

Studies in Computational Intelligence 1143

Hocine Cherifi
Luis M. Rocha
Chantal Cherifi
Murat Donduran *Editors*

Complex Networks & Their Applications XII

Proceedings of The Twelfth International
Conference on Complex Networks
and their Applications: COMPLEX
NETWORKS 2023 Volume 3

 Springer

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.

Hocine Cherifi · Luis M. Rocha ·
Chantal Cherifi · Murat Donduran
Editors

Complex Networks & Their Applications XII

Proceedings of The Twelfth International
Conference on Complex Networks and their
Applications: COMPLEX NETWORKS 2023
Volume 3

Editors

Hocine Cherifi 
University of Burgundy
Dijon Cedex, France

Chantal Cherifi
IUT Lumière - Université Lyon 2
University of Lyon
Bron, France

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

Murat Donduran
Department of Economics
Yildiz Technical University
Istanbul, Türkiye

ISSN 1860-949X

ISSN 1860-9503 (electronic)

Studies in Computational Intelligence

ISBN 978-3-031-53471-3

ISBN 978-3-031-53472-0 (eBook)

<https://doi.org/10.1007/978-3-031-53472-0>

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2024

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

Paper in this product is recyclable.

Preface

Dear Colleagues, Participants, and Readers,

We present the 12th Complex Networks Conference proceedings with great pleasure and enthusiasm. Like its predecessors, this edition proves complex network research's ever-growing significance and interdisciplinary nature. As we navigate the intricate web of connections that define our world, understanding complex systems, their emergent properties, and the underlying structures that govern them has become increasingly crucial.

The Complex Networks Conference has established itself as a pivotal platform for researchers, scholars, and experts from various fields to converge, exchange ideas, and push the boundaries of knowledge in this captivating domain. Over the past twelve years, we have witnessed remarkable progress, breakthroughs, and paradigm shifts highlighting the dynamic and complex tapestry of networks surrounding us, from biological systems and social interactions to technological infrastructures and economic networks.

This year's conference brought together an exceptional cohort of experts, including our keynote speakers:

- Michael Bronstein, University of Oxford, UK, enlightened us on “Physics-inspired Graph Neural Networks”
- Kathleen Carley, Carnegie Mellon University, USA, explored “Coupling in High Dimensional Networks”
- Manlio De Domenico, University of Padua, Italy, introduced “An Emerging Framework for the Functional Analysis of Complex Interconnected Systems”
- Danai Koutra, University of Michigan, USA, shared insights on “Advances in Graph Neural Networks: Heterophily and Beyond”
- Romualdo Pastor-Satorras, UPC, Spain, discussed “Opinion Depolarization in Interdependent Topics and the Effects of Heterogeneous Social Interactions”
- Tao Zhou, USTC, China, engaged us in “Recent Debates in Link Prediction”

These renowned experts addressed a spectrum of critical topics and the latest methodological advances, underscoring the continued expansion of this field into ever more domains.

We were also fortunate to benefit from the expertise of our tutorial speakers on November 27, 2023:

- Tiago de Paula Peixoto, CEU Vienna, Austria, guided “Network Inference and Reconstruction”
- Maria Liakata, Queen Mary University of London, UK, led us through “Longitudinal language processing from user-generated content”

We want to express our deepest gratitude to all the authors, presenters, reviewers, and attendees who have dedicated their time, expertise, and enthusiasm to make this event successful. The peer-review process, a cornerstone of scientific quality, ensures

that the papers in these proceedings have undergone rigorous evaluation, resulting in high-quality contributions.

We encourage you to explore the rich tapestry of knowledge and ideas as we dive into these four proceedings volumes. The papers presented here represent not only the diverse areas of research but also the collaborative and interdisciplinary spirit that defines the complex networks community.

In closing, we extend our heartfelt thanks to the organizing committees and volunteers who have worked tirelessly to make this conference a reality. We hope these proceedings inspire future research, innovation, and collaboration, ultimately helping us better understand the world's networks and their profound impacts on science, technology, and society.

We hope that the pleasure you have reading these papers matches our enthusiasm for organizing the conference and assembling this collection of articles.

Hocine Cherifi
Luis M. Rocha
Chantal Cherifi
Murat Donduran

Organization and Committees

General Chairs

Hocine Cherifi
Luis M. Rocha

University of Burgundy, France
Binghamton University, USA

Advisory Board

Jon Crowcroft
Raissa D'Souza
Eugene Stanley
Ben Y. Zhao

University of Cambridge, UK
Univ. of California, Davis, USA
Boston University, USA
University of Chicago, USA

Program Chairs

Chantal Cherifi
Murat Donduran

University of Lyon, France
Yildiz Technical University, Turkey

Lightning Chairs

Konstantin Avrachenkov
Mathieu Desroches
Huijuan Wang

Inria Université Côte d'Azur, France
Inria Université Côte d'Azur, France
TU Delft, Netherlands

Poster Chairs

Christophe Crespelle
Manuel Marques Pita
Laura Ricci

Université Côte d'Azur, France
Universidade Lusófona, Portugal
University of Pisa, Italy

Special Issues Chair

Sabrina Gaito University of Milan, Italy

Publicity Chairs

Fabian Braesemann University of Oxford, UK
Zachary Neal Michigan State University, USA
Xiangjie Kong Dalian University of Technology, China

Tutorial Chairs

Luca Maria Aiello Nokia-Bell Labs, UK
Leto Peel Maastricht University, Netherlands

Social Media Chair

Brennan Klein Northeastern University, USA

Sponsor Chairs

Roberto Interdonato CIRAD - UMR TETIS, France
Christophe Cruz University of Burgundy, France

Sustainability Chair

Madeleine Aurelle City School International De Ferney-Voltaire,
France

Local Committee Chair

Charlie Joyez Université Côte d'Azur, France

Publication Chair

Matteo Zignani University of Milan, Italy

Submission Chair

Cheick Ba Queen Mary University of London, UK

Web Chairs

Stephany Rajeh Sorbonne University, France
Alessia Galdeman University of Milan, Italy

Program Committee

Jacobo Aguirre	Centro de Astrobiología (CAB), Spain
Luca Maria Aiello	ITU Copenhagen, Denmark
Esra Akbas	Georgia State University, USA
Sinan G. Aksoy	Pacific Northwest National Laboratory, USA
Mehmet Aktas	Georgia State University, USA
Tatsuya Akutsu	Kyoto University, Japan
Reka Albert	Pennsylvania State University, USA
Alberto Aleta	University of Zaragoza, Spain
Claudio Altafini	Linköping University, Sweden
Viviana Amati	University of Milano-Bicocca, Unknown
Frederic Amblard	Université Toulouse 1 Capitole, IRIT, France
Enrico Amico	EPFL, Switzerland
Yuri Antonacci	University of Palermo, Italy
Alberto Antonioni	Carlos III University of Madrid, Spain
Nino Antulov-Fantulin	ETH Zurich, Switzerland
Mehrnaz Anvari	Fraunhofer SCAI, Germany
David Aparicio	Zendesck, Portugal
Nuno Araujo	Univ. de Lisboa, Portugal
Panos Argyrakis	Aristotle University of Thessaloniki, Greece
Oriol Artime	University of Barcelona, Spain
Malbor Asllani	Florida State University, USA
Tomaso Aste	University College London, UK
Martin Atzmueller	Osnabrück University & DFKI, Germany
Konstantin Avrachenkov	Inria Sophia-Antipolis, France

Giacomo Baggio	University of Padova, Italy
Franco Bagnoli	Università di Firenze, Italy
James Bagrow	University of Vermont, USA
Yiguang Bai	Xidian University, China
Sven Banisch	Karlsruhe Institute of Technology, Germany
Annalisa Barla	Università degli Studi di Genova, Italy
Nikita Basov	The University of Manchester, UK
Anais Baudot	CNRS, AMU, France
Gareth J. Baxter	University of Aveiro, Portugal
Loredana Bellantuono	University of Bari Aldo Moro, Italy
Andras Benczur	SZTAKI, Hungary
Rosa M. Benito	Universidad Politécnica de Madrid, Spain
Ginestra Bianconi	Queen Mary University of London, UK
Ofer Biham	The Hebrew University, Israel
Romain Billot	IMT Atlantique, France
Livio Bioglio	University of Turin, Italy
Hanjo D. Boekhout	Leiden University, Netherlands
Anthony Bonato	Toronto Metropolitan University, Canada
Anton Borg	Blekinge Institute of Technology, Sweden
Cecile Bothorel	IMT Atlantique, France
Federico Botta	University of Exeter, UK
Romain Bourqui	University of Bordeaux, France
Alexandre Bovet	University of Zurich, Switzerland
Dan Braha	New England Complex Systems Institute, USA
Ulrik Brandes	ETH Zürich, Switzerland
Rion Brattig Correia	Instituto Gulbenkian de Ciência, Portugal
Chico Camargo	University of Exeter, UK
Gian Maria Campedelli	Fondazione Bruno Kessler, Italy
M. Abdullah Canbaz	University at Albany SUNY, USA
Vincenza Carchiolo	DIEEI, Italy
Dino Carpentras	ETH Zürich, Switzerland
Giona Casiraghi	ETH Zürich, Switzerland
Douglas Castilho	Federal Inst. of South of Minas Gerais, Brazil
Costanza Catalano	University of Florence, Italy
Lucia Cavallaro	Free University of Bozen/Bolzano, Italy
Remy Cazabet	University of Lyon, France
Jianrui Chen	Shaanxi Normal University, China
Po-An Chen	National Yang Ming Chiao Tung Univ., Taiwan
Xihui Chen	University of Luxembourg, Luxembourg
Sang Chin	Boston University, USA
Daniela Cialfi	Institute for Complex Systems, Italy
Giulio Cimini	University of Rome Tor Vergata, Italy

Matteo Cinelli	Sapienza University of Rome, Italy
Salvatore Citraro	University of Pisa, Italy
Jonathan Clarke	Imperial College London, UK
Richard Clegg	QMUL, UK
Reuven Cohen	Bar-Ilan University, Israel
Jean-Paul Comet	Université Côte d'Azur, France
Marco Coraggio	Scuola Superiore Meridionale, Italy
Michele Coscia	ITU Copenhagen, Denmark
Christophe Crespelle	Université Côte d'Azur, France
Regino H. Criado Herrero	Universidad Rey Juan Carlos, Spain
Marcelo V. Cunha	Instituto Federal da Bahia, Brazil
David Soriano-Paños	Instituto Gulbenkian de Ciência, Portugal
Joern Davidsen	University of Calgary, Canada
Toby Davies	University of Leeds, UK
Caterina De Bacco	Max Planck Inst. for Intelligent Systems, Germany
Pietro De Lellis	University of Naples Federico II, Italy
Pasquale De Meo	University of Messina, Italy
Domenico De Stefano	University of Trieste, Italy
Fabrizio De Vico Fallani	Inria-ICM, France
Charo I. del Genio	Coventry University, UK
Robin Delabays	HES-SO, Switzerland
Yong Deng	Univ. of Electronic Science and Tech., China
Mathieu Desroches	Inria Centre at Université Côte d'Azur, France
Carl P. Dettmann	University of Bristol, UK
Zengru Di	Beijing Normal University, China
Riccardo Di Clemente	Northeastern University London, UK
Branco Di Fátima	University of Beira Interior (UBI), Portugal
Alessandro Di Stefano	Teesside University, UK
Ming Dong	Central China Normal University, China
Constantine Dovrolis	Georgia Tech, USA
Maximilien Dreveton	EPFL, Switzerland
Ahlem Drif	University of Setif, Algeria
Johan L. Dubbeldam	Delft University of Technology, Netherlands
Jordi Duch	Universitat Rovira i Virgili, Spain
Cesar Ducruet	CNRS, France
Mohammed El Hassouni	Mohammed V University in Rabat, Morocco
Frank Emmert-Streib	Tampere University, Finland
Gunes Ercal	Southern Illinois University Edwardsville, USA
Alejandro Espinosa-Rada	ETH Zürich, Switzerland
Alexandre Evsukoff	Universidade Federal do Rio de Janeiro, Brazil
Mauro Faccin	University of Bologna, Italy

Max Falkenberg	City University, UK
Guilherme Ferraz de Arruda	CENTAI Institute, Italy
Andrea Flori	Politecnico di Milano, Italy
Manuel Foerster	Bielefeld University, Germany
Emma Fraxanet Morales	Pompeu Fabra University, Spain
Angelo Furno	LICIT-ECO7, France
Sergio Gómez	Universitat Rovira i Virgili, Spain
Sabrina Gaito	Università degli Studi di Milano, Italy
José Manuel Galán	Universidad de Burgos, Spain
Alessandro Galeazzi	Ca' Foscari university of Venice, Italy
Lazaros K. Gallos	Rutgers University, USA
Joao Gama	INESC TEC—LIAAD, Portugal
Jianxi Gao	Rensselaer Polytechnic Institute, USA
David Garcia	University of Konstanz, Germany
Floriana Gargiulo	CNRS, France
Michael T. Gastner	Singapore Institute of Technology, Singapore
Alexander Gates	University of Virginia, USA
Alexandra M. Gerbasi	Exeter Business School, UK
Fakhteh Ghanbarnejad	Potsdam Inst. for Climate Impact Res., Germany
Cheol-Min Ghim	Ulsan National Inst. of Science and Tech., South Korea
Tommaso Gili	IMT School for Advanced Studies Lucca, Italy
Silvia Giordano	Univ. of Applied Sciences of Southern Switzerland, Switzerland
Rosalba Giugno	University of Verona, Italy
Kimberly Glass	Brigham and Women's Hospital, USA
David Gleich	Purdue University, USA
Antonia Godoy Lorite	UCL, UK
Kwang-Il Goh	Korea University, South Korea
Carlos Gracia	University of Zaragoza, Spain
Oscar M. Granados	Universidad Jorge Tadeo Lozano, Colombia
Michel Grossetti	CNRS, France
Guillaume Guerard	ESILV, France
Jean-Loup Guillaume	Université de la Rochelle, France
Furkan Gursoy	Bogazici University, Turkey
Philipp Hövel	Saarland University, Germany
Meesoon Ha	Chosun University, South Korea
Bianca H. Habermann	AMU, CNRS, IBDM UMR 7288, France
Chris Hankin	Imperial College London, UK
Yukio Hayashi	JAIST, Japan
Marina Hennig	Johannes Gutenberg University of Mainz, Germany

Takayuki Hiraoka	Aalto University, Finland
Marion Hoffman	Institute for Advanced Study in Toulouse, France
Bernie Hogan	University of Oxford, UK
Seok-Hee Hong	University of Sydney, Australia
Yujie Hu	University of Florida, USA
Flavio Iannelli	UZH, Switzerland
Yuichi Ikeda	Kyoto University, Japan
Roberto Interdonato	CIRAD, France
Antonio Iovanella	Univ. degli Studi Internazionali di Roma, Italy
Arkadiusz Jędrzejewski	CY Cergy Paris Université, France
Tao Jia	Southwest University, China
Jiaojiao Jiang	UNSW Sydney, Australia
Di Jin	University of Michigan, USA
Ivan Jokifá	Technology University of Delft, Netherlands
Charlie Joyez	GREDEG, Université Côte d'Azur, France
Bogumil Kamiński	SGH Warsaw School of Economics, Poland
Marton Karsai	Central European University, Austria
Eytan Katzav	Hebrew University of Jerusalem, Israel
Mehmet Kaya	Firat University, Turkey
Domokos Kelen	SZTAKI, Hungary
Mohammad Khansari	Sharif University of Technology, Iran
Jinseok Kim	University of Michigan, USA
Pan-Jun Kim	Hong Kong Baptist University, Hong Kong
Maksim Kitsak	TU Delft, Netherlands
Mikko Kivelä	Aalto University, Finland
Brennan Klein	Northeastern University, UK
Konstantin Klemm	IFISC (CSIC-UIB), Spain
Xiangjie Kong	Zhejiang University of Technology, China
Onerva Korhonen	University of Eastern Finland, Finland
Miklós Krész	InnoRenew CoE, Slovenia
Prosenjit Kundu	DA-IICT, Gandhinagar, Gujarat, India
Haewoon Kwak	Indiana University Bloomington, USA
Richard La	University of Maryland, USA
Josè Lages	Université de Franche-Comté, France
Renaud Lambiotte	University of Oxford, UK
Aniello Lampo	UC3M, Spain
Jennifer Larson	Vanderbilt University, USA
Paul J. Laurienti	Wake Forest, USA
Anna T. Lawniczak	University of Guelph, Canada
Deok-Sun Lee	KIAS, South Korea
Harlin Lee	Univ. of North Carolina at Chapel Hill, USA
Juergen Lerner	University of Konstanz, Germany

Lasse Leskelä	Aalto University, Finland
Petri Leskinen	Aalto University/SeCo, Finland
Inmaculada Leyva	Universidad Rey Juan Carlos, Spain
Cong Li	Fudan University, China
Longjie Li	Lanzhou University, China
Ruiqi Li	Beijing Univ. of Chemical Technology, China
Xiangtao Li	Jilin University, China
Hao Liao	Shenzhen University, China
Fabrizio Lillo	Università di Bologna, Italy
Giacomo Livan	University of Pavia, Italy
Giosue' Lo Bosco	Università di Palermo, Italy
Hao Long	Jiangxi Normal University, China
Juan Carlos Losada	Universidad Politécnica de Madrid, Spain
Laura Lotero	Universidad Nacional de Colombia, Colombia
Yang Lou	National Yang Ming Chiao Tung Univ., Taiwan
Meilian Lu	Beijing Univ. of Posts and Telecom., China
Maxime Lucas	CENTAI, Italy
Lorenzo Lucchini	Bocconi University, Italy
Hanbaek Lyu	UW-Madison, USA
Vince Lyzinski	University of Maryland, College Park, USA
Morten Mørup	Technical University of Denmark, Denmark
Leonardo Maccari	Ca'Foscari University of Venice, Italy
Matteo Magnani	Uppsala University, Sweden
Maria Malek	CY Cergy Paris University, France
Giuseppe Mangioni	University of Catania, Italy
Andrea Mannocci	CNR-ISTI, Italy
Rosario N. Mantegna	University of Palermo, Italy
Manuel Sebastian Mariani	University of Zurich, Switzerland
Radek Marik	CTU in Prague, Czech Republic
Daniele Marinazzo	Ghent University, Belgium
Andrea Marino	University of Florence, Italy
Malvina Marku	INSERM, CRCT, France
Antonio G. Marques	King Juan Carlos University, Spain
Christoph Martin	Hamburg University of Applied Sciences, Germany
Samuel Martin-Gutierrez	Complexity Science Hub Vienna, Austria
Cristina Masoller	Universitat Politecnica de Catalunya, Spain
Rossana Mastrandrea	IMT School for Advanced Studies, Italy
John D. Matta	Southern Illinois Univ. Edwardsville, USA
Carolina Mattsson	CENTAI Institute, Italy
Fintan McGee	Luxembourg IST, Luxembourg
Matus Medo	University of Bern, Switzerland

Ronaldo Menezes	University of Exeter, UK
Humphrey Mensah	Epsilon Data Management, LLC, USA
Anke Meyer-Baese	Florida State University, USA
Salvatore Micciche	UNIPA DiFC, Italy
Letizia Milli	University of Pisa, Italy
Marija Mitrovic	Institute of Physics Belgrade, Serbia
Andrzej Mizera	University of Warsaw, Poland
Chiara Mocenni	University of Siena, Italy
Roland Molontay	Budapest UTE, Hungary
Sifat Afroj Moon	University of Virginia, USA
Alfredo Morales	MIT, USA
Andres Moreira	UTFSM, Chile
Greg Morrison	University of Houston, USA
Igor Mozetic	Jozef Stefan Institute, Slovenia
Sarah Muldoon	State University of New York, Buffalo, USA
Tsuyoshi Murata	Tokyo Institute of Technology, Japan
Jose Nacher	Toho University, Japan
Nishit Narang	NIT Delhi, India
Filipi Nascimento Silva	Indiana University, USA
Muaz A. Niazi	National Univ. of Science & Technology, Pakistan
Peter Niemeyer	Leuphana University Lueneburg, Germany
Jordi Nin	ESADE, Universitat Ramon Llull, Spain
Rogier Noldus	Ericsson, Netherlands
Masaki Ogura	Osaka University, Japan
Andrea Omicini	Università di Bologna, Italy
Gergely Palla	Eötvös University, Hungary
Fragkiskos Papadopoulos	Cyprus University of Technology, Cyprus
Symeon Papadopoulos	Centre for Research & Technology, Greece
Alice Patania	University of Vermont, USA
Leto Peel	Maastricht University, Netherlands
Hernane B. B. Pereira	Senai Cimatec, Brazil
Josep Perelló	Universitat de Barcelona, Spain
Anthony Perez	Université d'Orléans, France
Juergen Pfeffer	Technical University of Munich, Germany
Carlo Piccardi	Politecnico di Milano, Italy
Pietro Hiram Guzzi	Univ. Magna Gracia of Catanzaro, Italy
Yoann Pigné	Université Le Havre Normandie, France
Bruno Pinaud	University of Bordeaux, France
Flavio L. Pinheiro	Universidade Nova de Lisboa, Portugal
Manuel Pita	Universidade Lusófona, Portugal
Clara Pizzuti	CNR-ICAR, Italy

Jan Platos	VSB - Technical University of Ostrava, Czech Republic
Pawel Pralat	Toronto Metropolitan University, Canada
Rafael Prieto-Curiel	Complexity Science Hub, Austria
Daniele Proverbio	University of Trento, Italy
Giulia Pullano	Georgetown University, USA
Rami Puzis	Ben-Gurion University of the Negev, Israel
Christian Quadri	Università degli Studi di Milano, Italy
Hamid R. Rabiee	Sharif University of Technology, Iran
Filippo Radicchi	Indiana University, USA
Giancarlo Ragozini	University of Naples Federico II, Italy
Juste Raimbault	IGN-ENSG, France
Sarah Rajtmajer	Penn State, USA
Gesine D. Reinert	University of Oxford, UK
Élisabeth Remy	Institut de Mathématiques de Marseille, France
Xiao-Long Ren	Univ. of Electronic Science and Tech., China
Laura Ricci	University of Pisa, Italy
Albano Rikani	INSERM, France
Luis M. Rocha	Binghamton University, USA
Luis E. C. Rocha	Ghent University, Belgium
Fernando E. Rosas	Imperial College London, UK
Giulio Rossetti	CNR-ISTI, Italy
Camille Roth	CNRS/CMB/EHESS, France
Celine Rozenblat	UNIL, Switzerland
Giancarlo Ruffo	Univ. degli Studi del Piemonte Orientale, Italy
Arnaud Sallaberry	University of Montpellier, France
Hillel Sanhedrai	Northeastern University, USA
Iraj Saniee	Bell Labs, Nokia, USA
Antonio Scala	CNR Institute for Complex Systems, Italy
Michael T. Schaub	RWTH Aachen University, Germany
Irene Sendiña-Nadal	Universidad Rey Juan Carlos, Spain
Mattia Sensi	Politecnico di Torino, Italy
Ke-ke Shang	Nanjing University, China
Julian Sienkiewicz	Warsaw University of Technology, Poland
Per Sebastian Skardal	Trinity College, Ireland
Fiona Skerman	Uppsala University, Sweden
Oskar Skibski	University of Warsaw, Poland
Keith M. Smith	University of Strathclyde, UK
Igor Smolyarenko	Brunel University, UK
Zbigniew Smoreda	Orange Innovation, France
Annalisa Socievole	ICAR-CNR, Italy
Igor M. Sokolov	Humboldt University Berlin, Germany

Albert Solé-Ribalta	Universitat Oberta de Catalunya, Spain
Sara Sottile	University of Trento, Italy
Sucheta Soundarajan	Syracuse University, USA
Jaya Sreevalsan-Nair	IIIT Bangalore, India
Christoph Stadtfeld	ETH Zürich, Switzerland
Clara Stegehuis	University of Twente, Netherlands
Lovro Šubelj	University of Ljubljana, Slovenia
Xiaoqian Sun	Beihang University, China
Michael Szell	IT University of Copenhagen, Denmark
Boleslaw Szymanski	Rensselaer Polytechnic Institute, USA
Andrea Tagarelli	University of Calabria, Italy
Kazuhiro Takemoto	Kyushu Institute of Technology, Japan
Frank W. Takes	Leiden University, Netherlands
Fabien Tarissan	CNRS & ENS Paris-Saclay, France
Laura Temime	Cnam, France
François Théberge	TIMC, France
Guy Theraulaz	Université Paul Sabatier and CNRS, France
I-Hsien Ting	National University of Kaohsiung, Taiwan
Michele Tizzani	ISI Foundation, Italy
Michele Tizzoni	University of Trento, Italy
Olivier Togni	University of Burgundy, France
Leo Torres	Northeastern University, USA
Sho Tsugawa	University of Tsukuba, Japan
Francesco Tudisco	The University of Edinburgh, UK
Melvyn S. Tyloo	Los Alamos National Lab, USA
Stephen M. Uzzo	National Museum of Mathematics, USA
Lucas D. Valdez	IFIMAR-UNMdP, Argentina
Pim Van der Hoorn	Eindhoven University of Technology, Netherlands
Piet Van Mieghem	Delft University of Technology, Netherlands
Fabio Vanni	University of Insubria, Italy
Christian L. Vestergaard	Institut Pasteur, France
Tiphaine Viard	Télécom Paris, France
Julian Vicens	Eurecat, Spain
Blai Vidiella	CSIC, Spain
Pablo Villegas	Enrico Fermi Research Center (CREAF), Italy
Maria Prosperina Vitale	University of Salerno, Italy
Pierpaolo Vivo	King's College London, UK
Johannes Wachs	Corvinus University of Budapest, Hungary
Huijuan Wang	Delft University of Technology, Netherlands
Lei Wang	Beihang University, China
Guanghui Wen	Southeast University, Nanjing, China
Mateusz Wilinski	Los Alamos National Laboratory, USA

Dirk Witthaut	Forschungszentrum Jülich, Germany
Bin Wu	Beijing Univ. of Posts and Telecom., China
Mincheng Wu	Zhejiang University of Technology, China
Tao Wu	Chongqing Univ. of Posts and Telecom., China
Haoxiang Xia	Dalian University of Technology, China
Gaoxi Xiao	Nanyang Technological University, Singapore
Nenggang Xie	Anhui University of Technology, China
Takahiro Yabe	MIT, USA
Kaicheng Yang	Northeastern University, USA
Yian Yin	Cornell University, USA
Jean-Gabriel Young	University of Vermont, USA
Irfan Yousuf	Univ. of Engineering and Technology, Pakistan
Yongguang Yu	Beijing Jiaotong University, China
Paolo Zeppini	University Cote d'Azur, France
Shi Zhou	University College London (UCL), UK
Wei-Xing Zhou	East China Univ. of Science and Techno., China
Eugenio Zimeo	University of Sannio, Italy
Lorenzo Zino	Politecnico di Torino, Italy
Michal R. Zochowski	University of Michigan, USA
Claudia Zucca	Tilburg University, Netherlands

Contents

Multilayer/Multiplex

Eigenvector Centrality for Multilayer Networks with Dependent Node Importance	3
<i>Hildreth Robert Frost</i>	
Identifying Contextualized Focal Structures in Multisource Social Networks by Leveraging Knowledge Graphs	15
<i>Abiola Akinnubi, Mustafa Alassad, Nitin Agarwal, and Ridwan Amure</i>	
How Information Spreads Through Multi-layer Networks: A Case Study of Rural Uganda	28
<i>Jennifer M. Larson and Janet I. Lewis</i>	
Classification of Following Intentions Using Multi-layer Motif Analysis of Communication Density and Symmetry Among Users	37
<i>Takayasu Fushimi and Takumi Miyazaki</i>	
Generalized Densest Subgraph in Multiplex Networks	49
<i>Ali Behrouz and Farnoosh Hashemi</i>	
Influence Robustness of Nodes in Multiplex Networks Against Attacks	62
<i>Boqian Ma, Hao Ren, and Jiaojiao Jiang</i>	
Efficient Complex Network Representation Using Prime Numbers	75
<i>Konstantinos Bougiatiotis and Georgios Paliouras</i>	

Network Analysis

Approximation Algorithms for k-Median Problems on Complex Networks: Theory and Practice	89
<i>Roldan Pozo</i>	
Score and Rank Semi-monotonicity for Closeness, Betweenness and Harmonic Centrality	102
<i>Paolo Boldi, Davide D'Ascenzo, Flavio Furia, and Sebastiano Vigna</i>	
Non Parametric Differential Network Analysis for Biological Data	114
<i>Pietro Hiram Guzzi, Arkaprava Roy, Francesca Cortese, and Pierangelo Veltri</i>	

Bowlership: Examining the Existence of Bowler Synergies in Cricket 124
Praharsh Nanavati and Amit Anil Nanavati

A Correction to the Heuristic Algorithm MinimalFlipSet to Balance Unbalanced Graphs 134
Sukhamay Kundu and Amit A. Nanavati

Influential Node Detection on Graph on Event Sequence 147
Zehao Lu, Shihan Wang, Xiao-Long Ren, Rodrigo Costas, and Tamara Metze

Decentralized Control Methods in Hypergraph Distributed Optimization 159
Ioannis Papastaikoudis and Ioannis Lestas

Topic-Based Analysis of Structural Transitions of Temporal Hypergraphs Derived from Recipe Sharing Sites 171
Keisuke Uga, Masahito Kumano, and Masahiro Kimura

I Like You if You Are Like Me: How the Italians’ Opinion on Twitter About Migrants Changed After the 2022 Russo-Ukrainian Conflict 183
Giulio Cordova, Luca Palla, Martina Sustrico, and Giulio Rossetti

Modeling the Association Between Physician Risky-Prescribing and the Complex Network Structure of Physician Shared-Patient Relationships 194
Xin Ran, Ellen R. Meara, Nancy E. Morden, Erika L. Moen, Daniel N. Rockmore, and A. James O’Malley

Focal Structures Behavior in Dynamic Social Networks 208
Mustafa Alassad and Nitin Agarwal

Unified Logic Maze Generation Using Network Science 222
Johnathon Henke and Dinesh Mehta

INDoRI: Indian Dataset of Recipes and Ingredients and Its Ingredient Network 234
Sandeep Khanna, Chiranjoy Chattopadhyay, and Suman Kundu

Optimizing Neonatal Respiratory Support Through Network Modeling: A New Approach to Post-birth Infant Care 245
Yassine Sebahi, Fakhra Jabeen, Jan Treur, H. Rob Taal, and Peter H. M. P. Roelofsma

Generalized Gromov Wasserstein Distance for Seed-Informed Network Alignment 258
Mengzhen Li and Mehmet Koyutürk

Orderliness of Navigation Patterns in Hyperbolic Complex Networks 271
Dániel Ficzer, Gergely Hollósi, Attila Frankó, Pál Varga, and József Biró

Multiplex Financial Network Regionalization Scenarios as a Result of Re-globalization: Does Geographical Proximity Still Matter? 283
Otilija Jurakovaite and Asta Gaigaliene

A Modular Network Exploration of Backbone Extraction Techniques 296
Ali Yassin, Hocine Cherifi, Hamida Seba, and Olivier Togni

IS-PEW: Identifying Influential Spreaders Using Potential Edge Weight in Complex Networks 309
Suman Nandi, Mariana Curado Malta, Giridhar Maji, and Animesh Dutta

Robustness of Centrality Measures Under Incomplete Data 321
Natalia Meshcheryakova and Sergey Shvydun

ATEM: A Topic Evolution Model for the Detection of Emerging Topics in Scientific Archives 332
Hamed Rahimi, Hubert Naacke, Camelia Constantin, and Bernd Amann

Analysis and Characterization of ERC-20 Token Network Topologies 344
Matteo Loporchio, Damiano Di Francesco Maesa, Anna Bernasconi, and Laura Ricci

Network Geometry

Modeling the Invisible Internet 359
Jacques Bou Abdo and Liaquat Hossain

Modeling the Dynamics of Bitcoin Overlay Network 371
Jacques Bou Abdo, Shivalaxmi Dass, Basheer Qolomany, and Liaquat Hossain

Graph Based Approach for Galaxy Filament Extraction 384
Louis Hauseux, Konstantin Avrachenkov, and Josiane Zerubia

Metric Invariants for Networks' Classification 397
Eldad Kronfeld and Emil Saucan

The Hidden-Degree Geometric Block Model	409
<i>Stefano Guarino, Enrico Mastrostefano, and Davide Torre</i>	
Networks in Finance and Economics	
Interactions Within Complex Economic System	423
<i>Daniela Cialfi</i>	
Demand Shocks and Export Surges in Trade Networks	435
<i>John Schoeneman, Marten Brienen, Lixia Lambert, Dayton Lambert, and Violet Rebek</i>	
Properties of B2B Invoice Graphs and Detection of Structures	444
<i>Joannès Guichon, Nazim Fatès, Sylvain Contassot-Vivier, and Massimo Amato</i>	
A Model and Structural Analysis of Networked Bitcoin Transaction Flows	456
<i>Min-Hsueh Chiu and Mayank Kejriwal</i>	
Rank Is All You Need: Robust Estimation of Complex Causal Networks	468
<i>Cameron Cornell, Lewis Mitchell, and Matthew Roughan</i>	
Author Index	483

Multilayer/Multiplex



Eigenvector Centrality for Multilayer Networks with Dependent Node Importance

Hildreth Robert Frost^(✉)

Dartmouth College, Hanover, NH 03755, USA
rob.frost@dartmouth.edu

Abstract. We present a novel approach for computing a variant of eigenvector centrality for multilayer networks with inter-layer constraints on node importance. Specifically, we consider a multilayer network defined by multiple edge-weighted, potentially directed, graphs over the same set of nodes with each graph representing one layer of the network and no inter-layer edges. As in the standard eigenvector centrality construction, the importance of each node in a given layer is based on the weighted sum of the importance of adjacent nodes in that same layer. Unlike standard eigenvector centrality, we assume that the adjacency relationship and the importance of adjacent nodes may be based on distinct layers. Importantly, this type of centrality constraint is only partially supported by existing frameworks for multilayer eigenvector centrality that use edges between nodes in different layers to capture inter-layer dependencies. For our model, constrained, layer-specific eigenvector centrality values are defined by a system of independent eigenvalue problems and dependent pseudo-eigenvalue problems, whose solution can be efficiently realized using an interleaved power iteration algorithm. An R package implementing this method along with example vignettes is available at <https://hrfrost.host.dartmouth.edu/CMLC/>.

Keywords: eigenvector centrality · multilayer networks

1 Eigenvector Centrality for Multilayer Networks

Computation of node importance via centrality measures is an important task in network analysis and a large number of centrality measures have been developed that prioritize different node and network properties [7]. The most widely used centrality measures are a function of the network adjacency matrix, \mathbf{A} , which, for an edge-weighted network defined over p nodes is the $p \times p$ matrix:

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,p} \\ \vdots & \ddots & \vdots \\ a_{p,1} & \cdots & a_{p,p} \end{bmatrix} \quad (1)$$

where $a_{i,j}$ captures the weight of the edge between nodes i and j or 0 if no edge exists between these nodes. Self-edges are represented by elements on the

diagonal. If the network is directed, then $a_{i,j}$ and $a_{j,i}$ capture distinct edges and \mathbf{A} is asymmetric; if the network is undirected, $a_{i,j} = a_{j,i}$ and \mathbf{A} is symmetric.

Modeling node importance as the weighted sum of the importance of adjacent nodes leads a version of centrality called eigenvector centrality, which is solved by computing the principal eigenvector of the following eigenvalue problem:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \quad (2)$$

Specifically, the eigenvector centrality for node n is given by element n of the principal eigenvector \mathbf{x} corresponding to the largest eigenvalue [7]. When \mathbf{A} is irreducible (i.e., the network is strongly connected), then the Perron-Frobenius theorem [8] guarantees that there is a unique largest real eigenvalue whose corresponding eigenvector can be chosen to have strictly positive elements. For directed graphs, left and right versions of eigenvector centrality are possible, i.e., the solution to the eigenvalue problem for \mathbf{A}^T or \mathbf{A} . For the methods developed below, we focus on the right eigenvector centrality, however, the same approach can be employed to compute left eigenvector centrality by considering \mathbf{A}^T instead of \mathbf{A} .

In this paper, we are interested in eigenvector centrality and how that measure of node importance generalizes to multilayer (or multiplex) networks [1, 5]. We assume a multilayer network comprised by k layers that each represent a potentially directed, edge-weighted graph over the same p nodes. The graph for layer $j \in \{1, \dots, k\}$ can be represented by the adjacency matrix \mathbf{A}_j :

$$\mathbf{A}_j = \begin{bmatrix} a_{j,1,1} & \cdots & a_{j,1,p} \\ \vdots & \ddots & \vdots \\ a_{j,p,1} & \cdots & a_{j,p,p} \end{bmatrix} \quad (3)$$

where $a_{j,n,m}$ holds the weight of the edge from node n to node m within the layer j graph. Although the terms network and graph are synonymous in this context, we will generally use the term network to refer to the entire multilayer network and the term graph to refer to the network that defines a single layer. In the context of a multilayer network, node eigenvector centrality can be evaluated at the level of a specific layer (i.e., a given node has separate centrality values for each of the k layers) or at the level of the entire multilayer network (i.e., a given node has a single centrality value that captures the importance of the node across all k layers). In the development below, we focus on layer-specific measures of eigenvector centrality.

If the k layers are independent, then eigenvector centrality can simply be computed separately for each layer. However, if dependencies exist between the layers, then a multilayer version of eigenvector centrality must be employed that can account for the inter-layer constraints. A number of approaches for modeling and computing multilayer eigenvector centrality have been explored over the last decade (e.g., [3, 4, 9–11]). Most of these approaches assume that inter-layer constraints can be modeled by edges between the nodes in one layer and nodes in other layers. This type of approach is exemplified by the recent work of Taylor

et al. [10] that details a flexible model for a "uniformly and diagonally coupled multiplex network". Specifically, Taylor et al. represent inter-layer dependencies by equally weighted edges connecting the nodes in one layer to the same nodes in a dependent layer. Taylor et al. represent the structure of these dependencies using a $k \times k$ adjacency matrix $\tilde{\mathbf{A}}$:

$$\tilde{\mathbf{A}} = \begin{bmatrix} \tilde{a}_{1,1} & \cdots & \tilde{a}_{1,k} \\ \vdots & \ddots & \vdots \\ \tilde{a}_{k,k} & \cdots & \tilde{a}_{k,k} \end{bmatrix} \quad (4)$$

where $\tilde{a}_{i,j}$ represents the weight of the edges from nodes in layer i to nodes in layer j . Computation of multilayer eigenvector centralities is then based on the principal eigenvector of a $kp \times kp$ supercentrality matrix $\mathbb{C}(\omega)$:

$$\mathbb{C}(\omega) = \hat{\mathbb{C}} + \omega \hat{\mathbb{A}} \quad (5)$$

where $\hat{\mathbb{C}} = \text{diag}[\mathbf{A}_1, \dots, \mathbf{A}_k]$ (i.e., a $kp \times kp$ block diagonal matrix that has the adjacency matrices for each of the k layers along the diagonal), $\hat{\mathbb{A}} = \tilde{\mathbf{A}} \otimes \mathbf{I}$ (i.e., the Kronecker product of $\tilde{\mathbf{A}}$ and \mathbf{I}), and ω is the coupling strength. The principal eigenvector of $\mathbb{C}(\omega)$ can then be used to find joint, marginal, and conditional eigenvector centralities. Specifically, the principal eigenvector elements are divided into k sequential blocks of p elements, with the block corresponding to layer i representing the joint centrality values for the nodes in layer i . To calculate the marginal centralities for either nodes or layers, the joint centralities are summed over all layers for a given node or all nodes for a given layer. To calculate conditional centralities for either nodes or layers, the joint centrality value for a given node/layer pair is divided by either the marginal centrality for the layer or the marginal centrality for the node.

2 Eigenvector Centrality for Multilayer Networks with Inter-layer Constraints on Adjacent Node Importance

Our model for multilayer eigenvector centrality extends the standard single network version given by (2) to support the scenario where the importance of a given node in layer i is proportional to the weighted sum of the importance of adjacent nodes with adjacency and weights based on layer i but adjacent node importance based on potentially distinct layers. This model is conceptually and mathematically distinct from approaches like Taylor et al. that add edges between the same node in different layers to capture inter-layer dependencies. To illustrate, assume we have a multilayer network with just two layers i and j . If we assume the importance for nodes in layer i has the standard definition, i.e., it is not dependent on another layer, the solution is given by the typical eigenvalue problem:

$$\mathbf{A}_i \mathbf{x}_i = \lambda_i \mathbf{x}_i \quad (6)$$

However, if we assume the importance for nodes in layer j is based on the importance of adjacent nodes in layer i , then solution for layer j is given by the following linear model (note that both \mathbf{x}_i and \mathbf{x}_j are included):

$$\mathbf{A}_j \mathbf{x}_i = \lambda_j \mathbf{x}_j \quad (7)$$

Although the linear model (7) is not technically an eigenvalue problem since it contains distinct \mathbf{x}_i and \mathbf{x}_j vectors, we will refer to it as a pseudo-eigenvalue problem given the structural similarities to (6) and the fact that \mathbf{x}_i may represent an eigenvector. As detailed below, we will also use the pseudo-eigenvalue label to describe more complex scenarios where \mathbf{x}_i found on both sides of the equation. For this example, the solution for the entire multilayer network is given by a system of an independent eigenvalue problem and a dependent pseudo-eigenvalue problem:

$$\begin{aligned} \mathbf{A}_i \mathbf{x}_i &= \lambda_i \mathbf{x}_i \\ \mathbf{A}_j \mathbf{x}_i &= \lambda_j \mathbf{x}_j \end{aligned} \quad (8)$$

In this case, the solution can be obtained by first solving the eigenvalue problem for layer i to find \mathbf{x}_i and then computing \mathbf{x}_j as $\mathbf{x}_j = 1/\lambda_j \mathbf{A}_j \mathbf{x}_i$ with the value of λ_j set to ensure \mathbf{x}_j is unit length.

This simple example can be generalized to a multilayer network with k layers and arbitrary node importance constraints encoded by a graph whose nodes represent layers and whose weighted and directed edges represent inter-layer dependencies. Let this inter-layer dependency graph be represented by the $k \times k$ adjacency matrix $\tilde{\mathbf{A}}$ that is similar in structure to the $\tilde{\mathbf{A}}$ used by Taylor et al. and defined in (4) but with the added constraint that the rows must sum to 1 (i.e., $\forall_{i \in 1, \dots, k} \sum_{j=1}^k \tilde{a}_{i,j} = 1$). Element $\tilde{a}_{i,j}$ of $\tilde{\mathbf{A}}$ represents the strength of the dependency between adjacent node importance in layer i and node importance in layer j with the sum of all dependencies for a given layer equal to 1. If $\tilde{\mathbf{A}} = \mathbf{I}$, all of the layers are independent. For the 2 layer example represented by (8), $\tilde{\mathbf{A}}$ is:

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \quad (9)$$

If the length p vector \mathbf{x}_i represents node importance in layer i , we can define an adjacent node importance function $\mathbf{c}(i, \tilde{\mathbf{A}})$ as:

$$\mathbf{c}(i, \tilde{\mathbf{A}}) = \sum_{j=1}^k \tilde{a}_{i,j} \mathbf{x}_j \quad (10)$$

In other words, the importance of nodes in layer i is based on a weighted sum of the importance of adjacent nodes in other layers (note that adjacency is only based on the topology of layer i). Given the function $\mathbf{c}()$, we can compute a constrained multilayer version of eigenvector centrality for the network by solving the following system of k interdependent eigenvalue and pseudo-eigenvalue problems:

$$\begin{aligned}
\mathbf{A}_1 \mathbf{c}(1, \tilde{\mathbf{A}}) &= \lambda_1 \mathbf{x}_1 \\
\mathbf{A}_2 \mathbf{c}(2, \tilde{\mathbf{A}}) &= \lambda_2 \mathbf{x}_2 \\
&\vdots \\
\mathbf{A}_k \mathbf{c}(k, \tilde{\mathbf{A}}) &= \lambda_k \mathbf{x}_k
\end{aligned} \tag{11}$$

Importantly, the supercentrality approach of Taylor et al. cannot in general solve systems such as (8), i.e., system (11) cannot be directly mapped to an eigenvalue problem involving a supercentrality matrix of the form defined by (5).

In the special case where $\tilde{\mathbf{A}} = \mathbf{I}$, all $\mathbf{c}(i, \tilde{\mathbf{A}}) = \mathbf{x}_i$ and (11) becomes a system of k independent eigenvalue problems:

$$\begin{aligned}
\mathbf{A}_1 \mathbf{x}_1 &= \lambda_1 \mathbf{x}_1 \\
\mathbf{A}_2 \mathbf{x}_2 &= \lambda_2 \mathbf{x}_2 \\
&\vdots \\
\mathbf{A}_k \mathbf{x}_k &= \lambda_k \mathbf{x}_k
\end{aligned} \tag{12}$$

More generally, the dependency structure for a given layer i falls into one of three cases:

- A $\tilde{a}_{i,i} = 1$: In this scenario, the eigenvector centrality for layer i is given by the principal eigenvector of the independent eigenvalue problem $\mathbf{A}_i \mathbf{x}_i = \lambda_i \mathbf{x}_i$.
- B $\tilde{a}_{i,i} = 0$: In this scenario, the centrality for layer i is a linear function of the centrality values of other network layers.
- C $0 < \tilde{a}_{i,i} < 1$: In this scenario, the centrality for layer i is given by a pseudo-eigenvalue problem that can be rewritten as $\mathbf{A}_i \mathbf{x}_i + \mathbf{d} = \lambda_i \mathbf{x}_i$, where \mathbf{d} captures the part of $\mathbf{A}_i \mathbf{c}(i, \tilde{\mathbf{A}})$ not due to \mathbf{x}_i .

If all layers in the multilayer network fall into case A or B and no cycles exist in the graph defined by $\tilde{\mathbf{A}}$, then the constrained eigenvector centralities can be computed using a relatively straightforward two-step procedure:

1. Solve the independent eigenvalue problems for all layers in case A using an algorithm like power iteration [6].
2. Sequentially solve the linear models for all layers in case B with the order of solution given by the inter-layer constraints.

If any layers fall into case C or cycles exist in the inter-layer dependency graph, then the solution must be obtained via an iterative algorithm similar to the interleaved power iteration approach detailed in Sect. 3 as Algorithm 1.

3 Interleaved Power Iteration Algorithm for a System of Dependent Pseudo-eigenvalue Problems

For an arbitrary inter-layer dependency matrix $\tilde{\mathbf{A}}$, the joint solution for system (11) can be found via a interleaved version of the power iteration method,

detailed in Algorithm 1, that is applied across all k linear problems. It should be noted that this specification of Algorithm 1 does not include features important for many practical implementations, e.g, checks to ensure the input matrices \mathbf{X}_i are well conditioned, options for the use of stochastic initialization of the eigenvectors, use of techniques like accelerated stochastic power iteration [12] to improve computational performance, alternate stopping conditions, etc. The CMLC (Constrained Multilayer Centrality) R package implementing this algorithm along with example vignettes (that generate the results shown in Sects. 4 and 5) is available at <https://hrfrost.host.dartmouth.edu/CMLC/>.

Algorithm 1. Interleaved power iteration for dependent pseudo-eigenvalue problems

Input:

- Set of k $p \times p$ irreducible matrices, $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_k\}$
- Dependencies between the principal pseudo-eigenvectors of \mathbf{X}_i encoded as a $k \times k$ matrix $\tilde{\mathbf{A}}$ whose rows sum to 1 (see (11)) The pseudo-eigenvalue problem for \mathbf{X}_i is given by $\mathbf{X}_i \mathbf{c}(i, \tilde{\mathbf{A}}) = \lambda_i \mathbf{x}_i$, where $\mathbf{c}(i, \tilde{\mathbf{A}}) = \sum_{j=1}^k \tilde{a}_{i,j} \mathbf{x}_j$
- Positive integer *maxIter* that represents the maximum number of iterations
- Positive real number *tol* that represents the stopping criteria as the proportional change in the mean of the k pseudo-eigenvalues between iterations

Output:

- Estimated principal pseudo-eigenvectors of the input \mathbf{X}_i : $\{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \dots, \hat{\mathbf{v}}_k\}$
- Estimated principal pseudo-eigenvalues of the input \mathbf{X}_i : $\{\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_k\}$
- Number of iterations completed

Notation:

- Let $\mathbf{v}_{n,m}$ and $\lambda_{n,m}$ represent the principal pseudo-eigenvector and pseudo-eigenvalue for matrix \mathbf{X}_n as computed on the m^{th} iteration of the algorithm
- 1: $\forall_{j \in 1 \dots k} \mathbf{v}_{j,0} = \{1/\sqrt{p}, \dots, 1/\sqrt{p}\}$ \triangleright Initialize principal pseudo-eigenvectors to unit length vectors with all values equal to $1/\sqrt{p}$
 - 2: **for** $i \in \{1, \dots, \text{maxIter}\}$ **do**
 - 3: $\forall_{j \in 1 \dots k} \mathbf{v}_{j,i} = \mathbf{X}_j \mathbf{c}(i, \tilde{\mathbf{A}})$ \triangleright Update principal pseudo-eigenvectors based on dependencies
 - 4: $\forall_{j \in 1 \dots k} \mathbf{v}_{j,i} = \mathbf{v}_{j,i} / \|\mathbf{v}_{j,i}\|$ \triangleright Normalize pseudo-eigenvectors to unit length
 - 5: $\forall_{j \in 1 \dots k} \lambda_{j,i} = (\mathbf{v}_{j,i})^T \mathbf{X}_j \mathbf{c}(i, \tilde{\mathbf{A}})$ \triangleright Update principal pseudo-eigenvalues
 - 6: **if** $i > 1$ **then**
 - 7: $\Delta = (1/k \sum_{j=1}^k |\lambda_{j,i-1} - \lambda_{j,i}|) / (1/k \sum_{j=1}^k \lambda_{j,i})$ \triangleright Compute proportional change in mean pseudo-eigenvalue
 - 8: **if** $\Delta < \text{tol}$ **then**
 - 9: **break** \triangleright If proportion change is less than *tol*, exit
- return** $\{\mathbf{v}_{1,i}, \mathbf{v}_{2,i}, \dots, \mathbf{v}_{k,i}\}, \{\lambda_{1,i}, \lambda_{2,i}, \dots, \lambda_{k,i}\}, i$
-

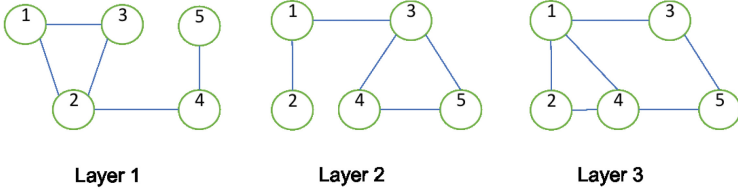


Fig. 1. Example undirected and unweighted multilayer network.

4 Simple Example

To illustrate the constrained multilayer model detailed in Sect. 2 and the performance of the interleaved power iteration algorithm detailed in Sect. 3, we consider a simple multilayer network comprised by three layers that each define an undirected and non-weighted network with five nodes. The structure of this multilayer network is shown in Fig. 1. For this example network, the symmetric adjacency matrices for the three layers are given by:

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \mathbf{A}_3 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

We consider four different inter-layer dependency scenarios:

1. No dependencies

If no dependencies exist between the layers (i.e., $\tilde{\mathbf{A}} = \mathbf{I}$), the eigenvector centralities (rounded to two decimal places) for each layer are:

$$\begin{aligned} \mathbf{v}_1 &= \{0.50, 0.60, 0.50, 0.34, 0.15\} \\ \mathbf{v}_2 &= \{0.34, 0.15, 0.60, 0.50, 0.50\} \\ \mathbf{v}_3 &= \{0.53, 0.43, 0.36, 0.53, 0.36\} \end{aligned}$$

As expected given the structure of layer 1, node 2 has the largest eigenvector centrality, followed by nodes 1 and 3 with node 5 having the lowest. Similarly for layer 2, node 3 has the largest centrality, followed by nodes 4 and 5 with node 2 having the lowest centrality. For layer 3, nodes 1 and 4 are tied for the largest centrality with nodes 3 and 5 tied for the lowest.

2. Mixture of layer dependency cases A and B

If layer 1 is independent, layer 2 is dependent on just layer 1 and layer 3 is dependent on layer 2, the $\tilde{\mathbf{A}}$ matrix takes the form:

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

and the constrained eigenvector centralities are:

$$\begin{aligned}\mathbf{v}_1 &= \{0.50, 0.60, 0.50, 0.34, 0.15\} \\ \mathbf{v}_2 &= \{0.58, 0.26, 0.53, 0.34, 0.44\} \\ \mathbf{v}_3 &= \{0.48, 0.39, 0.43, 0.54, 0.37\}\end{aligned}$$

Since layer 1 is still independent in this scenario, it has the same centrality values as the prior case. For layer 2, we see the expected increase in the centrality of node 1 relative to node 3 given the importance of their adjacent nodes in layer 1 (i.e., node 1 is adjacent to node 2, which has the largest centrality value in layer 1; node 3 is adjacent to nodes 4 and 5, which have the lowest centrality values in layer 1). For layer 3, the centrality for node 3 has the largest change (an increase) relative to the independent scenario, which is expected given that it is adjacent to the node with the largest centrality value in layer 2 (node 1).

3. Mixture of layer dependency cases A, B and C

If layer 1 is independent, layer 2 is dependent on just layer 1 and layer 3 is equally dependent on both layer 2 and itself, the $\tilde{\mathbf{A}}$ matrix takes the form:

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0.5 & 0.5 \end{bmatrix}$$

and the constrained eigenvector centralities are:

$$\begin{aligned}\mathbf{v}_1 &= \{0.50, 0.60, 0.50, 0.34, 0.15\} \\ \mathbf{v}_2 &= \{0.58, 0.26, 0.53, 0.34, 0.44\} \\ \mathbf{v}_3 &= \{0.51, 0.41, 0.39, 0.53, 0.37\}\end{aligned}$$

Since layers 1 and 2 have the same dependency structure as the prior scenario, the centrality values are unchanged. As expected, the equally divided dependency structure for layer 3 yields centrality values that are between those computed in the first two scenarios.

4. All layers are dependency case B with cycle

If layer 1 is dependent on layer 3, layer 2 dependent on layer 1 and layer 3 dependent on layer 2, a cycle is introduced in the layer dependency graph, the $\tilde{\mathbf{A}}$ matrix takes the form:

$$\tilde{\mathbf{A}} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

and the constrained eigenvector centralities are:

$$\begin{aligned}\mathbf{v}_1 &= \{0.40, 0.68, 0.42, 0.38, 0.25\} \\ \mathbf{v}_2 &= \{0.58, 0.21, 0.55, 0.36, 0.43\} \\ \mathbf{v}_3 &= \{0.48, 0.40, 0.43, 0.52, 0.39\}\end{aligned}$$

5 Random Graph Example

A more complex example of our constrained multilayer model involves the analysis of interdependent random graphs. Specifically, we simulated a two layer network where each layer was generated as an interconnected group of 5 Erdos-Renyi random graphs with 20 nodes using the `igraph` R package [2] function call `sample_islands(islands.n=5, islands.size=20, islands.pin=0.2, n.inter=1)`. Constrained eigenvector centrality values were then computed using the proposed algorithm for five different dependency structures:

$$\tilde{\mathbf{A}}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tilde{\mathbf{A}}_2 = \begin{bmatrix} 0.9 & 0.1 \\ 0 & 1 \end{bmatrix}, \tilde{\mathbf{A}}_3 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}, \tilde{\mathbf{A}}_4 = \begin{bmatrix} 1 & 0 \\ 0.1 & 0.9 \end{bmatrix}, \tilde{\mathbf{A}}_5 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

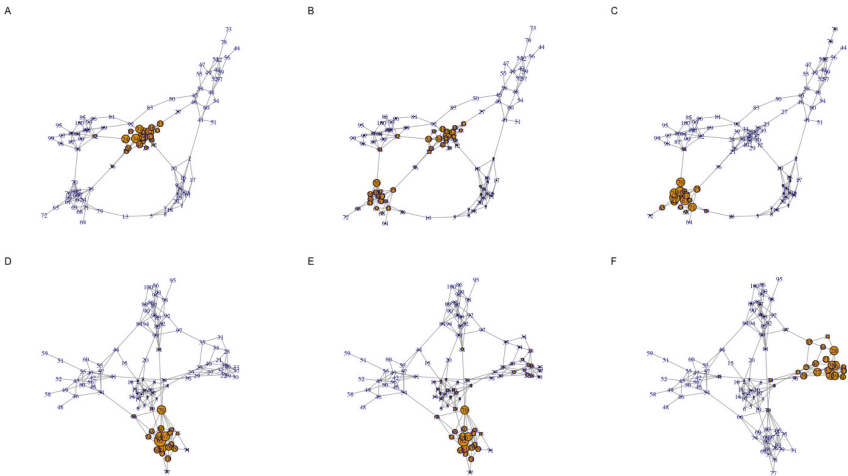


Fig. 2. Centrality visualization for a two layer random graph example. Each row corresponds to a separate layer generated as a interconnected group of 5 Erdos-Renyi random graphs that each have 20 nodes. Panels A and D visualize node eigenvector centrality values when the layers are independent. Panels B and E visualize eigenvector centrality values when 10% kept independent. Panels C and F visualize node eigenvector centrality when adjacent node importance is entirely based on the other layer.

These dependency structures, and the associated results in Fig. 2, have the following interpretations:

1. This represents dependency case A, i.e., the two layers are completely independent. Panels A and D in Fig. 2 visualize the eigenvector centrality values for each layer in this scenario.

2. This represents a mixture of dependency cases A and C with layer 2 independent and layer 1 having adjacent node importance that is a mixture of 10% layer 2 and 90% layer 1, i.e., close to the independence case. Panels B and D in Fig. 2 visualize the corresponding eigenvector centrality values. Interestingly, the small dependence of layer 1 on layer 2 results in a significant shift in eigenvector centrality values but with the distribution of values still dictated by the layer 1 structure.
3. This represents a mixture of dependency cases A and B with layer 2 independent and adjacent node importance for layer 1 completely based on layer 2. Panels C and D reflect the eigenvector centrality values for this case. As expected, the centrality values for layer 1 are quite distinct from the independence case seen in panel A but with the overall pattern still constrained by the layer 1 topology.
4. Similar to case 2, this is a mixture of dependency cases A and C but with the roles of layer 1 and 2 reversed. Panels A and E capture the eigenvector centrality values. In this case, there is a less dramatic shift in the dominant eigenvector centralities for layer 2 relative to the independence case.
5. Similar to case 3, this is a mixture of dependency cases A and B with layers 1 and 2 reversed. Similar to case 3, the eigenvector centrality values for layer 2 are completely distinct from the independence case while still being constrained by the layer 2 topology.

6 Applications and Future Directions

We believe the inter-layer dependency model outlined in this paper has utility for a number of real world multilayer network analysis problems where node adjacency and adjacent node importance are captured by distinct networks, e.g., transportation networks. One specific example of such a real world problem involves the characterization of ligand/receptor mediated cell-cell communication within a tissue. This cell signaling problem was in fact the original motivation for our method. A simplistic model for this problem uses a fully connected network whose nodes represent cells and with edge weights based on the inverse squared Euclidean distance between each cell to capture secreted protein diffusion. In this scenario, we assume that each cell is one of several distinct cell types, e.g., CD8⁺ T cell, and that each cell type is capable of presenting a set of membrane-bound receptor proteins on its surface with the set of receptors associated with different cell types potentially overlapping. We additionally assume that each receptor has a unique cognate secreted ligand protein that can bind to it and that each ligand is produced as a consequence of one or more receptor signaling pathways, i.e., binding of a given receptor by its associated ligand will trigger production of other ligands by the cell.

Given this simple ligand/receptor signaling model and the distribution of cells within a tissue, a key question is to estimate the steady-state activity of each receptor signaling pathway. One approach for answering that question creates a multilayer network with one layer per receptor protein with the activity of

a specific receptor signaling pathway in a given cell represented by the centrality of the associated node. Although we can assume that the adjacency matrices for all receptor layers are identical, a more realistic model would vary the edge weights (potentially with thresholding) based on the dispersion properties of each cognate ligand. Simply computing the eigenvector centrality for each layer, however, does not yield the appropriate answer since the importance of adjacent cells in a given receptor layer reflects that activity of that receptor, which most likely does not impact the activity of that same receptor in other cells, i.e., binding of a given receptor does in general result in secretion of the associated ligand. Instead, one wants to use the importance of cells in the layers corresponding to receptors that produce the cognate ligand. This type of inter-layer dependency structure is exactly what our proposed model supports. We believe that a more general class of systems biology questions may map to similar interdependent multilayer networks.

Our future work in this area includes exploring the theoretical properties of our multilayer network model and interleaved power iteration algorithm (e.g., iteration convergence), characterizing the computational performance on a range of simulated and real networks, performing a comparative evaluation against other multilayer centrality approaches, and applying our technique to study the example cell signaling problem using tissue imaging and genomic profiling data.

Acknowledgments. This work was funded by National Institutes of Health grants R35GM146586, R21CA253408, P20GM130454 and P30CA023108. We would like to thank Peter Bucha and James O'Malley for the helpful discussions and feedback. We would also like to acknowledge the supportive environment at the Geisel School of Medicine at Dartmouth where this research was performed.

References

1. Battiston, F., Nicosia, V., Latora, V.: Structural measures for multiplex networks. *Phys. Rev. E* **89**, 032804 (2014). <https://link.aps.org/doi/10.1103/PhysRevE.89.032804>
2. Csardi, G., Nepusz, T.: The igraph software package for complex network research. *Inter Journal Complex Systems*, 1695 (2006). <https://igraph.org>
3. De Domenico, M., Solé-Ribalta, A., Omodei, E., Gómez, S., Arenas, A.: Ranking in interconnected multilayer networks reveals versatile nodes. *Nat. Commun.* **6**(1), 6868 (2015). <https://doi.org/10.1038/ncomms7868>
4. DeFord, D.R., Pauls, S.D.: A new framework for dynamical models on multiplex networks. *J. Complex Netw.* **6**(3), 353–381 (2018). <https://doi.org/10.1093/comnet/cnx041>
5. Kivela, M., Arenas, A., Barthélemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. *J. Complex Netw.* **2**(3), 203–271 (2014). <https://doi.org/10.1093/comnet/cnu016>
6. von Mises, R., Pollaczek-Geiringer, H.: Praktische verfahren der gleichungsaufloesung. *Zamm-zeitschrift Fur Angewandte Mathematik Und Mechanik* **9**, 152–164
7. Newman, M.E.J.: *Networks: An Introduction*. Oxford University Press, Oxford (2010)

8. Perron, O.: Zur theorie der matrices. *Math. Ann.* **64**(2), 248–263 (1907). <https://doi.org/10.1007/BF01449896>
9. Solá, L., Romance, M., Criado, R., Flores, J., García del Amo, A., Boccaletti, S.: Eigenvector centrality of nodes in multiplex networks. *Chaos: Interdisc. J. Nonlinear Sci.* **23**(3), 033131 (2013). <https://doi.org/10.1063/1.4818544>
10. Taylor, D., Porter, M.A., Mucha, P.J.: Tunable eigenvector-based centralities for multiplex and temporal networks. *Multiscale Model. Simul.* **19**(1), 113–147 (2021). <https://doi.org/10.1137/19M1262632>
11. Tudisco, F., Arrigo, F., Gautier, A.: Node and layer eigenvector centralities for multiplex networks. *SIAM J. Appl. Math.* **78**(2), 853–876 (2018). <https://doi.org/10.1137/17M1137668>
12. Xu, P., He, B., De Sa, C., Mitliagkas, I., Re, C.: Accelerated stochastic power iteration. In: Storkey, A., Perez-Cruz, F. (eds.) *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*, vol. 84, pp. 58–67. PMLR (2018). <https://proceedings.mlr.press/v84/xu18a.html>



Identifying Contextualized Focal Structures in Multisource Social Networks by Leveraging Knowledge Graphs

Abiola Akinnubi^(✉), Mustafa Alassad, Nitin Agarwal, and Ridwan Amure

COSMOS Research Center, UA, Little Rock, Arkansas, USA
{[asakinnubi](mailto:asakinnubi@ualr.edu),[mmalassad](mailto:mmalassad@ualr.edu),[nxagarwal](mailto:nxagarwal@ualr.edu),[raamure](mailto:raamure@ualr.edu)}@ualr.edu

Abstract. Social media and online data pose challenges in information mining, network analysis, opinion mining, and combating misinformation. However, no previous work has been able to apply knowledge graph (KG) and contextual focal structure analysis (CFSA) on multisource data to study situational awareness in public discussion and establish information propagation such as the Belt and Road Initiative (BRI). This research uses multisource data, a knowledge graph model, and a CFSA, which we term KG-CFSA. We extract entities and topics from documents and correlate them with third-party data sources such as Wikidata and Diffbot. We establish relationships using a Cartesian product merge function to develop a graph model. The merge function uses search algorithms and pairwise matching to establish relationships. The model is divided into three instances: document-entity, document-document, and topic-topic. For the document-document instance, we used topics and entities and topic overlaps to establish a relationship while we used co-occurrence for the topic-topic instance. The study identified 276 focal sets; the top two focal sets are focal sets 275 and 276. The most important focal content comes from an Indonesian Twitter user, who operates a personal blog on opinion and story covers. The findings highlight the effectiveness of multisource KG-CFSA in establishing context for a social network analysis.

Keywords: Multisource knowledge Graph · Contextual Focal Structure Analysis · Knowledge Graph

1 Introduction

Social media and online data pose challenges in information mining, network analysis, opinion mining, and combating misinformation. Platforms such as Twitter and Reddit provide valuable information and have been used in campaigns such as COVID-19, China's Belt and Road Initiative, and traditional news platforms. Blogs and video transcripts enable scientists to mine multiple entities from the corpus, identify groups and individuals within communication channels,

and leverage user entities to maximize influence. Topic modeling extracts themes from documents, whereas knowledge graphs allow modeling relationships based on facts. Discovering groups and communities from multisource data requires identifying a focal set and leveraging knowledge graphs and contextual focal structures in social network analyses.

Social media platforms and blogs are ideal for identifying groups and individuals within communication channels, such as people, places, or organizations. User entities within different groups can leverage each other to maximize their influence [4]. Topic modeling is a widely used method in data and platform analysis to understand the overall theme of the collected data. It provides an unsupervised deterministic approach for understanding large text corpora and identifying key themes. Knowledge graph models use topic modeling to establish relationships [2] and contextualize focal structures [17] within online social network discussions. Knowledge graph embedding integrates knowledge graphs into traditional machine learning tasks such as classification, clustering, and prediction, allowing them to fit into tasks that would be difficult on graph-structured data. Research has shown the importance of knowledge embedding in news recommendations, with algorithms based on item-base similarity, knowledge graph similarity, and actuality [10, 17]. Existing knowledge embeddings such as KKGvec2go [14], Wembedder [13], and Pytorch-Biggraph [11] have been evaluated, and pyRDF2Vec [16] has been trained to recommend related news tags for news articles. However, there is limited usage of knowledge embedding in the combination of long and short texts, multiple source social media data, and studying public discussions such as the Belt and Road Initiative. Social network analysis requires mining data from different platforms to extract insights, especially for understanding Belt and Road Initiative conversations. Without focal structures, important information may be buried online.

The study by [5] used the contextual focal structure (CFS) framework to reveal individual interest using Twitter data. This allowed them to measure influence and identify communities using modularity measures and average clustering coefficient values. This study involved community extraction, user-to-user network discovery, and hashtag discovery to create a multiplex network for contextual focal structure analysis (CFSA) detection. Our study explores the implementation and adoption of knowledge graphs, knowledge embedding, and CFSA, and considers factors such as the entity being discussed, the dominant theme, and their relation across topic documents.

This study contextualizes entities as focal communication structures using knowledge embedding and contextual focal structures. This study also explores the application of knowledge graph-contextual focal structure analysis on heterogeneous data from multiple social media platforms, including blogs. Knowledge graph embedding is crucial for identifying focal structures and classifying entities, but existing research has not considered long-text data such as blogs. This study models knowledge graphs from blogs, Twitter, and Reddit using topic modeling and entity extraction. The Indo-Pacific Belt and Road Initiative is an example of the data collection used in this study. We further leverage CFSA

on a multiplex knowledge graph data which consists of a topic-topic model, document-entity model, and document-document model to arrive at other interesting findings. The rest of this article is organized as follows. In Sect. 2, we review the existing literature. In Sect. 3, we highlight the combined framework and methodology (KG-CFSA) used in this study. In Sect. 4, we discuss the results of the combined framework and an example use case on the Belt and Road Initiative discussion from a multisource data and multiplex network point of view. In Sect. 5, we highlight the future work we plan to undertake as well as how other researchers can strengthen our current efforts by leveraging our results and framework.

2 Literature Review

This section reviews existing literature on knowledge graph construction and contextual focal structures. In [17] and [8] the authors proposed a topic model knowledge graph for measuring entity similarity and topic coherence. They extracted topic models from the text and used them to list related papers. The work done by [1] focused on extracting knowledge graphs from Wikidata. They employed cosine similarity between entity vectors and sentence vectors, coupled with Latent Dirichlet Allocation (LDA) on Wikipedia articles, to select the top topics for potential permutations. This approach improved the property selection for graph models by 85%.

Another approach suggested by [7] is to use main topics from learning resources to extract knowledge by merging information and updating it using a subject-category look-up on Wikipedia. The algorithm consists of three layers: text extraction, keyword extraction, and category extraction. It uses Wikipedia, TextRank, and Genism to extract keywords and categories, similar to [1], but with a different platform focus. Knowledge graphs have gained popularity in representing data from social media. Knowledge graph embedding enables the application of linear and nonlinear machine learning [9]. The TransET model was used to model entities and relations in a low-dimensional space. It achieved state-of-the-art accuracy in link prediction and triple classification [18]. Knowledge embedding has also been applied to studying Twitter data, classifying tweets, and identifying user groups that utilize negative narrative framing [2]. Dimensionality reduction was implemented to study the YouTube commenters to identify suspicious behavior across different channels [15]. Contextual focal structure analysis is a method that uses multiplex network data to identify coordinated activities in a network. The authors in [12] used a semi-supervised novel contextualized approach to node representation for effective representation across a multiplex network. This approach yielded embedding that achieved state-of-the-art performance in classification and clustering tasks. The authors described a multiplex network as a graph structure in which entities are connected via multiple relationships, where each relationship represents a distinct layer. This approach maximizes mutual information shared between local nodes, allowing for classification and clustering tasks.

Community detection has become an important step in social network analysis [3], as seen in [6]. The authors suggested that influential nodes have links to trending contexts using contextual focal structure analysis in complex networks. The study by [4] identified influential users in a user hashtag propagated network using a multiplex network. This suggests that contextual focal structure analysis in complex networks requires multiple methods, influential nodes to have links to trending context, and partisanship to exist between coordinating groups. In addition, the works of [5] measured the influence of contextual focal sets and applied network structural measures on data sets from social movements such as Black Lives Matter and COVID-19 discourse. The results are evaluated using the ranking correlation coefficient in real-world scenarios.

3 Method of the Study

This section is divided into three subsections. Section 3.1 describes the dataset used and how it was collected. Section 3.2 describes the approach used in modeling the knowledge graph for the multisource data, and Sect. 3.3 describes the CFSA and the KG-CFSA model, which is a combination of the generated KG and the existing CFSA model by [4].

3.1 Data Collection

The data sources used in this study are shown in Table 1. The data were extracted using the following query phrases: *'antek', 'aseng asing', 'Tiongkok', 'Tionghoa', 'Indonesia', 'Cina', 'OBOR', 'BRI', 'kebijakan', 'luar', 'negeri', 'proyek', 'pekerja', 'Cina', 'China', 'Tionghoa', 'Tiongkok', 'Pembangkit Listrik Tenaga Batubara', 'Cina China', 'Perusahaan Listrik Negara', 'BRI OBOR', 'Proyek 35000 Megawatt', 'Maritime Silk Road', 'Jakarta Indonesia', 'Global Maritime Fulcrum', 'Jokowi', 'Tiongkok Tionghoa', 'Menguasai', 'Tiongkok Tionghoa', 'ekonomi', 'Pekerja Cina pulang', 'Chinese workers go home'.*

We selected 10,000 documents for each platform because each platform provides three different secondary data: Topics for each document, entities, and high probability words generated from the extracted topics. This provides a denser relationship between the documents and the platforms used in this study.

3.2 Multisource Knowledge Graph Model

Our approach involved extracting entities from each document. We then queried third-party knowledge databases such as Wikidata and Diffbot to enhance the extracted entities with more information. This also helped validate some popular entities. Once the entities were extracted and enhanced, we proceeded to extract the topics and other important themes from each document. Although we limited the number of topics to 10 for this study, each topic can contain numerous amounts of topic words. We then modeled topics and entities in a pairwise relationship using a Cartesian product to establish the relationship between the

Table 1. Data collection statistics for Belt and Road Initiative data across multiple platforms.

Platform	Quantity	Year range
Blog	10,000	2019–2022
Reddit	10,000	2019–2022
Twitter	10,000	2019–2022
YouTube	10,000	2019–2022

entities, topics, and documents. We call the algorithm for the Cartesian product that was specifically modeled for this study a merger function because it uses both search algorithms and pairwise matching to establish relationships. Each network layer in the multiplex network is composed of D-D and T-T where document-document is computed based on entities mentioned, similarity, and topic overlap. T-T is computed based on their overlap across entities and documents. The following points further explain how we establish these relationships:

We model document-entity-document (D-E-D) relationships where a document can have a mentioned entity in an undirected graph. We also modeled the extracted topics that were established from LDA and crossed with Wikidata in a document-topic-document relationship (D-T-D) where the document can belong to different topics or themes. We model topic-entity-topic relationships in which a document belongs to a similar entity and has an undirected graph structure.

We then use a binary search algorithm integrated into our Cartesian product-based merge function to develop a multiplex network instance. The multiplex network was categorized into two instances. One instance uses the topic word extracted, the topic number they belong to, and the documents from which these items were extracted, with the entities extracted as the edges forming a triple of (document, entity, document), (topic, entity, topics), and (word, entity, word) respectively. The second instance uses the topic number to which the document, entity, and word belong as the basis for establishing the relationship for the network construction.

Equations 1-8 and Algorithm 1 below show the step-by-step implementation of the graph model and merge function used in this study to build the knowledge graph relationship between entities and documents. Note: T represents topics, E represents entities, D represents documents, M represents multisource data, G represents the graph, and W represents topic words.

$$E = \{E_1, E_2, E_3, \dots, E_n\} \quad (1)$$

$$T = \{T_1, T_2, T_3, \dots, T_n\} \quad (2)$$

$$D = \{T(E, W)_1, T(E, W)_2, T(E, W)_3, \dots, T(E, W)_n\} \quad (3)$$

$$M = \{D_1, D_2, D_3, \dots, D_n\} \quad (4)$$

$$D_i * D_j = \{(d_i, d_j) | d_i \in D_i \text{ and } d_j \in D_j\} \quad (5)$$

therefore, represented as

$$\begin{aligned} & D_i * D_j \\ D_i * \dots D_n &= \{(D_i * D_j) \forall i, j \in \{0, 1, \dots, n\}\} \\ &= \{(d_i * d_j) | d_i \in D_i \text{ and } d_j \in D_j \forall i, j \in \{0, 1, \dots, n\}\} \end{aligned} \quad (6)$$

Graph G or KG for a multisource can be represented as

$$G = \prod_{i=1}^n D_i \quad (7)$$

3.3 KG-CSFA

The KG-CFSA model used in this study was designed to use the existing CFSA model developed by [4] and the knowledge graph construction model for a multi-source dataset, which we recently modeled for establishing relationships between data from multiple social media platforms. KG-CFSA is designed for document-to-document relationships, topic-to-topic relationships, and document-to-entity relationships, which can help contextualize the different layers of the multiplex network model. The model outcome is expected to represent influential entities, which are linked to other important extracted entities that are connected to topics and share the same contexts that exist in each document. Figure 1 Shows the overall CFSA model modified after [4] to accommodate data from multiple sources.

For an unweighted, undirected graph \mathbf{G} with \mathbf{N} nodes, the adjacency matrix is typically a symmetric $N * N$ matrix \mathbf{A} , where a_{ij} is 1 if there is an edge between nodes i and j in G and 0 otherwise. The adjacency matrix for a layer graph $G\alpha$ is a symmetric matrix A^α , where a_{ij} is 1 only if an edge exists between node (i, α) and j, α in G . Similar to this, the adjacency matrix for $G\beta$ is an n -by- m matrix $\rho = P_i\alpha$, where $p_i\alpha$ is 1 only if there is an edge connecting node i to layer α in the participation graph forming the participation matrix. The coupling graph G_f has an $N * N$ adjacency matrix $L = \{c_{ij}\}$, where c_{ij} is 1 only if there is an edge between the node-layer pair i and j in the coupling graph, which represents the same node in different layers. It is possible to stack rows and columns of L such that node-layer pairs of the same layer are adjacent and ordered with zero diagonal blocks. Using this arrangement, $c_{ij} = 1$, with $i, k = 1, \dots, N$ represents an edge between a node-layer pair in layer 1 (document-to-document) and node layer pair in layer 2 (document-to-entity) layer if $i < n_i < j < n_2$. This process was also extended to layer 3 (topic-to-topic). The supra-adjacency matrix, which is a synthetic representation of the entire multiplex M , is the adjacency matrix

Algorithm 1: Graph Model with a Merge Function for entity mapping

```

Data:  $size \leftarrow Integer$ 
 $CartesianProducts \leftarrow Array$ 
 $Entities \leftarrow \{E_1, E_2, ..E_n\}$ 
Result:  $GraphDataModel \leftarrow CartesianPairs, Entities$ 
Function  $CartesianPairs(size, CartesianProduct \leftarrow Array)$  is
   $results \leftarrow Array$ 
  foreach  $product \in CartesianProduct$  do
     $currentCollection \leftarrow \{\}$ 
    for  $i \in range(size)$  do
       $currentCollection[i][0] \leftarrow product[i][1]$ 
    end
     $results.append(currentCollection)$ 
  end
   $return results;$ 
end
Function  $CartesianProduct(array)$  is
   $results \leftarrow Array \subset Array$ 
  for  $i \in [0..size(array)]$  do
     $innerData \leftarrow Array$ 
    foreach  $item \in results$  do
      for  $element \in array[i]$  do
         $entry = item + [element]$ 
         $innerData.append(entry)$ 
      end
    end
     $results \leftarrow innerData$ 
  end
   $return results$ 
end
Function  $GraphDataModel(CartesianPairs, Entities \leftarrow \{\})$  is
   $documents \leftarrow \{entity \leftarrow [] \in \{\}\}$ 
  for  $entity, values \in documents$  do
     $entityCartesianPairs \leftarrow entity \in \subseteq$  of  $CartesianPairs$ 
     $edges \leftarrow entityCartesianPairs$ 
     $documents[entity].extend(edges)$ 
  end
   $return documents$ 
end

```

of the supra-graph G_M . The coupling matrix and intra-layer adjacency matrices can be used to derive it as shown in the equation below.

$$A = A^\alpha \bigoplus_{\alpha} L \quad (8)$$

The interlayer adjacency matrix can also be defined as follows $A = \bigoplus A^\alpha$. The goal of the CFSA model is to maximize network modularity values and user-level centrality values. It uses a spectral modularity method to calculate

the impact of entities on various layers. Additionally, the model uses vector parameters to transmit data between the entities and network levels. Equation (9) defines the objective function used to maximize the centrality values in the network.

$$\max \sum_{i=1}^n \sum_{j=1}^m (\delta_i^{DD} \oplus \beta_{ij}^{DT} \bar{h}_j^{TT}) \quad (9)$$

Here, n is the number of nodes in the users in layer DD; m is the number of nodes in layer TT. δ_i^{DD} is the sphere of influence for user i in DD. \bar{h}_j^{TT} is the number of j entities in TT connected by an edge to entity i in DD.

The network level of the model assesses the influence of user sets throughout the entire \bar{A} network. This measurement aims to understand how users impact the \bar{A} network when they become part of it. To gauge the influence of users identified at the user level, a spectral modularity method is applied. Additionally, a vector parameter \mathbb{C} is used to transfer user information between the user and network levels. The contextual focal structure is gathered using the equation below;

$$\mathbb{C}_{\varrho_j x} = \delta_{jx}(\bar{\mu}_{jx}^Q) \quad (10)$$

where $\delta_{jx}(\bar{\mu}_{jx}^Q)$ is the nondominated solution that maximizes the network's spectral modularity values used to transfer the results back to the user level. $\mathbb{C}_{\varrho_j x}$ elects the sets that gather all the criteria from both levels at each iteration x . The reader can refer to [4] for more information.

4 Discussion

This section discusses the results of the KG-CFSA model and its outcome using the China's Belt and Road Initiative as a case study. This section is divided into three subsections. Section 4.1 discusses the results obtained from the CFSA model for the document-entity-documents and topic-entity-topic inputs.

4.1 Contextual Focal Structure Analysis

The case study implemented in this research focuses on the China's Belt and Road Initiative. The documents identified were groups whose platforms propagated information about the China's Belt and Road Initiative on a multisource platform. The CFSA model identified 276 CFS sets in a multiplex network (Document-Topic layer), and the sets are different in sizes, topics, and entities.

The results from the topic modeling are presented in Table 2. There are more topics in the NS276 set. The tweet from this set will be the focal point of the subsequent discussion.

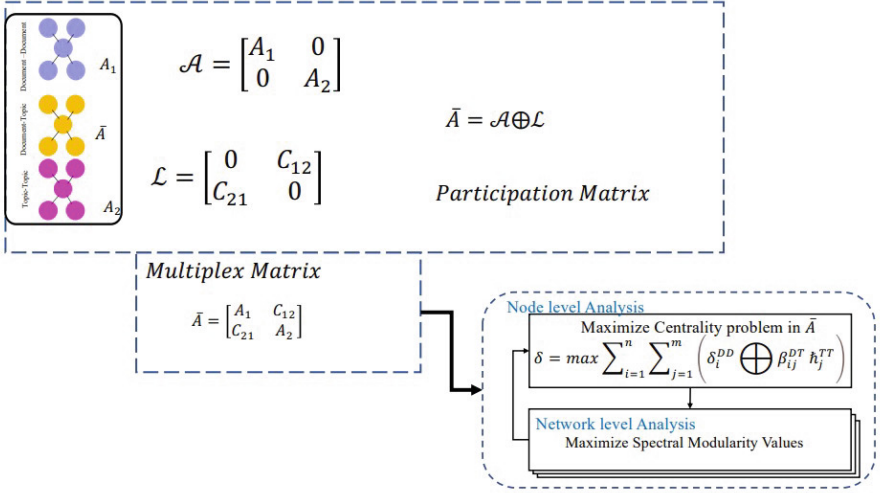


Fig. 1. The CFSA model takes an initial set of document-document multisource network and topic-topic (co-occurrence) multisource network. Then, we generate the matrix.

Table 2. Result of the topic modelling on the Knowledge graph

CFS Set	No of Document	No of Topics	No of Community
NS229	49	1 (Topic 9)	4
NS230	34	1 (Topic 0)	3
MOD254	37	1 (Topic 9)	4
MOD253	61	1 (Topic 9)	5
NS276	73	3(Topic 9, Topic 5, Topic 0)	3

The plot in Fig. 2 shows the focal point around two influential tweets. The tweets reflect the discussion on the Maritime and Digital Silk Road which was the subject of discussion as shown in Table 3. The keywords in the topics under consideration (0, 5, and 9) are displayed in Table 3.

The words in Topic 0 refer to activities of countries such as China (commonly referred to as “Tiongkok” in Indonesian and Malay), the USA (referred to as “AS” or an abbreviation of “Amerika Serikat”), and Japan (referred to as “japang”). The second topic with fewer descriptive words is related to Topic 0. It talks about China’s influence over the maritime Silk Road. Topic 9 describes the external influence and contributions of Europe, the USA, NATO, WHO, and other international organizations and sovereign nations on the situation in the Indo-Pacific region.

A careful examination of Fig. 2 shows the presence of keywords that made the tweets and users the focal point. Both users are concerned with Indonesia as a focus for the digital Silk Road in the Indo-Pacific region. This indicates there are diverse

groups talking about the Silk Road; one group used the term for the maritime Silk Road, while another group of users used it to discuss a digital Silk Road.

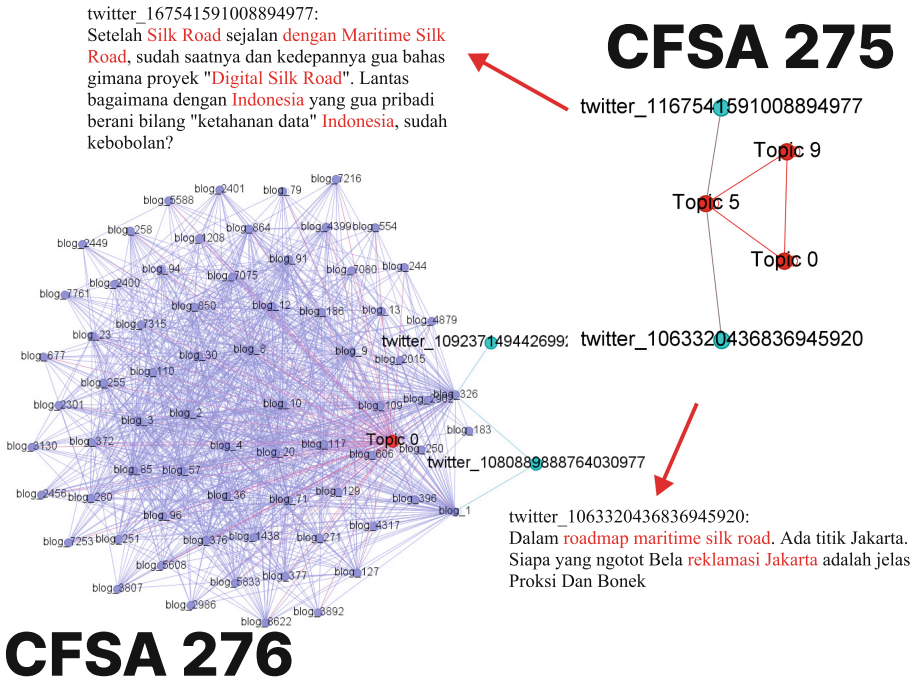


Fig. 2. KG-CFSA results for the top two focal sets (276 on the left and 275 on the right). Context and entity mapping highlighting important tweets by the most important document nodes.

Table 3. Topics and keywords extracted from the Knowledge graph.

	Words and Entities
Topic 0	Tiongkok, Yang, Jokowi, Dengan, Tionghoa, Anda, Saya, Dalam, Karena, Untuk, Setelah, AS, telah, Kita, Kami, Kalau, Lalu, Tidak', Maka, Bahkan, Jadi, Selain, itu, Mereka, Salah, Seputar Tiongkok, Jepang, Hal, Begitu, Mungkin
Topic 5	kedepannya gua, Presiden Xi Jinping, Jalur Sutra Maritim, Asia, Karena, Dalam, Bela, Proksi Dan Bonek, Di Indonesia
Topic 9	NATO, Washington, Xi Jinping, Xi, West, Biden, CCP, Donald Trump, Putin, WHO, CIA, Taliban, Trump, Indo-Pacific, Obama, Huawei, UN, coronavirus, Chinese Communist Party, PLA, European Union, ASEAN, AI, FBI, Pompeo, Democrats, Coronavirus, CNN, EU

5 Conclusion and Future Research

Social network analysis has become vital for understanding the dynamics of online communication, and our analysis continues to grow in complexity. It is important to understand the important role played by each entity and its relationship with other entities that form the focal set. This relationship is formulated by considering the entities as a focal set and contextualizing the focal set across multisource social networks. This study has leveraged developing a multisource knowledge graph model where data is a combination of different social media platforms such as Twitter and Reddit and also from personal blogging and news platforms. The knowledge graph model helps in achieving a more informative contextualization of information spread. We then applied to it the CFSA framework developed by [4] to further explore how information can be contextualized when such a multisource knowledge graph model is developed. This work then uses the combination of the CFSA and KG model (KG-CFSA) to contextualize the Belt and Road Initiative using data collected between 2019 and 2022. We arrived at the following findings:

We identified 276 focal structures and focused on the most dominant focal structure. The dominant focal structure that spread the information and spanned across three dominant topics belonged to an Indonesian Twitter user who operates a personal blog at <https://voxjax.wordpress.com/>. The three topics, i.e., Topic 0, Topic 9, and Topic 5, discuss “[a] roadmap [for] maritime silk road” a term synonymous with the Belt and Road Initiative.

Finally, our findings show the strength of leveraging a multisource knowledge graph model in helping to identify the stance and context of online users by arriving at a context faster. This is visible from our obtained results, which would have otherwise been challenging to unravel without multisource multiplex network data.

In the future, we believe our study can establish more on stance detection, bias, and areas such as online epidemiological modeling to uncover insights in these areas. In the future, this work can attempt to see how the KG-CFSA model can help provide situational awareness in uprisings and social campaigns.

Acknowledgement. This research is funded in part by the U.S. National Science Foundation (OIA-1946391, OIA-1920920, IIS-1636933, ACI-1429160, and IIS-1110868), U.S. Office of the Under Secretary of Defense for Research and Engineering (FA9550-22-1-0332), U.S. Office of Naval Research (N00014-10-1-0091, N00014-14-1-0489, N00014-15-P-1187, N00014-16-1-2016, N00014-16-1-2412, N00014-17-1-2675, N00014-17-1-2605, N68335-19-C-0359, N00014-19-1-2336, N68335-20-C-0540, N00014-21-1-2121, N00014-21-1-2765, N00014-22-1-2318), U.S. Air Force Research Laboratory, U.S. Army Research Office (W911NF-20-1-0262, W911NF-16-1-0189, W911NF-23-1-0011), U.S. Defense Advanced Research Projects Agency (W31P4Q-17-C-0059), Arkansas Research Alliance, the Jerry L. Maulden/Entergy Endowment at the University of Arkansas at Little Rock, and the Australian Department of Defense Strategic Policy Grants Program (SPGP) (award number: 2020-106-094). Any opinions, findings, conclusions, or recommendations expressed in this

material are those of the authors and do not necessarily reflect the views of the funding organizations. The researchers gratefully acknowledge the support.

References

1. Abels, P.B., Ahmadi, Z., Burkhardt, S., Schiller, B., Gurevych, I., Kramer, S.: Focusing Knowledge-based Graph Argument Mining via Topic Modeling. ArXiv (2021). <https://www.semanticscholar.org/paper/Focusing-Knowledge-based-Graph-Argument-Mining-via-Abels-Ahmadi/bd429d49ac29aa8ba9c2267905657ac7aaacfe39>
2. Abu-Salih, B., et al.: Relational learning analysis of social politics using knowledge graph embedding. *Data Min. Knowl. Discov.* **35**(4), 1497–1536 (2021). <https://doi.org/10.1007/s10618-021-00760-w>, <https://link.springer.com/10.1007/s10618-021-00760-w>
3. Al-khateeb, S., Agarwal, N.: Modeling flash mobs in cybernetic space: evaluating threats of emerging socio-technical behaviors to human security. In: 2014 IEEE Joint Intelligence and Security Informatics Conference, pp. 328–328 (2014). <https://doi.org/10.1109/JISIC.2014.73>
4. Alassad, M., Agarwal, N.: Contextualizing focal structure analysis in social networks. *Soc. Netw. Anal. Min.* **12**(1), 103 (2022). <https://doi.org/10.1007/s13278-022-00938-0>, <https://doi.org/10.1007/s13278-022-00938-0>
5. Alassad, M., Agarwal, N.: A systematic approach for contextualizing focal structure analysis in social networks. In: Thomson, R., Dancy, C., Pyke, A. (eds.) *Social, Cultural, and Behavioral Modeling. SBP-BRiMS 2022. LNCS*, vol. 13558. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-17114-7_5
6. Alassad, M., Hussain, M.N., Agarwal, N.: Comprehensive decomposition optimization method for locating key sets of commenters spreading conspiracy theory in complex social networks. *Cent. Eur. J. Oper. Res.* **30**(1), 367–394 (2022). <https://doi.org/10.1007/s10100-021-00738-5>
7. Badawy, A., Fisteus, J.A., Mahmoud, T.M., Abd El-Hafeez, T.: Topic extraction and interactive knowledge graphs for learning resources. *Sustainability* **14**(1), 226 (2022). <https://doi.org/10.3390/su14010226>. Place: ST ALBAN-ANLAGE 66, CH-4052 BASEL, SWITZERLAND Publisher: MDPI Type: Article
8. Brambilla, M., Altinel, B.: Improving topic modeling for textual content with knowledge graph embeddings. In: *Improving Topic Modeling for Textual Content with Knowledge Graph Embeddings* (2019). URL <https://www.semanticscholar.org/paper/Improving-Topic-Modeling-for-Textual-Content-with-Brambilla-Altinel/ab3e352affeceabc35bab1b9628d5a2f6443acf2>
9. Costabello, L., Pai, S., Van, C.L., McGrath, R., McCarthy, N., Tabacof, P.: *AmpliGraph: a Library for Representation Learning on Knowledge Graphs* (2019). <https://doi.org/10.5281/zenodo.2595043>
10. Engleitner, N., Kreiner, W., Schwarz, N., Kopetzky, T., Ehrlinger, L.: Knowledge graph embeddings for news article tag recommendation. In: *Knowledge Graph Embeddings for News Article Tag Recommendation* (2021). <https://www.semanticscholar.org/paper/Knowledge-Graph-Embeddings-for-News-Article-Tag-Engleitner-Kreiner/5bde615b31c46338f8d3e0a404c3728238b5a322>
11. Lerer, A., et al.: *PyTorch-BigGraph: A Large-scale Graph Embedding System*. In: *Proceedings of the 2nd SysML Conference*. Palo Alto, CA, USA (2019)

12. Mitra, A., Vijayan, P., Sanasam, R., Goswami, D., Parthasarathy, S., Ravindran, B.: Semi-supervised deep learning for multiplex networks. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 1234–1244 (2021). <https://doi.org/10.1145/3447548.3467443>. URL <https://dl.acm.org/doi/10.1145/3447548.3467443>. Conference Name: KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining ISBN: 9781450383325 Place: Virtual Event Singapore Publisher: ACM
13. Nielsen, F.r.: Wembedder: Wikidata entity embedding web service (2017). <https://doi.org/10.48550/arXiv.1710.04099>, <http://arxiv.org/abs/1710.04099>
14. Portisch, J., Hladik, M., Paulheim, H.: KGvec2go - knowledge graph embeddings as a service. In: Proceedings of the Twelfth Language Resources and Evaluation Conference, pp. 5641–5647. European Language Resources Association, Marseille, France (2020). <https://aclanthology.org/2020.lrec-1.692>
15. Shajari, S., Agarwal, N., Al Assad, M.: Commenter Behavior Characterization on YouTube Channels (2023). <https://doi.org/10.48550/ARXIV.2304.07681>
16. Steenwinckel, B., Vandewiele, G., Agozzino, T., Ongenaes, F.: pyRDF2Vec: a python implementation and extension of RDF2Vec. In: Pesquita, C., et al. The Semantic Web. ESWC 2023. LNCS, vol. 13870. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-33455-9_28
17. Sun, H., Ren, R., Cai, H., Xu, B., Liu, Y., Li, T.: Topic model based knowledge graph for entity similarity measuring. In: 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE), pp. 94–101 (2018). <https://doi.org/10.1109/ICEBE.2018.00024>
18. Wang, P., Zhou, J., Liu, Y., Zhou, X.: TransET: knowledge graph embedding with entity types. *Electronics* **10**(12), 1407 (2021). <https://doi.org/10.3390/electronics10121407>, <https://www.mdpi.com/2079-9292/10/12/1407>



How Information Spreads Through Multi-layer Networks: A Case Study of Rural Uganda

Jennifer M. Larson¹(✉) and Janet I. Lewis²

¹ Vanderbilt University, Nashville, TN, USA
jennifer.larson@vanderbilt.edu

² George Washington University, Washington, D.C., USA

Abstract. The social networks that interconnect groups of people are often “multi-layered” – comprised of a variety of relationships and interaction types. Although researchers increasingly acknowledge the presence of multiple layers and even measure them separately, little is known about whether and how different layers *function* differently. We conducted a field experiment in twelve villages in rural Uganda that measured real multi-layer social networks and then tracked how each layer was used to discuss new information about refugees. A majority of respondents discussed refugees with someone to whom they were connected in the social network. The connections came from all four layers, though the layer indicating regular homestead visits was used most frequently. People did not discuss refugees with every one of their network neighbors; homophily in views, homophily in level of interest, and the alter’s interest in the topic best distinguish links that were used from those that were not.

Keywords: Multi-Layer Networks · Discussion Networks · Link Function · Uganda · Refugees

1 Introduction

Real social networks tend to be comprised of a rich variety of relationships and interaction types, and hence are “multi-layered” [5, 6, 10, 12, 15]. Scholars studying networks empirically often collect data on multiple layers, such as friends, kin, discussion partners, sources of assistance, and so on [2–4, 11, 16, 18, 20]. These networks are of interest because they likely *do* something—spread information, apply peer pressure, share resources—that matters to outcomes across the social sciences [7, 23, 26].

Understanding how exactly links function is an important step in the process of understanding when and why networks matter [19], especially since certain links may function differently than others. For instance, some links may be based on deep trust, facilitating the spread of sensitive information from person to person, while others may be shallower, only allowing non-sensitive information

to pass through [1, 14, 17]. When links are not interchangeable in their function, researchers need to account for this in their measurement strategy, and aggregating links across layers could be misleading [8, 9, 15, 21, 22]. An important question is then: which links do what and when?

This question is expansive, and a complete answer surely depends on the context in question. A productive way forward would be to amass a collection of studies of link functions in multi-layer networks across contexts. This article contributes one. It focuses on a case which allows deep exploration of the function of different links in the context of rural Ugandan villagers learning new information about refugees.

Specifically, we conducted a field experiment in twelve villages in northwestern Uganda in which we elicited four layers of social networks for all households: who shares meals with whom, who visits whose homesteads, who consults whom in the presence of rumors, and who would turn to whom to borrow money. The study also presented information about the experiences of refugees to a randomly selected half of households. Two weeks later, participants were surveyed again and asked to name the people with whom they had conversed about refugees in the interim. By matching these names with the social network, we can determine whether people used any of the four layers to discuss refugees.

Consistent with previous studies that measure multi-layer networks, we find that the overlap between layers is imperfect and each contributes distinct sets of links and structural features [13, 21, 22, 24, 25]. We find that a majority of respondents did turn to social network neighbors (as opposed to others in the village or beyond) to discuss the new information; in one village, 70% of respondents who talked to anyone did so with a network neighbor. Across villages, discussion partners were connected to the respondent most often in the visit layer (65%), followed by the meal layer (53%), then borrow (44%) and rumor (39%).

Our data also allow us to compare people linked to the respondent in the social network who were named as discussion partners (1212 total links) with people linked to the respondent who were not (6593 total links) to try to understand why respondents made use of the links they did. We consider whether alter characteristics such as personal experience as a refugee, social relationships with refugees, occupation, views on the topic, and interest in the topic matter. Of these, only the alter’s level of interest in refugees significantly differentiates the two groups: alters who see refugees as a very pressing issue are more likely to be named as discussion partners. We also consider whether homophily with respect to religion, language, personal refugee status, views on the topic, and level of interest in the topic matter. Of these, both views on refugees and interest in the topic do: alters who agree on the level of threat refugees pose and the importance of the topic are more likely to be selected by the respondent as a discussion partner.

2 Village Networks

We used four name-generator questions in a baseline survey to measure social networks in each of the twelve villages. Table 1 describes the resulting social

network, here represented as the union of the four layers, for each village. Nodes are households, links are directed, and the count of links indicates the number of times one household lists someone in another in response to at least one of the four name generator questions. The table also reports features of these networks, including the mean total degree, the maximum in-degree, the number of nodes who have in-degree or out-degree equal to zero, mean transitivity, and the proportion of households in the largest component.

Table 1. Aggregated social network by village

Village	Nodes	Links	Degree	Max In	0 Out	0 In	Trans	Lg Comp
1	132	799	12.11	33	5	12	0.30	0.99
2	114	505	8.86	34	3	16	0.21	1.00
3	148	962	13.00	27	5	13	0.29	0.99
4	125	938	15.01	34	5	18	0.29	0.99
5	163	1030	12.64	31	6	14	0.25	0.98
6	126	692	10.98	28	2	11	0.35	0.99
7	121	456	7.54	23	7	19	0.18	0.99
8	130	437	6.72	17	9	21	0.20	0.98
9	112	803	14.34	33	9	23	0.38	0.96
10	104	364	7.00	12	8	15	0.30	0.99
11	180	492	5.47	23	29	53	0.13	0.96
12	149	327	4.39	24	27	51	0.15	0.89

Table 2 separates the networks into the four layers and reports the same structural features. The values are reported as averages across the villages by layer. On average, a village has 134 household nodes in the network. Each layer contributes differently to the overall village network. The visit layer has the most links on average, though the rumor layer has the highest in-degree—more people point to the same person to vet rumors than to visit in their home. The meal layer has the highest transitivity; households who have members who share meals with the same household are more likely to share meals with one another as well. The borrow layer has the largest number of nodes with out-degree and in-degree equal to zero; many households have no one they would borrow money from, and many households would not be asked.

Table 2. Characteristics of each of the four layers averaged over the 12 villages

Layer	Nodes	Links	Degree	Max In	0 Out	0 In	Trans	Lg Comp
Meal	134	298	4.56	11	31	43	0.20	0.87
Visit	134	344	5.23	14	24	38	0.18	0.92
Rumor	134	220	3.31	15	39	57	0.13	0.81
Borrow	134	204	3.10	14	46	64	0.16	0.74

For illustration, we pick one of the villages and visualize the four layers. Figure 1 shows each of the layers for village 7, holding the node placement fixed. Nodes are sized proportional to degree.

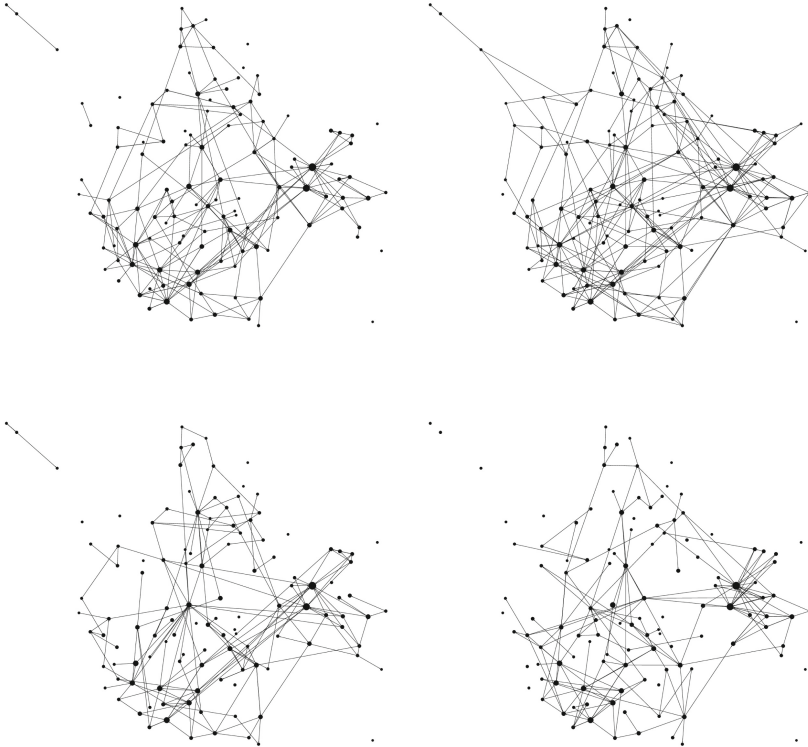


Fig. 1. The four layers of the multi-layer household network for Village 7. From top left to bottom right: shared meals, visit homestead, discuss rumors, borrow money.

3 Use of Village Social Networks to Discuss Refugees

In the second survey two weeks after the baseline, respondents were asked to think back over the past two weeks and name anyone with whom they had a conversation about refugees. Not everyone had done so, though a majority had. Table 3 shows the number of respondents who named any names and also reports this as a proportion of the village’s households. It also shows the proportion of respondents for whom at least one name offered was a neighbor in at least one layer of their village’s social network. We do see variation across villages, ranging from village 11 with 36% for whom this is the case to village 8 with 70%.

We next zoom in on the people who said they did have a conversation about refugees with anyone in the past two weeks. These respondents were invited to name up to five of their discussion partners. The average number of names offered across villages ranges from 2.79 (in village 11) to 4.02 (in village 2).

Table 3. Who discussed refugees, were they network neighbors?

Village #	Names > 0	Prop > 0	Any In NW
1	58	0.44	0.64
2	63	0.55	0.63
3	78	0.53	0.67
4	66	0.53	0.56
5	116	0.71	0.62
6	69	0.55	0.64
7	83	0.69	0.66
8	82	0.63	0.70
9	58	0.52	0.64
10	49	0.47	0.59
11	108	0.60	0.36
12	86	0.58	0.40

Table 4 shows the total number of the names respondents offered that also appear as their neighbors in at least one layer of the social network on average across respondents within each village. The four subsequent columns break these totals apart into the number of names that appear as a link in each of the four layers of the social network, reported as an average number of names. For village 1, on average 1.07 people listed are also network neighbors; these people are distributed across the four layers as .43 names in the meal layer, .62 in the visit layer, .38 in the rumor layer, and .47 in the borrow layer. The four layers do not sum to the total number of people because they are not mutually exclusive; a link between a respondent and an alter can appear in more than one layer, so a name can appear in more than one layer for a respondent.

4 When Are Links Most Likely to Be Used?

Next we investigate why the links in the network that were used to discuss refugees were in fact used. That is, for each respondent, we know the set of network neighbors across all layers, and we know that some, but not all, of them were selected as discussion partners about refugees. Was the selection random with respect to link, or do we observe differences between used and unused links?

We investigate two sets of attributes of the links. One set centers around attributes of the alter. We might think that alters who have relevant experience,

Table 4. Breakdown of discussion partners by layers of network

Village	Total in NW	#inMeal	#inVisit	#inRumor	#inBorrow
1	1.07	0.43	0.62	0.38	0.47
2	1.35	0.81	0.83	0.43	0.67
3	1.18	0.56	0.58	0.41	0.38
4	1.11	0.56	0.70	0.38	0.53
5	1.20	0.53	0.66	0.54	0.43
6	1.13	0.68	0.74	0.25	0.49
7	1.23	0.67	0.89	0.52	0.52
8	1.22	0.74	0.84	0.62	0.51
9	1.14	0.41	0.71	0.28	0.47
10	0.90	0.43	0.63	0.31	0.39
11	0.47	0.24	0.25	0.27	0.19
12	0.63	0.24	0.38	0.36	0.23
Pooled	0.53	0.65	0.39	0.44	

for instance by having been a refugee once themselves (this is true for about a third of our respondents) or who themselves know refugees personally, would be prioritized. Or we might think that alters who have a connection to the land, one of the key resources in question when refugee issues come up, in their occupation as farmers, would be prioritized. Or maybe an alter’s views on refugees¹ or the extent to which she finds refugees to be a pressing issue are important to respondents when selecting discussion partners.² Out of all of these alter characteristics, the only one that distinguishes the alters selected from those that are not is the alter’s interest in refugees: links to alters who find the issue of refugees to be more pressing are more likely to be used to discuss refugees.

Likewise, we consider homophily as a possible distinguishing factor between links in the social network used to discuss refugees and those that were not. We consider both religious and language homophily to see if common values or assured ability to communicate are relevant. We also consider shared refugee status, which would be relevant if respondents who were once refugees sought out their network neighbors who also shared this experience (or respondents who have never been a refugee might seek out like neighbors as well). Shared views about refugees, and a shared interest in the topic, could also facilitate conversations. In fact shared interest in refugees distinguishes links used from

¹ Our survey asks respondents to react to the statement “Refugees threaten the way of life in my community” with a five point scale from strongly agree to strongly disagree. Larger values indicate stronger disagreement, and hence warmer attitudes towards refugees.

² Our survey asks respondents how important they find the issue of refugees to be on a five point scale. Smaller values indicate greater importance.

those that were not in the network, and shared views does as well, though at a lower level of statistical significance.³

Table 5. Comparing the links in the multilayer social network that were used to discuss refugees to those that were not.

	Network link used	Network link not used	p-value
Link Count	1212	6593	
Alter was refugee	0.33	0.32	0.55
Alter knows refugee	0.73	0.72	0.37
Alter farmer	0.82	0.81	0.54
Alter’s views	3.61	3.54	0.21
Alter’s interest	1.36	1.48	0.00
Relig homoph	0.76	0.74	0.11
Language homoph	0.85	0.84	0.46
Refugee status homoph	0.62	0.64	0.41
Refugee views homoph	0.36	0.33	0.08
Interest homoph	0.56	0.52	0.01

Overall, these comparisons paint a picture of villagers using their social network as one source of discussion partners. They do not necessarily discuss the topic with everyone, nor do they necessarily select among their network neighbors at random. Alters in the network who see refugees as a pressing issue are more likely to be discussion partners. Respondents also seem to seek out their alters with whom they agree on the level of importance of the topic and whose views align (whether they are positive or negative). Other attributes of the alter and bases for homophily do not distinguish the used from the unused links (Table 5).

5 Conclusion

Villagers in rural Uganda have social networks with four quite different layers when measured in terms of shared meals, regular homestead visits, gossip partners, and borrowing sources. When these villagers are presented with new information, in this case about the experiences of refugees, they do turn to some of these network neighbors to discuss it. Not everyone they turn to is a network neighbor in one of these layers, and not every network neighbor is selected as a discussion partner. The visits layer is the most popular choice— alters selected as discussion partners are more frequently linked to the respondent in the visit layer across the twelve villages, though this layer is also the most dense.

The choice of discussion partner from among the network neighbors appears to be orthogonal to the occupation, refugee experience, and attitudes towards

³ The p-value reports the result of a two-tailed t-test comparing links used with links not used in terms of the link attribute in question.

refugees of the alter. It also appears orthogonal to shared language, religion, or personal refugee status. Instead, what distinguishes the network links used to discuss refugees is the level of importance that the person ascribes to the topic. Links to alters who find the issue more pressing are more likely to be used, and links to alters who agree with the respondent about the level of importance are also more likely to be used. Shared views about refugees—agreement on the extent to which refugees do or do not threaten the village’s way of life—also predicts link use to discuss refugees, though with less precision.

Overall, these findings paint a picture that in the context of new information about a topic salient to rural villagers in Uganda, social networks play an important role in discussing it. Shared views on the topic and its importance can pave the way for discussion, as can having alters who find the topic especially important. Some layers are used more than others, though all were used in all villages. That no one layer dominates the others suggests that these conversations were not particularly sensitive or rigidly tailored to a certain kind of relationship. The information that would spread as a result is unlikely to exhibit tie-specific diffusion, which indicates that aggregating the layers to understand the consequences of conversations such as these may not mask results to a great extent [21, 22].

Of course these results come from a single instance of network use—discussing new information about refugees—in a single context—rural Uganda. The more cases of networks in action that can be studied in more contexts, the better our understanding of the true role of multi-layer networks will be.

References

1. Aral, S., Van Alstyne, M.: The diversity-bandwidth trade-off. *Am. J. Sociol.* **117**(1), 90–171 (2011)
2. Atwell, P., Nathan, N.L.: Channels for influence or maps of behavior? A field experiment on social networks and cooperation. *Am. J. Polit. Sci.* **66**(3), 696–713 (2022)
3. Bandiera, O., Rasul, I.: Social networks and technology adoption in northern mozambique. *Econ. J.* **116**(514), 869–902 (2006)
4. Banerjee, A., Chandrasekhar, A.G., Duflo, E., Jackson, M.O.: The diffusion of microfinance. *Science* **341**(6144), 1236–1248 (2013)
5. Bianconi, G.: *Multilayer Networks: Structure and Function*. Oxford University Press (2018)
6. Boccaletti, S., et al.: The structure and dynamics of multilayer networks. *Phys. Rep.* **544**(1), 1–122 (2014)
7. Bramoullé, Y., Galeotti, A., Rogers, B.W.: *The Oxford Handbook of the Economics of Networks*. Oxford University Press (2016)
8. Cozzo, E., et al.: Clustering coefficients in multiplex networks (2013). arXiv preprint [arXiv:1307.6780](https://arxiv.org/abs/1307.6780)
9. De Domenico, M., Nicosia, V., Arenas, A., Latora, V.: Structural reducibility of multilayer networks. *Nat. Commun.* **6**(1), 6864 (2015)
10. Dickison, M.E., Magnani, M., Rossi, L.: *Multilayer Social Networks*. Cambridge University Press, Cambridge (2016)

11. Ferrali, R., Grossman, G., Platas, M., Rodden, J.: Peer effects and externalities in technology adoption: Evidence from community reporting in Uganda. SSRN (2018). <https://goo.gl/NcGSvV>
12. Gondal, N.: Multiplexity as a lens to investigate the cultural meanings of interpersonal ties. *Soc. Netw.* **68**, 209–217 (2022)
13. González-Bailón, S., Borge-Holthoefer, J., Rivero, A., Moreno, Y.: The dynamics of protest recruitment through an online network. *Sci. Rep.* **1**(1), 1–7 (2011)
14. Granovetter, M.S.: The strength of weak ties. *Am. J. Sociol.* **78**(6), 1360–1380 (1973)
15. Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. *J. Complex Netw.* **2**(3), 203–271 (2014)
16. Kremer, M., Miguel, E.: The illusion of sustainability. *Q. J. Econ.* **122**(3), 1007–1065 (2007)
17. Larson, J.M.: The weakness of weak ties for novel information diffusion. *Appl. Netw. Sci.* **2**(1), 1–15 (2017)
18. Larson, J.M., Lewis, J.I.: Ethnic networks. *Am. J. Polit. Sci.* **61**(2), 350–364 (2017)
19. Larson, J.M., Lewis, J.I.: Measuring networks in the field. *Polit. Sci. Res. Methods* **8**(1), 123–135 (2020)
20. Larson, J.M., Lewis, J.I., Rodriguez, P.L.: From chatter to action: how social networks inform and motivate in rural Uganda. *Br. J. Polit. Sci.* **52**(4), 1769–1789 (2022)
21. Larson, J.M., Rodríguez, P.L.: Sometimes less is more: when aggregating networks masks effects. In: Cherifi, H., Mantegna, R.N., Rocha, L.M., Cherifi, C., Miccichè, S. (eds.) *Complex Networks and Their Applications XI. COMPLEX NETWORKS 2016 2022. Studies in Computational Intelligence*, vol. 1077. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-21127-0_18
22. Larson, J.M., Rodriguez, P.L.: The risk of aggregating networks when diffusion is tie-specific. *Appl. Netw. Sci.* **8**(1), 21 (2023)
23. Light, R., Moody, J.: *The Oxford Handbook of Social Networks*. Oxford University Press (2020)
24. Maoz, Z.: Preferential attachment, homophily, and the structure of international networks, 1816–2003. *Confl. Manag. Peace Sci.* **29**(3), 341–369 (2012)
25. Szell, M., Lambiotte, R., Thurner, S.: Multirelational organization of large-scale social networks in an online world. *Proc. Natl. Acad. Sci.* **107**(31), 13636–13641 (2010)
26. Victor, J.N., Montgomery, A.H., Lubell, M.: *The Oxford Handbook of Political Networks*. Oxford University Press (2017)



Classification of Following Intentions Using Multi-layer Motif Analysis of Communication Density and Symmetry Among Users

Takayasu Fushimi^(✉) and Takumi Miyazaki

School of Computer Science, Tokyo University of Technology, Hachioji 192-0982,
Japan

{fushimity,c0b2016043}@edu.teu.ac.jp

Abstract. The information diffusion on social media shows no signs of stopping, and for many marketers, having influencers spread information has become a common advertising method. However, the use of social media has diversified, and the intentions behind following other users can vary greatly. Correspondingly, there are several patterns of communication on social media, and based on the density and symmetry of these communications, it is believed that one can infer the intentions of the users who follow others. In this study, we consider retweets, replies, and mentions, three types of communication in the context of follower relationships on Twitter, as a multi-layer graph. We propose multi-layer motifs by categorizing the edges, and we associate motif patterns with follower intentions to infer users' follow intents. Through experiments using real data, we confirm that our proposed multi-layer motifs can extract link patterns leading to follow intentions that would not be detectable using traditional single-layer motifs.

1 Introduction

In recent years, the process from product awareness to purchase has undergone significant changes due to the advancement of the Internet. While in the past, the common purchase journey involved becoming acquainted with a product through television commercials and advertisements before making a purchase, recently, social media has become the predominant catalyst for purchasing. Within social media, the emergence of users known as influencers has led to a substantial transformation in the way companies promote their products. Identifying these influencers, who hold considerable significance, from various social media platforms holds paramount importance for the analysis of corporate marketing strategies and information dissemination.

The number of users utilizing social networking services (SNS) has been steadily increasing year by year. Due to the diverse range of purposes for which SNS is used, it is believed that the intentions behind users following other users

have also become more varied. The following intentions include “gathering information,” “spreading information,” “wanting to become friends,” “bookmarking useful information,” “connecting with acquaintances,” “wanting approval,” and “discussing.”

Indeed, with different intentions, the roles played also differ. In other words, within the context of information dissemination, there are edges in the follower network through which information flows more readily, as well as those through which it doesn’t flow as effectively. Therefore, categorizing someone as an influencer solely based on a high number of followers would be a misconception. It’s important to recognize that the definition of an influencer goes beyond just follower count due to the diversity of motivations and the varying effectiveness of information flow within follower networks. Hence, as a web marketer, it can be said that in order to identify influencers who genuinely help spread the desired information, it’s essential to discern the individual follower intentions and create a categorized graph based on them. By doing so, you can effectively extract influencers who align with your content and objectives, going beyond superficial follower metrics. This approach acknowledges the nuanced nature of influencer identification and helps ensure that the chosen influencers are more likely to authentically amplify the intended message.

Therefore, in this study, we focus on Twitter as the subject and aim to classify follower edges by examining the density and symmetry of diverse communications occurring within the follower network. By doing so, we attempt to understand and categorize the various types of interactions taking place on the platform based on the relationships between users in the follower network. This approach allows us to gain insights into the nature of communication patterns and potentially identify influencers who align with specific communication styles or objectives.

In the field of network science, there exists a technique called motif counting, which involves classifying edges or subgraphs and understanding network characteristics. Motif counting focuses on counting specific-shaped small subgraphs, referred to as motifs, within a network. This concept originated from Milo et al.’s research [6] and has been studied extensively in various fields over the years [1, 8]. Research has actively expanded the concept of motifs, and one such extension involves the analysis of motif roles based on structural equivalence in directed 3-node motifs. This extension defines the roles of individual nodes within motifs and has been utilized to derive analytical results [5, 7].

In this study, we propose an edge classification methodology within the context of a multi-layered graph that considers relationships between nodes through multiple communication edges such as mentions and replies. This approach differs from conventional motifs that target graphs without layers or graphs that don’t distinguish between layers. Here, we recognize layers as representations of various communication edges and their relationships between nodes. This multi-layered graph framework allows us to capture the complexity of interactions more comprehensively, offering a novel approach to edge classification.

2 Related Work

We will overview some related work in terms of classification of follow intention, analysis of Twitter communication, and motif analysis.

2.1 Follow Intention Classification

Tanaka et al. attempted to classify the intention that Twitter users follow others such as seeking information, engaging in personal communication, or staying updated on content by celebrities or influencers [11]. To this end, the work introduces a classification scheme that categorizes Twitter follow links based on users' motivations. The scheme consists of three primary dimensions: user-orientation, content-orientation, and relationship type (mutuality). As a result, the study found that user-orientation and content-orientation are correlated, suggesting that users who follow others for personal engagement also tend to follow for content-related reasons. Content-oriented follows are prevalent, even among users who primarily use Twitter for personal communication. A notable proportion of follow links lacked a clear and identifiable intention.

Takemura et al. proposed an approach to classify Twitter follow links [10]. The classification is based on three axes: user-orientation, content-orientation, and mutuality. The combination of these axes is designed to comprehensively categorize the diverse intentions of Twitter followers. To enhance understanding and classification of follow links, the research developed classifiers using various features related to followers, followees, and their interactions. Additionally, the paper discusses a method for categorizing Twitter lists into information lists and community lists, considering the types of accounts included in those lists. The experiments revealed that no single feature is a dominant discriminator for follow link classification, and the accuracy of classifying follow links for information-seeking users was higher compared to communication-oriented users.

These studies focus on communication history and its reciprocity, but they differ from our study in that they use list information. Another difference is that these studies do not comprehensively utilize the communication methods of "retweets," "replies," and "mentions." Furthermore, when training a support vector machine, there is a problem in that a large amount of manual training data is required.

2.2 Twitter Communication Analysis

Kato et al. centers its attention on three fundamental functions within the Twitter platform: favoriting tweets, following other users, and mentioning users in tweets [4]. These functions are essential for interaction and engagement on the platform. The research employs network analysis techniques to investigate the relationships and structures formed by these Twitter functions. It explores how these functions connect users and shape the Twitter community. The study examines the patterns of favoriting tweets and identifies users who tend to receive more favorites. It explores how favoriting behavior can reflect popularity

and engagement with content. The research delves into the patterns of following other users on Twitter. It examines factors that influence users' decisions to follow others, such as common interests or mutual connections. Mentioning other users in tweets is analyzed to understand how these interactions form connections and relationships on Twitter. It explores the dynamics of conversations and mentions within the platform. The insights gained from this network analysis have implications for understanding user engagement, content popularity, and network structures on Twitter. They can inform strategies for content creators, marketers, and platform developers. Like our study, it targets multiple communication histories on Twitter, but differs from our study in that it focuses on the characteristics of the entire graph and analyzes three types of communication separately.

2.3 Motif Analysis

Research is being conducted to use motifs to infer the roles and meanings that can be understood from the structure [5,7,9]. Przulj constructed a vector of 73 kinds of orbits (motif-based roles) obtained from 2- to 5-node graphlets and attempted to quantify the similarity among graphs or nodes [9]. McDonnell et al. proposed a transformation matrix from motif-frequency vector to role-frequency vector to efficiently compute the number of roles for each node or the whole graph [5]. Onishi et al. have proposed a method for estimating the role of each node based on its appearance position in a directed three-node motif, targeting intercompany transaction networks [7]. As a result, the authors claim that they were able to partition nodes into economically meaningful groups. This study is similar to our study in that it estimates the meaning that can be seen from local structures called motifs. This research differs from ours in that it does not use the concept of "communication," which is what this research calls the symmetry of transactions between companies.

3 Proposed Method

In this study, we propose a method to count the number of communication times per unit time for each kind of communication on the follow network, and to classify the following edges based on the symmetry.

3.1 Symmetry of Communication Density

We consider the follow relation graph $G = (\mathcal{V}, \mathcal{E})$ consisting of a set of nodes representing users $\mathcal{V} = \{u_1, \dots, u_N\}$ and a set of directed edges representing follow relationships (FF) between users $\mathcal{E} = \{e_1 = (u, v), \dots, e_M\} \subset \mathcal{V} \times \mathcal{V}$. Here, we define $\mathcal{R} = \{\{u, v\}; (u, v) \in \mathcal{E} \vee (v, u) \in \mathcal{E}\}$ as the set of node pairs that have a following relationship regardless of whether they are unidirectional or bidirectional.

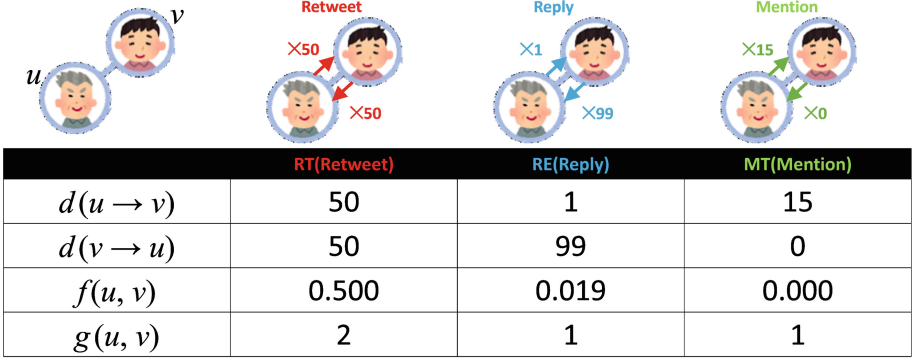


Fig. 1. Toy example

In Twitter, there are Retweet (RT), Reply (RE), and Mention (MT) as communication through the follow edge. For these three types of communication, we measure the number of communications per unit time, that is, the communication density, and quantify the symmetry. For two nodes, u and v , in a certain follow relationship, the communication density from u to v is denoted as $d(u \rightarrow v)$, and from v to u as $d(v \rightarrow u)$. The symmetry of communication density between the two nodes is defined using the harmonic mean as follows:

$$f(u, v) = \frac{2}{\frac{d(u \rightarrow v) + d(v \rightarrow u)}{d(u \rightarrow v)} + \frac{d(u \rightarrow v) + d(v \rightarrow u)}{d(v \rightarrow u)}}.$$

When the ratio of communication density is imbalanced, the value of the harmonic mean tends to approach 0, while it approaches 0.5 when balanced. For instance, if $d(u \rightarrow v) = 50$ and $d(v \rightarrow u) = 50$, then $f(u, v) = 0.500$, if $d(u \rightarrow v) = 1$ and $d(v \rightarrow u) = 99$, then $f(u, v) = 0.019$ and if $d(u \rightarrow v) = 15$ and $d(v \rightarrow u) = 0$, then $f(u, v) = 0.000$ (See Fig. 1). This value is multiplied by 2 and added by 1 then rounded to achieve values of 1 or 2:

$$g(u, v) = \begin{cases} 0 & \text{if } d(u \rightarrow v) + d(v \rightarrow u) = 0 \\ \lceil 2f(u, v) + 1 \rceil & \text{otherwise} \end{cases}$$

Consequently, $g(u, v) = 0$ indicates no communication, $g(u, v) = 1$ represents unidirectional communication, and $g(u, v) = 2$ signifies bidirectional communication (See Fig. 2).

3.2 Multi-layer Motif

The three types of communication are each classified into three categories based on symmetry. For the categorization of follow relationships into either “unidirectional” or “bidirectional,” follow relationships are classified into 54 categories

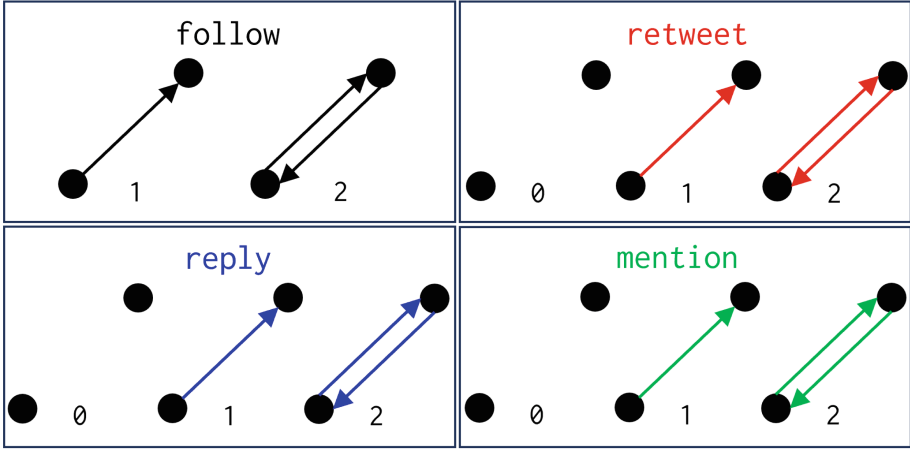


Fig. 2. Encoding symmetry of communication density

as a result of $2 \cdot 3^3$ possible combinations. In reality, there are 81 patterns since it's possible to retweet or reply without a follow relationship. However, since the focus of this study is on classifying follow intentions, “no follow relationship” is excluded in this case. Figure 3 depicts all the patterns of directed 2-node multi-layer motifs based on the symmetry of communication density defined in this study. Each pattern is represented in ternary notation, taking values of 0, 1, or 2:

$$\mathbf{g}(u, v) = [g_{FF}(u, v), g_{RT}(u, v), g_{RE}(u, v), g_{MT}(u, v)]_{(3)}.$$

Every user pair $\{u, v\} \in \mathcal{R}$ in the follow relationships is categorized into one of the 54 patterns. Based on this categorization, the frequencies of the 54 patterns are counted within each network or community:

$$\mathbf{h}(\mathcal{R}) = [|\{r; \mathbf{g}(r) = k\}|]_{27 \leq k \leq 80}.$$

This classification allows for the categorization of follow intentions and the characterization of the target network.

3.3 Follow Intention Classification

The patterns of multi-layer motifs essentially depict ways of communication. From these communication patterns, an attempt is made to classify the follow intentions, which is the main objective of this study. In order to classify follow intentions, we consider what purposes or meanings the utilized forms of communication entail. Then, for a subset of the defined 54 motif patterns in the previous section, we associate follow intentions. In this study, four follow intentions are defined: “Acquaintance,” “Information gathering,” “Interest in user,” and “Discussion.”

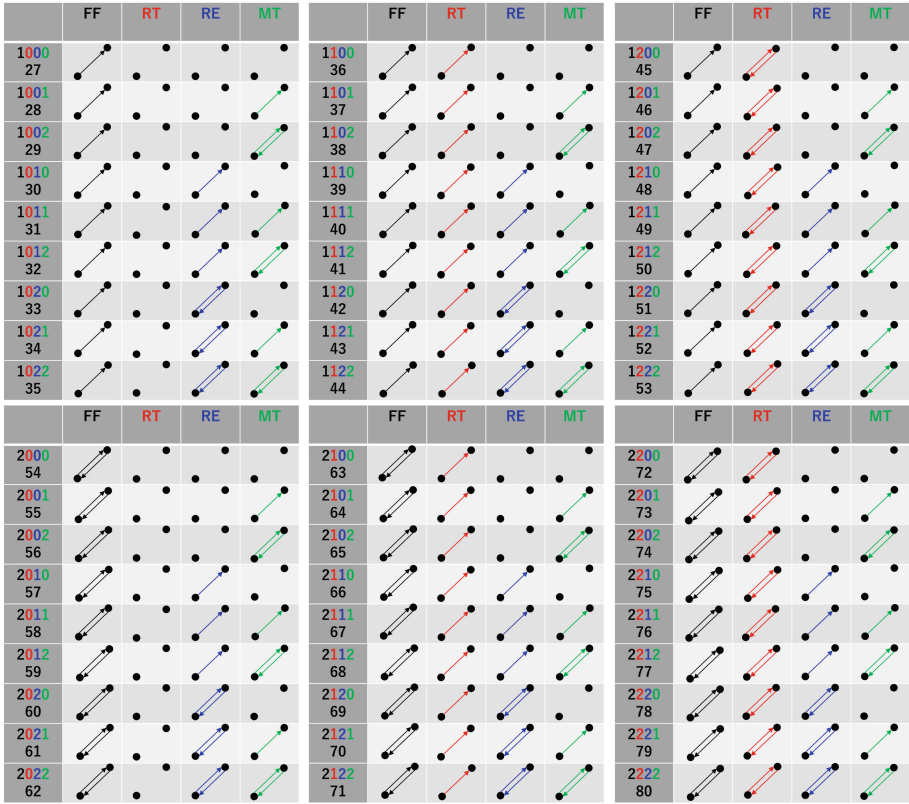


Fig. 3. Multi-layer motifs based on symmetry of communication density

The three types of communication-retweets, replies, and mentions-reflect the ways users are trying to establish relationships with others. Each of their purposes and meanings are as follows:

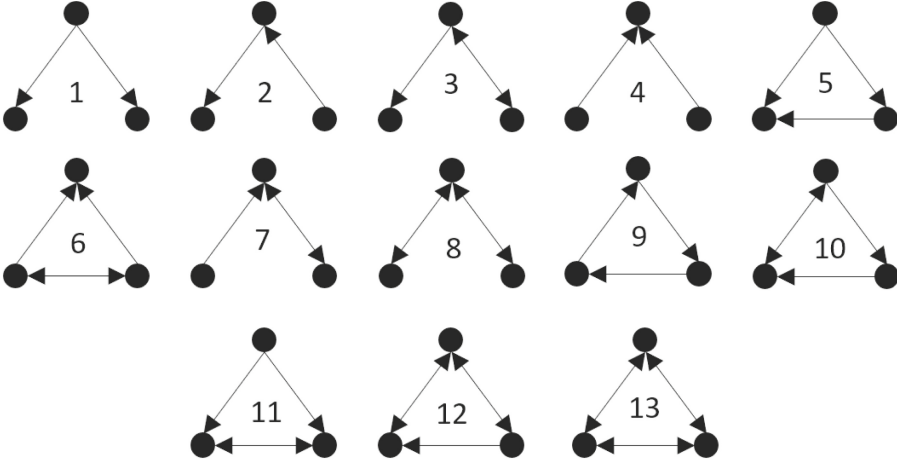
- Retweets: Users retweet to disseminate specific information or opinions, aiming to expand their influence, or to collect tweets on their timeline as a form of note-taking.
- Replies: Users engage in replies to exchange opinions and participate in discussions on particular topics, aiming to deepen connections through conversations and share opinions and information.
- Mentions: Mentions prioritize interactions and connections, aiming for one-on-one dialogues with individual users. They are often used for commenting, questioning, or sharing opinions on specific tweets or topics.

From these purposes, we consider how they align with the follow intentions of “Acquaintance,” “Information gathering,” “Interest in user,” and “Discussion.”

In the case of a follow relationship between “Acquaintances,” it’s common for the relationship to be bidirectional, and conversations often occur using replies

Table 1. Correspondence between follow intentions and multi-layer motifs

follow intention	FF	RT	RE	MT	motifs
Acquaintances	2	Any	2	2	62, 71, 80
Information gathering	1	1	0	1 or 2	37, 38
Interest in users	1	1	1 or 2	1 or 2	40, 41, 43, 44
Discussion	Any	1	1	1	40, 67

**Fig. 4.** Existing single-layer motif

and mentions. For follow relationships with the intention of “Information gathering,” it’s often a unidirectional follow relationship, and users post tweets using retweets and mentions to keep information on their timelines. In the case of following due to “Interest in users” such as celebrities, the follow relationship is typically unidirectional. Users unilaterally retweet tweets from the user of interest and engage in communication using replies and mentions. Some celebrities might also respond as part of fan service. For follow relationships with the purpose of “Discussion,” users reply to or mention other users’ tweets to engage in discussions on specific topics. However, bidirectional follow relationships are not always common. Therefore, we associate the multi-layer motifs as Table 1.

4 Experimental Evaluation

4.1 Dataset

In this study, we use the Higgs Twitter Dataset¹ [3]. This dataset focuses on messages related to the discovery of the Higgs boson posted on Twitter between July

¹ <https://snap.stanford.edu/data/higgs-twitter.html>.

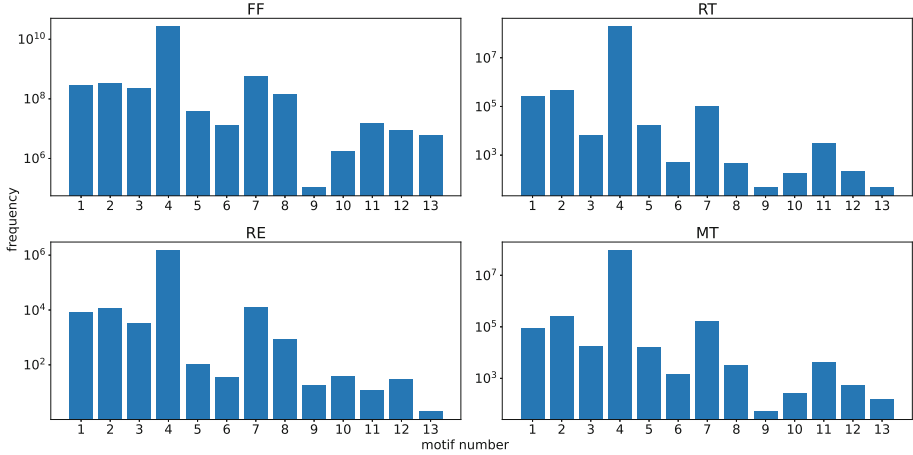


Fig. 5. Appearance frequency of the existing motifs (triad)

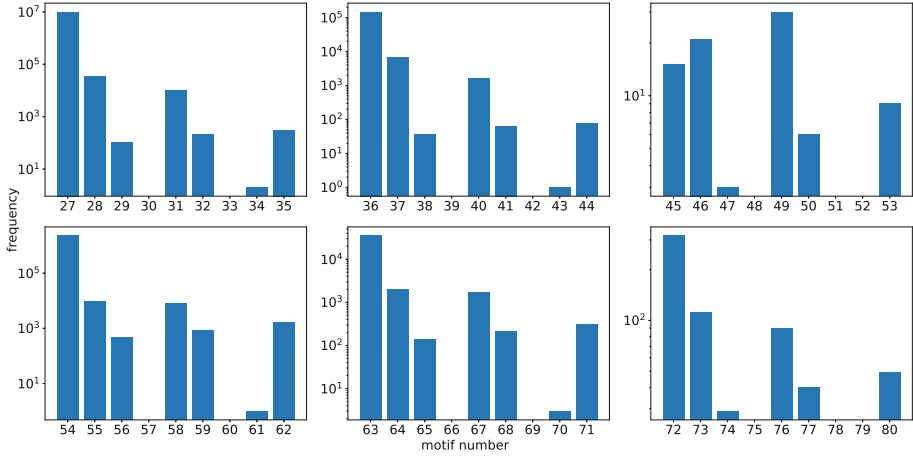


Fig. 6. Appearance frequency of the multi-layer motifs (diad)

1st and 7th, 2012. It includes messages posted by users during that time period, as well as interactions between users, such as retweets, replies, and mentions. The dataset's size is as follows: 456,626 users; 14,855,842 follow relationships; 328,132 retweet relationships; 32,523 reply relationships; 150,818 mention relationships;

4.2 Results

Figure 5 illustrates the frequency distribution of existing motif patterns for each graph of follow-relationships, retweet, reply, and mention. Observing the motif distributions for the four graphs, we notice that the motifs 4, 1, 2 and 7 are consistently high. According to the Fig. 4, motif 4 represents a structure in which

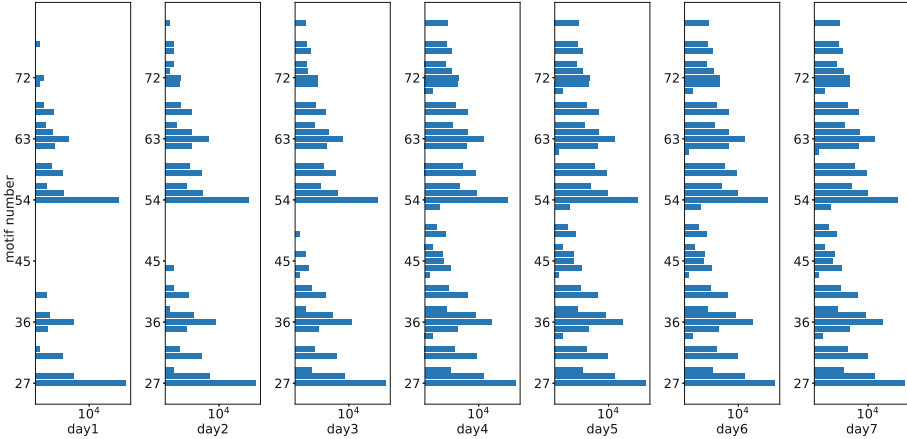


Fig. 7. Cumulative (daily) layer motif frequency distribution

general users unilaterally send follows, retweets, replies, and mentions to some hub users (such as experts). Motif 1 also represents a structure in which a general user unilaterally sends follows, retweets, replies, and mentions to multiple users (such as experts). Motif 2 represents a structure in which information flows like retweets or replies are chained, and information is spread. Motif 7 represents a structure in which two-way communication between acquaintances and one-way communication with hub users are mixed. On the contrary, cycles such as motifs from number 9 onwards are characteristically less frequent. Structures like Motif 13, where three nodes are mutually connected, do appear to some extent in the follow relationship graph. However, the existing motif analysis does not provide insight into the extent of communication occurring over these structures and whether it is unidirectional or bidirectional. As such, with existing motifs, since each layer is considered separately, it is not possible to conduct a detailed analysis regarding follow intentions from communication over each follow relationship.

Figure 6 displays the frequency distribution of proposed motif patterns for the higgs multi-layer graph. It consists of six subfigures corresponding to the six blocks in Fig. 3. Upon observing the figure, it becomes evident that 1st, 2nd, and 5th motifs within each block are notably frequent. The first multi-layer motifs including 27, 36, 45, 54, 63, and 72 are the “just following” relationship or “occasional unilateral retweet” relationship that is often observed on Twitter as a whole. The second and fifth multi-layer motifs including 28, 31, 37, 40, 55, 58, and all that, exhibit structures involving one-way mentions. In constructing the Higgs dataset graph, when a mention tweet is retweeted, it is also treated as a mention link. The frequent occurrence of these motifs is thought to be due to general users collecting information and bookmarking information by retweeting discussions (MTs) between experts. From the results of multi-layer motif frequency, it can be inferred that in this multilayer graph, there are many follow intentions focused on ‘discussion’ as indicated by the frequent use of RE

and MT, and ‘information gathering,’ as indicated by the frequent use of RT without RE.

Next, we examined how the frequency of motifs changes over time. Figure 7 depicts the motif occurrences, with the horizontal axis representing the motif count and the vertical axis denoting motif number, arranged by daily intervals from July 1st to 7th. For instance, the leftmost subfigure represents the frequency of multilayer motifs of communications on July 1st, while the second subfigure illustrates the frequency of such motifs over the two days, July 1st and 2nd. That is to say, the rightmost subfigure shows that over the seven days and aligns with Fig. 6.

Upon observing Fig. 7, it becomes evident that there is not a large variation day by day, and the counts linearly increase across all days. Additionally, 1st, 2nd, and 5th motifs, as mentioned in the previous experiment, continue to appear frequently compared to other motifs. This trend remains consistent even when considering not only daily but also hourly and half-day intervals. From these results, it can be concluded that the Higgs multi-layer graph exhibits a time-invariant characteristic, independent of the time intervals used to measure communication density. Specifically, it is characterized by the frequent occurrence of communication-based on the intentions of ‘discussion’ and ‘information gathering.’ This fact is consistent with the claim in [2].

5 Conclusion

In influencer marketing, the market size is expanding year by year due to the ability to create empathetic and persuasive PR from the consumer’s perspective. Therefore, accurately extracting influencers is a crucial research topic. While it’s possible to disseminate information by following other users, the ease of dissemination varies depending on the intention. To achieve higher accuracy in influencer extraction, we classify follow intentions. Hence, we proposed a method to classify follow intentions using layer motif patterns based on communication density in Twitter data.

Evaluation experiments using the Higgs Twitter dataset revealed that existing single-layer motifs involve a lot of information diffusion, convergence, and lateral flow in each communication. However, a detailed analysis of a single follow edge was not possible. On the other hand, with the multi-layer motif we proposed, we were able to classify follow intentions by focusing on a single follow edge and its communication density. Furthermore, the Higgs dataset exhibited time-invariant characteristics, as the same follow intentions were extracted regardless of the time window used to measure “communication density” (e.g., one week, one day, one hour).

Although we currently define four intentions, we plan to explore the possibility of expanding them or investigating alternative methods of definition. While the data in this study showed a prevalence of intentions related to “Discussion” and “Information Gathering,” it is important to validate these findings with other datasets.

Acknowledgments. This material is based upon work supported by JSPS Grant-in-Aid for Scientific Research (C) (JP22K12279).

References

1. Ahmed, N.K., Neville, J., Rossi, R.A., Duffield, N.: Efficient graphlet counting for large networks. In: 2015 IEEE International Conference on Data Mining, pp. 1–10 (2015)
2. Braha, D., Bar-Yam, Y.: Time-dependent complex networks: dynamic centrality, dynamic motifs, and cycles of social interactions. In: Gross, T., Sayama, H. (eds.) *Adaptive Networks. Understanding Complex Systems*. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01284-6_3
3. De Domenico, M., Lima, A., Mougél, P., Musolesi, M.: The anatomy of a scientific rumor. *Sci. Rep.* **3**, 2980 (2013)
4. Kato, S., Koide, A., Fushimi, T., Saito, K., Motoda, H.: Network analysis of three twitter functions: favorite, follow and mention. In: Richards, D., Kang, B.H. (eds.) *PKAW 2012. LNCS (LNAI)*, vol. 7457, pp. 298–312. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32541-0_26
5. McDonnell, M.D., Yaveroglu, O.N., Schmerl, B.A., Iannella, N., Ward, L.M.: Motif-role-fingerprints: the building-blocks of motifs, clustering-coefficients and transitivities in directed networks. *PLOS ONE* **9**(12), 1–25 (2014). <https://doi.org/10.1371/journal.pone.0114503>
6. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. *Science* (New York, N.Y.) **298**(5594), 824–827 (2002)
7. Ohnishi, T., Takayasu, H., Takayasu, M.: Network motifs in an inter-firm network. *J. Econ. Interac. Coord.* **5**(2), 171–180 (2010)
8. Pinar, A., Seshadhri, C., Vishal, V.: ESCAPE: efficiently counting all 5-vertex subgraphs. In: *Proceedings of the 26th International Conference on World Wide Web*, pp. 1431–1440. WWW 2017, Republic and Canton of Geneva, CHE (2017). <https://doi.org/10.1145/3038912.3052597>
9. Pržulj, N.: Biological network comparison using graphlet degree distribution. *Bioinformatics* **23**(2), 177–183 (2007). <https://doi.org/10.1093/bioinformatics/btl301>
10. Takemura, H., Tanaka, A., Tajima, K.: Classification of twitter follow links based on the followers’ intention. In: *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pp. 1174–1180. SAC 2015, Association for Computing Machinery, New York, NY, USA (2015). <https://doi.org/10.1145/2695664.2695940>
11. Tanaka, A., Takemura, H., Tajima, K.: Why you follow: a classification scheme for twitter follow links. In: *Proceedings of the 25th ACM Conference on Hypertext and Social Media*. pp. 324–326. HT 2014, Association for Computing Machinery, New York, NY, USA (2014). <https://doi.org/10.1145/2631775.2631790>



Generalized Densest Subgraph in Multiplex Networks

Ali Behrouz^(✉) and Farnoosh Hashemi

Cornell University, Ithaca, NY, USA
{ab2947, Sh2574}@cornell.edu

Abstract. Finding dense subgraphs of a large network is a fundamental problem in graph mining that has been studied extensively both for its theoretical richness and its many practical applications over the last five decades. However, most existing studies have focused on graphs with a single type of connection. In applications such as biological, social, and transportation networks, interactions between objects span multiple aspects, yielding multiplex graphs. Existing dense subgraph mining methods in multiplex graphs consider the same importance for different types of connections, while in real-world applications, one relation type can be noisy, insignificant, or irrelevant. Moreover, they are limited to the edge-density measure, unable to change the emphasis on larger/smaller degrees depending on the application. To this end, we define a new family of dense subgraph objectives, parametrized by two variables p and β , that can (1) consider different importance weights for each relation type, and (2) change the emphasis on the larger/smaller degrees, depending on the application. Due to the NP-hardness of this problem, we first extend the FirmCore, k -core counterpart in multiplex graphs, to layer-weighted multiplex graphs, and based on it, we propose two polynomial-time approximation algorithms for the generalized densest subgraph problem, when $p \geq 1$ and the general case. Our experimental results show the importance of considering different weights for different relation types and the effectiveness and efficiency of our algorithms.

Keywords: Multiplex Networks · Dense Subgraphs · FirmCore · p -mean

1 Introduction

Multiplex (ML) networks [24] have become popular in various applications involving complex networks such as social, transportation, and biological networks. These networks involve interactions between objects that span different aspects. For instance, interactions between individuals can be categorized as social, family, or professional, and professional interactions can vary depending on the topic. ML networks allow nodes to have interactions in multiple relation types and represent the graph of each relation type as a layer in the network.

Detecting Dense structures in a graph has become a key graph mining primitive with a wide range of applications [13, 15, 19]. The common method for identifying dense subgraphs is to formulate an objective function (called density) that

captures the density of each node set within a graph and then solve it via combinatorial optimization methods [14, 18, 20]. While the problem of finding the densest subgraph in simple graphs is a well-studied problem in the literature and its recent advancements bring the problem close to being fully resolved [25], extracting dense subgraphs from ML networks recently attracts attention [6, 16, 22]. Due to the complex interactions between nodes in ML networks, the definition of edge density is challenging. To this end, several studies [16, 22, 23] introduced new density objective functions to capture complex dense subgraphs; however, in practice, it can be challenging to evaluate tradeoffs between density measures and decide which density is more useful. Accordingly, there is a lack of a unified framework that can generalize all the existing density measures to formalize the tradeoff between them.

One of the main advantages of ML networks is their ability to provide complementary information by different relation types [22]. That is, some dense subgraphs can be missed if we only look at one relation type or the aggregated network [22]. However, taking advantage of this complementary information is challenging as in real-world applications, different relation types have different importance (e.g., some layers might be noisy/insignificant [2, 3, 16, 22], or have different roles in the applications [4, 5, 7]). Existing dense subgraph models treat relation types equally, which means noisy/insignificant layers (or less important layers) are considered as important as other layers, causing suboptimal performance and missing some dense subgraphs (we support this claim in Sect. 4).

To overcome the above challenges, we introduce a new family of density objectives in ML networks, p -mean multiplex densest subgraph (p -mean MDS), that: ① is able to handle different weights for layers, addressing different importance of relation types; ② given a parameter p , inspired by Veldt et al. [28], it uses p -mean of node degrees in different layers. This design gives us the flexibility to emphasize smaller/larger degrees and allows us to uncover a hierarchy of dense subgraphs in the same ML graph; ③ unifies the *existing* definition of density in ML networks, which allows evaluating the tradeoffs between them. The multiplex p -mean density objective uses parameter β to model the trade-off between high density and the cumulative importance of layers exhibiting the high density, and uses parameter p to define p -mean of node degrees within a subgraph as a measure of high density (we formally define it in Sect. 3). Inspired by FirmCore structure [22], we further extend the concept of k -core to weighted layer ML networks and define weighted (k, λ) -FirmCore ((k, λ) -GFirmCore) as a maximal subgraph in which every node is connected to at least k other nodes within that subgraph, in a set of layers with cumulative importance of at least λ . We discuss that given λ , weighted FirmCore has linear time decomposition in terms of the graph size, and can provide two tight approximation algorithms for the two cases of the p -mean MDS problem when ① $p \geq 1$ and ② the general case.

2 Related Work and Background

Given the wide variety of applications for dense subgraph discovery [13, 15, 19], several variants of the densest subgraph problem with different objective

functions have been designed [9, 14, 20, 28]. Recently, Veldt et al. [28] unifies most existing density objective functions and suggests using p -mean of node degrees within the subgraph as its density. In this case, when $p = 1$, $p = -\infty$, and $p = 2$ we have the traditional densest subgraph problem, maximal k -core, and F-density [14], respectively. Despite the usefulness of the family of p -mean density objectives, they are limited to simple graphs and their extension to ML networks is not straightforward.

In ML networks, Jethava and Beerenwinkel [23] formulate the densest common subgraph problem and develop a linear-programming formulation. Azimi-Tafreshi et al. [1] propose a new definition of core, \mathbf{k} -core, over ML graphs. Galimberti et al. [16] propose algorithms to find all possible \mathbf{k} -cores, and generalized the formulation of Jethava and Beerenwinkel [23] by defining the density of a subgraph in ML networks as a real-valued function $\rho : 2^V \rightarrow \mathbb{R}^+$:

$$\rho(S) = \max_{\hat{L} \subseteq L} \min_{\ell \in \hat{L}} \frac{|E_\ell[S]|}{|S|} |\hat{L}|^\beta, \quad (1)$$

where $E_\ell[S]$ is the number of internal edges of S in layer ℓ , and $\beta \geq 0$ is a real number. They further propose a core-based $\frac{1}{2|\hat{L}|^\beta}$ -approximation algorithm. However, their algorithm takes exponential time in the number of layers, rendering it impractical for large networks (see Sect. 4). Recently, Hashemi et al. [22] introduce FirmCore, a new family of dense subgraphs in ML network, as a maximal subgraph in which every node is connected to at least k other nodes within that subgraph, in each of at least λ individual layers.

Although the densest FirmCore approximates function $\rho(\cdot)$, which its optimization is NP-hard [17], with provable guarantee, it is limited to unweighted layer ML networks, missing some dense structures. Moreover, its approximation guarantee is limited to the objective function defined by Galimberti et al. [16], and its performance in our p -mean MDS is unexplored. For additional related work on the densest subgraph problem, we refer to the recent survey by Lanciano et al. [25].

3 p -Mean Multiplex Densest Subgraph

We let $G = (V, E, L, \mathbf{w})$ denote an ML graph, where V is the set of nodes, L is the set of layers, $E \subseteq V \times V \times L$ is the set of edges, and $\mathbf{w}(\cdot) : L \rightarrow \mathbb{R}^{\geq 0}$ is a function that assigns a weight to each layer. The set of neighbors of node $v \in V$ in layer $\ell \in L$ is denoted $N_\ell(v)$ and the degree of v in layer ℓ is $\deg_\ell(v) = |N_\ell(v)|$. For a set of nodes $H \subseteq V$, $G_\ell[H] = (H, E_\ell[H])$ shows the subgraph of G induced by H in layer ℓ , and $\deg_\ell^H(v)$ is the degree of v in this subgraph. We sometimes use $G_\ell[V]$ and $E_\ell[V]$ as G_ℓ and E_ℓ , respectively.

As discussed in [16], the density in ML networks should be modeled as a trade-off between the high density and the number of layers exhibiting the high density. Here, we use this intuition and first use p -mean density to measure the density of the subgraph in each layer, i.e.,

$$\Omega_\ell(S) = \left(\frac{1}{|S|} \sum_{u \in S} \text{deg}_\ell(u)^p \right)^{1/p}, \quad (2)$$

and then multiply it by the importance of the layer exhibiting this density:

$$\Xi_\ell(S) = \Omega_\ell(S) \mathbf{w}(\ell). \quad (3)$$

Based on this definition of density we define the p -mean MDS problem as follows:

Problem 1 (p -mean Multiplex Densest Subgraph). Given an ML graph $G = (V, E, L, \mathbf{w})$, real numbers $\beta \geq 0$ and $p \in \mathbb{R} \cup \{+\infty, -\infty\}$, and a real-valued function $\rho : 2^V \rightarrow \mathbb{R}^+$ defined as:

$$\rho(S) = \max_{\hat{L} \subseteq L} \min_{\ell \in \hat{L}} \Xi_\ell(S) \left(\sum_{\ell' \in \hat{L}} \mathbf{w}(\ell') \right)^\beta, \quad (4)$$

find a subset of vertices $S^* \subseteq V$ that maximizes ρ function.

Note that given layer weights $\mathbf{w}(\ell)$, we aim to solve a max-min problem over $\Xi_\ell(S)$. Also, given a layer ℓ , maximizing the $\Xi_\ell(S)$ is equivalent to maximizing $\Omega_\ell(S)^p$ for $p > 0$ and minimizing $\Omega_\ell(S)^p$ for $p < 0$. Therefore, for the sake of simplicity, in the following we aim to optimize (maximize or minimize) $\Omega_\ell(S)^p$. Following, we use $\Delta_\ell(S/\{u\}) = \Omega_\ell(S)^p - \Omega_\ell(S/\{u\})^p$, to denote the difference that removing a node u can cause to the density of layer ℓ . When $p = 1$ and $\mathbf{w}(\cdot) = 1$, the p -mean MDS problem reduces to ML densest subgraph problem [16].

3.1 Generalized FirmCore Decomposition

Next, inspired by the success FirmCore [22] in approximating the ML densest subgraph problem, we generalized it to layer-weighted ML networks and design an algorithm to find all existing FirmCores. In Sect. 3.2, we use the generalized FirmCore to approximate Problem 1.

There are two steps to generalize this concept: ① FirmCore treats all layers the same and consider the number of selected layers, accordingly. However, generalized FirmCore needs to consider the cumulative importance of selected layers, to take advantage of layer weights. ② In simple densest subgraph problem (i.e., $p = 1$), each node in a subgraph contributes the same to the denominator of the density function (i.e., subgraph size $|S|$), while each node's contribution to the numerator (i.e., number of edges) is as much as its degree. Traditionally, core structures attracts attention to approximate the densest subgraph as they provide lower bound for the minimum degree. However, in the p -mean density, the contribution of each node does not equal to its degree. As we discussed above,

removing each node makes $\Delta_\ell(S/\{u\}) = \Omega_\ell(S)^p - \Omega_\ell(S/\{u\})^p$ difference to the numerator of the $\Omega_\ell^p(S)$. Accordingly, in the general case $p \in \mathbb{R} \cup \{-\infty, \infty\}$, we want our generalized FirmCore to provide lower bound for the $\Delta_\ell(S/\{u\})$.

Definition 1 (Generalized FirmCore) *Given an ML graph G , a non-negative real-value threshold λ , an integer $k \geq 0$, and $p \in \mathbb{R} \cup \{-\infty, +\infty\}$, the (k, λ, p) -GFirmCore of G is a maximal subgraph $H = G[C_k] = (C_k, E[C_k], L)$ such that for each node $v \in C_k$ there are some layers with cumulative importance of at least λ (i.e., $\exists\{\ell_1, \dots, \ell_s\} \subseteq L$ with $\sum_{i=1}^s \mathbf{w}(\ell_i) \geq \lambda$) such that $\Delta_\ell(S/\{u\}) \geq k$, for $1 \leq i \leq s$.*

Proposition 1 *When $p = 1$ and $\mathbf{w}(\ell_i) = 1$ for all $\ell_i \in L$, (k, λ, p) -GFirmCore is equivalent to the (k, λ) -FirmCore [22].*

Proposition 2 (Hierarchical Structure) *Given a real-value threshold λ , an integer $k \geq 0$, and $p \in \mathbb{R} \cup \{-\infty, \infty\}$ the $(k+1, \lambda, p)$ -GFirmCore and $(k, \lambda+\epsilon, p)$ -GFirmCore of G are subgraphs of its (k, λ, p) -GFirmCore for any $\epsilon \in \mathbb{R}^+$.*

From now, to avoid confusion, when we refer to (k, λ) -GFirmCore, we assume that λ is maximal. That is, for at least one vertex u in (k, λ) -GFirmCore, there is a subset of layers with an exact summation of λ in which u has a degree not less than k . Next, we show that GFirmCore decomposition is strictly harder than the FirmCore decomposition, which is solvable in polynomial time, unless $P = NP$.

Theorem 1. *GFirmCore decomposition, which is finding all possible GFirmCores in an ML network, is NP-hard.*

Proof. Here we provide the proof sketch for the sake of space constraint. Given a sequence of layer weights $w_1, w_2, \dots, w_{|L|}$, the decision problem of whether there is a non-empty (k, λ, p) -GFirmCore can be simply reduced to the well-known NP-hard problem of the *Subset Sum* over $w_1, w_2, \dots, w_{|L|}$, as its YES (resp. NO) instance means there is (resp. is not) a subset of w_i s with summation of λ .

Algorithm. Here, we design a polynomial-time algorithm that finds all (k, λ, p) -GFirmCores for given λ and p . Given λ and p , we define the GFirmCore index of a node u , $\text{Gcore}_\lambda(u)$, as the set of all $k \in \mathbb{N}$, such that u is part of a (k, λ, p) -GFirmCore. For each node u in subgraph $G[H]$, we consider a vector $\Psi(u)$ that its ℓ -th element, $\Psi_\ell(u)$, shows $\Delta_\ell(H/\{u\})$'s in layer ℓ . We further define $\text{Top-}\lambda(\Psi(u))$ as the maximum value of k that there are some layers $\{\ell_1, \dots, \ell_t\}$ with a cumulative weight of at least λ in which $\Delta_\ell(H/\{u\}) \geq k$. To calculate the $\text{Top-}\lambda(\Psi(u))$, we can simply sort the vector $\Psi(u)$ and check if the cumulative weights of layers in which u has a $\Delta_\ell(H/\{u\})$ more than k is $\geq \lambda$ or not. This process takes $\mathcal{O}(|L| \log |L|)$ time. It is easy to see that u can be in at most (k, λ, p) -GFirmCore, where $k = \text{Top-}\lambda(\Psi(u))$. Accordingly, Algorithm 1 processes the nodes in increasing order of $\text{Top-}\lambda(\Psi(u))$. It uses a vector B of lists such that each element i contains all nodes with $\text{Top-}\lambda(\Psi(u)) = i$. This technique allows us to keep vertices sorted throughout the algorithm and to update each element in $\mathcal{O}(1)$ time. Algorithm 1 first initializes B with $\text{Top-}\lambda(\Psi(u))$ and then

Algorithm 1. Finding all (k, λ, p) -GFirmCores for a given λ

Input: An ML graph $G = (V, E, L, \mathbf{w})$, and a threshold $\lambda \in \mathbb{R}^{\geq 0}$ **Output:** GFirmCore index $\text{Gcore}_\lambda(v)$ for each $v \in V$

```

1: for  $v \in V$  do
2:    $I[v] \leftarrow \text{Top-}\lambda(\Psi(v))$ 
3:    $B[I[v]] \leftarrow B[I[v]] \cup \{v\}$ 
4: end for
5: for  $k = 1, 2, \dots, |V|$  do
6:   while  $B[k] \neq \emptyset$  do
7:     pick and remove  $v$  from  $B[k]$ 
8:      $\text{Gcore}_\lambda(v) \leftarrow k, N \leftarrow \emptyset$ 
9:     for  $(v, u, \ell) \in E$  and  $I[u] > k$  do
10:      update  $\Psi_\ell(u)$  and remove  $u$  from  $B[I[u]]$ 
11:      update  $I[u]$  and  $B[I[u]] \leftarrow B[I[u]] \cup \{u\}$ 
12:     end for
13:      $V \leftarrow V \setminus \{v\}$ 
14:   end while
15: end for

```

starts processing B 's elements in increasing order. If a node u is processed at iteration k , its Gcore_λ is assigned to k and removed from the graph. In order to remove a vertex from a graph, we need to update the degree of its neighbors in each layer, which leads to changing the $\text{Top-}\lambda(\Psi)$ of its neighbors and changing their bucket accordingly (lines 10–12). Note that it is simple to show that the above algorithm can find all (k, λ, p) -GFirmCores, given λ and p . That is, at the end of $(k - 1)$ -th iteration, each remaining nodes like u has $\text{Top-}\lambda(\Psi(u)) \geq k$ as we removed all nodes with $\text{Top-}\lambda(\Psi)$ less than k in the $(k - 1)$ -th iteration.

3.2 Approximation Algorithms

Algorithm 2 shows the pseudocode of the proposed approximation algorithm. Given a threshold α , we first construct a candidate set for the value of λ . To this end, we consider the set of summations of all possible subsets of layer weights with size $1 \leq s \leq \alpha$, denoted as \mathcal{M} . Next, we use Algorithm 1 for each $\lambda \in \mathcal{M}$, and then report the densest GFirmCore as the approximate solution. In our experiments, we observe that always $\alpha = 10$ results in a good approximate solution. Given p , let S_{SL}^* be the p -mean densest subgraph among all single-layer densest subgraphs, and ℓ^* denote its layer. Let C^* and S^* denote our found approximation solution and the optimal solution, respectively. Finally, we use \mathbf{w}^* , \mathbf{w}_{\min} , and \mathbf{w}_{\max} to refer to the summation of all layer weights, minimum weight, and maximum weight, respectively.

Lemma 1. *Let C be the (k, λ, p) -GFirmCore of G , we have:*

$$\rho(C) \geq \frac{k^{1/p}}{\mathbf{w}^{*1/p}} \times \max_{\hat{L} \subseteq \hat{L}} \left\{ \left(\lambda - \sum_{i=1}^{|\hat{L}|} \mathbf{w}(\ell_i) \right)^{1/p} \times \max_{\ell \in \hat{L}} \mathbf{w}(\ell) \times \left(\sum_{\ell \in \hat{L}} \mathbf{w}(\ell) \right)^\beta \right\} \quad (5)$$

$$\geq \frac{k^{1/p} \times \mathbf{w}_{\min}}{\mathbf{w}^{*1/p}} \times \max \left\{ \lambda^{1/p}, \lambda^{\beta/p} \right\}, \quad (6)$$

where \hat{L} is the first $|\hat{L}|$ -th element in sorted L with respect to the number of nodes like u with $\Psi_\ell(u) \geq k$ for $\ell \in \hat{L}$, and \mathbf{w}_{\min} is the smallest layer weights that contributed to C (i.e., removing it changes either k or λ).

Algorithm 2. Approximation algorithm for the p -mean MDS

Input: An ML graph $G = (V, E, L, \mathbf{w})$, a parameter $p \in \mathbb{R} \cup \{-\infty, \infty\}$, and parameter $\alpha \in \{1, \dots, L\}$.

Output: Approximation solution to p -mean MDS.

- 1: $\mathcal{M} \leftarrow$ summations of all possible subsets of layer weights with size $1 \leq s \leq \alpha$;
 - 2: **for** $\lambda \in \mathcal{M}$ **do**
 - 3: $\mathcal{Q}_\lambda \leftarrow$ find all (k, λ, p) -GFirmCore ▷ Using Algorithm 1
 - 4: $\hat{C}_\lambda \leftarrow$ calculate the density and find the densest (k, λ, p) -GFirmCore $\in \mathcal{Q}_\lambda$ $\rho(\cdot)$.
 - 5: **end for** **return** the densest subgraph among all \hat{C}_λ for $\lambda \in \mathcal{M}$.
-

Proof. By definition, each node $v \in C$ has at least $\Psi(u) \geq k$ in some layers with cumulative weights $\geq \lambda$, so based on the pigeonhole principle, there exists a layer ℓ' such that there are $\geq \frac{\lambda|C|}{\mathbf{w}^*}$ nodes like u that each has $\Psi_{\ell'}(u) \geq k$. So we have:

$$\Omega_{\ell'}(|C|) \geq \mathbf{w}(\ell') \times \left(\frac{k \times \frac{\lambda|C|}{\mathbf{w}^*}}{|C|} \right)^{1/p} = \mathbf{w}(\ell') \left(\frac{k \times \lambda}{\mathbf{w}^*} \right)^{1/p}.$$

Now, ignoring this layer, exploiting the definition of C , and re-using the pigeonhole principle, we can conclude that there exists a layer ℓ'' such that there are $\geq \frac{(\lambda - \mathbf{w}(\ell'))|C|}{\mathbf{w}^*}$ nodes like u that each has $\Psi_{\ell''}(u) \geq k$. By iterating this process, we can simply conclude the Inequality 6. Note that the last inequality is obtained from the first and last iterations of the above procedure.

Case 1: $p \geq 1$. Let C_{SL}^* be the $(p+1)^{1/p}$ approx solution for S_{SL}^* by [28] (it exists when $p \geq 1$), and $\mu = \min \Delta_{\ell^*}(C_{SL}^*)$. Since C_{SL}^* is the optimal obtained solution, removing a node cannot increase its p -mean density (if increases, then we find a better approx solution as it is certainly produced in the algorithm). Therefore, it is simple to see that $\Omega_{\ell^*}(S_{SL}^*)^p \leq \mathbf{w}(\ell^*)^p (p+1)\mu$. Based on the definition of μ and Δ , there is a non-empty (k^+, λ^+) -GFirmCore that $k^+ \geq \mu$. So we have $k^+ \geq \frac{\Omega_{\ell^*}(S_{SL}^*)^p}{\mathbf{w}(\ell^*)^p (p+1)}$.

Lemma 2. $\Omega_{\ell^*}(S_{SL}^*)\mathbf{w}^{*\beta} \geq \rho(S^*)$.

Proof. $\Omega_{\ell^*}(S_{SL}^*)\mathbf{w}^{*\beta} \geq \max_{\ell \in L} \Omega_{\ell}(S^*)\mathbf{w}^{*\beta} \geq \max_{\hat{L} \subseteq L} \min_{\ell \in \hat{L}} \Omega_{\ell}(S^*) (\sum_{\ell' \in \hat{L}} \mathbf{w}(\ell'))^{\beta}$.

Theorem 2 (Approximation Algorithm for $p \geq 1$).

$$\rho(C^*) \geq \frac{1}{(p+1)^{1/p}} \times \frac{\mathbf{w}_{\min} \times \max\{\lambda^{+1/p}, \lambda^{+\beta/p}\}}{\mathbf{w}_{\max} \mathbf{w}^{*\beta+1/p}} \times \rho(S^*), \quad (7)$$

Proof. The proof of this theorem is based on Lemmas 1 and 2, and the fact that $k^+ \geq \frac{\Omega_{\ell^*}(S_{SL}^*)^p}{\mathbf{w}(\ell^*)^p(p+1)}$.

Note that for the sake of simplicity, in the above theorem, we used Inequality 6. For a tighter bound, one can use Inequality 5 in Lemma 1. When $p = 1$ and $\mathbf{w}(\cdot) = 1$, the approximation guarantee matches the approximation guarantee by Hashemi et al. [22], which is the best existing guarantee for this special case. Note that, our work is the first algorithm for the generalized p -mean MDS case. **Case 2:** $p \in [-\infty, 1]$. In this part, we show that our approx solution to 1-mean MDS, can provide an approximation solution to p -mean MDS, when $p \in [-\infty, 1]$.

Theorem 3 (Approximation Algorithm for $-\infty \leq p \leq 1$).

$$\rho(C^*) \geq \frac{1}{(p+1)^{1/p}} \times \frac{\mathbf{w}_{\min} \times \max\{\lambda^{+1/p}, \lambda^{+\beta/p}\}}{2 \times \mathbf{w}_{\max} \mathbf{w}^{*\beta+1/p}} \times \rho(S^*), \quad (8)$$

Proof. Let $S_{SL}^{*(1)}$ be the optimal solution of $\Omega_{\ell^*}(S_{SL}^*)$ when $p = 1$. We know that $\min_{u \in S_{SL}^{*(1)}} \text{deg}_{\ell^*}(u) \geq \frac{1}{2} \Omega_{\ell^*}(S_{SL}^{*(1)}) = \frac{1}{2} \Omega_{\ell^*}^{(1)}(S_{SL}^{*(1)})$ for $p = 1$, since removing the node with the minimum degree cannot increase the density. On the other hand, as discussed by Chekuri and Torres [9], p -mean function over the degree of nodes in a graph is monotone. Therefore, we have:

$$\Omega_{\ell^*}(S_{SL}^{*(1)}) \geq \min_{u \in S_{SL}^{*(1)}} \text{deg}_{\ell^*}(u) \geq \frac{1}{2} \Omega_{\ell^*}^{(1)}(S_{SL}^{*(1)}) \geq \frac{1}{2} \Omega_{\ell^*}(S_{SL}^*) \quad (9)$$

The last inequality comes from the monotonicity of p -mean function over the degree of nodes in a graph. Using Lemma 2 and Theorem 2, we can simply show the above approximation guarantee.

Note that, while empirically the value of α can affect the performance, theoretically its value cannot affect the approx guarantee as we only need $\alpha = 1$.

4 Experiments

Setup. Designed algorithms and baselines are implemented in Python (compiled by Cython). All experiments are performed on a Linux machine with Intel Xeon 2.6 GHz CPU and 128 GB RAM.

Datasets. In our experiments, we use 10 real-world datasets [2, 6, 8, 10–12, 16, 21, 22, 26, 27] whose domains cover social, genetic, co-authorship, financial, and co-purchasing networks. The main characteristics are summarized in Table 1. We use an unsupervised learning method to learn the importance of each layer [3] and treat them as layer weights.

Results. Table 1 reports the average edge density and multiplex density for different values of p . Based on these results, our definition of density can find different and meaningful dense structures. Also, it is notable that the effect of p on the performance depends on the datasets, which again shows the importance of the flexibility that our formulation can provide. GFirmCore in all datasets finds a densest structure that is denser than the found solution by FirmCore, which shows the significance of considering weights for different layers.

Table 1. Comparison of the solutions found by GFirmCore and the state-of-the-art FirmCore [22]. The superior performance of GFirmCore with different p shows the importance of considering weights for different relation types.

	Metric	Dataset	Homo	Sacchcere	FAO	Brain	DBLP	Amazon	FTTtwitter	Friendfeed	StackO	Google+
		$ V $	18k	6.5k	214	190	513k	410k	155k	510k	2.6M	28.9M
		$ E $	153k	247k	319K	934K	1.0M	8.1M	13M	18M	47.9M	1.19B
		$ L $	7	7	364	520	10	4	2	3	24	4
GFirmCore	Edge Density $\frac{\sum_{r \in L} w_r E_r[S] }{w^* \times \binom{ S }{2}}$	$p = -\infty$	0.73	0.68	0.45	1.00	0.52	0.48	0.74	0.39	0.50	0.98
		$p = -1$	0.73	0.49	0.47	1.00	0.39	0.48	0.59	0.36	0.53	0.56
		$p = 0$	0.39	0.55	0.39	0.92	0.39	0.33	0.59	0.78	0.46	0.73
		$p = 1$	0.58	0.46	0.47	0.90	0.39	0.51	0.59	0.48	0.53	0.84
	Multiplex Density [16]	$p = -\infty$	28.36	20.79	1553.84	3941.55	77.46	41.89	111.42	163.58	96.20	153.99
		$p = -1$	30.17	19.53	1559.25	3941.55	81.17	42.01	98.50	165.72	97.18	172.87
		$p = 0$	28.49	31.26	1674.41	7180.09	82.46	40.51	98.73	183.76	99.03	148.16
		$p = 1$	31.14	28.59	1854.07	7935.29	82.91	61.38	99.26	216.74	118.33	173.81
	Runtime (s)	$p = -\infty$	38	96	7199	9207	930	992	894	4375	23698	71148
		$p = -1$	43	101	7418	9491	1061	1206	1089	4810	26056	74703
		$p = 0$	39	113	7407	9462	1128	1135	1103	4729	26114	74669
		$p = 1$	48	105	7369	9503	1076	1160	1057	4788	25671	74893
FirmCore	Edge Density $\frac{\sum_{r \in L} w_r E_r[S] }{w^* \times \binom{ S }{2}}$	$p = -\infty$	0.69	0.61	0.45	0.92	0.44	0.37	0.60	0.42	0.46	0.74
		$p = -1$	0.58	0.61	0.45	0.92	0.35	0.33	0.52	0.38	0.49	0.70
		$p = 0$	0.32	0.61	0.39	0.92	0.35	0.31	0.52	0.36	0.41	0.52
		$p = 1$	0.47	0.42	0.35	0.78	0.41	0.42	0.52	0.36	0.45	0.52
	Multiplex Density [16]	$p = -\infty$	27.85	22.91	1553.84	6997.12	75.19	39.28	98.46	167.19	98.51	162.43
		$p = -1$	28.14	23.69	1598.66	7034.50	75.83	39.15	98.03	167.56	100.03	163.88
		$p = 0$	28.53	25.82	1659.41	7180.09	76.11	39.64	99.12	168.44	100.98	162.07
		$p = 1$	29.74	25.87	1673.18	7163.89	78.91	43.52	100.24	170.87	107.09	164.81
	Runtime (s)	$p = -\infty$	19	36	2403	3169	322	348	297	799	6951	34814
		$p = -1$	21	37	2964	3613	438	489	386	841	8116	35726
		$p = 0$	20	46	2954	3486	447	467	394	835	8170	35482
		$p = 1$	20	41	2454	3273	362	394	359	891	8053	36027

Since there is no algorithm for exactly finding the multiplex densest subgraph, we generate two synthetic datasets, S1 and S2, both with $|V| = 100$, $|E| = 10000$, $|L| = 4$. We use the same approach as real-world datasets to obtain layer weights. We also inject the densest subgraph via clique density to S1 and average degree

density to S2. Figure 1 reports the ratio of the found solution and the optimal solution obtained by our algorithms ($p = 1, 2, 3$) and baselines FirmCore [22] and ML k -core [16]. Our algorithms outperform both baselines in both datasets and all values of p including $p = 1$, which they are designed for. This result shows the importance of handling different importance for different layers.

Figure 2 shows the running time of our algorithms and baselines. While our algorithms are much faster than ML k -core [16], FirmCore is more efficient than our algorithms. The main reason is that FirmCore does not consider different weights and as we discussed in Sect. 3, this relaxation can change the complexity of the decomposition (GFirmCore is NP-hard while FirmCore is polynomial). It is notable that our algorithms are scalable to graphs with billions of edges.

Case Study: Brain Networks. Detecting and monitoring functional systems in the human brain is a primary task in neuroscience. Brain Networks obtained from fMRI, are graph representations of the brain, where each node is a brain region and two nodes are connected if there is a high correlation between their functionality. However, the brain network generated from an individual can be noisy and incomplete. Using brain networks from many individuals can help to identify functional systems more accurately. A dense subgraph in a multiplex brain network, where each layer is the brain network of an individual, can be interpreted as a functional system in the brain. Figure 3 shows the densest subgraph including the occipital pole found by FirmCore and GFirmCore as well as the ground-truth functional system of the occipital pole (i.e., visual processing). The densest subgraph found by GFirmCore is more similar to ground truth than FirmCore. The main reason is that the brain network generated from an individual can be noisy/incomplete and FirmCore treats all layers the same.

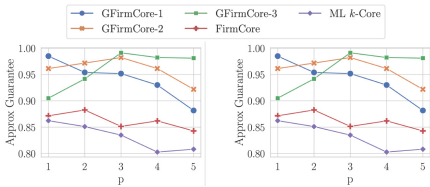


Fig. 1. The quality of found solution by GFirmCore and baselines. (Left) S1, (Right) S2 datasets.

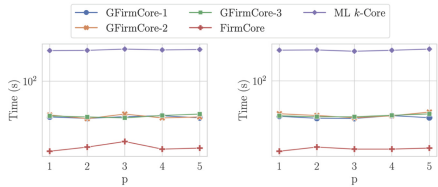


Fig. 2. The running time of GFirmCore and baselines. (Left) S1, (Right) S2 datasets.

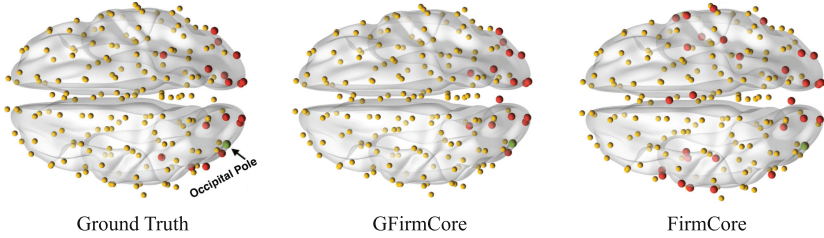


Fig. 3. The running time of GFirmCore and baselines. (Left) S1, (Right) S2 datasets.

5 Conclusion

In this paper, we propose and study a novel extended notion of core in layer-weighted multiplex networks, GFirmCore, where each layer has a weight that indicates the importance/significance of the layer. We show that theoretically this problem is more challenging than its layer-unweighted counterpart and is NP-hard. We further extend the notion of multiplex density to layer-weighted multiplex networks. For the sake of unifying existing density measures, we propose a new family of densest subgraph objectives, parameterized by a single parameter p that controls the importance of larger/smaller degrees in the subgraph. Using our GFirmCore, we propose the first polynomial approximation algorithm that provides approximation guarantee in the general case of p -mean densest subgraph problem. Our experimental results, show the efficiency and effectiveness of our algorithms and the significance of considering different weights for the layers in multiplex networks.

References

1. Azimi-Tafreshi, N., Gomez-Garde, J., Dorogovtsev, S.N.: k-corepercolation on multiplex networks. *Phys. Rev. E* **90**(3) (2014). ISSN 1550-2376
2. Behrouz, A., Hashemi, F.: CS-MLGCN: multiplex graph convolutional networks for community search in multiplex networks. In: Proceedings of the 31st ACM International Conference on Information and Knowledge Management, CIKM '22, pp. 3828–3832. New York, NY, USA (2022). Association for Computing Machinery. ISBN 9781450392365. <https://doi.org/10.1145/3511808.3557572>
3. Behrouz, A., Seltzer, M.: Anomaly detection in multiplex dynamic networks: from blockchain security to brain disease prediction. In: NeurIPS 2022 Temporal Graph Learning Workshop (2022). <https://openreview.net/forum?id=UDGZDfwmay>
4. Behrouz, A., Seltzer, M.: Anomaly detection in human brain via inductive learning on temporal multiplex networks. In: Machine Learning for Healthcare Conference, vol. 219. PMLR (2023)
5. Behrouz, A., Seltzer, M.: ADMIRE++: explainable anomaly detection in the human brain via inductive learning on temporal multiplex networks. In: ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH) (2023). <https://openreview.net/forum?id=t4H8acYudJ>

6. Behrouz, A., Hashemi, F., Lakshmanan, L.V.S.: Firmtruss community search in multilayer networks. *Proc. VLDB Endow.* **16**(3), 505–518 (2022). ISSN 2150-8097. <https://doi.org/10.14778/3570690.3570700>
7. Cardillo, A., et al.: Emergence of network features from multiplexity. *Sci. Rep.* **3**(1), 1–6 (2013)
8. Celli, F., Di Lascio, F.M.L., Magnani, M., Pacelli, B., Rossi, L.: Social network data and practices: the case of friendfeed. In: Chai, S.-K., Salerno, J.J., Mabry, P.L. (eds.) *SBP 2010. LNCS*, vol. 6007, pp. 346–353. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12079-4_43
9. Chekuri, C., Torres, M.R.: On the generalized mean densest subgraph problem: complexity and algorithms. arXiv preprint [arXiv:2306.02172](https://arxiv.org/abs/2306.02172) (2023)
10. De Domenico, M., Lima, A., Mougél, P., Musolesi, M.: The anatomy of a scientific rumor. *Sci. Rep.* **3**(1) (2013). ISSN 2045-2322
11. De Domenico, M., Porter, M.A., Arenas, A.: MuxViz: a tool for multilayer analysis and visualization of networks. *J. Complex Netw.* **3**(2), 159–176 (2014). ISSN 2051-1329. <https://doi.org/10.1093/comnet/cnu038>
12. De Domenico, M., Nicosia, V., Arenas, A., Latora, V.: Structural reducibility of multilayer networks. *Nat. Commun.* **6**, 6864 (2015)
13. Du, X., Jin, R., Ding, L., Lee, V.E., Thornton, J.H.: Migration motif: a spatial - temporal pattern mining approach for financial markets. In: *KDD*, pp. 1135–1144 (2009)
14. Faragó, A.: A general tractable density concept for graphs. *Math. Comput. Sci.* **1**(4), 689–699 (2008)
15. Fratkin, E., Naughton, B.T., Brutlag, D.L., Batzoglou, S.: MotifCut: regulatory motifs finding with maximum density subgraphs. *Bioinformatics* (Oxford, England), **22**(14) (2006)
16. Galimberti, E., Bonchi, F., Gullo, F.: Core decomposition and densest subgraph in multilayer networks. In: *Conference on Information and Knowledge Management (CIKM)* (2017)
17. Galimberti, E., Bonchi, F., Gullo, F., Lanciano, T.: Core decomposition in multilayer networks: theory, algorithms, and applications. *ACM Trans. Knowl. Discov. Data* **14**(1) (2020). ISSN 1556-4681. <https://doi.org/10.1145/3369872>
18. Gallo, G., Grigoriadis, M.D., Tarjan, R.E.: A fast parametric maximum flow algorithm and applications. *SIAM J. Comput.* **18**(1), 30-55 (1989). ISSN 0097-5397. <https://doi.org/10.1137/0218003>
19. Gibson, D., Kumar, R., Tomkins, A.: Discovering large dense subgraphs in massive graphs. In: *VLDB*, pp. 721–732 (2005)
20. Goldberg, A.: Finding a maximum density subgraph. Technical report (1984)
21. Zhenqiang Gong, N., et al.: Evolution of social-attribute networks: Measurements, modeling, and implications using google+. In: *Internet Measurement Conference*, pp. 131–144, NY, USA. ACM (2012)
22. Hashemi, F., Behrouz, A., Lakshmanan, L.V.S.: Firmcore decomposition of multilayer networks. In: *Proceedings of the ACM Web Conference 2022, WWW '22*, page 1589-1600, New York, NY, USA (2022). Association for Computing Machinery. ISBN 9781450390965. <https://doi.org/10.1145/3485447.3512205>
23. Jethava, V., Beerenwinkel, N.: Finding dense subgraphs in relational graphs. In: Appice, A., Rodrigues, P.P., Santos Costa, V., Gama, J., Jorge, A., Soares, C. (eds.) *Machine Learning and Knowledge Discovery in Databases. LNCS (LNAI)*, vol. 9285, pp. 641–654. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23525-7_39

24. Kivela, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. *J. Complex Netw.* **2**, 203–271 (2014). ISSN 2051-1310. <https://doi.org/10.1093/comnet/cnu016>
25. Lanciano, T., Miyauchi, A., Fazzino, A., Bonchi, F.: A survey on the densest subgraph problem and its variants (2023)
26. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. *ACM Trans. Web* **1**(1), 5-es (2007). ISSN 1559-1131
27. Omodei, E., De Domenico, M., Arenas, A.: Characterizing interactions in online social networks during exceptional events. *Front. Phys.* **3**, 59 (2015). ISSN 2296-424X. <https://doi.org/10.3389/fphy.2015.00059>
28. Veldt, N., Benson, A.R., Kleinberg, J.: The generalized mean densest subgraph problem. In: Proceedings of the 27th ACM SIGKDD, KDD '21, pp. 1604–1614, New York, NY, USA (2021). ACM. <https://doi.org/10.1145/3447548.3467398>



Influence Robustness of Nodes in Multiplex Networks Against Attacks

Boqian Ma[✉], Hao Ren[✉], and Jiaojiao Jiang^(✉)

School of Computer Science and Engineering, University of New South Wales,
Kensington, NSW 2052, Australia
{boqian.ma, hao.ren}@student.unsw.edu.au, jiaojiao.jiang@unsw.edu.au

Abstract. Recent advances have focused mainly on the resilience of the monoplex network in attacks targeting random nodes or links, as well as the robustness of the network against cascading attacks. However, very little research has been done to investigate the robustness of nodes in multiplex networks against targeted attacks. In this paper, we first propose a new measure, **MultiCoreRank**, to calculate the global influence of nodes in a multiplex network. The measure models the influence propagation on the core lattice of a multiplex network after the core decomposition. Then, to study how the structural features can affect the influence robustness of nodes, we compare the dynamics of node influence on three types of multiplex networks: assortative, neutral, and disassortative, where the assortativity is measured by the correlation coefficient of the degrees of nodes across different layers. We found that assortative networks have higher resilience against attack than neutral and disassortative networks. The structure of disassortative networks tends to break down quicker under attack.

Keywords: Multiplex network · Resilience · Complex network · Centrality

1 Introduction

Many studies have been conducted to analyse the resilience of different types of networks, such as monoplex, interconnected, or multiplex networks, against different types of attacks (such as random, targeted, or cascading attacks). In monoplex networks, Albert *et al.* [1] found that networks with a broad degree distribution (such as scale-free) exhibit a low degree of resilience if the attack happened on a large degree node, and a high degree of resilience otherwise. A similar phenomenon occurs when cascading attacks occur in monoplex networks [10, 27]. In interconnected networks, the malfunction of nodes within one network might trigger the collapse of reliant nodes in separate networks. Contrary to the behavior observed in single-layer networks, Buldyrev *et al.* [9] demonstrated that a more heterogeneous degree distribution amplifies the susceptibility of independent networks to stochastic failures. Within the context of multiplex networks, which are composed of multiple layers sharing a common set of nodes

[4], various studies have indicated that correlated interconnections can influence the structural resilience of these networks in a complex manner [7, 25].

The resilience of a network against attacks is often measured from the perspective of network functionality, such as the probability of the existence of giant connected components [9]. However, this could not reflect the robustness of the node influence, which is of great significance. For example, in a power grid network, if a high-degree node is removed, many nodes that are connected to it will also be removed. Such removal will result in changes in the influence of the neighbouring nodes and beyond. To maintain the communication efficiency of a network, we often need to retain the robustness of influential nodes. Little research has been done to study the influence robustness of nodes in multiplex networks against attacks. In monoplex networks, Jiang *et al.* [21] used the notion of coreness to measure the global influence of nodes. They found that nodes with high *coreness* in assortative networks tend to maintain their degree of coreness even after the influential nodes are removed. On the other hand, in disassortative networks, the node’s influence is distorted when influential nodes are removed.

In this paper, we extend the study of influence robustness of nodes against attacks from monoplex to multiplex networks. We first develop a new node centrality, **MultiCoreRank**, that measures the global influence of nodes in a multiplex network. Current centrality measures in multiplex networks are based on (1) projecting all layers into a monoplex network before applying the metrics on monoplex networks or (2) calculating the metrics in each individual layer separately, before aggregating to form a value for each node [17, 30]. However, these methods overlooked the multi-relation nature of multiplex networks, which could cause information loss in the process. To address this gap, we extend the idea of core decomposition in multiplex networks presented in [20] and calculate the global influence of nodes through propagation of node influence along the “core lattice”.

The main contributions of this paper include:

- We propose, **MultiCoreRank**, a new node centrality that measures the global influence of nodes in multiplex networks.
- We analyse the influence robustness of nodes across different types of multiplex network: assortative, neutral, and disassortative networks.
- The experimental results demonstrate that the assortative multiplex networks have greater robustness and are more resilient against targeted attack.

The rest of the paper is organised as follows. Section 2 introduces some related work. In Sect. 3, we introduce the proposed centrality measure. Section 4 outlines our experimental results, followed by the conclusion in Sect. 5. In addition, code is available at <https://github.com/Boqian-Ma/MultiCoreRank>.

2 Related Work

Let $G = (V, E, L)$ be a multiplex network, where V is a set of vertices, L is a set of layers and $E \subseteq V \times V \times L$ is a set of links. Each layer of G is a monoplex network $G^{[\alpha]}$, $\alpha \in L$. Each layer α is associated with an adjacency matrix $A^{[\alpha]} = \left(a_{ij}^{[\alpha]} \right)$,

where $a_{ij}^{[\alpha]} = 1$ if there is a link between i and j on layer α , and 0 otherwise. In the following, we first introduce some existing centrality measures, and then we discuss some related work on network resilience.

2.1 Node Centrality Measures

Various centrality measures have been developed to calculate the influence of nodes on monoplex and multiplex networks. **Degree Centrality** quantifies the number of edges attached to a specific node in a monoplex network. It was extended into **Overlapping Degree** in multiplex networks by summing the node's degree across various layers [3]. A node is considered influential if it is connected to a high number of edges. Bonacich *et al.* formulated **Eigenvector Centrality**, and proposed that the principal eigenvector of an adjacency matrix serves as an effective indicator of a node's centrality within the network [5]. Extending this to multiplex networks, Sola *et al.* [31] introduced multiple alternative metrics to evaluate the significance of nodes. **Betweenness centrality** measures the importance of a node by considering how often a node v lies in a shortest path between i and j [6]. Chakraborty *et al.* [11] extend betweenness centrality to multiplex networks and introduced cross-layer betweenness centrality. **Closeness Centrality** [29] quantifies the proximity of a given node to all other nodes within a network by calculating the average distance via the shortest pathways to all other nodes. A node gains significant importance if it is situated closer to every other node within the network. Mittal *et al.* [26] introduced cross-layer closeness centrality for multiplex networks. We note the above measures as classical centrality measures and their counterparts on multiplex networks.

More recently, other novel centrality methods have been proposed based on random walks [12, 18], gravity model [14], and *posteriori* measures [23].

Table 1 provides a list of classical centrality measures in monoplex and multiplex networks. Note that the counterpart of each centrality measure on a multiplex network is simply the sum of node centralities obtained on the different layers. For more complicated centrality measures, the readers can refer to [16].

Table 1. Node centralities in monoplex networks and multiplex networks. In the formula, $\lambda^{[\alpha]}$ represents the principal eigenvalue corresponding to the adjacency matrix $A^{[\alpha]}$. $\sigma_{pq}(i)$ signifies the aggregate number of shortest paths from node p to node q that traverse through node i , while σ_{pq} indicates the overall number of shortest paths between nodes p and q . $\text{dist}(i^{[\alpha]}, j^{[\alpha]})$ is used to describe the minimal path distance between nodes i and j within layer α .

Centrality	Monoplex	Multiplex
Degree	$d_i^{[\alpha]} = \sum_{j \in V} a_{ij}^{[\alpha]}$	$d_i = \sum_{\alpha \in L} d_i^{[\alpha]}$
Eigenvector	$e_i^{[\alpha]} = \frac{1}{\lambda^{[\alpha]}} \sum_{j \in V} a_{i,j}^{[\alpha]} e_j^{[\alpha]}$	$e_i = \sum_{\alpha \in L} e_i^{[\alpha]}$
Betweenness	$b_i^{[\alpha]} = \sum_{i \neq p \neq q \in V} \frac{\sigma_{p^{[\alpha]}q^{[\alpha]}}(i^{[\alpha]})}{\sigma_{p^{[\alpha]}q^{[\alpha]}}$	$b_i = \sum_{\alpha \in L} b_i^{[\alpha]}$
Closeness	$c_i^{[\alpha]} = \frac{n-1}{\sum_{j \in V} \text{dist}(i^{[\alpha]}, j^{[\alpha]})}$	$c_i = \sum_{\alpha \in L} c_i^{[\alpha]}$

2.2 Network Resilience

Network resilience is measured by the ability of a network to retain its structure when some nodes in the network are removed [13]. It can be measured by network assortativity, which describes the tendency of nodes in a network to connect to other nodes that are similar (or different) in some way. In recent decades, extensive contributions have been made to network resilience analysis [1, 10, 21, 27]. Understanding network resilience is of high research interest because it will allow us to design fail-safe networks such as transportation or energy networks.

In terms of the robustness of multilayer networks, Buldyrev *et al.* [9] found that an interconnected network is vulnerable to random failures if it presents a broader degree distribution, which is the opposite of the phenomenon in monoplex networks. De *et al.* [15] employed random walks to establish an analytical model for examining the time required for random walks to cover interconnected networks. Their findings indicate that such interconnected structures exhibit greater resilience to stochastic failures compared to their standalone layers. Min *et al.* [25] studied the resilience of multiplex networks and found that correlated coupling can affect the structural robustness of multiplex networks in diversified fashion. Brummitt *et al.* [8] generalised the threshold cascade model [32] to study the impact of multiplex networks on cascade dynamics. They found that multiplex networks are more vulnerable to global cascades than monoplex networks.

More recently, Fan *et al.* [19] proposed a multiplex network resilience metric and studied link addition strategies to improve resilience against targeted attacks.

Kazawa *et al.* [22] proposed effective link-addition strategies for improving the robustness of multiplex networks against degree-based attacks.

Recent studies mainly analyse resilience from a network functionality perspective, such as the probability of the existence of giant connected components [9]. This work extends from Jiang *et al.*'s previous work on the influence robustness of nodes on monoplex networks. In this paper, we study the resilience of nodes in **multiplex networks** under **targeted** (i.e. attacking nodes based on their influence) and **uniformly random** attacks. Before that, we first develop a method to measure node influence based on core decomposition in multiplex networks (see Sect. 3).

3 Proposed Node Centrality

3.1 Preliminaries

Given a multiplex network $G = (V, E, L)$ and a subset $S \subseteq V$, we use $G[S] = (S, E[S], L)$ to denote the subgraph of G , where $E[S]$ is the set of all the links in E connecting the nodes in S and L is the set of layers. We use $\tau^{[\alpha]}[S]$ to denote the minimum degree of nodes on layer α in the sub-graph. The core decomposition in multiplex networks is defined as follows.

Definition 1 (k-core percolation** [2]).** Given a multiplex network $G = (V, E, L)$ and an $|L|$ -dimensional integer vector $\mathbf{k} = [k^{[\alpha]}]_{\alpha \in L}$, the **k-core** of G is defined as the maximum subgraph $G[C] = (C, E[C], L)$ such that $\forall \alpha \in L : \tau^{[\alpha]}[C] \geq k^{[\alpha]}$. \mathbf{k} is termed as a core vector.

Hence, \mathbf{k} -core is the maximal subgraph that each node has at least $k^{[\alpha]}$ edges of each layer, $\alpha \in L$. The \mathbf{k} -core of a multiplex network could be calculated by removing nodes iteratively until $k^{[\alpha]} \in \mathbf{k}$, $\alpha \in L$ no longer satisfied. Taking the two-layer graph in Fig. 1(a) as an example, the (1, 2)-core is $\{A, B, D, E\}$ and the (2, 2)-core is $\{B, D, E\}$.

Theorem 1 (Core containment [20]). *Given a multiplex network $G = (V, E, L)$, let $C_{\mathbf{k}}$ and $C_{\mathbf{k}'}$ be the cores given by $\mathbf{k} = [k^{[\alpha]}]_{\alpha \in L}$ and $\mathbf{k}' = [k'^{[\alpha]}]_{\alpha \in L}$, respectively. It follows that if $\forall \alpha \in L : k'^{[\alpha]} \leq k^{[\alpha]}$, then $C_{\mathbf{k}} \subseteq C_{\mathbf{k}'}$.*

The partial containment of all cores can be represented by a lattice structure known as the *core lattice*. The core lattice of the example network in Fig. 1(a) is shown in Fig. 3. The nodes in the lattice represent cores and edges represent the containment relationship between cores where the “father” core contains all of its “child” cores (i.e. all cores from a core to the root). Using the core lattice structure, Galimberti *et al.* [20] developed three algorithms for efficiently computing cores in multiplex networks: DFS-based, BFS-based, and hybrid approaches. In the centrality method we are proposing in the next subsection, we use the BFS-based approach because, in order to update a node’s influence given a core, all father cores must be calculated first. The approach based on Breadth-First Search (BFS) leverages two key observations: (1) a non-empty \mathbf{k} -core is a subset of the intersection of all its preceding cores’ fathers”, and (2) the quantity of such preceding cores for any non-empty \mathbf{k} -core is commensurate with the number of non-zero components in its associated core vector \mathbf{k} .

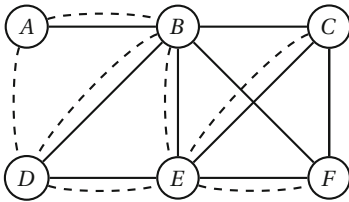


Fig. 1. An example two-layer network, where solid lines signify edges belonging to the first layer, while dashed lines indicate edges associated with the second layer.

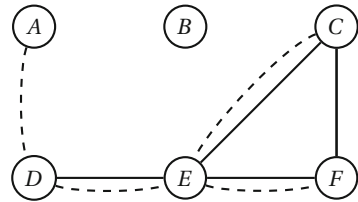


Fig. 2. The network after removing node B and its edges in Fig. 1.

3.2 MultiCoreRank Centrality

On a core lattice, we can observe that (1) nodes that appear in deeper level cores are more connected, which makes them more influential than those that only appear in shallower levels, and (2) for nodes on the same lattice level, those that appear in fewer cores are less influential than those that appear in more cores, as they have higher chances to have child cores, according to the core

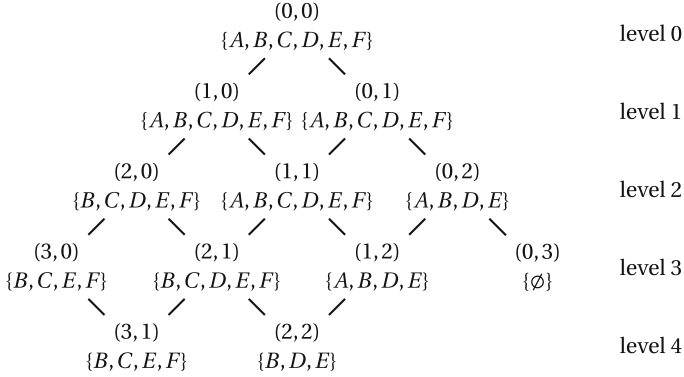


Fig. 3. The core lattice of the network in Fig. 1. The numbers in the core vectors are the minimum degrees of each layer in a core. (i.e. (2,1) consists of nodes with at least degree 2 on layer 1 and at least degree 1 on layer 2.) The level in which a core is in the lattice is given by the $L-1$ norm of its core vector. The core vector (0,3) is shown as an example of empty core while other empty cores are omitted.

containment theory in Theorem 1. We argue that an *ideal* centrality measure based on the core lattice should at least consider these two points.

Using the two observations above, we consider the calculation of the overall influence of a node $v \in V$ as a message passing process, which iteratively calculates the influence of node v on layer $l+1$ based on its influence on level l on the lattice.

Before introducing `MultiCoreRank`, we first define a father-child relationship, $\mathbf{k}_{l+1} \xrightarrow{\text{father}} \mathbf{k}_l$, of the two corresponding core vectors on levels $l+1$ and l , respectively, if there exists an edge between \mathbf{k}_{l+1} and \mathbf{k}_l on the core lattice. For an arbitrary node $v \in V$, we use $\text{inf}_l(v)$ to denote the influence of node v on level l of the core lattice, if there exists a \mathbf{k}_l -core such that $v \in C_{\mathbf{k}_l}$. Also, let $|\text{father}_{\mathbf{k}_l}|$ be the number of fathers of the core given by \mathbf{k}_l .

Now, considering observation (1), for an arbitrary level l , we assign it a weight l , allowing nodes at deeper lattice levels to have a larger weight. Next, considering observation (2), for an arbitrary node v , if v appears at both level l and level $l+1$ on the lattice, we aggregate v 's influences from all the cores that contain node v on level l as its influence on level $l+1$. In this way, nodes with more appearances will be assigned a higher influence than those with fewer appearances. The following formula gives the detailed calculation of the `MultiCoreRank` influence of a node v on level $l+1$ of a core lattice:

$$\text{inf}_{l+1}(v) = \sum_{\{\mathbf{k}_{l+1} \mid v \in C_{\mathbf{k}_{l+1}}, v \in C_{\mathbf{k}_l}, \mathbf{k}_{l+1} \xrightarrow{\text{father}} \mathbf{k}_l\}} (l+1) \cdot \text{inf}_l(v) \cdot \text{inf}(C_{\mathbf{k}_l}) \cdot |\text{father}_{\mathbf{k}_{l+1}}|, \quad (1)$$

where $\text{inf}(C_{k_l})$ is the influence of the k_l -core, calculated by

$$\text{inf}(C_{k_l}) = \frac{\sum_{v \in C_{k_l}} \text{inf}_l(v)}{|C_{k_l}|}. \quad (2)$$

Note that our proposed influence measure can be calculated using the BFS-based approach mentioned in Sect. 3.1 because of the layer-by-layer and iterative nature of this method. The influence of a node v on layer $l + 1$ is calculated on the basis of its influence on level l , hence all cores on layer l must be calculated before moving onto layer $l + 1$. Referring to Fig. 3, the order in which the cores are calculated from layer 0 to 2 is $(0, 0)$, $(1, 0)$, $(0, 1)$, $(2, 0)$, $(1, 1)$, $(0, 2)$.

To illustrate our method, consider node A in Fig. 1 and the lattice in Fig. 3. At level 0 of the lattice, since the $(0, 0)$ -core contains the entire network, we initialise the influence of all nodes to 1. On level 1, after the $(1, 0)$ -core and the $(0, 1)$ -core are found using BFS, we apply Eq. (1) to node A to get

$$\text{inf}_{l=1}(A) = \sum_{k_l \in \{(0,1), (1,0)\}} 1 \cdot \text{inf}_{l=0}(A) \cdot \text{inf}(C_{k_{l-1}}) \cdot |\text{fathers}_{k_l}| = 1 + 1 = 2.$$

On level 2, we have the following equation:

$$\begin{aligned} \text{inf}_{l=2}(A) &= \sum_{k_l \in \{(1,1), (0,2)\}} 2 \cdot \text{inf}_{l=1}(A) \cdot \text{inf}(C_{(k_{l-1})}) \cdot |\text{fathers}_{k_l}| \\ &= (2 \cdot 2 \cdot 2 \cdot 2) + (2 \cdot 2 \cdot 2 \cdot 1) = 24. \end{aligned}$$

The rest can be deduced accordingly¹.

4 Empirical Analysis of Influence Robustness of Nodes in Multiplex Networks

In this section, we commence by delineating the assortativity metric employed for gauging the structural characteristics of multiplex networks. Subsequently, we provide an overview of the data sets utilized for experimental validation. Following that, we assess the performance efficacy of the proposed centrality metric for nodes. Lastly, we undertake an analysis of the robustness of multiplex networks under varying levels of assortativity.

4.1 Multiplex Network Assortativity

We study the resilience of multiplex networks by analysing the dynamics of node influence when the most influential nodes are removed. We particularly consider the structural feature of *assortativity* of multiplex networks.

¹ When implementing this method on a large-scale network, appropriate normalisation techniques are required when the network is large to prevent numeric overflow.

The assortativity of a multiplex network is measured by the average layer-layer degree correlation [28]. If we denote $\mathbf{d}^{[\alpha]} = (d_1^{[\alpha]}, \dots, d_{|V|}^{[\alpha]})$ and $\mathbf{d}^{[\beta]} = (d_1^{[\beta]}, \dots, d_{|V|}^{[\beta]})$ as the degree vectors of layer α and β respectively, the layer-layer degree correlation between these two layers is given by

$$r_{\alpha,\beta} = \frac{\langle \mathbf{d}^{[\alpha]} \mathbf{d}^{[\beta]} \rangle - \langle \mathbf{d}^{[\alpha]} \rangle \langle \mathbf{d}^{[\beta]} \rangle}{\sigma_{\mathbf{d}^{[\alpha]}} \sigma_{\mathbf{d}^{[\beta]}}}, \quad (3)$$

where $\sigma_{\mathbf{d}^{[\alpha]}} = \sqrt{\langle \mathbf{d}^{[\alpha]} \mathbf{d}^{[\alpha]} \rangle - \langle \mathbf{d}^{[\alpha]} \rangle^2}$. $r_{\alpha,\beta}$ is the Spearman coefficient of $\mathbf{d}^{[\alpha]}$ and $\mathbf{d}^{[\beta]}$. $r_{\alpha,\beta}$ being close to 1 (assortative) means that the nodes in layers α and β are likely to have a similar tendency when connecting with their neighbours (i.e. a node has relative high/low degrees in both layers), whereas $r_{\alpha,\beta}$ being close to -1 , means that the nodes in α and β are less likely to have a similar tendency when connecting with their neighbours (i.e. a node has relative high/low degrees in one layer by the opposite in the other layer).

In this paper, we compute the assortativity of a multiplex network as the average of all layer-layer degree correlation given by

$$r_G = \frac{\sum_{\alpha < \beta \in L} r_{\alpha,\beta}}{|L|^2 - L} \quad (4)$$

Without losing generality, we disregards all correlations of a layer to itself.

4.2 Datasets

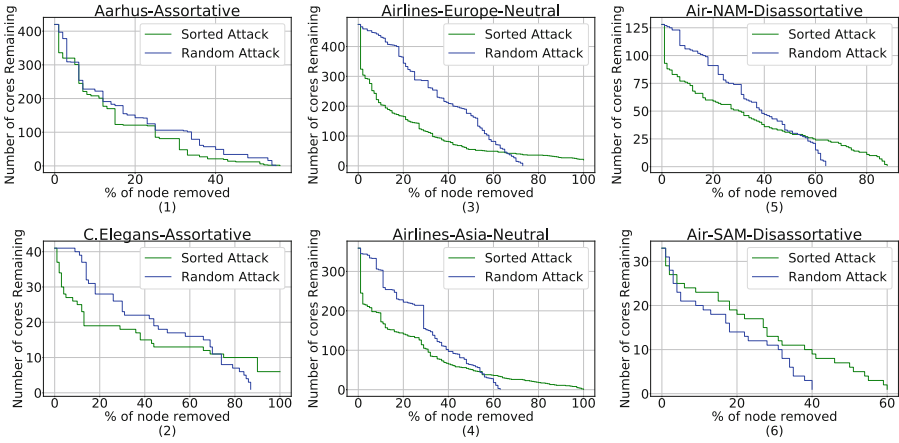
We selected two datasets for each of the assortative, neutral, and disassortative networks for our experiments. The details of the datasets are as follows, and Table 2 gives the basic statistics of the datasets. **C.elegans**² [28] is the neural network of the C.elegans nematode worm that consists of two layers representing synapses and gap junctions. **Aarhus** [24] is a five-layer network that encapsulates five different types of interactions (Facebook, Leisure, Work, Co-authorship, and Lunch) among employees within the Computer Science department at Aarhus University. **OpenFlight continental airport networks**³ [28] consists of international flight routes, where layers represent an airline company, node represent airports and edges represent routes provided by the airlines. We selected layers of **South America** and **North America** such that the network is disassortative and layers of **Asia** and **Europe** such that the network is neutral.

² <https://manliodedomenico.com/data.php>.

³ <https://openflights.org/>.

Table 2. Statistics of the datasets used in this paper, where r_G is the assortativity calculated from Eq. (4).

Dataset	Network	#nodes	#links	# Selected layers	r_G
C.elegans	Assortative	281	2476	2	0.6414
Aarhus	Assortative	61	620	5	0.2160
Airlines-Europe (Air-EU)	Neutral	476	3068	75	0.0139
Airlines-Asia (Air-Asia)	Neutral	348	1281	63	0.0125
Airlines-SouthAmerica (Air-SAM)	Disassortative	129	272	13	-0.0141
Airlines-NorthAmerica (Air-NAM)	Disassortative	528	1699	33	-0.0052

**Fig. 4.** The number of cores remaining in the network after a percentage of nodes are removed. Two types of removal are performed 1) sorted attack based on **MultiCoreRank** and 2) uniformly random attack. (1) and (2) correspond to the two assortative networks, (3) and (4) correspond to the two neutral networks, and (5) and (6) correspond to the two disassortative networks.

4.3 Effectiveness of the Proposed Centrality

To evaluate the effectiveness of our method, we calculated the Spearman’s Coefficient between our measurement and other node centralities. The results are shown in Table 3. We found that our method correlates the most with overlapping degree (d_i). This is justifiable since k -core is obtained by removing nodes that no longer satisfy the degrees given by a coreness vector, leaving only the nodes with higher degrees. Hence, our method is also based on the degree of a node. For the assortative and neutral datasets, our method has shown strong correlations with the eigenvector, betweenness, and closeness centralities. However, with the disassortative datasets, SouthAmerica and NorthAmerica, eigenvector and betweenness centrality show relatively weak correlation.

In addition, we compare the effectiveness of our method when random and influential nodes are removed from the network. Figure 4 simulates the changes in the network structure when proportions of nodes is removed, which simulate attacks. In general, the quality of our centrality measure is demonstrated through the sharp decrease in the number of cores at the beginning of the attacks (i.e. Figure 4 (3, 4, 5, 6)). This corresponds to the effectiveness of our method in identifying highly influential nodes because removing them caused significant structural changes.

4.4 Influence Robustness of Nodes

Continuing on the node removal experiment from the previous section, we also analyse the impact on the overall network assortativity when nodes are removed. Figure 5 (1,3,5) shows the change in the percentage of nodes when the influential nodes are removed in order. Comparing Fig. 5 (1) with (3,5), we see that the change in the number of remaining cores is less drastic in the assortative networks when influential nodes are removed. That is to say, 1) assortative networks are more robust under targeted attacks, 2) the removal of high-influence nodes in a neutral and disassortative network has a high impact on the network robustness. On the other hand, when nodes are removed randomly as shown in Fig. 5 (2,4,6), the network structure does not show visible trends in terms of changes.

Table 4 presents the change in assortativity when a percentage of high-influence nodes are removed. The assortative networks (C.Elegans, Aarhus) remained relatively assortative after node removal. The disassortative networks remained disassortative. The neutral networks remained neutral. However, there

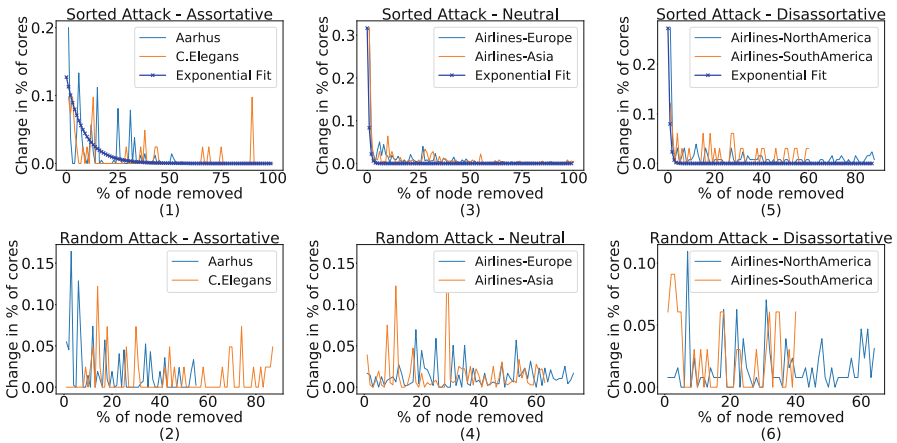


Fig. 5. The change in percentage of cores as a percentage of nodes are removed in different types of networks. The top row shows the sorted attack results. The bottom row shows the random attack results. The exponential fit on the top row is an exponential function, $y = ae^{-x^b}$, fitted on the average y-value given by each dataset at each x-value.

Table 3. Spearman’s rank correlation coefficients between node influence rankings from our method and other centrality measures in a complete network.

Dataset	d_i	λ_i	b_i	c_i
C.Elegans	0.85	0.65	0.65	0.78
Aarhus	0.82	0.76	0.49	0.81
Air-EU	0.51	0.45	0.38	0.55
Air-Asia	0.76	0.56	0.50	0.72
Air-SAM	0.93	0.29	0.62	0.86
Air-NAM	0.83	0.37	0.49	0.60

Table 4. Calculated coefficients of degree correlation between network layers following the targeted removal of the top 10%, 20%, and 30% of nodes.

Dataset	0%	10%	20%	30%
C.Elegans	0.6414	0.4174	0.4182	0.4315
Aarhus	0.2163	0.2723	0.3364	0.4110
Air-EU	0.0139	0.0061	0.0081	0.0042
Air-Asia	0.0125	0.0065	0.0072	0.0057
Air-SAM	-0.0141	-0.0050	0.0005	-0.0273
Air-NAM	-0.0052	-0.0042	-0.0057	-0.0062

is a trend in decreasing in assortativity in all types of network. This suggests that the initial network assortativity is a good indicator of the robustness of a given network.

Overall, from the above experiments, we can see that, similar to monoplex networks, assortative networks have shown higher robustness against attack than neutral and disassortative networks. The change in the overall k -core structure of the networks is smaller for the assortative networks.

5 Conclusion

In summary we developed a new node centrality measure, **MultiCoreRank** node centrality, based on core decomposition in multiplex networks. This measure takes into account the multi-relation nature of such networks and has shown consistency with existing methods through empirical comparisons. We then analysed the influence robustness of nodes across different types of multiplex networks: assortative, neutral and disassortative networks. We found that, in assortative networks, the k -core structure remains more consistent when nodes of high influence are removed. However, in neutral and disassortative networks, the number of k -cores tends to quickly decrease when they are under attack. In future work, we aim to study defence mechanisms to increase the robustness of multiplex networks and extend our method to multi-layer networks.

References

1. Albert, R., Jeong, H., Barabási, A.L.: Error and attack tolerance of complex networks. *Nature* **406**(6794), 378–382 (2000)
2. Azimi-Tafreshi, N., Gómez-Gardenes, J., Dorogovtsev, S.: k -core percolation on multiplex networks. *Phys. Rev. E* **90**(3), 032, 816 (2014)
3. Battiston, F., Nicosia, V., Latora, V.: Structural measures for multiplex networks. *Phys. Rev. E* **89**(3), 032, 804 (2014)

4. Bianconi, G.: *Multilayer Networks: Structure and Function*. Oxford University Press, Oxford (2018)
5. Bonacich, P.: Factoring and weighing approaches to clique identification. *J. Math. Sociol.* **92**, 1170–1182 (1971)
6. Brandes, U.: A faster algorithm for betweenness centrality. *J. Math. Sociol.* **25**(2), 163–177 (2001)
7. Brummitt, C.D., Kobayashi, T.: Cascades in multiplex financial networks with debts of different seniority. *Phys. Rev. E* **91**(6), 062, 813 (2015)
8. Brummitt, C.D., Lee, K.M., Goh, K.I.: Multiplexity-facilitated cascades in networks. *Phys. Rev. E* **85**(4), 045, 102 (2012)
9. Buldyrev, S.V., Parshani, R., Paul, G., Stanley, H.E., Havlin, S.: Catastrophic cascade of failures in interdependent networks. *Nature* **464**(7291), 1025–1028 (2010)
10. 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)
11. Chakraborty, T., Narayanam, R.: Cross-layer betweenness centrality in multiplex networks with applications. In: *ICDE*, pp. 397–408. IEEE (2016)
12. Chang, Y.C., Lai, K.T., Chou, S.C.T., Chiang, W.C., Lin, Y.C.: Who is the boss? Identifying key roles in telecom fraud network via centrality-guided deep random walk. *Data Technol. Appl.* **55**(1), 1–18 (2021)
13. Cohen, R., Havlin, S.: *Complex Networks: Structure Robustness and Function*. Cambridge University Press, Cambridge (2010)
14. Curado, M., Tortosa, L., Vicent, J.F.: A novel measure to identify influential nodes: return random walk gravity centrality. *Inf. Sci.* **628**, 177–195 (2023)
15. De Domenico, M., Solé-Ribalta, A., Gómez, S., Arenas, A.: Navigability of interconnected networks under random failures. *PNAS* **111**(23), 8351–8356 (2014)
16. De Domenico, M., Solé-Ribalta, A., Omodei, E., Gómez, S., Arenas, A.: Centrality in interconnected multilayer networks. arXiv preprint [arXiv:1311.2906](https://arxiv.org/abs/1311.2906) (2013)
17. De Domenico, M., Solé-Ribalta, A., Omodei, E., Gómez, S., Arenas, A.: Ranking in interconnected multilayer networks reveals versatile nodes. *Nat. Commun.* **6**(1), 1–6 (2015)
18. De Meo, P., Levene, M., Messina, F., Provetti, A.: A general centrality framework-based on node navigability. *IEEE Trans. Knowl. Data Eng.* **32**(11), 2088–2100 (2019)
19. Fan, D., et al.: A modified connectivity link addition strategy to improve the resilience of multiplex networks against attacks. *Reliab. Eng. Syst. Safety* **221**, 108, 294 (2022)
20. Galimberti, E., Bonchi, F., Gullo, F., Lanciano, T.: Core decomposition in multi-layer networks: theory, algorithms, and applications. *ACM Trans. Knowl. Discov. Data (TKDD)* **14**(1), 1–40 (2020)
21. Jiang, J., Wen, S., Yu, S., Zhou, W., Qian, Y.: Analysis of the spreading influence variations for online social users under attacks. In: *GLOBECOM*, pp. 1–6 (2016). <https://doi.org/10.1109/GLOCOM.2016.7841605>
22. Kazawa, Y., Tsugawa, S.: Effectiveness of link-addition strategies for improving the robustness of both multiplex and interdependent networks. *Physica A: Stat. Mech. its Appl.* **545**, 123, 586 (2020)
23. Lou, Y., Wang, L., Chen, G.: Structural robustness of complex networks: a survey of a posteriori measures [feature]. *IEEE Circuits Syst. Mag.* **23**(1), 12–35 (2023)
24. Magnani, M., Micenkova, B., Rossi, L.: Combinatorial analysis of multiple networks. arXiv preprint [arXiv:1303.4986](https://arxiv.org/abs/1303.4986) (2013)
25. Min, B., Do Yi, S., Lee, K.M., Goh, K.I.: Network robustness of multiplex networks with interlayer degree correlations. *Phys. Rev. E* **89**(4), 042, 811 (2014)

26. Mittal, R., Bhatia, M.P.S.: Cross-layer closeness centrality in multiplex social networks. In: ICCCNT, pp. 1–5. IEEE (2018)
27. Motter, A.E., Lai, Y.C.: Cascade-based attacks on complex networks. *Phys. Rev. E* **66**(6), 065, 102 (2002)
28. Nicosia, V., Latora, V.: Measuring and modeling correlations in multiplex networks. *Phys. Rev. E* **92**(3), 032, 805 (2015)
29. Nieminen, J.: On the centrality in a graph. *Scand. J. Psychol.* **15**(1), 332–336 (1974)
30. Salehi, M., Sharma, R., Marzolla, M., Magnani, M., Siyari, P., Montesi, D.: Spreading processes in multilayer networks. *IEEE Trans. Netw. Sci. Eng.* **2**(2), 65–83 (2015)
31. Solá, L., Romance, M., Criado, R., Flores, J., García del Amo, A., Boccaletti, S.: Eigenvector centrality of nodes in multiplex networks. *Chaos: Interdisc. J. Nonlinear Sci.* **23**(3), 033, 131 (2013)
32. Watts, D.J.: A simple model of global cascades on random networks. *PNAS* **99**(9), 5766–5771 (2002)



Efficient Complex Network Representation Using Prime Numbers

Konstantinos Bougiatiotis^{1,2(✉)} and Georgios Paliouras²

¹ Department of Informatics and Telecommunications,
National and Kapodistrian University, Athens, Greece
kbogas@di.uoa.gr

² Institute of Informatics and Telecommunications,
National Center Scientific Research Demokritos, Athens, Greece
{bogas.ko,paliourg}@iit.demokritos.gr

Abstract. In this work, we propose a novel representation of complex networks, which is compact and enables very efficient network analysis. Multi-relational networks capture complex data relationships and have a wide range of applications. As they get to be used with ever larger quantities of data, it is crucial to find efficient ways to represent and analyse them. This paper introduces the concept of Prime Adjacency Matrices (PAMs), which utilize prime numbers, to represent the relations of the network. Due to the Fundamental Theorem of Arithmetic, this allows for a lossless, compact representation of a complete multi-relational graph, using a single adjacency matrix. Moreover, this representation enables the fast computation of multi-hop adjacency matrices, which can be useful for a variety of downstream tasks. We illustrate the benefits of using the proposed approach through various network analysis tasks.

Keywords: complex networks modeling · graph classification · relation prediction · efficient graph analytics

1 Introduction

In recent years, research on complex networks has matured, and they have been the focus of study in multiple domains, such as biological, social, financial, and others [1]. This is because they allow us to model arbitrarily complex relationships between the data, thus making them very useful in real-world scenarios where complex structures arise. The observation that entities (i.e. nodes) in a complex network may be connected through multiple types of links has resulted in the study of *multi-relational* networks and their variants such as multi-layer, multi-dimensional or multi-plex networks. In this work, we will use the term complex graph/network as an umbrella term to express all kinds of data collections, that can be represented through triples of the form (h, r, t) , where h and t correspond to head and tail entities and r is the relation connecting them.

The goal when analyzing such graphs is to generate insights by aggregating the information expressed through each relation. There are many approaches to

analyzing networks for different downstream tasks, such as generating embeddings for the nodes and the relations in the graph [20], symbolic methodologies [7] and more recently graph neural networks [24]. However, many of these approaches make use only of the direct relations between entities, without being able to capture relations that are expressed through multiple hops in the graph [14]. Moreover, in many domains [5, 10] the paths connecting entities are useful for identifying the true nature of their relationship, the role of each entity, and finally help with the task at hand. Due to these requirements, there is a need for a framework that will facilitate easy and fast calculations of representations that capture the rich multi-hop information of the network.

To this end, we propose the *Prime Adjacency Matrix* (PAM) representation for complex networks. This representation compacts, in a lossless manner, all one-hop relations of the original network in a single adjacency matrix. By mapping each relation type to a distinct prime, we can construct the PAM in a manner that allows us to express all the information of the original graph without loss. Then, having at our disposal one adjacency matrix for the whole graph, we can easily calculate its powers and generate multi-hop adjacency matrices for the graph. This process is very fast and can scale easily to large, complex networks that cover many real-world applications.

These higher-order PAMs contain multi-hop information that is easily accessible; simply by looking up the values of the matrices. We motivate multiple scenarios where this representation can be used to generate structurally rich representations for graphs, nodes, pairs of nodes, etc. In this first exposure to the new representation, we design simple processes and present experimental results on tasks such as graph classification and relation prediction.

The main contributions of this work are the following:

- We introduce a new paradigm for representing complex networks in a single adjacency matrix using primes. To the best of our knowledge, this is the first work to model the full multi-relational graph in a single adjacency matrix in a lossless fashion.
- We use this compact representation for the fast calculation of multi-hop adjacency matrices for the complex graph, emphasizing its value for network analysis.
- We showcase the usefulness of the framework by conducting experiments on relation prediction and graph classification, where we greatly improve runtime using simple models while performing on par with commonly used models.

The rest of the paper is structured as follows: Sect. 2 introduces the PAM framework in detail. Then we present its application on different downstream tasks in Sect. 3. Finally, in Sect. 4, we summarize the main aspects of the novel method and propose future work.¹

¹ The code and related scripts can be found in <https://github.com/SubmissionUser/CN-PAM>.

2 Methodology

In this section, we introduce the proposed framework and highlight its main features.

2.1 Definition

Let us start with an unweighted, directed, complex graph G , with N nodes and R unique relation types. We can represent all possible edges between the different nodes with an adjacency tensor A of shape $N \times N \times R$:

$$A[i, j, r] = \begin{cases} 1, & \text{if } r \text{ connects nodes } i, j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

We now associate each unique relation type $r \in R$ with a distinct prime number p_r , through a mapping function φ , such that: $\forall r \in R : \varphi(r) = p_r$, where p_r is prime and $p_i = p_j \iff i = j$. This mapping function is a design choice and simply allocates distinct prime numbers to each $r \in R$. In its simplest form, we would randomly order the relations and allocate the first prime to the first relation, the second prime to the second one, and so forth.

With this mapping in place, we can construct the *Prime Adjacency Matrix* (PAM) P of shape $N \times N$ in the following form:

$$P[i, j] = \begin{cases} \prod_{r:A[i,j,r]=1} p_r, & \text{if } \exists r : A[i, j, r] = 1 \\ 0, & \text{if } \forall r : A[i, j, r] = 0 \end{cases} \quad (2)$$

As we can see in Eq. (2), each non-zero element $P[i, j]$ is the product of the primes p_r for all relations r that connect node i to j . Due to the Fundamental Theorem of Arithmetic (FTA), we can decompose each product to the original primes that constitute it (i.e. the distinct relations that connect the two nodes), thus preserving the full structure of G in P without any loss.

We will also define here P_+ , a variant of the above matrix, which aggregates the relations between two cells through their sum instead of their product, as shown in Eq. (3):

$$P_+[i, j] = \begin{cases} \sum_{r:A[i,j,r]=1} p_r, & \text{if } \exists r : A[i, j, r] = 1 \\ 0, & \text{if } \forall r : A[i, j, r] = 0 \end{cases} \quad (3)$$

As a note here, if each pair of nodes i, j exhibits at most one relation between them (i.e. only one edge connects them directly), we can see that $P = P_+$, from Eq. (2) and Eq. (3).

2.2 A Simple Example

Let us consider a simple toy graph, as the one shown in Fig. 1, where we have 5 nodes and 3 types of relation mapped to 3 (green), 5 (blue) and 7 (magenta), accordingly. The resulting PAM would be (with node A corresponding to index

$$0, \text{ node B to index 1, and so forth): } P = \begin{pmatrix} 0 & 3 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 7 & 0 & 0 & 0 \\ 3 & 7 & 3 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 \end{pmatrix}.$$

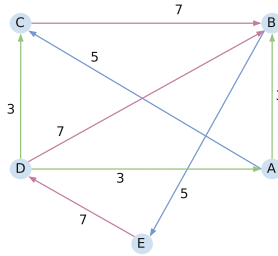


Fig. 1. A toy complex graph with 5 nodes and 3 types of relation.

Hence, for the edge $A \xrightarrow{3} B$ we have $P[0, 1] = 3$, for $A \xrightarrow{5} C$ we have $P[0, 2] = 5$ and so on, expressing all the edges in the graph. Even in this toy graph, the compact PAM representation facilitates interesting observations. For example, we can see all the incoming/outgoing edges and their types by simply looking at the corresponding columns/rows of P . So, looking at $P[0, :]$ and $P[:, 0]$ we see that node A has two outgoing edges (i.e. non-zero elements) of types 3 and 5, and one incoming edge of type 3. Another graph property that can be easily inferred is the frequency of different relations. If we simply count the occurrences of the non-zero elements of P , we get the distribution of edges per relation type, which is $\{3 : 3, 5 : 2, 7 : 3\}$.

2.3 Moving to Multi-hop Relationships

Having a single adjacency matrix for the whole G allows us to utilize tools from classical network analysis. Most importantly, we can easily obtain the powers of the adjacency matrix. In a single-relational and unweighted network, the element (i, j) of the power k of an adjacency matrix, contains the number of paths of length k from node i to node j . Generalizing this property to the PAM representation, where each value in the matrix also represents a specific type of the relation, the values of $P^k[i, j]$ allow us to keep track of the relational chain linking two nodes.

For instance, the second-order PAM for the example graph of Fig. 1 will be:

$$P^2 = P \times P = \begin{pmatrix} 0 & 35 & 0 & 0 & 15 \\ 0 & 0 & 0 & 35 & 0 \\ 0 & 0 & 0 & 0 & 35 \\ 0 & 30 & 15 & 0 & 35 \\ 21 & 49 & 21 & 0 & 0 \end{pmatrix}.$$

Let us examine the values of this matrix, by starting with the node pair (A, B) for which we have $P^2[0, 1] = P^2[A, B] = 35$. We can see from Fig. 1, that we can get from node A to node B in two hops only through node C, by following the directed path $A \xrightarrow{5} C \xrightarrow{7} B$. The relations 5 and 7 that are exhibited along this 2-hop path, are directly expressed in the value of $P^2[A, B] = 35$, through its prime factors, as $35 = 5 * 7$. The same goes for the rest of the matrix: there is $P[A, E] = 15 = 3 * 5$ corresponding to $A \xrightarrow{3} B \xrightarrow{5} E$, and also $P[E, A] = 21 = 7 * 3$ corresponding to $E \xrightarrow{7} B \xrightarrow{3} A$, and so on. Hence, using this representation the products in P^k express the relational k -chains linking two nodes in the graph.

It is also important to note the case of $P^2[D, B] = 30$, which is the sum of the two possible paths $30 = 9 + 21 = 3 * 3 + 3 * 7$, corresponding to paths $D \xrightarrow{3} A \xrightarrow{3} B$ and $D \xrightarrow{3} C \xrightarrow{7} B$ accordingly. This case shows that each cell (i, j) in P^k aggregates all “path-products” of k -hops that lead from i to j . This is aligned with the notion of adjacency matrix powers in classical graph theory, with the added benefit of encoding the types of relations in the value of the cell.

Moreover, we can easily extract structural characteristics for nodes, pairs, subgraphs, and the whole graph, by looking up these higher-order PAMs. For instance, we can calculate the frequency of the two-hop paths as in the one-hop case, by simply counting the occurrences of non-zero values in P^2 , which in this case are: $\{15 : 2, 21 : 2, 30 : 1, 35 : 4, 49 : 1\}$. These can be used for further analysis according to the task at hand. In general, k -order structural characteristics about the graph can be easily extracted through simple operations on the corresponding P^k .

It is important to note that the value $P^2[D, B] = 30$ could be also decomposed into $30 = 15 + 15 = 3 * 5 + 3 * 5$, rather than $9 + 21 = 3 * 3 + 3 * 7$ which is the actual case. In this case, without any further validation (e.g. checking whether the node B has incoming edges of type 5 that would indicate that the first decomposition is correct), the exact paths can't be reconstructed using the value 30 alone. This means that in k -hop PAMs we have some loss of information (for $k > 1$, as the 1-hop PAM is lossless as shown in Eq. (2)). As the k -hop PAMs are lossy by design due to such collisions (i.e. aggregates of different paths that result in the same sum value) and for computational reasons, we will use P_+ as introduced in Eq.(3) as a starting point for the calculations of P^k , as both P and P_+ will result in lossy P^k .

To sum up, if we want to represent the full graph G without loss, using a 1-hop matrix, we will need to use Eq. (2) and generate P . When we are interested in calculating the k -hop PAMs, it is better to start directly with P_+ from Eq. (3). Nonetheless, these collisions are not so common in real-world scenarios (and can be greatly reduced with sparsely-spaced primes, which is not the scope of this work), allowing k -hop PAMs to retain useful information, despite their lossy nature, as we will showcase in the following section.

3 Applications

In the following subsections, we will showcase the usefulness of the framework, first by showcasing its usability and then by utilizing the PAMs to generate expressive feature vectors for downstream tasks.

3.1 Calculating Prime Adjacency Matrices

To showcase the usability of the PAM representation and the simplicity of the calculations needed, we used some of the most common benchmark knowledge graphs and generated their P^k matrices. Specifically, we experimented on WN18RR [4], YAGO3-10 [11] FB15k-237 [17], CoDEX-S [13] and HetioNet [6]. The first three are some of the most well-known and commonly used datasets for link prediction in knowledge graphs, CoDEX-S was selected to showcase results in a small use case, while, on the other hand, the biomedical knowledge graph HetioNet was selected as the largest use case (in terms of the number of edges).

Table 1. Main characteristics of KG datasets and the time needed to calculate the corresponding P^5 PAMs.

Dataset	N	R	# Edges	P^5 (sec.)
CoDEX-S	2,034	42	32,888	0.2
WN18RR	40,493	11	86,835	0.3
FB15k-237	14,541	237	272,115	39.0
YAGO3-10	123,182	37	1,079,040	23.9
HetioNet	45,158	24	2,250,197	213.2

The basic characteristics of the datasets can be seen in Table 1, along with the total time needed to set up PAM and calculate all PAMs up to P^5 . We can see that for small and medium-scale KGs the whole process takes less than a minute. Interestingly, the time needed to calculate P^5 for HetioNet is disproportionately longer than for YAGO3-10, which is of comparable size, and this is mainly due to the structure of the dataset. It is 5 times denser, leading to denser PAMs, which takes a toll on the time needed for their calculation. Still, the whole process is completed in a matter of minutes. As a final note here, we have not optimized the generation procedure for PAMs, and we simply multiply sparse matrices iteratively.

3.2 Relation Prediction

The first task which we will tackle using PAMs is *relation prediction*. This task consists of predicting the most probable relation that should connect two existing nodes in a graph. Essentially, we need to complete the triple $(h, ?, t)$ where h is

the head entity and t is the tail entity, by connecting them with a relation r from a set of known relations.

Our idea is that we can use the PAMs to construct expressive feature vectors for each pair, as they contain complex relational patterns. To this end, we devised a nearest neighbor scheme, where for each training sample (h, r, t) , the pair (h, t) is embedded in a feature space as a point and r is used as a label for that point. This feature space is created by the user utilizing the PAMs as shown next. At inference time, given a query pair (h_q, t_q) , we embed it in the same space and the missing relation is inferred via the labels (i.e. relations) of its nearest neighbors. Because of the simplicity of this approach (no trainable parameters), the representation of the pairs must be rich enough to capture the semantics needed to make the correct prediction.

To create such a representation for a given pair of nodes (h, t) we designed a simple procedure, that utilizes both information about the nodes h, t , and the paths that connect them. Specifically, the feature vector for a pair (h, t) is simply the concatenation of the representations for the head node, the tail node, and the paths that connect them. More formally, we express this as:

$$R(h, t) = [Path(h, t) || Path(t, h) || R(h) || R(t)] \tag{4}$$

where $Path(h, t)$ denotes the feature vector of the path connecting h to t , $R(x)$ denotes the feature vector of node x and the symbol $||$ denotes the concatenation of vectors.

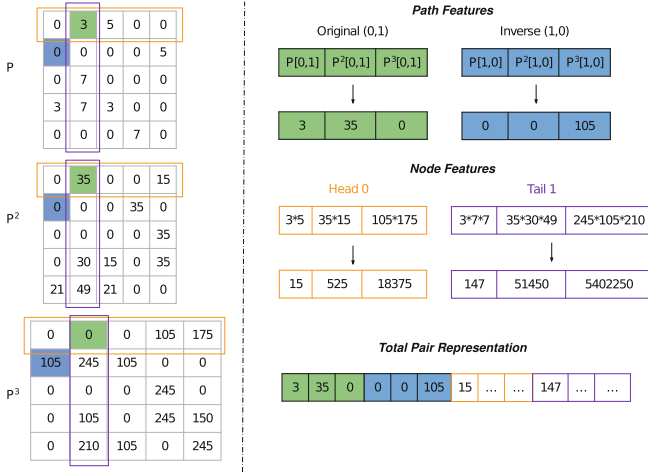


Fig. 2. The construction of the feature vector of the pair (0, 1) for the graph in Fig. 1, using the PAMs up to P^3 . On the left side, we see the PAM matrices, while on the right side the feature vector construction process. From top to bottom, we have the pair features, the node features, and the final pair representation, which is the concatenation of the individual vectors.

This procedure is highlighted in Fig. 2 for the pair $(0, 1)$ of the graph in Fig. 1. First, we create the feature vectors for the paths that connect the head to the tail and vice versa. The feature vector $Path(h, t)$ is simply a k -sized vector where each cell contains the corresponding value from the cell $P^k[h, t]$, that is:

$$Path(h, t) = [P[h, t], P^2[h, t], \dots]$$

We can see in the top-right of Fig. 2 that creating these path feature vectors is very easy; essentially accessing the values of the appropriate P^k matrix cells. In the example shown, the green ones correspond to the original path from $0 \rightarrow 1$, while the blue ones correspond to the inverse path from $1 \rightarrow 0$.

For the feature vectors of the head (tail) entity, we simply keep track of the products of the non-zero elements of the corresponding row (column), which essentially expresses the outgoing (incoming) relations and metapaths that the node exhibits. For the head h (tail t) entity, this simply is:

$$R(h) = [\prod P[h, :], \prod P^2[h, :], \dots] \quad R(t) = [\prod P[:, t], \prod P^2[:, t], \dots]$$

The idea of using different representations for the head and the tail (i.e. using the rows that represent the outgoing paths for the head, versus using the columns that represent the incoming paths for the tail), accentuates the different roles these entities play in a relational triple. Finally, using the above representations for the paths and the entities themselves, we simply concatenate them to create the final feature vector for the pair as per Eq. (4). In the example of Fig. 2, the outcome can be seen in the bottom right.

In order to evaluate this model, which we name *PAM-knn* we will follow the experimental setup presented in [19]. There, the authors experiment on this task with 6 KG datasets, but we will focus on the 3 most difficult ones which were: NELL995 [21], a collection of triples extracted from [3], WN18RR [4], as introduced in Sect. 3.1, and DDB14, which was created by the authors and is based on the Disease Database, a medical database containing biomedical entities and their relationships.

We compare *PAM-knn* to several widely used graph embedding models, namely TransE [2], ComplEx [18], DistMult [22], RotatE [16] and QuatE [23]. For a given entity pair (h, t) in the test set, we rank the ground-truth relation type r against all other candidate relation types according to each model. We use *MRR* (mean reciprocal rank) and *Hit@3* (hit ratio with cut-off values of 3) as evaluation metrics, as in the original work. The performance of the competing models is reported as found in [19]. More details on the models, their hyper-parameters, and the experimental setup can be found in the original article. Regarding the hyper-parameters of *PAM-knn*, namely the number of neighbors to take into account and the power of k in P^k , we used the validation part of each dataset to perform a simple grid-search and select the optimal ones.

The results of the experiments are presented in Table 2. Once again, our goal was to highlight the usefulness of the PAM framework using a simple model that heavily relies on the expressivity of the framework and performs comparably well. We can see from Table 2 that the simple *PAM-knn* approach outperformed the competing models in WN18RR, while in the other 2 datasets its performance

was not far from the competition. It is important to note here that *PAM-knn*, has no trainable parameters and the whole procedure can be completed in a few minutes on all datasets using a CPU, while the competing models were trained for hours using a GPU. This can be seen in the last column, where the number of trainable parameters for each model on DDB14 are reported, which are in the order of millions for the embedding models, while PAM-knn has none.

Table 2. Results of relation prediction on all datasets. The best results are highlighted in bold. For DDB14 the number of parameters for each model is shown as well.

	NELL995		WN18RR		DDB14		
	MRR	H@3	MRR	H@3	MRR	H@3	# Params
TransE	0.784	0.870	0.841	0.889	0.966	0.980	3.7M
CompleX	0.840	0.880	0.703	0.765	0.953	0.968	7.4M
DistMult	0.847	0.891	0.634	0.720	0.927	0.961	3.7M
RotatE	0.799	0.823	0.729	0.756	0.953	0.964	7.4M
QuatE	0.823	0.852	0.752	0.783	0.946	0.962	14.7M
PAM-knn	0.740	0.843	0.852	0.957	0.915	0.961	0

To sum up, we presented a simple model for relation prediction that relies on the expressive representations of node pairs, which can be naturally constructed using the PAMs and, as shown in the results, performs on par with many widely used graph embedding methodologies. The method is very fast, has no trainable parameters, and can be used as a strong baseline for relation prediction. We could devise more sophisticated representations for node pairs or train a model using the same feature vectors in a supervised setting, but this simple approach serves to highlight the inherent expressiveness and efficiency of the PAM framework.

3.3 Graph Classification

Another application in which the structural properties of the graph play an important role is *graph classification*. There are several ways to use the P^k matrices of a given graph to capture complex relational patterns. In this work, we propose the following simple procedure. First, we calculate all the PAMs up to a pre-defined k . Then, from each matrix P^k , we calculate the product of its non-zero values $g_k = \prod_{P^k[i,j]>0} P^k[i,j]$ as a single representative feature for the matrix.

For example, as we can see in the top matrix of Fig. 2, the resulting g_1 would be $g_1 = 231525$, the product of all the primes in the graph (i.e. $\{3, 3, 3, 5, 5, 7, 7, 7\}$). We call this approach *PowerProducts (PP)*.

The intuition behind PP lies in the fact that these non-zero values express the paths found at that k -hop. By design, the g_k number captures the information of the distribution of different relations and k -hop paths in the graph in a single number, which acts as a “fingerprint” for the structure of the graph. Having

generated all individual g_k values, up to a certain k , the final feature vector that represents the graph is simply $F(G) = [g_1, g_2, \dots, g_k]$. By combining all g_k in a feature vector, we aim at capturing the structure of the different sub-graphs residing in a complex graph.

We experimented with the PP method in the task of graph (binary) classification, utilizing the benchmark datasets from [12]. We use the multi-relational ones, which are mainly small-molecule datasets, with graphs being molecules exhibiting specific biological activities. It is also worth noting that all the nodes have labels in these experiments (i.e. the type of the atom). We compared the PowerProducts approach, in terms of time and accuracy, versus one of the best-performing graph kernels [9], the Weisfeiler-Lehman Optimal Assignment (WL-OA) kernel [8]. We opted for comparison with graph kernels, as they are computationally efficient and not far from the state-of-the-art in these datasets. Simulating the behavior of graph kernels, we used a Radial Basis Function (RBF) kernel to calculate the similarity of the graphs given their PP feature vectors.

Moreover, we created a variant that takes into account the node labels, using a Vertex Histogram (VH) kernel [15]. We call this variant *PP-VH*, and it simply is the average of the similarity matrices as generated by the PP and VH models. This variant will allow us to check the impact of utilizing the node-label information, which is not used in PP. As this is a classification task, we use a Support Vector Machine (SVM) (with the similarity kernel precomputed by the underlying model). We use a nested cross-validation (cv) scheme, with an outer 5-fold cv for evaluation and an inner 3-fold cv for tuning the penalty parameter C of the SVM. The rest of the parameters for WL-OA and VH are left to their default values, as proposed in [9]. For PP and PP-VH we use $k = 3$ for all datasets.

Table 3. Results on graph classification. Firstly, the characteristics of the datasets are shown (averaged per dataset). Then, the presented values are percentage point differences in the performance of the proposed models versus the WL-OA kernel.

Dataset	Graphs	Nodes	Edges	PP		PP - VH	
				$\Delta Acc\%$	$\Delta Time\%$	$\Delta Acc\%$	$\Delta Time\%$
AIDS	2,000	15.69	16.20	-1.45	-99.65	+0.40	-99.44
BZR_MD	306	21.30	225.06	+5.66	-94.78	+12.22	-88.42
COX2_MD	303	26.28	335.12	-20.20	-95.79	+0.15	-93.26
DHFR_MD	393	23.87	283.01	-16.99	-93.78	-48.90	-18.50
ER_MD	446	21.33	234.85	-0.38	-92.10	+7.45	-89.90
MUTAG	188	17.93	19.79	-8.69	-89.88	+2.21	-82.93
Mutagenicity	4,337	30.32	30.77	-28.53	-99.75	-20.23	-95.72
PTC_FM	349	14.11	14.48	-2.99	-95.37	-18.06	-67.87
PTC_FR	351	14.56	15.00	-10.25	-94.16	+10.97	+234.68
PTC_MM	336	13.97	14.32	-18.35	-93.99	-7.73	-27.64
PTC_MR	344	14.29	14.69	-9.83	-95.45	-14.39	-72.10

The results are shown in Table 3. For brevity, we report the percent change in accuracy $\Delta Acc\%$ and time $\Delta Time\%$ of PP and PP-VH over the results of WL-OA which is used as a strong baseline. In terms of accuracy, the WL-OA outperforms PP in almost every dataset, except BZR_MD. However, PP-VH, which uses the node-label information, outperforms WL-OA in 6/11 datasets. The good results of PP-VH indicate that for many datasets in the small-molecule classification task, structure alone is not sufficient and the types of the atoms in the molecule play an important role as well. In terms of computational performance, PP takes up only 10% of the WL-OA runtime, across all datasets. Even when adding the VH kernel, the resulting model is more than 65% faster than WL-OA in 8/11 datasets. Thus, using the PP-VH variant we can have less than half of the WL-OA runtime, while also improving the performance in 6/11 datasets.

To sum up, we have proposed a simple graph representation methodology that capitalizes on the information captured by PAMs and can be used for graph classification. We showcased its usefulness, as it is (on average) much faster while also performing comparably or better with one of the best graph kernels. Moreover, the proposed feature extraction methodology using PAMs is a very simple one and more sophisticated ones may yield greater results.

4 Conclusions

In this work, we presented the Prime Adjacency Matrix (PAM) framework for complex networks. It is a compact representation that allows representing the one-hop relations of a network losslessly through a single-adjacency matrix. This, in turn, leads to efficient ways of generating higher-order adjacency matrices, that encapsulate rich structural information. We showcased that the representations created are rich enough to be useful in different downstream tasks, even when utilized by simple models. These models perform on par with commonly-used graph models, while greatly improving the runtime. In the future, we aim to strengthen the methodology, by addressing details of the framework (e.g. the effect of the mapping function ϕ) and experimenting on other downstream tasks.

References

1. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U.: Complex networks: structure and dynamics. *Phys. Rep.* **424**(4–5), 175–308 (2006)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: *Advances in Neural Information Processing Systems*, vol. 26 (2013)
3. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: *Twenty-Fourth AAAI Conference on Artificial Intelligence* (2010)
4. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32 (2018)

5. Edwards, G., Nilsson, S., Rozemberczki, B., Papa, E.: Explainable biomedical recommendations via reinforcement learning reasoning on knowledge graphs. arXiv preprint [arXiv:2111.10625](https://arxiv.org/abs/2111.10625) (2021)
6. Himmelstein, D.S., et al.: Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *Elife* **6** (2017)
7. Ji, S., Pan, S., Cambria, E., Marttinen, P., Philip, S.Y.: A survey on knowledge graphs: representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* (2021)
8. Kriege, N.M., Giscard, P.L., Wilson, R.: On valid optimal assignment kernels and applications to graph classification. In: *Advances in Neural Information Processing Systems*, vol. 29 (2016)
9. Kriege, N.M., Johansson, F.D., Morris, C.: A survey on graph kernels. *CoRR abs/1903.11835* (2019)
10. Liu, G., Yang, Q., Wang, H., Lin, X., Wittie, M.P.: Assessment of multi-hop interpersonal trust in social networks by three-valued subjective logic. In: *IEEE INFOCOM Conference on Computer Communications*, pp. 1698–1706. IEEE (2014)
11. Mahdisoltani, F., Biega, J., Suchanek, F.M.: YAGO3: a knowledge base from multilingual wikipedias. In: *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research*, Asilomar, USA (2015)
12. Morris, C., Kriege, N.M., Bause, F., Kersting, K., Mutzel, P., Neumann, M.: TUDataset: a collection of benchmark datasets for learning with graphs. In: *GRL+ Workshop in ICML 2020 Workshop* (2020)
13. Safavi, T., Koutra, D.: Codex: a comprehensive knowledge graph completion benchmark (2020)
14. Sato, R.: A survey on the expressive power of graph neural networks. arXiv preprint [arXiv:2003.04078](https://arxiv.org/abs/2003.04078) (2020)
15. Sugiyama, M., Borgwardt, K.: Halting in random walk kernels. In: *Advances in Neural Information Processing Systems*, vol. 28 (2015)
16. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: knowledge graph embedding by relational rotation in complex space. arXiv preprint [arXiv:1902.10197](https://arxiv.org/abs/1902.10197) (2019)
17. Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., Gamon, M.: Representing text for joint embedding of text and knowledge bases. In: *Proceedings of the EMNLP Conference*, pp. 1499–1509 (2015)
18. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: *International Conference on Machine Learning*, pp. 2071–2080. PMLR (2016)
19. Wang, H., Ren, H., Leskovec, J.: Relational message passing for knowledge graph completion. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery, Data Mining, KDD 2021*, pp. 1697–1707. NY, USA (2021)
20. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **29**(12), 2724–2743 (2017)
21. Xiong, W., Hoang, T., Wang, W.Y.: DeepPath: a reinforcement learning method for knowledge graph reasoning. *CoRR abs/1707.06690* (2017)
22. Yang, B., Yih, W.T., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint [arXiv:1412.6575](https://arxiv.org/abs/1412.6575) (2014)
23. Zhang, S., Tay, Y., Yao, L., Liu, Q.: Quaternion knowledge graph embeddings. In: *Advances in Neural Information Processing Systems*, vol. 32 (2019)
24. Zhou, J., et al.: Graph neural networks: a review of methods and applications. In: *AI Open*, pp. 57–81 (2020)

Network Analysis



Approximation Algorithms for k -Median Problems on Complex Networks: Theory and Practice

Roldan Pozo^(✉)

National Institute of Standards & Technology, Gaithersburg, MD 20899, USA

pozo@nist.gov

<https://www.nist.gov/people/roldan-pozo>

Abstract. Finding the k -median in a network involves identifying a subset of k vertices that minimize the total distance to all other vertices in a graph. While known to be computationally challenging (NP-hard) several approximation algorithms have been proposed, most with high-order polynomial-time complexity. However, the graph topology of complex networks with heavy-tailed degree distributions present characteristics that can be exploited to yield custom-tailored algorithms. We compare eight algorithms specifically designed for complex networks and evaluate their performance based on accuracy and efficiency for problems of varying sizes and application areas. Rather than relying on a small number of problems, we conduct over 16,000 experiments covering a wide range of network sizes and k -median values. While individual results vary, a few methods provide consistently good results. We draw general conclusions about how algorithms perform in practice and provide general guidelines for solutions.

1 Introduction

The k -median problem is an important and fundamental problem in graph theory, and various application areas. Given a connected network, it seeks to find k vertices which are the closest (in terms of average distance) to the remaining vertices in the network graph. This is crucial in the spread of viral messages in social networks, disease contagion in epidemiological models, operation and distribution costs for goods and services, marketing and advertising, design layout of communication networks, and other wide-ranging applications.

Finding such an optimal set of vertices is referred to as the influence maximization problem [8] and numerous algorithms have been proposed to address this issue. Although these algorithms were not explicitly designed for the k -median problem, they share mathematical similarities, as they attempt to find influential vertices that are similarly well-connected and can quickly disseminate information throughout the network. The level of *influence* can be measured in various ways, typically employing diffusion models such as independent cascade model [4] or epidemiological models of the Susceptible-Infected (SI) and

Susceptible-Infected-Recovered (SIR) [5] types. Formally, the k -median problem on network graphs has been shown to be NP-hard (in terms of k) by reduction to the dominant cover problem [7] and is computationally intractable for large network graphs.

2 Notation and Definitions

Definition 1. k -median problem: *Given an integer k and a connected graph $G = (V, E)$, find a set of S of k vertices that minimize the summation of distances from S to the remaining vertices of G .*

Using $d(v, u)$ as the distance (length of shortest path) between vertices v and u , we can define the distance $d(v, S)$ between a single vertex and a vertex set S as the minimum distance between v and any vertex in S . We can then define the *average distance* $A(S)$ as

$$A(S) \doteq \frac{\sum_{v \in V} d(v, S)}{|V - S|} \quad (1)$$

In this context, we can restate the k -median problem on a network graph G to be the identification of a set of k -vertices which minimize the average distance to the remaining $|V| - k$ vertices:

$$M^*(k) \doteq \min_{|S|=k} A(S) \quad (2)$$

We refer to $M^*(k)$ as the true *optimal value* of the k -median problem on a graph G , and let $M(k)$ denote an approximation to this solution using the methods described in Sect. 3. When necessary, we use $M_{\text{method}}(k)$ to avoid ambiguity.

3 Approximation Algorithms and Related Problems

The field of approximation methods to the k -median problem is quite large. Resse [17] provides a comprehensive overview of over 100 methods from linear programming, genetic algorithms, simulated annealing, vertex substitution, and other approaches. In general, the algorithm with the best guaranteed approximation ratio is a local search and swap method [1] which provides a bound of $3 + 2/p$ where p is the number of vertices simultaneously swapped. Its computation time is $O(n^p)$, where n is the number of vertices. Thus, even for a quadratic-order complexity $O(n^2)$, which is quite limiting for large networks, the best guarantee we can get is a factor of four from optimal. While these approaches were adequate for small networks, the higher-order polynomial time complexity makes them unfeasible for networks with thousands or million of vertices [18].

Instead, researchers have turned their attention to algorithms for finding effective spreaders in connected networks with heavy-tailed degree distributions, often employing an Susceptible-Infected (SI) or Susceptible-Infected-Recovered

(SIR) model of spread [14, 15], where $I(t)$ denotes the number of infected nodes at time t , and $I(0)$ is the number of initially infected nodes. The k -median can be thought of a special case of an SI model where the probability of an infected node transmitting the disease to a susceptible neighbor is 1.0, or an SIR model with the probability of recovery for each node is 0.0. In which case, the solution to the k -median problem can be thought of as maximizing the integral of the number of infected nodes over the propagation steps, starting with k initial infected vertices.

3.1 Degree Ordering

Approximating the k -median solution as the top k hubs of the network is perhaps the most straightforward approach:

$$X_{\text{degree}}(k) \leftarrow \underset{v \in V}{\operatorname{argmax}[k]} \deg(v) \quad (3)$$

The idea here is that the hubs (high-degree vertices) serve as efficient spreaders since they are connected to large number of neighbors. This is countered by the notion that there may be significant overlap among their aggregate neighborhoods, with other vertices potentially covering the graph more effectively. This is a common criticism of degree ordering for this problem, but our experimental results show that this may not be as critical an issue in practice (see Sect. 6).

3.2 Extended Degree Ordering

A more sophisticated approach is the extended degree ordering, which measures the sum of degrees for neighboring vertices, and uses the top k values as an approximation of the k -median solution:

$$X_{\text{degree}+}(k) \leftarrow \underset{v \in V}{\operatorname{argmax}[k]} \sum_{x \in N(v)} \deg(x) \quad (4)$$

This is a semi-local algorithm, utilizing more information about the network's topology by analyzing the second-level neighborhood, i.e. neighbors of neighbors. The motivation for this centrality measure is that it uses more information about the graph topology and can lead to an improved metric for identify candidate vertices for the set S .

3.3 PageRank Ordering

PageRank is a variant of the eigenvalue centrality, which treats the network as a flow graph and values vertices with high eigenvalues. It is the basis for some commercial web search engines. [14]. Given a damping factor, $0 \leq \delta \leq 1$, the PageRank centrality is given as the convergence of the iteration

$$\text{PageRank}(v) = (1 - \delta) + \delta \sum_{u \in N(v)} \frac{\text{PageRank}(u)}{\deg(u)} \quad (5)$$

typically a value of $\delta = 0.85$ is used in these calculations [14]. The corresponding approximation for the k -median solution is

$$X_{\text{PRank}}(k) \leftarrow \operatorname{argmax}_{v \in V}[k] \text{PageRank}(v) \quad (6)$$

3.4 VoteRank Ordering

A method developed by Zhang *et al.* [18], is the VoteRank algorithm, which uses an iterative voting methodology to determine the best influencer nodes. Each vertex i has a pair of values (S_i, T_i) denoting the collective (incoming) votes from neighbors S_i and the number of (outgoing) votes to give out in each voting round, T_i . At each voting round (complete pass through the graph) a vertex with the maximum (incoming) vote score is selected (i^*) and its (S_i^*, T_i^*) values are set to zero, effectively taking it out of future voting in subsequent rounds. The neighbors of vertex i^* have their respective T_i votes reduced by a fixed value f , and the process is repeated until k vertices are found. In their paper, the authors use $f = 1/\langle d \rangle$, where $\langle d \rangle$ is the average degree of the graph, and this value is fixed throughout the algorithm. Typically, one would choose f such that $kf \ll 1$ but this implementation does not allow the T_i values to go negative. The VRank k -median approximation is given by

$$X_{\text{VRank}}(k) \leftarrow \operatorname{argmax}_{v \in V}[k] \text{VoterRank}(v) \quad (7)$$

3.5 Coreness Ordering

Another vertex centrality measure that has been proposed for finding effective spreaders is based on the degeneracy of network graphs. The i -core of a graph is collection of connected components that remain after all vertices with degrees less than n have been removed. (This is often referred to as the k -core of a graph in the literature, but we use i to avoid conflict with the k used in the k -median formulation.) To compute the i -core of a graph, we remove all vertices of degree $i - 1$ or less. This process is repeated until there are no vertices of the graph with degrees less than n . The notion here is that vertices in higher value i -cores represent the inner backbone of the network, as opposed to lower-valued i -cores which lie at its periphery, and serve as better-connected vertices to efficiently spread information throughout the network. The *core-number* of a vertex is the largest value i which it belongs to the i -core of the graph. Although one could use this centrality to identify candidate vertices [9], one problem that has been noted is that the core values of the highest vertices are often the same and hence are not distinguishable to form a proper ordering [2]. To remedy this, a slightly extended centrality has been proposed that replaces the i -shell value of a vertex with the sum of its neighbors' core-number. That is, if $c(v)$ is the core-number of v , then the **core** algorithm is

$$X_{\text{core}}(k) \leftarrow \operatorname{argmax}_{v \in V}[k] \sum_{u \in N(v)} c(u) \quad (8)$$

3.6 Extended Coreness Ordering

Extensions to the $C(v)$ centrality have also been proposed [2] as an improved measure for influence. In a similar manner to deg^+ , *neighborhood coreness*, or core^+ , uses the values of its neighbor’s core centrality. We refer to this algorithm as **core+**

$$X_{\text{core}^+}(k) \leftarrow \operatorname{argmax}_{v \in V}[k] \sum_{u \in N(v)} C(u) \tag{9}$$

3.7 H-Index Ordering

The Hirsch index or H -index [6], originally intended to measure the impact of authors and journals by way of citations, has also been studied as centrality to measure ranking of influence and its relation to other centralities [12]. The original measure for an author or journal was determined as the number of n publications that have at least n citations. In terms of a network graph, the Hirsch index of a vertex v , given as $H(v)$, can be represented as the maximal number of n neighbors that each have at a degree of n or more. That is, if $h(v, n)$ is the number of neighbors of v with degree at least n ,

$$h(v, n) \doteq \{e \mid \text{deg}(e) \geq n, e \in N(v)\} \tag{10}$$

then

$$H(v) \doteq \max_n \{|h(v, n)| \geq n\} \tag{11}$$

The k -median approximation can be then be given as the **H-index** algorithm:

$$X_{\text{H-index}}(k) \leftarrow \operatorname{argmax}_{v \in V}[k] H(v) \tag{12}$$

3.8 Expected Value (Random)

The **mean** average-distance of every k -element vertex set is simply the expected value of all possible combinations:

$$E^*(k) \doteq \frac{1}{\binom{|V|}{k}} \sum_{|S|=k} A(S) \tag{13}$$

That is, the average value of a **random** guess chosen from a uniform distribution of all $\binom{|V|}{k}$ possible sets. This can be computed exactly by brute force for small networks and small k -values. For larger cases, the expected value is approximated by sampling a finite subset of these possibilities and use of the Central Limit Theorem (CTL).

4 Experiments and Methodology

For these experiments, we focused on connected simple graphs that were undirected, unweighted, with no self-loops or multi-edges. This represents the least common denominator for graph topologies, as not all datasets have edge weights and other metadata. Directed graphs were represented as undirected by making each edge bi-directional. For disconnected networks, we used the largest connected component. Additionally, the input networks had their vertices renumbered to be contiguous for optimized operations, and therefore did not necessarily match the vertex numbers in the original sources.

Table 1. Application network topologies used in this study (largest connected component of undirected graph). The average degree is $\langle d \rangle$ and the maximum degree is Δ .

Network	Application	$ V $	$ E $	$\langle d \rangle$	Δ	$\Delta/\langle d \rangle$
Zebra	animal contact network	23	105	9.13	14	1.5
Dolphin	animal contact network	62	159	5.13	12	2.3
Terrorist network	social network	64	243	7.59	29	3.8
High School	social network	70	274	7.83	19	2.4
MIT students	mobile social network	96	2,539	52.90	92	1.7
Hypertext 2009	social interaction	113	2,196	38.9	98	2.5
Florida ecosystem wet	food network	128	2,075	32.42	110	3.4
PDZBase	metabolic network	161	209	2.59	21	8.1
Jazz	collaboration network	198	2,742	27.79	100	3.6
GE_200	top-level web graph	200	1,202	12.02	124	10.3
Chevron_200	top-level web graph	200	5,450	54.50	189	3.5
Abilene218	computer network	218	226	2.07	10	4.8
Bethesda	top-level web graph	255	422	3.31	81	24.5
<i>C. Elegans</i>	neural network	297	2,148	14.46	134	9.3
NetScience	co-authorship	379	914	4.82	34	7.0
Arenas-email	email communications	1,133	5,451	9.62	71	7.4
FAA air traffic	infrastructure	1,226	2,408	3.9	34	8.6
Human protein	protein interaction	2,217	6,418	8.94	314	26.5
ca-GrQc	co-authorship	4,158	13,422	6.46	81	12.5
ca-HepTh	co-authorship	8,638	24,806	5.74	65	11.3
ca-HepPh	co-authorship	11,204	117,619	23.38	491	21.0
ca-CondMat	co-authorship	21,363	91,286	8.54	279	32.6
email-Enron	email communications	33,696	180,811	10.73	1,383	128.9
cit-HepPh	citation network	34,401	420,784	24.46	846	34.6
flickrEdges	online social network	105,722	2,316,668	43.83	5,425	123.8
email-EuAll	email communications	224,832	339,925	3.02	7,636	2,525.3
com-YouTube	online social network	1,134,890	2,987,624	5.27	28,754	5,631.3
soc-Pokec	online social network	1,632,803	22,301,964	27.32	14,854	543.8
soc-LiveJournal	online social network	4,846,609	42,851,237	17.68	20,333	1,149.9

The dataset comprised a wide range of application areas, including social, mobile, metabolic, neural, email, biological, and collaboration networks listed in Table 1. Examples were collected from network databases Konect [10], SNAP [11], and UC Irvine [3], as well as several webgraphs generated by examining public websites. Network sizes ranged from less than 100 vertices (for exact verification of k-median problems) to networks with over 1 million vertices, with most networks containing several thousand vertices. This study focused on 32 of these networks, comparing the eight algorithms from Sect. 3 for k-values from 1 to 100, resulting in roughly 16,000 experiments of graph, algorithm, and k-value combinations. This provided a clearer view of the performance landscape and algorithm behavior.

Table 2. Average error (%) to true-optimal for small graphs ($1 \leq k \leq 5$)

Network	random	degree	degree+	VRank	PRank	core	core+	H-index
Abilene218	68.9	11.1	3.4	10.2	54.0	4.4	3.4	8.4
USAir87	60.0	2.6	2.7	2.6	3.1	2.7	2.7	10.7
ca-HepTh	46.4	4.7	5.8	4.7	4.7	26.2	26.2	34.9
ca-netscience	68.6	32.7	65.3	18.0	17.0	56.0	67.0	66.8
celegans	50.2	1.8	2.1	1.8	3.3	1.8	2.1	25.9
faa	48.4	10.7	5.0	10.7	8.9	4.8	5.3	12.6
foodweb_florida_wet	53.8	5.6	5.6	5.6	5.6	5.6	5.6	5.0
hypertext_2009	38.8	3.3	0.6	3.0	3.3	0.6	0.6	7.5
jazz	43.7	7.7	10.1	7.7	5.8	10.1	10.1	15.0
pdzbase	64.3	10.7	22.9	10.7	10.7	20.0	14.2	32.6

Computational experiments were conducted on a desktop workstation, running Ubuntu Linux 5.15.0-46, with an AMD Ryzen 7 1700x (8-core) processor running at 3.4 GHz, and outfitted with 32 MB of RAM¹. The algorithms were coded in C++, and compiled under GNU g++ 11.1.0 with the following optimization and standardization flags: [-O3 -funroll-loops -march=native -std="c++11"]. Modules were used from the NGraph C++ library and Network Toolkit [16], as well as optimized C++ implementations of algorithms noted in the paper.

5 Results

The results of the computational experiments on real networks showed significant variations. Despite claims made for any particular approach, we did not see a

¹ Certain commercial products or company names are identified here to describe our study adequately. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products or names identified are necessarily the best available for the purpose.

single method consistently producing a winner in every case. Instead, we were presented with trade-offs for varying network topologies. Nevertheless, we were able to form general observations about the expected behavior on classes of networks and provide some guidelines for choosing appropriate methods.

5.1 Comparisons with Optimal k -median solutions

For small networks, we compared the accuracy of the approximation algorithms by running them for various values, up to $k = 5$, except where limited by the computation effort. The results reveal several interesting patterns: (1) guessing a solution (**random**) performs, on average, within a factor of 2 from optimal, (2) for $k > 2$ some approximation methods (**degree+**, **core**, **core+**, **H-index**) can perform **worse** than random guessing, (3) most methods (excluding **random**) stay within a 1.5 factor of optimal, (4) **VRank** and **PRank** seem to be perform best, staying within 1.2 of optimal for $1 \leq k \leq 5$, and (5) **core**, **core+**, and **H-index**, typically perform worse, underperformed only by **random**.

The *C. Elegans* network, for example, shows extremely good approximations (relative error less than 5%) for all methods, except for **random** and **H-index**. In the *Abilene218* network, we encounter quite different behavior: the **PRank** method performs substantially worse than every other one, except random guessing. This is in sharp contrast to other examples, where **VRank** and **PRank** methods perform similarly and are often outperformed other algorithms. Finally, the *USAir87* network illustrates that the approximation algorithms are capable of calculating good-quality solutions (even **H-index**) that are significantly better than random guessing.

Table 3. Ranking of methods by actual error (%) in small graphs: average relative errors for each method in Table 2

method	error (%)
VRank	7.5
degree	9.1
PRank	11.6
degree+	12.3
core	13.2
core+	13.7
H-index	21.9
random	54.3

Table 2 provides a tabular form of similar results from a larger study of 10 networks. From this data we see that the behavior of these methods on real networks can vary significantly. Ignoring **random** and **H-index** momentarily, the remaining competitive methods can be quite accurate for these networks. For example,

C Elegans, *hypertext_2009*, *USAir87* and *foodweb_florida_wet* all exhibit approximations that are within 5% of optimal. The *ca-netscience* network was the sole outlier, with the best methods of the group exhibiting roughly a 20% error.

Taking the average error for each method across the networks we arrive at (Table 3) illustrating that **VRank** performed the best overall, with the other methods not too far behind. In this experiment, **degree+**, **core**, and **core+** did not perform as badly, while **H-index** and **random** fared significantly worse. This table represents the analysis for small network comparisons with exact solutions. It illustrates that the top methods generally work quite well, typically *within 10% to 20% of optimal* and seem reasonable candidates for testing on larger networks.

5.2 Case Studies: Million-Node Networks

Here we focus on three larger examples with millions of vertices and edges used in the study of large social networks [13]. For the *YouTube* network ($V = 1,134,890$ $E = 2,987,624$), the various heuristics perform about the same: roughly 35% better than a random guess. In this case, all methods yield nearly identical values, and one can simply use the fastest one (**degree**) to generate competitive results.

Table 4 lists the how each method ranked in the top 1%, 10%, and 100% of solutions. For example, **degree** scored in the top 10% solutions about 3/4 of the time, while random guessing always remained within a factor 2 of the best solution. Using the fastest method (**degree**) as a reference, we see that **VRank** and **PRank** provided the best solutions, but at a computational cost of nearly three orders of magnitude.

Similar results are seen for the *soc-pokec* social network ($V=1,632,803$, $E=22,301,964$). From these two examples, one may be tempted to conclude that the algorithms perform equally well for large networks. However, computations for the *LiveJournal* social network ($V=4,846,609$ $E= 42,851,237$) show a significant difference between various methods, with **PRank**, **VRank**, and **degree** performing better than most other heuristics and roughly 30% better than **random**.

Table 4. Percentage of cases where each method scored within $x\%$ of best solution ($k = 1, \dots, 100$) for million-vertex network (YouTube). Nearly all methods are within a factor of two of best solution.

method	0% (best)	1%	10%	100%
degree	12.5	24.0	75.7	100.0
degree+	8.5	11.5	42.7	99.9
VRank	20.4	44.2	91.9	100.0
PRank	18.5	33.7	91.6	100.0
core	9.1	15.5	50.1	99.0
core+	6.3	8.7	41.0	97.8
H-index	3.9	6.0	39.7	95.5
random	0.6	3.1	12.3	99.4

5.3 Overall Results

Table 5 describes the overall performance of approximation algorithms on the 32 networks under consideration. The values are described as relative error to the best solution for each method. For example, a value of 10 signifies that particular method performed on average within 10% above the best possible heuristic for each k from one to one hundred. From here we see that **VRank** and **PRank** come in first and second position, respectively, for the majority of cases while **degree** comes in a close third position.

Table 5. Quality of methods for large graphs ($k = 1, 2, \dots, 100$). Relative performance (%) from best solution. A value of 10 signifies that on average that method performed 10% above best possible value from all heuristics.

Network	degree	degree+	VRank	PRank	core	core+	H-index	random
Abilene218	8.2	11.1	3.2	10.1	8.6	12.8	13.4	79.7
USAir87	10.0	18.1	0.4	1.8	18.0	18.9	19.7	43.7
amazon0302	0.5	3.0	0.3	0.4	0.8	3.0	3.9	48.0
areans_email	3.3	9.4	0.0	0.8	7.3	11.1	10.9	30.9
as20000102	1.4	12.5	0.0	0.5	4.5	11.4	7.0	76.1
bethesda	3.6	10.9	0.8	1.5	17.4	11.5	40.7	76.8
ca-CondMat	4.3	11.3	1.3	0.0	11.6	13.9	12.6	40.0
ca-GrQc	40.1	51.3	1.3	1.4	51.7	51.7	56.2	37.9
ca-HepPh	11.7	13.1	1.4	1.1	14.1	14.7	16.6	20.2
ca-HepTh	4.8	21.1	0.2	0.9	61.5	63.0	47.4	34.4
ca-netscience	12.6	34.9	0.3	3.2	35.6	59.5	48.0	50.9
celegans	0.9	6.9	0.1	0.9	1.6	6.6	7.8	25.0
chevron_top200	0.0	0.5	0.0	0.0	0.5	1.1	1.1	7.8
cit-HepPh	3.2	11.3	2.3	0.0	8.8	15.2	18.8	38.6
com-youtube	0.8	2.7	0.1	0.4	2.5	3.2	3.6	51.3
d1mf	3.1	7.6	0.0	2.0	8.5	7.5	8.3	67.2
email-Enron	2.7	11.6	0.7	0.4	9.0	12.1	13.3	59.1
email-EuAll	2.4	11.8	2.4	3.2	5.0	10.5	12.5	66.9
faa	8.6	31.0	0.7	1.8	18.0	41.1	30.8	40.3
flickrEdges	8.2	79.4	2.3	0.5	88.8	89.2	89.5	35.6
foodweb_florida_wet	0.1	0.6	0.1	0.1	0.4	0.7	1.5	8.7
ge_top200	3.0	14.3	0.3	0.3	9.4	17.4	22.8	36.0
human_protein_gcc	2.7	30.4	0.1	1.3	6.2	30.0	20.6	58.4
hypertext_2009	0.3	0.0	0.3	0.3	0.0	0.0	0.8	8.5
jazz	5.1	12.4	1.2	0.1	14.2	14.8	13.6	11.2
p2p-Gnuetalla31	0.3	1.5	0.0	1.5	0.6	2.6	5.5	32.3
pdzbase	7.1	69.6	0.1	3.7	19.3	70.9	55.2	71.1
roadNet-PA	8.8	17.5	8.7	1.2	19.1	32.1	351.3	3.5
soc-Epinions	1.2	4.0	0.1	0.2	3.3	4.4	5.0	48.8
soc-Slashdot0922	0.8	1.8	0.2	0.5	1.1	2.8	3.6	45.8
web-Stanford	2.2	21.3	0.8	0.7	12.4	20.9	26.5	37.0
wiki-Vote	1.6	2.0	0.6	0.1	2.8	2.0	2.2	38.5

Table 6. Efficiency of k-median approximations on large networks: computation time (secs)

Network	degree	degree+	VRank	PRank	core	core+	H-index	random
<i>soc-LiveJournal</i>	0.01	0.8	50.8	35.2	5.4	11.5	5.6	166.9
<i>soc-pokec</i>	0.01	0.5	24.4	20.0	2.2	4.7	3.28	62.5
<i>com-youtube</i>	0.01	0.04	2.1	1.4	0.1	0.32	0.75	7.0

Table 7. Ranked performance of k-median approximations. A value of x signifies that $M_{method}(k)$, on average, was within $x\%$ of the best solution for each graph and k-value combination from Table 5.

method	performance (%)
VRank	0.9
PRank	1.3
degree	5.1
core	14.4
degree+	16.7
core+	20.5
H-index	30.3
random	41.6

Table 7 summarizes these results, where we compute the overall error of each method from the best solution for each k -value. For example, **degree** is typically about 5% greater than the best solution, while **random** produced, on average, a solution that was less than 50% greater than the best algorithm.

6 Conclusion

We have compared eight k -median approximation methods for various k -values (typically 1 to 100) on 32 networks over a diverse range of application areas. After conducting thousands of experiments, we have observed patterns and formulated guidance for solving the k -median problem on a broad range of application network problems. Overall, these approximation algorithms are efficient and some can produce good-quality solutions on complex networks. However, they do not replace traditional methods[17] for general graphs without heavy-tailed degree distributions.

We have demonstrated that the algorithms in this study can indeed yield high-quality results on smaller networks where we can compute the optimal solution explicitly (Sect. 5.1, Table 3) with **degree**, **VRank**, and **PRank** achieving roughly a 1.1 factor of the true solution. By contrast, the best algorithms for general graphs provide a guaranteed factor of 3 or higher.

For larger networks, the exact optimal solution is not computationally tractable and we can only compare the approximation methods against themselves, and one may reach an incomplete or premature conclusions by examining only a small number of networks. Exploring a larger and more diverse dataset, however, reveals certain patterns that aid in algorithm choices.

Like many approximation heuristics, the practical question comes down to a trade-off between performance (computational cost) and quality of solution. If one is willing to accept a factor of 2 from the best methods, then simply choosing k random vertices from an uniform distribution may suffice. (This may come as a unexpected result, as hard problems typically do not behave in this manner.) If a higher quality solution is needed, then we can consult Table 7 which summarizes the results of over 16,000 experiments. Here we see that **VRank** and **PRank** perform, on average, within about a 1.01 factor of the best method in every k -value in the [1:100] range. The simple **degree** method yields results on average within about 1.05 factor of the best method while exhibiting a performance speedup of three orders of magnitude over **VRank** and **PRank**.

Thus, we can form a general best-practices guide for choosing the appropriate algorithms:

- if a quality factor of 2 is sufficient, choose k random vertices from uniform distribution (**random**)
- if a better quality solution is needed, choose the top k hubs (**degree**)
- if quality still not sufficient, use **VRank** or **PRank** for slight improvement (at a 10^2 to 10^4x computational cost)

In practice, these approximation algorithms remain efficient, even for networks containing millions of elements. The more expensive algorithms (**VRank**, **PRank**) require about a minute to approximate k -median solutions for up to $k = 100$ on a personal computer (Table 6). Thus, one possible approach would create an amalgamate *super-algorithm* which would run the seven methods (**degree**, **degree+**, **VRank**, **PRank**, **core**, **core+**, **H-index**) concurrently and choose the best one for each k -value. A final step could compare this to the expected value (**random**) to give an indication how well the approximation methods have improved the solution.

In summary, the methods presented here do a reasonable job at estimating the k -median problem on complex networks. Despite the challenges of this fundamental problem, these methods provide a reasonable approximation and can be used efficiently to formulate approximations to this important problem, providing researchers with practical tools in studying large-scale complex networks.

References

1. Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., Pandit, V.: Local search heuristic for k -median and facility location problems. In: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, pp. 21–29 (2001)

2. Bae, J., Kim, S.: Identifying and ranking influential spreaders in complex networks by neighborhood coreness. *Physica A: Stat. Mech. Appl.* **395**, 549–559 (2014). <https://doi.org/10.1016/j.physa.2013.10.047>
3. DuBois, C.L.: UCI Network Data Repository. University of California, School of Information and Computer Sciences, Irvine, CA (2008). <http://networkdata.ics.uci.edu>
4. Granovetter, M.: Threshold models of collective behavior. *Am. J. Sociol.* **83**(6), 1420–1443 (1978)
5. Hethcote, H.W.: The mathematics of infectious diseases. *SIAM Rev.* **42**(4), 599–653 (2000)
6. Hirsch, J.E.: An index to quantify an individual’s scientific research output that takes into account the effect of multiple coauthorship. *Scientometrics* **85**(3), 741–754 (2010). <https://doi.org/10.1007/s11192-010-0193-9>
7. Kariv, O., Hakimi, S.L.: An algorithmic approach to network location problems. i: The p-centers. *SIAM J. Appl. Math.* **37**(3), 513–538 (1979)
8. 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)
9. Kitsak, M., et al.: Identification of influential spreaders in complex networks. *Nat. Phys.* **6**(11), 888–893 (2010)
10. Kunegis, J.: Konect - The koblenz network collection. In: Proceedings of the International Web Observatory Workshop, pp. 1343–1350. Boston, MA (2013). <http://konect.uni-koblenz.de/networks>
11. Leskovec, J., Krevl, A.: Snap datasets: stanford large network dataset collection (2014). <http://snap.stanford.edu/data>
12. Lü, L., Zhou, T., Zhang, Q.M., Stanley, H.E.: The h-index of a network node and its relation to degree and coreness. *Nature Communications* **7**(1), 10, 168 (2016). <https://doi.org/10.1038/ncomms10168>
13. Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC’07). San Diego, CA (2007)
14. Newman, M.: *Networks*. Oxford University Press, Oxford (2018)
15. Porter, M.A., Gleeson, J.P.: *Dynamical Systems on Networks*. FADSRT, vol. 4. Springer, Cham (2016). <https://doi.org/10.1007/978-3-319-26641-1>
16. Pozo, R.: NGraph Network Toolkit (2019). <https://math.nist.gov/~RPozo/ngraph>
17. Reese, J.: Solution methods for the p-median problem: an annotated bibliography. *Networks* **48**(3), 125–142 (2006). <https://doi.org/10.1002/net.20128>
18. Zhang, J.X., Chen, D.B., Dong, Q., Zhao, Z.D.: Identifying a set of influential spreaders in complex networks. *Sci. Rep.* **6**(1), 27, 823 (2016). <https://doi.org/10.1038/srep27823>



Score and Rank Semi-monotonicity for Closeness, Betweenness and Harmonic Centrality

Paolo Boldi, Davide D'Ascenzo, Flavio Furi, and Sebastiano Vigna^(✉)

Dipartimento di Informatica, Università degli Studi di Milano, Milan, Italy
{paolo.boldi,sebastiano.vigna}@unimi.it

Abstract. In the study of the behavior of centrality measures with respect to network modifications, *score monotonicity* means that adding an arc increases the centrality score of the target of the arc; *rank monotonicity* means that adding an arc improves the importance of the target of the arc relative to the remaining nodes. It is known [7, 8] that score and rank monotonicity hold in directed graphs for almost all the classical centrality measures. In undirected graphs one expects that the corresponding properties (where both endpoints of the new edge enjoy the increase in score/rank) hold when adding a new edge. However, recent results [6] have shown that in undirected networks this is not true: for many centrality measures, it is possible to find situations where adding an edge reduces the rank of one of its two endpoints. In this paper we introduce a weaker condition for undirected networks, *semi-monotonicity*, in which just one of the endpoints of a new edge is required to enjoy score or rank monotonicity. We show that this condition is satisfied by closeness and betweenness centrality, and that harmonic centrality satisfies it in an even stronger sense.

1 Introduction and Definitions

In this paper we discuss the behavior of centrality measures in undirected networks after the addition of a new edge. In particular, we are interested in the following question: if a new edge is added to a network, does the importance of *at least one* of its two endpoints increase? This question was left open in [6], where the authors proved that for many centrality measures it is possible to find situations where adding an edge reduces the rank of one of its two endpoints. Note that these results are in jarring contrast with the corresponding properties for directed networks, where it is known [7, 8] that score and rank monotonicity hold for almost all centrality measures.

Formally, in this paper we introduce *semi-monotonicity*, a weaker condition than monotonicity for undirected networks in which we require that *at least one* endpoint of the new edge enjoys monotonicity. Score semi-monotonicity, in particular, means that adding a new edge increases the score of at least one of the two endpoints:

Definition 1 (Score semi-monotonicity). *Given an undirected graph G , a centrality c is said to be score semi-monotone on G iff for every pair of non-adjacent vertices x and y we have that*

$$c_G(x) < c_{G'}(x) \quad \text{or} \quad c_G(y) < c_{G'}(y),$$

where G' is the graph obtained adding the edge $x - y$ to G . We say that c is score semi-monotone on a set of graphs iff it is score semi-monotone on all the graphs from the set.

As we already know from the directed case, a score increase does not imply that the rank relations between the two vertices involved in the new edge and the other vertices in the network remain unchanged. For this reason, rank monotonicity was introduced, where we require that every vertex that used to be dominated is still dominated after the addition of the new edge. Formally, the request for at least one of the two endpoints can be expressed as follows:

Definition 2 (Rank semi-monotonicity). *Given an undirected graph G , a centrality c is said to be rank semi-monotone on G iff for every pair of non-adjacent vertices x and y at least one of the following two statements holds:*

– for all vertices $z \neq x, y$:

$$\begin{aligned} c_G(z) < c_G(x) &\Rightarrow c_{G'}(z) < c_{G'}(x) \quad \text{and} \\ c_G(z) = c_G(x) &\Rightarrow c_{G'}(z) \leq c_{G'}(x), \end{aligned}$$

– for all vertices $z \neq x, y$:

$$\begin{aligned} c_G(z) < c_G(y) &\Rightarrow c_{G'}(z) < c_{G'}(y) \quad \text{and} \\ c_G(z) = c_G(y) &\Rightarrow c_{G'}(z) \leq c_{G'}(y), \end{aligned}$$

where G' is the graph obtained adding the edge $x - y$ to G . We say that c is rank semi-monotone on a set of graphs iff it is rank semi-monotone on all the graphs from the set.

In particular, we say that c is rank semi-monotone at x if the first statement holds, and rank semi-monotone at y if the second statement holds (if both statements hold, c is rank monotone).

Definition 3 (Strict rank semi-monotonicity). *Given an undirected graph G , a centrality c is said to be strictly rank semi-monotone on G iff for every pair of non-adjacent vertices x and y at least one of the following two statements holds:*

- for all vertices $z \neq x, y$: $c_G(z) \leq c_G(x) \Rightarrow c_{G'}(z) < c_{G'}(x)$,
- for all vertices $z \neq x, y$: $c_G(z) \leq c_G(y) \Rightarrow c_{G'}(z) < c_{G'}(y)$,

where G' is the graph obtained adding the edge $x - y$ to G . We say that c is strictly rank semi-monotone on a set of graphs iff it is strictly rank semi-monotone on all the graphs from the set.

Again, we say that it is strictly rank semi-monotone at x if the first statement holds, and strictly rank semi-monotone at y if the second statement holds (if both statements hold, c is strictly rank monotone).

In the rest of the paper, we assume that we are given an undirected connected graph G , and two non-adjacent vertices $x, y \in N_G$; G' will be the graph obtained by adding the edge $x - y$ to G . From now on, d_{uv} will refer to the distance (i.e., the length of a shortest path) between u and v in G (i.e., before the $x - y$ addition), and d'_{uv} will refer to the distance in G' instead. In general, we will use the prime symbol to refer to any property or function of G when translated to G' .

2 Distances and Basins

Geometric centrality measures [8] depend only on distances between vertices. In the next two sections we are going to prove new results about the semi-monotonicity of two geometric centrality measures—*closeness centrality* [2,3] and *harmonic centrality* [4,8]. To understand the semi-monotonic behavior of these centrality measures, we introduce the following notion:

Definition 4 (Basin). *Given an undirected graph G and two non-adjacent vertices x and y we define the basin of x (with respect to y) K_{xy} and the basin of y (with respect to x) K_{yx} as*

$$\begin{aligned} K_{xy} &:= \{u \in N_G \mid d_{ux} \leq d_{uy}\} \\ K_{yx} &:= \{u \in N_G \mid d_{uy} \leq d_{ux}\} \end{aligned}$$

That is, the basin of x contains those vertices that are not farther from x than from y : see Fig. 1 for an example. Note that the vertices that are equidistant from x and y are included in both basins, and the score of such vertices cannot change in any geometric centrality when adding the edge $x - y$.

Let us consider a key property:

Definition 5 (Basin dominance). *A centrality c is said to be basin dominant on an undirected graph G iff for every pair of non-adjacent vertices x and y we have that*

$$\begin{aligned} c'(u) - c(u) &\leq c'(x) - c(x) && \text{for every } u \in K_{xy}, u \neq x \\ c'(v) - c(v) &\leq c'(y) - c(y) && \text{for every } v \in K_{yx}, v \neq y. \end{aligned} \tag{1}$$

It is strictly basin dominant iff the same conditions are satisfied, but inequalities (1) hold with the $<$ sign.

Intuitively, basin dominance means that the increase in score of x and y is at least as large as (or larger than, in the strict case) the increase in score of all other nodes in their respective basin.

The following theorems will be used throughout the paper:

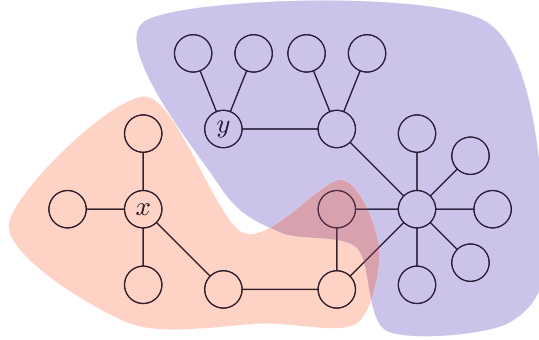


Fig. 1. An undirected graph G , with K_{xy} (the basin of x w.r.t. y) shown in red and K_{yx} (the basin of y w.r.t. x) in blue.

Theorem 1. *If a centrality measure is strictly basin dominant on a graph then it is strictly rank semi-monotone on the same graph.*

Proof. Let c be strictly basin dominant, and let us assume by contradiction that c is not strictly rank semi-monotone. This implies that we should be able to find u, v such that:

$$\begin{cases} c(x) \geq c(v) \\ c(y) \geq c(u) \\ c'(v) \geq c'(x) \\ c'(u) \geq c'(y). \end{cases} \tag{2}$$

As a consequence of (2), $c'(v) - c(v) \geq c'(x) - c(x)$ and $c'(u) - c(u) \geq c'(y) - c(y)$, which by the assumption of strict basin dominance imply $v \notin K_{xy}$ and $u \notin K_{yx}$. Therefore $v \in K_{yx}$ and $u \in K_{xy}$. But then again using strict basin dominance and (2) we have $c'(y) - c(y) > c'(v) - c(v) \geq c'(x) - c(x) > c'(u) - c(u) \geq c'(y) - c(y)$, a contradiction. \square

For the non-strict case, a similar result holds:

Theorem 2. *If a centrality measure is basin dominant on a graph then it is rank semi-monotone on the same graph.*

Proof. Let c be basin dominant, and let us assume by contradiction that c is not rank semi-monotone. This implies that we should be able to find u, v satisfying one of the following four sets of inequalities:

$$\begin{cases} c(x) > c(v) \\ c(y) > c(u) \\ c'(v) \geq c'(x) \\ c'(u) \geq c'(y) \end{cases} \quad \begin{cases} c(x) = c(v) \\ c(y) > c(u) \\ c'(v) > c'(x) \\ c'(u) \geq c'(y) \end{cases} \quad \begin{cases} c(x) > c(v) \\ c(y) = c(u) \\ c'(v) \geq c'(x) \\ c'(u) > c'(y) \end{cases} \quad \begin{cases} c(x) = c(v) \\ c(y) = c(u) \\ c'(v) > c'(x) \\ c'(u) > c'(y). \end{cases} \tag{3}$$

The proof is similar to the strict case. In all sets of inequalities we have $c'(v) - c(v) > c'(x) - c(x)$ and $c'(u) - c(u) > c'(y) - c(y)$, which by the assumption of basin dominance imply $v \notin K_{xy}$ and $u \notin K_{yx}$. Therefore $v \in K_{yx}$ and $u \in K_{xy}$. Using basin dominance and (3), in all cases we have $c'(y) - c(y) \geq c'(v) - c(v) > c'(x) - c(x) \geq c'(u) - c(u) > c'(y) - c(y)$, a contradiction. \square

3 Closeness Centrality

Closeness centrality [2,3] was shown to be score monotone but not rank monotone on connected undirected networks [6]. In the latter paper, it was left open the problem of whether closeness was (in our terminology) rank semi-monotone or not. In the rest of this section, we will solve this open problem by showing that closeness is in fact rank semi-monotone, but not in strict form.

Recall that, given an undirected connected graph $G = (N_G, A_G)$, the *peripherality* of a vertex $u \in N_G$ is the sum of the distances from u to all the other vertices of G :

$$p(u) = \sum_{v \in N_G} d_{uv}.$$

The *closeness centrality* of u is just the reciprocal of its peripherality:

$$c(u) = \frac{1}{p(u)}.$$

Consistently with the previous notation, we will use $p(u)$ for the peripherality of u in G , and $p'(u)$ for the peripherality of u in G' .

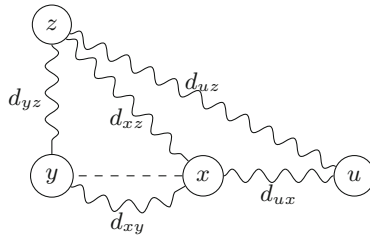


Fig. 2. Path labels represent the distance between the two endpoints. The dashed edge is the $x - y$ edge that we add to G , obtaining G' .

We have that:

Lemma 1. *Closeness centrality is basin dominant on connected undirected graphs.*

Proof. We first show that for every vertex $u \in K_{xy}$ and for every $z \neq u, x$:

$$d_{uz} - d'_{uz} \leq d_{xz} - d'_{xz}. \tag{4}$$

Then, the result follows by adding both sides for all z (as for $z = u$ or $z = x$ the inequality trivializes) and using the fact that closeness is reciprocal to peripherality. To prove (4) we consider two cases (see Fig. 2):

- if $d'_{uz} = d_{uz}$ then the inequality trivially holds (because $d_{xz} \geq d'_{xz}$ always);
- if $d'_{uz} < d_{uz}$ then $d'_{uz} = d_{ux} + 1 + d_{yz} < d_{uz}$ and $d'_{xz} = d_{yz} + 1$. Using $d_{uz} \leq d_{ux} + d_{xz}$ from the triangle inequality we have

$$\begin{aligned} d_{uz} - d'_{uz} &= d_{uz} - (d_{ux} + 1 + d_{yz}) \leq \\ &= (d_{ux} + d_{xz}) - (d_{ux} + 1 + d_{yz}) = d_{xz} - (d_{yz} + 1) = d_{xz} - d'_{xz}. \end{aligned}$$

□

This lemma is the undirected version of Lemma 2 in [7] (provided that in the latter you sum the inequality in the statement over all nodes w): it is interesting to observe that in the directed case the inequality holds *for all* u 's, while here it holds only within the basin.

Applying Lemma 1 and using Theorem 2, we obtain that:

Theorem 3. *Closeness centrality is rank semi-monotone on connected undirected graphs.*

Interestingly, and regardless of their initial score, we can always tell which of the two endpoints of the edge $x - y$ will have smaller peripherality (i.e., higher centrality) in G' . In fact:

Lemma 2. *The following property holds:*

$$p'(x) - p'(y) = |K_{yx}| - |K_{xy}|.$$

Proof. We can write the peripherality of x and y in G' as

$$\begin{aligned} p'(x) &= \sum_{u \in K_{xy}} d_{ux} + \sum_{u \in K_{yx}} (1 + d_{uy}) - \sum_{u \in K_{xy} \cap K_{yx}} (1 + d_{uy}) \\ p'(y) &= \sum_{u \in K_{xy}} (1 + d_{ux}) + \sum_{u \in K_{yx}} d_{uy} - \sum_{u \in K_{xy} \cap K_{yx}} (1 + d_{ux}). \end{aligned}$$

Note that for each $u \in K_{xy} \cap K_{yx}$ we have $d_{ux} = d_{uy}$. Computing the difference between the two expressions gives the result. □

It is not hard to build a graph G where x has a smaller basin than y but a greater score: Lemma 2 tells us that y becomes more central than x in G' , due to having a greater basin.

We conclude this section by showing that:

Theorem 4. *Closeness centrality is not strictly rank semi-monotone on (an infinite family of) connected undirected graphs.*

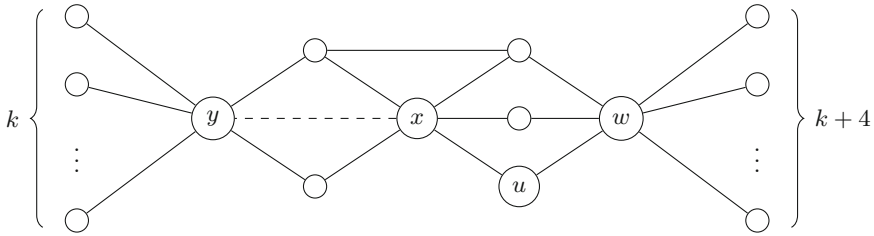


Fig. 3. A counterexample to strict rank semi-monotonicity for closeness centrality. For all $k \geq 10$, u and x have the same score before and after the addition of the edge $x - y$. Moreover, u has the same score of y (or smaller) before the addition, but a higher score after the addition, breaking strict rank semi-monotonicity.

Proof. Consider the graphs in Fig. 3, where $u \in K_{xy}$. This is an infinite family of graphs with a parameter k which controls the sizes of the two stars around vertices w and y . Computing the peripheralities of u , x and y before and after the addition of $x - y$, we obtain

$$\begin{array}{ll}
 p(u) = 2 \cdot (k + 4) + 4 \cdot k + 13 & p'(u) = 2 \cdot (k + 4) + 3 \cdot k + 12 \\
 p(x) = 3 \cdot (k + 4) + 3 \cdot k + 9 & p'(x) = 3 \cdot (k + 4) + 2 \cdot k + 8 \\
 p(y) = 4 \cdot (k + 4) + k + 15 & p'(y) = 4 \cdot (k + 4) + k + 12.
 \end{array}$$

For all $k \geq 10$, we have that

$$p(x) = p(u), \quad p'(x) = p'(u), \quad p(y) \leq p(u), \quad p'(y) > p'(u),$$

showing that closeness is not semi-monotone at y (because y used to be at least as central as u , but it is less central after the addition of the edge) and not strictly rank semi-monotone at x (it is always as central as x , before and after adding the edge). □

4 Harmonic Centrality

Harmonic centrality [4] solves the issue of unreachable vertices in closeness centrality. In particular, if we assume $\infty^{-1} = 0$, we can define it as

$$h(u) = \sum_{v \in N_G \setminus \{u\}} \frac{1}{d_{uv}},$$

so that unreachable vertices have a null impact on the summation and, thus, on the final centrality score of the node. Being a geometric measure, it is trivially score monotone but not rank monotone, as shown in [6], where the same counterexample disproving rank monotonicity for closeness centrality also shows that harmonic centrality fails at satisfying this axiom.

Lemma 3. *Harmonic centrality is strictly basin dominant on connected undirected graphs.*

Proof. We first show that for every vertex $u \in K_{xy}$ and for every $z \neq u, x$:

$$\frac{1}{d'_{uz}} - \frac{1}{d_{uz}} \leq \frac{1}{d'_{xz}} - \frac{1}{d_{xz}}. \quad (5)$$

Then, the result follows by adding both sides for all z . Note that the unique term in $h(u)$ and $h'(u)$ (i.e., when $z = x$) is equal to the unique term in $h(x)$ and $h'(x)$ (i.e., when $z = u$), and they are both equal to 0. Also, we remark that (5) holds with a strict inequality at least in one case, i.e., when $z = y$, since $d'_{xy} < d_{xy}$ always.

We consider two cases:

- if $d'_{uz} = d_{uz}$ then the inequality trivially holds (because $d'_{xz} \leq d_{xz}$ always);
- if $d'_{uz} < d_{uz}$ then $d'_{uz} = d_{ux} + 1 + d_{yz} < d_{uz}$ and $d'_{xz} = d_{yz} + 1$. Using $d_{uz} \leq d_{ux} + d_{xz}$ from the triangle inequality we have

$$\begin{aligned} \frac{1}{d'_{uz}} - \frac{1}{d_{uz}} - \left(\frac{1}{d'_{xz}} - \frac{1}{d_{xz}} \right) &= \frac{1}{d_{ux} + 1 + d_{yz}} - \frac{1}{d_{uz}} - \left(\frac{1}{d_{yz} + 1} - \frac{1}{d_{xz}} \right) \\ &= \frac{(d_{yz} + 1) - (d_{ux} + 1 + d_{yz})}{(d_{ux} + 1 + d_{yz})(d_{yz} + 1)} + \frac{d_{uz} - d_{xz}}{d_{uz}d_{xz}} \\ &= -\frac{d_{ux}}{(d_{ux} + 1 + d_{yz})(d_{yz} + 1)} + \frac{d_{uz} - d_{xz}}{d_{uz}d_{xz}} \\ &< -\frac{d_{ux}}{d_{uz}d_{xz}} + \frac{(d_{ux} + d_{xz}) - d_{xz}}{d_{uz}d_{xz}} = 0, \end{aligned}$$

which proves the inequality. □

Using Lemma 3 and Theorem 1, we obtain that:

Theorem 5. *Harmonic centrality is strictly rank semi-monotone on connected undirected graphs.*

The stronger result we can give for harmonic centrality should be compared to the fact that on strongly connected graphs harmonic centrality is strictly rank monotone, whereas closeness centrality is just rank monotone [6].

5 Betweenness Centrality

Betweenness centrality [1,9] focuses not only on the length of shortest paths, but also on how many of them involve a given node, trying to estimate the amount of flow passing through nodes in a network. Note that betweenness is not a geometric measure.

Formally, if we call σ_{vw} the number of shortest paths between two vertices v and w and $\sigma_{vw}(u)$ the number of such paths passing through u , then we can define the betweenness centrality of a vertex $u \in N_G$ as

$$b(u) = \sum_{\substack{i,j \neq u, \\ \sigma_{ij} > 0}} \frac{\sigma_{ij}(u)}{\sigma_{ij}}.$$

In the following, we let $N_G(u)$ denote the set of neighbors of u in G , and $G[u]$ the subgraph of G induced by $N_G(u)$ (sometimes called the *ego network* of u). As in the previous section, we denote with σ and σ' the number of shortest paths before and after the addition of an edge $x-y$, and with b and b' the betweenness centrality before and after the addition of the edge.

We know from [6] that this centrality measure is neither rank nor score monotone. Nonetheless, we can show that the betweenness centrality of two vertices can never decrease after we link them with a new edge. In fact:

Lemma 4. *The following properties hold:*

$$\frac{\sigma'_{ij}(x)}{\sigma'_{ij}} - \frac{\sigma_{ij}(x)}{\sigma_{ij}} \geq 0 \quad \text{for all } i, j \neq x.$$

As a consequence, $b'(x) \geq b(x)$.

Proof. For all $i, j \in N_G$ such that $i, j \neq x$, let us call p_x ($p_{\bar{x}}$, respectively) the number of shortest paths between i and j passing (not passing, resp.) through x . We have to show that, for each such pair $i, j \neq x$, the following holds:

$$\frac{\sigma'_{ij}(x)}{\sigma'_{ij}} - \frac{\sigma_{ij}(x)}{\sigma_{ij}} = \frac{p'_x}{p'_x + p'_{\bar{x}}} - \frac{p_x}{p_x + p_{\bar{x}}} \geq 0. \tag{6}$$

Summing over all $i, j \neq x$ proves the second part of the statement.

To show that (6) is indeed true we consider two cases:

- $d'_{ij} < d_{ij}$, meaning that in G' all the shortest paths $i \sim j$ pass through the edge $x-y$ (in particular through x). Thus, we obtain:

$$1 - \frac{p_x}{p_x + p_{\bar{x}}} \geq 0,$$

which is clearly true, since the second term of the left side of the inequality is a value between 0 and 1.

- $d'_{ij} = d_{ij}$, which implies that $p'_{\bar{x}} = p_{\bar{x}}$ and $p'_x \geq p_x$. We express p'_x as $p_x + \alpha$ with $\alpha \geq 0$, obtaining:

$$\begin{aligned} \frac{p_x + \alpha}{p_x + \alpha + p_{\bar{x}}} - \frac{p_x}{p_x + p_{\bar{x}}} &= \frac{p_x^2 + p_x p_{\bar{x}} + \alpha p_x + \alpha p_{\bar{x}} - (p_x^2 + \alpha p_x + p_x p_{\bar{x}})}{(p_x + \alpha + p_{\bar{x}})(p_x + p_{\bar{x}})} = \\ &= \frac{\alpha p_{\bar{x}}}{(p_x + \alpha + p_{\bar{x}})(p_x + p_{\bar{x}})} \geq 0, \end{aligned}$$

which is again true, concluding the proof. □

Incidentally, we can always tell if the betweenness centrality of a vertex is zero without actually computing it. In fact,

Lemma 5. *Let G be a connected undirected graph and $u \in N_G$. Then:*

$$b(u) = 0 \iff G[u] \text{ is a clique.}$$

Proof. If $b(u) = 0$ no shortest paths are passing through u : but then any two neighbors of u must be adjacent (or otherwise they would have distance 2, and the path through u has length 2). Conversely, suppose that $G[u]$ is a clique and let $i, j \neq u$. A path from i to j cannot involve u , otherwise it would touch two neighbors of u , say i', j' , and we might shorten it by skipping u and taking the $i' - j'$ edge instead of $i' - u - j'$. \square

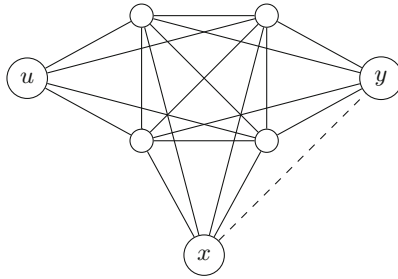


Fig. 4. Simple counterexample for score semi-monotonicity and strict rank semi-monotonicity for betweenness centrality. The dashed edge is the $x - y$ edge that we add to G , obtaining G' . The betweenness score of vertices x , y and u is 0 both in G and G' .

As a consequence, and differently from geometric measures, we can show that

Theorem 6. *Betweenness centrality is not score semi-monotone on (an infinite family of) connected undirected graphs.*

Proof. Consider a graph G such that $G[x]$ and $G[y]$ are cliques and moreover, $N_G(x) = N_G(y)$, that is, x and y have the same neighborhood in G (see Fig. 4 for an example). Then, by Lemma 5 we know that $b(x) = b(y) = 0$. It is easy to observe that $G'[x]$ and $G'[y]$ are still cliques, hence we can use the same lemma and say that $b'(x) = b'(y) = 0$, meaning that the addition of the $x - y$ leaves the score of both the endpoints unchanged. \square

Moreover, we can say that

Theorem 7. *Betweenness centrality is not strictly rank semi-monotone on (an infinite family of) connected undirected graphs.*

Proof. Consider a graph G with three vertices x, y and u adjacent to a clique, but with x, y , and u not adjacent to each other, as in Fig. 4. Then, by Lemma 5 we know that $b(x) = b(y) = b(u) = 0$; if we add the edge $x-y$ to G , obtaining G' , we also have $b'(x) = b'(y) = b'(u) = 0$ by the same lemma. \square

We are now going to show that, somehow unexpectedly, betweenness centrality is rank semi-monotone on connected undirected graphs. In fact, we show that it enjoys the same dominance property of closeness centrality, in spite of being a non-geometric measure:

Lemma 6. *Betweenness centrality is basin dominant on connected undirected graphs.*

Proof. Let us call $\Delta_z = b'(z) - b(z)$ the score difference for a vertex $z \in N_G$, and for every pair of nodes i, j (with $i \neq j$) let also

$$\Delta_z(i, j) = \frac{\sigma'_{ij}(z)}{\sigma'_{ij}} - \frac{\sigma_{ij}(z)}{\sigma_{ij}}.$$

Obviously

$$\Delta_z = \sum_{i, j \neq z} \Delta_z(i, j).$$

We want to show that $\Delta_u \leq \Delta_x$ for every $u \in K_{xy}$. We do this by analyzing the summands $\Delta_u(i, j)$ and $\Delta_x(i, j)$ separately. For reasons of space, the remaining part of the proof appears only in the full version available on arXiv [5]. \square

Hence, applying Theorem 2 with Lemma 6 we have that:

Theorem 8. *Betweenness centrality is rank semi-monotone on connected undirected graphs.*

6 Conclusions and Future Work

Table 1 summarizes the results of this paper along with those of [6–8]. For all the negative results, we have an infinite family of counterexamples (for instance, there are infinitely many graphs on which closeness is shown to be not strictly semi-monotone).

The notion of basin dominance turned out to be the key idea in all proofs of semi-monotonicity. It would be interesting to investigate whether basin dominance applies to other geometric measures, or even other centrality measures based on shortest paths, as in that case one gets immediately rank semi-monotonicity.

Proving or disproving score and (strict) rank semi-monotonicity for other measures (in particular, for the spectral ones) remains an open problem.

Table 1. Summary of the results about monotonicity obtained in this paper (in bold-face) and in [6–8]. All results are about (strongly) connected graphs.

	undirected		directed [7,8]	
	score	rank	score	rank
Closeness	monotone [6]	semi-monotone	monotone	monotone
Harmonic centrality	monotone [6]	strictly semi-mon.	monotone	strictly monotone
Betweenness	not semi-monotone	semi-monotone	not monotone	not monotone

Acknowledgments. This work was supported in part by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU. Davide D’Ascenzo has been financially supported by the Italian National PhD Program in Artificial Intelligence (DM 351 intervento M4C1 - Inv. 4.1 - Ricerca PNRR), funded by EU - NGEU.

References

1. Anthonisse, J.M.: The rush in a directed graph. *J. Comput. Phys.*, 1–10 (1971)
2. Bavelas, A.: A mathematical model for group structures. *Hum. Organ.* **7**, 16–30 (1948)
3. Bavelas, A.: Communication patterns in task-oriented groups. *J. Acoust. Soc. Am.* **22**, 725–730 (1950)
4. Beauchamp, M.A.: An improved index of centrality. *Behav. Sci.* **10**(2), 161–163 (1965)
5. Boldi, P., D’Ascenzo, D., Furia, F., Vigna, S.: Score and rank semi-monotonicity for closeness, betweenness and harmonic centrality (2023)
6. Boldi, P., Furia, F., Vigna, S.: Monotonicity in undirected networks. *Netw. Sci.*, 1–23 (2023). <https://doi.org/10.1017/nws.2022.42>
7. Boldi, P., Luongo, A., Vigna, S.: Rank monotonicity in centrality measures. *Netw. Sci.* **5**(4), 529–550 (2017). <https://doi.org/10.1017/nws.2017.21>
8. Boldi, P., Vigna, S.: Axioms for centrality. *Internet Math.* **10**, 222–262 (2014)
9. Freeman, L.C.: Centrality in social networks conceptual clarification. *Soc. Netw.* **1**, 215–239 (1978)



Non Parametric Differential Network Analysis for Biological Data

Pietro Hiram Guzzi¹(✉), Arkaprava Roy², Francesca Cortese¹,
and Pierangelo Veltri³

¹ Magna Graecia University, Catanzaro, Italy
hguzzi@unicz.it

² University of Florida, Gainesville, USA

³ DIMES, University of Calabria, Cosenza, Italy
pierangelo.veltri@unical.it

Abstract. Rewiring of molecular interactions under different conditions causes different phenotypic responses. Differential Network Analysis (also indicated as DNA) aims to investigate the rewiring of gene and protein networks. DNA algorithms combine statistical learning and graph theory to explore the changes in the interaction patterns starting from experimental observation. Despite there exist many methods to model rewiring in networks, we propose to use age and gender factors to guide rewiring algorithms. We present a novel differential network analysis method that consider the differential expression of genes by means of sex and gender attributes. We hypothesise that the expression of genes may be represented by using a non-gaussian process. We quantify changes in non-parametric correlations between gene pairs and changes in expression levels for individual genes. We apply our method to identify the differential networks between males and females in public expression datasets related to mellitus diabetes in liver tissue. Results show that this method can find biologically relevant differential networks.

Keywords: Non parametric Differential Network Analysis · Algorithms · Biological Networks

1 Introduction

The always more performing high-throughput technologies for studying genes, proteins and non-coding RNA allowed us to identify and study many associations among changes in their abundance and diseases [2, 7, 8, 11, 25]. The availability of data has also motivated the introduction of novel methods of analysis based on a systemic perspective by using network science. Networks have been used to represent the set of associations among biological molecules in a given state starting from experimental observation [10]. An interesting application of network modelling is the possibility of gathering information by comparing biological networks associated with different conditions (e.g. healthy vs diseases). Differential Network Analysis (DNA) has been introduced to model differences

between two conditions represented as two distinct networks in a single network (differential network) which model the differences [14]. DNA methods have been used to compare experimental configurations (e.g. two different drugs) or phenotypes. Recently, many observations have evidenced that the age and sex of patients may have a role in causing different responses to drugs, outcomes of diseases and incidence of comorbidities for many complex chronic diseases [12, 26]. Experimental observations evidenced that both the incidence and progression of some diseases have remarkable differences considering sex and age as factors. For instance, patients affected by diabetes are more likely to develop comorbidities as they grow older, as well as studies about mortality caused by COVID-19 pandemic, showed higher number of deaths in older males wrt female [4, 18]. Consequently, the necessity of defining novel algorithms to identify motivations or possible candidates that causes such differences at the molecular level and depending on age or genders, arises.

DNA algorithms aim to identify changes in measures of association in terms of network rewiring, which can distinguish different biological conditions $\mathcal{C}_\infty, \mathcal{C}_\epsilon$. Let $\mathcal{C}_\infty, \mathcal{C}_\epsilon$ be two different biological conditions in two different networks representing molecular interactions, DNA algorithms aim to identify changes in network rewiring (and differences) among the two networks that may be possible candidates associated to changes among C_1 and C_2 . Given two gene expression datasets corresponding to two experimental conditions, DNA algorithms first derive two networks $\mathcal{N}_\infty, \mathcal{N}_\epsilon$ corresponding to the examined condition. Each network usually has a node for each gene of the dataset, while a weighted edge between nodes represents the association or the casual dependency among them and the weight represents the strength of the association. Finally, a single network \mathcal{N}_Γ describing the changes is calculated. The final network represents the rewiring of association in condition, and it has been used in the past to study changes associated with pathological conditions [3, 12]. Given two datasets corresponding to gene expressions related to two different biological experiments, DNA is based on the following steps. Firstly two networks $\mathcal{N}_\infty, \mathcal{N}_\epsilon$ corresponding to the examined conditions are defined. Each node in a network corresponds to a gene in the dataset, and weighted edges between nodes represent the association or the casual dependency among genes, while the weight on an edge represents the strength of the associations among genes. Then, DNA algorithmic implementations compute a new single network \mathcal{N}_Γ describing the changes (i.e. the different among networks \mathcal{N}_∞ and \mathcal{N}_ϵ). The resulting network represents the rewired associations that can be used to study biological triggers causing rewiring differences. This may be used to study variations associated with pathological conditions as reported in [3, 12].

Many existing methods are based on the hypothesis that experimental observation such as gene expression values come from parametric distribution, e.g. normal, paranormal, binomial or Poisson distribution [7, 13, 17, 19]. NGS data, different to microarray technology, are similar to count data, thus parametric distribution hypothesis sometimes does not hold. Consequently, there is a need for the introduction of non-parametric methods for DNA analysis.

We here propose a novel DNA method to identify differential edges among two networks and integrate differential expressions between nodes (i.e. genes) also using gender-based differences [6]. Moreover, gene expression level is statistically predicted by using multivariate count data, and the conditional dependence graph is built by using pairwise Markov random fields [22]. Differently to existing methods where gene expression values are obtained by using parametric distribution (such as normal, paranormal, binomial or Poisson) [7, 13, 17, 19], DNA analysis requires non-parametric methods. In a nutshell, the proposed DNA algorithm works as follows. We build two graphs for the two tested conditions; then, we derive the final graph from the previous two. Finally, we prune the resulting graph by admitting only edges incident to at least one differential expressed gene. The proposed DNA has been used with genes dataset related to patients affected by diabetes and we use results to identify the differential networks also using gender attributes (i.e. male and female patients).

The paper is structured as follows: Sect. 2 discusses state-of-the-art methods; Sect. 3 presents the proposed approach; Sect. 4 discusses the results of some experiments; finally Sect. 5 concludes the paper.

2 Related Work

DNA is largely used to identify differentially expressed genes between groups of samples, and thus useful to compare genes from patients with a particular disease compared to healthy subjects. In molecular biology and bioinformatics it is used for identifying genes that are differentially expressed between groups of samples, such as those from patients with a particular disease compared to healthy individuals.

DNA algorithms aim to identify changes in the network structures between two states, or conditions [24]. In biology, DNA algorithms have been used to identify changes between the healthy and diseased status of the same biological system [10]. There exist some different formulations of the problem, we here focus on networks with the same node sets and different edge sets. Formally, given two different conditions \mathcal{C}_1 , and \mathcal{C}_2 , represented by means of two graphs $G_1(V, E_1)$ and $G_2(V, E_2)$, DNA aims to identify changes between them.

When dealing with biological systems it should be noted, that nodes are directly measurable, while edges among them should be derived from a set of observations over time. For instance, when considering gene networks derived from microarray experiments, nodes are fixed while edges should be inferred from the observations by means of *statistical graphical models* [9, 15, 23]. In a statistical graphical model, we use a graph $G = (V, E)$, and each node $v \in V$ is associated with a set of m random variables X_1, \dots, X_m representing quantitative measurements of v , and edges are inferred from X_1, \dots, X_m . We focus on **undirected** graphs. In such models differential associations are measured by analysing the difference of partial correlations between experimental data of two conditions. Changes are measured by means of specific statistics test defined to measure the modification among correlation between entities. Moreover, the

changes in gene expression levels are quantified by using the classical Student's t-test statistics [28]. Then, the two test statistics are integrated into a single optimisation model which aims to evidence the hierarchical structures of networks. Nevertheless, some hypotheses of the previous models (e.g. gaussian distribution of data) are not valid in all the experimental conditions, therefore non-parametric methods have been introduced. These methods are in general computationally efficient and often easier to implement and the results can be more interpretable. The main limitation of these methods is that they require the data to adhere to specific distributional assumptions, and if these assumptions are violated, the results can be biased or incorrect.

Some works considered a nonparanormal distribution of data (or Gaussian copula) instead of normality or multivariate normality of data [1] and they used a rank-based correlation matrix, such as Spearman correlation or Kendal's τ . Since the nonparanormal model presents some restrictions on the nature of data, some conditionally-specified additive graphical models have been proposed such as graphical random forest and kernel-based estimators [24].

In particular such models have been used for brain data and counts based data, such as sequencing. To overcome the time limitations of the non-parametric methods, efficient Bayesian models have been proposed [24]. Such methods are based on the calculation of probabilities of the edges among data by inferring their likelihood. Some of the proposed methods used different heuristics to infer such probabilities which are hard to derive from data, such as in [22]. We here selected this last method which outperforms the other state-of-the-art methods.

Non-parametric methods make fewer assumptions about the underlying data distribution and they are based on data-driven techniques to assess the differences in network connectivity between conditions. These methods are more flexible and robust, as they do not assume a specific data distribution and can handle complex and non-linear relationships between nodes in the network. On the other hand, they can be computationally intensive and less interpretable.

Choosing between parametric and non-parametric differential network analysis depends on the nature of the data, the underlying assumptions, and the research question. Researchers often perform sensitivity analyses and cross-validate their results to ensure the robustness and reliability of the findings. Additionally, combining information from both approaches may provide a more comprehensive understanding of the differential network structure.

3 The Proposed Pipeline

This section explains the method we designed and implemented as depicted in Fig. 1. The method starts by gathering expression data grouped by tissues and for each tissue we build two different datasets filtered by using gender of the individual they belongs to.

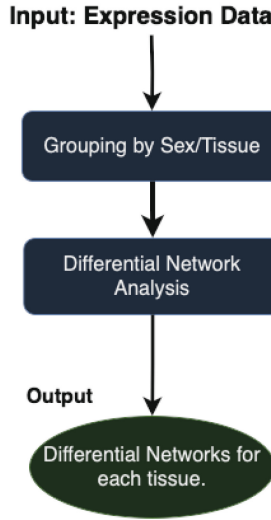


Fig. 1. Figure depicts the main steps of the experiments we performed.

3.1 Non Parametric Differential Network Analysis

Let us suppose that two biological conditions \mathcal{C}_1 \mathcal{C}_2 , have been investigated by means of gene expression analysis giving two different expression datasets encoded in two matrices $N \times M$ (N samples, M genes) X^1, X^2 . Each row of X^j stores the expression values of M genes of the i sample. Therefore $X_{i,j}^c, j$ ($c = 1,2, i = 1..n, j = 1..m$) denotes a pair of $N \times M$ matrices.

Many approaches suppose that gene expression datasets are samples from two multivariate normal distributions. We here do not hold this hypothesis, so we hypothesise that two gene expression datasets are samples from non-parametric distributions.

Let $\mathcal{P}_{1,2}$ ($\|P\| = n \times n$) be two matrices representing the relation among nodes. Both matrices represent the conditional independence among nodes [22].

We may define the differential matrix between two conditions as the difference as \mathcal{P}_1 and \mathcal{P}_2 .

Following pair-wise MRF [5,27], we consider the following joint probability mass function for P -dimensional count-valued data X as in [22],

$$\Pr(X_1, \dots, X_P) \propto \exp \left(\sum_{j=1}^P [\alpha_j X_j - \log(X_j!)] - \sum_{\ell=2}^P \sum_{j < \ell} \beta_{j\ell} F(X_j) F(X_\ell) \right),$$

where $F(\cdot)$ is a monotone increasing bounded function with support $[0, \infty)$. We let $F(\cdot) = (\tan^{-1}(\cdot))^\theta$ for some positive $\theta \in \mathbb{R}^+$ to define a flexible class of monotone increasing bounded functions. The exponent θ provides additional flexibility, including impacting the range of $F(X)$, $(0, (\frac{\pi}{2})^\theta)$. The parameter θ can be estimated along with the other parameters, including the baseline parameters α

controlling the marginal count distributions and the coefficients β_{jl} controlling the graphical dependence structure. For simplicity and interpretability, we estimate θ to minimize the difference in covariance between $F(X)$ and X following [22]. For detailed descriptions of the method, readers are encouraged to check [22].

If $\beta_{j\ell} = 0$, we have X_j and X_ℓ to be conditionally independent i.e. $P(X_j, X_\ell | X_{-(j,\ell)}) = P(X_j | X_{-(j,\ell)})P(X_\ell | X_{-(j,\ell)})$, where $X_{-(j,\ell)}$ stands for all the variables excluding X_j and X_ℓ . Our estimated graphical relation would rely on $\hat{\beta}_{j\ell}$'s, and thus, our model is a probabilistic model that encodes the conditional independence structure in a graph.

Consequently, we define a differential network as the difference $\beta_{j,k}^{(1)} - \beta_{j,k}^{(2)}$ for each edge (j,k) where $\beta_{j,k}^{(1)}$ and $\beta_{j,k}^{(2)}$ are estimated coefficients under two conditions 1 and 2. From the MCMC samples, we can get the posterior estimates of these differences as $\hat{\beta}_{j,k}^{(1)} - \hat{\beta}_{j,k}^{(2)}$. Alternatively, we may be able to compute other posterior summaries such as $P(|\beta_{j,k}^{(1)} - \beta_{j,k}^{(2)}| > c | D_1, D_2)$, which is the posterior probability that $|\beta_{j,k}^{(1)} - \beta_{j,k}^{(2)}|$ is greater than some pre-specified cutoff given the two datasets, denoted as D_1 and D_2 .

4 Experimental Results

Figure 1 depicts the main steps of our experiments. We start by considering GTEx gene-expression data [16]. Genotype-Tissue Expression (GTEx) data portal [16], which is a publicly available resource containing expression data of patients integrated with information related to the tissue of provenance, sex and age (grouped into six classes). The current version of the GTEx database accessed on February 01st stores 17382 samples of 54 tissues of 948 donors, see at <https://gtexportal.org/home/tissueSummaryPage> [18, 20, 21]. We downloaded data and integrated it with information related to sex and age with an ad-hoc realised script. Then for each tissue, we split data into male and female samples. For each tissue, we randomly selected the same number of samples.

We used Conga [22] R package to derive non-parametric differential networks. Conga receives as input two datasets corresponding to the different experimental conditions. We ran conga using default parameters. We here show results for genes related to diabetes expressed in the liver tissue in Fig. 2.

To explain this network's biological and medical relevance, we perform a Gene Ontology analysis using the String Database as depicted in Fig. 3.

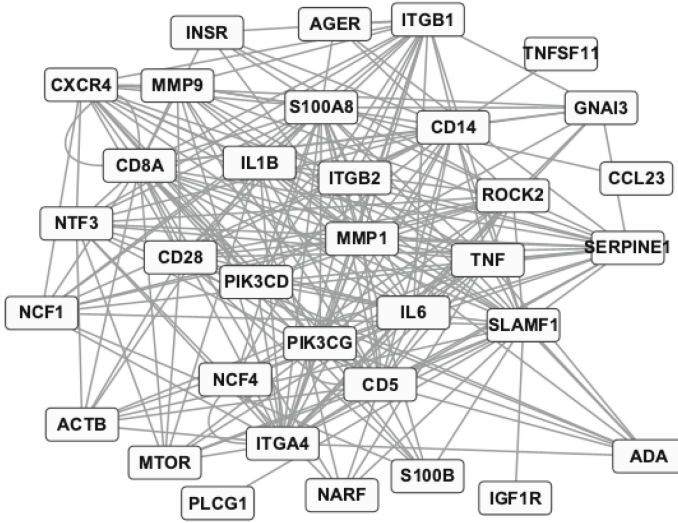
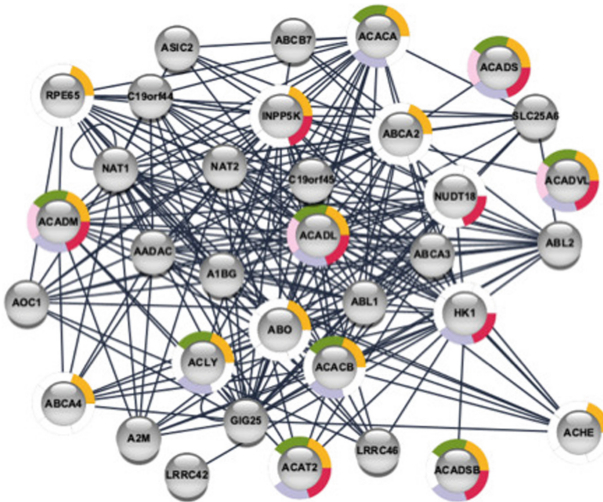


Fig. 2. Figure represents a differential network between males and females in liver tissue



category	chart color	term name	description	FDR value
GO Biological Process	Yellow	GO:0044255	Cellular lipid metabolic process	1,83E-7
GO Biological Process	Green	GO:0006631	Fatty acid metabolic process	1,16E-5
GO Biological Process	Pink	GO:0033539	Fatty acid beta-oxidation using acyl-CoA de...	2,4E-5
GO Biological Process	Purple	GO:0032787	Monocarboxylic acid metabolic process	3,3E-5
GO Biological Process	Red	GO:0044282	Small molecule catabolic process	7,94E-5

Fig. 3. Figure depicts the main enriched function of the obtained network. We used the String Database and we selected Gene Ontology Biological Process. All the functions are enriched with a p-value after false discovery rate correction value less than 0.01

5 Conclusion

The results presented in the article demonstrate the effectiveness of the proposed non-parametric approach for Differential Network Analysis (DNA) in uncovering meaningful biological insights from gene expression data. The analysis focuses on gender-based differences in liver tissue using the Genotype-Tissue Expression (GTEx) database.

The differential network provides a visual representation of how gene interactions change between males and females, offering insights into gender-specific molecular relationships. To understand the biological relevance of the identified differential associations, we performed a Gene Ontology analysis using the STRING Database. The enriched Gene Ontology terms shed light on the biological processes that may be influenced by gender-specific gene interactions in liver tissue.

The results highlight specific genes and interactions that exhibit gender-based differences. This information is valuable for understanding why certain diseases or conditions might affect males and females differently. By identifying gender-specific molecular associations, researchers can potentially uncover molecular mechanisms underlying sex-related disparities in disease susceptibility, progression, or response to treatments.

The success of the approach in identifying known gender-specific molecular differences validates the effectiveness of the method. If the identified differential network aligns with existing knowledge about gender-related gene interactions or processes, it strengthens the credibility of the approach and its ability to reveal biologically relevant insights.

While the results provided promising insights, the analysis may also be influenced by factors like data quality, sample size, and the choice of statistical parameters. Additionally, the results might be specific to the dataset used (GTEx), and their generalizability to other datasets or tissues should be considered.

Understanding gender-specific molecular differences has implications for personalized medicine. The identified genes and interactions could potentially serve as biomarkers for predicting disease risk, prognosis, or treatment response based on an individual's gender. This could lead to more tailored and effective medical interventions.

In summary, the results presented in the article showcase the potential of the non-parametric Differential Network Analysis method to uncover gender-based differences in gene interactions. The identified differential network and enriched Gene Ontology terms provide insights into the molecular underpinnings of gender-related disparities in liver tissue. This approach has broader implications for understanding sex-specific responses to diseases and treatments, advancing the field of personalized medicine.

References

1. Allen, G.I., Liu, Z.: A local poisson graphical model for inferring networks from sequencing data. *IEEE Trans. Nanobiosci.* **12**(3), 189–198 (2013)

2. Buccitelli, C., Selbach, M.: mRNAs, proteins and the emerging principles of gene expression control. *Nat. Rev. Genet.* **21**(10), 630–644 (2020)
3. Cannataro, M., Guzzi, P.H., Mazza, T., Tradigo, G., Veltri, P.: Using ontologies for preprocessing and mining spectra data on the grid. *Futur. Gener. Comput. Syst.* **23**(1), 55–60 (2007)
4. Cannistraci, C.V., Valsecchi, M.G., Capua, I.: Age-sex population adjusted analysis of disease severity in epidemics as a tool to devise public health policies for COVID-19. *Sci. Rep.* **11**(1), 1–8 (2021)
5. Chen, S., Witten, D.M., Shojaie, A.: Selection and estimation for mixed graphical models. *Biometrika* **102**(1), 47–64 (2014)
6. Chiarella, G., et al.: Vestibular disorders in euthyroid patients with hashimoto's thyroiditis: role of thyroid autoimmunity. *Clin. Endocrinol.* **81**(4), 600–605 (2014)
7. Cho, Y.R., Mina, M., Lu, Y., Kwon, N., Guzzi, P.H.: M-finder: uncovering functionally associated proteins from interactome data integrated with go annotations. *Proteome Sci.* **11**(1), 1–12 (2013)
8. Cookson, W., Liang, L., Abecasis, G., Moffatt, M., Lathrop, M.: Mapping complex disease traits with global gene expression. *Nat. Rev. Genet.* **10**(3), 184–194 (2009)
9. Galicia, J.C., Guzzi, P.H., Giorgi, F.M., Khan, A.A.: Predicting the response of the dental pulp to SARS-CoV2 infection: a transcriptome-wide effect cross-analysis. *Genes Immun.* **21**(5), 360–363 (2020)
10. Grimes, T., Potter, S.S., Datta, S.: Integrating gene regulatory pathways into differential network analysis of gene expression data. *Sci. Rep.* **9**(1), 1–12 (2019)
11. Gu, S., Jiang, M., Guzzi, P.H., Milenković, T.: Modeling multi-scale data via a network of networks. *Bioinformatics* **38**(9), 2544–2553 (2022)
12. Guzzi, P.H., et al.: Analysis of age-dependent gene-expression in human tissues for studying diabetes comorbidities. *Sci. Rep.* **13**(1), 10372 (2023)
13. Guzzi, P.H., et al.: Differential network analysis between sex of the genes related to comorbidities of type 2 mellitus diabetes. *Appl. Network Sci.* **8**(1), 1–16 (2023)
14. Ideker, T., Krogan, N.J.: Differential network biology. *Mol. Syst. Biol.* **8**(1), 565 (2012)
15. Lauritzen, S.L.: *Graphical Models*, vol. 17. Clarendon Press (1996)
16. Lonsdale, J., et al.: The genotype-tissue expression (GTEx) project. *Nat. Genet.* **45**(6), 580–585 (2013)
17. Mangoni, M., et al.: Investigating mitochondrial gene expression patterns in drosophila melanogaster using network analysis to understand aging mechanisms. *Appl. Sci.* **13**(12), 7342 (2023)
18. Mercatelli, D., Pedace, E., Veltri, P., Giorgi, F.M., Guzzi, P.H.: Exploiting the molecular basis of age and gender differences in outcomes of SARS-CoV-2 infections. *Comput. Struct. Biotechnol. J.* **19**, 4092–4100 (2021)
19. Milano, M., et al.: An extensive assessment of network alignment algorithms for comparison of brain connectomes. *BMC Bioinf.* **18**, 31–45 (2017)
20. Ortuso, F., Mercatelli, D., Guzzi, P.H., Giorgi, F.M.: Structural genetics of circulating variants affecting the SARS-CoV-2 spike/human ace2 complex. *J. Biomol. Struct. Dyn.* **40**(14), 6545–6555 (2021)
21. Pressler, M.P., Horvath, A., Entcheva, E.: Sex-dependent transcription of cardiac electrophysiology and links to acetylation modifiers based on the GTEx database. *Front. Cardiovasc. Med.* **9**, 941890 (2022)
22. Roy, A., Dunson, D.B.: Nonparametric graphical model for counts. *J. Mach. Learn. Res.* **21**(1), 9353–9373 (2020)

23. Roy, S., Manners, H.N., Jha, M., Guzzi, P.H., Kalita, J.K.: Soft computing approaches to extract biologically significant gene network modules. In: Purohit, H.J., Kalia, V.C., More, R.P. (eds.) *Soft Computing for Biological Systems*, pp. 23–37. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-7455-4_3
24. Shojaie, A.: Differential network analysis: a statistical perspective. *Wiley Interdiscip. Rev. Comput. Stat.* **13**(2), e1508 (2021)
25. Stark, R., Grzelak, M., Hadfield, J.: RNA sequencing: the teenage years. *Nat. Rev. Genet.* **20**(11), 631–656 (2019)
26. Succurro, E., et al.: Sex-specific differences in prevalence of nonalcoholic fatty liver disease in subjects with prediabetes and type 2 diabetes. *Diabetes Res. Clin. Pract.* **190**, 110027 (2022)
27. Wainwright, M.J., Lafferty, J.D., Ravikumar, P.K.: High-dimensional graphical model selection using ℓ_1 regularized logistic regression. In: *Advances in Neural Information Processing Systems*, pp. 1465–1472 (2007)
28. Zimmerman, D.W.: Comparative power of student T test and Mann-Whitney U test for unequal sample sizes and variances. *J. Exp. Educ.* **55**(3), 171–174 (1987)



Bowlership: Examining the Existence of Bowler Synergies in Cricket

Praharsh Nanavati¹ and Amit Anil Nanavati²(✉)

¹ Department of Data Science and Engineering, Indian Institute of Science Education and Research (IISER), Bhopal, India

praharsh19@iiserb.ac.in

² School of Engineering and Applied Science, Ahmedabad University, Ahmedabad, India

amit.nanavati@ahduni.edu.in

Abstract. Player synergies are a salient feature of team sports. In the team game of cricket, player synergies may be reflected in batting partnerships. Batting partnerships have been analysed extensively. In this paper, we introduce and precisely define bowling partnerships. We explain their importance, and analyse ball-by-ball data from three formats of the game: 2,034 one-day international matches, 634 Test matches and 1,432 Twenty-20 international matches, in order to find such bowling partnerships (“bowlerships”). We find that bowlerships exist. We construct bowlership networks based on these pairwise synergies. We assert that these bowlership networks can be analysed for team selection before a match, and making bowling changes during the match. We present Algorithm *bowler-select* that selects a team based on *bowlerships*.

Keywords: Data Mining · Cricket Analytics · Network Science

1 Introduction

Team sports are all about player synergies; deciding when to let your partner take the shot in Tennis doubles or anticipating the next pass in Football. This is why often the best single’s champions are not necessarily the best doubles’ champions in Tennis, and the team with more star players does not necessarily win the Football match.

Cricket is a team sport played in most Commonwealth countries. There are various formats of the game varying from five-day long matches to 3-hour matches. Batsmen and Bowlers in Cricket are traditionally ranked according to their batting and bowling averages respectively [6]. Batsmen bat in pairs, and the pair is known as a partnership. A partnership continues batting until one of the batsmen is dismissed. It has long been believed that synergies exist between effective batting partners. Some batting pairs are part of legend. For example, Matthew Hayden and Justin Langer of Australia and Desmond Haynes and Gordon Greenidge of the West Indies are mentioned by Wisden [14]. In their

paper [10], the authors investigate the importance of batting partnerships in Test and one-day cricket with respect to improved performance. Based on their statistical analyses, the authors conclude that synergies in opening partnerships may be considered a sporting myth.

In this paper, we investigate bowler pairs. We introduce and precisely define a bowling partnership. Bowlers bowl in pairs from opposite ends of the field alternately. It is often the case that a pair of bowlers bowl several consecutive overs alternately. We define such pairs of bowlers a *bowling pair*. Are some bowling pairs more effective than others? Are they effective in terms of saving runs or taking wickets?

Wisden also lists famous bowling partnerships [15]. The bowling partnership list contains pairs of bowlers who bowled in the same match but *not necessarily together*. Bowlers have two goals: saving runs and taking wickets. Commentators sometimes anecdotally observe that when one of the bowlers is economical (saving runs), the other bowler is targeted by the batsmen leading them to make errors and give away wickets. This raises the question whether two bowlers are more effective as a pair with each other, in saving runs or taking wickets or both.

Definition 1. *A bowler's economy rate is the average number of runs he/she has conceded per over bowled. The lower the economy rate is, the better the bowler is performing.*

Definition 2. *A bowler's hitrate is defined as the average number of wickets he/she has taken per over bowled. The higher the hitrate, the better the bowler is performing.*

Deciding which bowler should bowl from which end and when is a crucial decision to be made by the captain. As with other things, bowler (team) selection (before the match) and deciding bowling changes (during the match) takes skill and experience. Such decisions depend upon a number of complex factors including pitch conditions and assessing the competitor team's strengths and weaknesses. If effective bowling partnerships do in fact exist, then this could inform the strategies for team selection as well as bowling changes.

Bowlers are typically assessed on two aspects – the wickets they take (the more the better) and the runs they concede (the less the better). While the former matters the most in Test cricket (5-day matches), in the limited overs versions (one-day internationals and T20 matches) the latter matters too. The Bowling strike rate is defined for a bowler as the average number of balls bowled per wicket taken [12], and the economy rate is the average number of runs they have conceded per over bowled [11]. Therefore a bowling pair can be deemed effective (synergistic) if the pair together saves more runs and/or takes more wickets. This could happen in three ways: (i) both the bowlers have improved economies, (ii) both the bowlers take more wickets, (iii) one of them has an improved economy while the other gets more wickets.

Our analysis involves a comparison of the performance of a bowler together with his partners to investigate if the performance of the bowler with a given partner is statistically superior to his performance with the rest. The findings of such analysis can inform bowler selection and bowling changes.

We analysed data across all formats of the game — Test Cricket, ODIs and T20Is. In Sect. 2, we define and explore these bowling partnerships, and investigate if there are any statistically significant partnerships. In Sect. 3, we present results from the 3 formats of the game and compare them. Some discussion and concluding remarks are then provided in Sect. 4.

2 Methodology

We acquired ball-by-ball data of Men’s Test Matches, One-day internationals and T20 Internationals in separate files in the YAML format from cricsheet.org [2]. These include metadata like venue, date, participating teams, toss details including striker, non-striker, bowler, runs scored on each ball, wickets taken (if any), type of dismissal, extras information and outcome of the game.

We analysed 2,034 ODIs, 634 Test matches and 1,432 T20Is. There were a total of 1148 ODI bowlers, 495 test bowlers and 1518 T20I bowlers. In general, the falling of a wicket is a rare event. This is quite unlike scoring runs. Roughly, the falling of a wicket occurs once in 3, 9 and 12 overs respectively in T20Is, ODIs and Tests.

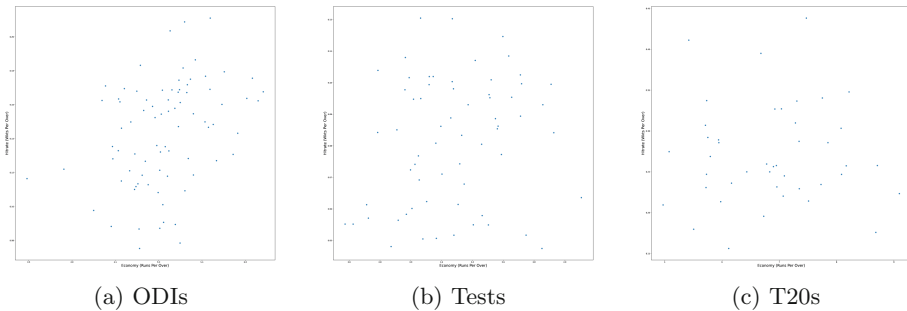


Fig. 1. Hitrate vs. Economy: This scatter plot show the economy on the X-axis along with their respective hitrates on the Y-axis for all the bowlers. Bowlers to the top left are good at both traits – conceding fewer runs and taking more wickets whereas bowlers to the bottom right are conceding the most runs and taking the fewest wickets per over.

It is uncommon to concede too few or too many runs in an over. Hence, we expected the distribution of runs conceded by all bowlers across all overs to be normal. Figure 1 shows the Economy v Hitrate plots indicating the trends of all the bowlers in each format. To measure a bowler’s performance, Croucher [3] defines the bowling index as:

$$\text{Bowling Index} = \text{Bowling average} \times \text{Bowling Strikerate}$$

where, bowling average is the number of runs conceded by a bowler per wicket taken and bowling strikerate is the average number of balls bowled for every

Table 1. The results of 3 statistical tests of Normality. The test of normality failed several data points.

Normality Test	ODIs		Tests		T20Is	
	Fail	Pass	Fail	Pass	Fail	Pass
Chi-square	578 (60%)	393 (40%)	295 (70%)	129 (30%)	232 (28%)	609 (72%)
Shapiro-Wilk	674 (71%)	297 (29%)	360 (85%)	64 (15%)	316 (38%)	525 (62%)
Anderson-Darling	785 (81%)	186 (19%)	390 (92%)	34 (8%)	463 (55%)	378 (45%)

wicket taken. A bowler is successful if he takes wickets and/or gives away few runs and hence the Economy and the Hitrate are separate metrics and are analysed separately as well. For each bowler, we plotted the runs conceded per over and checked if these distributions were normal. Table 1 summarises the results. We conducted three normality tests: Chi-square test [4], Shapiro-Wilk test [7] and Anderson-Darling test [8]. Except in the case of T20Is for the former two tests, more bowlers fail than pass the test. Therefore, for most bowlers, the distribution is not normal. Since the falling of a wicket is a rare event, the distribution of wickets per over for each bowler is not normal either.

Definition 3. A pair of bowlers constitute a *bowling pair* at individual threshold t_i and pairing-threshold t_p iff: (a) each bowler has bowled at least t_i overs in his career, and (b) together they bowl at least t_p consecutive overs alternately over all the matches.

In order to exclude trivial bowling pairs, we set the following conditions:

- T1.** (for t_i) The individual bowlers in a bowling pair should have bowled at least 300 overs (in Tests), 300 overs (in ODIs) and 80 overs (in T20Is) throughout the span of their careers.
- T2.** (for t_p) In order for a pair of bowlers to be considered a bowling pair, we set the pairing-threshold – the number of consecutive overs that they should have bowled alternately – to 60 (in Tests), 60 (in ODIs) and 16 (in T20Is).

Based on condition T1: 64 Test bowlers (out of 495) and 80 ODI bowlers (out of 1148) have bowled at least 300 overs each. We found 45 T20I bowlers (out of 1518) have bowled at least 80 overs. Further filtering based on the additional condition T2, we got: 81 Test bowler pairs, 41 ODI bowler pairs and 18 T20I bowler pairs who have together bowled at least 60, 60 and 16 overs respectively.

While these numbers appear to have been arbitrarily chosen, the basic rationale is to ensure that an individual bowler has bowled enough overs and the choice of the pairing-threshold is such that it allows a bowler to have a few potential bowling pairs (5 with the above values). The analysis detailed in this section could easily be conducted by choosing other values to obtain corresponding results. We discuss this further in Sect. 4. We need to compare

the performance of an individual bowler with the bowling pairs that the bowler is a part of. Since the individual distributions are not normal, we use the non-parametric Mann-Whitney U test [13], to do the comparisons.

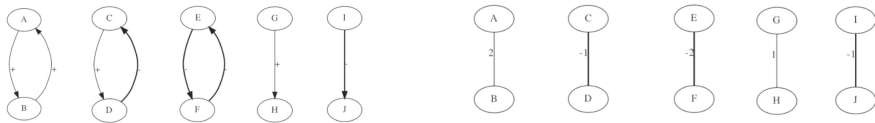
2.1 Mann-Whitney U Test

We consider the set of all the overs that bowler A bowls with a particular partner B (bowlership set) and the set of all overs bowled by the bowler A (individual set). Since not all bowlers' economy rates or wickets per over aren't normal, we use the Mann-Whitney U test to compare an individual bowler's performance with a paired performance of the same bowler with a partner. The number of runs conceded in each over is considered for this test. We conduct three tests:

1. "greater" | H_0 : Individual Economy better than or same as the Bowlership Economy.
2. "two-sided" | H_0 : Individual Economy is same as the Bowlership Economy.
3. "less" | H_0 : Individual Economy worse or same as the Bowlership Economy.

If the first two tests fail, then the two null hypotheses can be rejected and we can conclude that the bowlership pair performs better than the individual. In this case, we say that a *positive bowlership* exists from A to B . This relationship is not symmetric. Bowler A may bowl better with a Bowler B , while the opposite need not be true. If the last two tests fail, then the two null hypotheses can be rejected and we can conclude that the bowlership pair performs worse than the individual. In this case, we say that a *negative bowlership* exists from A to B . We conduct similar tests for Bowlership Hitrates.

2.2 Bowlership Networks



(a) Examples of directed signed graphs on a pair of vertices.

(b) Corresponding weighted undirected versions.

Fig. 2. Conversion of directed signed graph into a weighted undirected graph.

We can construct a directed signed graph $G_d = (V, \vec{E})$ where V is the set of bowlers, and we draw a *positive directed* edge from a bowler A to B if A bowls better with B and a *negative directed* edge if A bowls worse with B . Figure 2a indicates the various cases between a pair of bowlers.

We can analyse these graphs to suggest bowling changes during the match as well as team (bowler) selection before the match. The basic idea is to select a set of bowlers with as many positive bowlership pairs as possible to give the captain maximum flexibility in making bowling changes during the match.

We first convert the directed signed graph G_d into an undirected weighted graph G_u . Each signed directed edge is replaced by an undirected weighted edge as depicted in Fig. 2. While the directed version for nodes C, D as well as nodes I, J are different, they result in the same weighted undirected graph. This is because a negative edge from A to B nullifies the effect of a positive edge from B to A , since A and B are essentially incompatible. After this transformation, the weight of an edge indicates the strength of bowlership between the endpoints.

Definition 4. A subgraph S of the undirected weighted graph G_u (which may contain negative edges), has an associated *average weighted degree* $W(S) = \frac{\sum w(S)}{|S|}$ where $w(S)$ is the sum of weights of all edges induced by S and $|S|$ is the number of vertices in S .

Algorithm *create-weighted-graph*

Input: directed signed graph G_d

Output: undirected weighted graph G_u

- 1: For any given pair of vertices A, B in G_d , given the directed signed edges between them, replace them with their corresponding undirected signed weighted version depicted in Figure 2. This results in G_u , the undirected weighted version of the graph. G_u may have negative edges.
-

Algorithm *bowler-select*

Input: undirected weighted graph G_u , required bowlers k // ($k = 5$ or 6 .)

Output: k bowlers maximising positive bowlerships

- 1: Let C be the set of disconnected components of G_u .
 - 2: **for** each component c in C **do**
 - 3: **for** $i \in \{2, \dots, k\}$ **do**
 - 4: Find all subgraphs $S_i = \{S_{i1}, \dots, S_{ip}\}$ of size i .
 - 5: For each subgraph, calculate $W(S_{ij}) = \frac{\sum w(S_{ij})}{|S_{ij}|}$,
 where $w(S)$ is the sum of the edge weights induced by S .
 - 6: **end for**
 - 7: **end for**
 - 8: Output subgraph S_{max} with maximum $W(S_{ij})$.
 - 9: **for** $|S_{ij}| \in \{1, \dots, (k - |S_{max}|)\}$ **do**
 - 10: Let X_{ij} be the set of cross edges connecting S_{ij} and S_{max} , and $w(X_{ij})$ be the sum of the weights of such edges.
 - 11: $WT(S_{ij}) \leftarrow (W(S_{ij}) + w(X_{ij}))$
 - 12: **end for**
 - 13: $remain \leftarrow (k - |S_{max}|)$.
 - 14: $size \leftarrow remain$.
 - 15: **while** $remain \neq 0$ **do**
 - 16: **if** $size > remain$ **then**
 - 17: $size \leftarrow remain$
 - 18: **end if**
 - 19: Select S_{ij} with $|S_{ij}| = size$ with maximum $WT(S_{ij})$.
 - 20: **if** no such S_{ij} exists, **then**
 - 21: $size \leftarrow size - 1$.
 - 22: **continue**
 - 23: **end if**
 - 24: Output S_{ij} .
 - 25: $remain \leftarrow remain - |S_{ij}|$.
 - 26: **end while**
-

The higher the average weighted degree of a subgraph, the more the bowler-ship synergy among the corresponding bowlers. Maximising the average weighted degree of the subgraph of the selected bowlers during team (bowler) selection before the match increases the flexibility of bowling changes during the match.

Algorithm **bowler-select** takes as input G_u and a number k of bowlers to be selected and returns a set of bowlers such that the average weighted degree of the selected bowlers is greedily maximised. Steps 2–6 computes the average weighted degree for each *connected* subgraph of size $\leq k$, the required number of bowlers. Step 8 outputs a subgraph S_{max} with the maximum average weighted degree. If the size of this subgraph equals k , then we are done. Otherwise, we need to find another subgraph to fulfill the k bowler requirement. For this, we need to take into account not just the weight of a candidate subgraph in consideration, but also the weight of its connectivity with S_{max} . This total weight is calculated in Steps 9–12. The selection of any remaining bowlers is done in Steps 13–26, until the required number of bowlers k is reached. The number k of required bowlers in our setting is small, so the exhaustive computation in steps 2–6 is practical. While there are other algorithms in literature [1, 5, 9], only [9] supports negative edge weights, but they are simultaneously trying to maximise the sum of positively weighted edges (reward) and minimise the sum of negatively weighted edges (risk), unlike our case where we consider a single sum.

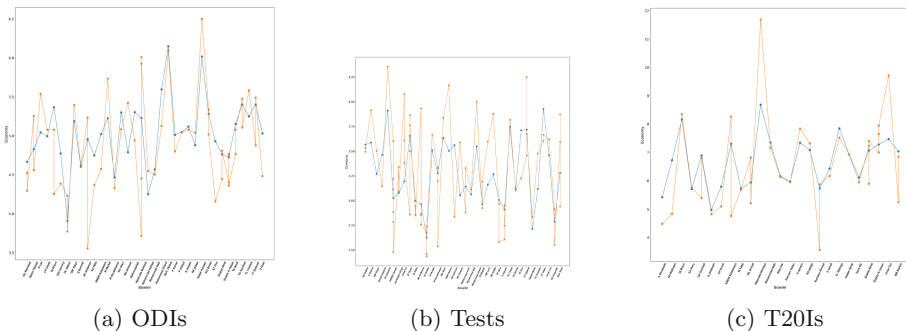


Fig. 3. Individual and bowler-ship economy: The X-axis has the name of the bowler and the Y-axis depicts the economy. The plot points in orange show the bowler-ship economy whereas the plot points in blue depict the bowler’s economy with all other bowler partners combined. The bowler-ship economy need not be better than the overall economy for a bowler-ship to be positive.

3 Results

Mann-Whitney Analysis: As we can see in Fig. 3, the overall economy of a bowler throughout his career in comparison with his overall economy in the bowler-ship has no role in determining whether the bowler-ship is ‘better’ than the

individual. In these figures, the points in orange depict the bowlership economies, which, for a majority of bowlers, lie below their individual average economies. This shows that merely taking averages across all overs is not enough.

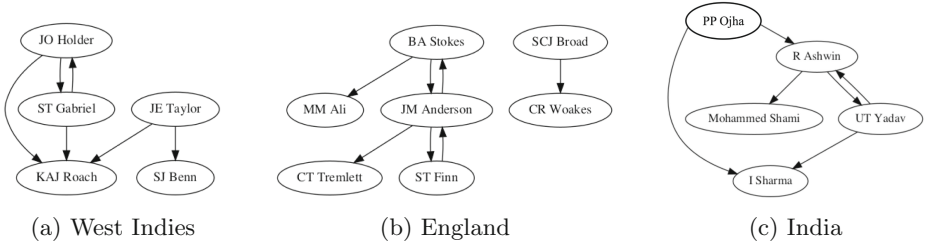


Fig. 4. Test Bowlership Network.

Interestingly, we weren't able to reject the null hypotheses of the 'two-sided' or the 'less' tests for any of the formed bowler pairs for various confidence levels. This means, we did not have any negative bowlerships. Hence, we refer to bowlerships that aren't positive but couldn't be proved negative as non-bowlerships (no edge exists between the pair of bowlers). For the Mann-Whitney experiments with *wickets taken per over* for bowlers who had bowled at least a certain number of overs, we were not able to reject any of the null hypotheses. This may be due to the reason that the taking of a wicket is a rare event.

Bowlership Networks: Figure 4 depicts the bowlerships networks for Tests:

- The test network is the densest and the T20 Network(not shown) is sparsest. This could be due to frequent bowling changes in the shorter formats.
- While it is an artefact of the chosen parameters (T_1 and T_2), two way positive bowlerships are quite uncommon across all formats. We also expected to see fewer disconnected components.
- For the Test Bowlership Network 4, the results of running Algorithm **bowler-select** look quite intuitive. The algorithm may run efficiently since the sizes of these graphs is small.

A positive bowlership does not necessarily imply better economy. In practice, we could select positive bowlerships that also have better economies. The more the outgoing edges from a bowler, the more the number of bowlers he bowls better with. The more the more incoming edges to a bowler, the more the number of bowlers who bowl better with him. For the threshold values chosen, the bowlership networks have disconnected components, with no components having more than 5 vertices. Such networks can therefore be visually analysed.

4 Conclusion

Player synergies are an integral part of team sports. Unlike games such as football and tennis, where all players are simultaneously participating in the one activity at a time, cricket and baseball have one team fielding while the other is batting. In cricket, bowling partnerships have not been analysed extensively. We define what constitutes a bowling partnership, and then analyse all formats of the game in search of effective bowler partnerships. i.e., “bowlerships”.

Our analyses showed that bowlerships exist. These bowlerships can be leverages both strategically (for team formation) and tactically (for bowling changes while the match is in progress). We presented Algorithm *bowler-select* to select bowlers which account for bowler synergies during team selection.

In future work, it would be very interesting to investigate the various bowler-ship patterns that emerge based on varying the thresholds. Is there a systematic way to determine the thresholds? Also, we need to look deeper into the reason why negative bowlerships were not found. How do we compare a pair of positive bowlerships? Can we add weights to the directed signed graph G_d ? This would help us differentiate between stronger and weaker bowler-ship pairs.

Acknowledgement. We gratefully thank the anonymous reviewers for their encouraging and insightful comments, and also suggesting reference [5].

Dedication. To Reena Kaki, who has been a source of determination, inspiration, and fearlessness for us.

References

1. Charikar, M.: Greedy approximation algorithms for finding dense components in a graph. In: Jansen, K., Khuller, S. (eds.) APPROX 2000. LNCS, vol. 1913, pp. 84–95. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44436-X_10
2. Cricsheet: Cricsheet (2021). <https://cricsheet.org/>. Accessed 20 Apr 2021
3. Croucher, J.S.: Player ratings in one-day cricket. In: Proceedings of the Fifth Australian Conference on Mathematics and Computers in Sport, pp. 95–106. Sydney University of Technology Sydney, NSW (2000)
4. d’Agostino, R.B.: An omnibus test of normality for moderate and large size samples. *Biometrika* **58**(2), 341–348 (1971)
5. Goldberg, A.V.: Finding a maximum density subgraph (1984)
6. Mukherjee, S.: Quantifying individual performance in cricket—a network analysis of batsmen and bowlers. *Phys. A* **393**, 624–637 (2014)
7. Shapiro, S.S., Wilk, M.B.: An analysis of variance test for normality (complete samples). *Biometrika* **52**(3/4), 591–611 (1965)
8. Stephens, M.A.: EDF statistics for goodness of fit and some comparisons. *J. Am. Stat. Assoc.* **69**(347), 730–737 (1974)
9. Tsourakakis, C.E., Chen, T., Kakimura, N., Pachocki, J.: Novel dense subgraph discovery primitives: risk aversion and exclusion queries. In: Brefeld, U., Fromont, E., Hotho, A., Knobbe, A., Maathuis, M., Robardet, C. (eds.) ECML PKDD 2019. LNCS (LNAI), vol. 11906, pp. 378–394. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-46150-8_23

10. Valero, J., Swartz, T.B.: An investigation of synergy between batsmen in opening partnerships. *Sri Lankan J. Appl. Stat.* **13**, 87–98 (2012)
11. Wikipedia: Bowling economy rate (2021). https://en.wikipedia.org/wiki/Economy_rate. Accessed 24 Apr 2021
12. Wikipedia: Bowling strike rate (2021). https://en.wikipedia.org/wiki/Strike_rate. Accessed 24 Apr 2021
13. Wikipedia: Mann-whitney U test (2021). https://en.wikipedia.org/wiki/Mann%E2%80%93U_test. Accessed 03 Jun 2021
14. Wisden: Wisden (2021). <https://wisden.com/stories/archive/the-ten-best-opening-partnerships-ever>. Accessed 24 Apr 2021
15. Wisden: Wisden (2021). <https://wisden.com/stories/archive/the-ten/the-ten-greatest-bowling-partnerships-from-laker-and-lock-to-wasim-and-waqar>. Accessed 24 Apr 2021



A Correction to the Heuristic Algorithm MinimalFlipSet to Balance Unbalanced Graphs

Sukhamay Kundu¹ and Amit A. Nanavati²(✉)

¹ Louisiana State University, Baton Rouge, LA 70803, USA
kundu@csc.lsu.edu

² Ahmedabad University, Ahmedabad 380009, India
amit.nanavati@ahduni.edu.in

Abstract. We present here a critical correction of the heuristic algorithm MinimalFlipSet in [8] for the *NP*-hard problem of finding a minimum size subset of edges in an unbalanced signed graph G whose ‘+’/‘-’ edge-labels can be flipped to balance G .

Keywords: balancing signed graph · heuristic algorithm · spanning tree

1 The Problem of Balancing an Unbalanced Graph

In a signed graph $G = (V, E)$, where V is the set of nodes and E is the set of edges, each edge $(x, y) \in E$ has a ‘+’/‘-’ label or sign denoted by $s(x, y)$. A political or social network can be modeled [3, 4] by a signed graph G , where V represents the individuals, E represents the pairs of individuals who communicate directly with each other, $s(x, y) = ‘+’$ indicates that the individuals x and y agree on some given issue such as voting the same way (‘yes’/‘no’) on the issue, and $s(x, y) = ‘-’$ indicates x and y disagree, voting the opposite way. The signed graphs are also used in modeling intra-cellular regulatory system [9].

A signed graph G is called balanced if each cycle in G is balanced, i.e., has an even number of ‘-’ edges. If G has at least one ‘-’ edge, this is equivalent to saying [5] that we can write $V = V_1 \cup V_2$, where V_1 and V_2 are disjoint non-empty subsets, such that each ‘+’ edge (in short, p -edge) connects two nodes in the same V_i and each ‘-’ edge (in short, n -edge) connects two nodes in different V_i ’s. A social network G , where no one lies about his/her opinion or vote, is always balanced and we can let V_1 represent the people who voted ‘yes’ and V_2 the people who voted ‘no’. For a signed graph G , the partition $V = V_1 \cup V_2$ is unique if and only if G is connected. Clearly, G is balanced if and only if each connected component of G is balanced. Now imagine that for some subset of edges $E' = \{(x_i, y_i) : 1 \leq i \leq m\} \subseteq E$ in a social network G , both x_i and y_i lie to each other about their yes/no votes. In that case, G is still balanced. However, if for each $(x_i, y_i) \in E'$ exactly one of x_i and y_i lies, then depending on E' the

graph G may be unbalanced. The two simplest ways of balancing a signed graph G are: (1) flip the signs of a subset of edges $E_{flip} \subseteq E$ such that each cycle ξ in G becomes balanced, i.e., $|E_{flip} \cap \xi|$ is even or odd according as ξ is balanced or not, and (2) delete a subset of edges $E_{del} \subseteq E$ such that $|E_{del} \cap \xi| \geq 1$ for each unbalanced cycle ξ in G . (For a social network G , an E_{del} is a set of edges (x_i, y_i) where exactly one of x_i and y_i lies and hence the importance of finding an E_{optDel} , i.e., a minimum size E_{del} because it gives the minimum number of one-sided lies, if any, among the links in G .)

It is known [6] that the optimal (minimum size) flipping edge-sets $E_{optFlip}(G)$ of a signed graph G are the same as its optimal (minimum size) deletion edge-sets $E_{optDel}(G)$. Thus, finding an $E_{optFlip}(G)$ equivalent to and equally important as that of finding an $E_{optDel}(G)$. Henceforth, we use the shorter notations $E_{optFlip}$ and E_{optDel} when the underlying graph G is clear from the context. If each edge in G is an n -edge, then finding an E_{optDel} is the same as finding a maximum size (in terms of $\#(\text{edges})$) bipartite subgraph of G . Because the latter problem is known to be NP -hard, the problem of finding an $E_{optFlip}$ for a signed graph G is also NP -hard. Henceforth, $E_{malFlip}(G)$ (in short $E_{malFlip}$) will denote a minimal flipping edge-set E_{flip} , which may or may not be optimal.

We write K_n^+ (resp., K_n^-) for a complete graph on n nodes with only p -edges (resp., n -edges) and K_n^b for a balanced signed complete graph on n nodes. Clearly, each K_n^+ is a K_n^b and no K_n^- is a K_n^b for $n > 2$. If $G = (V, E)$ is a balanced graph on n nodes, then the result in [5] on the partition $V = V_1 \cup V_2$ that we mentioned earlier shows that G is a subgraph of some K_n^b .

Example 1. The p -edges in Fig. 1(i) show an $E_{optFlip}(K_5^-)$; it consists of 4 edges forming a K_3^+ (top 3 nodes) and a vertex disjoint K_2^+ (bottom 2 nodes). Thus, $\#(E_{optFlip}(K_5^-)) = 10$ assuming the nodes are labeled. The p -edges in Fig. 1(ii) show an $E_{malFlip}(K_5^-)$ that is not an $E_{optFlip}(K_5^-)$; it consists of 6 edges forming a K_4^+ and there are 5 such edge-sets. This gives the total $\#(E_{malFlip}(K_5^-)) = 10 + 5 = 15$. We will see that for $G = K_5^-$ our algorithm always gives an $E_{optFlip}$ whereas the algorithm in [2] sometimes gives a non-optimal minimal $E_{malFlip}$. Figures 1(i)-(ii) are the only two possible structures of a K_5^b other than K_5^+ . ■

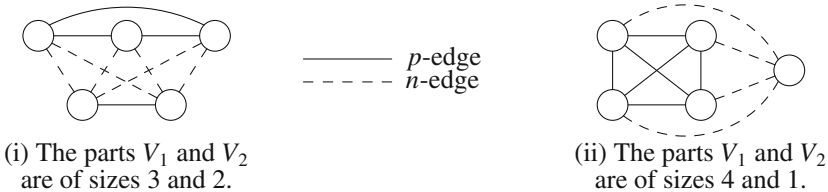


Fig. 1. The structures of a K_5^b with at least one n -edge.

In [1], bounds on $|E_{optFlip}(G)|$ are derived in terms of $|V|$ and $|E|$. For $k = |E_{optFlip}(G)|$, an $O(2^k|E|^2)$ algorithm for balancing G is given in [7]. In [10], a related NP -hard problem is considered where one wants to find a maximum size node set $V' \subseteq V$ such that the induced subgraph of G on V' is balanced.

A heuristic algorithm based on signed spectral theory and perturbations of the graph Laplacian is given in [11]. The problem of balancing G as much as possible by deleting up to b edges is considered in [12].

The critical correction of the heuristic algorithm MinimalFlipSet in [8] provided here is the following. That algorithm works with an arbitrarily selected spanning tree T in the given G (similar to the algorithm in [2]) and it computes a flipping edge-set E_{flip} , which may or may not be minimal, i.e., an $E_{malFlip}$, and which may consist of both edges in T and edges not in T . (Although the algorithm in [2] always computes an $E_{malFlip}(G)$ and that $E_{malFlip}(G)$ consists only of edges not in T , unlike the algorithm in [8], we do not know of any combination of a graph G and a spanning tree T in it for which the latter computes a larger $E_{flip}(G)$ than the $E_{malFlip}(G)$ computed by the former.) We show here that in some rare cases the algorithm in [8] selects a tree-edge e for flipping more than once and this was overlooked in [8]. Moreover, the algorithm in [8] wrongly includes such a tree-edge e in the computed E_{flip} even if e is selected for flipping an even (>0) number of times although this is equivalent to not flipping e . In addition, keeping such an e in the intermediate stages of computing the final E_{flip} may lead to include other unnecessary edges in the final E_{flip} . If we force the algorithm in [8] to not consider a tree-edge e after it is added the first time in E_{flip} , the algorithm can perform poorly and give a large E_{flip} .

2 Preliminaries

For a p -edge (x, y) , we think of its label $s(x, y) = '+'$ as a short form of '+1' and, likewise, for an n -edge we think of its label $s(x, y) = '-'$ as a short form of '-1'. For a cycle $\xi = \langle x_1, x_2, \dots, x_m, x_1 \rangle$ of length $m \geq 3$, we write $s(\xi) = \prod_{j \leq m} s(x_j, x_{j+1})$, where $x_{m+1} = x_1$. Thus, ξ is balanced if and only if $s(\xi) = '+'$, i.e., $\#(n\text{-edges in } \xi)$ is even. The cycle ξ is called simple if the nodes x_1, x_2, \dots , and x_m are distinct. If ξ is not simple and unbalanced, then there is a simple unbalanced cycle ξ' whose edges are a subset of the edges of ξ . Henceforth, by a cycle we will mean a simple cycle. Thus, G is balanced if and only if every (simple) cycle in G is balanced. Because each (simple) cycle in G is contained in a bicomponent of G , it follows that G is balanced if and only if each of its bicomponents is balanced. Henceforth, by "graph" we mean a biconnected signed graph, i.e., having just 1 bicomponent, with $|V| \geq 3$.

Definition 1. *Given a spanning tree T of G , we say an edge $(x, y) \notin T$ covers an edge $(u, v) \in T$ if (u, v) is in the unique xy -path $\pi_{x,y}$ in T . By abuse of language, we also say (u, v) covers (x, y) and this should not cause any confusion. The cycle $\xi_{x,y}$ formed by $(x, y) \notin T$ and $\pi_{x,y}$ is called the fundamental cycle of (x, y) . We write $\xi(G, T)$ or $\xi(G)$, in short, for the fundamental cycles of G for a given T .*

We sometimes use the short notations k -path for a path connecting $k + 1$ distinct nodes (and hence having k edges), k -cycle for a (simple) cycle of $k (\geq 3)$ edges, k -set for a set of $k (\geq 1)$ items, and $\max\text{Edjuc}(G)$ for the maximum $\#(\text{edge-disjoint unbalanced cycles in } G)$. The following theorem, which gives a

tight lower bound on $E_{optFlip}$ and $E_{malFlip}$ of a graph G in terms $\max\text{Edjuc}(G)$, is straightforward and is stated here without proof. We use it to show that an $E_{malFlip}(G)$ is an $E_{optFlip}(G)$ by showing that the former has size $\max\text{Edjuc}(G)$.

Theorem 1. *For each graph G , $|E_{malFlip}(G)| \geq |E_{optFlip}(G)| \geq \max\text{Edjuc}(G)$.*

Example 2. The graph G_1 in Fig. 2(i) has 8 edges and hence ≤ 2 edge-disjoint cycles. The cycles $\langle x_1, x_2, x_3, x_1 \rangle$ and $\langle x_3, x_4, x_5, x_3 \rangle$ show that $\max\text{Edjuc}(G_1) = 2$. Here, $\{(x_1, x_2), (x_4, x_5)\}$ and $\{(x_2, x_3), (x_3, x_5)\}$ are two of the three possible $E_{optFlip}(G_1)$. For the graph $G_2 = K_4^-$ in Fig. 2(ii), each of its four 3-cycles is unbalanced, and it is easy to see that $\max\text{Edjuc}(G_2) = 1 < 2 = |E_{optFlip}(G_2)|$, with $\{(x_1, x_2), (x_3, x_4)\}$ being one of 3 possible $E_{optFlip}(G_2)$. ■

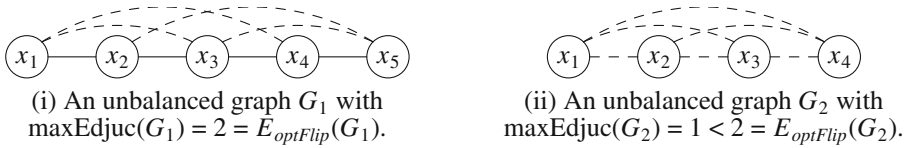


Fig. 2. Illustration of Theorem 1.

The choice of a spanning tree T in a given G greatly affects the fundamental cycles $\xi(G, T)$. This, in turn, greatly affects the $E_{malFlip}$ computed by the algorithm in [2]. Theorem 2 and Lemma 1 below show that the choice of a T and hence the computed $E_{malFlip}$ in [2] is independent of ‘+’/‘-’ labels of the edges in T in some sense. (The problem of choosing a T so that the computed $E_{malFlip}$ in [2] is an $E_{optFlip}$ is *NP*-hard.)

Theorem 2. *Given a signed graph $G = (V, E)$, a spanning tree T of G , and an arbitrary relabeling of the edges in T , there is a relabeling of 0 or more edges in $G - T$ such that there is no change in the balancedness of the fundamental cycles $\xi(G, T)$ and hence in the edge-sets $E_{malFlip}(G)$ and $E_{optFlip}(G)$.*

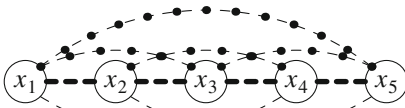
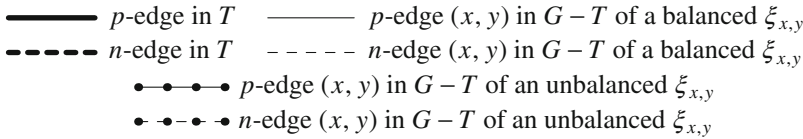
Proof. Let $(u, v) \in T$ be an edge which is relabeled (from a p -edge to an n -edge or vice-versa). We then reverse the label of each $(x, y) \in G - T$ covered by (u, v) and let G' be the new relabeled form of G . Clearly, the relabeling does not change the balancedness (or unbalancedness) of any fundamental cycles in $\xi(G, T)$. Thus, if flipping the edges $E' \subset E$ in G balances all fundamental cycles in $\xi(G, T)$ then the same is true for $\xi(G', T)$; the converse is also true. Thus, G and G' have the same edge-sets $E_{malFlip}$ and also the same edge-sets $E_{optFlip}$. We can now repeat the process for each relabeled edge $(u, v) \in T$. (Note that an edge $(x, y) \in G - T$ will have its label unchanged in the final G' if and only if $|\{(u, v) \in T : (u, v) \text{ is relabeled and } (x, y) \text{ is covered by } (u, v)\}|$ is even.) ■

Because of Theorem 2 and Lemma 1, the spanning trees in each of Figs. 5, 6(i), and 7 are shown with p -edges. By the same reason, we can make all spanning tree-edges in Fig. 4(i) p -edges by replacing the whole graph by that in Fig. 3(ii), without affecting the behavior of the algorithm *NewFlipSet* in Sect. 3.2.

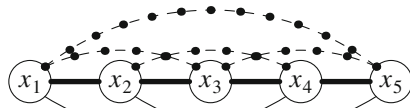
Lemma 1. For an unbalanced graph G and a spanning tree T of G , the algorithm in [2] gives the same $E_{malFlip}$ for G and its modified form G' based on a relabeling of the edges in T as in Theorem 2. The same holds for our algorithm *NewFlipSet* in Sect. 3.2.

Proof. The proof follows immediately from the following two facts: (1) the flipping edge-set generated by both the algorithms depend only the balancedness of the fundamental cycles in $\xi(G, T)$ and not on the labels of the edges in G , and (2) The graphs G and G' in the lemma have the same balancedness of the fundamental cycles. ■

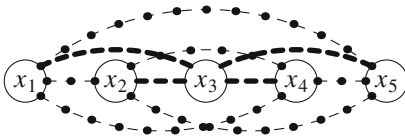
Example 3. Figures 3(i), 3(iii), and 3(v) show the three possible structurally different spanning trees T in $G = K_5^-$ and Figs. 3(ii), 3(iv), and 3(vi) show the corresponding new graph G' obtained in the proof of Theorem 2 when we relabel each n -edge in T to a p -edge. In particular, for Fig. 3(iii), we do not change the label of any edge in $G - T$, i.e., G and G' have the same label (sign) for the edges in $G - T$. It is easy to see that in all three cases of T , $E_{malFlip}(G) = E_{malFlip}(G')$ and $E_{optFlip}(G) = E_{optFlip}(G')$. ■



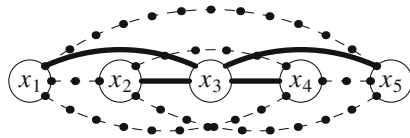
(i) A spanning tree T in $G = K_5^-$ shown by the thick lines.



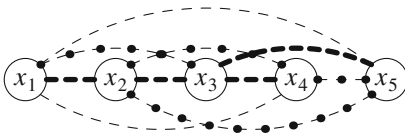
(ii) The graph G' in Theorem 2 for relabeling n -edges in T in (i) to p -edges.



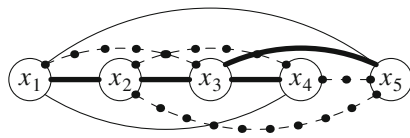
(iii) A structurally different spanning tree T in K_5^- than (i).



(iv) The graph G' in Theorem 2 for relabeling n -edges in T in (iii) to p -edges.



(v) Another structurally different spanning tree T in K_5^- than (i) and (iii).



(vi) The graph G' in Theorem 2 for relabeling n -edges in T in (v) to p -edges.

Fig. 3. Illustration of Theorem 2 for $G = K_5^-$.

2.1 Verifying Balancedness of G via the Node Labels $s(x)$

Given a signed connected graph G and a spanning tree T in G , we can assign a '+'/'-' sign (label) $s(x)$ to each node x as follows. Choose an arbitrary node, say, x_1 as the root of T and let $s(x_1) = '+'$. For each node $x_i \neq x_1$, if $\pi(x_i) = \langle x_1, x_2, \dots, x_i \rangle$ is the unique x_1x_i -path in T , then let $s(x_i) = \prod_{j < i} s(x_j, x_{j+1})$, i.e., $s(x_i) = '+'$ if $\#(n\text{-edges in } \pi(x_i))$ is even and let $s(x_i) = '-'$ otherwise. In particular, if node y is a child of node x or, equivalently, $x = \text{par}(y)$, the parent of y in T , then $s(y) = s(x)s(x, y)$, which is the same as $s(x, y) = s(x)s(y)$. If we start with the opposite label $s(x_1) = '-'$ for $x_1 = \text{root}(T)$, then the new label of each node x_i would be the opposite of its previous label. Moreover, if we choose a different node $x_i \neq x_1$ as $\text{root}(T)$ and label the nodes of T starting with the current label $s(x_i)$ for the new root x_i of T , then for each node x_j its new label is the same as its current label $s(x_j)$. In this sense, we can say that the above method gives a unique '+'/'-' labeling of the nodes of G based on T . Henceforth, the node labels $s(x)$ will correspond to those obtained with some choice of $\text{root}(T)$ and its label '+'. We say x is a p -node (resp., an n -node) if $s(x) = '+'$ (resp., '-'). Clearly, the computation of all node labels $s(x)$ takes $O(|V|)$ time. Note that if G is balanced, then the product $\prod s(x_j, x_{j+1})$ of the labels of the edges in an xy -path in G is independent of the xy -path because two xy -paths would form a cycle (which may not be simple) and that cycle is balanced. Thus, the node labels $s(x)$ are independent of T for a balanced G .

Consider now a fixed rooted spanning tree T in a graph G and the associated node labels $s(x)$ based on T . For an edge $(x, y) \in G - T$, the fundamental cycle $\xi_{x,y}$ is balanced means $s(\xi_{x,y}) = s(x, y)s(\pi_{x,y}) = +1$, i.e., $s(x, y) = s(\pi_{x,y})$, where $\pi_{x,y}$ is the xy -path in T . If z is the nearest common ancestor in T of x and y , then $\pi_{x,y} = \pi_{x,z}\pi_{z,y}$, the concatenation of the paths $\pi_{x,z}$ and $\pi_{z,y}$ in T . If $z = x$, say, then we take $\pi_{z,x}$ to be empty-path with no edges and $s(\pi_{z,x}) = +1$. We have $s(x)s(y) = s(\pi(z))s(\pi_{z,x})s(\pi(z))s(\pi_{z,y}) = s(\pi_{z,x})s(\pi_{z,y}) = s(\pi_{x,y})$. This gives Lemma 2 below to test the balancedness of the fundamental cycle $\xi_{x,y}$ for $(x, y) \notin T$. (Recall that the equation in Lemma 2 holds if $(x, y) \in T$.)

Lemma 2. *A fundamental cycle $\xi_{x,y}$ is balanced if and only if $s(x, y) = s(x)s(y)$.*

2.2 An Algorithm for E_{malFlip} by Alabandi et al.

Let G be a signed graph and T a spanning tree in G . The algorithm by Alabandi et al. [2] shown below uses the fact that G is balanced if and only if each fundamental cycle in $\xi(G, T)$ is balanced. It gives the $E_{\text{malFlip}} = \{(x, y) \in G - T: \xi_{x,y} \text{ is unbalanced}\} \subseteq G - T$, which depends on T . We will show that this $E_{\text{malFlip}}(G)$ can have size as big as $|E_{\text{optFlip}}(G)| \times O(|V|^2)$. For $(x, y) \in G - T$, we can determine whether $\xi_{x,y}$ is balanced or not using Lemma 2 in $O(1)$ time compared to the $O(|V|)$ time in [2]. This led to the more efficient (by a factor of $O(|V|)$) implementation `MinimalFlipSetOfNonTreeEdges` in [8] of the algorithm in [2].

3 Flipping Edges in T to Balance G

We need a few definitions to discuss the impact of flipping an edge in a spanning tree T of G on balancing G , i.e., its fundamental cycles $\xi(G, T)$.

Algorithm NonTreeEdgesFlipSet of Alabandi et. al in [2]:

Input: A signed graph $G = (V, E)$ and a spanning tree T of G .

Output: A minimal flipping edge set $E_{malFlip} \subseteq G - T$ to balance G .

1. For each $(x, y) \in G - T$, include it is $E_{malFlip}$ if it is unbalanced.

Definition 2. We write $\xi_{cov}(u, v) = \{\xi_{x,y} : (x, y) \notin T \text{ covers } (u, v)\}$, $\xi_{bcov}(u, v) = \{\xi_{x,y} \in \xi_{cov}(u, v) : \xi_{x,y} \text{ is balanced}\}$, and $\xi_{ucov}(u, v) = \{\xi_{x,y} \in \xi_{cov}(u, v) : \xi_{x,y} \text{ is unbalanced}\}$. Clearly, $\xi_{cov}(u, v) = \xi_{bcov}(u, v) \cup \xi_{ucov}(u, v)$, a disjoint union.

Table 1 shows $\xi_{bcov}(u, v)$ and $\xi_{ucov}(u, v)$ for the graph G and its spanning tree T in Fig. 3(iii); here, each $\xi_{bcov}(u, v) = \text{empty-set}$ because each $\xi_{x,y} \in \xi(G, T)$ is unbalanced. For the same graph G and its spanning tree shown in Fig. 3(i), $\xi_{bcov}(x_1, x_2) = \{(x_1, x_4)\}$ and $\xi_{ucov}(x_1, x_2) = \{(x_1, x_3), (x_1, x_5)\}$.

Table 1. The edge-sets $\xi_{bcov}(u, v)$ and $\xi_{ucov}(u, v)$ for $(u, v) \in T$ in Fig. 3(iii).

Edge $(u, v) \in T$	Edges $(x, y) \notin T$ such that $\xi_{x,y}$ in $\xi_{bcov}(u, v)$	Edges $(x, y) \notin T$ such that $\xi_{x,y}$ in $\xi_{ucov}(u, v)$
(x_1, x_3)	none	$(x_1, x_2), (x_1, x_4), (x_1, x_5)$
(x_2, x_3)	none	$(x_1, x_2), (x_2, x_4), (x_2, x_5)$
(x_3, x_4)	none	$(x_1, x_4), (x_2, x_4), (x_4, x_5)$
(x_3, x_5)	none	$(x_1, x_5), (x_2, x_5), (x_4, x_5)$

For a given spanning tree T of G , flipping the label of an edge $(x, y) \in G - T$ changes the balancedness of only $\xi_{x,y}$ and has no effect on any other fundamental cycle in $\xi(G, T)$. However, flipping the label of an edge $(u, v) \in T$ changes the balancedness of all fundamental cycles $\xi_{x,y} \in \xi_{cov}(u, v)$. If $\xi_{x,y} \in \xi_{bcov}(u, v)$, then flipping the label of (u, v) makes $\xi_{x,y}$ unbalanced and we need to rebalance $\xi_{x,y}$ by flipping the label of some other edge in T that covers $\xi_{x,y}$ or by flipping the label of (x, y) itself. We can often (see Example 4) obtain a smaller $E_{malFlip}$ when we use a combination of edges in T and edges in $G - T$ than that when we use only the edges in $G - T$ as in [2].

3.1 Selection Criteria for an Edge $(u, v) \in T$ for Flipping

If we flip the label of $(u, v) \in T$, then to rebalance each $\xi_{x,y} \in \xi_{bcov}(u, v)$ we can flip the label of (x, y) . Thus, flipping the label of $(u, v) \in T$ is better in reducing $\#(\text{unbalanced } \xi_{x,y} \text{ in } G)$ even with flipping each $(x, y) \in \xi_{bcov}(u, v)$ than flipping each $(x, y) \in \xi_{ucov}(u, v)$ when the inequality in eqn. (1) below is strict. This suggests the definitions of $gain(u, v)$ and $G_{unbal}(T)$ (in short, G_{unbal}) in eqns. (2)-(3). If $gain(u, v) > 1$, then it is advantageous to flip $(u, v) \in T$ to

balance G . If we write $\text{gain}(x, y) = 1$ for an unbalanced $\xi_{x,y}$ and $\text{gain}(x, y) = -1$ for a balanced $\xi_{x,y}$, then flipping (x, y) reduces G_{unbal} by $\text{gain}(x, y)$ just as flipping $(u, v) \in T$ reduces G_{unbal} by $\text{gain}(u, v)$, which can be ≤ 0 .

$$\text{Condition for flipping } (u, v) \in T : |\xi_{bcov}(u, v)| \leq |\xi_{ucov}(u, v)| - 1 \quad (1)$$

$$\text{gain}(u, v) = |\xi_{ucov}(u, v)| - |\xi_{bcov}(u, v)| \text{ for } (u, v) \in T \quad (2)$$

$$G_{\text{unbal}}(T) = \#(\text{unbalanced } \xi_{x,y} \text{ based on current labels of } E) \quad (3)$$

3.2 Corrected Form of MinimalFlipSet in [8]

As in the MinimalFlipSet algorithm in [8], the algorithm NewFlipSet below first successively selects an edge $(u, v) \in T$ in a greedy fashion to reduce G_{unbal} by the maximum amount, i.e., with maximum $\text{gain}(u, v) \geq 1$ and flips them. When no $(u, v) \in T$ has $\text{gain}(u, v) \geq 1$, it chooses the remaining unbalanced edges $(x, y) \in G - T$ in the modified G due to the flipping of the selected tree-edges to include in the final edge-set, denoted by $E_{\text{smallFlip}}$, to balance G . The 2nd half of step 3(b.1) is the correction added here to the algorithm in [8]. Because the algorithm may choose the same tree-edge (u, v) more than once and flip it, the only tree-edges (u, v) that were selected an odd number of times will be in the final $E_{\text{smallFlip}}$. In that sense, we may say that the algorithm NewFlipSet is not a truly greedy algorithm. The tree-edges that are selected an even number of times and hence not included in the final $E_{\text{smallFlip}}$ can be thought of as a kind of ‘‘corrections’’ to the greedy choice.

Algorithm NewFlipSet (corrected form of MinimalFlipSet in [8]):

Input: A connected signed graph $G = (V, E)$ and a spanning tree T of G .

Output: A small flipping edge-set $E_{\text{smallFlip}}$ consisting of possibly edges from both T and $G - T$ for balancing G .

1. Initialize $E_{\text{smallFlip}} = \text{empty-set}$.
2. For each edge $(u, v) \in T$, determine $\xi_{bcov}(u, v)$, $\xi_{ucov}(u, v)$, and $\text{gain}(u, v)$.
3. Repeat steps (a)-(c):
 - (a) Select an arbitrary tree-edge $(u', v') \in T$ such that $\text{gain}(u', v') = M = \max \{\text{gain}(u, v) : (u, v) \in T\}$.
 - (b) If $(M \geq 1)$, then do the following:
 - (b.1) Modify G by flipping the label of the edges (u', v') and $(x, y) \in \xi_{cov}(u', v')$. Add (u', v') to $E_{\text{smallFlip}}$ if $(u', v') \notin E_{\text{smallFlip}}$ and otherwise remove it from $E_{\text{smallFlip}}$.
 - (b.2) Let $T(u', v') = \{(u'', v'') \in T : (u'', v'') \text{ is covered by some } (x, y) \in \xi_{cov}(u', v')\}$, and recompute $\xi_{bcov}(u'', v'')$, $\xi_{ucov}(u'', v'')$, and $\text{gain}(u'', v'')$ for each $(u'', v'') \in T(u', v')$.
- while $(M \geq 1)$.
4. Add the non-tree edges $\{(x, y) \in G - T : \xi_{x,y} \text{ is unbalanced}\}$ to $E_{\text{smallFlip}}$.

It is worth mentioning that if $(u, v) \in T$ is selected in step 3(a) at some point, then the same (u, v) cannot be selected in the very next iteration of step 3(a). This is because $\text{gain}(u, v)$ immediately after flipping (u, v) equals the negative of $\text{gain}(u, v)$ immediately before flipping (u, v) .

The following example and Fig. 4 are the same as in [8] because no tree-edge is selected here more than once. It is included here for the sake of completeness.

Example 4. Figure 4 illustrates algorithm NewFlipSet for $G = K_5^-$. The spanning tree T in Fig. 4(i) is a depth-first tree G for root(T) = x_1 . Figure 4(ii)-(iv) show a sequence of choices of tree-edges for flipping and the results of flipping them. Finally, we flip $(x_3, x_5) \in G - T$ to balance G . The resulting $E_{\text{smallFlip}} = \{(x_1, x_2), (x_3, x_4), (x_4, x_5), (x_3, x_5)\}$ is an E_{optFlip} . If the first two choices are (x_2, x_3) and (x_3, x_4) in that order, then we must choose (x_1, x_5) and (x_2, x_4) next, again giving an E_{optFlip} . The algorithm in [2] here also gives an E_{optFlip} consisting of the four unbalanced edges in Fig. 4(i). The situation is quite different if we choose T as in Fig. 3(iii). Each $E_{\text{smallFlip}}$ determined by NewFlipSet is now an E_{optFlip} , involving 2 edges selected first from T and 2 edges selected next from $G - T$, but the algorithm in [2] gives the non-optimal $E_{\text{malFlip}} = G - T$ of size 6. ■

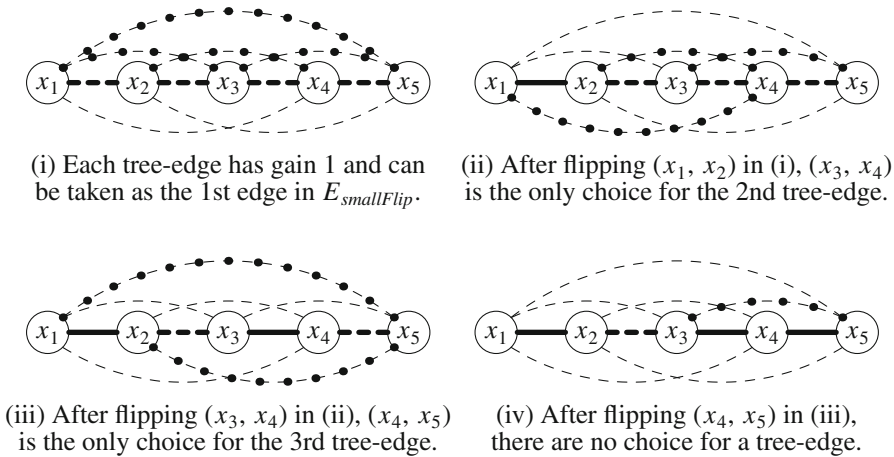
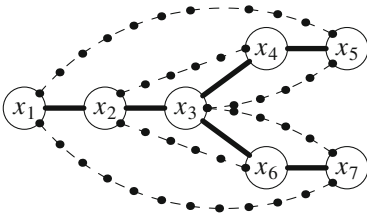


Fig. 4. Illustration of the algorithm NewFlipSet for $G = K_5^-$.

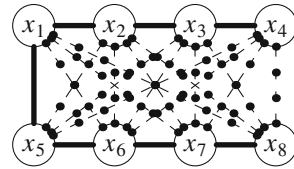
Theorem 3. *The algorithm NewFlipSet always terminates.*

Proof. Because G_{unbal} is reduced by $\text{gain}(u, v) \geq 1$ (in the current G) after flipping an $(u, v) \in T$ and by 1 after flipping an (x, y) for $\xi_{x,y} \in \xi(G, T)$, it immediately follows that the algorithm terminates. (Note that in the latter case $\xi_{x,y}$ might have been initially unbalanced or have become unbalanced due to flipping of some of tree-edges.) ■

Example 5. Fig. 5(i) shows a graph G , a spanning tree T in G , and an $E_{optFlip}$ of size 2. However, the algorithm NewFlipSet gives a minimal but non-optimal solution $\{(x_2, x_3), (x_3, x_5), (x_3, x_7)\}$ of size 3 by first choosing the tree-edge (x_2, x_3) . (The same problem arises for the algorithm in [8].) It is easy to see that if G' is the result of flipping (x_2, x_3) in G then $\max \text{Edjuc}(G') = 2$ and thus any $E_{malFlip}$ of G containing (x_2, x_3) will have size $\geq 1 + 2 = 3$. Here, the pattern of the edges within the nodes $X_1 = \{x_3, x_4, x_5\}$ and those from X_1 to $\{x_1, x_2\}$ is repeated for $X_2 = \{x_3, x_6, x_7\}$. If we have $k \geq 2$ such sets of nodes, then the algorithm NewFlipSet gives an $E_{malFlip}(G)$ of size $k + 1 > k = |E_{optFlip}(G)|$. Figure 5(ii) on the other hand shows a type of graph G and a spanning tree T in G for which the algorithm in [2] has the worst performance in that it gives the $E_{malFlip} = G - T$ of size $O(|V|^2)$ while $|E_{optFlip}| = 1$ and which is found by the algorithm NewFlipSet (and also by the algorithm MinimalFlipSet in [8]). ■



(i) The algorithm NewFlipSet gives here a non-optimal flip-set of size 3 containing (x_3, x_4) but $E_{optFlip} = \{(x_3, x_4), (x_3, x_6)\}$.



(ii) The algorithm in [2] gives here $E_{malFlip} = G - T$ of size 15 but $E_{optFlip} = \{(x_1, x_5)\}$ has size 1.

Fig. 5. Examples of two kinds of exceptional graphs, one for the algorithm NewFlipSet and one for the algorithm in [2].

3.3 Repeated Flipping of an Edge

We now give examples of the pairs (G, T) such that the algorithm NewFlipSet (and also the algorithm in [8]) flips a tree-edge multiple times.

Example 6. For the graph G in Fig. 6(i), the pattern of edges within the nodes $X_1 = \{x_1, x_2, x_3\}$ and the edges from the nodes in X_1 to the nodes $\{x_{13}, x_{14}\}$ is repeated 3 more times for nodes $X_j = \{x_j, x_{j+1}, x_{j+2}\}$, $j = 4, 7, \text{ and } 10$. Here, the algorithm NewFlipSet first selects and flips the tree-edge (x_{13}, x_{14}) and reduces G_{unbal} by $\text{gain}(x_{13}, x_{14}) = 4$. Figure 6(ii) shows the graph after flipping (x_{13}, x_{14}) . The next three iterations of step 3(a) in NewFlipSet can select any three of the tree-edges $\{(x_2, x_3), (x_5, x_6), (x_8, x_9), (x_{11}, x_{12})\}$, each of gain 1, for flipping in some order. Figure 6(iii) shows the result of flipping the edges $\{(x_2, x_3), (x_5, x_6), (x_8, x_9)\}$, reducing G_{unbal} by 1 each time. Then, NewFlipSet flips (x_{13}, x_{14}) again with current $\text{gain}(x_{13}, x_{14}) = 2$, making (x_{13}, x_{14}) an n -edge as in Fig. 6(i). Finally, the algorithm selects and flips the tree-edge (x_{11}, x_{12}) to balance G and gives the unique $E_{optFlip} = E_{smallFlip} = \{(x_2, x_3), (x_5, x_6)\}$,

$(x_8, x_9), (x_{11}, x_{12})\}$. The same $E_{optFlip}$ is obtained for other alternative choices of the tree-edges in the next three iterations of step 3(a) in NewFlipSet for the graph in Fig. 6(ii). (The algorithm MinimalFlipSet in [8] behaves exactly the same way as NewFlipSet for the graph in Fig. 6(i).) It is not difficult to create variations of the graph in Fig. 6(ii) with more nodes and edges in which NewFlipSet can flip a tree-edge an arbitrary number (≥ 2) of times if G is sufficiently large. ■

Example 7. If we modify the graph in Fig. 6(i) by deleting the edge (x_4, x_{14}) so that the spanning tree shown in thick lines becomes a depth-first tree, then the algorithm NewFlipSet finds the same unique optimal $E_{smallFlip} = E_{optFlip}$ as before without flipping a tree-edge more than once. Figure 7 shows a graph G , which is closely related to that in Fig. 6(i), and a depth-first spanning T in G with root(T) = x_1 . Here, nodes $\{x_{16}, x_{17}\}$ play the roles of nodes $\{x_{13}, x_{14}\}$ in Fig. 6(i) except that the edge (x_4, x_{17}) is deleted here so that the tree T shown in the thick lines is a depth-first tree. The edges among the extra nodes $\{x_{13}, x_{14}, x_{15}\}$ and the edges from them to the nodes $\{x_{16}, x_{17}\}$ have the same pattern as the other 3-node groups. The algorithm NewFlipSet now flips the tree-edges in the order $(x_{16}, x_{17}), (x_5, x_6), (x_2, x_3), (x_8, x_9), (x_{11}, x_{12}), (x_{16}, x_{17}),$ and $(x_{14}, x_{15}),$ with the tree-edge (x_{16}, x_{17}) flipped twice. We get the final $E_{smallFlip} = E_{optFlip} = \{(x_2, x_3), (x_5, x_6), (x_8, x_9), (x_{11}, x_{12}), (x_{14}, x_{15})\}$. ■

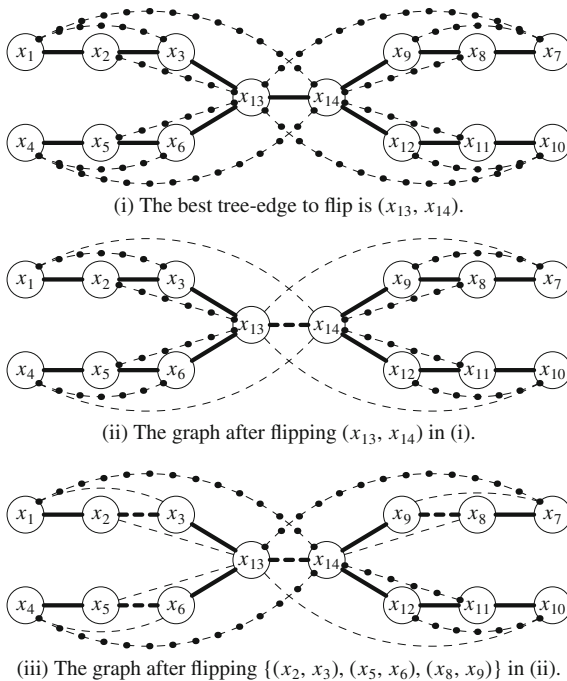


Fig. 6. The algorithm NewFlipSet flips tree-edge (x_{13}, x_{14}) twice for the graph in (i).

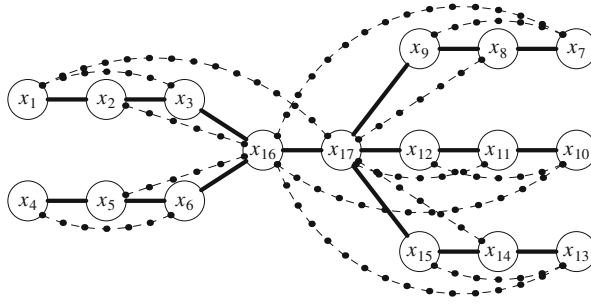


Fig. 7. A graph related to Fig. 6(i), with the thick lines forming a depth-first tree.

4 Conclusion

We present here a critical correction to our heuristic algorithm in [8] to balance a connected unbalanced signed graph G based on a spanning tree T of G . An extensive experimental runs of the new algorithm NewFlipSet showed that it often succeeds in finding an optimal (minimum size) flipping edge-set $E_{optFlip}(G)$ in cases where the algorithm in [2] failed. In a few rare cases, where NewFlipSet failed to find an $E_{optFlip}(G)$ we observed that the use of the well-known look-ahead technique [13] improves its ability to find an $E_{optFlip}(G)$.

References

1. Akiyama, J., Avis, D., Chvátal, V., Era, H.: Balancing signed graphs. *Discret. Appl. Math.* **3**(4), 227–233 (1981)
2. Alabandi, G., Tešić, J., Rusnak, L., Burtscher, M.: Discovering and balancing fundamental cycles in large signed graphs, In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–17 (2021)
3. Aref, S., Neal, Z.: Detecting coalitions by optimally partitioning signed networks of political collaboration. *Sci. Rep.* **10**(1), 1–10 (2020)
4. Cartwright, D., Harary, F.: Structural balance: a generalization of heider’s theory., *Psychological Rev.* **63**(5), 277 (1956)
5. Harary, F.: On the notion of balance of a signed graph. *Michigan Math. J.* **2**(2), 143–146 (1953)
6. Harary, F.: On the measurement of structural balance. *Behav. Sci.* **4**(4), 316–323 (1959)
7. Hüffner, F., Betzler, N., Niedermeier, R.: Optimal edge deletions for signed graph balancing. In: Demetrescu, C. (ed.) *WEA 2007. LNCS*, vol. 4525, pp. 297–310. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72845-0_23
8. Kundu, S., Nanavati, A.A.: A more powerful heuristic for balancing an unbalanced graph. In: Cherifi, H., Mantegna, R.N., Rocha, L.M., Cherifi, C., Micciche, S. (eds.) *Complex Networks and Their Applications XI: Proceedings of The Eleventh International Conference on Complex Networks and their Applications: COMPLEX NETWORKS 2022 — Volume 2*, pp. 31–42. Springer International Publishing, Cham (2023). https://doi.org/10.1007/978-3-031-21131-7_3

9. Ma'ayan, A., Lipshtat, A., Iyengar, R., Sontag, E.D.: Proximity of intracellular regulatory networks to monotone systems. *IET Syst. Biol.* **2**(3), 103–112 (2008)
10. Figueiredo, R., Frota, Y.: The maximum balanced subgraph of a signed graph: applications and solution approaches. *Eur. J. Oper. Res.* **236**(2), 473–487 (2014)
11. Ordozgoiti, B., Matakos, A., Gionis, A.: Finding large balanced subgraphs in signed networks. In: *Proceedings of The Web Conference 2020*, pp. 1378–1388 (2020)
12. Sharma, K., Gillani, I.A., Medya, S., Ranu, S., Bagchi, A.: Balance maximization in signed networks via edge deletions. In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021*, pp. 752–760 (2021)
13. Russell, S.J., Norvig, P.: *Artificial Intelligence: A modern approach* (4th ed.), Prentice-Hall



Influential Node Detection on Graph on Event Sequence

Zehao Lu^{1(✉)}, Shihan Wang^{1(✉)}, Xiao-Long Ren², Rodrigo Costas³,
and Tamara Metzke⁴

¹ Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
com3dian@outlook.com, s.wang2@uu.nl

² Yangtze Delta Region Institute (Huzhou), University of Electronic Science
and Technology of China, Chengdu, People's Republic of China

³ Centre for Science and Technology Studies, Leiden University,
Leiden, The Netherlands

⁴ Technology, Policy and Management, Delft University of Technology,
Delft, The Netherlands

Abstract. Numerous research efforts have centered on identifying the most influential players in networked social systems. This problem is immensely crucial in the research of complex networks. Most existing techniques either model social dynamics on static networks only and ignore the underlying time-serial nature or model the social interactions as temporal edges without considering the influential relationship between them. In this paper, we propose a novel perspective of modeling social interaction data as the graph on event sequence, as well as the Soft K-Shell algorithm that analyzes not only the network's local and global structural aspects, but also the underlying spreading dynamics. The extensive experiments validated the efficiency and feasibility of our method in various social networks from real world data. To the best of our knowledge, this work is the first of its kind.

Keywords: Influential Node Detection · Dynamics of Network · Non-epidemic Spreading

1 Introduction

Real-world networks exhibit high complexity as there are a large number and variety of nodes, interactions, or relationships. Therefore, modeling the spreading phenomenon (which could be either informative or physical) is difficult yet critical in a variety of fields such as infectious disease research [1], social media study [2], communication study [3]. The most intuitive way to find those opinion leaders in a network is to rank the nodes according to their influence. Numerous studies have been done to identify opinion leaders in various complex networks [4,5]. The majority of research on opinion leader detection in complex networks has assumed that the network is a static model, in which each node corresponds to an individual and the edges represent their long-term relationships.

However, many real-world social systems cannot be accurately depicted using static graphs due to their inability to account for temporal fluctuations in interactions and assume that the graph’s structure remains unchanged [6]. The existing temporal network models mainly constructed network models by adding time-respecting edges on static models, therefore being able to model edge dissolving network phenomena.

Although the already existing network models are able to describe some network spreading mechanisms, these models overlook the relevant causal relationships within a sequence of events. Additionally, they fail to provide a quantification of influence. In some non-epidemic spreading situations (e.g. information spreading), people actively make decisions or react to received information rather than being passively affected [7]. In this case, the process of one person influencing another becomes a ‘two-step’ process: an actor first posts a message or commits a change, and another network member then reacts by doing the same (like posting or committing). The causal relationship here hinges on the chain of events, not just the network of individuals, as these active participants react to content or changes rather than just the sender [8]. These responsive actions or interactions can be summarized as ‘events’ in a non-epidemic network. See the example in Fig. 1.

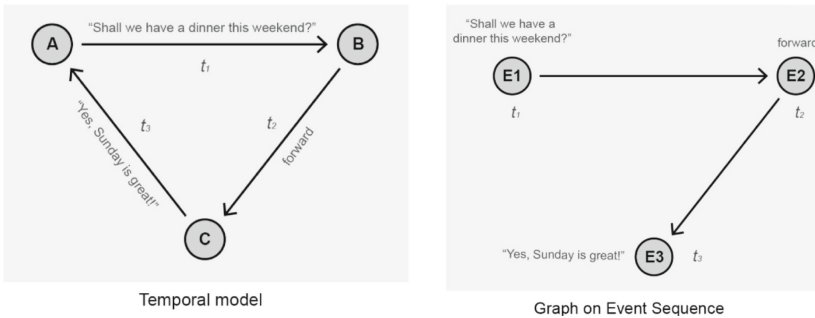


Fig. 1. A texting network example. The temporal model is not able to capture the affection relationship $E1 \rightarrow E2 \rightarrow E3$, as the adjacent edges do not necessarily have a direct affection relationship in a temporal graph.

In a texting network, the chain of messages is usually highly informative and has long-distance influences. Thus, sending the message can be conceptualized as an *event*. When A sends a message to B saying ‘*shall we (A,B,C) have a dinner this weekend?*’ (first event), and B forward this to C afterward for confirmation (second event). Then, C texts A with ‘*Yes, Sunday is great!*’ (third event). In this example, the third event is obviously a direct result of the first event, in other words, the first event influences the third event. However, the influence and chain-like relationships will not be captured in the current static or temporal network models (demonstrated in Fig. 1 left), as those models can only treat

individuals as nodes and interactions as edges, while the relationship between edges (events) is usually ignored [6].

To fix the mentioned gap, this study offers a new perspective to model the graph on event sequence and to detect influential nodes in dynamic networks. In this model, we focus on social networks constructed with chains of informative events, and the influence of each social opinion is precisely measured by applying the Hawkes process model [9], which is a stochastic processes model that describes the progression of events through time, wherein the occurrence of prior events can influence the probability of future events. Furthermore, we propose a novel opinion leader mining method, the Soft K-shell, which functions on the graph on event sequence. The Soft K-shell algorithm applies the Hawkes process model on influence measurement and is able to use a variety of node properties (both topological and contextual) to find influential nodes. We conduct experiments on networks of different sizes and types to assess the Soft K-Shell's performance. The experimental results show that the proposed algorithm is feasible to perform better than the current benchmark algorithms.

2 Proposed Method

This section presents the proposed model (the graph on event sequence), and a related novel influential node detection algorithm (namely Soft K-shell).

2.1 Graph on Event Sequence

This research proposes a new type of social network graph model: the graph on event sequence. To make it clearer, the explanation of this term is given in this subsection. Unpack:

- 1: The **graph** is a directed structure (V, E) made of nodes V and edges E .
- 2: The **event sequence** is a sequence of interacting nodes $\{v_i\}$ in successive order. The temporal feature of the informatic flow is stored in the event sequence. The event sequence can be generated according to various situations. In the Fig. 2 example, to identify influential social media users we may define the top right event sequence, whereas the bottom right event sequence is more suitable for studying content of messages (e.g. influential scientific papers) in the social media community.
- 3: The term '**on**' here simply means that the graph itself is extracted from the event sequence.

Why 'Graph on Event Sequence'? One major disadvantage of the static graph model for studying social media is that they do not take the temporal features of information diffusion into account. That is to say, all interactions within the populations are considered equally in a static model, even if they are not. On the other hand, existing temporal models mainly consider network

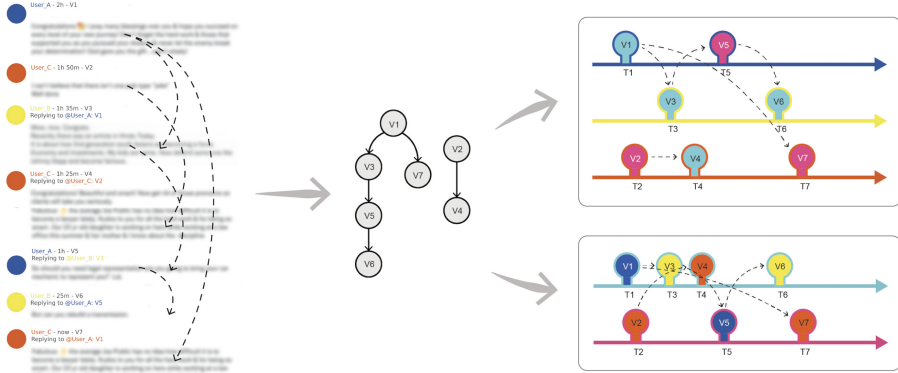


Fig. 2. A example of a graph on event sequence. The left plot illustrates an online social media screenshot, the colored circles (or ‘bulbs’) symbolize the network elements connected by the topological graph. Each ‘bulb’ (node) represents a post on social media. The extracted graph is shown in the middle, where each edge is an interaction between posts (e.g. reply or retweet). Interestingly, the same graph can be extracted from different event sequences (the right plots). For example, the top event sequences represent the posts from three users, where each user’s posts are lined on the same arrow and the interaction between posts is represented by the dotted arrow. Here, the same contour color means the same users, while the filled color suggests the topic that the post is concerning. On the other hand, the bottom event sequences indicate the posts concerning two topics. In this event sequence, the contour color shows the topic and the filled color shows the publisher (e.g. Twitter user).

models only consisting of individuals while failing to construct a network for events. The graph on event sequence model provides an approach of combining the topological and temporal features for influential measurement to address the above issues, the graph on event sequence modeled the underlying dynamical process of events chain as a Hawkes process. Therefore, it is able to accurately describe the impact of each network event on its respondents. The graph on the event sequence model primarily serves as a content impact mining method, but it can also measure an individual’s influence within a social network by summing the influences of events in a sequence, offering versatility for analyzing the impact of different content attributes, such as ranking influential journals based on their overall influence in a citation network.

2.2 Hawkes Process for Influence Measurement

The Hawkes process [9] is a mathematical model used in statistics and stochastic processes to describe the probability of occurrence of events $\{v_i\}$ over time using an intensity function. Hawkes processes is widely used in the area of influence measurement of non-epidemic spreading process [10–13]. The classic Hawkes Process is a counting process that models ‘self-excited’ events over a time period. In

this research, a typical type of the multivariate Hawkes processes, the topological Hawkes process model is applied to measure the influence of receiving information on reaction [14,15]. We propose to measure the impact of each network event on its respondents using the Hawkes process.

Definition 1. *A graph on event sequence is a direct graph $G = (V, E)$ that each node $v \in V$ is an element in an event sequence, a directed edge $e \in E$ represents the interaction relationship between the two nodes.*

Definition 2. *Given a threshold R , a graph $G = (V, E)$ and an influential function $I = I(v)$ where v is the node and $I(v)$ is the rank, a node $v \in V$ is called opinion leader if and only if it is in the top R nodes among the influentially ranking list that is sorted by $I(v)$.*

Definition 3. *The weight of each edge is defined as $W(e) = e^{-\beta(T(u)-T(v))}$ where u, v are the nodes connected by edge $e = v \rightarrow u$, $T(v)$ represents the timestamp when v occurred, and β is the scale parameter.*

Lemma 1. *Each connected component of the graph on event sequence is an acyclic graph.*

Proof. For any path in the graph on event sequence, the timestamp of successor nodes is later than their precursor’s. Therefore if there is a cycle in any connected component, a time machine is invented.

Theorem 1. *Each connected component of the graph $G = (E, V)$ is a feed-forward network. For each node v in the graph, its direct successor nodes $S(v)$ is $\{u|u \in V, (v, u) \in E\}$, and the total influence of node v on its neighbor is,*

$$I(v) = \sum_{u \in S(v)} \alpha(u)e^{-\beta(T(u)-T(v))} \tag{1}$$

Proof. By lemma 1, we have that each connected component of the graph $G = (E, V)$ is directed and acyclic, thus, it is a feed-forward network. For each $u \in S(v)$, the influence of v on u is $\alpha(u)e^{-\beta T(u)-T(v)}$. By summing up those influences we have the above result. Note that the term α only depends on the property of node u , therefore it can be estimated by various machine learning or statistical algorithms.

Lemma 2. *For any two nodes v and u , the influence of v on u is $m \times \alpha(u)e^{-\beta(T(u)-T(v))}$ if $T(v) \leq T(u)$ and u, v both belong to a same connected component of G , otherwise the influence is 0. Here m is the number of distinct paths from v to u . If we do not consider any node properties then α is set to 1.*

Proof. Suppose u and v are below to same connected component of G , by Lemme 1, this connected component is an acyclic directed network. Thus there are m paths from v to u and none from u to v . Select one chain p , let $\{w_i\}$ denote the nodes on one of those chains from v to u as follows,

$$p : v \rightarrow w_1 \rightarrow w_2 \rightarrow \dots \rightarrow w_n \rightarrow u \tag{2}$$

By the definition of intensity function and formula (1), there is,

$$I_u^p(v) = \alpha(u)e^{-\beta(T(u)-T(v))}$$

If there are m different chains from v to u , the summed influence should be,

$$I_u(v) = \sum_{p \in P(v,u)} I_u^p(v) = \sum_{p \in P(v,u)} \alpha(u)e^{-\beta(T(u)-T(v))} = m\alpha(u)e^{-\beta(T(u)-T(v))} \tag{3}$$

where $P(v, u)$ represents the set of paths from v to u . The influence of a node v on itself is $\alpha(v)e^{-\beta(T(v)-T(v))} = \alpha(v)$. If u, v do not belongs to a same connected component of G , $I_u(v)$ is naturally 0.

Theorem 2. *The overall influence of a node v in the graph can be obtained by directly computing its accumulated influence, which is defined as*

$$A(v) = \alpha(v) + \sum_{u \in S(v)} e^{-\beta(T(u)-T(v))} A(u) \tag{4}$$

where $S(v)$ is the set of direct successors of v . Note that m is not shown in formula (5) as there will only exist one direct path (an edge) for each neighbouring (u, v) pair.

Let $S'(v)$ denote the set of successors of v in G . By Lemma 2, there is

$$I_w^p(v) = \alpha(w)e^{-\beta(T(w)-T(v))} \quad \text{and,} \quad A(v) = \sum_{w \in S'(v)} \sum_{p \in \{P(v,w)\}} I_w^p(v)$$

where $p(v, w)$ denotes a direct path from v to w . Therefore,

$$\begin{aligned} A(v) &= \sum_{w \in S'(v)} \sum_{p \in \{P(v,w)\}} \alpha(w)e^{-\beta(T(w)-T(v))} \\ &= \alpha(v) + \sum_{w \neq v, w \in S'(v)} \sum_{p \in P(v,w)} \alpha(w)e^{-\beta(T(w)-T(v))} \end{aligned} \tag{5}$$

The summed form in (5) can also be written as $\sum_{w \neq v, w \in S'(v)} \sum_{p \in P(v,w)} \cdot = \sum_{p \in P(v, \cdot)} \cdot$ where $P(v, \cdot)$ represents all paths in G that start from v . Further, Note that any path p in G must include at least one node u which is a direct successor of v , one can split $P(v, \cdot)$ into $\bigcup_{u \in S(v)} P(u, \cdot)$, where u is the closest node to v on path p . And for u in $S(v)$,

$$e^{-\beta(T(w)-T(v))} = e^{-\beta(T(w)-T(u))} e^{-\beta(T(u)-T(v))} \tag{6}$$

Therefore, the summed form in (5) is equal to

$$\sum_w \sum_p \cdot = \sum_{p \in P(v, \cdot)} \alpha(w)e^{-\beta T(w)-T(v)} = \sum_{u \in S(v)} e^{-\beta(T(u)-T(v))} A(u) \tag{7}$$

By combining formula (5) and (7) there is,

$$A(v) = \alpha(v) + \sum_{u \in S(v)} e^{-\beta(T(u)-T(v))} A(u) \tag{8}$$

2.3 Soft K-Shell Algorithm

Accordingly, we propose a novel method, namely the Soft k-shell algorithm, that considers the topological Hawkes process of interaction. The proposed method addresses the task of detecting influential nodes by assessing the global influence of individual nodes in the graph using the Hawkes process and subsequently ranking them based on their overall influence¹. The algorithm is executed as follows: first, if the node attribute is used, each node v 's ranking is initially set to $\alpha(v)$, as it is proved by Theorem 2, the influence of a node v on itself is $\alpha(v)$. Otherwise, $\alpha(v) = 1$. It is advised that $\alpha(v)$ be set to a value between 0 and 1. Second, the global influence of node v whose direct successor u has a 0 degree ($\{u|u \in V, (v, u) \in E, degree(u) = 0\}$) is calculated recursively by adding its self influence (which is initialized as α) and its influence over u , which is $e^{-\beta(T(u)-T(v))}A(u)$. After performing this computation, node u will be permanently removed from the graph, and $A(u)$ is the final result of u 's global influence. By repeatedly removing zero degree nodes us and updating the global influence of their predecessor vs , the graph G itself is also shrinking. As the graph is acyclic, there will always be new zero-out-degree nodes until all nodes are removed. We name this method the Soft K-shell algorithm. Two versions of the Soft K-Shell algorithm are considered here. If node-property is true then the property of nodes is used as $\alpha(v)$ ², otherwise $\alpha(v)$ is set to 1.

3 Experiments and Results

This study compares the performance of our method with four other methods for detecting opinion leaders in dynamic social networks. To evaluate the feasibility and generalizability of our method, six real world data sets of various types are used. Four of them are a collection of long-term Dutch tweets containing Coronavirus tags from February 2020 to January 2021 [16], which are split into four different data sets (NCF(reply), NCF(quote), NCF(retweet), NCF(together)) depending on their interaction type. The fifth data set (NCJ) contains short-term Dutch tweets related to the COVID-19 pandemic (three hours around a pandemic press conference on 14th January 2022) [17], while the sixth data set (DBLP V1) is the first version of the DBLP dataset (citation network) [18]. The specifications of the data sets utilized in the studies and the source code can be found in the paper's repository.

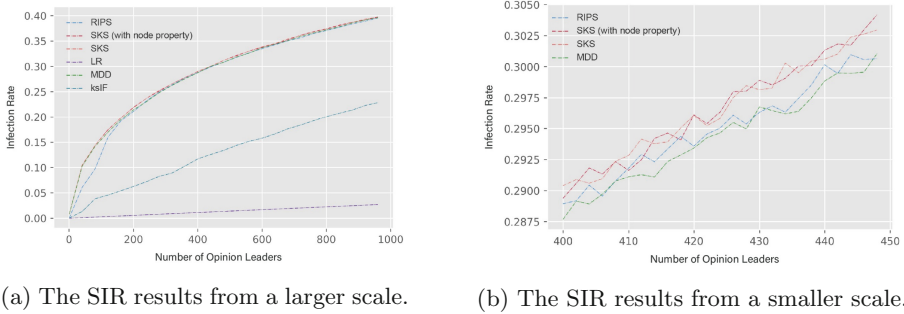
3.1 SIR Simulation Results

In this study, we compare the experimental results of our method with those of four other state-of-the-art algorithms, utilizing the Susceptible-Infected-

¹ A pseudocode of the proposed Soft K-shell algorithm could be found in this paper's repository, <https://github.com/com3dian/SoftKShell>.

² For the soft k-shell model, the parameter of node properties α is user-defined. In our conceptual framework, this value is assumed to be calculated using other machine learning techniques and given as the prior knowledge. Consequently, we do not delve into the methodologies for obtaining this parameter in the paper.

Removed (SIR) [19] simulation infection rate and computational complexity. The SIR model is a widely used framework for simulating the diffusion of information within networks. In scenarios where opinion leaders are designated as the initial set of infected nodes, upon achieving convergence, this model yields an infection rate that serves as a quantitative measure of the influence exerted by these initial opinion leaders throughout the entire information diffusion process. The transmission rate τ of each edge is 0.98 and the recovery rate γ of each node is 0.02. In every simulation, the top 5% ranked nodes were selected and infected initially. The simulations were conducted for 50 rounds. During our experiments, we used the number of followers as α for the NCF/NCJ datasets and citation numbers as α for the DBLP dataset.



(a) The SIR results from a larger scale.

(b) The SIR results from a smaller scale.

Fig. 3. Following the SIR model, the infection rates of various approaches on the NCF(together) network are presented (with different zoom scales). The x-axis shows the number of the initial opinion leaders (seed set), while the y-axis shows the final proportion of infection (in percentage). All comparable methods use the same size of opinion leaders' set as the seed set of SIR. The plot shows that the proposed method, Soft K-Shell, outperforms all other methods. The final SIR infection rate does not significantly differ between the two versions of the Soft K-Shell (with and without node properties). MMD and RIPS also perform well.

The results of Soft K-Shell (SKS) and several other state-of-the-art methods in the literature were used to validate the proposed model and algorithm. Leader-Rank algorithm (LR) [20], mixed degree decomposition (MDD) [21], k-shell iteration factor (ksIF) [22], and Randomized Influence Paths Selection (RIPS) [23] are selected as the baseline. The first three algorithms are widely used baselines in opinion leader mining tasks, while the RIPS is the state-of-art algorithm according to its results [23]. The parameters used in the following experiment are suggested by its authors.

In particular, four versions of the RIPS algorithm are used for comparison. Two versions consider the Hawkes intensity $e^{-\beta(T(u)-T(v))}$ as edge weight while the other two use equal weight for all edges. Additionally, we consider two versions of the Soft K-Shell algorithm in the experiments (with node property or not). Figures 3a and 4 depict the SIR model infection rates on the NCF(together)

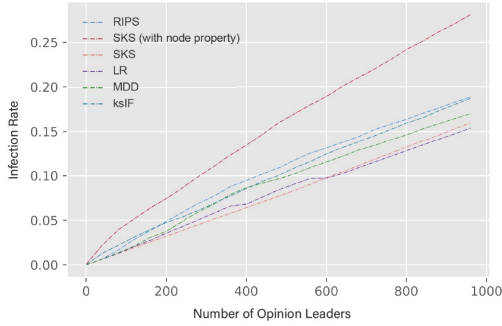


Fig. 4. In the NCJ dataset, the Soft K-Shell algorithm with node property achieves the highest infection rate, while the K-Shell iterative factor algorithm also performs well. Notably, the performance between the two versions of Soft K-Shell differs a lot, indicating the significant impact of node property in enhancing algorithm performance. This enhancement can be attributed to the characteristics of the NCJ dataset, which mainly consists of short-term Twitter interactions. In short-term information diffusion, users with a larger follower count exert a more significant influence. This phenomenon enhances the advantage of the Soft K-Shell algorithm with respect to its feasibility in combining node properties.

data set and the NCJ data set respectively. To better show the subtle difference in Fig. 3a, we also present plots of infection rates from the smaller scale in Fig. 3b.

Table 1 presents the highest infection rates obtained through SIR simulations across all six datasets, considering various algorithm settings. Our proposed algorithm consistently achieved the highest infection rate across datasets of varying sizes and social interactions. This demonstrates the feasibility and generalizability of our opinion leader mining method.

Table 1. The infection rate after SIR model reached the convergence of different methods results. Top 5% nodes found by each algorithm are used in the SIR model. The highest score on each data set is bolded. The Soft K-shell algorithm outperforms every other model.

Datasets	LR	MMD	Ksif	RIPS	SKS	SKS with node property
NCF(reply)	0.048	0.087	0.102	0.092	0.133	0.133
NCF(quote)	0.155	0.240	0.199	0.212	0.250	0.250
NCF(retweet)	0.050	0.481	0.373	0.469	0.490	0.488
NCF(together)	0.050	0.481	0.373	0.469	0.489	0.490
NCJ	0.056	0.067	0.067	0.076	0.050	0.108
DBLP V1	0.100	0.164	0.140	0.181	0.198	0.210

Table 2. Computational complexity of different methods

Algorithms	Complexity
K-Shell	$O(V ^2)$
MDD	$O(V ^2)$
ksIF	$O(V ^2 + E)$
RIPS	$O(V \log(V) + 2 E)$
Soft K-Shell	$O(V ^2)$

3.2 Computational Complexity Results

In this paper, we also assess the proposed method based on the computational complexity. The time complexity of five algorithms is listed in Table 2. The Soft K-shell algorithm has the same time complexity as the K-shell algorithm. In terms of the number of nodes $|V|$, RIPS appears to be the most effective algorithm because its highest ordered component is $|V|\log(|V|)$, whereas other algorithms have the term $|V|^2$. However, in many real world networks, the amount of edges $|E|$ has the same order as $|V|^2$. Additionally, because the RIPS technique uses a Monte-Carlo-based methodology, the constant coefficients in the asymptotic complexity term are significantly larger than those in the other four methods.

3.3 Soft Shell Decomposition

Besides the quantitative results, we had another interesting discovery even though the Soft K-Shell algorithm does not compute a ‘hard’ decomposition of the network, its computed node ranking follows a multimodal distribution. As demonstrated in Fig. 5, the scatter plot of the NCF(together) data set has four ‘shells’ which together can be described as a soft shell decomposition. This is also the inspiration for naming this algorithm. The four ‘shells’ from the inside out each represent one type of posts in the NCF(together) dataset. The most centered shell (purple) represents the most influential tweets; while the second shell (deep blue) represents the posts that have a small range impact in the ‘local’ social network; the third shell (green) represents the most ordinary posts; the fourth shell (yellow) represents the ‘dead’ posts that are rarely noticed by any other people. It can be concluded that the Soft K-shell ranking result is also able to reveal the soft shell nature of a network. The above soft shell decomposition result not only gives the ranking of posts regarding their influence, but also gives a distribution of their influence. With the help of that distribution, a more specific community opinion study on the purple shell can be carried out, since the purple shell is highly representative of the community and is much smaller compared to the entire network. As a result, the soft shell decomposition can immensely benefit the analysis of social media platforms’ processes for forming opinions based on their content.

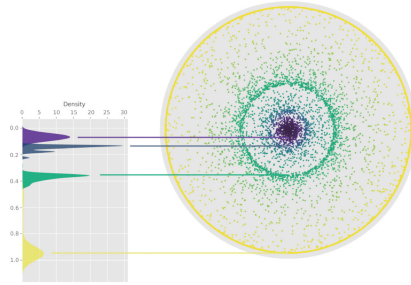


Fig. 5. The shell decomposition or the coreness decomposition of the NCF(together) data set. Each data point is a rank of a node in NCF(together)(which representing a post on Twitter), which has been scaled by exponential function $e^{1-rank(v)}$, where $rank(v)$ is ranging from 1 to $+\infty$. The coreness trait is also seen in the result of Soft K-Shell algorithm, despite the fact that it does not use the original K-Shell technique.

4 Conclusion

With the increasing popularity of social networks, resolving essential problems about these networks, such as opinion leader detection, has gained a lot of interest. However, most of the existing static and temporal methods fail to explain the underlying dynamic of non-epidemic spreading process in the networks. Therefore, a new method that considers the Hawkes process of information flow to combine the topological and temporal features for influential measurement, is proposed in this paper. The proposed model outperforms the state-of-the-art model by a significant margin while keeping a competitive time complexity. In future work, more theoretical study shall be accomplished on finding the scale parameter β and the node's property α . Also, we plan to investigate more into using the time serial structure of the Hawkes process to forecast social network dynamics and its impact on public opinion.

References

1. Goh, K.-I., Cusick, M.E., Valle, D., Childs, B., Vidal, M., Barabási, A.-L.: The human disease network. *Proc. Natl. Acad. Sci. USA* **104**, 8685–8690 (2007)
2. Vespignani, A.: Modelling dynamical processes in complex socio-technical systems. *Nat. Phys.* **8**, 32–39 (2012)
3. Garton, L., Haythornthwaite, C., Wellman, B.: Studying online social networks. *J. Comput.-Mediat. Commun.* **3**, JCMC313 (1997)
4. Bamakan, S.M.H., Nurgaliev, I., Qu, Q.: Opinion leader detection: a methodological review. *Expert Syst. Appl.* **115**, 200–222 (2019)
5. 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)
6. Holme, P., Saramäki, J.: Temporal networks. *Phys. Rep.* **519**(3), 97–125 (2012). *Temporal Networks*

7. Zheng, M., Lü, L., Zhao, M.: Spreading in online social networks: the role of social reinforcement. *Phys. Rev. E* **88**, 012818 (2013)
8. Inwagen, P.V.: *An Essay on Free Will*. Oxford University Press, New York (1983)
9. Hawkes, A.G.: Spectra of some self-exciting and mutually exciting point processes. *Biometrika* **58**(1), 83–90 (1971)
10. Kobayashi, R., Lambiotte, R.: TiDeH: Time-dependent Hawkes process for predicting retweet dynamics. *Proc. Int. AAAI Conf. Weblogs Soc. Media* **10**, 191–200 (2021)
11. Zadeh, A.H., Sharda, R.: Hawkes point processes for social media analytics. In: Iyer, L.S., Power, D.J. (eds.) *Reshaping Society through Analytics, Collaboration, and Decision Support*. AIS, vol. 18, pp. 51–66. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-11575-7_5
12. Alvari, H., Shakarian, P.: Hawkes process for understanding the influence of pathogenic social media accounts. In: 2019 2nd International Conference on Data Intelligence and Security (ICDIS), pp. 36–42, IEEE (2019)
13. Filimonov, V., Sornette, D.: Quantifying reflexivity in financial markets: toward a prediction of flash crashes. *Phys. Rev. E* **85**, 056108 (2012)
14. Cai, R., Wu, S., Qiao, J., Hao, Z., Zhang, K., Zhang, X.: THP: topological hawkes processes for learning granger causality on event sequences. *arXiv preprint arXiv:2105.10884* (2021)
15. Embrechts, P., Liniger, T., Lin, L.: Multivariate hawkes processes: an application to financial data. *J. Appl. Probab.* **48**(A), 367–378 (2011)
16. Colavizza, G., Costas, R., Traag, V.A., van Eck, N.J., van Leeuwen, T., Waltman, L.: A scientometric overview of covid-19. *PLoS ONE* **16**(1), e0244839 (2021)
17. Wang, S., Schraagen, M., Sang, E.T.K., Dastani, M.: Dutch general public reaction on governmental COVID-19 measures and announcements in twitter data. *arXiv preprint arXiv:2006.07283* (2020)
18. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: AAAI (2015)
19. Ross, R.: An application of the theory of probabilities to the study of a priori pathometry.-Part I. *Proc. Roy. Soc. Lond. A* **92**(638), 204–230 (1916)
20. Lü, L., Zhang, Y.-C., Yeung, C.H., Zhou, T.: Leaders in social networks, the delicious case. *PLoS ONE* **6**(6), e21202 (2011)
21. Zeng, A., Zhang, C.-J.: Ranking spreaders by decomposing complex networks. *Phys. Lett. A* **377**(14), 1031–1035 (2013)
22. Wang, Z., Zhao, Y., Xi, J., Du, C.: Fast ranking influential nodes in complex networks using a K-shell iteration factor. *Phys. A: Stat. Mech. Appl.* **461**, 171–181 (2016)
23. Rezaei, A.A., Jalili, M., Khayyam, H.: Influential node ranking in complex networks using a randomized dynamics-sensitive approach. *arXiv preprint arXiv:2112.02927* (2021)



Decentralized Control Methods in Hypergraph Distributed Optimization

Ioannis Papastaikoudis^(✉) and Ioannis Lestas

Department of Engineering, University of Cambridge, Trumpington Street,
Cambridge CB2 1PZ, UK
{ip352, ic120}@cam.ac.uk

Abstract. We study a distributed optimization problem which uses an undirected, unweighted hypergraph communication from a dynamical system viewpoint. The stability analysis of the dynamical system is conducted with the use of approaches relevant to non linear decentralized control where we also use the hypergraph Laplacian matrix for the decomposition of the dynamical system. Additionally, we present a Laplacian matrix for the case of a directed and weighted hypergraph and we show how this Laplacian matrix is decomposed for the stability analysis of the distributed optimization problem with the specific directed and weighted hypergraph communication structure.

Keywords: Hypergraphs · Distributed Optimization · Decentralized Control

1 Introduction

Distributed optimization can be traced back to the seminal works of [5] and [6] and a common way of solving such problems is the use of first order methods as in [12] and [13]. In this paper we study a distributed optimization problem which uses a hypergraph communication despite the fact that graph communication is the most commonly used graphical structure in distributed optimization problems [14]. Hypergraphs were introduced in [7] as a generalization of graphs. The importance of the hypergraph lies in the fact that it allows more than two nodes to be linked in the same edge (hyperedge). As a result, a hypergraph can depict more complex relationships compared to the communication structure of a graph. This different communication structure exists in reality e.g. large online social networks, supply chain management, etc. Various advantages of the hypergraph communication are presented in [8–10]. It is important to note that a hypergraph based distributed optimization problem can also be viewed as a multiple consensus problem, a consensus must be achieved among the nodes (agents) that are attached to each hyperedge. A review of distributed optimization and consensus theory can be found in [14] and [15] respectively. As the communication matrix of the distributed optimization problem we will use the Bolla's Laplacian for hypergraphs [11].

Our contribution in this work is the study of the primal dual algorithm for the hypergraph distributed optimization problem and its interpretation from a dynamical system perspective. We prove that the equilibrium point of the resulting primal dual dynamical system is the optimal solution of the hypergraph distributed optimization problem and we show its convergence with the use of non linear control theoretic techniques from passivity and Lyapunov theory. We utilize in the decomposition process of the communication matrix the fact that hypergraph Laplacian in our given information structure is also a projection matrix.

Finally, we will present a formula for a Laplacian matrix in the case of a directed weighted hypergraph by introducing a new type of incidence matrix. We will also show how this new Laplacian matrix can be decomposed in the stability analysis setting that we will present for the unweighted/undirected case. Various Laplacian matrices have been proposed in the literature for the cases of weighted and/or oriented hypergraphs such as in [16] but in most cases these matrices are constructed for specific case studies and they do not satisfy many of the usual properties of a Laplacian matrix as these are presented in the preliminaries section.

The paper is organized as follows: Sect. 2 presents the mathematical tools that will be used from non linear control theory [2], hypergraph theory [3], optimization theory [1] and matrix theory [4]. Section 3 introduces the hypergraph distributed optimization problem while in Sect. 4 we present the primal dual algorithm along with the respective equilibrium and convergence proofs. Finally, in Sect. 5 we present a Laplacian matrix formula for a directed weighted hypergraph and we show how this matrix is decomposed for the stability analysis setting that we studied previously. Numerical examples are provided alongside with the evolution of the theory.

2 Preliminaries

2.1 Notation

The set of real numbers is \mathbb{R} . For $x \in \mathbb{R}^n$, $x \geq 0$ ($x > 0$) means that all components of x are nonnegative (positive). We use $\|\cdot\|_2$ to denote the Euclidean norm in \mathbb{R}^n . We use $|C|$ to denote the cardinality of set C .

2.2 Non Linear Control Theory

We will study a continuous, autonomous nonlinear system

$$\dot{x}(t) = f(x(t)) \tag{1}$$

where $x(t) \in \mathbb{R}^n$ and $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is continuous. A function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ that satisfies $\|x\| \rightarrow \infty \Rightarrow V(x) \rightarrow \infty$ is called radially unbounded. The function $\dot{V} : \mathbb{R}^n \rightarrow \mathbb{R}$ is called the Lie derivative of V and is defined as:

$$\dot{V}(x(t)) = \nabla V(x(t))^T \cdot \dot{x}(t) = \nabla V(x(t))^T \cdot f(x(t)).$$

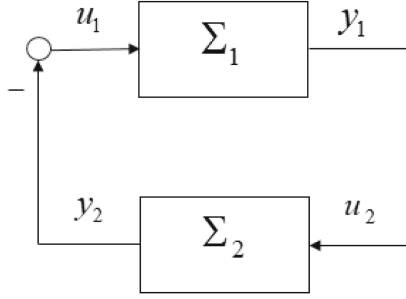


Fig. 1. Negative Feedback Interconnection of Systems.

Theorem 1. *Let x^* be an equilibrium point of (1). If $V : \mathbb{R}^n \rightarrow \mathbb{R}$ is radially unbounded and $\dot{V}(x) < 0, \forall x \neq x^*$ then x^* is globally asymptotically stable and V is a valid Lyapunov function for (1).*

Definition 1. *Consider a system Σ , with the following state space expression*

$$\dot{x}(t) = f(x(t), u(t))$$

$$y(t) = h(x(t))$$

with state $x(t) \in \mathbb{R}^n$, input $u(t) \in \mathbb{R}^n$, output $y(t) \in \mathbb{R}^n$ and $f, h : \mathbb{R}^n \rightarrow \mathbb{R}^n$. The system Σ is said to be passive if there exists a function $S : \mathbb{R}^n \rightarrow \mathbb{R}_+ := (0, \infty)$, called storage function, such that

$$\dot{S}(x(t)) \leq u^T(t)y(t) \quad (2)$$

holds for all states $x(t) \in \mathbb{R}^n$, all inputs $u(t) \in \mathbb{R}^n$ and all outputs $y(t) \in \mathbb{R}^n$. In the case that we have:

$$\dot{S}(x(t)) \leq u^T(t)y(t) - \psi(x(t)) \quad (3)$$

for some positive definite function ψ then the system is called strictly passive.

Theorem 2. *The negative feedback interconnection of a passive and a strictly passive system is a stable system (Fig. 1).*

Definition 2. *A domain $\mathcal{D} \subseteq \mathbb{R}^n$ is called invariant for the system $\dot{x} = f(x)$ if*

$$\forall x(t_0) \in \mathcal{D} \Rightarrow x(t) \in \mathcal{D}, \forall t \in \mathbb{R}.$$

Definition 3. *A domain $\mathcal{D} \subseteq \mathbb{R}^n$ is called positively invariant for the system $\dot{x} = f(x)$, if*

$$\forall x(t_0) \in \mathcal{D} \Rightarrow x(t) \in \mathcal{D}, \forall t \geq t_0.$$

Theorem 3 (LaSalle’s Invariance Principle). *Let $\Omega \subset D$ be a compact positively invariant set with respect to $\dot{x} = f(x)$. Let $V : D \rightarrow \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(x) \leq 0$ in Ω . Let \mathcal{X} be the set of all points in Ω where $\dot{V}(x) = 0$. Let M be the largest invariant set in \mathcal{X} . Then every solution starting in Ω approaches M as $t \rightarrow \infty$.*

Lemma 1. *Consider a convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Then, its gradient $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is incrementally passive, i.e., the following inequality holds for any $x, y \in \mathbb{R}^n$,*

$$(\nabla f(x) - \nabla f(y))^T(x - y) \geq 0.$$

If f is strictly convex, the inequality strictly holds as long as $x \neq y$. In that case, ∇f is strictly incrementally passive.

Lemma 2. *A function $f : A \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ is locally Lipschitz if for each $x, x_0 \in A$, there exist constant $M > 0$ and $\delta_0 > 0$ such that $\|x - x_0\| < \delta_0 \Rightarrow \|f(x) - f(x_0)\| \leq M\|x - x_0\|$.*

2.3 Hypergraphs

A hypergraph is a pair $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_1, \dots, v_n\}$ is a finite set of nodes and $\mathcal{E} = \{\mathcal{E}_1, \dots, \mathcal{E}_m\}$ is the set of hyperedges. Each hyperedge can join any number of nodes and not just two as it is in the case of a graph. A hypergraph \mathcal{H} is connected if there is a path from any node to any other node in \mathcal{H} . The degree of a node v_i denoted by $|v_i|$ is the total number of hyperedges adjacent to this node and the degree of a hyperedge \mathcal{E}_j denoted by $|\mathcal{E}_j|$ is the total number of nodes adjacent to this hyperedge. We define by D_V the diagonal $|\mathcal{V}| \times |\mathcal{V}|$ matrix whose entries are the degrees of each node, i.e., $D_V = \text{diag}\{|v_1|, \dots, |v_n|\}$ and by D_E the diagonal $|\mathcal{E}| \times |\mathcal{E}|$ matrix whose entries are the degrees of each hyperedge, i.e., $D_E = \text{diag}\{|\mathcal{E}_1|, \dots, |\mathcal{E}_m|\}$. For a hypergraph \mathcal{H} , the incidence matrix, denoted by E is a $|\mathcal{V}| \times |\mathcal{E}|$ matrix whose (i, j) -th entry is defined as:

$$E = \begin{cases} 1, & v_i \in \mathcal{E}_j \\ 0, & \text{otherwise.} \end{cases}$$

The hypergraph Laplacian (Bolla’s Laplacian) Q is a $|\mathcal{V}| \times |\mathcal{V}|$ matrix given by the formula:

$$Q = D_V - ED_E^{-1}E^T.$$

2.4 Optimization Theory

For the following equality constrained optimization problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t.} & Ax = b \end{aligned} \tag{4}$$

x is the decision variable, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function, $A \in \mathbb{R}^{r \times n}$ and $b \in \mathbb{R}^r$ are the constant matrix and vector for equality constraints respectively. We define the Lagrange function of (4) as $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}$ given by:

$$\mathcal{L}(x, v) = f(x) + v^T(Ax - b)$$

where $x \in \mathbb{R}^n$ is the primal variable and $v \in \mathbb{R}^r$ is the dual variable. The primal-dual dynamics as a solution to (4) are given by:

$$\dot{x} = -\nabla f(x) - A^T v$$

$$\dot{v} = Ax - b$$

and the set of the optimality solutions is defined as

$$X^* := \{(x^*, v^*) \in \mathbb{R}^n \times \mathbb{R}^r\}$$

where x^* and v^* satisfy the following equations

$$\nabla f(x^*) + A^T v^* = 0, \quad Ax^* - b = 0.$$

2.5 Matrix Theory

For a matrix $M \in \mathbb{R}^{n \times n}$ by M^T and $S(M)$ we denote its transpose and its spectrum respectively. Matrix $P \in \mathbb{R}^{n \times n}$ is an orthogonal projection matrix when $P^2 = P = P^T$ with spectrum $S(P) = \{0, 1\}$. A symmetric matrix M is called positive (semi)definite $M(\succeq) \succ 0$ if and only if $x^T M x(\geq) > 0$ for every nonzero $x \in \mathbb{R}^n$. The Laplacian matrix L of a connected graphical structure is symmetric ($L = L^T$), positive semidefinite, M -matrix (its off-diagonal elements are nonpositive), every row and column sum to zero and its eigenvalues satisfy the following inequalities $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$. The eigendecomposition of a positive semi definite matrix M is $M = C^T B C$ where B is the diagonal matrix with the eigenvalues of matrix M and C is the matrix with the respective eigenvectors. The square root of matrix M is $M^{1/2} = C^T B^{1/2} C$ where $B^{1/2}$ is the diagonal matrix with the respective square roots of the eigenvalues of M .

3 The Hypergraph Distributed Optimization Problem

We have the following hypergraph distributed optimization problem for K subsystems with hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ describing the coupling variables of the different subsystems,

$$\begin{aligned} \min_{x=[x_1, \dots, x_K]^T} & \sum_{i=1}^K f_i(x_i) \\ \text{s.t.} & \quad Qx = 0. \end{aligned} \tag{5}$$

where

- $f_i : \mathbb{R}^{p_i} \rightarrow \mathbb{R}$ is the objective function of i th subsystem and is considered to be strictly convex, continuously differentiable with its gradient ∇f_i being locally Lipschitz.
- Vectors $x_i \in \mathbb{R}^{p_i}, \forall 1 \leq i \leq K$ denote the variables of the subsystems which we assume are coupling (i.e. there are common components among different variables). We assume that the total number of common values for the coupling components is N .
- The node set \mathcal{H} is partitioned into $\mathcal{V} = \{\mathcal{V}_1, \dots, \mathcal{V}_K\}$ where each node in subset \mathcal{V}_i is associated with a component of variable x_i .
- Each hyperedge $\mathcal{E}_j \forall 1 \leq j \leq N$ is associated with the j th common value of coupling variable components. We assume that each node of \mathcal{H} can be adjacent to only one hyperedge.
- Matrix Q is the hypergraph Laplacian matrix and in our information structure it can also be expressed as

$$Q = I - E(E^T E)^{-1} E^T$$

since $D_V = I$ and $D_E = E^T E$. Matrix Q allocates the coupling variable components of the different subsystems to their respective common values.

For the hypergraph \mathcal{H} we have,

$$|\mathcal{V}| = p_1 + \dots + p_K = p, |\mathcal{E}| = N$$

$$D_V = I_{p \times p}, D_E = \text{diag}\{|\mathcal{E}_1|, \dots, |\mathcal{E}_N|\} \text{ and } E = \begin{bmatrix} E_1 \\ \vdots \\ E_K \end{bmatrix}, \text{ where}$$

E_i is a $p_i \times N$ matrix whose (l, j) -th entry is given by

$$E_i^{lj} = \begin{cases} 1, & \text{if } x_i^l = \mathcal{E}_j, \forall 1 \leq l \leq p_i, \forall 1 \leq j \leq N \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

with x_i^l denoting the l th component of variable x_i .

Lemma 3. *The hypergraph Laplacian matrix Q in (5) is also an orthogonal projection matrix.*

Proof. We have that

$$\begin{aligned} Q^2 &= [I - E(E^T E)^{-1} E^T]^2 \\ &= [I - E(E^T E)^{-1} E^T][I - E(E^T E)^{-1} E^T] \\ &= I - 2E(E^T E)^{-1} E^T + E(E^T E)^{-1} E^T E(E^T E)^{-1} E^T \\ &= I - 2E(E^T E)^{-1} E^T + E(E^T E)^{-1} E^T \\ &= I - E(E^T E)^{-1} E^T \\ &= Q \end{aligned}$$

and also $Q = Q^T$. As a result, Q is an orthogonal projection matrix.

Example 1. We have the following sum of objective functions

$$f_1(x_1^1) + f_2(x_2^1) + f_3(x_3^1, x_3^2) + f_4(x_4^2)$$

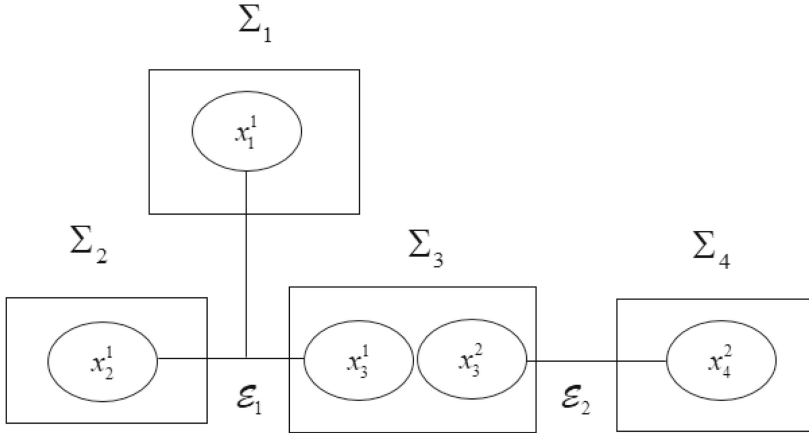


Fig. 2. Hypergraph Communication.

with two coupling variable components that have the hypergraph representation of Fig. 2. The nodes associated with variables $\{x_1^1, x_2^1, x_3^1\}$ are attached to hyperedge \mathcal{E}_1 while the nodes associated with variables $\{x_3^2, x_4^2\}$ are attached to hyperedge \mathcal{E}_2 . We also have that

$$|\mathcal{V}| = 5, |\mathcal{E}| = 2, D_V = I_{5 \times 5}, D_E = \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix} \text{ and } E = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \end{bmatrix} = \begin{pmatrix} [1 & 0] \\ [1 & 0] \\ [1 & 0] \\ [0 & 1] \\ [0 & 1] \end{pmatrix}$$

respectively. The hypergraph Laplacian matrix Q is

$$Q = I - E(D_E)^{-1}E^T = \begin{pmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 0 & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

4 Primal Dual Algorithm

In this section we propose a primal dual algorithm for the distributed optimization problem (5). We define the Lagrangian of (5) to be

$$\mathcal{L}(x, v) = f(x) - v^T Qx$$

where $f(x) = \sum_{i=1}^K f_i(x_i)$ and $v \in \mathbb{R}^p$ are the respective dual variables. The primal dual dynamics of (5) are:

$$\dot{x}(t) = -\nabla \mathcal{L}_x = -\nabla f(x(t)) + Qv(t) \tag{8a}$$

$$\dot{v}(t) = \nabla \mathcal{L}_v = -Qx(t). \tag{8b}$$

Theorem 4. *Let (x^*, v^*) be an equilibrium point of the dynamical system (8a)–(8b) then (x^*, v^*) satisfies the optimality conditions of (5).*

Proof. We find the equilibrium point of the dynamical system from $\dot{x}(t) = 0 \Rightarrow \nabla f(x^*) = Qv^*$ and $\dot{v}(t) = 0 \Rightarrow Qx^* = 0$. We notice that the equilibrium point satisfies the optimality conditions as they were presented in the preliminaries section and as a result, the equilibrium point of the dynamical system (8a)–(8b) solves the optimization problem (5).

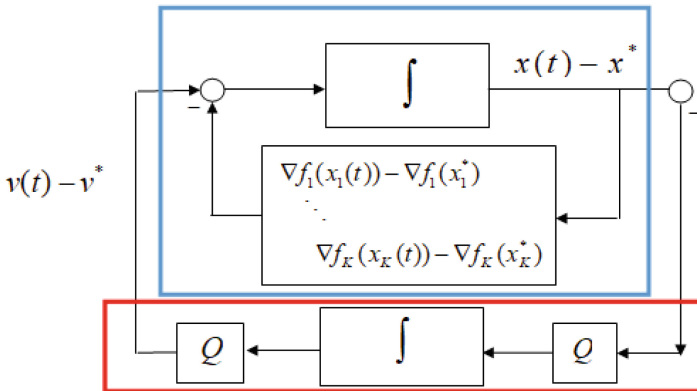


Fig. 3. Primal Dual Decomposition.

Theorem 5. *The dynamical system (8a)–(8b) is a negative feedback interconnection of a passive and a strictly passive system.*

Proof. The dynamical system (8a)–(8b) can be seen as a negative feedback interconnection of a passive and a strictly passive system in Fig. 3.

The first system which is enclosed by the blue parallelogram refers to the primal dynamics and is a negative feedback interconnection of two systems where the first system is the system of non-linearities which is $\nabla f(x(t)) = [\nabla f_1(x_1(t)) - \nabla f_1(x_1^*), \dots, \nabla f_K(x_K(t)) - \nabla f_K(x_K^*)]$. Each $\nabla f_i(x_i(t)), i = 1, \dots, K$

is strictly incrementally passive and as a result, the system of the non-linearities is strictly passive. The other system is an integrator system and in order to be passive there must exist a storage function $S_1(t) : \mathbb{R}^p \rightarrow \mathbb{R}$ such that $\dot{S}_1(t) \leq u_1^T(t)y_1(t)$ where $u_1(t) = -(\nabla f(x(t)) - \nabla f(x^*)) + Q(v(t) - v^*)$ and $y_1(t) = x(t) - x^*$. We define the storage function to be $S_1(t) = \frac{1}{2}\|x(t) - x^*\|^2$ where $S_1(t) : \mathbb{R}^p \rightarrow \mathbb{R}$. The Lie derivative of the storage function is $\dot{S}_1(t) = \dot{x}(t)^T(x(t) - x^*) = [-(\nabla f(x(t)) - \nabla f(x^*))^T + Q(v(t) - v^*)](x(t) - x^*) \leq u_1(t)y_1(t)$ where we have used that $\nabla f(x^*) = Qv^*$ and as a result, the primal dynamics integrator system is passive. The overall system enclosed by the blue parallelogram is strictly passive.

The second system which is enclosed by the red parallelogram refers to the dual dynamics. The system is an integrator which is premultiplied and post-multiplied by Q . This pre/post multiplication preserves passivity since matrix $Q^2 = Q$ is positive semidefinite as an orthogonal projection matrix. For the integrator system we have $u_2(t) = -Q(x(t) - x^*)$ and $y_2(t) = Q(v(t) - v^*)$. In order for this system to be passive there must exist a storage function $S_2(t) : \mathbb{R}^p \rightarrow \mathbb{R}$ such that $\dot{S}_2(t) \leq u_2^T(t)y_2(t) = -(x(t) - x^*)^T Q^T Q(v(t) - v^*) = -(x(t) - x^*)^T Q(v(t) - v^*)$ since $Q^T Q = Q^2 = Q$. We define the storage function to be $S_2(t) = \frac{1}{2}\|v(t) - v^*\|^2$ where $S_2(t) : \mathbb{R}^p \rightarrow \mathbb{R}$. The Lie derivative of the storage functions is $\dot{S}_2(t) = \dot{v}(t)^T v(t) = -(Qx(t))^T(v(t) - v^*) = -(Qx(t) - Qx^*)^T(v(t) - v^*) = -(x(t) - x^*)^T Q(v(t) - v^*)$ where we have used that $Qx^* = 0$. As a result, $\dot{S}_2(t) = -(x(t) - x^*)^T Q(v(t) - v^*) \leq u_2^T(t)y_2(t)$ and the second system is passive. In conclusion we have a passive and a strictly passive system interconnected in negative feedback.

Theorem 6. *The equilibrium point of the dynamical system (8a)–(8b) is globally asymptotically stable.*

Proof. The Lyapunov function of the dynamical system (8a)–(8b) can be constructed as the sum of the respective storage functions. We choose as candidate Lyapunov function $V(t) : \mathbb{R}^p \rightarrow \mathbb{R}$ where $V(t) = \frac{1}{2}\|x(t) - x^*\|_2^2 + \frac{1}{2}\|v(t) - v^*\|_2^2$. The respective Lie derivative is

$$\begin{aligned}
\dot{V}(t) &= \dot{x}^T(t)(x(t) - x^*) + \dot{v}^T(t)(v(t) - v^*) \\
&= (-\nabla f(x(t)) + Qv(t))^T(x(t) - x^*) + (-Qx(t))^T(v(t) - v^*) \\
&= -\nabla f(x(t))^T(x(t) - x^*) + v^T(t)Q(x(t) - x^*) - x^T(t)Q(v(t) - v^*) \\
&= -\nabla f(x(t))^T(x(t) - x^*) + v^T(t)Qx(t) - v^T(t)Qx^* - x^T(t)Qv(t) + \\
&\quad + x^T(t)Qv^* \\
&= -\nabla f(x(t))^T(x(t) - x^*) - v^T(t)Qx^* + x^T(t)Qv^* \\
&= -\nabla f(x(t))^T(x(t) - x^*) + \nabla f(x^*)^T x(t) - \nabla f(x^*)^T x^* \\
&= -(\nabla f(x(t)) - \nabla f(x^*))^T(x(t) - x^*) < 0, \quad \forall x \neq x^*
\end{aligned}$$

since $\nabla f(x)$ is strictly incrementally passive. From LaSalle's invariance principle we have convergence to the largest invariant set for which $\dot{V} = 0$, i.e. $x = x^*$.

This set includes only the equilibrium point (x^*, v^*) . As a result, the dynamical system (8a)–(8b) converges asymptotically to the solution (x^*, v^*) . In the fifth line of the proof we have used that $v^T(t)Qx(t) = x^T(t)Qv(t)$ while in the sixth line of the proof we have used that $v^T(t)Qx^* = 0, x^T(t)Qv^* = \nabla f(x^*)^T x(t)$ and $\nabla f(x^*)^T x^* = 0$ from the equilibrium properties.

4.1 Directed Weighted Laplacian

The stability methodology and results of our previous study can also be extended for the case of a distributed optimization problem that uses a weighted and directed hypergraph. The only difference of such a problem with the case that we studied previously would be the structure of the hypergraph Laplacian matrix. We define the Laplacian matrix of a directed and weighted hypergraph as

$$Q' = ZWZ^T \tag{10}$$

with W being the positive definite diagonal $p \times p$ weight matrix with $p = p_1 + \dots + p_K$ representing the sum of the respective dimensions of the K subsystems and the matrix Z with dimensions $p \times N$ to be an incidence matrix of the form

$$Z = [Z_1, \dots, Z_N]$$

where

$$Z_i = D_{\text{in}}^i - D_{\text{out}}^i, \forall 1 \leq i \leq N$$

with $D_{\text{in}}^i, D_{\text{out}}^i$ being the $p \times 1$ in-degree and out-degree vectors of i th hyperedge respectively. For the unweighted case, (10) becomes

$$Q' = ZD_E^{-1}Z^T$$

where D_E is the hyperedge degree matrix. A suitable decomposition of (10) for the stability analysis that we conducted throughout this work would be $Q' = AA^T$ where $A = ZW^{1/2}$ with $W^{1/2}$ being the square root of matrix W .

Remark 1. The proposed Laplacian matrix in (10) satisfies all the properties of a Laplacian matrix presented in the introduction except of the M -matrix property which is natural since the hyperedges may have multiple directions.

Example 2. Given the unweighted directed hypergraph of Fig. 4 we have

$$Z_1 = \begin{pmatrix} -2 \\ 1 \\ 1 \\ 0 \end{pmatrix}, Z_2 = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 1 \end{pmatrix}, Z = \begin{pmatrix} -2 & 0 \\ 1 & 0 \\ 1 & -1 \\ 0 & 1 \end{pmatrix} \text{ and } D_E = \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix}.$$

As a result, the directed hypergraph Laplacian is

$$Q' = ZD_E^{-1}Z^T = \begin{pmatrix} \frac{4}{3} & \frac{-2}{3} & \frac{-2}{3} & 0 \\ \frac{-2}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{5}{3} & \frac{-1}{2} \\ 0 & 0 & \frac{-1}{2} & \frac{1}{2} \end{pmatrix}.$$

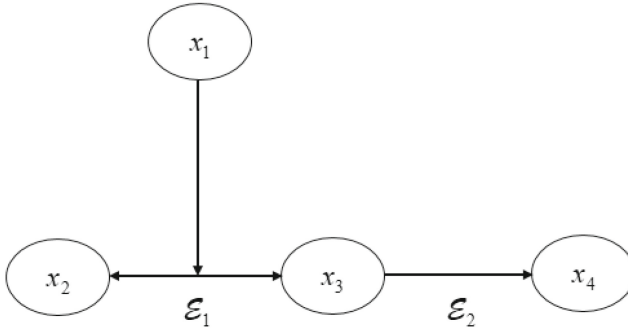


Fig. 4. Directed Hypergraph.

5 Conclusion

We have studied the primal dual algorithm of a distributed optimization problem that uses an undirected and unweighted hypergraph as its communication structure from a dynamical system approach. We proved the stability of this dynamical system with the use of non linear control theory and an appropriate decomposition of the respective hypergraph Laplacian matrix. Finally, we have extended our study for the case of a weighted and directed hypergraph where we presented a Laplacian matrix for this communication structure and we proposed a decomposition of this matrix for the stability analysis setting studied throughout this work.

References

1. Stephen, P.: Boyd and Lieven Vandenberghe. Cambridge University Press, Convex optimization (2004)
2. Khalil, H., Grizzle, J.W.: Nonlinear Systems, vol. 3. Prentice hall, Upper Saddle River, NJ (2002)
3. Voloshin, V.I.: Introduction to Graph and Hypergraph Theory. Nova Science Publication (2009)
4. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press (2012)
5. Tsitsiklis, J.: Problems in decentralized decision making and computation, Doctoral dissertation, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems (1984)
6. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods, 2nd. (1989)
7. Claude, B.: Graphs and Hypergraphs (1973)
8. Heintz, B., Hong, R., Singh, S., Khandelwal, G., Tesdahl, C., Chandra, A.: MESH: a flexible distributed hypergraph processing system. In: 2019 IEEE International Conference on Cloud Engineering (IC2E), pp. 12–22. IEEE(2019)
9. Wolf, M.M., Klinvex, A.M., Dunlavy, D.M.: Advantages to modeling relational data using hypergraphs versus graphs. In: 2016 IEEE High Performance Extreme Computing Conference (HPEC). IEEE (2016)

10. Heintz, B., Chandra, A.: Beyond graphs: toward scalable hypergraph analysis systems. *ACM SIGMETRICS Performance Eval. Rev.* **41**(4), 94–97 (2014)
11. Bolla, M.: Spectra, euclidean representations and clusterings of hypergraphs. *Discrete Math.* **117**, 19–39 (1993)
12. Kelly, F.P., Maulloo, A.K., Tan, D.K.H.: Rate control for communication networks: shadow prices, proportional fairness and stability. *J. Oper. Res. Soc.* **49**(3), 237–252 (1998)
13. Nedic, A.: Distributed gradient methods for convex machine learning problems in networks: distributed optimization. *IEEE Signal Process. Mag.* **37**(3), 92–101 (2020)
14. Yang, T., et al.: A survey of distributed optimization. *Ann. Rev. Control* **47**, 278–305 (2019)
15. Kia, S.S., Van Scoy, B., Cortes, J., Freeman, R.A., Lynch, K.M., Martinez, S.: Tutorial on dynamic average consensus: the problem, its applications, and the algorithms. *IEEE Control Syst. Mag.* **39**(3), 40–72 (2019)
16. Jost, J., Raffaella, M.: Normalized Laplace operators for hypergraphs with real coefficients. *J. Complex Networks* **9.1**, cnab009 (2021)



Topic-Based Analysis of Structural Transitions of Temporal Hypergraphs Derived from Recipe Sharing Sites

Keisuke Uga, Masahito Kumano, and Masahiro Kimura^(✉)

Faculty of Advanced Science and Technology, Ryukoku University, Otsu, Japan
kimura@rins.ryukoku.ac.jp

Abstract. We analyze a recipe stream created on a social media site dedicated to sharing homemade recipes in terms of a temporal hypergraph over a set of ingredients. Unlike the previous studies for transition analysis of temporal higher-order networks, we propose a novel analysis method based on topics and projected graphs to effectively characterize the structural transitions of the temporal hypergraph immediately before and after the occurrences of hyperedges. First, we propose a probabilistic model to extract the topics of hyperedges on the basis of the trends and seasonality of recipes, and present its Bayesian inference method. Next, we propose employing the projected graph of the entire hypergraph, and examining whether each of its main edges is present or not in the temporal hypergraph, both immediately before and after the occurrences of hyperedges for each topic. Using real data of a Japanese recipe sharing site, we empirically demonstrate the effectiveness of the proposed analysis method, and reveal several interesting properties in the evolution of Japanese homemade recipes.

Keywords: ingredient network · social media analysis · temporal higher-order network · transition analysis

1 Introduction

The growing popularity of social media sites dedicated to sharing cooking recipes has opened up an opportunity to explore the evolution of creative homemade recipes made by ordinary people. Recently, there has been an increasing interest in food science and computing [15], leading to the use of network science methods [2] to analyze ingredient co-occurrence properties in recipes [1, 9, 18]. Networks offer a fundamental tool of modeling complex systems, and have been successfully applied in various fields including social media analysis. Traditional network-based models use graphs with nodes representing basic elements and edges encoding their pairwise interactions. However, in many real-world settings such as a human interaction in a group and a combination of ingredients in a recipe, it becomes crucial to analyze interactions among more than two elements. Thus, attention has recently been devoted to an analysis of higher-order networks [3, 4, 20].

From the perspective of link prediction, many studies have been conducted on the evolution of traditional dyadic networks [13]; however, little is known about the evolution of higher-order networks since their analysis can be computationally challenging. Benson et al. [4] have presented a framework to examine the evolution of higher-order networks, focusing on *simplicial closure*, which is a unique phenomenon of higher-order structures not captured by conventional network analysis such as *triadic closure*. On the other hand, Cencetti et al. [6] have analyzed the configuration transitions for the evolution of higher-order networks of human proximity interactions, unlike link prediction tasks such as simplicial closure. More specifically, they examined the transitions of configurations around higher-order links of a given size immediately before and after their occurrences, and found several interesting properties in five different social settings. Fujisawa et al. [8] have extended the work of Cencetti et al. [6] in the case of temporal *simplicial complexes*, and presented an effective framework for analyzing the transitions of *boundary-based active configurations* around new simplices of a given dimension immediately before and after their occurrences. However, these previous studies basically restricted their transition analysis to higher-order links of a given size that is relatively small. Thus, it is desirable to develop an effective framework for transition analysis that is independent of the size of target higher-order links.

In this paper, we investigate a recipe stream generated on a social media site dedicated to sharing homemade recipes in terms of a temporal hypergraph on a set of ingredients. To effectively characterize the structural transitions of this temporal hypergraph immediately before and after the occurrences of hyperedges (i.e., recipes), we propose a novel method that leverages topics and projected graphs. To extract the topics of hyperedges that relate to the trends and seasonality of recipes, we first propose a probabilistic model and develop its Bayesian inference method. Next, as an effective characterization framework that is independent of the size of generated hyperedges, we focus on the projected graph of the entire hypergraph and propose examining whether each of its main edges is present or not in the temporal hypergraph, both immediately before and after the occurrences of hyperedges for each topic. Using real data of a Japanese recipe sharing site, we empirically evaluate the proposed probabilistic model, and explore the characteristics in the evolution of Japanese homemade recipes by applying the proposed analysis method.

2 Related Work

Recently, there has been a lot of research in the field of food science and computing, leading to a wide range of food-oriented applications being explored [15]. From the perspective of network science, several researchers have analyzed ingredient networks based on flavor compounds [1, 9, 14, 16]. Also, Kikuchi et al. [12] have analyzed the dynamic changes in ingredient pairs jointly used together in recipes by leveraging temporal ingredient networks. Other studies have examined population-wide dietary preferences through recipe queries on the Web [21], and

investigated cuisines and culinary habits in the world, considering ingredients, flavors, and nutritional values derived from large-scale online recipe data [17]. For cross-region recipe analysis, visualizations of recipe density and ingredient categories have allowed for comparisons of food cultures from around the world [10]. Concerning recipe recommendation, researchers have explored the use of complement and substitute networks for ingredients [18], and discussed the relationship between algorithmic solutions and recipe healthiness [19]. In this paper, we explore the characteristics in the evolution of Japanese homemade recipes through an analysis of the structural transitions of temporal hypergraphs derived from recipe sharing sites.

This paper also has a relationship with probabilistic topic models for network generation. For the generation of traditional networks (i.e., graphs), a variety of studies on probabilistic topic models, such as stochastic blockmodels, have been conducted. These models have been successfully applied to network clustering (see e.g., [11]). On the other hand, there has been a small number of studies that have devised probabilistic topic models for generating clustered hypergraphs and successfully applied them to hypergraph clustering (see e.g., [7]). Note that these studies partition the set of nodes in a hypergraph by assigning a topic to each node. In contrast, in this paper, we employ a probabilistic topic model to partition the set of hyperedges in a hypergraph by assigning a topic to each hyperedge, considering the trends and seasonality of recipes as well.

3 Preliminaries

3.1 Temporal Hypergraphs of Recipe Streams

We focus on a social media site dedicated to sharing homemade recipes, where active interactions among users are performed and these interactions can promote the creation of better recipes. We analyze the characteristics of recipe streams generated from the social media site, where each recipe stream consists of recipes with time-stamps.

From a perspective of a temporal hypergraph $\{H_t = (V, \mathcal{H}_t)\}_{t \in \mathcal{T}}$, we investigate a recipe stream \mathcal{R} during a time-span \mathcal{T} , where the day is used as our time unit. We fix a set of main ingredients V , and refer to each element of V as a *node* in hypergraph H_t for any $t \in \mathcal{T}$. We set

$$V = \{v_1, \dots, v_N\}, \quad (1)$$

where N is the number of main ingredients. For each recipe $r \in \mathcal{R}$, let $\tau(r)$ denote its time-stamp, meaning that recipe r is published on the site at time $\tau(r)$. Using the cooking procedure of recipe r in terms of V , we first express r as a sequence of nodes $\langle w_1(r), \dots, w_{n(r)}(r) \rangle$, where $w_1(r), \dots, w_{n(r)}(r) \in V$ and $n(r)$ is an integer more than one. Note that there might be two nodes $w_i(r)$ and $w_j(r)$ ($1 \leq i, j \leq n(r)$, $i \neq j$) such that $w_i(r) = w_j(r)$. Then, we identify recipe r as an unordered node tuple with repeated nodes, $[w_1(r), \dots, w_{n(r)}(r)]$, and represent r as an N -dimensional vector

$$h(r) = (h_1(r), \dots, h_N(r)), \quad (2)$$

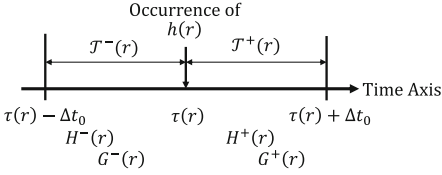


Fig. 1. Transition analysis of temporal hypergraph $\{H_t\}$ for a hyperedge $h(r)$

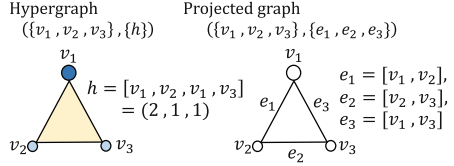


Fig. 2. Example of the projected graph of a hypergraph.

where each $h_i(r)$ is the number of times that node v_i appears in $\langle w_1(r), \dots, w_{n(r)}(r) \rangle$ (see Eq. (1)). Note that

$$h_1(r) + \dots + h_N(r) = n(r).$$

Let $n_1(r), \dots, n_{m(r)}(r)$ denote the nonzero components of vector $h(r)$, where $n_1(r) < \dots < n_{m(r)}(r)$. Note that $m(r)$ is the number of nonzero components of vector $h(r)$. We refer to $h(r)$ as an $m(r)$ -hyperedge that is placed on node set $\{v_{n_1(r)}, \dots, v_{n_{m(r)}(r)}\}$ with multiplicity $(h_{n_1(r)}(r), \dots, h_{n_{m(r)}(r)}(r))$ and that is generated at time $\tau(r)$. Let \mathcal{H}_t denote the set of all hyperedges generated at time $t \in \mathcal{T}$. Note that $h(r) \in \mathcal{H}_{\tau(r)}$. We say that $\{H_t = (V, \mathcal{H}_t)\}_{t \in \mathcal{T}}$ is the temporal hypergraph derived from recipe stream \mathcal{R} during time-span \mathcal{T} .

3.2 Structural Transitions of Temporal Hypergraphs

We consider the structural transition of the temporal hypergraph $\{H_t = (V, \mathcal{H}_t)\}$ before and after the occurrence of each hyperedge $h(r)$ (see Fig. 1). We focus on a time-span $\mathcal{T}^-(r)$ immediately before the occurrence of $h(r)$ and a time-span $\mathcal{T}^+(r)$ immediately after it, where

$$\mathcal{T}^-(r) = [\tau(r) - \Delta t_0, \tau(r)) \subset \mathcal{T}, \quad \mathcal{T}^+(r) = (\tau(r), \tau(r) + \Delta t_0] \subset \mathcal{T}, \quad (3)$$

and Δt_0 is a positive integer indicating the length of the “Before-After” investigation period. We examine the hypergraph $H^-(r) = (V, \mathcal{H}^-(r))$ immediately before the occurrence of $h(r)$ and the hypergraph $H^+(r) = (V, \mathcal{H}^+(r))$ immediately after it, which are defined as

$$\mathcal{H}^-(r) = \{h(r') \in \mathcal{H}_t \mid t \in \mathcal{T}^-(r)\}, \quad \mathcal{H}^+(r) = \{h(r') \in \mathcal{H}_t \mid t \in \mathcal{T}^+(r)\}.$$

Let $G^-(r) = (V, E^-(r))$ and $G^+(r) = (V, E^+(r))$ denote the projected graphs of hypergraphs $H^-(r)$ and $H^+(r)$, respectively. Here, $G' = (V', E')$ is referred to as the projected graph of a hypergraph $H' = (V', \mathcal{H}')$, when each undirected edge $e' = [u', v']$ in G' ($u', v' \in V'$) is derived from the projection of some hyperedge h' in H' , i.e., $e' \subset h'$, and the projections of any hyperedge in H' are edges in G' (see Fig. 2). Note that G' represents the basic graph structure inherent in H' .

For a relatively small positive integer $m \geq 3$, Cencetti et al. [6] and Fujisawa et al. [8] dealt with configurations around each m -hyperedge created, and

examined the characteristics of the configuration transitions before and after the occurrences of those m -hyperedges. Unlike these previous studies, in this paper, we focus on the projected graphs $G^-(r)$ and $G^+(r)$ for analyzing the structural transition of $\{H_t\}$ before and after the occurrence of $h(r)$. Moreover, we investigate the structural transition of $\{H_t\}$ from the perspective of the topics of created hyperedges (i.e., the topics of created recipes), rather than based on the size m of the created hyperedges.

4 Analysis Method

We consider the temporal hypergraph $\{H_t = (V, \mathcal{H}_t)\}_{t \in \mathcal{T}}$ derived from recipe stream \mathcal{R} . Let $\mathcal{T}_* \subset \mathcal{T}$ be a time-span to be analyzed. We focus on the set of hyperedges created within \mathcal{T}_* , $\mathcal{H}_* = \{h(r) \in \mathcal{H}_t \mid t \in \mathcal{T}_*\}$. We explore the structural transition of $\{H_t\}_{t \in \mathcal{T}}$ immediately before and after the occurrences of hyperedges in \mathcal{H}_* from the perspectives of topics and projected graphs.

4.1 Extraction of Topics

First, we extract the topics of hyperedges from \mathcal{H}_* . To this end, we simply assume a Dirichlet mixture of multinomial distributions over V as a probabilistic model generating hyperedges with topics. To take into account the trends and seasonality of recipes (i.e., hyperedges) in time-span \mathcal{T} and automatically estimate the number of topics, we propose incorporating a *distance dependent Chinese restaurant process (ddCRP)* by Blei and Frazier [5].

The proposed probabilistic model generates each hyperedge $h(r)$ (see Eq. (2)) in the following way: First, a collection of topic assignments $Z = (z(r))_{r \in \mathcal{R}}$ is drawn from a ddCRP, where a positive integer $z(r)$ represents the topic of hyperedge $h(r)$ (i.e., the topic of recipe r). More specifically, for each recipe $r \in \mathcal{R}$, the ddCRP independently generates a recipe assignment $c_r \in \mathcal{R}$ from the probability distribution

$$p(c_r \mid \Delta t_1, \gamma) \propto \begin{cases} \mathcal{I}(|\tau(c_r) - \tau(r)| < \Delta t_1) & \text{if } c_r \neq r, \\ \gamma & \text{if } c_r = r, \end{cases} \quad (4)$$

where $\Delta t_1 > 0$ and $\gamma > 0$ are hyper-parameters, and $\mathcal{I}(q)$ is the indicator function of a proposition q such that $\mathcal{I}(q) = 1$ if q is true and $\mathcal{I}(q) = 0$ if q is false. For the collection of recipe assignments $C = (c_r)_{r \in \mathcal{R}}$, we consider the graph $\mathcal{G}(C)$ over \mathcal{R} that is determined by C . Then, $Z = Z(C)$ is the partition of \mathcal{R} that is derived from the connected component decomposition of the graph $\mathcal{G}(C)$. Let $K = K(Z)$ be the number of topics in Z , i.e., the number of different values in $\{z(r) \mid r \in \mathcal{R}\}$. Next, for each topic k , a multinomial parameter $\theta_k = (\theta_{k,1}, \dots, \theta_{k,N})$ is drawn from a Dirichlet distribution

$$p(\theta_k \mid \mu) = \frac{\Gamma\left(\sum_{i=1}^N \mu_i\right)}{\prod_{i=1}^N \Gamma(\mu_i)} \prod_{i=1}^N \{\theta_{k,i}\}^{\mu_i - 1}, \quad (5)$$

where $\boldsymbol{\mu} = (\mu_1, \dots, \mu_N)$, $\mu_1, \dots, \mu_N > 0$ are hyper-parameters, and $\Gamma(s)$ is the gamma function. We put $\Theta = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K)$. Finally, given the number $n(r)$ of main ingredients in recipe r , hyperedge $h(r)$ (see Eq. (2)) is generated according to the multinomial distribution

$$p(h(r) | Z(C), \Theta) = \frac{n(r)!}{\prod_{i=1}^N h_i(r)!} \prod_{i=1}^N \{\theta_{z(r),i}\}^{h_i(r)}. \quad (6)$$

We now consider extracting the collection of topics Z for the hyperedges in the observed data \mathcal{H}_* on the basis of a Bayesian inference framework. Then, the likelihood of \mathcal{H}_* is given by

$$p(\mathcal{H}_* | Z(C), \boldsymbol{\mu}) = \int_{\Omega} \prod_{h(r) \in \mathcal{H}_*} p(h(r) | Z(C), \Theta) \prod_{k=1}^{K(Z)} p(\boldsymbol{\theta}_k | \boldsymbol{\mu}) d\boldsymbol{\theta}_k, \quad (7)$$

where Ω stands for the appropriate domain of integration. Note that Eq. (7) is analytically calculated from Eqs. (5) and (6). We estimate the collection of recipe assignments C from \mathcal{H}_* by employing the Gibbs sampling

$$p(c_r^{\text{new}} | C^{-r}, \mathcal{H}_*, \Delta t_1, \gamma, \boldsymbol{\mu}) \propto p(c_r^{\text{new}} | \Delta t_1, \gamma) p(\mathcal{H}_* | Z(C^{-r} \cup c_r^{\text{new}}), \boldsymbol{\mu}) \quad (8)$$

for each recipe r , and consequently extract the collection of topic assignments $Z(C)$. Here, C^{-r} indicates removing the current assignment c_r of r from C , and $C^{-r} \cup c_r^{\text{new}}$ indicates adding a new assignment c_r^{new} of r to C^{-r} . Note that the Gibbs sampler (see Eq. (8)) can be efficiently calculated by using several properties of ddCRP (see [5]). Due to space constraints, we omit the details. We will provide a more comprehensive explanation of our inference method in the extended version of the paper.

4.2 Topic-Based Analysis of Structural Transitions

For each hyperedge $h(r) \in \mathcal{H}_*$, we investigate the structural transition of temporal hypergraph $\{H_t\}$ before and after the occurrence of $h(r)$.

To examine the structural transition of $\{H_t\}$ in terms of projected graphs, we first consider the hypergraph $H = (V, \mathcal{H})$ derived from \mathcal{R} during the entire time-span \mathcal{T} and its projected graph $G = (V, E)$, where $\mathcal{H} = \bigcup_{t \in \mathcal{T}} \mathcal{H}_t$. We introduce a set of main edges in E (see Sect. 5.3 for more details),

$$E^{\text{main}} = \{e_1, \dots, e_M\} \subset E, \quad (9)$$

where each e_j indicates a main pair of ingredients during \mathcal{T} . To examine the structural transition of $\{H_t\}$ in terms of topics, we also consider the topic decomposition of \mathcal{H}_* ,

$$\mathcal{H}_* = \mathcal{H}_{*,1} \cup \dots \cup \mathcal{H}_{*,K} \quad (\text{disjoint union}), \quad (10)$$

which is extracted by the method described in Sect. 4.1. Note that each $\mathcal{H}_{*,k}$ consists of the hyperedges (i.e., recipes) in topic k during time-span \mathcal{T}_* .

For every hyperedge $h(r) \in \mathcal{H}_*$, we first examine whether each $e_j \in E^{\text{main}}$ is present or not in the projected graphs $G^-(r)$ and $G^+(r)$ to characterize the structural transition of $\{H_t\}$ before and after the occurrence of $h(r)$. Then, we classify each $e_j \in E^{\text{main}}$ into one of the *four classes* “p \rightarrow p”, “p \rightarrow n”, “n \rightarrow p” and “n \rightarrow n” with respect to hyperedge $h(r)$ based on the following conditions: if “ $e_j \in E^-(r)$ and $e_j \in E^+(r)$ ” is satisfied, it belongs to class “p \rightarrow p”; if “ $e_j \in E^-(r)$ and $e_j \notin E^+(r)$ ” is satisfied, it belongs to class “p \rightarrow n”; if “ $e_j \notin E^-(r)$ and $e_j \in E^+(r)$ ” is satisfied, it belongs to class “n \rightarrow p”; and if “ $e_j \notin E^-(r)$ and $e_j \notin E^+(r)$ ” is satisfied, it belongs to class “n \rightarrow n”.

Moreover, for every topic k , we investigate the classes of each $e_j \in E^{\text{main}}$ with respect to the hyperedges in $\mathcal{H}_{*,k}$ (see Eq. (10)). Let $P_k(\text{p} \rightarrow \text{p} | e_j)$, $P_k(\text{p} \rightarrow \text{n} | e_j)$, $P_k(\text{n} \rightarrow \text{p} | e_j)$ and $P_k(\text{n} \rightarrow \text{n} | e_j)$ denote, respectively, the fractions of hyperedges $h(r)$ in $\mathcal{H}_{*,k}$ such that e_j belongs to classes “p \rightarrow p”, “p \rightarrow n”, “n \rightarrow p” and “n \rightarrow n” with respect to $h(r)$. Note that

$$P_k(\text{p} \rightarrow \text{p} | e_j) + P_k(\text{p} \rightarrow \text{n} | e_j) + P_k(\text{n} \rightarrow \text{p} | e_j) + P_k(\text{n} \rightarrow \text{n} | e_j) = 1.$$

Then, we further classify each $e_j \in E^{\text{main}}$ into one of the *four classes* “p \rightarrow p”, “p \rightarrow n”, “n \rightarrow p” and “n \rightarrow n” with respect to topic k based on the maximum value of posterior probabilities $P_k(\text{p} \rightarrow \text{p} | e_j)$, $P_k(\text{p} \rightarrow \text{n} | e_j)$, $P_k(\text{n} \rightarrow \text{p} | e_j)$ and $P_k(\text{n} \rightarrow \text{n} | e_j)$. To analyze the structural transitions of temporal hypergraph $\{H_t\}$ immediately before and after the occurrences of hyperedges for any topic k , we propose characterizing them in terms of the class of each $e_j \in E^{\text{main}}$ with respect to the topic k .

5 Experiments

We investigated Dessert, Fish-dish, Meat-dish and Vegetable-dish categories on Japanese recipe-sharing site “Cookpad”¹ during time-span \mathcal{T} from Dec 4, 2011 to Jan 28, 2013.

5.1 Datasets and Experimental Settings

By taking into account the attributes of Cookpad, we set the length of the “Before-After” investigation period Δt_0 as 28 days (see Eq. (3)), and analyzed the creation of hyperedges (i.e., the creation of recipes) within time-span \mathcal{T}_* from Jan 1, 2012 to Dec 31, 2012.

We constructed four datasets, each of which corresponds to one of the four categories described above. For each dataset, we identified the set of main ingredients² (i.e., the set of nodes V). Then, the numbers of nodes for the Dessert, Fish-dish, Meat-dish and Vegetable-dish datasets were 3, 157, 1, 538, 2, 713 and

¹ <https://cookpad.com/>.

² We first excluded common ingredients for Japanese food such as soy sauce, salt, sugar, water, edible oil, and the like. Then, we identified the ingredients that appeared in two or more recipes for each dataset.

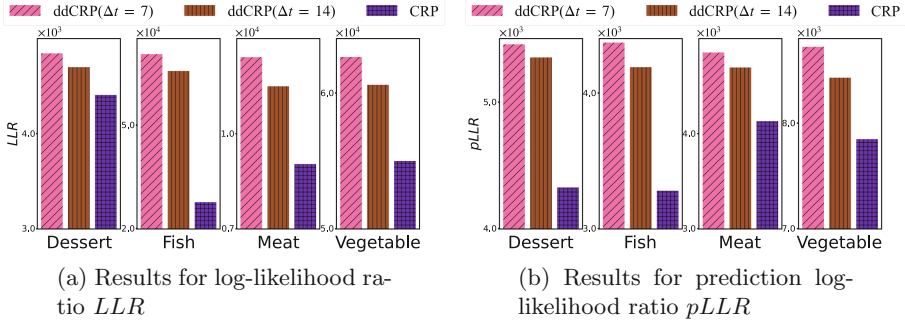


Fig. 3. Evaluation results of the proposed model for the four datasets.

4,985, respectively. Moreover, we focused on the recipes containing at least two main ingredients (i.e., the m -hyperedges with $m \geq 2$). Then, the numbers of hyperedges for the Dessert, Fish-dish, Meat-dish and Vegetable-dish datasets were 654, 495, 511 and 712, respectively.

5.2 Evaluation of Proposed Model

For extracting the topics of hyperedges from \mathcal{H}_* , we evaluated the proposed probabilistic model (see Sect. 4.1).

We compared the proposed model to three baseline models. As the first baseline model, we considered a multinomial distribution over V , which is same as the proposed model with only one topic. As the second baseline model, we adopted a Dirichlet mixture model of multinomial distributions over V that incorporates a *Chinese restaurant process* (CRP). Note that this is regarded as the proposed model with $\Delta t_1 = \infty$ and ignores the temporal trends and seasonality for topics in the proposed model. In the experiments, we set the hyper-parameter Δt_1 of the proposed model to $\Delta t_1 = 7$ days, considering the attributes of Cookpad. To confirm the suitability of this hyper-parameter setting, we also considered the proposed model using $\Delta t_1 = 14$ days as the third baseline model, and compared it with our proposed model using $\Delta t_1 = 7$ days.

We evaluated the proposed model in two different manners. First, we estimated the four target probabilistic models from the entire data \mathcal{H}_* , and evaluated how well they fit \mathcal{H}_* in terms of *log-likelihood ratio* LLR , where LLR is defined as the difference between each target probabilistic model and the first baseline model with respect to the log-likelihood for \mathcal{H}_* (see Eq. (7)). Next, we divided \mathcal{H}_* into the training set $\mathcal{H}_*^{\text{train}}$ and the test set $\mathcal{H}_*^{\text{test}}$, where $\mathcal{H}_*^{\text{train}}$ is the set of hyperedges (i.e., recipes) created from Jan 1, 2012 to Nov 30, 2012, and $\mathcal{H}_*^{\text{test}}$ is the set of hyperedges (i.e., recipes) created from Dec 1, 2012 to Dec 31, 2012. We estimated the four target probabilistic models from the training set $\mathcal{H}_*^{\text{train}}$, and evaluated their prediction performance in terms of *prediction log-likelihood ratio* $pLLR$, where $pLLR$ is defined as the difference between each target probabilistic model and the first baseline model with respect to

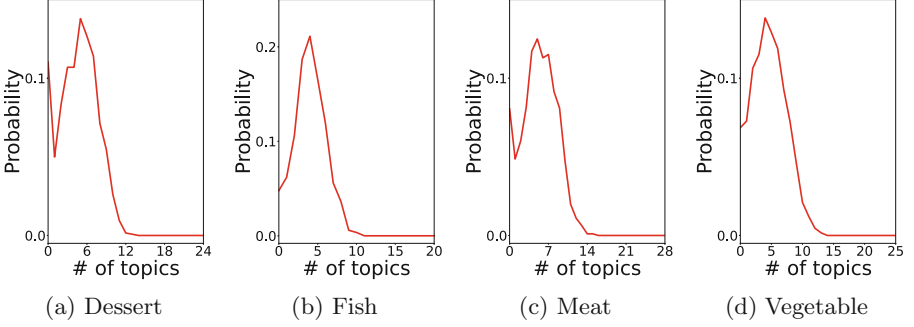


Fig. 4. Distribution of the number of topics with respect to which a main edge in the projected graph G of the entire hypergraph H belongs to class “ $n \rightarrow p$ ”.

the log-likelihood for the test set $\mathcal{H}_*^{\text{test}}$ (see Eq. (7)). As for learning the target probabilistic models, we set the hyper-parameters γ and μ to $\gamma = 1$ and $\mu_1 = \dots = \mu_N = 2$, and implemented 500 iterations with 50 burn-in iterations.

Figure 3 shows the evaluation results of the proposed model (ddCRP($\Delta t_1 = 7$)), the second baseline model (CRP) and the third baseline model (ddCRP($\Delta t_1 = 14$)) for the Dessert, Fish-dish, Meat-dish and Vegetable-dish datasets, where Fig. 3a and 3b indicate the results for log-likelihood ratio LLR and the results for prediction log-likelihood ratio $pLLR$, respectively. We see that the proposed model with $\Delta t_1 = 7$ significantly outperforms the three baseline models for the four datasets. These results imply that there exists a topic structure for the set of hyperedges created in the temporal hypergraph $\{H_t\}$, and their topics are related to the temporal trends and seasonality. Thus, we consider that the topics extracted from \mathcal{H}_* by the proposed model (see Eq. (10)) have significance.

5.3 Analysis Results

We investigated the structural transition of the temporal hypergraph $\{H_t\}$ for the four datasets using the proposed analysis method (see Sect. 4.2). For each dataset, we identified the set of main edges E^{main} (see Eq. (9)) by the criterion that an edge e_j in the projected graph G is *main*, i.e., $e_j \in E^{\text{main}}$, if there are ten or more hyperedges $h(r) \in \mathcal{H}_*$ such that $e_j \subset h(r)$. Also, we extracted the topics from \mathcal{H}_* by the proposed model (see Eq. (10)), where we excluded topics including fewer than ten hyperedges (i.e., recipes) for simplicity. Then, the number of main edges M (see Eq. (9)) and the number of topics K (see Eq. (10)) were $(M = 1,262, K = 24)$, $(M = 857, K = 20)$, $(M = 1,912, K = 25)$ and $(M = 2,378, K = 28)$ for the Dessert, Fish-dish, Meat-dish and Vegetable-dish datasets, respectively.

For each topic k , we analyzed the main edges e_j belonging to class “ $n \rightarrow p$ ” with respect to topic k since they tend to be newly created immediately after the occurrence of a hyperedge (i.e., recipe) in topic k . To this end, we first examined the number of topics with respect to which each main edge $e_j \in E^{\text{main}}$

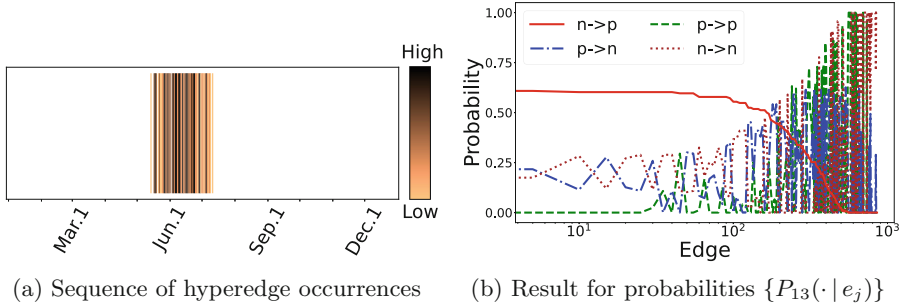


Fig. 5. Analysis results for topic “ $k = 13$ ” in the Fish-dish dataset.

belongs to class “ $n \rightarrow p$ ” for the four datasets. Figure 4 shows the distribution of the number of topics for any main edge. We observe that there were a few main edges belonging to class “ $n \rightarrow p$ ” with respect to a majority of topics, while there were a considerable number of main edges not belonging to class “ $n \rightarrow p$ ” with respect to every topic. We also see that for the four datasets, the distribution exhibited a bell-shaped-like curve having a peak at a relatively small number of topics (i.e., four or five topics). Here, we focused on the main edges belonging to class “ $n \rightarrow p$ ” with respect to only one topic since they are considered to represent the characteristic structural transitions unique to the corresponding topics. For the Fish-dish dataset, for instance, the seven unordered pairs of ingredients [‘mackerel’, ‘ginger’], [‘octopus’, ‘sake’], [‘righteye flounder’, ‘ginger’], [‘squid’, ‘cucumber’], [‘macrophyll’, ‘rice’], [‘shishito green pepper’, ‘sake’] and [‘myoga’, ‘sake’] became main edges satisfying the above condition, and topic “ $k = 13$ ” was their corresponding topic.

Due to space constraints, we focus on an analysis of topic “ $k = 13$ ” in the Fish-dish dataset as an example. Figure 5 shows the results. Here, Fig. 5a indicates the occurrence sequence of hyperedges (i.e., recipes) in topic “ $k = 13$ ” for the Fish-dish dataset. We observe that the hyperedges in topic “ $k = 13$ ” concentrated on some period of early summer, implying that topic “ $k = 13$ ” was related to seasonality. After carefully reviewing the recipes in topic “ $k = 13$ ”, we found that topic “ $k = 13$ ” represents Japanese-style fish dishes that go well with beer in early summer. In fact, for example, the recipes in topic “ $k = 13$ ” that received many *Cooksnap*³ were “Easy Pan-fried Japanese Amberjack Teriyaki”, “Easy Sesame Mayonnaise Salad with Exquisite Crispy Cucumbers” and “Grilled Salmon with Sesame Miso Mayonnaise”. Figure 5b indicates the probabilities $P_{13}(n \rightarrow p | e_j)$, $P_{13}(p \rightarrow n | e_j)$, $P_{13}(p \rightarrow p | e_j)$ and $P_{13}(n \rightarrow n | e_j)$ for each main edge e_j , where the main edges $e_j \in E^{\text{main}}$ are arranged in decreasing order according to their $P_{13}(n \rightarrow p | e_j)$ values. The seven main edges described above

³ On Cookpad, when users love a posted recipe, they can show their appreciation by sending a “Thank You” message along with a photo of the dish they actually cooked. This type of message is called a *Cooksnap*.

were ranked within the top 200 in the $P_{13}(n \rightarrow p | e_j)$ value-based ranking of main edges. For the top-ranked main edges in this ranking, their $P_{13}(n \rightarrow p | e_j)$ values were significantly higher than their $P_{13}(p \rightarrow n | e_j)$, $P_{13}(p \rightarrow p | e_j)$ and $P_{13}(n \rightarrow n | e_j)$ values. Thus, such top-ranked main edges represented class “ $n \rightarrow p$ ” in topic “ $k = 13$ ”. Here, the four main edges [‘basil’, ‘onion’], [‘baking powder’, ‘wheat flour’], [‘ketchup’, ‘wheat flour’] and [‘garlic’, ‘grape tomato’] were top-ranked in this ranking although they also belonged to class “ $n \rightarrow p$ ” with respect to topics other than topic “ $k = 13$ ”. These pairs of ingredients also characterize the structural transition of the temporal hypergraph $\{H_t\}$ for topic “ $k = 13$ ”. Here, we emphasize that our topic-based analysis method, investigating the structural transition of $\{H_t\}$ derived from recipe stream \mathcal{R} , has offered novel insights not uncovered by the previous studies [6, 8]. It has also revealed several intriguing properties in the evolution of Japanese homemade recipes. These results demonstrate its effectiveness.

6 Conclusion

We have explored a recipe stream created on a social media site dedicated to sharing homemade recipes in terms of a temporal hypergraph over a set of ingredients. Aiming to uncover the characteristics in the evolution of homemade recipes on the site, we have analyzed the structural transitions of the temporal hypergraph immediately before and after hyperedges are created. Unlike the previous studies for transition analysis of temporal higher-order networks by Cencetti et al. [6] and Fujisawa et al. [8], we have proposed a novel method based on topics and projected graphs for this “Before-After” analysis of created hyperedges. To identify the topics of hyperedges by taking into account the trends and seasonality of recipes, we have first proposed a probabilistic model with a ddCRP prior and presented its Bayesian inference method. Next, to provide an effective characterization framework that is independent of the size of created hyperedges, we have focused on the projected graph of the entire hypergraph, and proposed examining whether each of its main edges is present or not in the temporal hypergraph, both before and after the occurrences of hyperedges for each topic. Using real data of Japanese recipe sharing site “Cookpad”, we have empirically showed the effectiveness of the proposed probabilistic model, and uncovered several intriguing properties in the evolution of Japanese homemade recipes by employing the proposed analysis method.

Acknowledgement. This work was supported in part by JSPS KAKENHI Grant Number JP21K12152. The Cookpad dataset we used in this paper was provided by Cookpad Inc. and National Institute of Informatics.


References

1. Ahn, Y.Y., Ahnert, S.E., Bagrow, J.P., Barabási, A.L.: Flavor network and the principles of food pairing. *Sci. Rep.* **1**, 196:1-196:7 (2011)

2. Barabási, A.L.: *Network Science*. Cambridge University Press (2016)
3. Battiston, F., et al.: Networks beyond pairwise interactions: structure and dynamics. *Phys. Rep.* **874**, 1–92 (2020)
4. Benson, A.R., Abebe, R., Schaub, M.T., Jadbabaie, A., Kleinberg, J.: Simplicial closure and higher-order link prediction. *Proc. Natl. Acad. Sci. U.S.A.* **115**(48), E11221–E11230 (2019)
5. Blei, D., Frazier, P.: Distance dependent Chinese restaurant processes. *J. Mach. Learn. Res.* **12**, 2461–2488 (2011)
6. Cencetti, G., Battiston, F., Lepri, B., Karsai, M.: Temporal properties of higher-order interactions in social networks. *Sci. Rep.* **11**, 7028:1-7028:10 (2021)
7. Chodrow, P., Veldt, N., Benson, A.: Generative hypergraph clustering: from block-models to modularity. *Sci. Adv.* **7**, 1303:1-1303:13 (2021)
8. Fujisawa, K., Kumano, M., Kimura, M.: Transition analysis of boundary-based active configurations in temporal simplicial complexes for ingredient co-occurrences in recipe streams. *Appl. Network Sci.* **8**, 48:1-48:21 (2023)
9. Jain, A., Rakhi, N.K., Bagler, G.: Analysis of food pairing in regional cuisines of India. *PLoS One* **10**(10), 0139539:1-0139539:17 (2015)
10. Jiang, Y., Skufca, J.D., Sun, J.: Bifold visualization of bipartite datasets. *EPJ Data Sci.* **6**, 2:1-2:19 (2017)
11. Karrer, B., Newman, M.: Stochastic blockmodels and community structure in networks. *Phys. Rev. E* **83**, 016107:1-016107:10 (2011)
12. Kikuchi, K., Kumano, M., Kimura, M.: Analyzing dynamical activities of co-occurrence patterns for cooking ingredients. In: *Proceedings of ICDMW 2017*, pp. 17–24 (2017)
13. Lü, L., Zhou, T.: Link prediction in complex networks: a survey. *Phys. A* **390**(6), 1150–1170 (2011)
14. Makinei, L., Hazarika, M.: Flavour network-based analysis of food pairing: application to the recipes of the sub-cuisines from northeast India. *Curr. Res. Food Sci.* **5**, 1038–1046 (2022)
15. Min, W., Jiang, S., Liu, L.: A survey on food computing. *ACM Comput. Surv.* **52**(5), 92:1-92:36 (2019)
16. Park, D., Kim, K., Kim, S., Spranger, M., Kang, J.: Flavorgraph: a large-scale food-chemical graph for generating food representations and recommending food pairings. *Sci. Rep.* **11**, 931:1-931:13 (2021)
17. Sajadmanesh, S., et al.: Kissing cuisines: exploring worldwide culinary habits on the web. In: *Proceedings of WWW 2017 Companion*, pp. 1013–1021 (2017)
18. Teng, C.Y., Lin, Y.R., Adamic, L.A.: Recipe recommendation using ingredient networks. In: *Proceedings of WebSci 2012*, pp. 298–307 (2012)
19. Trattner, C., Elswiler, D.: Implications for meal planning and recommender systems. In: *Proceedings of WWW 2017*, pp. 489–498 (2017)
20. Veldt, N., Benson, A., Kleinberg, J.: Minimizing localized ratio cut objectives in hypergraphs. In: *Proceedings of KDD 2020*, pp. 1708–1718 (2020)
21. West, R., White, R.W., Horvitz, E.: From cookies to cooks: insights on dietary patterns via analysis of web usage logs. In: *Proceedings of WWW 2013*, pp. 1399–1410 (2013)



I Like You if You Are Like Me: How the Italians' Opinion on Twitter About Migrants Changed After the 2022 Russo-Ukrainian Conflict

Giulio Cordova¹, Luca Palla², Martina Sustrico², and Giulio Rossetti³

¹ Dipartimento di Fisica E. Fermi, Università di Pisa, Pisa, Italy
g.cordova@studenti.unipi.it

² Computer Science Department, University of Pisa, Pisa, Italy
{l.palla5,m.sustrico}@studenti.unipi.it

³ Institute of Information Science and Technologies “A. Faedo” (ISTI),
National Research Council (CNR), Pisa, Italy
giulio.rossetti@isti.cnr.it

Abstract. For the past decade, immigration has taken center stage in the Italian public discourse, providing fertile ground for far-right parties to cultivate fear and ultimately nurturing sentiments of animosity and racism. The Russian invasion of Ukraine introduced a unique narrative as European white refugees sought shelter, diverging from the conventional refugee archetype ingrained within the Italian collective consciousness. Employing the tools of Network Science, this study aims to dissect whether the Russo-Ukrainian conflict in 2022 has triggered shifts in Italians' viewpoints toward refugees. Specifically, we delve into whether the proximity of the conflict and the ethnic parallels between refugees and Italians have prompted changes in attitudes toward refugees. Through an exploration of the intricate interplay between international conflict, social network dynamics, and evolving sentiments, this research contributes to understanding the intricate dynamics that underlie shifts in public sentiment regarding migration.

Keywords: Refugees · Russo-Ukrainian Conflict · Migrants · Social Network Analysis · Twitter · Opinion Dynamics · Community Discovery

1 Introduction

Italy's geographical location has historically positioned it as a primary route for undocumented migrants, encompassing those migrating for economic motives and asylum seekers alike. Notable events such as the Arab Spring in 2011 and the Syrian War in 2014 contributed to a significant rise in boat arrivals on Italian shores [15]. As these numbers surged, far-right political entities in Italy adeptly

steered the migration discourse toward themes of “threat” and “security,” effectively integrating migration as a pivotal element within their political agendas. This strategic maneuver led to the establishment of echo-chambers, distorting the factual migrant and asylum seeker statistics. An Ipsos study conducted in 2019 [6], revealed a discrepancy between the perceived migrant presence on Italian soil (31%) and the actual data (9%). Moreover, 63% of Italians acknowledged exposure to misinformation on immigration, which led them to erroneously attribute a majority of crimes to migrants (33% of respondents) and perceive immigration as a menace to Italy (57% of respondents).

As expounded by Dylan Patrick McGinnis in an article within the *Yale Review of International Studies* [9], the strategies employed by political figures such as Matteo Salvini (Lega’s leader) and Giorgia Meloni (leader of Fratelli d’Italia) have positioned immigration policies within the far-right ideological spectrum. This approach hinges on cultivating an “us versus them” dichotomy, casting migrants as the ‘other’ and ascribing Italy’s economic woes to immigrant populations. In accordance with these postulates, the presence of a proximate conflict, exemplified by the Russian invasion of Ukraine, holds the potential to elicit empathetic sentiments among Italians toward refugees. Additionally, ethnic similarities might dismantle Salvini’s entrenched “us versus them” dichotomy.

This article seeks to delve into the dynamics shaping Italians’ perceptions of refugees, with a particular emphasis on scrutinizing the evolution and transformation of opinions prior to and following the 2022 Russo-Ukrainian conflict. The analysis is rooted in an examination of a dataset comprising tweets written in Italian.

2 Related Work

In recent decades, the issue of immigration has become increasingly relevant both in the social and academic contexts. Studies by Ambrosini [1] and Venturini [13] demonstrate how this phenomenon impacts various aspects of our lives, from the economic to the political sphere. Several publications, including Hampshire’s work [5], have highlighted how politics has often been inadequate in controlling migratory flows, especially during critical periods such as the Libyan war or the Arab Spring [4]. Public opinion and the media’s treatment of the subject have been frequently investigated to understand their evolution over time. Kosho’s study [8] reports that mass media exert significant influence on the opinions of individual citizens and regulators, and how these actors often influence each other in a continuous cycle. A media coverage report [10] centered on the main Italian newspapers highlighted a positive correlation between their consumption and a positive attitude towards migrants. With the evolution of technology and the widespread use of social networks, the debate has increasingly shifted to the internet. Vilella et al. [14] have proposed a study based on data extracted from Twitter, showing that citizens’ opinions are not strongly correlated with their geographic location but are more closely tied to their political orientation. These results have also been confirmed by the study published by Radicioni [11].

Looking at the behavior of migrants and native on Twitter has been showed that migrants have more followers than friends and that they tends to connect more based on nationality despite the country of residence [7]. It is often observed that war refugees and economic migrants are perceived by citizens as a single entity, raising the question of whether there is a genuine awareness of the difference between the two. A passage from Torres’ study [12] reports that, despite European citizens’ understanding of this difference, a sort of “us against them” dynamic is often created, fueled by anti-migrant populist narratives. Building upon some established results over time, our study aims to investigate whether the outbreak of the war between Russia and Ukraine has changed opinions about migrants within the Italian territory.

3 Italians’ Perception of Migrants on Twitter

In this section, we delve into the investigation of public opinion dynamics exhibited by a representative subset of Italian speaking Twitter users with regard to refugees. Our primary focus centers on the temporal trajectory preceding and subsequent to the Russo-Ukrainian conflict in 2022 [16].

Methodology and Experimental Setup: Our analysis proceeds through two essential stages. Firstly, we detail the methodology employed for data analysis and classification. Subsequently, we elaborate on our experimental environment, emphasizing the evolution of opinions across time. Two discrete datasets are formulated for distinct purposes:

– *Graph Construction and Analysis Dataset (GCAD):*

To construct the social interaction graph, a comprehensive dataset was acquired by scraping Italian tweets containing relevant hashtags associated with immigration¹ from September 1, 2020, to September 1, 2022. The initial dataset comprised 71,735 tweets, encompassing 13,575 distinct users and 69,580 conversations. Naturally, certain samples were discarded due to inconsistencies, duplicated entries, and other data anomalies. Exploiting conversation IDs, we further collected all tweets affiliated with those conversations to reconstruct complete discourse.

– *Classifier Training Dataset (CTD):*

we created a specialized dataset focused on tweets featuring polarized hashtags associated with a common theme. The collection period spanned from September 2018 to September 2022. Each of these tweets is assigned a label that indicates the extent of positive sentiment towards migrants conveyed in the text. Tweets marked with #restiamoumani² were designated with a label of 1, signifying a pro-refugee stance, while those containing #portichiusi³ were

¹ In Italian: [#rifugiato, #rifugiati, #profugo, #profughi, #migrante, #migranti].
English: [#refugee, #refugees, #migrant, #migrants].

² In English #stayhuman.

³ In English #closedharbors.

assigned label 0, indicating an anti-refugee leaning. This dataset serves as the foundation for training a classifier, which subsequently operates to categorize the tweets extracted from the GCAD.

Nodes Labeling: The CTD serves uniquely as a training dataset for developing a classification model fine-tuned to categorize the tweets of the GCAD. The classifier assigns a tweet label (TL) to each tweet of the GCAD:

- TL = 1: positive leaning about refugees;
- TL = 0: negative leaning about refugees.

Throughout the experimentation phase, an assortment of models underwent evaluation. Among these models, the Logistic Regression model emerged as the optimal candidate, displaying the highest performance metrics, including an accuracy score of 0.78, along with effective probability calibration.

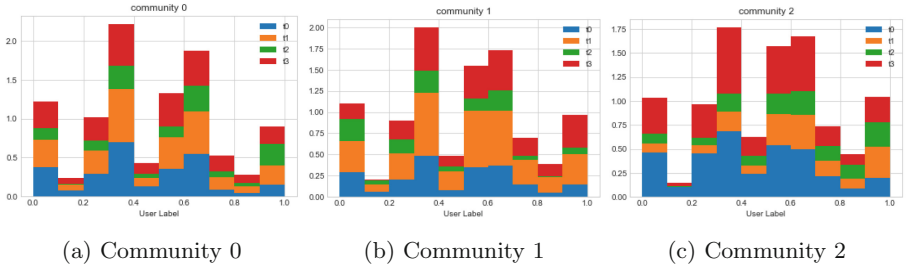
In pursuit of a user-centric viewpoint, we introduced a unique continuous score computation for each user, predicated on the average predicted leaning of their associated TL. This user label (UL) encapsulates the collective sentiment trajectory exhibited by each user towards the subject matter. For instance, if a user contributed two tweets, characterized by labels 1 and 0, their UL would manifest as 0.5. By adopting this approach, we attain a comprehensive portrayal of the prevailing inclination of each user concerning the focal theme.

Network Characterization: The network $G = (V, E)$ employed in our analysis is constructed based on the GCAD. Within this framework, users were designated as nodes, represented by V , while the interactions in the form of replies between users were accounted for as edges, depicted as E . Due to the presence of isolated nodes, our focus was directed towards the giant component of the network. Post the cleansing process, the resulting G comprises $|V| = 46,978$ nodes and $|E| = 88,029$ edges. G has a density of $7.98 \cdot 10^{-05}$, and it exhibits an average Clustering Coefficient of 0.078, coupled with an average shortest path length of 3.39. Notably, G showcases a minor degree of disassortativity, as inferred from its Newman Assortativity coefficient of $R = -0.2$.

Meso-scale Topologies: The interactions and information assimilation within discussions lead individuals to be influenced by their neighbors and the broader context. Consequently, delving into how users cluster on a meso-scale level, such as forming communities, during the two-year observation period, becomes pivotal. This endeavor facilitates the comprehension of network dynamics and offers insights into the composition of debates and the amalgamation of ideological perspectives. To gain insights into how or if the ongoing debate impacted the network's topology, we partitioned the dataset into four distinct semesters. Subsequently, we extracted the giant component for each semester. From these snapshots, we derived node clusters using the Principled Clustering algorithm [3]. Across the four snapshots, the count of communities generated varied between eight and nine. This variation highlights the occurrence of diverse community dynamics during the observation period. A brief summary of the results and

Table 1. Results and performance of Principled Clustering (PC) algorithm on the four different snapshots

Snapshot	#communities	Modularity	Conductance
Sep 2020–Feb 2021	8	0.5670	0.2860
Mar 2021–Aug 2021	9	0.5504	0.3119
Sep 2021–Feb 2022	8	0.5919	0.2496
Mar 2022–Aug 2022	9	0.5685	0.2970

**Fig. 1.** User label distribution across a sample of communities. The histograms offer a cumulative perspective across four timestamps. The counts are color-coded: blue for the first semester, yellow for the second semester, green for the third semester, and red for the fourth semester.

performances of the PC algorithm is shown in Table 1. To assess the relevance of community changes w.r.t. ideological composition, we studied the user label distribution within each community over time. As depicted in Fig. 1, it becomes evident that communities predominantly do not consist of users who exclusively share a singular opinion. This characteristic is consistent across all community-label composition distributions through time and is supported by both the high variance and the similarity in means for each distribution. Therefore, the composition of communities appears to exclude the presence of polarized conversations in the network under analysis, instead fostering heated discussions and a fluent flow of ideas and information.

A Comparison of the User Label Distribution: We conducted a comparative analysis to determine whether tweets concerning Ukraine exhibited a more positive sentiment compared to those discussing migrants in a broader context. To achieve this, we visualized the distribution of user labels (as computed in the nodes labeling subsection) in Fig. 2, considering two distinct scenarios: tweets containing the substring *ucr* (the first three letters of the Italian word *Ucraina* - Ukraine) and tweets without it. Figure 2a, centered on the former scenario, vividly demonstrates a distinct shift in the UL distribution towards 1, indicating a pronounced positive inclination. In contrast, Fig. 2b, which addresses the latter scenario, reveals a more evenly distributed UL. This trend gains further

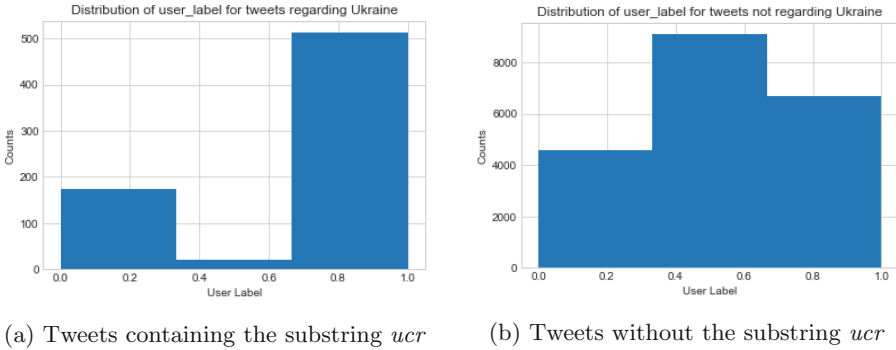


Fig. 2. Distribution of the user label in the two cases.

validation from the mean user label values: in the former case, the mean computes to 0.72, while in the latter, it stands at 0.51. This observation underscores that tweets related to Ukraine tend to manifest a more optimistic sentiment compared to the broader dataset.

Pre- and Post-Russo-Ukrainian Conflict Comparison: The comparison of opinions before and after the Russo-Ukrainian conflict entails partitioning the dataset GCAD into two distinct time frames:

- “before the war”: all the tweets written before the Russian invasion of Ukraine (24/2/2022)
- “after the war”: all the tweets written after the Russian invasion of Ukraine (24/2/2022).

Subsequently, the user label underwent re-computation, resulting in the mean value of each tweet label within each temporal frame, contingent on users who composed at least two tweets within each temporal snapshot. For enhanced data visualization and manageability, a decision was made to categorize the user labels into four distinct bins, incorporating the user label within:

- 0–0.25 (bin 0): negative leaning about migrants;
- 0.25–0.5 (bin 1): slightly negative leaning about migrants;
- 0.5–0.75 (bin 2): slightly positive leaning about migrants;
- 0.75–1 (bin 3): positive leaning about migrants;

To provide empirical support for the evolving dynamics of users’ opinions, we employed a Sankey Plot, as depicted in Fig. 3. This visual representation effectively illustrates the shifts in users’ opinion distributions across the previously identified temporal snapshots.

To validate the observed shifts, we infer whether we can reject the hypothesis H_0 which asserts that the observed change of opinions are only due to random fluctuations. Under this hypothesis of total randomness, each user with opinion

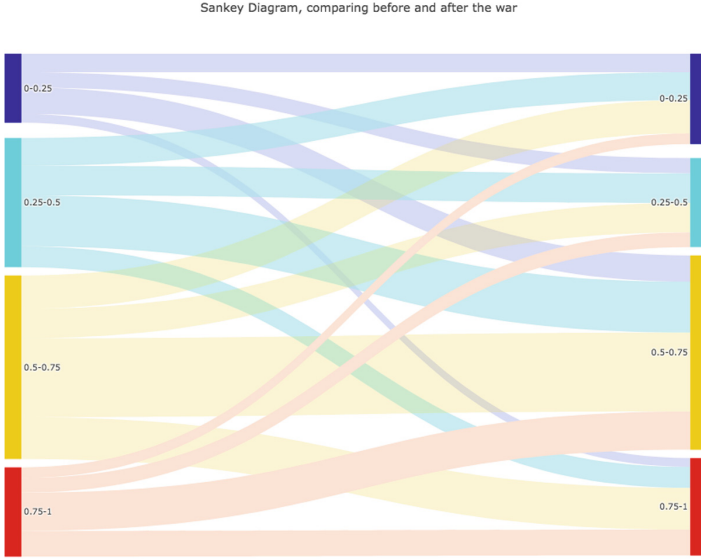


Fig. 3. Sankey Diagram of user label in the two temporal snapshots, before and after the 2022 Russo-Ukrainian conflict

x has the same probability of changing his opinion to some value x' . Thus, the aleatory variable x' is distributed according to a continuous uniform distribution $\mathcal{U}(0, 1)$. Since x' is independent of the original opinion x , we can treat each original bin i (i.e. the bins on the left side of Fig. 3) independently from the other ones. Under the hypothesis H_0 , the fluxes coming from each bin i behave in the same way as 4 histograms h_i of n_i samples extracted from $\mathcal{U}(0, 1)$. As a statistical test, we adopt a goodness of fit based on the likelihood ratio statistic λ [2], which tends to a non-central χ^2 in the asymptotic limit, as stated by Wilk's Theorem. Therefore, we can test the counts in each flux with the classic Pearson's χ^2 , defined as

$$\lambda(n_{ij}) = \frac{(n_{ij} - f_{ij})^2}{\sigma_{ij}^2}, \tag{1}$$

where:

- n_{ij} is the number of counts in the flux from the bin i to the bin j ;
- $f_{ij} = n_i/4$ is the mean of the multinomial distribution of the counts y_{ij} bin j , i.e. the integral of the pdf $\mathcal{U}(0, 1)$ over the range of the bin j ;
- $\sigma_{ij}^2 = \frac{n_{ij}}{n_i}(n_i - n_{ij})$ is the variance of the multinomial distribution of the counts y_{ij} bin j .

For the calculation of the mean f_{ij} and variance σ_{ij}^2 , we employ the multinomial distribution since the parameter n_i is fixed apriori. The critical region for the test is then

$$\lambda(n_{ij}) > q_\alpha \quad \text{with } q_\alpha : 1 - F_{\chi_1^2}(q_\alpha) = \alpha$$

Table 2. Values used and results for the goodness of fit test of the hypothesis H_0 of random change of opinion. n_{ij} is the number of counts in the flux from the bin i to the bin j ; $\lambda(n_{ij})$ is the statistic based on the likelihood ratio, defined in (1); $q_{5\%}$ is the significant threshold; the last two rows indicate whether the test is significant and which p-value is associated to it.

	0→0	0→1	0→2	0→3	bin 0	1→0	1→1	1→2	1→3	bin 1
n_{ij}	99	82	141	47	/	149	158	272	112	/
$\lambda(n_{ij})$	0.658	1.519	34.350	29.594	66.101	4.353	1.679	76.029	28.485	110.546
$q_{5\%}$	3.841	3.841	3.841	3.841	9.488	3.841	3.841	3.841	3.841	9.488
significant	N	N	Y	Y	Y	Y	N	Y	Y	Y
p-value	0.417	0.218	4.6e-9	5.3e-8	1.5e-13	0.037	0.195	2.8e-18	9.4e-8	5.6e-23

	2→0	2→1	2→2	2→3	bin 2	3→0	3→1	3→2	3→3	bin 3
n_{ij}	178	158	422	224	/	57	78	205	138	/
$\lambda(n_{ij})$	24.745	41.582	169.191	2.511	238.029	43.584	19.216	81.565	3.819	648.184
$q_{5\%}$	3.841	3.841	3.841	3.841	9.488	3.841	3.841	3.841	3.841	9.488
significant	Y	Y	Y	N	Y	Y	Y	Y	N	Y
p-value	6.5e-7	1.1e-10	1.1e-38	0.113	2.4e-50	4.1e-11	1.2e-5	1.7e-19	0.074	6.3e-149

We set $\alpha = 0.05$ and calculate the p-value as

$$p = 1 - F_{\chi^2_1}(\lambda)$$

In Table 2 we report a brief of the p-value obtained for each flux ij and a combination of them.

The test results are significant for each bin. Moreover, the general p-value p can be obtained by combining the individual p_i obtained for each bin i by creating the statistics

$$\lambda'(p - i) = -2 \log \prod p_i = -2 \log \sum p_i \simeq 602, \tag{2}$$

which also follows a χ^2 distribution with $n = \sum_i i$ degrees of freedom. Since the value of lambda is much larger than the threshold value of 26.296 (calculated from the χ^2 distribution with $n = 16$ dof), we conclude that the test is significant. With this test, we can then infer that the fluxes are generally not random, with the exception of the fluxes $0 \Rightarrow 0$, $0 \Rightarrow 1$, $1 \Rightarrow 1$, $2 \Rightarrow 3$ and $3 \Rightarrow 3$. We can thus conclude that the majority of the observed changes in opinion are not due to random fluctuations, but rather, there is some underlying effect responsible for users transitioning from one category to another.

The result shown in Fig. 3 highlights the natural behaviour of a user to have a moderate change of opinion instead of a drastic one. Another noticeable trend is the enlargement of the bins 0 and 3 in the snapshot following the conflict. This intensified polarization might stem from the influx of Ukrainian refugees or could be an outcome of the proximity to the Italian election. Within our dataset, a substantial quantity of tweets were authored post-May 2022, with a pronounced peak in July 2022, coinciding with the commencement of the electoral campaign for the Italian election scheduled in September 2022. Consequently, this temporal

Table 3. Percentage variation in the User Label (UL) within each bin (or bin combination). BtA represents the alteration in user labels from before to after the 2022 Russo-Ukrainian Conflict; BtA/Ukr signifies the same, with tweets containing the substring “ucr” removed; “one to two” denotes the change between the initial and second temporal snapshots outlined in the *Meso-scale topologies* section.

	0-0.5	0.5-1	0-0.25	0.25-0.5	0.5-0.75	0.75-1
BtA	-9%	+7%	+31%	-31%	+6%	+9%
BtA/Ukr	-9%	+6%	+33%	-31%	+5%	+9%
one to two	-5%	+4%	-2%	-9%	-0.1%	+6%

interval hosts a significant volume of tweets authored by politicians who possess distinct viewpoints on immigration. This, in turn, can trigger a cascade effect, leading users to adopt more definite positions as well.

Given our objective of discerning shifts in opinions regarding migrants and refugees, we additionally investigate the distribution of the user label difference, denoted as $x_{ij} = x_j - x_i$. This quantifies the extent to which a user’s opinion has altered from one temporal snapshot i to another j . To explore the potential influence of the conflict on these shifts in opinions, our analysis centers on the temporal snapshots labeled as “before the war” and “after the war.” The dataset capturing these calculated opinion differences is denoted as “Before to After (BtA).” In our initial analysis, we concentrate on examining the percentage increase or decrease of each bin count between the two temporal snapshots. This approach provides us with a quantitative gauge of the extent of opinion change across these periods. Furthermore, we extend our investigation to encompass a dataset excluding tweets containing the substring “ucr”, referred to as “BtA/Ukr”. This exclusion aims to mitigate potential biases, since the tweets regarding Ukraine are in general more positive, as depicted in Fig. 2a. Beyond the changes in individual bins, it is insightful to consider the cumulative values within the bins 0-1, and 2-3, which correspond, respectively, to negative and positive leaning. To validate the analysis, we repeat the analysis between the first two semestral snapshot created in the *Meso-scale topologies* paragraph, calling this dataset “one to two”. Table 3 presents the outcomes of this analysis. Comparing the rows for BtA and BtA/Ukr reveals their similarity, indicating no significant alteration. However, a subtle effect is observed in the slight decrease of the positive class (0.5-1), accompanied by a corresponding increase in the negative leaning class, aligning with our expectations. Comparing the changes in “on to two”, these are significantly smaller than the ones in row BtA. This may be a hint that the Russo-Ukrainian conflict (and also the Italian election) could have had an effect on the Italians’ opinion about migrants. To validate this kind of inference, a 2-samples Kolmogorov-Smirnov (KS) test is designed, where we test whether the distribution of x_{ij} is different in the case of “Before to After” with respect to a random split like “one to two”.

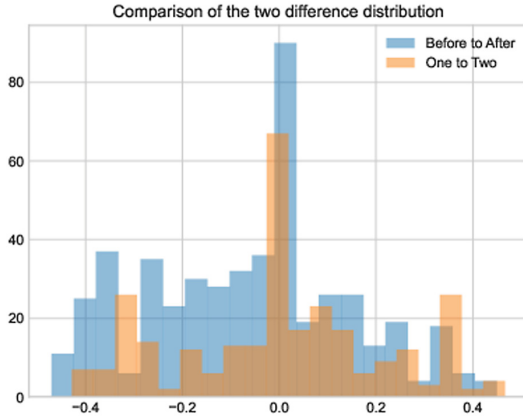


Fig. 4. Distribution of the difference of UL between to temporal snapshot. In orange we reported the difference between the temporal snapshot before and after the conflict, while in blue between the first and second semester defined in the *Meso-scale topologies* paragraph

The test result significant, with a p-value of $p = 2.44 \cdot 10^{-5}$, i.e. the two distribution are really different from one another. The overlap of the distribution is reported in Fig. 4.

4 Conclusions

The goal of this article was to investigate the dynamics of Italians’ opinions about refugees, with a specific focus on the comparison and shifting opinions before and after the 2022 Russian-Ukrainian conflict. Our analysis underlies that most of the observed change of opinions between the studied temporal snapshots are not random, conversely the test we designed highlights that there is an undergoing effect. We studied whether this effect is different between the temporal snapshots regarding the war or between two randomly chosen intervals. The KS test we designed resulted significant. Our hypothesis is that this difference is due to the Russian-Ukrainian conflict: such a variation emerged indeed, however, we can not conclude on the relative importance of different confounders that generated such an effect.

Acknowledgment. This work is supported by: the EU NextGenerationEU programme under the funding schemes PNRR-PE-AI FAIR (Future Artificial Intelligence Research); the EU - Horizon 2020 Program under the scheme “INFRAIA-01-2018-2019 - Integrating Activities for Advanced Communities” (G.A. n.871042) “SoBigData++: European Integrated Infrastructure for Social Mining and Big Data Analytics” (<http://www.sobigdata.eu>); PNRR-“SoBigData.it - Strengthening the Italian RI for Social Mining and Big Data Analytics” - Prot. IR0000013.

References

1. Ambrosini, M.: Immigration in Italy: between economic acceptance and political rejection. *Int. Migr. Integr.* **14**, 175–194 (2013). <https://doi.org/10.1007/s12134-011-0231-3>
2. Baker, S., Cousins, R.D.: Clarification of the use of chi-square and likelihood functions in fits to histograms. *Nucl. Instrum. Methods Phys. Res.* **221**(2), 437–442 (1984). [https://doi.org/10.1016/0167-5087\(84\)90016-4](https://doi.org/10.1016/0167-5087(84)90016-4). <https://www.sciencedirect.com/science/article/pii/0167508784900164>
3. Ball, B., Karrer, B., Newman, M.E.J.: Efficient and principled method for detecting communities in networks. *Phys. Rev. E* **84**, 036,103 (2011). <https://doi.org/10.1103/PhysRevE.84.036103>. <https://link.aps.org/doi/10.1103/PhysRevE.84.036103>
4. Fargues, P., Fandrich, C.: Migration after the Arab spring. Migration Policy Centre Research Report 2012/09 (2012). <https://hdl.handle.net/1814/23504>
5. Hampshire, J.: Europe’s migration crisis. *Polit. Insight* **6**(3), 8–11 (2015). <https://doi.org/10.1111/2041-9066.12106>
6. IPSOS: Ciak migraction: indagine sulla percezione del fenomeno migratorio in Italia. WeWorld Onlus (2019). <https://www.ipsos.com/it-it/ciak-migration-indagine-sulla-percezione-del-fenomeno-migratorio-italia>
7. Kim, J., Pratesi, F., Rossetti, G., Sirbu, A., Giannotti, F.: Where do migrants and natives belong in a community: a twitter case study and privacy risk analysis. *Soc. Netw. Anal. Mining* **13** (2022). <https://api.semanticscholar.org/CorpusID:255292156>
8. Kosho, J.: Media influence on public opinion attitudes toward the migration crisis. *Int. J. Sci. Technol. Res.* **5**(5), 86–91 (2016)
9. McGinnis, D.P.: Anti-immigrant populism in Italy: an analysis of Matteo Salvini’s strategy to push Italy’s immigration policy to the far right. *Yale Rev. Int. Stud.* (2021). <http://yris.yira.org/winter-issue/4659>
10. Mertens, S., De Coninck, D., d’Haenens, L.: A report on legacy media coverage of migrants (2021)
11. Radicioni, T., Squartini, T., Pavan, E., Saracco, F.: Networked partisanship and framing: a socio-semantic network analysis of the Italian debate on migration. *PLoS one* **16**(8), e0256,705 (2021)
12. Torres, M.J.: Public Opinion Toward Immigration, Refugees, and Identity in Europe: A Closer Look at What Europeans Think and How Immigration Debates Have Become So Relevant. *IEMed Mediterranean Yearbook*, Rome (2019)
13. Venturini, A.: Do immigrants working illegally reduce the natives’ legal employment? Evidence from Italy. *J. Popul. Econ.* **12**, 135–154 (1999). <https://doi.org/10.1007/s001480050094>
14. Vilella, S., et al.: Immigration as a divisive topic: clusters and content diffusion in the Italian twitter debate. *Future Internet* **12**(10), 173 (2020)
15. Villa, M.: Fact-checking: migrazioni 2021. ISPI (2021)
16. Wikipedia: Russo-Ukrainian war (2023). https://en.wikipedia.org/wiki/Russo-Ukrainian_War. Accessed on 28 Aug 2023



Modeling the Association Between Physician Risky-Prescribing and the Complex Network Structure of Physician Shared-Patient Relationships

Xin Ran^{1,2}, Ellen R. Meara³, Nancy E. Morden^{2,4}, Erika L. Moen^{1,2}, Daniel N. Rockmore^{5,6}, and A. James O'Malley^{1,2}(✉)

¹ Department of Biomedical Data Science, Geisel School of Medicine at Dartmouth, Lebanon, NH, USA

James.OMalley@Dartmouth.edu

² The Dartmouth Institute for Health Policy and Clinical Practice, Geisel School of Medicine at Dartmouth, Lebanon, NH, USA

³ Department of Health Policy and Management, Harvard T.H. Chan School of Public Health, Boston, MA, USA

⁴ United HealthCare, Minnetonka, MN, USA

⁵ Department of Mathematics and Department of Computer Science, Dartmouth College, Hanover, NH, USA

⁶ The Santa Fe Institute, Santa Fe, NM, USA

Abstract. Homophily is the social network phenomenon whereby similar individuals have a greater propensity to form ties. Motivated by concerns of risky-prescribing among older patients in the United States, we developed exponential random graph models to estimate the effect of homophily of risky prescribing net of other physician characteristics and network features in a complex network. We also developed novel network measures and associated non-parametric statistical tests that allow for greater homophily in specific triadic configurations (“super-homophily”). Using a shared-patient network of all physicians who treated patients residing in the US state of Ohio in 2014, we found statistical evidence of physician homophily (both in level and heterogeneity across regions) and triadic homophily on risky prescribing. Our findings may explain the emergence of prescriber communities, motivate group-level prescriber interventions to directly reduce risky-prescribing, and motivate interventions that reshape physician shared-patient networks to indirectly reduce risky-prescribing.

Keywords: Complex networks · Homophily · Polypharmacy · Risky prescribing · Shared-patient physician network · State-space · Transition matrix

1 Introduction

Risky prescribing among the older population is a health concern for which public health interventions are highly sought-after. Risky prescribing commonly

refers to the excessive prescribing (“polypharmacy”) of unwarranted prescriptions that deviate from guidelines [5, 10, 13]. The older population in the United States (U.S.) consumes more than one-third of prescription medications, yet they consist of around 15% of the U.S. population [12, 32]. Even more concerning are the adverse events associated with risky prescribing. Specifically, the combined usage of opioids and benzodiazepines or non-benzodiazepine sedative-hypnotics (sedative-hypnotics) is reported to have a higher risk of overdose than using opioids alone [7, 9, 30]. Social network analysis has proven to be effective for studying collaborations among physicians and their association with patients’ health outcomes [4, 11, 20, 24]. Therefore, social network analysis has great potential to provide insights for intervening on physicians to help combat risky prescribing.

Understanding how different prescribing behaviors are embedded in a shared-patient physician network may help identify the most important physicians to intervene on and subgroups of physicians where the intervention is likely to have greatest impact. For example, if actors with certain traits in common underlie the network, then an intervention that targets groups of connected actors with similar traits might be the most effective form of intervention. Homophily in the healthcare setting among physicians can reveal important factors driving physicians’ communications and collaborations. Previous studies have found that physicians in closer geographic proximity or with similar patient panels were more likely to share patients, and physicians with similar organizational affiliations were more likely to develop professional relationships [17, 19]. Another study found that homophily in a network of opioid users was associated with the number, type, and daily dosage of opioid prescriptions [3]. The existence of homophily can reinforce the influence between dyads (a pair of connected individuals in a social network) such that individuals are more prone to interact with individuals they resemble than those lacking traits in common [6]. By comparing physician homophily associated with prescribing at the state and HRR levels, we examine whether physician prescribing intensity clusters across geographic regions, extending the health care variations literature in a unique way.

Exponential random graph models (ERGMs) provide a general modeling framework for relating network phenomena and actor attributes to the likelihood of observing a network. One challenge with ERGMs is model degeneracy, the phenomenon in which a model puts most of its mass on a very dense or sparse network. Degeneracy has been commonly encountered by researchers seeking to estimate the extent to which dyadic-dependent network phenomena such as transitivity underlie the network [15, 16]. When using ERGMs to study homophily, degeneracy often limits our ability to isolate the true level of homophily from confounding effects of other network effects such as the various forms of triadic dependence (network dependence involving three actors), including transitivity. To overcome this problem, we introduce two new network statistics and randomization tests of whether their prevalence in the network exceeds that expected in the absence of homophily, gaining unique insights into the extent to which risky prescribing is associated with the network structure of physician relationships.

In this study of the relationship of physician networks to risky-prescribing, we developed several innovations related to the study of complex networks. These include novel triadic network statistics and non-parametric statistical tests to study prescribing-associated homophily within dyads to partially overcome the challenge of triadic dependency in ERGMs and development of a state-space framework to quantify physician prescribing behavior by attributing their contribution to prescribing and deprescribing based on patient drug status change. With the above novelties, we shed unique light on the emergence of prescriber communities and motivate group-level prescriber interventions to reduce risky prescribing.

2 Methods

2.1 Study Overview

Medicare Part D prescription fill records for three classes of risky drugs (opioids, benzodiazepines, and sedative-hypnotics) along with their corresponding prescribers were extracted for a 40% random sample of beneficiaries with Part D claims in 2014. Separately, we used a 40% random sample of all Medicare fee-for-service beneficiaries residing in the state of Ohio in 2014 to extract relevant physician-patient encounters for constructing a unipartite physician network. A visit to physician i followed by another visit to physician j by the same patient within a certain time window (a “patient referral”) may provide evidence of a meaningful professional relationship from physician i to j [1, 2, 22]. In the binary-undirected physician network that we constructed, physicians i and j were connected if they had directed edges in both directions during 2014 (i.e., they shared at least one patient in each direction). The network was limited to physicians who had at least prescribed one drug in the aforementioned three drug classes and was reduced to its largest connected component (LCC) to eliminate isolated dyads of physicians who only shared patients among themselves as they are likely practicing in a reduced manner. To further study homophily associated with risky prescribing in hospital referral regions (HRRs) and its possible variation across different HRRs, we partitioned the LCC network into HRR sub-networks. More details of the study cohort definition, workflow, and physician network construction are in Sections 1.1 and 1.2 and Figure S1 of the supplemental online appendix (see GitHub link at the end of this paper).

2.2 Exponential Random Graph Models (ERGMs)

An ERGM is an exponential family model designed for relational data. Standard regression models cannot handle network data if the status of the edges (ties) in the network are statistically dependent, such as in a complex network, as this violates the independence and no interference assumptions of standard regression models [8]. ERGMs overcome this issue and allow nodal attributes, edge attributes, dyadic dependencies, and some higher-order network dependencies to be simultaneously accounted for when modeling the network [26, 27, 29].

ERGMs model the probability distribution of all possible networks given a set of nodes, and in estimation, seek the values of the parameters of the network statistics that make the observed network the most likely compared to all other possible realizations of the network [26]. Mathematically, the model is given by,

$$Pr(Y = y) = \left(\frac{1}{\kappa}\right) \exp\left\{\sum_A \eta_A g_A(y)\right\} \quad (1)$$

where y represents the observed network ($y_{ij} = 1$ if there is an edge between node i and j , and 0 otherwise) and $g_A(y)$ represents possible network statistics such as the number of edges, the number of reciprocated or mutual edges (for directed networks), certain degree-related configurations (e.g., k-stars), triadic configurations, and nodal or edge-level attributes. The set A indexes the network statistics included in a model vector $g(y)$. The parameter η_A is the coefficient of certain network statistics, which corresponds to the conditional log odds of a tie with a one-unit change in the network statistics holding the rest of the terms in the model fixed. A positive value of an element of η_A indicates that the network statistic represented by the corresponding element of $g_A(y)$ is more prominent in the observed network than expected by chance given the other network statistics in the model. The quantity κ is a normalizing constant equal to the sum of $\exp\{\sum_A \eta_A g_A(y)\}$ over all possible realizations of the network with the given number of nodes [14].

A wide range of network statistics capturing various elements of network structure may be included as predictors in an ERGM [21]. Network statistics that capture the level of homophily of specified attributes in the network are of primary interest in our application (see Section 1.3 and Table S1 of the supplemental online appendix for the mathematical specifications of these statistics).

2.3 New Network Statistics: Triadic Homophily Associated with Risky Prescribing

Models with any combination of the network statistics that are dyadic independent can be estimated straightforwardly. However, triadic terms introduce statistics that induce dependence across dyads (this is seen from the fact that changing the status of one of the three dyads comprising a triad restricts the possible statuses of the other two dyads). To overcome model degeneracy encountered when including triadic terms, we computed two triadic statistics that are restricted through the involvement of attribute information: 1) the proportion of closed triangles with the same node attribute Tri_1 , and 2) the proportion of open two-paths (2-stars or open-triangles) with the same node attribute that are closed Tri_2 . Suppose $\mathbf{A} = [a_{ij}]$ is the adjacency matrix of the binary-undirected network, and $a_{ij} = 1$ if physician i and j shared at least one patient during 2014. Let $\{x_i, x_j, x_k\}$ denote the attribute of nodes i , j , and k . For a binary node attribute taking the value of 0 or 1, the statistic Tri_1 is defined as,

$$Tri_1 = \frac{\sum x_i x_j x_k \cdot a_{ij} a_{jk} a_{ki}}{\sum a_{ij} a_{jk} a_{ki}}. \quad (2)$$

The statistic Tri_2 is defined as,

$$Tri_2 = \frac{\sum x_i x_j x_k \cdot a_{ij} a_{jk} a_{ki}}{\sum x_i x_j x_k \cdot a_{ij} a_{ik}}. \tag{3}$$

If x_h denotes whether physician h has contributed to risky prescribing, Tri_1 is the proportion of times that three physicians who shared patients among themselves all contributed to risky prescribing. The interpretation of Tri_2 is the proportion of triads in the 2-star with physician i as the apex (an undirected path of length 2 from j to k via i) that are closed (physician j and k also shared patients) among those for which nodes i, j , and k are all risky prescribers. Thus, Tri_2 can be viewed as an attribute-restricted version of node transitivity [18]; see Fig. 1 for illustration of Tri_1 and Tri_2 .

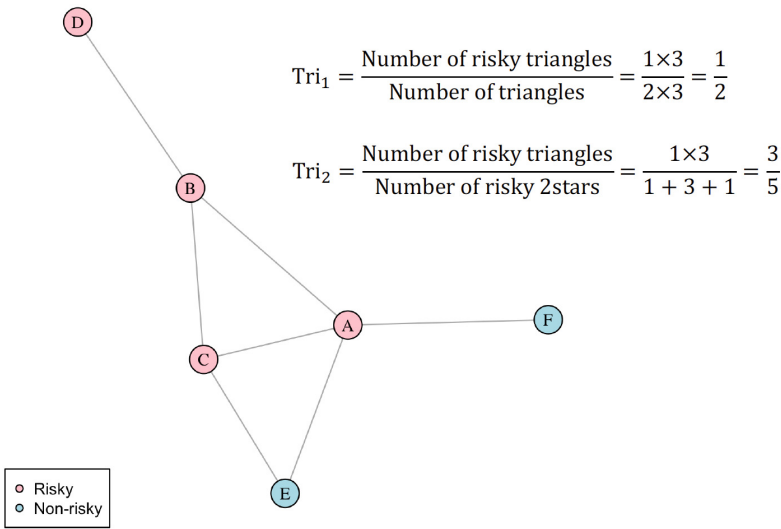


Fig. 1. Diagram of computing triadic homophily statistics Tri_1 and Tri_2 in an example network. Suppose nodes A, B, C, and D are physicians who have contributed to risky prescribing, and nodes E and F are non-risky-prescribing physicians. The number of risky 2-stars with nodes A, B, C, and D being the center vertex is 1, 3, 1, and 0, respectively. Therefore, the total number of 2-stars among risky prescribing physicians is five.

2.4 Non-parametric Test for Triadic Homophily

The numerator and denominator in Eq. 2 are available as ERGM terms in the statnet package [15]. However, to the best of our knowledge, the ratio of them is not. Similarly, the denominator in Eq. 3, the total number of 2-stars among nodes with a certain attribute, is a statistic that is not directly available in

statnet. Therefore, we develop a non-parametric test based on randomly re-distributing the node attribute in question across the nodes. The test preserves the total number of nodes, the number of nodes with a certain attribute, and the structure of the observed network. In each test, we repeatedly re-assigned the attribute of interest at random to the nodes across the network 30 times. On each permuted-attribute network we computed Tri_1 and Tri_2 to form a null distribution of what is expected by chance under the null hypothesis of no homophily in the given attribute conditional on the structure of the network. We compared the resulting distributions with the corresponding observed values and computed a p-value as a measure of statistical significance. These two triadic homophily statistics generalize to continuous attributes standardized to have a range from 0 and 1. All the analyses were performed using Python 3.7 and R. [25,31]

3 Application to Study of Homophily in Physician Prescribing and Deprescribing

We applied the general methodology in Sect. 2 to study the homophily of risky prescribing in our physician shared-patient network. We constructed novel measures to quantify physician risky prescribing based on Medicare Part D data from 2014 that involved prescriptions of the aforementioned three drug classes of interest, including opioids (O), benzodiazepines (B), and non-benzodiazepine sedative-hypnotics (S). A series of indexes were computed for each individual physician to reflect their involvement in risky prescribing or deprescribing, including 1) I_{OBS} : the extent of a physician’s involvement in simultaneously prescribing drugs in each of the three risky drug classes to patients, 2) $I_{everOBS}$: the binary counterpart of I_{OBS} , 3) $I_{presc2mr}$: a physician’s contribution to simultaneously prescribing two or more drugs to patients and its deprescribing counterpart $I_{depresc2mr}$. The supplemental online appendix provides more details about data preprocessing for the risky-prescribing analyses, modeling patient drug status as a state space process, algorithms for attributing physicians to prescribing and deprescribing events based on patient drug status change, and mathematical specifications and contextual descriptions of the risky prescribing indices.

4 Results

4.1 Physician Shared-Patient Networks

The Ohio shared-patient physician network we constructed consists of 35,765 physicians who had clinical encounters with patients residing in Ohio in 2014 identified from Medicare fee-for-service claims. After linking physicians in this Ohio shared-patient network to their prescribing measures identified from Medicare Part D data, 22,655 physicians were included in the Ohio shared-patient prescribing network. Thus, approximately 63% of physicians in the Ohio shared-patient physician network were identified as prescribers of at least one opioid,

benzodiazepine, or sedative-hypnotic, and around half of the ties in the network took place among the prescribers indicating that the rate of risky prescribing is likely to be highly prevalent.

The largest connected component (LCC) of the Ohio shared-patient prescribing network contains 17,363 physicians, amounting to more than 76% of physicians and more than 98% of the ties in the full network. The prescribing network and its LCC were similar in terms of network statistics and physician prescribing measures, except that the physicians in the LCC had a slightly higher average node degree (hence, density was substantially greater in the prescribing network) and number of distinct Ohio patients encountered annually.

Table S2 in the supplemental online appendix provides a more detailed account of the network statistics of the Ohio shared-patient physician network, the prescribing network, and the LCC of the prescribing network. Descriptive statistics about the prescribing-deprescribing measures are in Section 3.1 and Figure S3 of the supplemental online appendix while Figure S4 is graphical depiction of the association between physician network position and involvement in certain types of risky prescribing.

For the HRR sub-network analyses, only the 12 HRR sub-networks with at least 100 physicians were included; 100 was the smallest network size for which prescribing behavior could be measured stably for all physicians in the network.

4.2 ERGMs for Adjusted Homophily

Table 1 shows estimated ERGM-adjusted homophily effects in the LCC of the shared-patient prescribing physician network. When controlling for network density and the main effects of nodal prescribing and deprescribing attributes, the network exhibited assortative patterns in terms of different prescribing measures. An overall state-wide homophily effect was found among physicians in terms of whether they have ever contributed to bringing patients to the OBS state ($est. = 0.037$ (odds-ratio of 1.038), $p < 0.001$). Physicians with a larger difference in their likelihood of bringing patients to OBS were less likely to be connected to each other ($est. = -1.200$ (odds ratio of 0.301), $p < 0.001$). A larger difference in the likelihood of prescribing two or more drugs to patients at once was associated with a lower likelihood of a tie between physicians ($est. = -0.619$ (odds-ratio of 0.538), $p < 0.001$). Physicians were also less likely to form ties with each other if there was an increased difference in their propensity to deprescribe two or more drugs ($est. = -0.203$ (odds-ratio of 0.816), $p < 0.01$). Because such a high proportion of the physicians in Ohio prescribed at least one risky drug and the physicians in the LCC dominate the prescribing network, these effects are clinically and statistically significant. Physicians' propensity to form ties with other physicians of the same specialty was consistent across models including different prescribing measures. Compared to primary care physicians, emergency medicine physicians, neurologists, and psychiatrists were less likely to have connections with other physicians in the network. After controlling for the main effect of physician specialties, primary care physicians and neurologists

were less likely to connect to peers in the network of the same specialty. In contrast, emergency medicine physicians and psychiatrists were more likely to form ties with physicians of the same specialty.

Table 1. ERGM adjusted homophily effects for models estimated on the largest connected component of the Ohio 2014 shared-patient physician prescribing network.

	Model 1			Model 2			Model 3			Model 4		
	Est.	SE	p	Est.	SE	p	Est.	SE	p	Est.	SE	p
Edges	-5.732	0.012	***	-5.614	0.006	***	-5.609	0.006	***	-5.620	0.006	***
Node attribute Prescribing												
<i>Binary</i>												
$I_{everOBS} = 1$	0.327	0.010	***									
<i>Continuous</i>												
I_{OBS}				1.202	0.109	***						
$I_{presc2mr}$							0.492	0.049	***			
$I_{depresc2mr}$										0.337	0.066	***
Specialty (ref. PC)												
EM	-0.584	0.006	***	-0.614	0.006	***	-0.614	0.006	***	-0.611	0.006	***
Neuro	-0.338	0.010	***	-0.358	0.010	***	-0.359	0.010	***	-0.357	0.010	***
Psych	-0.668	0.009	***	-0.654	0.009	***	-0.653	0.009	***	-0.651	0.009	***
Other	-0.396	0.004	***	-0.423	0.005	***	-0.423	0.005	***	-0.420	0.005	***
Prescribing homophily												
<i>Binary</i>												
$I_{everOBS} = 1$	0.037	0.011	***									
<i>Continuous</i>												
absdiff(I_{OBS})				-1.200	0.114	***						
absdiff($I_{presc2mr}$)							-0.619	0.054	***			
absdiff($I_{depresc2mr}$)										-0.203	0.068	**
Specialty homophily												
PC	-1.509	0.009	***	-1.509	0.009	***	-1.511	0.009	***	-1.509	0.009	***
EM	0.541	0.019	***	0.541	0.019	***	0.541	0.019	***	0.541	0.019	***
Neuro	-0.190	0.091	*	-0.190	0.091	*	-0.190	0.091	*	-0.190	0.091	*
Psych	0.673	0.050	***	0.670	0.050	***	0.674	0.050	***	0.673	0.050	***

Note: The node attribute term and the homophily term associated with the attribute were added one at a time in the model for each of the prescribing or deprescribing indexes, yielding five separate models. In each model, physician specialty and homophily of physician specialty (restricted to homogeneous effects across the specialties) were included in the model. Absdiff is the ERGM term for examining the homophily of a continuous node attribute, with a negative estimate indicating homophily (smaller differences imply a higher likelihood of a network connection). Abbreviations: PC = primary care, EM = emergency medicine, Neuro = neurology, Psych = psychology. Significance levels: *** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$.

At the HRR-level, 6 out of 12 HRRs show significant homophily for the risky prescribing index, I_{OBS} , and 10 of them show significant homophily for the index quantifying prescribing intensity, $I_{presc2mr}$ (Table 2). Further, the scale of homophily varies across the HRRs. For other prescribing or deprescribing

indexes, homophily is not as significant nor prevalent as at the state level. The discrepancy of prescribing-associated homophily between the state and HRR levels, especially the homophily found at the state-level but not in some of the HRRs, may indicate that some prescribing clusters at the state-level rely on cross-HRR physician patient-sharing.

Table 2. ERGM adjusted homophily effects in HRR shared-patient sub-networks in 2014.

Homophily effects of indexes										
Descriptive Stats			absdiff(I_{OBS})		absdiff($I_{presc2mr}$)		absdiff($I_{depresc2mr}$)		$I_{everOBS}$	
HRR	N	Density	Coef.	SE	Coef.	SE	Coef.	SE	Coef.	SE
180	129	0.116	-3.998	3.662	0.004	1.623	1.052	1.898	0.026	0.133
357	193	0.106	0.810	2.707	-1.541*	0.714	0.467	0.775	-0.268	0.149
331	256	0.128	-5.156	2.675	-1.792*	0.697	-0.325	0.773	0.047	0.117
332	415	0.048	-1.765*	0.745	-0.863***	0.250	0.299	2.410	0.056	0.079
335	550	0.060	-1.887*	0.920	-0.881**	0.305	-0.545	0.416	0.063	0.070
326	648	0.050	-1.281	0.795	-1.066***	0.306	44.210	280.321	-0.080	0.056
325	750	0.030	-0.469	0.814	-0.917**	0.279	0.496	0.699	0.0002	0.089
334	1039	0.029	-0.532	0.416	-1.190***	0.226	-0.301	0.209	0.018	0.045
330	1164	0.024	-1.205**	0.419	-0.339	0.196	-0.071	0.319	0.0002	0.037
327	1760	0.015	-1.711**	0.584	-0.783***	0.179	-0.362*	0.171	0.120**	0.044
328	2623	0.010	-1.603***	0.370	-0.897***	0.142	-0.193	0.227	0.060	0.034
329	3101	0.008	-1.181***	0.234	-0.754***	0.109	-0.327*	0.157	-0.018	0.025

Note: The HRR sub-networks were partitioned from the largest connected component of the Ohio 2014 shared-patient physician prescribing network and the HRR sub-networks were not restricted to their respective largest connected components thus they may not be fully connected. Significance levels: *** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$.

4.3 Triadic-Level Hyper Homophily

The triad-level risky prescribing indices evaluated through the involvement of $I_{everOBS}$ in Tri_1 and Tri_2 are 0.0015 and 0.0007, respectively (Fig. 2). The interpretation of the index for Tri_1 is that among 10,000 closed triangles (three nodes fully connected with one another), 15 of them include nodes with the same attribute (each physician contributed to bringing at least one patient to state OBS). The interpretation of Tri_2 is that among 10,000 open two-paths (2-stars) with the same node attribute ($I_{everOBS}$) in the network, 7 of them are closed. By the attribution re-distribution test, the values of Tri_1 and Tri_2 in the observed network are significantly higher than expected ($p = 0.000$). These results suggest that 1) when three physicians share patients among themselves, they are more likely to all be involved in risky prescribing than by chance; and 2) when two physicians share patients with a common third physician, and all three of them have been involved in risky prescribing, the two physicians are more likely to also share patients between them than by chance. These results

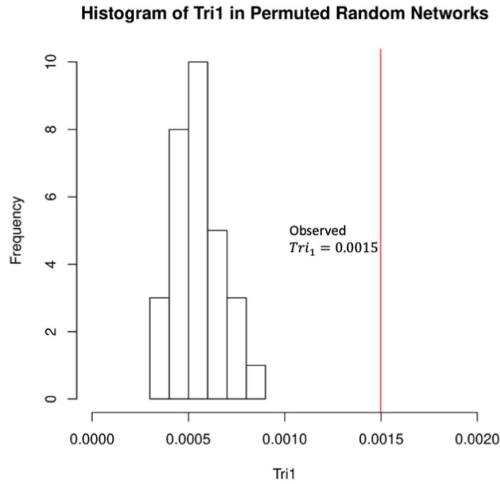
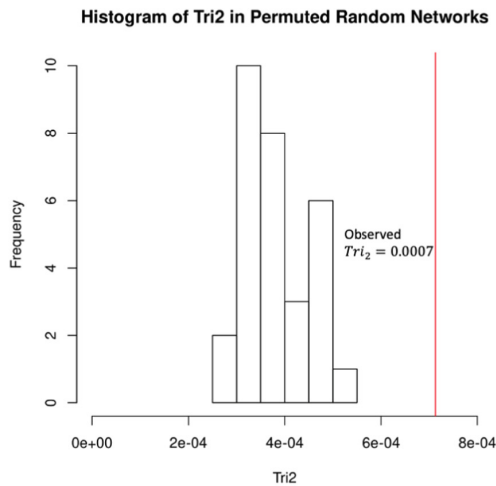
(a) Triadic homophily statistic Tri_1 (b) Triadic homophily statistic Tri_2

Fig. 2. Histogram of triadic homophily network statistics generated by the triadic homophily non-parametric test. The triadic homophily statistic Tri_1 is the proportion of closed triangles with the $I_{everOBS}$ node attribute (whether a physician has ever contributed to bringing patients to the riskiest prescription state OBS) in the network. The triadic homophily statistic Tri_2 is the proportion of open two-paths with all nodes having the same attribute that are closed in the network. Panel (a) is the histogram of Tri_1 and panel (b) is the histogram of Tri_2 calculated from 30 networks with randomly shuffled node attributes under the null hypothesis of no homophily with respect to the given prescribing index. The red vertical lines denote the values in the observed network.

further demonstrate the importance of homophily of risky prescribing and its intersectionality with triadic clustering net of dyadic-level homophily.

5 Conclusions

We developed a framework to quantify physicians' prescribing and deprescribing behaviors comprehensively and studied the homophily associated with prescribing in a shared-patient physician network. We discovered substantial homophily of prescribing behaviors among physicians, as well as assortative and disassortative mixing patterns associated with physician specialties in the prescribing network. We also found a level of triadic-level risky-prescribing homophily in the observed network statistically significantly greater than expected by chance. We found that physicians' level of involvement in prescribing and deprescribing varied across specialties and that there was heterogeneity in the level of prescribing-associated homophily across HRRs.

Our findings related to the homophily associated with physician prescribing behavior and their specialty in a complex shared-patient physician network provides a basis for promoting guideline-concordant prescribing practice and informing interventions. Previous literature revealed that physicians were more likely to share patients with those having similar traits, patient panels, and institutional affiliations. [17, 19] Our results add to this literature by demonstrating the influence of prescribing preference on the propensity of sharing patients and attempting to reveal the mechanism underlying the formation of prescriber communities. Previous literature suggests that homophily in professional networks may hinder the diffusion of innovations but may also promote healthcare consistency. [17] Homophily can be a roadblock to reducing non-compliant prescribing among heavy prescribers. The act of sharing patients can be a channel for behavior changes and so physicians who only share patients with risky prescribers might expose the focal physician to so much high-risk behavior that their own practice changes, forming a loop of reinforced problematic prescribing. Given this homophily-driven potential reinforcement of influence between physicians, [6] external interventions may be warranted to help break the cycle of risky prescribing among communities of guideline non-concordant prescribers.

Even without identifying the precise mechanism underlying observed clustering on risky prescribing, the detection of observed homophily patterns has its own merits. For example, the knowledge that homophily exists would motivate efforts to discover peer physicians defined according to shared-patient network ties or otherwise following identification of a risky prescribing candidate for potential intervention. Such efforts, which can be thought of as link tracing designs, are likely to identify more potential intervention candidates more efficiently, which may be critical given limited resources and budgets. Our finding of the variation in prescribing-associated homophily across HRRs also adds to previous literature on the geographic variability in healthcare utilization and outcomes and the use of physician patient-sharing network characteristics to provide new insights into previously unexplained variation [17].

Another methodological contribution is the introduction of two measures of triadic homophily that mitigate the common degeneracy issues encountered in ERGMs when dyad dependence is imposed by including triadic terms. These two measures elevate homophily from the dyad to the triad level, defining a form of super-homophily. Although we use a non-parametric random redistribution (partial permutation) test to compare the observed statistics to those expected by chance, this does not account for other network phenomena. One avenue for future research is to embed these terms in ERGM software packages to enable their adjusted effects net of other predictors to be easily estimated.

This study is subject to several limitations. First, the data used in this study was cross-sectional, which led to challenges in accounting for triadic dependence in the network. The availability of longitudinal network data would have allowed dyadic dependent network effects to be modeled as lagged variables, avoiding degeneracy [23], and helped distinguish social selection (i.e., the factors governing the selection of relationships [28]) from social influence (i.e., the influence of individuals on one another). Second, our study focused on the Medicare population, whereas the same research question within younger populations is also of interest. Thirdly, it is important to appreciate that not all risky prescribing is bad prescribing. There are instances in which the prescribing of risky drugs from one or multiple classes is warranted. Distinguishing these from those instances in which the prescribing is unnecessarily risky is challenging using claims data alone (the additional of electronic medical record data would help to determine the appropriateness of prescriptions and prescription regimes by providing more insights into the condition of a patient leading up to a prescription being written). Although the prescribing indices we studied are technically measures of physicians' involvement in potentially risky prescribing, not binary indicators of whether physicians intentionally conducted risky prescribing or not, the extent of the risky prescribing observed in these data far exceeds what could be considered reasonable (combinations of drugs such as OB, OS, and OBS should almost never be prescribed together).

In summary, we proposed a novel framework to model the relationship between physician professional networks and developed new measures for quantifying physicians' prescribing and deprescribing behavior. We found that homophily was associated with prescribing among physicians' connections measured through sharing patients. These findings provide important insights into the spread of risky prescribing among the older population in the United States and how communities of prescribers emerge and evolve, helping incentivize interventions to reduce guideline-non-compliant practices and promote safe practices among healthcare providers.

Supporting Information. The supplemental online appendix, tables, and figures referred to in the main text are available at the GitHub site: <https://github.com/xinran02/PolyRxNetworkHomophily/blob/main/Appendix>

References

1. An, C., O'Malley, A.J., Rockmore, D.N., Stock, C.D.: Analysis of the U.S. patient referral network. *Stat Med.* **37**(5), 847–866 (2018)
2. An, C., O'Malley, A.J., Rockmore, D.N.: Referral paths in the us physician network. *Appl. Netw. Sci.* **3**(1), 1–24 (2018)
3. Aroke, H., Katenka, N., Kogut, S., Buchanan, A.: Network-based analysis of prescription opioids dispensing using exponential random graph models (ERGMs). In: Benito, R.M., Cherifi, C., Cherifi, H., Moro, E., Rocha, L.M., Sales-Pardo, M. (eds.) *COMPLEX NETWORKS 2021*. SCS, vol. 1073, pp. 716–730. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-93413-2_59
4. Barnett, M.L., Christakis, N.A., O'Malley, A.J., Onnela, J.P., Keating, N.L., Landon, B.E.: Physician patient-sharing networks and the cost and intensity of care in US hospitals. *Med. Care* **50**(2), 152 (2012)
5. Bushardt, R.L., Massey, E.B., Simpson, T.W., Ariail, J.C., Simpson, K.N.: Polypharmacy: misleading, but manageable. *Clin. Interv. Aging* **3**(2), 383–389 (2008)
6. Centola, D.: An experimental study of homophily in the adoption of health behavior. *Science* **334**(6060), 1269–1272 (2011)
7. Cho, J., Spence, M.M., Niu, F., Hui, R.L., Gray, P., Steinberg, S.: Risk of overdose with exposure to prescription opioids, benzodiazepines, and non-benzodiazepine sedative-hypnotics in adults: a retrospective cohort study. *J. Gen. Intern. Med.* **35**(3), 696–703 (2020)
8. Contractor, N.S., Wasserman, S., Faust, K.: Testing multitheoretical, multilevel hypotheses about organizational networks: an analytic framework and empirical example. *Acad. Manag. Rev.* **31**(3), 681–703 (2006)
9. Centers for Disease Control and Prevention, et al.: Guideline for prescribing opioids for chronic pain. *J. Pain Palliative Care Pharmacother.* **30**(2), 138–140 (2016)
10. Dreischulte, T., Guthrie, B.: High-risk prescribing and monitoring in primary care: how common is it, and how can it be improved? *Ther. Adv. Drug Saf.* **3**, 175–184 (2012)
11. Fattore, G., Frosini, F., Salvatore, D., Tozzi, V.: Social network analysis in primary care: the impact of interactions on prescribing behaviour. *Health Policy* **92**(2–3), 141–148 (2009)
12. Fulton, M.M., Riley Allen, E.: Polypharmacy in the elderly: a literature review. *J. Am. Acad. Nurse Pract.* **17**(4), 123–132 (2005)
13. Gnjdjic, D., et al.: High risk prescribing in older adults: prevalence, clinical and economic implications and potential for intervention at the population level. *BMC Public Health* **13**(1), 1–9 (2013)
14. Goodreau, S.M.: Advances in exponential random graph (p^*) models applied to a large social network. *Soc. Netw.* **29**(2), 231–248 (2007)
15. Handcock, M.S., Hunter, D.R., Butts, C.T., Goodreau, S.M., Morris, M.: Statnet: software tools for the representation, visualization, analysis and simulation of network data. *J. Stat. Softw.* **24**(1), 1548 (2008)
16. Handcock, M.S., Robins, G., Snijders, T., Moody, J., Besag, J.: Assessing degeneracy in statistical models of social networks. Center for Statistics and Social Sciences Working paper #39 (2003)
17. Landon, B.E., et al.: Variation in patient-sharing networks of physicians across the united states. *JAMA* **308**(3), 265–273 (2012)

18. Latapy, M., Magnien, C., Del Vecchio, N.: Basic notions for the analysis of large two-mode networks. *Soc. Netw.* **30**(1), 31–48 (2008)
19. Mascia, D., Di Vincenzo, F., Iacopino, V., Fantini, M.P., Cicchetti, A.: Unfolding similarity in interphysician networks: the impact of institutional and professional homophily. *BMC Health Serv. Res.* **15**(1), 1–8 (2015)
20. Moen, E.L., Austin, A.M., Bynum, J.P., Skinner, J.S., O'Malley, A.J.: An analysis of patient-sharing physician networks and implantable cardioverter defibrillator therapy. *Health Serv. Outcomes Res. Method.* **16**(3), 132–153 (2016)
21. Morris, M., Handcock, M.S., Hunter, D.R.: Specification of exponential-family random graph models: terms and computational aspects. *J. Stat. Softw.* **24**(4), 1548 (2008)
22. O'Malley, A.J., Ran, X., An, C., Rockmore, D.N.: Optimal physician shared-patient networks and the diffusion of medical technologies. *J. Data Sci.* **21** (2022)
23. Paul, S., O'Malley, A.J.: Hierarchical longitudinal models of relationships in social networks. *J. Roy. Stat. Soc. Ser. C (Appl. Stat.)* **62**(5), 705–722 (2013)
24. Pollack, C.E., Soulos, P.R., Gross, C.P.: Physician's peer exposure and the adoption of a new cancer treatment modality. *Cancer* **121**(16), 2799–2807 (2015)
25. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2022). <https://www.R-project.org/>
26. Robins, G., Pattison, P., Kalish, Y., Lusher, D.: An introduction to exponential random graph (p^*) models for social networks. *Soc. Netw.* **29**(2), 173–191 (2007)
27. Robins, G., Snijders, T., Wang, P., Handcock, M., Pattison, P.: Recent developments in exponential random graph (p^*) models for social networks. *Soc. Netw.* **29**(2), 192–215 (2007)
28. Runciman, W.G., et al.: *The Theory of Cultural and Social Selection*. Cambridge University Press, New York (2009)
29. Snijders, T.A., Pattison, P.E., Robins, G.L., Handcock, M.S.: New specifications for exponential random graph models. *Sociol. Methodol.* **36**(1), 99–153 (2006)
30. Sun, E.C., Dixit, A., Humphreys, K., Darnall, B.D., Baker, L.C., Mackey, S.: Association between concurrent use of prescription opioids and benzodiazepines and overdose: retrospective analysis. *BMJ* **356** (2017)
31. Van Rossum, G., Drake, F.L.: *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA (2009)
32. Werder, S.F., Preskorn, S.H.: Managing polypharmacy: walking the fine line between help and harm. *Curr. Psychiatr. Online* **2**(2), 24–36 (2003)



Focal Structures Behavior in Dynamic Social Networks

Mustafa Alassad^(✉) and Nitin Agarwal

COSMOS Research Center, UA-Little Rock, Little Rock, AR, USA
{mmalassad, nxagarwal}@ualr.edu

Abstract. The expansion of coordinating communities via focal information spreaders on online social networks has attained much-needed attention over the past few years. Several methods have been applied to investigate the influential communities of information spreaders in static social networks. However, investigating static social networks does not entirely reflect the activities and the dynamics of evolving communities over time. Researchers have applied advanced operational methods such as game theory and evolving complex graphs to describe the change in the regular communities in dynamic social networks. Yet, these methods need the ability to describe the focal information spreaders in dynamic social networks. For this purpose, in this research, we propose a systematic approach to measure the influence of focal information spreaders and track their evolution in social networks over time. This novel approach combines the focal structure analysis model and the adaptation algorithm to identify the coordinating communities of information spreaders in social networks and illustrate their development in the network over time, respectively. We evaluate our findings using a real-world dynamic Twitter network collected from the Saudi Arabian women's Right to Drive campaign coordination in 2013. The outcomes of this approach allow observing, predicting, tracking, and measuring the coordination among the focal information spreaders over time. Correspondingly, this approach investigates and illustrates when the information spreaders will escalate their activities, where they concentrate their influence in the network, and what coordinating communities of spreaders are more tactical than others in the network.

Keywords: Dynamic Social Networks · Focal Structure Analysis · Adaptation Algorithm · Betweenness Centrality · Modularity Method · focal information spreaders

1 Introduction

Millions of people worldwide use social platforms like Facebook, Twitter, and Instagram to communicate, shop, trade, advertise, book flights, and catch up with relatives, friends, and co-workers. The widespread use of such platforms welcomes all users, communities, and agencies to share public announcements and breaking news to quickly influence the maximum number of users. However, in the past few years, most online social networks (OSN) have witnessed, discovered, and suffered from coordinated online users

campaigns spreading massive amounts of information across networks and mobilizing crowds. For example, we witnessed that online campaigns were able to spread misinformation and fake news and damage numerous economic systems around the world, as reported in [1], seen in influenced political and election campaigns [2], and evident recently in the volatility of the stock markets [3]. The stock markets' volatility and fluctuations in the price of GameStop (GME) stocks is a perfect example demonstrating that coordinated campaigns on social networks have a crucial impact on the real-world market and life. Moreover, such campaigns, now being termed as OccupyWall 2.0, started on Reddit and quickly gained traction, leading hordes of redditors to buy and sell the stock in a coordinated fashion [3]. Likewise, in a different event, a group of online users on Twitter organized an armed movement against the COVID-19 lockdown in Michigan state in May 2020 [4]. In this event, Twitter was used to coordinate the date and time of the protest.

In response to these ongoing problems in social networks, several researchers [5, 6] have attempted to identify and limit the activities of focal information spreaders (influential sets of users coordinating to spread information) on Twitter, Facebook, and other platforms. These studies used traditional community detection methods such as the centrality method to locate the influential users and the modularity method to explore patterns of users/communities in complex OSNs [7]. However, studies in [5, 6] aggregated temporal local and global social interactions into one static snapshot, ignoring the development of the communities and the dynamic aspects of the social networks over time [8]. Also, many other scholars studied and clustered the regular communities in dynamic social networks, as presented in Sect. 2, but our goal in this research is to study the behavior of the focal information spreaders in dynamic social networks and present their influence over time.

For this purpose, the research proposed in this paper considers that all regular users/communities are evolving and that their behavior changes over time. In addition, the model presented here identifies focal information spreaders, measures their influence and development over time, and estimates when they will disappear from the network after serving their purpose. Moreover, the dynamic analysis of these focal information spreaders unveils their ability to change in size and space from one time period to another, making analysis a complex and intrinsically dynamic process. To investigate such NP-hard problems [7], the main contributions and challenges within this research are as follows:

- to identify and track focal information spreaders in evolving real-world social networks.
- to measure, record, illustrate, and investigate the growth/shrink of focal information spreaders in evolving real-world social networks.

To overcome the above challenges, the main objective of this research is to create a systematic approach that integrates the focal structure analysis model presented in [9], which identifies the focal sets spreading information in dynamic social networks, with the adaptation algorithm presented in [10], which spots the behaviors of focal information spreaders over time. The resultant approach is an integrated systematic model that utilizes the decomposition optimization model and adaptation method to project the development of focal information spreaders over time.

The rest of the paper is organized as follows. Section 2 reviews the related works. Section 3 describes the proposed methodology. Section 4 explains the experimental results. Section 5 is the research conclusion and future work.

2 Related Work

The main point of interest is modeling the focal sets' behavior and their growth in dynamic OSNs. Currently, most of the implemented community detection methods were deployed using static OSNs, as presented below.

Şen et al. (2016) introduced the focal structure analysis model into OSNs; the authors identified the smallest possible influential sets of online users that were able to mobilize crowds, spread information to thousands of users, and organize protest campaigns through social networks [5]; the authors proposed a greedy algorithm to identify the focal structures responsible for spreading information in static Facebook and Twitter networks. Also, the authors stated that these influential sets could not be discovered by regular community detection algorithms such as HITS [11], PageRank [12] or centrality methods [5]. Alassad et al. (2019) presented a bi-level centrality-modularity model to examine the intensive groups of co-commenters spreading information in a static YouTube network; the authors explored the hidden focal groups of commenters and ranked them for further investigations [6]. Authors in [2] examined the key information spreaders in complex static social networks: they designed a decomposition optimization method to reveal the focal sets of users influencing other users. In an extended study, Alassad et al. (2021), used a computational social science technique to identify coordinated groups spreading information on social networks about the smart cities' infrastructure. In this research, the authors implemented a static Twitter network to measure the model's applicability [1]. The model explored the intensive sets of spreaders and measured their power to influence other users across the network. The authors in [9] implemented a comprehensive decomposition optimization model for locating the key sets of commenters that spread information in the static social networks. Alassad et al. (2021) studied the computational social science techniques to identify coordinated actions on social networks; the authors explained the behavior toward the smart cities' infrastructure from the influential coordinating users on social networks [13]. Shajari et al. (2023) studied the commenter behavior characterization on different YouTube channels to identify suspicious behavior [14].

However, the methods mentioned were applied only to static social networks. But, in this research we focus on the dynamic social networks, the behavior of focal information spreaders in the evolving social networks, and their development and activities in the network over time, and we project the influence of focal information spreaders across the social networks over time.

Moreover, many other studies have investigated the influential nodes in large-scale networks, their resources in such networks, and what positions they occupy in the structure of such networks. The authors in [15] studied when influential bloggers were able to impact other bloggers and explored the challenges of identifying such influential nodes in communities. Blondel et al. (2008) studied the fast unfolding of communities in large networks by using the modularity method [16]. Xu et al. (2020) investigated the development of communities—the new and easy ways of social mass movements and flash

mobs influenced by social networks [17]. Chen et al. (2017) used label propagation to identify communities and the direct neighbor relationship in social networks [18]. However, none of the mentioned papers considered the dynamicity of communities and the evolution of the network over time.

In addition, Zhu et al. (2020) designed two vital node algorithms to identify the influential nodes in complex networks [19]; the authors used the nodes' removal technique to measure network changes. Kitsak et al. (2010) used the K-shell decomposition method to identify key spreaders in complex networks, where they found that influential spreaders occupied positions in the core of the network's structure [20]. Chen et al. (2012) identified influential nodes in complex networks; the authors proposed a semi-local centrality method to overcome the gap in the analysis of the betweenness and closeness centrality methods and to identify influential nodes in complex networks [21]. However, none of the mentioned methods studied the evolution of the influential nodes in dynamic networks or measured their development over time.

Nonetheless, many scholars have excessively investigated regular community detection in dynamic social networks. Alvari et al. (2014) applied a game theory method to measure an agent's utilities over time and captured the regular communities in dynamic networks [22]. Dakiche et al. (2019) identified two types of growth for a regular community in dynamic networks. The authors defined community diffusion growth as attracting new members through ties to existing members [23]. Their second definition referred to the non-diffusion growth communities, where individuals with no prior ties become part of other communities. Dakiche et al. (2019) predicted the lifespan of a regular community in dynamic networks based on a consistent set of structural features [24], where the authors extracted some features of the communities from the profiles of the users and communities to predict the dynamic lifespan. Takaffoli et al. (2014) implemented a similarity function to match the regular communities in dynamic networks from one time step to another to detect changes such as merging, splitting, dissolving, and surviving [25]. Bródka et al. (2012) modeled a classifier to discover events in dynamic social networks, where the authors utilized the changes between snapshots to measure the patterns over time [26].

The studies mentioned above were implemented to study and cluster the regular communities in dynamic social networks; however, our goal in this research is to study the behavior of focal information spreaders in dynamic social network and to present their influence over time.

3 Methodology

The main objective in this research is to design a systematic approach that integrates the focal structure analysis presented in [9] and the adaptation method presented in [10], while relaxing the complexity of time dimension in the analysis, to identify and study the focal information spreaders in evolving real-world events on social networks.

Consider G is a network of an event on an online social network consisting of a set of S snapshots $S = \{1, 2, \dots, n\}$, where $G = \{G_1, G_2, \dots, G_n\}$ and time $T = \{t_1, t_2, \dots, t_n\}$. Each snapshot is $G_i = (V_i, E_i, T_i)$ and represents the snapshot S_i with total number of nodes $|V_i|$, and total number of edges $|E_i|$ at time t_i . Each snapshot G_i

is used to represent a connected social network at time t_i . In other words, (V_i, E_i, T_i) is when users (v_i, v_j) mention or retweet each other while implementing the edge e_i at time t_i . Introducing the dynamic (temporal) focal structure analysis model presented in this research, given $G_i = G_F$ as shown in Fig. 1, we find the set K_{G_F} that can influence the maximum number of users and increase the information spread in G over time, or the focal sets of users in G_F , where $K_{G_F} = \{k_{1G_F}, k_{2G_F}, \dots, k_{jG_F}, k_{mG_F}\}$, and $j \leq m$.

3.1 Definitions

Focal Structure Analysis: As mentioned in Sect. 2, various studies are applied to identify the focal sets responsible for information diffusion in static social networks. The focal structure analysis model identifies the focal influential sets of users coordinating to spread information to the maximum number of users, mobilize crowds, and participate in different communities across the OSNs. The authors in [2, 6, 9] applied models to identify focal sets in static social networks; however, a static network is not sufficient to reveal the events and the development of the users/communities during a real-world event on OSNs. In this research, we present an extended focal structure analysis (FSA) model to identify and study the behavior of focal information spreaders as the network changes over time.

Adaptation Algorithms: Using the adaptive algorithm presented in [10] helps avoid recalculation methods and repeatedly measures all instances and changes in every snapshot in complex dynamic OSNs. The adaptive algorithm helps to overcome problems such as having expensive execution time, getting trapped in the local optimal solutions, and receiving the same reactions to tiny changes to inactive local communities in dynamic OSNs.

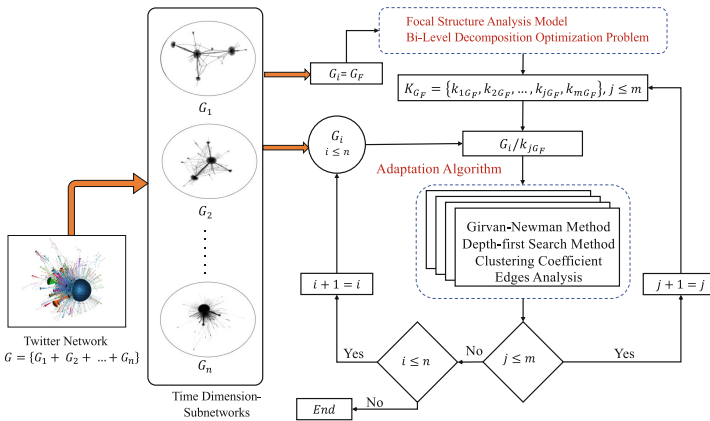


Fig. 1. Overall structure of the model.

In addition, the adaptive algorithm provides the ability to track the qualitative and quantitative changes generated by k_{jG_i} over time. Relatedly, the main advantages for implementing this method to the social network analysis are the following:

- This method is less computationally expensive compared to repeatedly recomputing the development of OSNs from scratch every snapshot.
- This method is less time consuming and avoids the difficulty of continuously recomputing the multitude of variables in every snapshot.
- This method can observe the local campaigns on OSNs and illustrate the significant transformation of the communities or the focal information spreaders over a long duration in dynamic OSNs.

The methodology presented in this research applies into any dynamic social networks as presented in the steps below:

Step 1: from the dynamic social network G , we selected $G_F \in G$ as shown in Fig. 1.

Step 2: The focal structure analysis model presented in [9] was applied to find $K_{G_F} \in G_F$. The resultant K_{G_F} , is the influential sets of users (focal information spreaders) in G_i , where the users in each set K_{jG_F} communicate to each other over time, and the communications should remain as the network evolves with each snapshot.

Step 3: Using the adaptation algorithm [10] and the methods stated in Sect. 3.1, the impacts of K_{G_F} to other snapshots $G_{F_{i \neq j}} \in G$ were measured.

3.2 Validation and Verification

This section introduces the methods used to validate the results of the model and quantitatively measure the impacts of focal information spreaders in dynamic social networks. The methods explained below should reveal guidelines about where, when, and which focal information spreaders are more active than others in dynamic social networks:

- The modularity method introduced by Newman-Girvan [27] and the adaptation method [10] implemented measure the changes on the communities' level in the dynamic network. We employed the modularity method to find the patterns of users in each snapshot before and after suspending each set of the focal information spreaders from $G_i \in G$. The adaptation method [10] employed illustrates, records, and compares the network's development in each snapshot before and after suspending the set of the focal information spreaders from G , as shown in Fig. 1.
- The depth-first search and linear graph algorithm [28] and the adaptation method [10] were implemented to measure the changes on the users' level in the dynamic network. The depth-first search algorithm [28] is used to measure the weakly connected users in social networks. This algorithm is employed to measure the weakly connected users in the network before and after suspending the sets of the focal information in each snapshot $G_i \in G$. The adaptation method [10] is used to for calculating the transformation of the users before and after suspending each focal information spreaders from G , as presented in Fig. 1.

Finally, a real-world dynamic Twitter network was implemented to verify the accuracy and applicability of the proposed approach.

4 Results

For the purpose of this research, the model presented here tracks the development of the focal information spreaders in all snapshots in G . The examination was applied to a network of an event on the Twitter platform related to the Saudi Arabian women’s collective actions of the “Oct26Driving” campaign network as shown in Fig. 2 [5].

4.1 Development of the Campaign on Twitter Over Time

In this section, we present the general aspects of the dynamic Twitter network as follows.

Dataset. The dataset used is a real-world event on Twitter network that was generated during the Saudi Arabia women’s activities to drive campaign in October 2013. The dataset was collected from Twitter using a Twitter API from Oct 9th to Oct 30th, 2013. The Oct26Driving dataset consists of 70,000 tweets posted from more than 100 countries as described in [5] and presented in Fig. 2. The network includes a large number of users/edges that evolved during the campaign as presented in Table 1, and the network was structured into 20 days (snapshots).

Figure 3 illustrates the growth of the network on Twitter, where this growth was based on the usage of dominant hashtags dedicated to the campaign such as ‘#oct26driving’ [5]. The dataset changes with respect to the users’ frequencies varying between hundreds to thousands of users’ tweets about the campaign per day. Likewise, the reader can observe that the campaign received more popularity after snapshot # 3, G_3 , on Oct. 13th as shown in Fig. 4. After this time slot, the number of users increased and then started to bend the curve on and after Oct. 28th, three days after the campaign day Oct. 25th.

Table 1. Twitter network statistics.

Min # of Users	413
Max # of Users	6933
Min # of Edges	461
Max # of Edges	8399

Figure 4 observes the users’ communications behavior over time, where all users went into massive activities, made frequent actions, and spread information about the event on Twitter after snapshot # G_3 , Oct. 13th. The clustering coefficient values measure the level of friendship between users and their neighbors in the network [7]. We implemented this method to observe the communications between users and their neighbors, where the reader can see the values highly increased on Oct. 20th, snapshot G_{10} , compared to Oct. 9th. In other words, the increase in the clustering coefficient values projects the increase of the communications between users a few days before the campaign day on Oct. 25th.

In addition, the average path length method defined [7] was implemented to measure how quickly the information transfers between users in the network. Figure 3 shows the increase in the transfer of information between users in the network over time. These



Fig. 2. Static Twitter Network.

values highly decreased compared to Oct. 9th, where the results indicated that speedier information was transferred in the network before Oct. 25th.

Likewise, the average path length values minimized on Oct. 18th through Oct. 20th, suggesting a surge in the users’ activities a week before Oct. 25th.

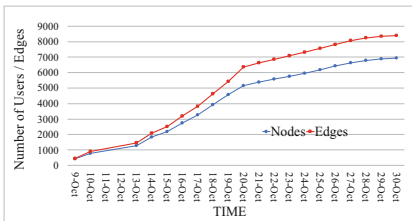


Fig. 3. Users and links change in an evolving social network over time.

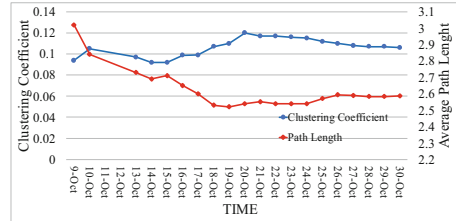


Fig. 4. The communication behavior of the users in an evolving Twitter network over time.

4.2 Focal Structure Analysis in Dynamic Networks

Step 1 in Sect. 3.1, mentions selecting a snapshot, and Step 2 then deploys upon the snapshot the focal structure analysis model presented in [9].

For this purpose, $G_F = G_1$ on Oct. 9th was selected. This snapshot includes 413 users and 461 edges. Alassad et al. (2021) had mentioned that Oct. 9th is when the campaign started spreading information on the Twitter network publicly and when the users began to coordinate with others to spread the words on Twitter extensively [9]. In addition, implementing G_1 has an advantage for the purpose of the focal structure analysis model, since the model focuses on identifying the active seed groups on a network and then tracing and analyzing the development of these groups on the Twitter network over time. Also, selecting an early snapshot helps to validate the model’s predictability feature, since the stakeholders can observe the focal information spreaders at the very beginning of the campaign’s life cycle. In addition, this process is a systematic method to limit the spread of information, rather than suspending random influential users from the network in different time windows. Furthermore, an early detection approach helps to track the focal sets’ evolutions and observe when they would merge with other communities and disappear from the network.

The focal structure analysis model in [9] initially identified $K_{G_1} = 13$ focal information spreaders in G_1 , where these focal sets consist of influential users and include users acting in different communities. For example, Fig. 5 presents the network on Oct. 9th before and after suspending focal set # 5 (k_{5G_1}) from the network. As presented, when suspending this focal set, (which included only 35 users, 8.5% of the total number of users on Oct. 9th and 0.5% of users on Oct. 28th) the network shifted from the connected and highly dense network shown in Fig. 5 (left side) into a completely disconnected and scattered network as presented in Fig. 5 (right side). In summary, the model apparently projects a large number of users disconnected from others, and the spread of the information was limited only to a few users instead of the whole network on Oct. 9th.

4.3 Validation and Evaluation in Dynamic Social Networks

Step 3 in Sect. 3.1 was to implement the adaptation algorithm to measure, present, and track focal spreaders' behavior changes in the activities over time.

For this purpose, the model suspended each focal information spreader K_{G_1} from each snapshot in G , as presented in Fig. 1 and explained in Sect. 3.1. The model recorded and compared the changes in the network G after suspending each focal information spreader in set K_{G_1} . In addition, we utilized other criteria, such as the Clustering Coefficient method [7] to measure and reflect other important changes in G , as presented below.

4.3.1 Changes in Clustering Coefficient Values Over Time

As part of step 3 in Sect. 3.1, the Clustering Coefficient method was utilized to provide robust information on the links between users and their neighbors in the network [7]. In this section, the adaptation algorithm was implemented to record the changes in the clustering coefficient over time.

Consequently, by definition, the focal structure sets are participating in different activities in different parts of the networks [5]; therefore, suspending each focal information set should disconnect a large number of links (edges) and decrease the connectivity of the users in each time slot in the dynamic network. In other words, after suspending any focal information spreaders, the clustering coefficient values would decrease dramatically compared to the values before appending any focal information spreaders as presented earlier in Fig. 3.

Figure 6 shows the changes in the clustering coefficient values after suspending each focal information spreaders K_{G_1} from other snapshots in G . In addition, the focal information spreaders (k_{4G_1} , k_{5G_1} , and k_{6G_1}) were able to decrease the clustering coefficient values more than other focal sets, scattering the larger complex communities into smaller more powerless groups and disrupting the connectivity of the users with their neighbors in G .

4.3.2 Changes in Modularity Values Over Time

The modularity method was implemented to observe the changes in the regular communities in each snapshot before and after suspending the focal information spreaders from G [7]. In other words, eliminating any focal information spreaders should scatter the

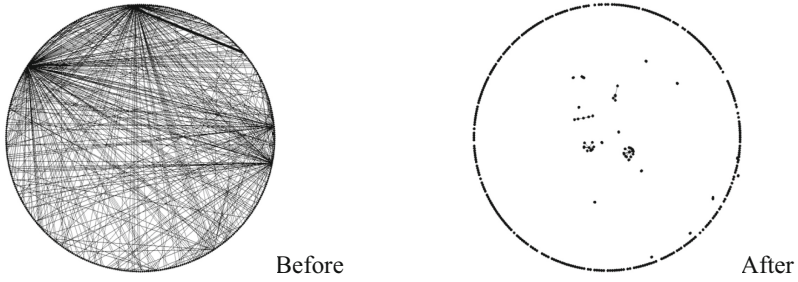


Fig. 5. Original Oct. 9th snapshot (left). Oct. 9th snapshot after suspending focal set # 5 (right).

network into smaller communities and increase the modularity values in each snapshot in G .

Figure 7 shows the changes in the modularity values after suspending each focal information spreaders K_{G_1} from other snapshots in G over time. Furthermore, the results show a huge increase in the modularity values in G compared to the values presented in Fig. 3.

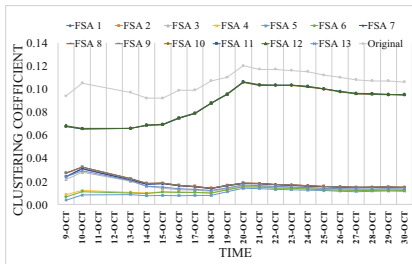


Fig. 6. Changes in the communication behavior of the users after suspending K_{G_1} from G .

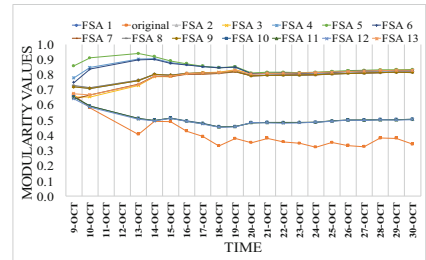


Fig. 7. Changes in the modularity values and behavior of the communities after suspending K_{G_1} from G .

Moreover, Fig. 8 shows that the focal information sets # (k_{4G_1} , k_{5G_1} , and k_{6G_1}) increased the modularity values in G more than other sets.

4.3.3 Changes in Network’s Edges Over Time

In this section, we show the changes in the number of edges between users after suspending each focal information spreaders from G . The adaptation algorithm and the depth-first search illustrated a significant decrease in the users’ connectivity after suspending each focal information spreader from G ; where the focal information spreaders in K_{G_1} , occupying critical positions in the structure of the network. Figure 8 shows a huge decrease in users’ connectivity after suspending K_{G_1} from G over time.

Moreover, the focal information spreaders # (k_{4G_1} , k_{5G_1} , and k_{6G_1}) were able to decrease the number of edges in G more than other focal information spreaders compared to the number of edges shown in Fig. 3.

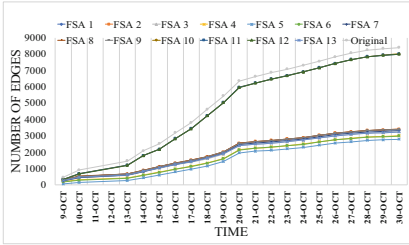


Fig. 8. Changes in the number of edges after suspending K_{G_1} from G .

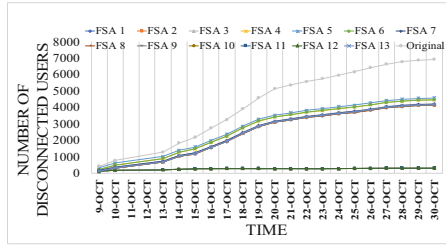


Fig. 9. Changes in the number of disconnected users after suspending K_{G_1} from G .

4.3.4 Changes in the Connectivity of the Users Over Time

The depth-first search method revealed information about the users’ connectivity in dynamic social networks. This method measures the disconnected users after suspending each focal information spreader from G . Figure 9 shows the development of the online users in G after suspending K_{G_1} , where a massive decrease in the number of users is reported.

Moreover, suspending the focal information spreaders # (k_{4G_1} , k_{5G_1} , and k_{6G_1}) increased the number of disconnected users compared to the original number of users reported in Fig. 3. In other words, these sets were close to disconnected from the entire network, where they could disconnect hundreds and thousands of users over time. Also, the values represent the activities and communications of these sets over time.

5 Conclusion

In this research, we studied the dynamic aspects of focal information spreaders and their ability to spread information to the maximum number of users in dynamic OSNs. For this purpose, the focal structure analysis model was used to identify the focal sets of information spreaders, and the adaptation algorithm was utilized to observe the growth of focal sets of information spreaders in social networks over time. In addition, the modularity method, the depth-first search method, and the clustering coefficient method were implemented to measure and validate the development of focal sets of information spreaders and illustrate the behavior of the dynamic social network over time.

Furthermore, based on the analysis presented in this research, appending the focal sets of information spreaders from the dynamic OSNs would reduce the clustering coefficient values, reduce the number of edges between users, increase the modularity values in the network, and increase the number of disconnected users in dynamic OSNs. In addition, this research proposed a systematic and a simplified method to investigate the evolution of the focal sets of information spreaders over time, project their activities early, and systematically limit the information spread in an evolving Twitter network. Throughout this research, we were able to illustrate when information spreaders will increase their activities in a network, revealing which focal sets were more active than others in the dynamic network.

For future work, due to the limitations in the nodes’ removals, we would study alternative methods for removing focal information spreaders and improving the effectiveness

of the users' suspension from the network. The authors in [29] developed a modularity vitality method to calculate the exact change in modularity values in the network. Such research needs more investigation with respect to the focal structure analysis model in the dynamic social network and the robustness of the networks, as mentioned in [30, 31].

Acknowledgment. This research is funded in part by the U.S. National Science Foundation (OIA-1946391, OIA-1920920, IIS-1636933, ACI-1429160, and IIS-1110868), U.S. Office of the Under Secretary of Defense for Research and Engineering (FA9550-22-1-0332), U.S. Office of Naval Research (N00014-10-1-0091, N00014-14-1-0489, N00014-15-P-1187, N00014-16-1-2016, N00014-16-1-2412, N00014-17-1-2675, N00014-17-1-2605, N68335-19-C-0359, N00014-19-1-2336, N68335-20-C-0540, N00014-21-1-2121, N00014-21-1-2765, N00014-22-1-2318), U.S. Air Force Research Laboratory, U.S. Army Research Office (W911NF-20-1-0262, W911NF-16-1-0189, W911NF-23-1-0011), U.S. Defense Advanced Research Projects Agency (W31P4Q-17-C-0059), Arkansas Research Alliance, the Jerry L. Maulden/Entergy Endowment at the University of Arkansas at Little Rock, and the Australian Department of Defense Strategic Policy Grants Program (SPGP) (award number: 2020-106-094). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations. The researchers gratefully acknowledge the support.

References

1. Alassad, M., Spann, B., Agarwal, N.: Combining advanced computational social science and graph theoretic techniques to reveal adversarial information operations. *Inf. Process. Manag.* **58**(1), 102385 (2021)
2. Alassad, M., Hussain, M.N., Agarwal, N.: Finding fake news key spreaders in complex social networks by using bi-level decomposition optimization method. In: Agarwal, N., Sakalauskas, L., Weber, G.W. (eds.) *International Conference on Modelling and Simulation of Social-Behavioural Phenomena in Creative Societies*. CCIS, vol. 1079, pp. 41–54. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29862-3_4
3. Robinhood, Reddit CEOs to Testify Before Congress on GameStop. <https://www.investopedia.com/robinhood-reddit-ceos-to-testify-in-congress-on-gamestop-gme-5112714>. Accessed 16 Feb 2021
4. Coronavirus: Armed protesters enter Michigan statehouse - BBC News. <https://www.bbc.com/news/world-us-canada-52496514>. Accessed 29 Aug 2020
5. Şen, F., Wigand, R., Agarwal, N., Tokdemir, S., Kasprzyk, R.: Focal structures analysis: identifying influential sets of individuals in a social network. *Soc. Netw. Anal. Min.* **6**(1), 17 (2016)
6. Alassad, M., Agarwal, N., Hussain, M.N.: Examining intensive groups in YouTube commenter networks. In: Thomson, R., Bisgin, H., Dancy, C., Hyder, A. (eds.) *Proceedings of 12th International Conference, SBP-BRiMS 2019*. LNCS, vol. 11549, no. 12, pp. 224–233. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21741-9_23
7. Zafarani, R., Abbasi, M.A., Liu, H.: *Social Media Mining: An Introduction*. University Press, Cambridge (2014)
8. Wijenayake, S.: Understanding the dynamics of online social conformity. In: *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, pp. 189–194 (2020)

9. Alassad, M., Hussain, M.N., Agarwal, N.: Comprehensive decomposition optimization method for locating key sets of commenters spreading conspiracy theory in complex social networks. *Cent. Eur. J. Oper. Res.*, 1–28 (2021)
10. Nguyen, N.P., Dinh, T.N., Shen, Y., Thai, M.T.: Dynamic social community detection and its applications. *PLoS ONE* **9**(4), 91431 (2014)
11. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. In: *Proceedings ACM-SIAM Symposium Discrete Algorithms*, vol. 46, no. 5, pp. 604–632 (1999)
12. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web. In: *World Wide Web Internet Web Information Systems*, vol. 54, no. 1999–66, pp. 1–17 (1998)
13. Alassad, M., Spann, B., Al-khateeb, S., Agarwal, N.: Using computational social science techniques to identify coordinated cyber threats to smart city networks. In: El Dimeery, I., et al. (eds.) *Design and Construction of Smart Cities. JIC Smart Cities 2019. Sustainable Civil Infrastructures*, pp. 316–326. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-64217-4_35
14. Shajari, S., Agarwal, N., Alassad, M.: Commenter behavior characterization on YouTube channels, April 2023. <https://arxiv.org/abs/2304.07681v1>
15. Agarwal, N., Liu, H., Tang, L., Yu, P.S.: Identifying the influential bloggers in a community. In: *Proceedings 2008 International Conference Web Search Data Mining*, pp. 207–218 (2008)
16. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**(10), 10008 (2008)
17. Al-Khateeb, S., Agarwal, N.: Modeling flash mobs in cybernetic space: evaluating threats of emerging socio-technical behaviors to human security. In: *Proceedings - 2014 IEEE Joint Intelligence and Security Informatics Conference, JISIC 2014*, p. 328 (2014)
18. Chen, N., Liu, Y., Chen, H., Cheng, J.: Detecting communities in social networks using label propagation with information entropy. *Phys. A Stat. Mech. Appl.* **471**, 788–798 (2017)
19. Xu, X., Zhu, C., Wang, Q., Zhu, X., Zhou, Y.: Identifying vital nodes in complex networks by adjacency information entropy. *Sci. Rep.* **10**(1), 1–12 (2020)
20. Kitsak, M., et al.: Identification of influential spreaders in complex networks. *Nat. Phys.* **6**(11), 888–893 (2010)
21. Chen, D., Lü, L., Shang, M.S., Zhang, Y.C., Zhou, T.: Identifying influential nodes in complex networks. *Phys. A Stat. Mech. Appl.* **391**(4), 1777–1787 (2012)
22. Alvari, H., Hajibagheri, A., Sukthankar, G.: Community detection in dynamic social networks: a game-theoretic approach. In: *ASONAM 2014 - Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 101–107 (2014)
23. Dakiche, N., Slimani, Y., Tayeb, F.B.S., Benatchba, K.: Community evolution prediction in dynamic social networks using community features' change rates. In: *Proceedings of the ACM Symposium on Applied Computing*, vol. Part F147772, pp. 2078–2085 (2019)
24. Dakiche, N., Benbouzid-Si Tayeb, F., Slimani, Y., Benatchba, K.: Tracking community evolution in social networks: a survey. *Inf. Process. Manag.* **56**(3), 1084–1102 (2019)
25. Takaffoli, M., Rabbany, R., Zaïane, O.R.: Community evolution prediction in dynamic social networks. In: *ASONAM 2014 - Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 9–16 (2014)
26. Bródka, P., Kazienko, P., Kołoszczyk, B.: Predicting group evolution in the social network. In: Aberer, K., Flache, A., Jager, W., Liu, L., Tang, J., Guéret, C. (eds.) *Social Informatics. LNCS*, vol. 7710, pp. 54–67. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35386-4_5
27. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99**(12), 7821–7826 (2002)

28. Tarjan, R.: Depth-first search and linear graph algorithms. *SIAM J. Comput.* **1**(2), 146–160 (1972)
29. Magelinski, T., Bartulovic, M., Carley, K.M.: Measuring node contribution to community structure with modularity vitality. *IEEE Trans. Netw. Sci. Eng.* **8**(1), 707–723 (2021)
30. Holme, P., Kim, B.J., Yoon, C.N., Han, S.K.: Attack vulnerability of complex networks. *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.* **65**(5), 14 (2002)
31. Da Cunha, B.R., González-Avella, J.C., Gonçalves, S.: Fast fragmentation of networks using module-based attacks. *PLoS ONE* **10**(11), e0142824 (2015)



Unified Logic Maze Generation Using Network Science

Johnathon Henke^(✉) and Dinesh Mehta

Colorado School of Mines, Golden, CO 80401, USA

jhenke@mines.edu

<http://mines.edu>

Abstract. Solving maze puzzles is a recreational activity with long-standing roots in human civilization dating back several thousands of years. This paper considers the problem of automated maze generation for a more recent class of maze puzzles, the logic maze, popularized by Abbott in 1990. Although there are several distinct types of logic mazes, we present a single unified generation strategy based on a state graph representation. We capture desirable features of a maze in an objective function that consists of several network science metrics on the state graph and the original maze. We then optimize this objective function through the use of local state space search and obtain high-quality results.

Keywords: logic maze · state graph · maze characteristics · local search

1 Introduction and Related Work

Mazes and labyrinths have been discovered in multiple ancient civilizations [5]. These classical mazes can typically be viewed as a two-dimensional grid with adjacent elements sometimes separated by a barrier or wall. The goal is to trace a valid path from start to end. The maze is represented as a graph with a vertex for each grid square, while an edge denotes permitted moves between adjacent squares. Algorithmically, the problem is easily solved by initiating a simple traversal such as a Breadth-First Search (BFS) at the start vertex. This work explores a newer type of maze: the Logic Maze (originally referred to as a *Mad Maze*, a term coined by Abbott in his 1990 book). Logic Mazes [3] retain the notion of physical space as in a classical maze, but introduce additional rules that govern movement. They are also known as Multi-state mazes because a solution may require one to visit the same location multiple times in different logical “states.” These logic mazes present interesting and challenging graph modeling problems, and have been used by the authors in the undergraduate Algorithms class at the Colorado School of Mines. The primary purpose of this work is to generate logic mazes of desired size and difficulty, both to introduce students to the rules of each maze and test their implementations of a graph model and solution. Abbott and others designed much of their mazes by hand. To the best of our knowledge, the only work that focuses on this type of automated generation is Neller et al. [11], which we discuss in the next section.

2 Modeling Logic Mazes

Although our work has been used to generate different types of Logic Mazes, we focus solely on Jumping Mazes in this paper due to space limitations. The first Jumping Maze Abbott introduces is Maze #7 Jumping Jim [1]—this simple version of a jumping maze consists of a simple numeric grid. The grid has a number on each cell that indicates how far one must move (horizontally or vertically, but not diagonally) from that cell. Starting from the upper left-hand corner, the goal is to reach the bottom right-hand corner in the minimum number of “jumps” (note it is the number of jumps that we try to minimize and not the total length of the jumps). Once again, this maze can be intuitively modeled as a graph—followed by a BFS for an algorithmic solution. Neller et al. emphasize the generation of this specific type of jumping maze in their research [11]. Their work stands as the primary, if not the sole, contribution in this domain.

The Jumping Maze variant introduced in Maze #15 Jumping Jim’s Encore (JJE) is more complex [1]. It includes circled numbers that change the direction of movement (Fig. 1). If one lands on a circled number while moving horizontally or vertically, then one’s movement direction changes to diagonal until reaching another circled number, at which point it reverts to vertical/horizontal. This small change introduces the concept of “movement state” into the maze, making it possible to visit the same cell twice, once in the state of vertical/horizontal (or cardinal) movement and once in the state of diagonal movement. It is possible to solve this problem either by (1) developing a customized JJE-variant of BFS or (2) by modifying the underlying graph model and retaining the use of the original BFS Algorithm. The former approach results in a higher cognitive load because both the underlying data structure and algorithm are simultaneously modified,

6	1	2	4	4	2	7	2
1	3	2	2	1	6	1	5
2	5	4	4	1	2	5	6
2	6	1	2	3	2	1	3
3	2	2	4	4	4	3	2
4	4	2	4	2	4	2	2
1	6	4	2	3	4	3	4
6	2	2	7	5	1	3	GOAL

Fig. 1. Jumping maze instance with diagonal state changes. The shortest solution to this maze is (1,1), (7,1), (6,2), (2,2), (5,5), (1,1), (7,7), (4,4), (6,6), (2,6), (8,6), (8,5), (3,5), (4,5), (1,5), (5,1), (2,4), (4,6), (6,8), (8,6), (7,5), (4,8), (1,5), (5,5), (5,1), (2,1), (3,1), (3,3), (7,3), (7,7), (4,7), (4,6), (4,4), (6,4), (6,8), (8,8), where (1,1) and (8,8) respectively denote the top-left and bottom-right corners.

increasing the likelihood of design errors. The latter modeling approach can be viewed as a reduction to a state graph (or a maze without state) followed by standard BFS, which is available and reliably (i.e., without bugs) implemented in graph libraries associated with many programming languages. For these reasons, in our Algorithms class project, we require the latter approach to our instruction: both to modularize and simplify the design and as an example of good software engineering practice.

Further, this graph modeling paradigm allows for a *unified* solution to the problem we seek to address in this paper—the generation of suitable maze instances with desirable characteristics. This facilitates the use of similar components in the maze design function and enables the consideration of similar traits when rating the difficulty of multiple logic mazes. Additionally, there is no need to “reinvent the wheel” by devising new maze-specific metrics, as there are already existing network science metrics that can be used to evaluate graphs.

Given that we are modifying the graph and not the BFS, consider the following model for the JJE maze instance M : Let G be a directed, unweighted graph that will model M . For each cell s in M with uncircled number n , create two vertices in G , s_1 and s_2 , that respectively represent the cardinal and diagonal states of s . Add directed edges from s_1 to the cardinal vertices of distance n from s considering cardinal movement. Add directed edges from s_2 to the diagonal vertices of distance n considering diagonal movement. The process is similar for circled numbers, except the outgoing edges connect to vertices of the other movement type. Thus, we add directed edges from s_1 to the diagonal vertices of distance n considering diagonal movement and directed edges from s_2 to the cardinal vertices of distance n considering cardinal movement. Create only one vertex for the goal cell. When it is possible to reach the goal in one move whether in the cardinal state or the diagonal state, add an edge to this single goal vertex.

To eliminate the notion of “state”, which is the movement type, we have expanded the number of vertices in the graph relative to the number of grid cells. There are two states (vertices) per cell, and thus double the number of vertices in the state graph (minus one for the goal). In effect, we have created a two-level multilayer network [10] with a cardinal level and a diagonal level, linked by the circled numbers. This technique is common for logic mazes with state changes; we often increase the number of vertices in the graph relative to the original maze in order to capture its complexity. However, the result is not always a multilayer network—that is contingent on the initial puzzle.

Other mazes such as arrow mazes (Apollo and Diana/Apollo’s revenge), connections mazes (Grandpa’s Transit Map), and multiplayer mazes (Spacewreck, Meteor Storm) can also be modeled as state graphs [1]. These mazes were modeled and generated in the longer paper, but are omitted due to length constraints [7].

3 Maze Characteristics

The primary objective of this research was to create logic maze instances of varying sizes, both small enough to serve as instructional examples for students to become familiar with the rules of the maze, and large enough to test their implementations. A local search implementation resulted in the highest-quality mazes. Hence, it became vital to define qualities of a maze to determine a score for the purposes of local search.

The state graph reduction applied to all mazes in this work allows for analysis of both the underlying graph representation and the high-level problem instance, as they are reduced to an unweighted directed graph. This unique approach enables the use of an identical objective function when calculating the score of state graphs, regardless of maze type, which is a significant contribution of this work. Additional qualities can then be applied to the higher-level problem instances individually.

Neller et al. [11] discuss some desired maze attributes that will be introduced. Abbott also outlines several metrics by which he hand-designed his mazes, which will be mentioned as well. We found some additional maze qualities to be challenging based on our experience. These properties are all intuitive and combined in our work, but we caution that to our knowledge, none of the metrics presented have been formally evaluated via human testing to determine their validity.

Before delving into specific maze characteristics, we note at the outset that when designing a maze for humans to solve, a common solving method is to work backward from the goal instead of forward from the start [3]. Therefore, it follows that the maze ought to be equivalently difficult when attempting to solve it forward or backward, otherwise it will be easily solved through back-tracing. A practical method to address this is to compute a metric's value for both the state graph and its transpose and take the minimum when including it in the objective function.

3.1 Paths, Branching, Reachability

Abbott suggests leaving a substantial amount of space for several long false paths when deciding what portion of the vertices should be involved in the shortest path [2]. From experience with our program, a good number is 15–35%, but exceeding 35% could limit the potential for other characteristics that increase maze difficulty.

In addition to the length of the shortest path, the existence of multiple shortest paths can affect the solver's motivation. Neller et al. find there is a level of satisfaction achieved when one discovers the shortest (or best) solution, and the existence of a unique shortest solution can motivate solvers to continue working on a maze even after solving it [11].

Branching is a relatively intuitive characteristic that refers to the number of different locations one could be in after exactly X moves from the start, usually calculated as a percentage of the shortest path length. It is similar to the idea

of a “branching factor” in the growth of a search tree/space, and is used to help reduce sections of forced moves at the beginning or end of the maze.

A *reaching* vertex v is a vertex from which it is possible to reach the goal g . A *reachable* vertex is a vertex that can be reached from the start s [11]. A traversal from the start (finish) on the state graph (transpose) can be used to calculate the portion of reachable (reaching) vertices. Sixty percent is a good minimum portion of the maze to be reachable (reaching) from the start (finish). This restriction can greatly influence the creation of traps, as traps are often only accessible when moving in one direction through the maze.

Reachability also measures the efficiency of a maze because the number of vertices in the state graph is a function of the puzzle size, which usually does not change while generating an instance. Vertices that are both unreachable and unreaching are denoted *isolates*, and these represent wasted resources—vertices in the state graph that cannot be reached from the start nor backward from the finish. The puzzle instance perhaps ought to be redesigned so these are included as a part of the maze.

3.2 Traps and Holes

From the definitions of reachable and reaching vertices, we can define several different traps to entertain and confuse the maze solver. A dead end of a maze is a set of one, or many reachable, unreaching vertices. A reverse dead end is a set of reaching, unreachable vertices in the state graph.

A *black hole* is a set of strongly connected, reachable, unreaching vertices in the state graph. In effect, it is a false path that ends in loop(s) instead of at a singular dead end. Black holes, especially large black holes, can significantly increase a maze’s difficulty because a maze solver may spend a lot of time in the trap before realizing there is no escape. This definition is slightly different than the one given in [11] because we wanted to focus on the strongly connected vertices, which is the core of the trap, and neglect the fringes. Black holes usually force the solver to restart the maze once they realize there is no path to the solution because the solver does not remember how they initially entered the black hole [11]. A *white hole* is a set of strongly connected, reaching, unreachable vertices in the state graph. It is identical to a black hole when considering the transpose of the state graph (with the start becoming the finish and vice versa).

These types of traps only affect one direction of solving the maze and directly conflict with reachability. Reverse dead ends and white holes have no impact on individuals solving in the forward direction because they cannot be reached when moving forward. Similarly, dead ends and black holes do not impede solvers attempting to move backward from the solution. A *whirlpool* is a set of strongly connected, reaching, reachable vertices. In effect, it is a hole that can be reached in both directions. The placement of such traps is more difficult and important. A whirlpool located near the start is vulnerable to back-tracing from the finish, and the same can be said of a whirlpool close to the finish.

3.3 Decisions, Required Vertices, Bridges/Dominance

The presence of traps such as black/white holes in a maze does not guarantee it is difficult to solve. We have generated many mazes where multiple traps exist, but a solver may not encounter them unless they are unlucky while a lucky solver may never encounter these traps.

We want to maximize the chances for a solver to become lost in traps for them to truly increase the maze’s difficulty. The first solution that comes to mind is to consider the *decisions* that a maze solver must make along the shortest path. Consider each vertex on the shortest path n . From n , there are several cases for each immediate descendent d (the resulting vertex of each outgoing edge of n):

- d is the next vertex on the shortest path.
- d is in an unreaching trap (black hole/dead end).
- d is in a reaching trap (whirlpool) or on a suboptimal path to the solution.
- d is a previous vertex on the shortest path.

We aim to have as many decision points as possible that lead into reaching/unreaching traps or suboptimal paths. However, merely counting the number of outgoing edges from each vertex on the shortest path that satisfy these requirements is overly simplistic. Decisions within a few vertices of the solution are usually trivial, some traps ought to be weighted more than others (based on the furthest distance a solver can travel without retracing his/her steps), and most importantly, not all solvers may encounter every decision along the shortest path. Solvers may find longer paths that are in entirely separate parts of the graph, or multiple shortest paths may exist.

Because decisions are one of the primary factors in determining the difficulty of a maze [11], it is important to consider only the decisions that every solver, regardless of the path chosen, must make. Therefore, we need to determine which vertices are “required” vertices R that are present on all paths from the start vertex s to the goal vertex g . This question has been studied and solved in theory related to control-flow graphs and is referred to as *Dominance*. A vertex v *dominates* another vertex u if v lies on every path from the entry vertex to u . We need to determine which vertices dominate the goal vertex with an entry node of the starting vertex. Cooper, Harvey, and Kennedy [4] give a $\mathcal{O}(V^2)$ algorithm that in practice runs faster on graphs with less than 1000 vertices than the classical $\mathcal{O}(E \log(V))$ Lengauer-Tarjan Algorithm. State graphs of human-solvable typical mazes do not usually exceed this number of vertices. After computing the dominator tree, we can identify the vertices that dominate the goal vertex. By only including the decision scores associated with these vertices, we can ensure the maze’s perceived difficulty is based on decisions that all solvers must consider.

Mazes often have characteristics unique to the problem instance itself that ought to be included in the objective function, such as the number of circled locations that swap the direction of movement, u-turns, and revisiting the same location on the grid in both movement types [1], etc. Because the state graph abstracts away some qualities of the maze instance, these characteristics cannot

be captured by the state graph and must be separately included in the objective function of each specific maze type.

4 Local Search Generation

One efficient way to implement the detection of attributes discussed is to use the state graph reduction combined with a robust graph library such as Networkx in Python 3. This library includes pre-implemented versions of all the necessary functions to detect and score state graph attributes, such as dominance, traversals, and distance calculations [6]. As mentioned previously, local search maze generation does not start from scratch but rather takes a currently generated maze instance and makes small modifications that increase the value of an objective function. When defining local search solutions to problems, there are generally three items to define:

1. An objective function that returns a score given a solution instance to the problem. We seek to either minimize or maximize this function.
2. A notion of a neighborhood, or a set of small modifications made to a given solution to turn it into another solution then rated by the objective function. Typically, a solution has multiple neighbors.
3. A search algorithm, that is, a method to choose between the neighbors of a given solution [12].

In the context of maze generation, the objective function will be as defined previously and contain metrics to rate both the state graph and the maze instance. The neighborhood definition turns one maze instance into another, which involves making a small change to the maze that is simultaneously reflected in the state graph. For jumping mazes, a neighbor state is simply changing the number and/or the circling of a particular square on the grid.

5 Example Objective Function

In this section, we will provide an example search function used for state graph generation that focuses on decisions. Let n denote the number of vertices in the state graph. It is best (if possible) to try and scale terms using n to provide weightings that represent importance.

Additionally, depending on the size of the instances being generated, some terms in the objective function may necessitate coefficients for proper weighting (which we include below). Furthermore, the generator can adjust these coefficients based on the desired maze metrics they aim to ensure the instance possesses. Note that we will be attempting to maximize the score of the maze (and not minimize it). When logical, the score of a trait ought to be calculated for both the state graph and the transpose and the minimum added to the score, as mentioned previously.

The score is initialized to zero. The function comprises three primary components: a Path score, Reachability Score, and Decisions score, with some miscellaneous additional terms. We define the following function to score the state graph.

Path. Let L represent the ratio of the length of the shortest path(s) to n , indicating the proportion of vertices included in the shortest path. A bonus c_1n is provided for one shortest path as mentioned in Sect. 3.

$$P = \begin{cases} -n^3 & \text{no path} \\ -n^2 & \text{multiple shortest paths, } L \leq 0.15 \vee L \geq 0.35 \\ -n^2 + c_1n & \text{one shortest path, } L \leq 0.15 \vee L \geq 0.35 \\ c_1n & \text{one shortest path, } 0.15 \leq L \leq 0.35 \end{cases} \quad (1)$$

Branching and Reachability. Calculate the sum of vertices b (b_t on the transpose) that are reachable after a certain number of moves (10–20% of the shortest path length is a good starting point). Let r and r_t respectively denote the set of reachable and reaching vertices.

$$\text{Branching} = \min(b, b_t) \quad (2)$$

$$\text{Reachability} = \min(r, r_t) + |r \cup r_t| - (n - |r \cup r_t|)^2 \quad (3)$$

Decisions. The most important factor in determining the difficulty of a maze, given all other traits are equal, is the decision score that dictates the exact decisions every maze solver considers. Compute the required nodes that every path from the start to the finish contains [4]. Then, evaluate the decisions at these vertices in both the state graph (D) and its transpose D_t , as discussed in Sect. 3, take the minimum, (multiply by a constant to increase the weight if desired) and add to the score.

$$\text{Decisions} = c_2 \min(D, D_t) \quad (4)$$

Misc. and Maze Instance Terms. A dead end is a reaching or reachable vertex in a state graph that does not have any outgoing edges. Holes make better traps than dead ends. Subtract the number of dead ends d times a constant (the cost of each dead end) from the score. Although we use the objective function terms above to rate the state graphs, we still need to consider the higher-level maze instances during the generation process. Briefly, we introduce metrics to avoid large clusters of the same number as mentioned in Neller et al. [11], to avoid a large number of circled locations, and provide small bonuses for each time a movement change is required (from cardinal to diagonal or vice versa), each time a square is visited in both movement states, and each time a u-turn is encountered (multiple repeated moves forward and backward along the same row/column/diagonal). These are denoted as M .

Final Formula. In every instance generated, the final case for the path score was always met. Taking this into consideration, we have the following objective function:

$$F(n) = c_1n + \min(b, b_t) + \min(r, r_t) + |r \cup r_t| - (n - |r \cup r_t|)^2 + c_2 \min(D, D_t) - c_3d + M \quad (5)$$

6 Results

Table 1 shows the results of several mazes generated using this objective function and simulated annealing with random restart and offers several items to consider. The path score (SP score) is the same for each instance. Every maze has a single shortest path, and thus all have the c_1n resulting SP score. We purposefully chose $c_1 = 10$ to enforce this condition, and we appear to have been successful. The branching score gives a small bonus to the score as desired to help prevent long sections of forced moves.

There are fewer reachable/reaching vertices due to holes in the highest-rated mazes. This is a common theme we have noticed when generating instances: one can maximize reachability at the cost of traps, which results in a higher reachability score at the cost of a lower decisions score, or one can maximize hole sizes, which results in a higher decision score at the cost of a lower reachability score. Compare maze one to maze seven. Maze seven has the highest reachability score (241) of the instances presented in Table 1 with the highest proportion of reachable (91.3%), reaching (90.6%), and both reachable and reaching (81.9%) vertices. However, it has the lowest decision score.

The decision score dominates the overall score, usually representing about 40%. We claimed previously that decisions are the most important factor when considering the difficulty of a state graph, and this principle is reflected in the scoring criteria we have adopted.

The highest-rated mazes generally have a combination of the most required decisions, (which may or may not be the most required vertices), large black and white holes, and many entrances to these traps from required vertices (which is the BH/WH entrances table entry). The top two mazes exhibit these traits, and both have the highest decision scores. A very low R and R score is indicative of mazes with large unreaching/unreachable traps; in the most optimal version of these mazes (according to this objective function), only the shortest path vertices are reachable and reaching, and all other vertices are part of large strongly connected holes.

As designed, the state graph score plays a much larger role than the instance score. When using the same objective function to generate multiple different mazes, it would defeat the purpose of sharing a state graph objective function if it did not play a more significant role than the individual maze instance terms. The most exceptional of these mazes is depicted in Fig. 1 and its component graph is shown in Fig. 2.

Table 1. Jumping generation maze results are shown below with the primary components of the scores highlighted in **bold**. To generate this table, we used $c_1 = 10, c_2 = 2, c_3 = 5$. SP = shortest path, BH = black hole, WH = white hole, R or R = Reachable or Reaching, R and R = Reachable and Reaching. The hole entrances are only to the largest hole. The required decisions is the sum of the required vertices out degrees, which represents the minimum quantity of choices maze solvers will consider. Fwd and bwd decisions are the forward and backward decisions, and the other metrics are as specified previously.

Metric/Maze	1	2	3	4	5	6	7	Avg.
SP Length	36	38	34	27	22	36	43	33.7
SP Quantity	1	1	1	1	1	1	1	1
SP Score	1270	1270	1270	1270	1270	1270	1270	1270
Branching Score	24	31	31	15	15	15	17	21.1
Reachable (%)	68.5	66.9	72.4	72.4	69.3	88.2	91.3	75.6
Reaching (%)	62.2	66.1	74.0	89.0	92.9	89.8	90.6	80.7
R or R (%)	98.4	99.2	99.2	100	98.4	99.2	100	99.2
R and R (%)	32.3	33.9	47.2	61.4	63.8	78.7	81.9	57.0
Reachability Score	199	208	216	218	208	236	241	218
Required Vertices	36	38	33	26	20	32	39	32.0
Required Decisions	82	87	78	60	50	70	84	73.0
Largest BH	35	30	22	8	2	2	2	14.4
BH Entrances	18	16	15	2	1	1	1	7.7
Largest WH	25	24	22	4	17	2	2	13.7
WH Entrances	25	23	14	10	18	6	1	13.9
Fwd Decisions	1024	902	834	832	794	558	418	766
Bwd Decisions	1046	990	850	834	794	560	418	784.6
Decision Score	1024	902	834	832	794	558	418	766.0
Dead End Score	-50	-50	-60	-50	-80	-60	-60	-58.6
State Graph Score	2467	2361	2291	2285	2207	2019	1886	2216.6
Circled (%)	23.4	21.9	25	21.9	23.4	21.9	23.4	23.0
State Changes	7	7	10	4	6	12	12	8.3
Double Visited	8	10	6	4	3	8	11	7.1
U-turns	9	9	10	9	5	15	17	10.6
Instance Score	143	186	137	99	54	235	276	161.4
Overall Score	2610	2547	2428	2384	2261	2254	2162	2378

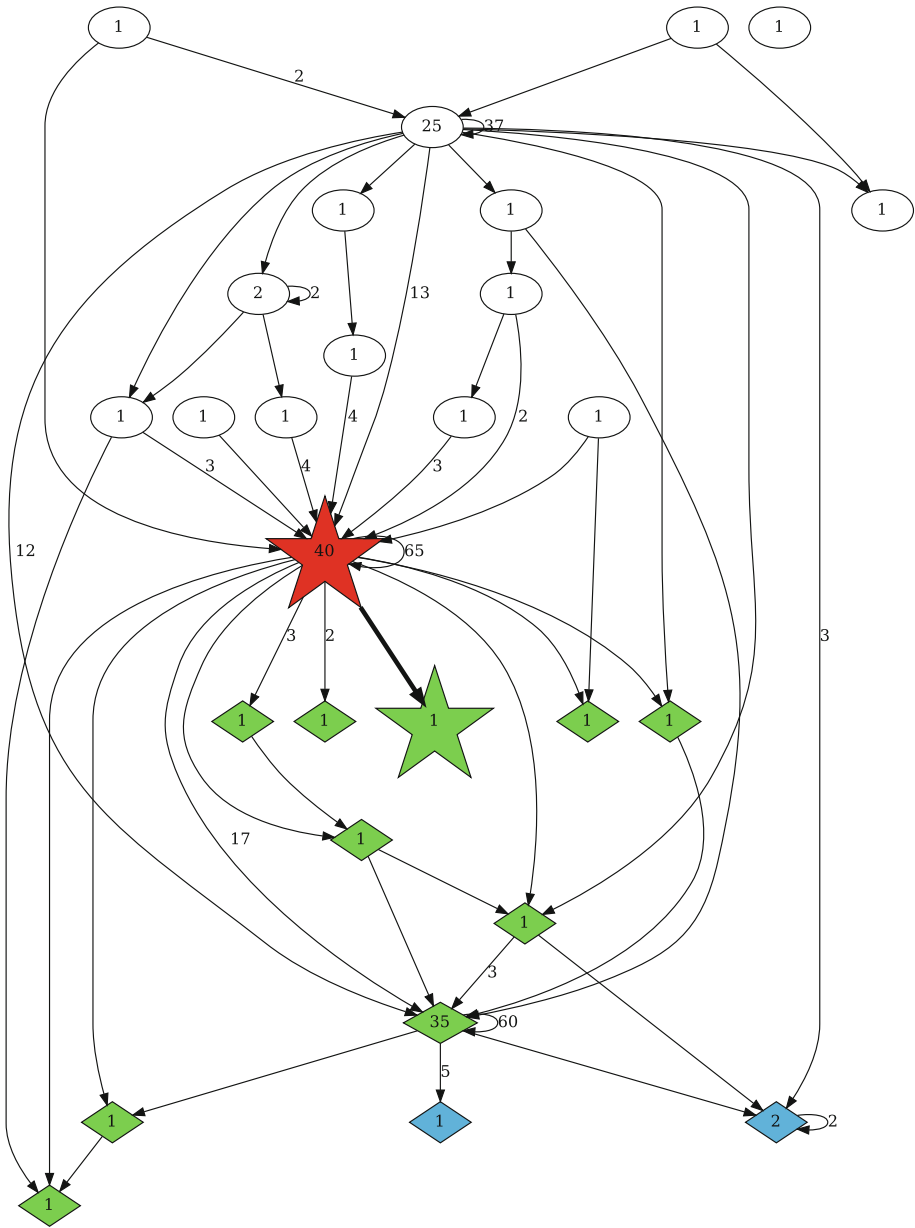


Fig. 2. Component graph of the excellent jumping maze instance. Colors indicate the distance from the start and finish (red = start, blue = finish, uncolored means unreachable). Vertex labels denote the number of vertices in each SCC. Diamond shapes indicate that it's impossible to reach the finish, star shapes represent the start/finish, and edge labels denote the number of direct connections between components. (Color figure online)

7 Conclusions and Future Work

In this work, we created challenging logic mazes for humans, crafting objective functions based on maze attributes, but these functions require validation. A potential validation method is simulating human interactions with these mazes, comparing results to our objectives, and further investigating human maze-solving methods as discussed in [13]. Karlsson [9] employed a DFS for this, but noted its limitations due to human non-deterministic choices at intersections. Future research could pinpoint an algorithm that aptly represents human exploration, serving as an alternate difficulty metric for state graphs.

Any game that can be modeled as a state space can be represented using this model, and its metrics and traits rated by the scoring methods we have developed. This has potential for applications in the gaming industry. Modeling a game as a state space and identifying qualities the space must contain to be entertaining and challenging for humans to solve is a potential application. Many other games such as Sokoban can be represented as state spaces [8] and rated in such a fashion as the logic mazes presented in this work.

References

1. Abbott, R.: *Mad Mazes: Intriguing Mind Twisters for Puzzle Buffs*. Adams Media Corporation, Game Nuts and Other Smart People (1990)
2. Abbott, R.: *SuperMazes: Mind Twisters for Puzzle Buffs*. Prima Publishing, Game Nuts and Other Smart People (1997)
3. Abbott, R.: Logic mazes (1999). <https://www.logicmazes.com/dec99.html>
4. Cooper, K., Harvey, T., Kennedy, K.: A simple, fast dominance algorithm. Rice University, CS Technical report 06-33870 (2006)
5. Fisher, A., Gerster, G.: *The Art of the Maze*. Seven Dials (2000)
6. Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using networkx. In: Varoquaux, G., Vaught, T., Millman, J. (eds.) *Proceedings of the 7th Python in Science Conference*, pp. 11 – 15, Pasadena, CA, USA (2008)
7. Henke, J.: Unified logic maze generation: combining state graph representations and local search for diverse puzzle design. Master’s thesis, Colorado School of Mines (2023)
8. Jarusek, P., Pelánek, R.: *Human Problem Solving: Sokoban Case Study* (2010)
9. Karlsson, A.: Evaluation of the complexity of procedurally generated maze algorithms (2018)
10. Kivelä, M., Arenas, A., Barthélemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. *J. Complex Netw.* **2**(3), 203–271 (2014). <https://doi.org/10.1093/comnet/cnu016>
11. Neller, T.W., Fisher, A., Choga, M.T., Lalvani, S.M., McCarty, K.D.: Rook jumping maze design considerations. In: van den Herik, H.J., Iida, H., Plaat, A. (eds.) *CG 2010. LNCS*, vol. 6515, pp. 188–198. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-17928-0_18
12. Russell, S.J., Norvig, P.: *Artificial Intelligence: A Modern Approach*, 4 edn. Pearson (2022)
13. Zhao, M., Marquez, A.G.: Understanding humans’ strategies in maze solving. *CoRR* **abs/1307.5713** (2013). <http://arxiv.org/abs/1307.5713>



INDoRI: Indian Dataset of Recipes and Ingredients and Its Ingredient Network

Sandeep Khanna¹(✉), Chiranjoy Chattopadhyay², and Suman Kundu¹

¹ Indian Institute of Technology Jodhpur, Jodhpur, India

{khanna.1, suman}@iitj.ac.in

² FLAME University, Pune, India

chiranjoy.chattopadhyay@flame.edu.in

Abstract. Exploring and comprehending the culinary heritage of a nation holds a captivating allure. It offers insights into the structure and qualities of its cuisine. The endeavor becomes more accessible with the availability of a well-organized dataset. In this paper, we present the introduction of INDORI (Indian Dataset of Recipes and Ingredients), a compilation drawn from seven distinct online platforms, representing 18 regions within the Indian subcontinent. This comprehensive geographical span ensures a portrayal of the rich variety within culinary practices. Furthermore, we introduce a unique collection of stop words, referred to as ISW (Ingredient Stop Words), manually tuned for the culinary domain. We assess the validity of ISW in the context of global cuisines beyond Indian culinary tradition. Subsequently, an ingredient network (InN) is constructed, highlighting interconnections among ingredients sourced from different recipes. We delve into both the defining attributes of INDORI and the communal dimensions of InN. Additionally, we outline the potential applications that can be developed leveraging this dataset. Addressing one of the applications, we demonstrated a research problem on InN with a simple weighted community detection algorithm. Furthermore, we provide a comparative analysis of the results obtained with this algorithm against those generated by two baselines.

Keywords: Dataset · Food Computing · Ingredient Network · Stop Words

1 Introduction

India, characterized by its rich tapestry of cultures, hosts a plethora of distinct cuisines. Tackling food computing challenges within this culinary landscape is indeed complex. One significant hurdle stems from the dearth of structured data that spans India's diverse cuisines despite numerous websites house extensive recipe databases. The reason for the same is that the information available therein is predominantly unstructured, comprising text and multimedia content.

This paper introduces the Indian Dataset of Recipes and Ingredients (INDORI), encompassing a total of 5187 recipes. Recipes were extracted and

gathered from seven different online platforms [1–7]. These recipes span a variety of Indian cuisines, reflecting the rich cultural diversity across regions such as Punjabi, Bengali, and Gujarati. INDoRI stands as a structured repository of recipes and their corresponding ingredients. Further, the dataset includes a graph-based representation of ingredient relationships, namely, ingredient network (InN). InN is formed by capturing ingredient relationships based on their co-occurrence within recipes.

Extracting meaningful information from widely available recipes from the web, required to remove several stop words apart from the natural language stop words. For instance, terms like “pinch” and “mix” appear with the list of ingredient in a recipe needs to be removed to extract actual ingredient. We introduced a novel set of 572 stop words aligning with food ingredients and named that set as Ingredient Stop Words (ISW). Furthermore, validity of ISW is checked with three other cuisines i.e., Japanese, American and Italian. The use of these stop words proves instrumental in effectively extracting and refining ingredient names.

In summary, the paper presents

1. Proposal of INDoRI, a dataset of Recipes and Ingredients of Indian cuisines. It includes over 5K recipes with 18 different cuisines. The characteristics and possible applications of the data set are reported.
2. A novel set of stop words ISW for the culinary domain.
3. Construction of the Ingredient Network (InN) on top of INDoRI.
4. Demonstrated a research problem on InN with a simple weighted community detection algorithm (WABCD).

2 Literature Survey

Datasets: Over the course of time, numerous benchmark food datasets have been introduced in research literature. For instance, Matsuda et al. [8] introduced a Japanese food image dataset in 2012, encompassing a collection of 14,361 images. In 2014, Bossard et al. [9] released the ETHZ Food-101 dataset. The year 2016 saw the unveiling of a large dataset by Rich et al. [10] containing 800 thousand images. Many of these existing datasets are focused on images, although a few exceptions exist in the form of datasets oriented towards recipes. Notably, three recipe-centric datasets emerged in 2018. These are: a recipe question-answering dataset by Semih et al. [11], comprising approximately 36k questions that users can query against the dataset; Epic Kitchen, introduced by Damen et al. [12], featuring cooking videos and accompanying recipes; and the extensive “Recipe1M” dataset containing both recipes and images, brought forth by Salvador et al. [13]. The work of Salvador et al. [13] notably focuses on embedding recipes and images. Furthermore, they extended their dataset to create “Recipe1M+” [14].

Ingredient Network: Over time, researchers have explored ingredient networks in various contexts. One such study [15] resulted in the creation of two ingredient networks: “complement” and “substitute”. The complement network exhibited two distinct communities, one centered around savory ingredients and the

other around sweet ingredients. On the other hand, the substitute network was constructed based on user-generated suggestions, offering alternative ingredient choices for specific recipes. Similarly, another work [16] focused on two types of networks: ingredient-ingredient and recipe-ingredient networks. These networks were designed to recommend recipes to users based on the ingredients they had available. By analyzing the relationships between ingredients and recipes, the system could suggest suitable recipes that aligned with the user's resources. Apart from recipe recommendations, ingredient networks have also been applied to food recognition tasks. For instance, Min et al. [17] achieved food recognition by developing an innovative Ingredient-Guided Cascaded Multi-Attention Network. This approach utilized the ingredient network to enhance the accuracy of food recognition systems, leveraging the knowledge of the associations among the food ingredients. However, we introduced INDoRI, which distinguishes itself by encompassing not only recipes, ingredients, and cooking instructions, but also comprehensive cuisine information representative of the entirety of India.

3 Indian Dataset of Recipes and Ingredients (INDoRI)

Creating a comprehensive dataset of Indian cuisines possesses unique challenges. One of them is to compiling recipes that span diverse cultural landscape of India. Due to the same reason one may not find all the recipes from one single web portal. As there is no common data format available, each portal present data differently and the data are unstructured. Hence the second challenge is to extract meaningful information from it. We consider seven different recipe websites to address the first challenge. All the unstructured data therein are crawled using Python script. Basic cleaning is performed on the collected data and the following methodology is used to structured it using both tabular and network structures.

Identification of Novel Stop Words for Food Ingredients (ISW):

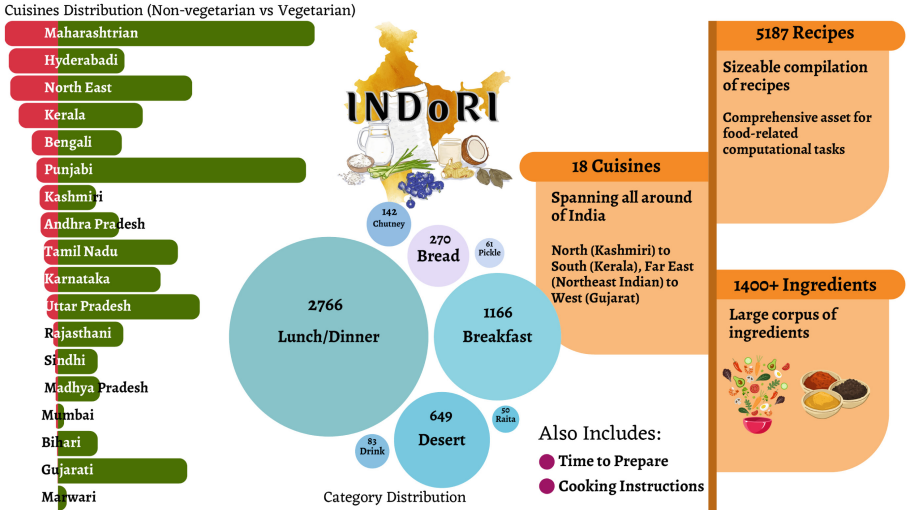
Amidst the data preparation phase, novel food-related stop words were introduced. Notable examples encompass 'kg,' 'gms,' 'cup,' 'tbls,' 'pinch,' 'chopped,' 'boiled,' 'sliced,' and 'split'. 527 specific keywords are identified, scrutinized, and extracted manually from the ingredient data. The validity of ISW was tested on other global cuisines, including Japanese [18], Italian [19] and American [20]. For each of the cuisine hundred recipes were taken along with the ingredients needed to prepare them. The ingredient names were extracted manually and through stop word removal using ISW. The details were reported in Table 1. A comprehensive breakdown of the calculations and results are presented in an online repository¹

Removal of Stop Words and Numbers: The exclusion of stop words and numerical values from ingredients yielded beneficial results in obtaining clean ingredient names. Solely ingredient names are employed to construct the

¹ Link to the supplementary material: <https://shorturl.at/gwzFN>.

Table 1. Cuisine wise accuracy statistics: ISW accuracy for global Cuisine.

Cuisine	Avg. Accuracy	Min. Accuracy	Max. Accuracy
Indian	81.98	80.0	92.85
Italian	75.42	68.75	83.33
Japanese	53.36	42.85	60.0
American	72.31	62.50	85.71

**Fig. 1.** Key Characteristics of INDoRI

ingredient network (InN). Nonetheless, these numerical values can potentially be considered for recommendation purposes hence kept separately.

Characteristics of INDoRI: INDoRI stands as a unique and innovative Indian recipes dataset, distinguishing itself from conventional counterparts. It contains a total of 5187 recipe, presenting a diverse array of culinary offerings. Additionally INDoRI encompasses additional attributes such as cuisine, category, and preparation time. All recipes are classified into 8 different types. Apart from 925 unclassified recipes rest are also categorized into 18 different cuisines. Figure 1 shows the key characteristics of INDoRI. In order to examine the inter-relationships among ingredients, we formed a network of ingredients referred to as the Ingredient Network (InN). Further information regarding this network is outlined in the subsequent Section.

3.1 Ingredient Network Construction

We constructed the ingredient network out of INDoRI where each node is an ingredient. A link is constructed when two ingredient appear in the same recipe.

Statistics	
Directed	No
Weighted	Yes
Nodes	1433
Edges	30464
Average Clustering Coefficient	0.8455
Number of Triangles	424048
Fraction of Closed Triangles	0.3485
Diameter	4
Average Edge Weight	39.7861

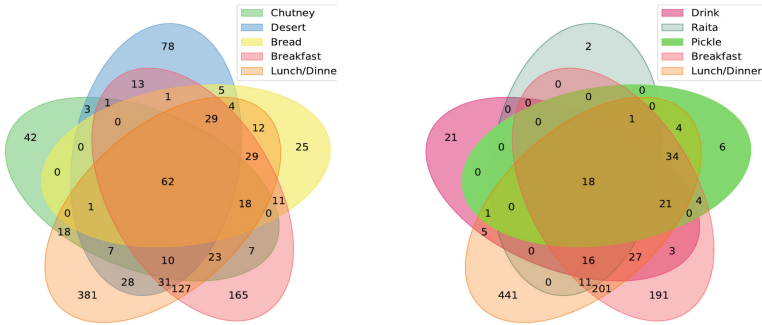


Fig. 2. Characteristics and statistics of InN and ingredients.

Total of 30,464 relationships were found among all ingredients. The ingredient network is a graph $G(V, E, w)$, where V is a set of ingredients, E is the connections between ingredients and $w : V \times V \rightarrow \mathbb{R}$ of an edge signifies the number of association between ingredients in different recipes. The more they appear together in diverse recipes, more stronger is the association. The strongest association, is between salt and oil, appearing together in 1958 recipes.

Characteristics of Ingredients and InN. Sample sub graph of InN is shown in top right of Fig. 2. Here thick edges represent stronger associations, while thinner edges represent weaker associations. The size of the node shows the degree. The bigger the size greater is the degree. Top left Table shows the statistics of the network InN. We also investigated the presence of ingredients in multiple recipe categories. The bottom images of Fig. 2 represents the ingredient overlaps. While the left image provide overlap across five recipe categories viz Chutney, Desert, Bread, Breakfast and Lunch/Dinner the right image shows the overlap among Drink, Raita, Pickle, Breakfast and Lunch/Dinner. It is evident that there were 62 ingredients shared among all categories in the left image and overlap of 18 ingredients is found in the right image. This gives an interesting observation of

the distinct communities for each category. The degree distribution of InN follows a power law, making it a scale-free network as shown in Figs. 3 (a) and (b) shows the cumulative degree distribution.

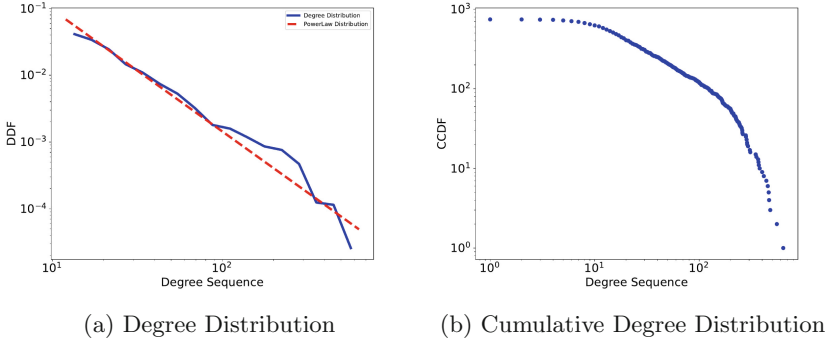


Fig. 3. Distribution Plot of InN.

3.2 Communities in InN

The average clustering coefficient of InN is measured as 0.8455, indicating a higher tendency for nodes to form clusters or groups within the network. We employed the weighted Leiden algorithm [21] to identify community structure of InN. The outcomes are presented in Sect. 4.1, highlighting that the network is partitioned into five distinct communities. We tried to uncover the inherent characteristics of each partition, leading us to recognize a distinct pattern. Specifically, we observed that the majority of categories, excluding dessert, are having strong associations with the first partition or community. Conversely, the exceptions displayed associations with the second partition. This observation presents a fascinating challenge for researchers to devise a weighted algorithm tailored for community detection within InN. Such an algorithm has the potential to identify diverse trends in the network structure.

4 Applications on INDoRI and InN

Food Computing is defined as the study of food and its properties using computational methods and methodologies [22]. One such method is modeling and simulation. It involves many tasks such as acquiring, analyzing, recognition [23], recommendation of food and recipes. Considering the characteristics of this dataset, researchers have the opportunity to delve into the tasks both on INDoRI and InN. Some of the potential applications are listed in Table 2. The outcomes of two applications of INDoRI are described in the online repository²

² Link to the supplementary material: <https://shorturl.at/gwzFN>.

Table 2. Potential Applications on INDoRI and InN.

INDoRI	
Application	Description
Recipe Categorization	Automatic categorization of recipes into categories such as breakfast, lunch, dinner etc. based on text descriptions like ingredients and cooking instructions.
Cuisine Classification	Automatic categorization of recipes into cuisines such as Punjabi, Bengali, Hyderabadi etc. based on text descriptions like ingredients and cooking instructions.
InN	
Application	Description
Community Identification	Algorithms to identify communities in the Ingredient Network (InN) where each community can be correlated with cuisine or category.
Ingredient Pair Prediction	Development of methods to predict occurring pairs of ingredients in recipes, aiding link prediction and recommendation.
INDoRI + InN	
Application	Description
Recipe Similarity	Design of techniques that measure recipe similarity or dissimilarity based on ingredient overlap, cooking techniques, and other attributes.
Ingredients based Recipe Recommendation	Proposal of recommendation algorithms predicting recipes based on ingredient availability, offering personalized suggestions

4.1 Example: Community Detection for Better Categorization of Ingredients

Addressing the challenge we have discussed in Sect. 3.2 for community identification in InN, we proposed a simple Weighted Association Based Community Detection (WABCD) algorithm that groups nodes based on the strong association between them. The input to the algorithm is a weighted graph $G(V, E, w)$ and it outputs community structure therein. The algorithm (Algorithm 1) works in the following manner. At the outset, every vertex denotes a unique community. In the first cycle, communities merge according to the most substantial weighted edge between two vertices. Starting from the second iteration, each vertex within a community is compared with vertices from other communities, and the average weight between the communities is calculated. Merging occurs based on the highest average value between two communities. The algorithm terminates when the average weight computed in an iteration is lower than that calculated in the previous iteration.

Comparison of WABCD with Baselines. We compare the proposed WABCD algorithm with other community detection algorithms. Baseline algorithms considered were weighted Leiden and weighted Louvain [24]. The results were shown in Fig. 4. The communities identified by weighed Leiden, Louvain and WABCD is 5, 4, 7 respectively. To uncover the inherent characteristics of each partition we have created multiple sub-graphs based on the category of recipes and compare them with the communities obtained from all three algorithms. The results were shown in Table 3. One may observed that with both

Algorithm 1: WABCD (G, V, E, w)

Input: G : InN graph, V : set of vertices of G , E : set of edges of G and w : set of weights on edges

Output: Acquired communities in *dictnew*

```

1 dictnew = {};
2 for i in range (0, len(V)) do
3   | dictnew[V[i]] = V[i];
4   end
5 while True do
6   dict = dictnew.copy();
7   for key1 in list(dict.keys()) do
8     bestinc = 0; c = 0; key = dict[key1];
9     for key2 in list(dict.keys()) do
10      newkey = dict[key2];
11      if key1 != key2 and len(key) > 0 and len(newkey) > 0 then
12        sumweight = 0;
13        for m in key do
14          for n in newkey do
15            if G.hasedge(m,n) then
16              sumweight = sumweight + G.getedgedata(m, n)[weight];
17              c+=1;
18            end
19            else
20              continue;
21            end
22          end
23        end
24        if c > 0 then
25          sumweight = sumweight / c;
26        end
27        accnode = sumweight;
28        if accnode > bestinc then
29          bestinc = accnode; k = newkey; q = key;
30        end
31      end
32    end
33  end
34  if bestinc > 0 then
35    if dictnew[key1] != -1 then
36      for qw in k do
37        | dictnew[key1].append(qw);
38      end
39      dictnew[key1] = -1;
40    end
41    delete dict[key1];
42  end
43 end
44 for key in list(dictnew.keys()) do
45   | if dictnew[key] == -1 then
46     | dictnew.pop(key);
47   end
48 end
49 if bestinc == 0 then
50   | break;
51 end
52 end

```

weighted Leiden and Louvain algorithm, the second community exhibit connection with recipe category Desert whereas the rest tend to have more association with Lunch/Dinner category. Conversely, the WABCD approach succeeds in identifying four prominent recipe categories: Bread, Lunch/Dinner, Drink, and Deserts. However the desired number of communities is 8 with overlap in between as shown in Fig. 2 and the problem remain open to solve.

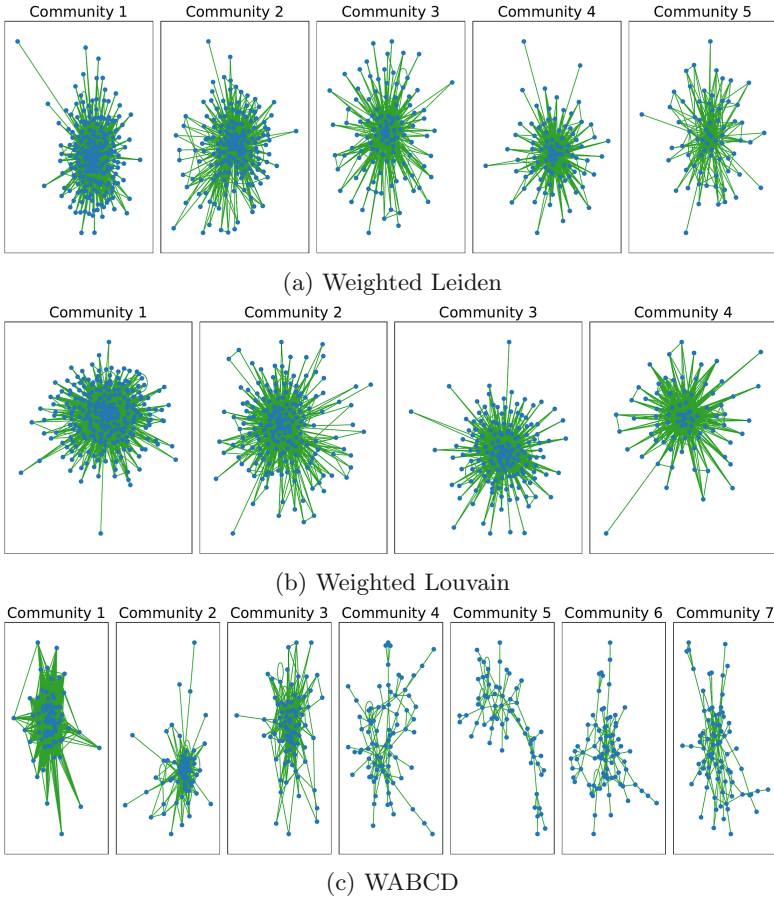


Fig. 4. Results from Different Community Detection Algorithms a) Weighted Leiden detects 5 communities b) Weighted Louvain detects 4 communities c) WABCD detects 7 communities

Table 3. Comparison of community detection algorithms

	Weighted Leiden	Weighted Louvain	WABCD
C1	Lunch/Dinner Recipes	Lunch/Dinner Recipes	Bread Recipes
C2	Desert Recipes	Desert Recipes	Bread Recipes
C3	Lunch/Dinner Recipes	Lunch/Dinner Recipes	Lunch/Dinner Recipes
C4	Lunch/Dinner Recipes	Lunch/Dinner Recipes	Drink Recipes
C5	Lunch/Dinner Recipes	-	Lunch/Dinner Recipes
C6	-	-	Dessert Recipes
C7	-	-	Lunch/Dinner Recipes

5 Conclusion

This paper presented our INDoRI dataset with a general characterization along with its ingredient network. We thoroughly examined and shown its distinctive features and attributes. Furthermore, we have put forth a set of novel stop words specifically tailored for the food ingredients. The creation of the Ingredient network (InN) from ingredient interconnections has been a focal point, with a comprehensive analysis on community identification. Our discourse extends to addressing the potential applications on top of INDoRI and InN. We present and compare the communities identified using WABCD and other baseline community detection algorithms. Overall, INDoRI and InN not only enriches our understanding of Indian cuisine but also opens up fresh avenues for research, encouraging a deeper exploration of its culinary intricacies.

References

1. Indian Food Forever. <https://nerdyfoodies.com/indian-spices-list-3291.html>. Accessed Jan 2023
2. Azmanov, V.: East Indian recipes. <https://eastindianrecipes.net>. Accessed Jan 2023
3. Dassana: Dassana's veg recipes. <https://www.vegrecipesofindia.com>. Accessed Jan 2023
4. Shreekanth, S.: Swasthi's recipes. <https://www.indianhealthyrecipes.com>. Accessed Jan 2023
5. Ahn, Y.-Y., Ahnert, S.E., Bagrow, J.P., Barabási, A.-L.: Flavor network and the principles of food pairing. *Sci. Rep.* **1**(1), 1–7 (2011)
6. Meredith Food Group: allrecipes. <https://www.allrecipes.com>. Accessed Jan 2023
7. Kapoor, S.: <https://www.sanjeevkapoor.com/>. Accessed Jan 2023
8. Matsuda, Y., Yanai, K.: Multiple-food recognition considering co-occurrence employing manifold ranking. In: Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), pp. 2017–2020. IEEE (2012)
9. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – mining discriminative components with random forests. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 446–461. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_29

10. Rich, J., Haddadi, H., Hospedales, T.M.: Towards bottom-up analysis of social food. In: Proceedings of the 6th International Conference on Digital Health Conference, pp. 111–120 (2016)
11. Yagcioglu, S., Erdem, A., Erdem, E., Ikizler-Cinbis, N.: RecipeQA: a challenge dataset for multimodal comprehension of cooking recipes. arXiv preprint: [arXiv:1809.00812](https://arxiv.org/abs/1809.00812) (2018)
12. Damen, D., et al.: Scaling egocentric vision: the epic-kitchens dataset. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11208, pp. 753–771. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01225-0_44
13. Salvador, A., et al.: Learning cross-modal embeddings for cooking recipes and food images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3020–3028 (2017)
14. Marin, J., et al.: Recipe1M+: a dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(1), 187–203 (2019)
15. Teng, C.-Y., Lin, Y.-R., Adamic, L.A.: Recipe recommendation using ingredient networks. In: Proceedings of the 4th Annual ACM Web Science Conference, pp. 298–307 (2012)
16. Nyati, U., Rawat, S., Gupta, D., Aggrawal, N., Arora, A.: Characterize ingredient network for recipe suggestion. *Int. J. Inf. Technol.* **13**, 2323–2330 (2021)
17. Min, W., Liu, L., Luo, Z., Jiang, S.: Ingredient-guided cascaded multi-attention network for food recognition. In: Proceedings of the 27th ACM International Conference on Multimedia, pp. 1331–1339 (2019)
18. deliciousteam: 100 Japanese recipes that aren't all sushi (2021). <https://www.delicious.com.au/recipes/collections/gallery/55-quick-and-easy-japanese-recipes-to-try-tonight/11y6brzi>. Accessed Jan 2023
19. TasteAtlas: 100 most popular Italian dishes. <https://www.tasteatlas.com/100-most-popular-dishes-in-italy>. Accessed Jan 2023
20. deliciousteam: 100 most popular American recipes. <https://www.deliciousmagazine.co.uk/cuisine/american-recipes>. Accessed Jan 2023
21. Traag, V.A., Waltman, L., Van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**(1), 1–12 (2019)
22. Min, W., Jiang, S., Liu, L., Rui, Y., Jain, R.: A survey on food computing. *ACM Comput. Surv. (CSUR)* **52**(5), 1–36 (2019)
23. Xu, R., Herranz, L., Jiang, S., Wang, S., Song, X., Jain, R.: Geolocalized modeling for dish recognition. *IEEE Trans. Multimedia* **17**(8), 1187–1199 (2015)
24. 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)



Optimizing Neonatal Respiratory Support Through Network Modeling: A New Approach to Post-birth Infant Care

Yassine Sebahi¹(✉), Fakhra Jabeen², Jan Treur¹, H. Rob Taal³,
and Peter H. M. P. Roelofsma³

¹ Social AI Group, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
y.sebahi@student.vu.nl, j.treur@vu.nl

² Safety and Security Section, Delft University of Technology, Delft, The Netherlands

³ Erasmus University Medical Center Rotterdam, Rotterdam, The Netherlands
h.taal@erasmusmc.nl, p.h.m.p.roelofsma@hhs.nl

Abstract. This paper presents an approach to enhancing neonatal care through the application of artificial intelligence (AI). Utilizing network-oriented modeling methodologies, the study aims to develop a network model to improve outcomes in neonatal respiratory support. The introduction sets the stage by outlining the significance of neonatal respiratory support and the challenges faced in this domain. The literature review delves into the existing body of work, highlighting the gaps and the need for a network modeling approach. The network-oriented modeling approach provides a robust framework that captures various states, such as world states, doctors' mental states, and AI coach states, facilitating a comprehensive understanding of the complex interactions in neonatal respiratory support. Through Matlab simulations, the study investigates multiple scenarios, from optimal conditions to deviations from standard protocol. The main contribution focuses on the introduction of an AI coach, which serves as a real-time intervention mechanism to fill in the doctor's knowledge gaps. The research serves as a seminal work in the intersection of artificial intelligence and healthcare, demonstrating the potential of network-oriented modeling in improving patient outcomes and streamlining healthcare protocols.

Keywords: Adaptive network model · Infant Care · AI Coach

1 Introduction

In the Netherlands, 166,891 babies were born in 2022. That is 457.24 per day (Cijfers over geboorte | Nederlands Jeugdinstituut 2023). Approximately 7% experience respiratory distress at birth, necessitating immediate and specialized medical intervention (Edwards and Kotecha 2013). This translates to a staggering number of infants requiring critical respiratory support each day, highlighting the urgency for effective and optimized neonatal care. Neonatal respiratory support is a critical aspect of infant care, as timely interventions can have significant impacts on survival and long-term health

outcomes (Kaltsogianni et al. 2023). With advancements in technology, Artificial Intelligence (AI) has emerged as a tool with potential applications across various healthcare domains, including neonatal care (Malak et al. 2018).

This paper explores the question, ‘How can the development and analysis of a network model, representing world, AI coach, and doctors’ mental states, provide insights into neonatal respiratory support through the simulation of the process of respiratory support of a newborn baby?’ The research builds on principles of network-oriented modeling by adaptive self-modeling networks (Treur, 2016; Treur 2020a, 2020b, 2020c), encompassing three key states and modeling three scenarios:

1. A Successful Scenario: Reflecting optimal processes as described in neonatal care guidelines.
2. An Error Scenario: Some deviation takes place: an often occurring error or omission.
3. An AI-Coached Error Detection & Knowledge Improvement Scenario: The AI coach detects an error and improves the knowledge of the doctor if needed.

By analyzing these scenarios, this paper aims to contribute to the understanding of the AI Coach’s role in optimizing neonatal respiratory support.

2 Background Literature

Neonatal respiratory support is a vital aspect of care for newborns, particularly in the moments immediately following birth. Roehr and Bohlin (2011) state that a protective respiratory support strategy from birth is essential as it may not only reduce breathing difficulties in the immediate neonatal period, but may also influence some known triggers for the development of BPD, such as inflammation, oxidative stress and lung growth. The importance of this intervention has been highlighted in various clinical guidelines, emphasizing the need for immediate assessment and support of breathing in newborns (Anne and Murki 2021). Advances in neonatal respiratory care have led to improved survival rates and outcomes for preterm infants and those with specific respiratory conditions.

Even with recent progress, there are still some hurdles in giving the best breathing support to newborns (Kaltsogianni et al., 2023b). These hurdles include identifying which babies need help, choosing the right treatments, deciding when to offer support, and avoiding mistakes. There’s also inconsistency in how treatments are given, making things even more complicated. However, technology like Artificial Intelligence (AI) could help overcome some of these issues (Kaltsogianni et al., 2023b). By using network models that show different situations related to breathing support, healthcare providers could get a clearer idea of how to best handle this crucial part of caring for newborns.

The integration of Artificial Intelligence (AI) into healthcare has marked a transformative era, revolutionizing various medical domains, from diagnostics to personalized treatment (Khan et al. 2022). The convergence of AI technologies with medical practices has led to improved efficiencies, enhanced patient outcomes, and the opening of new avenues for research and innovation. In the context of neonatal care, AI has demonstrated promising applications, including the analysis of complex medical data, predictive modeling for patient outcomes, and assistance in decision-making (Bajwa

et al. 2021). These applications extend to neonatal respiratory support, where timely and precise interventions are crucial.

One specific area where we can investigate if AI shows potential in the process of the respiratory support of neonatal is with the use of network modeling. This approach involves the construction of network models representing various states and relationships, enabling the simulation and analysis of different scenarios related to respiratory support (Treur 2016; Treur 2020a, 2020b, 2020c). For instance, network models can represent the world states, AI coach states, and doctors' states, each with specific roles and interactions. The development of such network models allows for a systematic exploration of neonatal respiratory support processes, including the simulation of optimal processes, common deviations, and AI-coached interventions. By leveraging the computational capabilities of AI, these models can provide insights, guide clinical decisions, and potentially optimize respiratory support for newborns. Network modeling and analysis in neonatal respiratory support offers a novel approach to understanding and enhancing care, with potential implications for both immediate neonatal outcomes and the future of technology-driven medical care.

The development of network models that represent various states and interactions is an innovative approach in healthcare, providing a computational framework to understand and analyze complex processes. In the context of neonatal respiratory support, these models can include states such as:

- **World States:** capturing states of the baby and the broader context and environment, including hospital settings, equipment, and external factors that may influence care.
- **AI Coach states:** Representing an intelligent entity that guides, monitors, and supports the healthcare process, offering insights and interventions when needed.
- **Doctors' mental and action states:** Reflecting the healthcare provider's actions, decisions, knowledge, and interactions with both the world and AI coach states.

These states and their interactions are covered by the network model, allowing for the simulation and analysis of different scenarios. The scenarios can include:

1. **Successful Processes:** Simulating the ideal process of neonatal respiratory support, serving as a baseline for understanding best practices and optimal outcomes.
2. **Common Deviations:** Modeling frequent errors or omissions, highlighting potential risks, and areas for improvement in care delivery.
3. **AI-Coached Error Detection and Knowledge Improvement:** Integrating an AI coach to detect and rectify mistakes in real-time, enhancing accuracy and safety. And utilizing AI to support healthcare workers in enhancing their knowledge, skills, and adherence to guidelines, thus improving overall care quality.

The ability to model and simulate these scenarios offers valuable insights into neonatal respiratory support, allowing for a nuanced understanding of the interactions and dependencies within the process. It opens opportunities for targeted interventions, continuous learning, and optimization of care, aligning with the broader goals of precision medicine and technology-driven healthcare.

By leveraging network modeling, this approach fosters a data-driven, evidence-based practice that transcends traditional boundaries, offering a new perspective on neonatal care and beyond.

Computational causal modeling is a powerful tool in AI that can help us understand complex healthcare situations better (Sarker 2022). In the case of helping newborns breathe, this type of modeling can map out how different factors like doctors, AI coaches, and the baby's condition interact. This approach is unique because it shows not only what directly causes what but also how changes in one area can affect the whole system (Squires and Uhler 2022). By using this method, researchers can simulate different outcomes, such as what happens when things go right, when they go wrong, how AI can spot mistakes, and how AI can help improve our knowledge (Campos and Fleury 2022).

This kind of modeling can help identify why certain treatments work or fail and point out where critical decisions should be made to improve care. The research aims to add to the growing field of network modeling in healthcare. The findings could impact not just how doctors treat newborns, but also broader healthcare policies and future studies, laying the groundwork for improving the care of newborns overall.

In conclusion, the development and application of network modeling, coupled with computational causal modeling, represent a novel and promising avenue in neonatal respiratory support. By drawing on relevant literature and innovative methodologies, this research aims to shed new light on the complexities of neonatal care and pave the way for technology-driven improvements in this vital area of healthcare.

3 Modeling Approach for Neonatal Respiratory Support

The research conducted for this paper employs a network-oriented modeling approach to understand and analyze the complex interactions and processes in neonatal respiratory support (Weigl et al. 2023). This methodology encompasses various states, such as the world, AI coach, and doctor states, capturing the interactions and causal impacts within the system. Key features characterize the structure of the network (Weigl et al. 2023). State are often indicated by X and Y ; they have activation values (real numbers, usually in the interval $[0, 1]$) $X(t)$ and $Y(t)$ that vary over time t . *Connectivity Characteristics* specify connections from a state X to a state Y as defined by their weights $\omega_{X,Y}$, symbolizing the strength of the causal impact from X to Y . *Aggregation Characteristics* are specified for any state Y by a combination function $\mathbf{c}_Y(\dots)$ outlines the aggregation applied to the single causal impacts $\omega_{X,Y}X(t)$ on Y from its incoming connections from states X . *Timing Characteristics* specify for each state Y a speed factor η_Y , indicating how quickly it changes for a given causal impact.

Based on these network characteristics a standard numerical format described by the difference equation defines the dynamics of the network model:

$$Y(t + \Delta t) = Y(t) + \eta_Y[\mathbf{c}_Y(\omega_{X_1,Y}X_1(t), \dots, \omega_{X_k,Y}X_k(t)) - Y(t)]\Delta t$$

Various combination functions are available to handle the aggregation of multiple impacts, with the specific functions used here detailed in Table 1.

The modeling approach also includes the concept of network reification or self-modeling network, extending the base model by additional states, referred to as reification states or self-model states (Treur 2020a, 2020b, 2020c). Examples are self-model states $\mathbf{W}_{X,Y}$, \mathbf{C}_Y , \mathbf{H}_Y (reification states) to represent the adaptive network structure characteristics $\omega_{X,Y}$, \mathbf{c}_Y , η_Y for a state Y of the base network. Such self-model states are called

W-states, **C**-states and **H**-states, respectively. This can be iterated to get higher-order self-model states. For example, the self-model state $\mathbf{W}_{\mathbf{W}_{X1,Y1}, \mathbf{W}_{X2,Y2}}$ is a second-order self-model state that indicates the weight of a communication channel from $\mathbf{W}_{X1,Y1}$ to $\mathbf{W}_{X2,Y2}$. This will be used in the introduced model for communication from AI Coach to Doctor.

This network-oriented methodology provides a robust framework to explore and analyze scenarios related to neonatal respiratory support. The modeling and simulations are conducted using Matlab, providing a comprehensive approach to simulating and analyzing the scenarios related to neonatal care. In the development of an network model for optimizing neonatal respiratory support, two essential mathematical functions as shown in Table 1 have been employed within the MATLAB environment.

Table 1. Combination functions used

Function	Notation	Formula	Parameters
Advanced logistic sum	$\mathbf{alogistic}_{\sigma, \tau}(V_1, \dots, V_k)$	$\left[\frac{1}{1 + e^{-\sigma(V_1 + \dots + V_k - \tau)}} - \frac{1}{1 + e^{\sigma\tau}} \right] (1 + e^{-\sigma\tau})$	steepness σ threshold τ
Identity	$\mathbf{id}(V_1, \dots, V_k)$	V_1	–

The advanced logistic sum function represents a nonlinear transformation that takes the weighted sum of the input variables V_1, \dots, V_k for incoming single causal impacts and applies a logistic function. This function is characterized by two parameters: the steepness σ and the threshold τ . The steepness parameter σ controls the slope of the logistic curve, whereas the threshold parameter τ determines the point at which the function transitions from one state to another. In the context of neonatal respiratory support, this function can be utilized to model complex relationships and transitions within the respiratory system, such as the response to different ventilatory support parameters. The identity function is a straightforward mathematical transformation that just returns the input value itself. In terms of the respiratory support model, the identity function can represent parameters or variables that are directly observed or controlled without the need for transformation or scaling. Together, these two functions serve critical roles within the network model. The advanced logistic sum provides the capability to capture nonlinear dynamics and complex relationships within the respiratory system, while the identity function ensures that certain aspects of the system can be modeled in a direct and unaltered manner. They enable the creation of sophisticated graphs that can be analyzed to better understand the underlying mechanisms of neonatal respiratory support. The ultimate goal is to leverage these insights to develop more effective and personalized interventions for post-birth infant care.

In (Appendix A, 2023) tables with explanations for all states and for the role matrices that are used for the simulation of the scenarios can be found. Here we explain the base world states. The pathway shown in Fig. 1 is followed. The red cross represents what happens from Scenario 3, where the doctor has no knowledge that you have to follow certain instructions after gasping, or he forgets to do this.

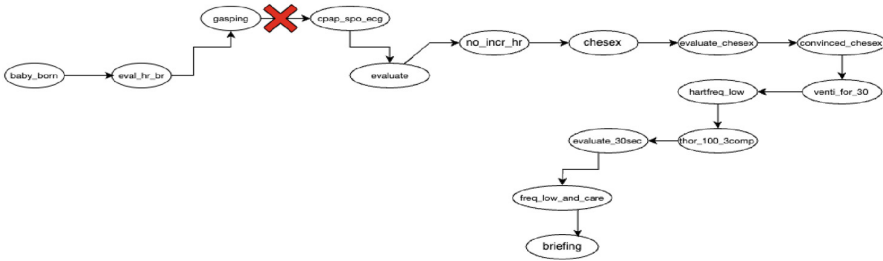


Fig. 1. The base level world states in the network model

The concept of a world state encompasses the environment in which neonatal respiratory support takes place. Understanding the world state is pivotal for the realistic simulation of scenarios that aim to optimize neonatal respiratory support. Within the network model, various states coexist to collectively influence the outcome of respiratory support for a neonate. These states can be categorized as follows (Table 2):

- **Context States:** These states provide information on specific conditions that could influence respiratory support. For example, Context State G Indicates that the baby is gasping in this scenario.
 - **Evaluation States:** These states, like `eval_hr_br` and `evaluate_30sec`, are pivotal for ongoing assessment of the baby’s physiological parameters.
 - **Intervention States:** These states dictate the medical interventions that should be considered, such as `cpap_spo_ecg` and `infl_spo_ecg`.
 - **Outcome States:** These states represent the outcomes of previous actions and evaluations, like `true_incr_hr` and `hartfreq_high`.
- `eval_hr_br` (Evaluate heart rate, breathing, color, and muscle tone): This state is crucial for the initial evaluation post-birth. It encapsulates the assessment of multiple physiological parameters to decide the subsequent course of action.
 - `infl_spo_ecg` (Open airway, give 5 inflation breaths (30 cm H2O), SpO2 and ECG monitoring): This state outlines the protocol for cases where initial assessment indicates respiratory distress, thereby requiring inflation breaths and continuous monitoring.
 - `evaluate_30sec` (Evaluate heart rate every 30 s): This state underscores the necessity for frequent re-evaluation to adapt the treatment strategy effectively.

The states are not static but interact dynamically within the network model. For example, if the state `hartfreq_low` is activated, the network transition to `thor_100_3comp` for immediate intervention. This dynamic interplay is essential for simulating the real-world complexity of neonatal respiratory care. The granularity and complexity of these states make them ideal candidates for network modeling. By applying advanced functions like $\text{alogistic}_{\sigma, \tau}$ for nonlinear relationships and id for direct variables, the model can simulate intricate scenarios that mimic real-life conditions. These simulations, therefore, hold the potential to significantly improve neonatal respiratory support protocols.

In this part, we’ll look at the roles of doctors and AI coaches in helping newborns breathe. See also Fig. 2, these roles are complex and include everything from the decisions

Table 2. World states and their explanations

State name	Description
context state N	Whether or not there is: no_incr_hr
context state L	Whether or not there is: hartfreq_low
context state G	Whether or not there is: gasping
context state I	Whether or not there is: inadequate
context state T	Whether or not there is: true_incr_hr
context state H	Whether or not there is: hartfreq_high
baby_born	Birth
eval_hr_br	Evaluate heart rate, breathing (color and muscle tone)
inadequate	Inadequate breathing
gasping	Gasps or apnea
cpap_spo_ecg	Open airway, consider CPAP SpO2 and ECG monitoring
infl_spo_ecg	Open airway, give 5 inflation breaths (30 cm H2O) SpO2 and ECG monitoring
evaluate	Evaluate heart rate
no_incr_hr	No increase in heart rate
true_incr_hr	Increase in heart rate
chesex	Check head and mask position. Consider alternate airway strategies. Repeat 5 inflation breaths
evaluate_chesex	Evaluate whether chest excursions had an effect on heart rate
convinced_chesex	Convinced of chest excursions
venti_for_30	Ventilation for 30 s
hartfreq_high	Heart rate is higher than 60/min
hartfreq_low	Heart rate is less than 60/min
thor_100_3comp	Start chest compressions. Increase oxygen percentage to 100%. 3 compressions on 1 breath
evaluate_30sec	Evaluate heart rate every 30 s
freq_low_and_care	If heart rate < 60/min: Provide i.v. access and give adrenaline Consider other causes (such as pneumothorax, hypovolemia, congenital abnormalities)
briefing	Inform parents, debrief with team and register

healthcare providers make to the medical guidelines they follow. The idea is to understand how doctors think and act in these situations. For example, if a doctor knows that a baby is having trouble breathing, they would follow a specific treatment plan, known as the CPAP_SPO_ECG procedure. This decision is based on the doctor's existing knowledge

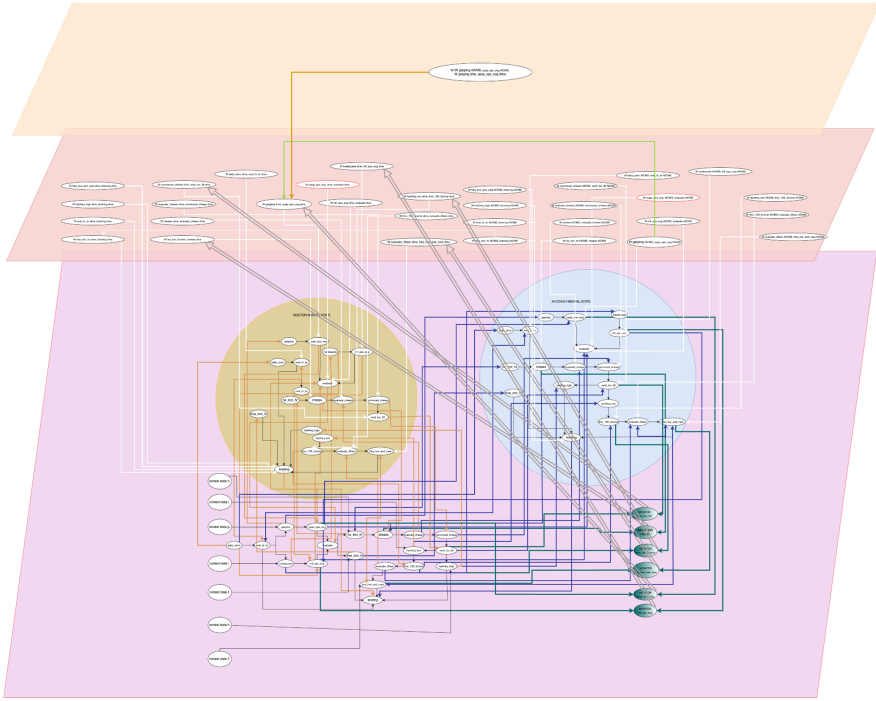


Fig. 2. The introduced overall network model

and experience. By breaking down the thought processes and actions of healthcare providers in this way, we can get a clearer picture of how decisions are made and treatments are administered in neonatal respiratory care.

So, if the \mathbf{W} -state of this relation has no value, the doctor would not know that it needs to do CPAP_SPO_ECG after he diagnoses that the baby has gasping. The doctor states encapsulate various functionalities:

- Intervention States: Such as cpap_spo_ecg (doctor MS) and infl_spo_ecg (doctor MS), dictating specific medical actions.
- Assessment States: Focused on continuous evaluations, e.g., evaluate (doctor MS) and evaluate_30sec (doctor MS).
- Outcome States: Representing the results of interventions, like hartfreq_high (doctor MS) and hartfreq_low (doctor MS).

In our model, \mathbf{W} -states serve as quantitative indicators of the doctor’s level of knowledge, confidence, or belief regarding the relationship between other states. For example:

- $\mathbf{W}_{\text{baby_born dms,eval_hr_br dms}}$ represents the doctor’s confidence in the necessity of immediate evaluations like heart rate and breathing following birth.
- $\mathbf{W}_{\text{gasping dms, cpap_spo_ecg dms}}$ reflects the level of belief the doctor has in initiating CPAP and monitoring when gasping or apnea is detected in a newborn.

These **W**-states are crucial for capturing the cognitive landscape of the medical practitioner, incorporating both objective knowledge and subjective beliefs into the decision-making model. Understanding the Doctor States, especially the cognitive aspects captured by **W**-states, is pivotal for our model aiming to interface effectively with healthcare providers. By modeling these intricate cognitive processes, the AI system can be trained to offer real-time, knowledge-aligned recommendations that can improve neonatal respiratory care outcomes.

The AI coach functions by monitoring various states and **W**-states in real-time, identifying gaps in the practitioner's actions or knowledge, and providing timely interventions to enhance learning and improve patient care. Learning states are specialized **W**-states for the AI Coach that interact with the corresponding **W**-states in the doctor model. They serve as the mechanism through which the AI Coach improves the practitioner's knowledge and decision-making. When the AI Coach detects a gap or a deviation in the doctor's actions, it uses these learning states to adjust the doctor's weight states, thus facilitating learning and improvement.

This goes as follows. The AI coach continuously monitors the doctor's actions. When it identifies a lapse, such as the doctor forgetting to initiate `cpap_spo_ecg` upon detecting gasping, it triggers the learning **W**-state

$$\mathbf{W}_{\text{gasping AICMS, cpap_spo_ecg AICMS}}; \mathbf{W}_{\text{gasping dms, cpap_spo_ecg dms}}$$

This second-order self-model state models a communication channel from AI Coach to Doctor that adjusts the corresponding doctor's **W**-state, $\mathbf{W}_{\text{gasping dms, cpap_spo_ecg dms}}$, to fill in the knowledge gap. This adjustment informs the doctor of the necessary action, thereby enhancing the doctor's knowledge and improving patient outcomes. In a scenario where a newborn is detected to be gasping, and the doctor fails to initiate `cpap_spo_ecg`, the AI coach monitors this and intervenes. Through this learning state, the AI coach updates the doctor's corresponding **W**-state, in turn making them aware of the need to initiate CPAP, thereby facilitating immediate and appropriate medical intervention.

The incorporation of an AI coach equipped with learning states into the network model offers several advantages:

- **Real-Time Intervention:** The AI coach provides immediate feedback, allowing for real-time adjustments in the doctor's actions.
- **Knowledge Enhancement:** The learning states serve as a conduit for knowledge transfer from the AI coach to the medical practitioner, ensuring that the doctor is always updated on the best course of action.
- **Adaptive Learning:** The model can adapt and evolve over time, capturing the nuances of each practitioner's learning curve and adjusting its coaching strategy accordingly.

By effectively utilizing learning states, the model becomes an invaluable tool for continuous professional development, ensuring that healthcare providers are always at the forefront of medical knowledge and practice, ultimately leading to improved patient outcomes.

Monitor states serve as an integral part of the network model, capturing real-time or near-real-time observations or measurements from the system. In the context of neonatal respiratory support, these states are crucial for continuously assessing various conditions

and parameters. They provide the data that informs the weight states, thus influencing the medical practitioner's decision-making process. The primary role of monitor states is to provide timely and accurate data for various attributes or conditions that are crucial in neonatal care. This data is then used to adjust the **W**-states, which represent the level of confidence or belief a medical practitioner might have in certain protocols or interventions. For example, if the doctor forgets to do the process that comes with the state `cpap_spo_ecg`, the monitoring of the AI coach will make sure that the doctors knowledge about this will be updated. Examples of monitoring states are:

- **MONITOR** `cpap_spo_ecg`: This monitor state observes the effectiveness of CPAP (Continuous Positive Airway Pressure) along with SpO2 and ECG monitoring. The data collected helps in dynamically adjusting the weight state `W_gasping_dms`, `cpap_spo_ecg_dms`, which influences the decision to initiate or continue CPAP.
- **MONITOR** `freq_low_and_care`: This state keeps track of the frequency and quality of care provided when the heart rate is below 60/min. The information is then used to adjust the weight at `W_evaluate_30_s_dms`, `freq_low_and_care_dms`, affecting the urgency and type of interventions considered.

Incorporating monitor states allows the AI system to make real-time adjustments based on current observations, making the model more adaptive and robust. These states serve as a bridge between the realworld conditions and the weight states, providing a dynamic feedback loop that enhances the model's predictive and decision-support capabilities. By understanding and effectively utilizing these monitor states, the model can offer more precise, timely, and context-sensitive recommendations, contributing to improved outcomes in neonatal respiratory care.

4 Findings from Network Model Simulations

This section presents the findings derived from simulations of the network model using Matlab. These simulation results are visualized as graphs, offering valuable insights into the model's effectiveness across various scenarios. The primary aim of this research is to address specific errors commonly made during the respiratory support of neonatal infants. In our Matlab simulations, we concentrated on rectifying a particular error: the omission of the `cpap_spo_ecg` action by doctors upon recognizing that the baby is gasping.

Importantly, all actions that a doctor can take are structured similarly within the network model. For the sake of simplicity and focus, we chose to zero in on the `cpap_spo_ecg` process. We posit that if our solution proves effective for this process, it should be generalizable to other processes as well. For all scenarios, see (Appendix A, 2023). Here we focus on Scenario 3. This scenario involves an AI coach that not only detects the error or omission with the help of monitor states but also aids healthcare workers in improving their knowledge.

In this scenario, both the `cpap_spo_ecg` state and the corresponding **W**-state representing the doctor's knowledge remain deactivated. However, the introduced AI Coach is connected to the doctor's knowledge **W**-state. A specialized higher-order **W**-state denoted as $\mathbf{W}_{\text{gasping AICMS, cpap_spo_ecg AICMS, W}_{\text{gasping dms, cpap_spo_ecg dms}}$ is in place to

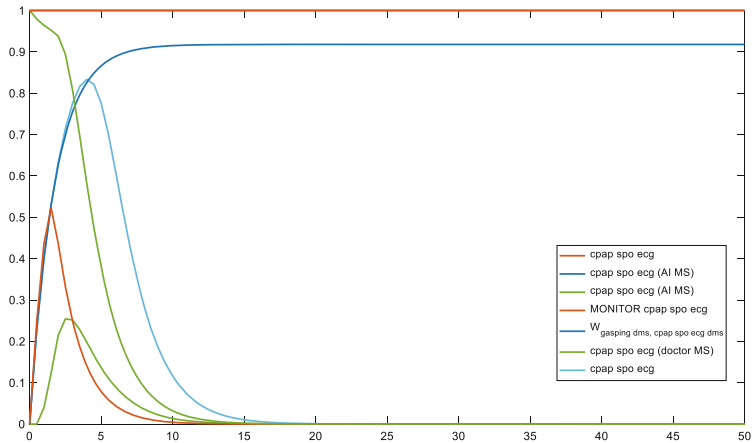


Fig. 3. Scenario 3 focused

facilitate the transfer of knowledge from the AI Coach to the doctor when needed. Essentially, if the state values are already optimal (e.g., knowledge value is 1), the AI coach will not intervene, and the monitor state will remain in observational mode without triggering any actions. The inclusion of the specialized higher-order **W**-state provides an opportunity for the doctor to gain knowledge from the AI Coach. As observed in Fig. 3, the doctor’s knowledge initially starts at 0 but increases to 0.9 due to the input from the AI coach. The concept underlying the connection between the AI coach’s knowledge and the doctor’s knowledge is that a single intervention should suffice for knowledge improvement. In other words, if the doctor receives guidance from the AI coach once, that guidance should be sufficient for future situations, thus negating the need for repeated AI interventions for the same action. The monitor state value for `cpap_spo_ecg` is notably low in Fig. 3, where the doctor’s knowledge stands at 0. Interestingly, the monitor state

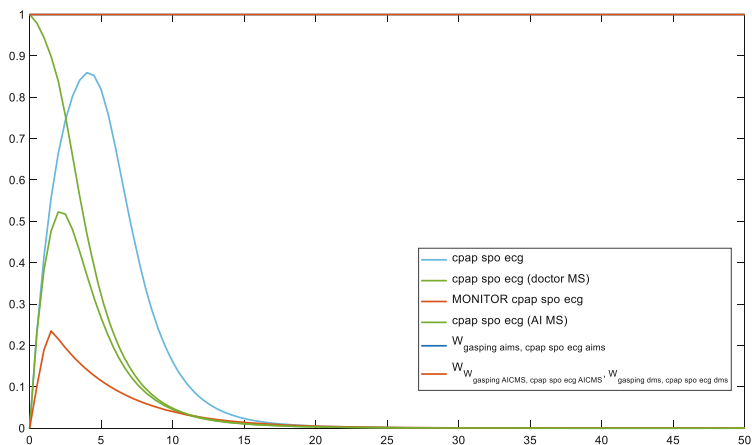


Fig. 4. `cpap_spo_ecg` knowledge is already adequate

value escalates significantly when the AI coach imparts knowledge to the doctor. This monitor state serves as an error-detection mechanism, intervening only when necessary. As evidenced in Fig. 4, when the doctor's knowledge level is already at 1 and the cpap_spo_ecg procedure has been performed, the monitor state peaks at 0.24. In contrast, in the absence of the doctor's knowledge, the peak is 0.52. In this instance, the monitor state only checks for errors and does not take any further action, as the necessary conditions are already met.

5 Discussion

The primary objective of this paper was to investigate how network modeling could optimize neonatal respiratory support protocols. Utilizing a network-oriented modeling approach as outlined in (Treur 2020a, 2020b, 2020c), various scenarios were created and analyzed within the Matlab environment. The findings from these scenarios contribute significantly to both the fields of neonatal respiratory support and network modeling. The second scenario emphasized the interconnectedness within the model. It revealed that missing links or incomplete knowledge could have far-reaching implications, affecting multiple aspects of neonatal respiratory support. This finding underscores the need for comprehensive and accurate data in models. The third scenario illustrated the potential of incorporating an AI coach into the model. The AI can not only act as a fail-safe tool but also as an educational tool. It provided real-time decision-making support and reinforced the doctor's knowledge base for future scenarios. The AI coach's intervention is a one-time requirement for each specific action or decision, equipping the doctor for future similar situations without additional AI assistance. Monitor states proved effective as safeguards, ensuring optimal performance and error minimization across different scenarios.

These findings provide evidence that an AI coach can be successfully applied to healthcare settings, particularly in the area of neonatal respiratory support. Some limitations concern that scaling up has not been addressed yet, the effectiveness has not yet been validated, and only some scenarios have been explored. Moreover, when dealing with healthcare data, you are primarily dealing with Personal Health Information (PHI), a category of data that is highly sensitive and heavily regulated to protect individuals' privacy. Health data can be a target for cyber attackers. Thus, robust security measures need to be in place to prevent unauthorized access and protect against data breaches. Future Research may address such limitations. For further details, see (Appendix 2023).

References

- Anne, R.P., Murki, S.: Noninvasive respiratory support in neonates: a review of current evidence and practices. *Indian J. Pediatr.* **88**(7), 670–678 (2021)
- Appendix: Linked Data at <https://www.researchgate.net/publication/373775834> (2023)
- Bajwa, J., Munir, U., Nori, A.V., Williams, B.: Artificial intelligence in Healthcare: Transforming the practice of medicine. *Future healthcare journal* **8**(2), e188–e194 (2021)
- Campos, A. B. A., Fleury, A. T.: Modeling, control strategies and design of a neonatal respiratory simulator. In: Bastos-Filho, T.F., de Oliveira Caldeira, E.M., Frizzera-Neto, A. (eds.) CBEB 2020. IP, vol. 83, pp. 563–571. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-70601-2_87

- Cijfers over geboorte | Nederlands Jeugdinstituut. <https://www.nji.nl/cijfers/geboorte> (2023, 2 juni)
- Edwards, M., Kotecha, S.: Respiratory distress of the term newborn infant. *Paediatr. Respir. Rev.* **14**(1), 29–37 (2013). <https://doi.org/10.1016/j.prrv.2012.02.002>
- Kaltsogianni, O., Dassios, T., Greenough, A.: Neonatal Respiratory Support Strategies—short and long-term respiratory outcomes. *Frontiers in Pediatrics* **11**. (2023a)
- Khan, M., Khurshid, M., Vatsa, M., Singh, R., Duggal, M., Singh, K.: On AI Approaches for Promoting Maternal and neonatal health in Low resource Settings: a review. *Frontiers in Public Health*, **10** (2022). <https://doi.org/10.3389/fpubh.2022.880034>
- Malak, J.S., Zeraati, H., Nayeri, F., Safdari, R., Shahraki, A.D.: Neonatal Intensive care decision support systems using artificial intelligence techniques: a systematic review. *Artif. Intell. Rev.* **52**(4), 2685–2704 (2018)
- Roehr, C.C., Bohlin, K.: Neonatal resuscitation and respiratory support in prevention of bronchopulmonary dysplasia. *Breathe* **8**(1), 14–23 (2011)
- Sarker, I.H.: AI-Based modeling: techniques, applications and research issues towards automation, intelligent and smart systems. *SN Comput. Sci.* **3**(2), 158 (2022)
- Squires, C., & Uhler, C.: Causal Structure Learning: A Combinatorial Perspective. *Foundations of Computational Mathematics*, 1–35. <https://doi.org/10.1007/s10208-022-09581-9> (2022)
- Treur, J.: Modeling higher-order adaptivity of a network by multilevel network reification. *Network Sci.* **8**, 110–144 (2020)
- Treur, J.: Network-Oriented modeling for Adaptive networks: Designing Higher-Order Adaptive biological, mental and social network models. *Springer Nature*. (2020). <https://doi.org/10.1007/978-3-030-31445-3>
- Treur, J.: Modeling multi-order adaptive processes by self-modeling networks (Keynote Speech). In: Antonio, J., Tallón-Ballesteros, Chen, C.-H. (eds.) *Proceedings of the 2nd International Conference on Machine Learning and Intelligent Systems, MLIS'20*. *Frontiers in Artificial Intelligence and Applications*, vol. 332, pp. 206 – 217. IOS Press (2020c)
- Weigl, L., Jabeen, F., Treur, J., Taal, H.R., Roelofsma, P.: Modelling learning for a better safety culture within an organization using a virtual safety coach: Reducing the risk of postpartum depression via improved communication with parents. *Cogn. Syst. Res.* **80**, 1–36 (2023). <https://doi.org/10.1016/j.cogsys.2023.01.009>



Generalized Gromov Wasserstein Distance for Seed-Informed Network Alignment

Mengzhen Li^(✉) and Mehmet Koyutürk

Department of Computer and Data Sciences, Case Western Reserve University,
Cleveland, OH, USA
{mx1994,mxk331}@case.edu

Abstract. Network alignment is a commonly encountered problem in many applications, where the objective is to match the nodes in different networks such that the incident edges of matched nodes are consistent. Gromov-Wasserstein (GW) distance, based on optimal transport, has been shown to be useful in assessing the topological (dis)similarity between two networks, as well as network alignment. In many practical applications of network alignment, there may be “seed” nodes with known matchings. However, GW distance assumes that no matchings are known. Here, we propose Generalized GW-based Network Alignment (GGWNA), with a loss/distance function that reflects the topological similarity of known matching nodes. We test the resulting framework using a large collection of real-world social networks. Our results show that, as compared to state-of-the-art network alignment algorithms, GGWNA can deliver more accurate alignment when the seed size is small. We also perform systematic simulation studies to characterize the performance of GGWNA as a function of seed size and noise, and find that GGWNA is more robust to noise as compared to competing algorithms. The implementation of GGWNA and the Supplementary Material can be found in <https://github.com/Meng-zhen-Li/Generalized-GW.git>.

Keywords: Gromov-Wasserstein Distance · Network Alignment

1 Introduction

Network alignment is the problem of aligning nodes that belong to the same entity from different networks based on the similarity of their connections [14]. In social networks, network alignment is often used to match the users that are the same person [12]. In biological networks, network alignment is used to identify molecules with similar evolutionary history and/or biological function [8].

Gromov-Wasserstein (GW) distance [9] is a measure that aims to quantify the distance between two networks (or similarity matrices) based on their topological (dis)-similarity. The formulation of GW derives an optimal transport

(OT) [16], which compares probability distributions and minimizes the transport cost between the distributions [11]. There are many existing variations of GW distance. Entropic GW distance [10] introduces an entropic regularizer to the loss function. Sliced GW [15] projects each distribution in an 1D form and improves efficiency.

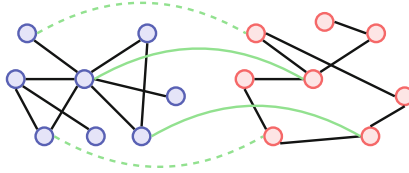


Fig. 1. Illustration of the seed-informed network alignment problem. Given the blue and red networks, and the known mappings of some nodes in the networks (solid green lines), the objective is to identify mappings of other nodes (dashed green lines) to maximize topological consistency.

The computation of GW distance between two networks also entails computation of a fuzzy mapping (the “transport” matrix) between the nodes of the two networks, which is useful for network alignment. Motivated by this observation, many recent studies develop GW-based methods for network alignment [2, 17]. GW is also shown to be useful in computing node embeddings for multiple networks, by jointly performing graph alignment and node embedding [17].

The classical formulation of GW distance and its existing variations assume that the mapping between the nodes of the two networks is unknown (or irrelevant) and formulate the optimization problem purely based on topology. However, in graph alignment applications involving real-world networks, there are some known matchings (Fig. 1), which can be used as prior knowledge in computing the mapping of remaining node pairs [4]. In this paper, we propose a novel framework for Gromov-Wasserstein based network alignment and introduce a new loss function that takes into account the known matchings between the two networks as “seed nodes” used to guide the alignment process. The proposed “generalized Gromov-Wasserstein distance” fixes the known matching of seed nodes in the optimal transport, while incorporating the topological consistency of these nodes in the loss function. We comprehensively assess the performance of the proposed Generalized Gromov-Wasserstein-based Network Alignment (GGWNA), in comparison to standard GW-based alignment, as well as other network alignment algorithms [5, 18]) on a rich corpus of social networks and synthetic datasets. We also investigate the effect of several factors and hyperparameters on the performance of GGWNA and other algorithms: 1) the number of seed nodes that are available, 2) the node overlap between the networks, 3) the divergence of edges between the two networks, and 4) the relative importance assigned to the topological consistency between seed vs. free matchings in our loss function. Our results show that (i) the use of seed matchings greatly improves the accuracy of GW-based alignment, (ii) GGWNA performs better when more

attention is given to the topological consistency of the seed nodes, (iii) GGWNA is drastically more robust than non-GW based algorithms to small seed sizes and more divergence between the networks. These results establish GW-based algorithms as a compelling alternative for seed-driven network alignment, while also enabling computation of GW distance for a broader range of networks.

2 Background

2.1 Optimal Transport

Optimal transport [10] minimizes the mapping cost between two probability distributions. Suppose that $p \in \mathbb{R}_+^m$ and $q \in \mathbb{R}_+^n$ are two distributions, given a cost matrix $C_{ij} \in \mathbb{R}^{m \times n}$ representing the transport cost from i to j . The optimal transport problem aims to find a matrix T to minimize the transport cost:

$$\text{minimize } \sum_{i,j} C_{i,j} T_{i,j} \quad \text{subject to: } T \in \mathbb{R}_+^{m \times n} : T \mathbb{1}_m = p, T^T \mathbb{1}_n = q. \quad (1)$$

This optimization problem can be solved using quadratic optimization[13]. In our experiments, we compute the optimal transport using the Python Optimal Transport (POT) package [3].

2.2 Gromov-Wasserstein Distance

Based on the optimal transport theory, GW distance [9] was proposed as a measure to quantify the (dis)similarity between two matrices. GW distance is defined between (C_1, p) and (C_2, q) , where C_1 and C_2 are two similarity matrices that represent the pairwise similarities or distances of elements, p and q are the two distributions that represent the relative importance of the elements [10].

This representation can be applied to quantifying the (topological) dissimilarity between two networks G_1 and G_2 , as $C_1 \in \mathbb{R}^{m \times m}$ and $C_2 \in \mathbb{R}^{n \times n}$ can be selected as the adjacency matrices of G_1 and G_2 . In its most general setting, the GW distance between two adjacency matrices C_1 and C_2 is defined as:

$$GW(C_1, C_2, p, q) = \min_T \sum_{i,j,k,l} L(C_1(i,k), C_2(j,l)) T(i,j) T(k,l) \quad (2)$$

where i and k refer to nodes in G_1 , j and l refer to nodes in G_2 , p and q are vectors representing the relative importance of the nodes in the two networks, $L(\cdot)$ is a loss function, and T is constrained by p and q as in (1). In common applications of Gromov-Wasserstein based network distance, quadratic loss $L(a, b) = \frac{1}{2}|a - b|^2$ is used along with uniform distributions for p and q , i.e., $p = \frac{1}{m} \mathbb{1}_m$ and $1 = \frac{1}{n} \mathbb{1}_n$ [10].

2.3 Network Alignment Problem

Network alignment aims to find a matching between the nodes of two networks, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ to maximize the consistency of the incident edges of matched nodes. Network alignment algorithms differ in terms of how they formulate an objective function to reflect this aim, as well as how they solve the resulting optimization problem(s) [5, 6, 18]. Network alignment algorithms can be supervised [18] or unsupervised [5]. GW-based network alignment formulates the problem as an optimization problem as in (2), where T represents the resulting mapping of the nodes. Here, we consider the seeded variant of the problem, where the matching between a subset of nodes $S = V_1 \cap V_2$ is known. The objective of seeded network alignment is to find a mapping between the nodes in $V_1 - S$ and $V_2 - S$ to maximize topological consistency.

3 Methods

3.1 Generalized Gromov-Wasserstein with Known Matching Nodes

Suppose that we have two networks $G_1 = \{V_1, E_1\}$ and $G_2 = \{V_2, E_2\}$ such that a subset of nodes $S = V_1 \cap V_2$ is common. We aim to compute $T, |V_1 - S| \times |V_2 - S|$ transport matrix such that $T(i, j)$ provides a mapping of the remaining nodes that maximizes topological consistency of the networks, given S .

Generalized Gromov-Wasserstein Distance. Let C_1 and C_2 denote the adjacency matrices of G_1 and G_2 . Reorganize matrix C_i ($i = 1, 2$) as follows:

$$S \left\{ \begin{array}{c|c} \overbrace{\hspace{1cm}}^S & \\ \hline A_i & B_i \\ \hline B'_i & D_i \end{array} \right.$$

Here, A_i corresponds to the edges between nodes that exist in both networks, B_i and B'_i correspond to edges between one node in S and one node outside S , and D_i corresponds to edges that are between nodes outside S . Since the mapping of nodes in S are fixed, the topological consistency of A_1 and A_2 is not informative on the mapping of the nodes in $V_1 - S$ vs. $V_2 - S$. Thus we consider the topological consistency of B_1 vs B_2 , B'_1 vs B'_2 , and D_1 vs D_2 to generalize Gromov-Wasserstein distance for this scenario:

$$L_1 = \sum_{\substack{i, k \in V_1 - S \\ j, l \in V_2 - S}} \frac{1}{2} (D_1(i, k) - D_2(j, l))^2 T(i, j) T(k, l) \tag{3}$$

$$L_2 = \sum_{\substack{i \in S \\ k \in V_1 - S \\ l \in V_2 - S}} \frac{1}{2} (B_1(i, k) - B_2(i, l))^2 T(k, l)^2 \tag{4}$$

Here, L_1 is the same as the GW distance between D_1 and D_2 . L_2 considers each common node $i \in S$, and penalizes the inconsistencies in the neighborhood of i created by the mapping of other nodes in the networks. We define the generalized Gromov-Wasserstein distance as the weighted sum of these two loss functions:

$$L_{generalized} = \min((1 - \alpha)L_1 + \alpha L_2) \tag{5}$$

Here, $0 \leq \alpha \leq 1$ is a parameter that balances the relative importance of prior information (edges with one side fixed) vs. free mappings (edges with both sides to be mapped). Increasing α assigns more weight to L_2 , so that the learning algorithm depends more on the known matchings instead of other nodes. $\alpha = 0$ corresponds to standard GW distance between D_1 and D_2 (ignoring the parts of G_1 and G_2 that are induced by the seeds), while $\alpha = 1$ corresponds to taking into account the edges incident to seeds only.

Peyré et al. [10] propose an efficient learning algorithm for computing the GW distance by incorporating a 4-way tensor \mathcal{L} and a tensor matrix multiplication $\mathcal{L} \otimes T$. The loss function of Gromov-Wasserstein distance can be rewritten as:

$$GW(C_1, C_2, T) = \langle \mathcal{L}(C_1, C_2) \otimes T, T \rangle \tag{6}$$

in which $\mathcal{L}(C_1, C_2) \otimes T$ is the cost matrix C in the optimal transport. A decomposition of $\mathcal{L}(C_1, C_2) \otimes T$ is also proposed to improve efficiency.

The optimal transport can be computed by solving a quadratic optimization problem [13]. Building on this approach, we propose an efficient learning algorithm for computing the Generalized GW distance by generalizing the quadratic problem to fit our objective function. For this purpose, we first define a $|V_1 - S| \times |V_2 - S|$ matrix:

$$E(k, l) = \sum_{i \in S} (B_1(i, k) - B_2(i, l))^2 \tag{7}$$

which is a constant matrix and can be computed by matrix operations. Then the loss function of the generalized Gromov-Wasserstein becomes:

$$L_{generalized} = \langle (1 - \alpha)\mathcal{L}(D_1, D_2) \otimes T + \alpha E \odot T, T \rangle \tag{8}$$

where $E \odot T$ is the element-wise multiplication of E and T , and $(1 - \alpha)\mathcal{L}(D_1, D_2) \otimes T + \alpha E \odot T$ is the cost matrix C .

We use Algorithm 1 to compute the Generalized GW distance of two networks. We first initialize the optimal transport T as the outer product of p and q (defined in Sect. 2.1). At each iteration, we compute the gradient direction of T and use Algorithm 2 to compute the optimal learning rate τ to minimize the cost of $T + \tau \Delta T$:

$$\tau = \arg \min_{0 \leq \tau \leq 1} L_{generalized}(T + \tau \Delta T) \tag{9}$$

The update function L_1 is derived in [13] as a quadratic function of τ . Using E as defined above, we derive the following update function for L_2 :

$$L_2(B_1, B_2, T + \tau \Delta T) = \sum_{\substack{k \in V_1 - S \\ l \in V_2 - S}} E(k, l) (\Delta T(k, l) \tau^2 + 2T(k, l) \Delta T(k, l) \tau + T(k, l)^2) \tag{10}$$

Thus the update function for $L_{generalized}$ can also be expressed as a quadratic function of τ . We then compute the optimal learning rate τ as the value that minimizes the resulting update function for $L_{generalized}$ and update T accordingly. When $\tau = 0$ or ΔT is less than a threshold, the process converges and stops. The complexity of the algorithm is $O(mn^2 + m^2n)$, where m and n are the number of nodes in $V_1 - S$ and $V_2 - S$.

Algorithm 1. Optimization for GGWNA

- 1: $T^{(0)} \leftarrow pq^T$
- 2: **for** $i = 1, 2, \dots$ **do**
- 3: $C \leftarrow$ cost matrix of the iteration
- 4: $T \leftarrow OT(C, T^{(i-1)})$
- 5: $\Delta T \leftarrow T - T^{(i-1)}$
- 6: $\tau^{(i)} \leftarrow$ line search using algorithm 2
- 7: $T^{(i)} \leftarrow T^{(i-1)} + \tau^{(i)} \Delta T$
- 8: **end for**

Algorithm 2. Line Search

- 1: $a \leftarrow -2(1 - \alpha) \langle D_1 \Delta T D_2, \Delta T \rangle + \alpha \langle E \odot T^{(i-1)}, T^{(i-1)} \rangle$
- 2: $b \leftarrow (1 - \alpha) \langle c_{D_1, D_2}, \Delta T \rangle - 2(1 - \alpha) (\langle D_1 \Delta T D_2, T^{(i-1)} \rangle + \langle D_1 T^{(i-1)} D_2, \Delta T \rangle) + 2\alpha \langle E \odot T^{(i-1)}, \Delta T \rangle$
- 3:
- 4: $c \leftarrow L_{gw}(T)$
- 5: **if** $a > 0$ **then**
- 6: $\tau \leftarrow \min(1, \max(0, \frac{-b}{2a}))$
- 7: **else**
- 8: $\tau \leftarrow 1$ if $a + b \leq 0$ else $\tau \leftarrow 0$
- 9: **end if**

Algorithm 3. Greedy Matching

Require: optimal transport $T \in \mathbb{R}^{p \times q}$

Ensure: array $M \in \mathbb{R}^{\min(p, q) \times 2}$ of matchings

- 1: $M \leftarrow \emptyset$
- 2: **while** size of $M < \min(p, q)$ **do**
- 3: $i, j \leftarrow$ the row and column indices of $\max(T)$
- 4: **if** $i \notin M(1)$ and $j \notin M(2)$ **then**
- 5: add node pair $[i, j]$ to M
- 6: **end if**
- 7: **end while**

3.2 Seeded Network Alignment Using Optimal Transport

Having computed the optimal transport matrix T , we aim to find an optimal matching between the two networks. For a pair of nodes $i \in G_1$ and $j \in G_2$, T_{ij} is assigned a larger value by the optimal transport algorithm if the local topology around them are more similar (also considering their edges with the nodes in S). While there are many algorithms in the literature to compute a discrete mapping of the nodes based on the weights in T [1], these algorithms are computationally costly. Here, since our focus is on computing T (as opposed to using T to compute a mapping), we use a simple greedy algorithm (Algorithm 3) to compute a mapping T , thereby enabling repeated computational experiments to compare the proposed algorithm against alternative algorithms. The framework we propose here can be used with any matching algorithm once T is computed using Algorithm 1. In each iteration of this algorithm, we find the row and column indices of the maximum value in T , and align the corresponding pair of nodes. If one of the nodes is already aligned, we skip the pair and find the next maximum value in T , until $\min(|V_1 - S|, |V_2 - S|)$ nodes are aligned.

Table 1. The networks used in the experiments. Left: Real network pairs. Right: Networks used to create network pairs in simulation studies.

Network Pairs	#Nodes	#Edges	#Matchings
Douban Offline	1118	1511	
Douban Online	3906	8164	1118
ACM	9872	39561	
DBLP	9916	44808	6325
Twitter	5120	130575	
Foursquare	5313	54233	1609
Phone	1000	41191	
Email	1003	4627	1000

Networks	#Nodes	#Edges
Facebook	4039	88234
lastfm	7624	27806
Arxiv	5242	14496

4 Experimental Results

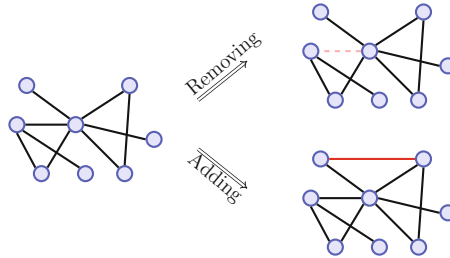
4.1 Datasets

We use real-world social network pairs to compare GGWNA with other network alignment algorithms. The network pairs [18] used in our experiments are shown in Table 1. Douban is an online social network providing user review and recommendation services for movies, books, and music. ACM and DBLP are two co-authorship networks, in which nodes indicate authors and edges indicate that the two authors published at least one paper together. The twitter-foursquare data includes friend relationships from two online social networks, and the overlaps are the people who are in both networks. In the Phone-Email dataset [19], the Phone and Email networks respectively correspond to communications among

people via phone and emails. For all datasets, the matchings are the users that are identified as the same person in different social networks.

Besides real-world network pairs, we perform simulation studies on real-world network [7] to generate synthetic network pairs with controlled characteristics. We assess the effect of the following variables in simulation studies (Fig. 2):

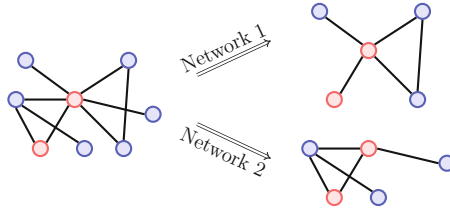
- **Network pairs with different levels of divergence:** For each network $G = (V, E)$ in Table 1, we generate 10 networks by adding or removing $\gamma|E|$ edges from G , where γ represents divergence (also referred to as noise, varying from 0.05 to 0.8). In the experiments, we align the 10 new networks with the original network G and assess the mean and variance of accuracy.
- **Divergent network pairs with identical degree distribution:** For each network $G = (V, E)$ in Table 1, we generate 10 divergent networks with γ : 0.05, 0.1, 0.2, 0.4, 0.8. To preserve degree distribution, we randomly remove two randomly selected edges $(i, j), (k, l) \in E$, and add edges (i, k) and (j, l) at each iteration of the randomization process (repeated $\gamma|E|/2$ times).
- **Network pairs with different levels of node overlap:** We simulate the case when two partial observations of a network are aligned. We generate 10



(a) Adding noise by adding/removing randomly selected edges. Left: Original network. Upper right: Dashed edge is removed. Lower right: Red edge is added to the network.



(b) Adding noise by swapping nodes in two randomly selected edges. Left: Original network. Right: Network after one edge swap.



(c) Splitting a network into two networks with fixed overlap. The red nodes selected from the original graph are the overlapping nodes of the new graphs.

Fig. 2. Simulation techniques used to generate synthetic network pairs.

network pairs with different levels of node overlap: 0.1, 0.2, 0.4, 0.8. For a network $G = (V, E)$ in Table 1(Right), we split it into two networks, where $\lambda|V|$ (λ denotes the overlap parameter) nodes appear in both networks, and other nodes are equally distributed in the two networks. If there is an edge $(i, j) \in E$, then edge (i, j) also appears in the new networks. After the pair is constructed, we add 20% noise to both networks as described above.

4.2 Baseline Methods

GW: The Gromov-Wasserstein distance was introduced in Sect. 2.2. We learn the optimal transport matrix using all nodes (including seed nodes) to apply the greedy matching algorithm, ignoring the seed matching.

FINAL. [18] is a supervised network alignment method for attributed networks. The FINAL algorithm leverages the node/edge attribute information to guide topology-based alignment process. In our experiments, the networks are not attributed networks, so the node attribute matrices are empty, and only topological consistencies are considered. We use the default hyperparameters of FINAL.

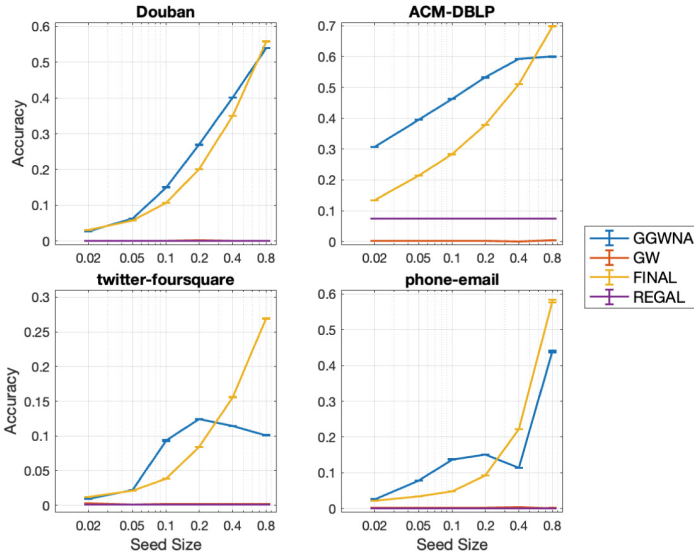


Fig. 3. Comparison of network alignment algorithms on real-world network pairs. The x-axis shows the percent of overlap that are used to train the models. The curves and error bars show the means and variances across 10 runs. Here, $\alpha = 0.8$ for GGWNA (the effect of this parameter is shown in Fig. 3).

REGAL. [5] first learns a node embedding for each network by a proposed matrix factorization technique (xNetMF). Then, the embeddings are used to compute the cross-network node similarities of each pair of nodes.

4.3 Experimental Setup

We compare our method with the baseline methods in terms of the accuracy of network alignment. Let $S' = V_1 \cap V_2$ denote the set of all known matching in the two networks. For a given “seed size σ (fraction of known matchings in the training set), we randomly select $\sigma|S'|$ nodes from S' to construct S . The remaining nodes in $S' - S$ become the test. For all algorithms S is provided as the set of seed matching and the resulting mapping of the node pairs in $S' - S$ is obtained by using the greedy matching algorithm (Algorithm 3) on the weighted mapping matrix returned by the algorithm. The accuracy of alignment is then computed as the fraction of correctly aligned pairs in $S' - S$. All the experiments are repeated 10 times, and the averages and variances are shown in the figures. The x-axes of all figures are in log scale.

4.4 Results on Real Network Pairs

The network alignment accuracy of the four algorithms on four different pairs of real network pairs as a function of seed size is shown in Fig. 4. On all datasets, GGWNA and FINAL clearly outperform GW and REGAL. In addition, for all datasets, GGWNA outperforms FINAL for smaller seed sizes, while FINAL outperforms GGWNA when the seed size is large. These results suggest that GGWNA is quite robust to smaller seed size.

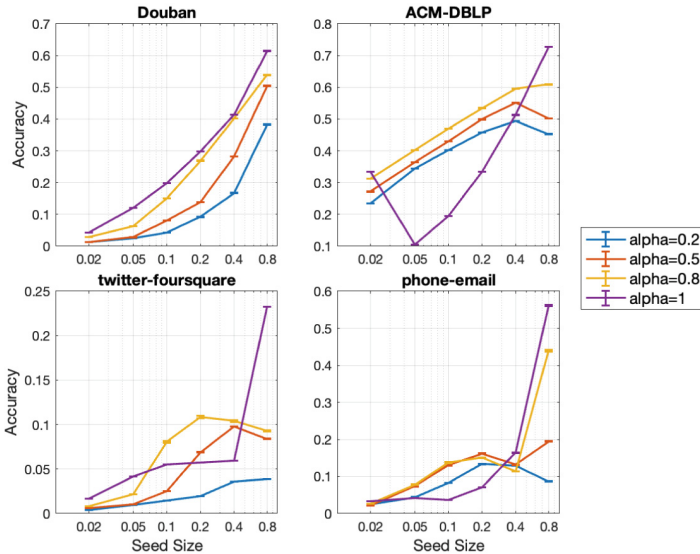


Fig. 4. The effect of α on the performance of GGWNA. The x-axis shows percent overlap used to train the models and the error bars show the variances.

In most cases, the accuracy of GGWNA increase as the seed size becomes larger, but the performance of GGWNA begins to decline when the seed size is too large (40% or 80%). The reason might be: As the seed size becomes larger, the D_i part of the adjacency matrices (Sect. 3.1) is smaller and will have less weight in the loss function. Therefore, the learning process depends more on the topological features of the seed nodes and less on the topological similarities of the nodes in the test set. FINAL performs better than GGWNA when more seed nodes are used in training, but 80% seed sizes can be unrealistic in practice. GW and REGAL do not work well on these datasets, and since they are unsupervised, the accuracy does not increase as the seed size becomes larger.

Figure 3 shows the effect of α on the performance of GGWNA. Overall, the accuracies of GGWNA increases as α increases from 0.2 to 0.8, but the performance goes down in most cases as we increase α to 1, since we depend too much on the known matchings instead of the topology. As α goes higher, the weight of the known matchings increases, and the optimal transport will depend more on B_1 and B_2 parts of the adjacency matrices.

4.5 Results on Simulated Pairs of Networks

We investigate the effect of various parameters using simulated network pairs (Fig. 2). We show the results on networks generated using the Facebook dataset here, the results of other datasets are in the Supplementary Material.

The Effect of Divergence/Noise. As seen in Fig. 5, the accuracy of the algorithms declines as the two networks diverge. GGWNA is most robust against

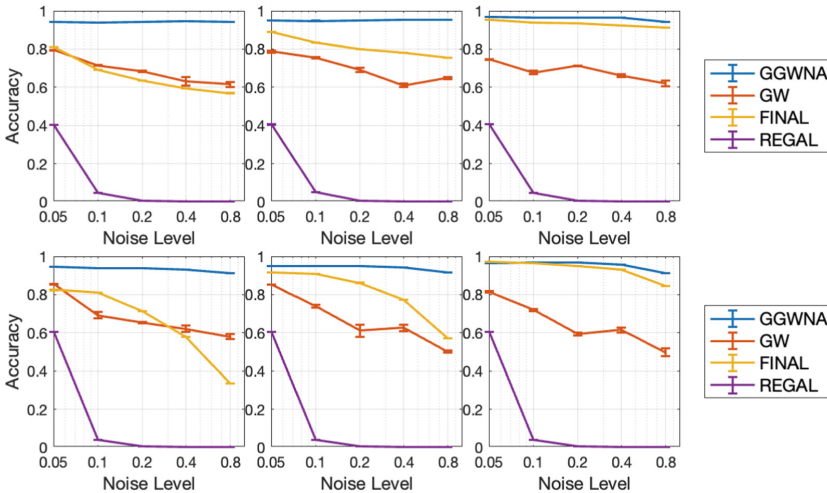


Fig. 5. Accuracy of network alignment accuracy as a function of noise/divergence between networks. Top: Uniform noise, Bottom: Degree-preserving noise. The seed sizes are 10% (left), 20% (center), and 50% (right).

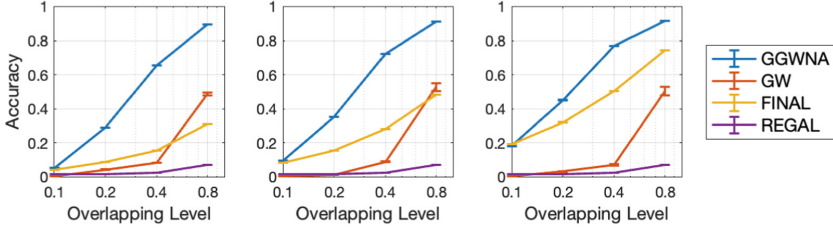


Fig. 6. Network alignment accuracy on partial observations of a network. Two networks are samples from the original network, with overlap levels from 10% to 80% as shown on the x-axis. 20% random noise is added to each network. Seed size: 10% (left), 20% (center), 50%(right) of the overlapping nodes.

noise, the accuracy decreases slightly as the noise level increases. The accuracy of FINAL improves as the seed size increases from 0.05 to 0.8, but GGWNA remains at a higher accuracy even when the seed size is small. Accuracy declines more sharply for degree-preserving noise (bottom panel), since this presents a more difficult instance for the algorithms (i.e., the algorithms cannot use node degree information to match the nodes), which can be more relevant in practice.

Partial Observations of a Network. From the results on real network pairs (Fig. 4), we observe that GGWNA works better than other techniques when the node overlap between nodes the networks is larger (e.g. the ACM-DBLP and phone-email datasets). In the experiments reported in Fig. 6, we investigate the effect of node overlap between two observations of a single network. As seen in the figure, the accuracy of all algorithms improves as the node overlap becomes larger, especially for GGWNA. However, GGWNA is still robust to smaller seed sizes as there is no obvious differences between the curves of the three subplots.

5 Conclusions

In this paper, we proposed generalized Gromov-Wasserstein for network alignment (GGWNA), by introducing a new loss function that takes into account the connectives of seed nodes for which matchings are known. We compared the accuracy of the algorithm on real network pairs as well as simulated pairs, and showed that our generalized GW outperforms other network alignment methods at most time, and it is robust to high divergence between networks and smaller seed sizes. Avenues for future research include introducing labels into the loss function, and applying generalized GW to a broader range of types of networks.

References

1. Caetano, T.S., McAuley, J.J., Cheng, L., Le, Q.V., Smola, A.J.: Learning graph matching. *IEEE TPAMI* **31**(6), 1048–1058 (2009)
2. Chowdhury, S., Needham, T.: Generalized spectral clustering via gromov-wasserstein learning. In: *ICAIS*, pp. 712–720 (2021)
3. Flamary, R.: Pot: python optimal transport. *JMLR* **22**(78), 1–8 (2021)
4. Halimi, A., Ayday, E.: Profile matching across online social networks. In: Meng, W., Gollmann, D., Jensen, C.D., Zhou, J. (eds.) *ICICS 2020*. LNCS, vol. 12282, pp. 54–70. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-61078-4_4
5. Heimann, M., Shen, H., Safavi, T., Koutra, D.: Regal: representation learning-based graph alignment. In: *ACM CIKM*, pp. 117–126 (2018)
6. Kazemi, E.: Proper: global protein interaction network alignment through percolation matching. *BMC Bioinformatics* **17**(1), 1–16 (2016)
7. Leskovec, J.: Snap datasets: Stanford large network dataset collection (2014)
8. Ma, L.: Heuristics and metaheuristics for biological network alignment: a review. *Neurocomputing* **491**, 426–441 (2022)
9. Mémoli, F.: Gromov-wasserstein distances and the metric approach to object matching. *Found. Comput. Math.* **11**(4), 417–487 (2011)
10. Peyré, G., Cuturi, M., Solomon, J.: Gromov-wasserstein averaging of kernel and distance matrices. In: *ICML*, pp. 2664–2672 (2016)
11. Santambrogio, F.: *Optimal transport for applied mathematicians*. Birkhäuser, NY **55**(58–63), 94 (2015)
12. Shu, K., Wang, S., Tang, J., Zafarani, R., Liu, H.: User identity linkage across online social networks: a review. *ACM SIGKDD Expl.* **18**(2), 5–17 (2017)
13. Titouan, V., Courty, N., Tavenard, R., Flamary, R.: Optimal transport for structured data with application on graphs. In: *ICML*, pp. 6275–6284 (2019)
14. Trung, H.T.: A comparative study on network alignment techniques. *Expert Syst. Appl.* **140**, 112,883 (2020)
15. Vayer, T., Flamary, R., Tavenard, R., Chapel, L., Courty, N.: Sliced gromov-wasserstein. *arXiv preprint arXiv:1905.10124* (2019)
16. Villani, C.: *Topics in optimal transportation*, vol. 58. AMS (2021)
17. Xu, H., Luo, D., Zha, H., Duke, L.C.: Gromov-wasserstein learning for graph matching and node embedding. In: *ICML*, pp. 6932–6941 (2019)
18. Zhang, S., Tong, H.: Final: Fast attributed network alignment. In: *ACM SIGKDD*, pp. 1345–1354 (2016)
19. Zhang, S., Tong, H., Jin, L., Xia, Y., Guo, Y.: Balancing consistency and disparity in network alignment. In: *ACM SIGKDD*, pp. 2212–2222 (2021)



Orderliness of Navigation Patterns in Hyperbolic Complex Networks

Dániel Ficzere^(✉), Gergely Hollósi, Attila Frankó, Pál Varga, and József Biró

Department of Telecommunications and Media Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, H-1111 Műegyetem rkp. 2, Budapest, Hungary
ficzere@tmit.bme.hu

Abstract. Navigation plays a pivotal role in the operation of real-world complex networks. In this paper, we delve into the extensive realm of the 'store and forward' principle, comprising two fundamental components: the addressing scheme for network nodes and the routing function responsible for establishing paths between network endpoints. Particularly, we show that the hyperbolic geometry of complex networks can be used to greatly improve the orderliness of navigation patterns in complex networks implementing the store and forward principle. By using entropy-based measures applied on the forwarding tables we provide a formal assessment for the orderliness which can also be used to estimate the memory requirements of navigation directly at individual nodes and in the whole network.

Keywords: hyperbolic complex networks · navigation · routing · entropy

1 Introduction

When compiling a roster of prevalent network functions, information routing invariably occupies a prominent position. Consequently, it's unsurprising that a multitude of networks have demonstrated navigability, enabling nodes to adeptly channel information throughout the network, even in cases where the overall structure remains undisclosed to individual nodes.

Among technological networks, the Internet stands out as a fundamental case, primarily designed to route information among computers. In the present day, the majority of computer networks are constructed upon the distributed hop-by-hop routing paradigm. Within this framework, routers uphold forwarding tables that correlate incoming packets with succeeding hop routers, relying on the destination address embedded within the packet headers. Subsequent routers adhere to the same mechanism, progressively transporting packets hop-by-hop toward their desired destinations. Consequently, routers are required to retain a sufficient amount of information within their internal memory, enabling them to accurately forward any packet – regardless of its destination address – towards the appropriate next-hop router [4, 9].

The utility of hyperbolic geometry has been notably impactful in the realms of network analysis and synthesis. In a seminal work [6] authors established a geometric framework for investigating the intricate structure and functionalities of complex networks. Expanding upon this hyperbolic geometric framework, [2] delved into the realm of large-scale Internet routing, and this exploration was further extended within Internet-like architectures as demonstrated in [10]. In addition, hyperbolic geometry has found successful applications within networked neuroscience. This fact is underscored in [1], which affirms that hyperbolic space offers a nearly impeccable method for charting navigable maps of connectomes across various species. This finding reveals that hyperbolic distances harmoniously align with the intricate structures inherent in brain networks.

In this paper we deal with the analysis of orderliness of forwarding patterns in hyperbolic complex networks. Assuming the store and forward (hop by hop) routing principle we follow the information-theoretic approach introduced in [5] to measure the orderliness of forwarding patterns. More specifically, the first order empirical entropy of forwarding tables as sequential strings of node addresses is measured and analyzed. We introduce a heuristic approach utilizing hyperbolic coordinates to create an address space that holds the potential to notably decrease the first-order entropy of forwarding strings at the network level. We have made numerical comparisons on several synthetic networks and a real-world network, and found that our method significantly increase the orderliness compared to the random choice of addresses and slightly better than other heuristic based on hierarchical clustering of nodes. We think that our results form an important step towards disclosing the intricate relationship between network structural dynamics, network geometry and address space optimization of hop by hop navigation.

2 Related Works

2.1 Hyperbolic Geometry of Complex Networks

In the original model of hyperbolic complex networks, N points are distributed (quasi-)uniformly across a two-dimensional hyperbolic disk with a radius of R [6]. Two points are connected if the distance between them does not exceed R . These points symbolize the network nodes, and the connections they form constitute the links within the resulting networks. By employing polar coordinates for a pair of points (u, v) , denoted as (r_u, ϕ_u) and (r_v, ϕ_v) , this elegantly straightforward generation rule can be formally defined as follows: connect points u and v if their hyperbolic distance satisfies $d(u, v) \leq R$. The hyperbolic distance $d(u, v)$ between u and v can be expressed using the hyperbolic cosine law:

$$\cosh(d(u, v)) = \cosh(r_u) \cosh(r_v) - \sinh(r_u) \sinh(r_v) \cos(\phi_u - \phi_v) \quad (1)$$

which can be used to calculate efficiently hyperbolic distances and angles.

The hyperbolically (quasi-)uniform node density implies that we assign the angular coordinates $\phi \in [0, 2\pi]$ to nodes with the uniform density $\rho(\phi) = \frac{1}{2\pi}$, while the density for the radial coordinate is exponential as

$$\rho(r, \alpha) := \alpha \frac{\sinh \alpha r}{\cosh \alpha R - 1} \quad (2)$$

where $0.5 \leq \alpha \leq 1$. When $\alpha = 1$ then the distribution of the nodes are uniform, otherwise the density is changing from the centre to the periphery with constant rate.

In our illustrations (see Fig. 1), we utilize the native representation of a hyperbolic plane, employing hyperbolic coordinates as if they were Euclidean. While this choice might lead to some peculiar visual effects, it enhances the comprehensibility of algorithmic descriptions and examples within the text. For example, such a peculiar phenomenon in our representations that nodes appear non-uniformly distributed across the disk. This apparent distortion originates from the non-isometric nature of the native representation. It's important to note that other representation models within the Euclidean plane lack isometry as well, mainly due to the fact that the hyperbolic plane inherently contains a significantly larger volume-exponentially so-than its Euclidean counterpart. Nevertheless, our calculations and derivations remain independent of any specific representation models and properties.

2.2 Modeling Forwarding Tables

The store and forward routing principle entails that packets carry essential global information about their destination node (identifier/address). At each intermediate node, the routing function identifies the neighboring node (next-hop) to which the packet should be forwarded on its journey toward its destination. In packet-oriented communication networks like on the Internet the global identifiers of the nodes are the IP addresses, in smaller administrative domains the routing function is usually providing the shortest path, and the forwarding decision is performed based on a forwarding table (often referred to as forwarding information base, FIB).

Definition 1 (Routing table). *Let $G(V, E)$ be a connected, undirected graph with N nodes. The routing table or routing function for a node $v \in V$ is the function $r_v : V \rightarrow N_G(v) \cup v$, where $N_G(v)$ stands for the neighbourhood of the node v .*

Remark 1. The r_v routing functions can be determined using different routing strategies (i.e. shortest path), however, in the paper we suppose, that the routing functions $\{r_v\}_{v \in V}$ are readily available.

Remark 2. Please note, that the value of $r_v(v)$ can be chosen freely. In the paper we suggest that $r_v(v) = v$.

For the purpose of modeling, we assume a flat address space over the nodes of the graph, and we assign unique continuous integer identifiers to nodes from the set $[1, 2, \dots, N]$, where N is the number of nodes in the graph.

Definition 2 (Node identifiers). *Let $\Sigma = [1, 2, \dots, N]$ be a finite set, with size $|\Sigma| = N$. The set Σ is called the alphabet or node identifiers.*

Definition 3 (Ordering). Let $G(V, E)$ be a connected, undirected graph with N nodes and let Σ be the alphabet. Also, let $p : V \rightarrow \Sigma$ be a bijection. The bijection p is called the ordering or permutation of the nodes.

Definition 4 (Routing string). Let $G(V, E)$ be a connected, undirected graph with N nodes, let Σ be the alphabet ($|\Sigma| = N$), let p be a node ordering and r_v is the routing function of node $v \in V$. The routing string or forwarding string for node v is the finite sequence defined by the function $f_v : \Sigma \rightarrow \Sigma$ and $f_v : p(\xi) \mapsto p(r_v(\xi))$ for every node $\xi \in V$.

Remark 3. Please note, that indeed, f_v defines a finite sequence, since it maps every node identifiers $[1, 2, \dots, N]$ to another node identifier (but not necessarily surjectively).

2.3 Measures to Orderliness

Definition 3 defines ordering as a bijective function p . However, to measure the orderliness of the routing table, we define different measures based on *routing strings*. These definitions only aim to measure the orderliness of the routing table of one node only – however, in the results we will show different statistics for orderliness of multiple nodes to characterize the whole graph.

Definition 5 (Sprint). Let $G(V, E)$ be a connected, undirected graph and Σ is the alphabet, and let $s_v = (a_1, a_2, \dots, a_N \mid a_i \in \Sigma)$ be a routing string for a node $v \in V$ and for some p ordering. A sprint of length $K > 0$ ($K \in \mathbb{N}$) at k is a sequence, denoted as S_k , where $|S_k| = K$, $a_k \neq a_{k-1}$ (if a_{k-1} exists) and $a_{k+K-1} \neq a_{k+K}$ (if a_{k+K} exists) and $a_i = a_{i+1}$ for every $i \in [k, k + 1, \dots, k + K - 2]$.

Remark 4. Note, that a sprint can be located at the beginning and at the end of a string (in this case a_{k-1} or a_{k+K} not exists).

Remark 5. Note also, that not every k can be paired to a sprint, since a_k might not equal to a_{k-1} . E.g. in the sequence $(1, 2, 2, 2, 3, 5, 5)$, sprints can be found at $k = 1, 2, 5, 6$, with $|S_1| = 1$, $|S_2| = 3$, $|S_5| = 1$ and $|S_6| = 2$.

Definition 6 (Longest sprint). The quantity $\max\{|S_k| \mid k \in \Sigma \text{ and } S_k \text{ exists}\}$ is called the longest sprint.

Depending on the amount of information used on the routing strings, entropy-based lower bounds have been introduced for hop-by-hop routing in [5]. These bounds can be used to estimate the memory requirements of encoding the forwarding strings.

Definition 7 (Zero-order empirical entropy). Let $G(V, E)$ be a connected, undirected graph and let s_v be a routing string for a node $v \in V$. The zero-order entropy for node $v \in V$ is defined as

$$H_0(s_v) = \sum_{i \in \Sigma} \frac{n_i}{|s_v|} \log \frac{|s_v|}{n_i} \tag{3}$$

where n_i is the number of times i appears in s_v .

Remark 6. In the paper, $0 \log 0$ and $0 \log \infty$ are supposed to be 0.

Remark 7. Please note, that the zero-order entropy is invariant under the ordering of the nodes, e.g. it does not change after some permutation of the node identifiers.

Higher-order entropies extend the concept further by encompassing not only the frequencies, but also the intricate sequences in which these IDs appear [3]. Higher-order empirical entropy is strongly related to the entropy of Markov-chains, however, it measures the entropy of a very specific sequence. In the paper, we use the first-order empirical entropy (as in [5]), which requires first to define the context string.

Definition 8 (Context string). Let $s = (a_1, a_2, \dots, a_N)$ be a routing string with an alphabet Σ ($|\Sigma| = N$). For a $t \in \Sigma$ context, the context string is the sequence $C_t = (a_i \mid a_{i-1} = t, 1 < i \leq N)$.

Definition 9 (First-order empirical entropy). Let $G(V, E)$ be a connected, undirected graph and let s_v be a routing string for a node $v \in V$ and for some p ordering. The first-order empirical entropy is defined as

$$H_1(v) = \frac{1}{|s_v|} \sum_{t \in \Sigma} |C_t| \cdot H_0(C_t) \tag{4}$$

where C_t is the context string of s_v for context $t \in \Sigma$, and H_0 is the zero-order entropy.

3 Data Sets

In our work, we evaluate our hypotheses on a synthetic network and on a real world example. Both networks have hyperbolic representations defined as follows.

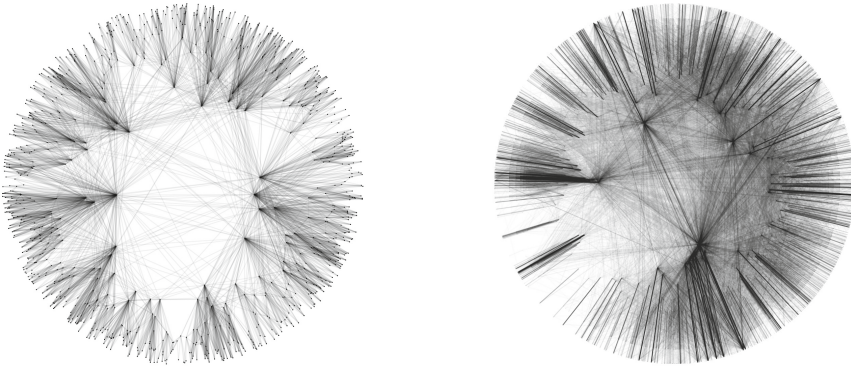
Definition 10 (Hyperbolic representation). Let $G(V, E)$ be a connected, undirected graph. Hyperbolic embedding is a function $h : V \rightarrow \mathbb{R}^2$, which assigns hyperbolic (polar) coordinates to every $v \in V$ and satisfy Eq. (1).

3.1 Synthetic Network Generation

To generate the synthetic network we used the following parameters: $N = 1000$, $R = 10$ and $\alpha = 1$. The resulting network, generated according to the rule outlined earlier, is depicted in Fig. 1a. The initial identifiers of nodes $1, 2, \dots, N$ align with the sequence of nodes' coordinates generated randomly. This string's structure, depicted in Fig. 3 (a), exhibits a noticeable horizontal orientation. The identifiers are interleaved randomly, and there are no longer continuous sequences of identical IDs in the forwarding string. The longest sequence of a single ID within this string is merely 4, with most runs consisting of just one. On average, the length of these sequences is 1.067.

3.2 Internet AS-Level Topology - A Real World Example

The Internet data set representing the global internet structure at the autonomous system (AS) level is from [2]. The topology contains 23748 nodes and 58414 connections. The average degree of a node is 4.92, however, the degrees are distributed in a wide range according to a scale free distribution. The network layout presented in Fig. 1b. The hyperbolic embedding of ASs are from [8] using the HyperMap algorithm. This algorithm is deterministic and is based on the previous observation that the latent geometry of scale-free and strongly clustered real networks is hyperbolic. Originally, in the dataset the nodes are ordered according to their level in the hierarchy and their degrees are also taken into account. This means that nodes with low values of IDs are strongly connected to each other forming the so-called core of high-rank ASs in the network. The core is also connected to the periphery of the network formed by lower level autonomous systems.



(a) A synthetic hyperbolic network example with $N = 1000$ and $R = 10$.

(b) The hyperbolic network embedding of the Internet AS network.

Fig. 1. The network layout of the examined graphs.

4 Methods

The main problem analyzed in the paper can be stated as follows: find a p ordering for a graph $G(V, E)$ with routing functions $\{r_v\}_{v \in V}$ to increase the orderliness of the routing strings. For orderliness, we use the measures defined in Subsect. 2.3.

4.1 Ordering IDs According to the Hyperbolic Angular Coordinates

In this subsection we present our new idea to increase the orderliness of forwarding strings of nodes in hyperbolic complex network. The idea is based on

generating a new address space by using the hyperbolic angle coordinates. As described earlier, in the hyperbolic generative model we assign the angular coordinates $\phi \in [0, 2\pi]$ to nodes with the uniform density $\rho(\phi) = \frac{1}{2\pi}$. There is no distinguished direction in the model, only the difference of the angle coordinates of nodes counts in the distance calculation.

Heuristic 1 (Angular ordering). *Let $G(V, E)$ be a connected, undirected graph, and $h : V \rightarrow \mathbb{R}^2$ the hyperbolic representation for the nodes $v \in V$. Choose p ordering such that $p(v) > p(w) \Leftrightarrow \phi_v > \phi_w$ for all $v, w \in V$, where ϕ_v is the angular coordinate in $h(v)$.*

Figure 3a and Fig. 3d presents the randomly organized FIB of the highest degree nodes. Similarly, Fig. 3b and Fig. 3e depict the newly organized FIB of the highest degree nodes. In this way, one can visually recognize that the highly interleaved horizontal clusters of points on Fig. 3a and Fig. 3d are collected into longer sequences of same IDs on Fig. 3b and Fig. 3e.

4.2 Ordering IDs Based on Hierarchical Clustering

For comparison, we use another heuristics for decreasing the first order entropy adopted from [5], the so called hierarchical clustering. Korosi et al. uses single-linkage clustering, however, their results seem to be based on complete-linkage clustering, since single-linkage would result in merging a cluster and a node periodically, which trivially turns into an astray clustering.

Definition 11 (Node cluster). *Let $G(V, E)$ be a graph. We call a set of nodes as node cluster, denoted as $C_i, i \in \mathbb{N}$, i.e. $C_i \subseteq V$.*

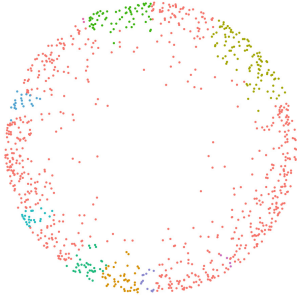
Algorithm 1 (Complete-linkage hierarchical clustering). *Let $G(V, E)$ be a connected, undirected graph, and $d(u, v)$ the length of the shortest path between $u \in V$ and $v \in V$. Let all nodes be in its own cluster, so the set of clusters is $C = \{\{v\}\}_{v \in V}$. In each step, clustering means the merging of two closest clusters, where the distance is measured between clusters is $d(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$.*

The clustering is ended, if $|C| = 1$. The result is a binary merging tree, called the dendrogram of clustering, where each node represents the merging of its two child clusters.

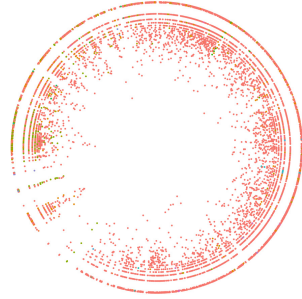
Heuristic 2 (Clustering based ordering). *Let $G(V, E)$ be a connected, undirected graph. Let D be a dendrogram of the complete-linkage hierarchical clustering of the graph G . Chose p ordering to be the permutation of nodes which is the result of post-order traversal of the dendrogram.*

Remark 8. Please note, since the dendrogram is not strictly an ordered tree, there can be many different permutations using this method.

The results of the hierarchical clustering for both networks (synthetic and AS Internet) are presented in Fig. 2. Furthermore, the FIB of the highest degree nodes are presented in Fig. 3c and Fig. 3f. It can be seen that the ordering increased compared to the randomly ordered nodes, but the relation to the angle-based approach cannot be judged; some kind of metric is needed for that.



(a) The hierarchical clustering of the synthetic network with 10 clusters.



(b) The hierarchical clustering of the Internet AS network.

Fig. 2. The hierarchical clustering results of the examined networks with 10 clusters.

5 Discussion

The statistics on H_1 and on the sprint-lengths clearly show that the angular-based addressing produces the highest orderliness. Figure 2 illustrates how the clustering method forms clusters with relatively low angular distances between nodes, essentially resembling sections of a disk. Therefore, both methods produce similar results in terms of H_1 and sprint-length statistics. Moreover, it is worth noting that the orderliness, as indicated by the statistics, is significantly greater when using angular-based address space, see Fig. 4, 5, 6. The numerical results for some notable metrics are summarized in Table 1. These results also support the hypothesis, that the angle-based ordering performs best. It's essential to emphasize that defining these statistics for every node in the network would have been valuable. However, this endeavor demands substantial computational resources, particularly in the case of the AS Internet network, where calculating the shortest paths for every node across the entire graph is necessary. Consequently, we opted to compute these features for a random sample of 100 nodes. Importantly, we verified that the results remained largely consistent irrespective of the sampling method.

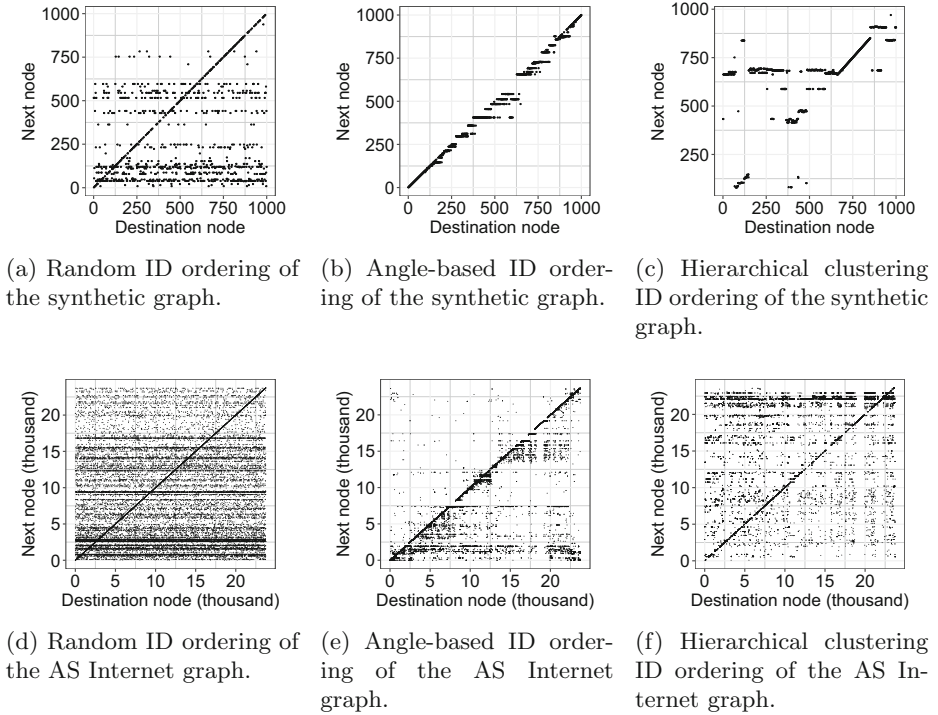
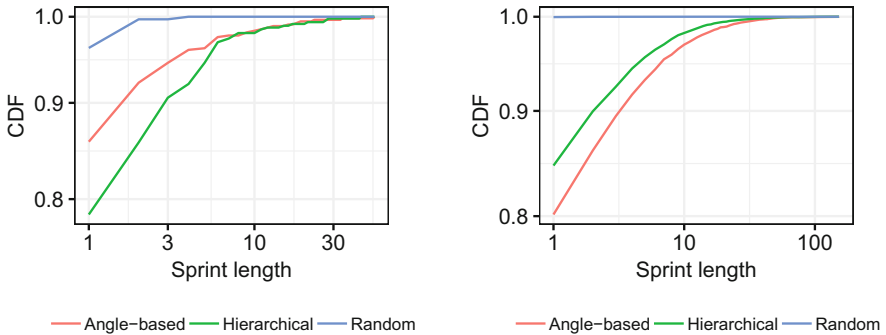


Fig. 3. The forwarding string representation of the highest degree node for the three examined methods.



(a) Sprint-length statistic of the synthetic network (b) Sprint-length statistic of the Internet AS network.

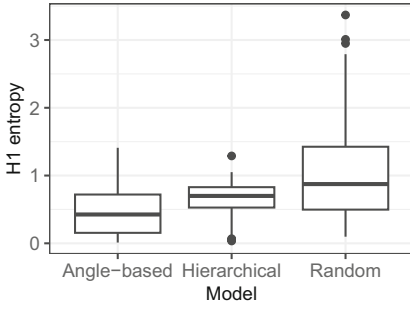
Fig. 4. The sequence length distribution of the highest degree node of the synthetic and the Internet AS network.

Table 1. Comparison of the statistics of 100 sampled nodes for both the synthetic and AS graph. The statistics of the 100 nodes are aggregated using averaging. (Random: random ordering, Angle: angle based ordering, HC: hierarchical clustering)

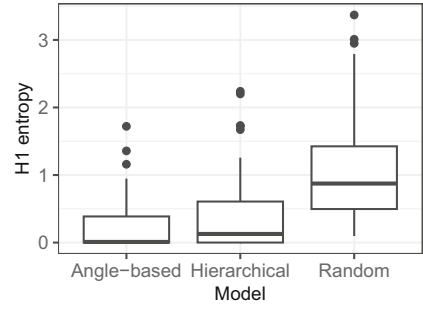
Network	Method	H_1		Sequence			
		mean	median	Average mean	Average median	Max mean	Max median
Synthetic	Random	0.96	0.91	11.35	6.01	55.84	26
	Angle	0.48	0.47	22.68	9.37	447.34	375
	HC	0.65	0.71	8.6	2.57	196.78	143
AS	Random	0.42	0.04	3579.63	617.25	7996.27	4207
	Angle	0.23	0.01	3750.90	1213.98	9872.99	11547
	HC	0.37	0.13	2709.13	16.70	6880.03	965

The intuition behind the use of hyperbolic angle coordinates as ordering rules is the following. On one hand it is known from previous studies that the short paths between the nodes are not very far from the geodesics of the nodes in the hyperbolic space [6]. The geodesics are the 'straight lines' and in this sense they are pointing from the source (or an intermediate node) to the destination. Hence, the initial step in a short path, the next hop should also be found more or less towards the destination, at least in a certain angle range containing also the destination node. Hence, it is very unlikely that the next hop from a node is seen in a very different direction than the direction in which the destination node is seen. On the other hand, the sequence in which nodes appear to an arbitrary 'observer' node on the hyperbolic plane while the observer revolves is significantly correlated to the order of node hyperbolic angle coordinates (which can be considered as the observer was in the center of the disk). According to our preliminary investigations, this correlation is very high when the observer node has higher radial coordinates lying between $R/2$ and R . Note that on the hyperbolic disk for reasonable N and R the radial coordinates of almost all points fall in the range $[R/2, R]$. Here, it is also worth noting that authors in [7] demonstrated the significance of angular differences between nodes as a similarity measure. This also strengthens our intuition.

For the time being the shortest path routing function is used in our analysis for orderliness of forwarding tables. Although the shortest path routing is omnipresent in communication networking and can be a reference in many other types of networks, in network science there is another widely studied navigation scheme, the so-called greedy routing or greedy navigation. The greedy forwarding scheme uses coordinates and distance calculations to decide the next hop for forwarding, and does not apply forwarding tables at all for saving memory space. Hence, greedy routing is suitable for complex networks embedded in a metric space like the hyperbolic plane. Here, we advocate that it will be worth analyzing greedy routing (we plan doing this) as if it had forwarding tables (which could be generated by the simple rule of greedy forwarding), because the entropies on (even hypothetical) forwarding tables are more general bounds, they are not

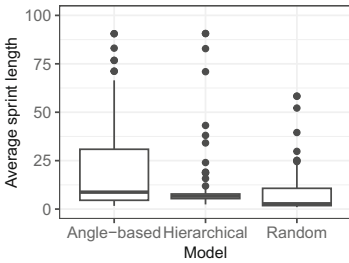


(a) H1 entropy statistic of the synthetic network

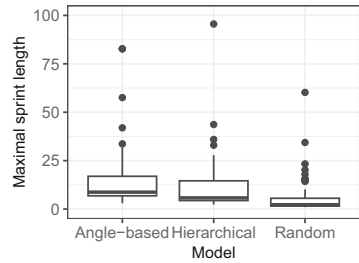


(b) H1 entropy statistic of the Internet AS network.

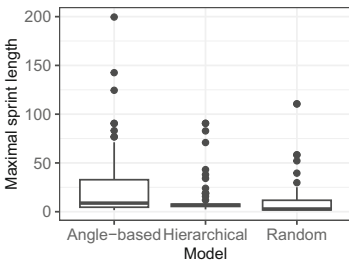
Fig. 5. H1 entropy statistic of 100 sampled nodes for the synthetic and the Internet AS network.



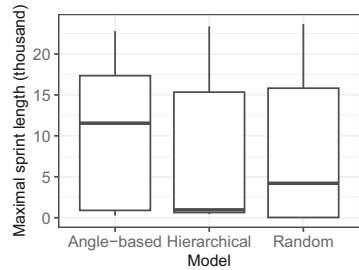
(a) Sprint mean statistic of the synthetic network



(b) Sprint mean statistic of the Internet AS network.



(c) Sprint max statistic of the synthetic network.



(d) Sprint max statistic of the Internet AS network.

Fig. 6. Sprint mean statistic of 100 sampled nodes for the synthetic and the Internet AS network.

only lower bounds for the memory requirements of tables but they may also forecast the overall complexity if such schemes which does not use forwarding tables at all.

6 Conclusion

Minimizing globally the first order entropy (thus maximizing the global orderliness) at network level is hard and seems to be hopelessly intractable. Other possibility is to use exhaustive search in the parameter space, but this is neither suitable even for networks with moderate size because of the exponentially growing number of permutations of node IDs. For very small networks and very special graphs the global optimization can be solved, however, these results have less substance in large scale real-world complex networks. Therefore, heuristic approaches have special significance in finding high level of orderliness in navigation patterns. Our approach presented is unique in a sense that we try to couple address space optimization with the hidden hyperbolic space of real-world complex networks. We feel that the hyperbolic geometry of complex networks may provide a rich set of possibilities for address space optimization heuristics. The ordering node IDs based on the hyperbolic angular coordinates is only the first and maybe the simplest way, other orderings are worth trying like those based on mutual hyperbolic distances between nodes or based on the hyperbolic trees.

References

1. Allard, A., Serrano, M.Á.: Navigable maps of structural brain networks across species. *PLoS Comput. Biol.* **16**(2), e1007584 (2020)
2. Boguná, M., Papadopoulos, F., Krioukov, D.: Sustaining the internet with hyperbolic mapping. *Nat. Commun.* **1**(1), 62 (2010)
3. Ferragina, P., Venturini, R.: A simple storage scheme for strings achieving entropy bounds. *Theoret. Comput. Sci.* **372**(1), 115–121 (2007)
4. Gulyás, A., Rétvári, G., Heszberger, Z., Agarwal, R.: On the scalability of routing with policies. *IEEE/ACM Trans. Networking* **23**(5), 1610–1618 (2014)
5. Kőrösi, A., Gulyás, A., Heszberger, Z., Bíró, J., Rétvári, G.: On the memory requirement of hop-by-hop routing: tight bounds and optimal address spaces. *IEEE/ACM Trans. Networking* **28**(3), 1353–1363 (2020)
6. Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., Boguná, M.: Hyperbolic geometry of complex networks. *Phys. Rev. E* **82**(3), 036106 (2010)
7. Papadopoulos, F., Kitsak, M., Serrano, M.Á., Boguná, M., Krioukov, D.: Popularity versus similarity in growing networks. *Nature* **489**(7417), 537–540 (2012)
8. Papadopoulos, F., Psomas, C., Krioukov, D.: Network mapping by replaying hyperbolic growth. *IEEE/ACM Trans. Networking* **23**(1), 198–211 (2014)
9. Rétvári, G., Tapolcai, J., Kőrösi, A., Majdán, A., Heszberger, Z.: Compressing IP forwarding tables: towards entropy bounds and beyond. In: *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, pp. 111–122 (2013)
10. Voitalov, I., Aldecoa, R., Wang, L., Krioukov, D.: Geohyperbolic routing and addressing schemes. *ACM SIGCOMM Comput. Commun. Rev.* **47**(3), 11–18 (2017)



Multiplex Financial Network Regionalization Scenarios as a Result of Re-globalization: Does Geographical Proximity Still Matter?

Otilija Jurakovaite^(✉) and Asta Gaigaliene

Vytautas Magnus University, Kaunas, Lithuania

{otilija.jurakovaite, asta.gaigaliene}@vdu.lt

Abstract. Re-globalization is a currently understudied topic and previous research focuses mostly on theoretical discussion of the problem. Empirical re-globalization related research suggests that re-globalization in terms of structural changes in financial network did not start recently, but was already observed after global financial crisis. It might have been further strengthened by pandemic and recent geopolitical tensions, but these tendencies have not been tested empirically. Among possible scenarios of re-globalization, most likely are discussed to be geographical regionalization or ally-based bipolar regionalization. Therefore, we aim to test these scenarios empirically. Using 5-layer multiplex financial network data of 2009–2020 from 234 countries, we found that multiplex financial network appears to be already highly regionalized, but regionalization and globalization appears to be not contradictory processes. Geographical regionalization did not increase in terms of shifting interregional investment to intraregional territory as interregional investment network density and value also increased as well as intraregional. The world appears to have become bipolarly ally-regionalized with 2 main communities - US & Europe vs. China. However, it is worth noting that Europe's role is still unclear as not all European countries belong to the same identified cluster. Future research could aim to explore in detail what are the main factors affecting ally-familiarity based region formation.

Keywords: Multiplex financial network · Re-globalization · Regionalization · Global financial network · Community analysis · Geographical proximity

1 Introduction

Re-globalization topic, which is currently dominating scientific and practical discussions, denotes deep change in structure of globalization with regard to counterparties, types of financial flows and their amounts. This topic became widely discussed after recent events of Covid-19 pandemic and following geopolitical tensions and conflicts in Europe, which highlighted drawbacks of being excessively interconnected with some particularly risky or unstable counterparties with regard to trade (supply chain) or finance (investments). These structural changes might be particular salient in financial sector where physical distance is much less important factor than in trade. Research suggests

that changes of countries' importance are happening what fosters new approach to globalization (Paul 2021; Scott and Wilkinson 2021). Multiplex financial network topology changed after global financial crisis (hereinafter - GFC) as number of strongly and weakly connected countries decreased and more countries became included in the network and same trends remain since (Lund et al. 2017; Lambert et al. 2015). Positions of separate countries in the network reveal tendency of decreased importance of developed countries (e.g., Europe) and increased importance of developing ones (e.g., especially Asia region) (Korniyenko et al. 2018), which also seems to be continuing after GFC, thus, proving this process longevity and supporting the notion of a new phase of globalization.

There is currently no unified belief towards where this new phase of globalization might lead. Several scenarios of re-globalization have been developed (Wray et al. 2022; Grosskurth et al. 2022) recently including mainly bipolar regionalization, geographical regionalization, localization and continued globalization. Both bipolar and regional segmentation scenarios imply regionalization, but at a different level. For instance, bipolar segmentation would treat US & Europe as one region while China and its' possible allies as another. Geographical regional segmentation might imply some segments based on geographical proximity. Regionalization is not a recent phenomenon and its increase was identified in post-GFC period research, especially in banking sector (Lund et al. 2017; Lambert et al. 2015; Gaigaliene et al. 2018). As re-globalization might have started already after GFC, regionalization observed during post-GFC period, could also characterize re-globalization. Thus, in this research we aim to test the regionalization scenarios of re-globalization empirically as suggested by re-globalization megatrend.

2 Literature

Re-globalization is a currently understudied topic, partly due to its recency, partly due to its concept still being formulated. Part of previous literature about re-globalization mostly focuses on theoretical discussion of the problem (Paul 2021; Scott and Wilkinson 2021). For example, Paul (2021) argues that the period of 'hyper-globalization' may need a change as increasing role of China and its allies pose some new challenges. Scott and Wilkinson (2021) suggest that world trade and financial flows have to re-globalize in order to avoid deglobalization.

There is a strand of empirical re-globalization research, which suggest that re-globalization in terms of structural financial network changes did not start recently, but appears to be observed after GFC. For instance, Korniyenko et al. (2018) study shock propagation within multiplex financial network and find rising role of Asian countries (China, Hong Kong SAR). Del Rio-Chanona et al. (2020) research revealed that several major European countries decreased in rank and that several major Asian countries increased in rank since 2008. This indicates structural changes happening in global financial system network regarding the rising role of Asian regions especially driven by China and Hong Kong SAR economies (Del Rio-Chanona et al. 2020).

Futurists also discuss some possible scenarios of re-globalization. The scenarios are not intended to predict the future but to present snapshots of a range of possible futures (Grosskurth et al. 2022) (Table 1).

Table 1. Re-globalization scenarios in previous literature

Author	Scenarios	Common grounds
Wray, J., Jones, O., Rickert McCaffrey, C., Krumbmüller, F. (2022)	4 <ol style="list-style-type: none"> 1. Self-reliance reigns 2. Globalization lite 3. Cold War II 4. Friends first 	<ol style="list-style-type: none"> 1. Bipolar peace (US & Europe vs. China) 2. Regionalization mosaic 3. Localization
Grosskurth, P., Karunska, K., Masabathula, S., Zahidi, S. (2022)	4 <ol style="list-style-type: none"> 1. Globalization 5.0: Reconnection 2. Analogue Networks: Virtual Nationalism 3. Digital dominance: Agile Platforms 4. Autarkic World: Systemic Fragmentation 	<ol style="list-style-type: none"> 4. Globalization reconnections continuing

Source: compiled by authors based on references in Table 1.

Wray et al. (2022) highlights that recent event of pandemic and the war in Ukraine, have accelerated a shift toward a multipolar world. Grosskurth et al. (2022) suggest 4 scenarios of re-globalization depending on how different economic centers of gravity will choose between physical and virtual integration, fragmentation or isolation until 2027. All scenarios could be summarized to some common grounds based on their essence (see Table 1). Bipolar peace between US & Europe vs. China could be summarized into first scenario, though the role of Europe is not completely clear. Another plausible scenario could be geographical regionalization. A tendency of localization of supply chains and investment could define third scenario. Finally, somewhat changed globalization may be continuing characterized by more responsible connections, end of geopolitical conflicts and stronger alliances. Localization scenario appears to be least likely as it would lower gains from international investment and trade. While continuance of reconnected globalization could be plausible, most recent research support regionalization-related scenarios (Del Rio-Chanona et al. 2020; Korniyenko et al. 2018; Lund et al. 2017; Lambert et al. 2015) as they would seem most likely at least for the nearest future.

Regionalization phenomena is mainly understood as countries grouping into regions in order to become more economically and politically important, working in a decentralized manner. Regionalization denotes increased connectedness at a regional level (Kim and Shin 2002). Regionalization is often analyzed using network approach as it allows to consider interconnectedness aspect. Regionalization measurement focuses mainly on intra-regional value or density comparison with inter-regional for geographical regions (Kim and Shin 2002).

Hence, as some increase in regionalization was already observed after GFC, current pandemic and geopolitical tensions might have further impacted regionalism preferences and regionalization. Thus, in this research we aim to evaluate bipolar (US & Europe vs. China) and geographical regionalization scenarios using multiplex financial network in re-globalization context.

3 Methodology

3.1 Logics and Methods

We construct multiplex financial network, which includes 5 layers used to encode different types of edges and represent diverse relationship between nodes by intralayer edges, which connect node-layer tuples within a layer. No interlayer edges, which connect node-layer tuples from different layers, are used. We construct 6 multiplex networks: (i) cross-border net direct investments in equity; (ii) cross-border net portfolio investments in equity; (iii) cross-border net portfolio investments in debt assets; (iv) cross-border net direct investments in debt assets; (v) cross-border net bank loans and deposits and (vi) aggregated network of bilateral international financial positions defined as a sum of the five individual networks. Each element (cell) x_{ij} in a matrix is a bilateral exposure from country j to country i . 12 years covered by the analysis (2009–2020) resulting in total 72 networks. We build networks for stock (positions) to capture the effect of overall position outstanding. Each country in the dataset is a node within the network. Directional links between nodes represent net cross-border investment claim positions outstanding from country j to country i . Links exist for strictly positive net positions, i.e., cross-border investment assets of a reporting country are higher than cross-border investment liabilities vis-à-vis another country ('net assets') channeled through financial system between the source and the destination country. This research is performed in 2 stages (see Table 2).

Stage 1 is aimed to reveal possibly increased trend towards geographical regionalization (Scenario 1) based on 6 world regions, i.e., Europe, Northern America, Latin America and the Caribbean, Asia, Africa and Australia and Oceania, for the first geographical regionalization scenario. Regions are divided based on United Nations world regions classification (2023), Americas region is divided into Latin America and the Caribbean and Northern America regions. Only Cyprus is reclassified from Western Asia region to Southern Europe sub-region within EU, as Cyprus belongs to European Union to keep coherency.

Table 2. Logics of the research

Stages	<i>Stage 1.</i> Multiplex network geographical regionalization (Scenario 1)	<i>Stage 2.</i> Multiplex network bipolar regionalization (Scenario 2)
Methods	Network intra and inter density, value and value to world GDP	Network communities' analysis
Data	Global financial network matrices	Global financial network matrices

Source: compiled by authors.

We then construct intraregional and interregional adjacency matrices. This is in line with the research by Kim and Shin (2002), which addressed regionalization issues. For these regions we calculate intraregional and interregional network density measures for different points in time. Simple network density is calculated using the following formula

(adapted from Martinez-Jaramillo et al. 2014):

$$d = \frac{\sum_{i=1, k}^{N, k} \sum_{j=1, k}^{N, k} x_{ijk}}{N_k(N_k - 1)} \tag{1}$$

where N is the number of nodes, k – layer index, and $d \in [0, 1]$. Comparison of density in different points in time allows to measure whether connectedness within the network increased or decreased. Intraregional density is calculated as a simple density (see formula 1), but of a network, which consists only of a certain region’s countries’ cross-border claim positions. Interregional density is also calculated as a simple density (see formula 1), but of a network, which consists only of between regions’ cross-border claim positions. We then sum up all regions’ actual connections and divide by sum of all regions’ possible connections to get intraregional and interregional density, total. We calculate such totals for each of three types of assets (layers), i.e., equity (direct and portfolio summed up), debt (direct and portfolio summed up) and banking.

We also calculate intraregional and interregional network nominal value and value to world GDP to capture network value trend for each region. We then sum up all regions’ intraregional and interregional values to get total values. We calculate them for all of three types of assets (layers). We also divide calculated total intraregional and interregional values by world GDP to eliminate nominal effect for each layer.

The aim of Scenario 1 testing is to analyze if the level of multiplex financial network geographical regionalization increased during post-crisis period. To test it we analyze data based on 3 criteria. All of them should be true to confirm that multiplex financial network geographical regionalization increased. The following criteria is used: 1) Intraregional density and value of geographical regions increased; 2) Interregional density and value of geographical regions decreased; 3) Intraregional density and value of geographical regions is higher than interregional density and value of geographical regions. For criteria 1, we compare intraregional density and value over analysis period to check if it has increasing tendency or decreasing. We follow same approach for criteria 2 for interregional density and value. In case regionalization would have increasing tendency, we would expect intraregional density and value to increase and interregional to decrease, thus indicating shift from interregional investment to intraregional. The idea behind criteria 3 is to reveal is there is state of higher intraregional density and value of geographical regions compared to interregional, thus revealing the state of regionalization. For criteria 1 and 2 testing, we use Paired t test for statistical significance analysis, where p-values are calculated using the formula (Shein-Chung et al. 2002):

$$p - value = \frac{\bar{x}_1 - \bar{x}_2}{\sigma / \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{2}$$

where \bar{x}_1 is 2017–2020 period mean, σ – standard deviation, \bar{x}_2 – 2009–2012 period mean, n_1 – number of observations of first sample period, n_2 – number of observations of first second sample period. We aim to test H_0 : intraregional and interregional each layer density and value mean of 2009–2012 is equal to the intraregional and interregional each layer density and value mean of 2017–2020. For criteria 3 testing, we use the same Paired t test for statistical significance analysis, where p-values are calculated using the formula

(2). We aim to test H_0 : intraregional 3 layer 2009–2020 mean is equal to interregional 3 layer 2009–2020 mean.

Graphical analysis of total intraregional and interregional density and value shows higher increase of density and value during the period 2009–2012 and slower increase during the period 2017–2020, thus, we separate year periods into two groups, i.e., 2009–2012 and 2017–2020 and calculate period means for each layer. We then compare period mean differences between 2 year groups for each layer and its statistical significance, to check if intraregional and interregional density and value increased. In case intraregional (interregional) either network value, network value to GDP or network density increases (decreases) significantly comparing periods 2009–2012 and 2017–2020 for any of the layers, then Criteria 1 (Criteria 2) is accepted.

Then, we calculate interregional and intraregional network value, network value to GDP and network density 3 layer mean and period 2009–2020 mean. We compare total interregional layer and period mean with intraregional layer and period mean to identify the higher. In case intraregional either network value, network value to GDP or network density is significantly higher than interregional, then Criteria 3 is accepted.

For *Stage 2* analysis Scenario 2 of bipolar regionalization is tested. Bipolar regionalization scenario suggests that 2 main regions (communities) are likely to form due to re-globalization, i.e., US and Europe (1st community) vs. China (2nd community). The aim of Scenario 2 testing is to analyze if bipolar regionalization has formed during post-crisis period with 2 main clusters - US & Europe vs. China. We calculate multiplex aggregate global financial networks' actual communities for the period 2009 and 2020 using NetMiner software. We analyze actual communities in 2009 and 2020 and compare them to communities US & Europe vs. China. In case in 2009, actual communities did not correspond to communities US & Europe vs. China, but in 2020 they did correspond, then Scenario 2 is accepted revealing the change towards communities US & Europe vs. China comparing 2009 and 2020.

3.2 Data

Research is limited by the data period available, i.e., post-GFC period from 2009 until 2020 could be covered as latest data of 2022–2023 is not yet available (see Table 3). Limitation of this research is that it includes only positive net cross-border investment positions, i.e., net positions of a reporting country vis-à-vis another country ('net assets') were used. All negative positions ('net liabilities') were replaced with zeros and ignored in the analysis in line with the research of Minoiu and Reyes (2012). Research is limited by data availability and gaps, which are caused by investment positions data not provided by some countries or suppressed due to confidentiality reasons.

For non-reporting countries or reporting, but for which data was not provided, mirror data was used.

Table 3. Research data

No.	Layer	Asset type	Countries*	Source
1	Banking	Net bank loans and deposits	208	BIS LBS by residence
2	CDIS equity	Net direct investment in equity	227	CDIS
3	CDIS debt	Net direct investment in debt	213	CDIS
4	CPIS equity	Net portfolio investment in equity	201	CPIS
5	CPIS debt	Net portfolio investment in debt	201	CPIS
6	Aggregate	All above aggregated	234	BIS, CPIS, CDIS

Source: compiled by authors based on BIS LBS by residence (2020), CPIS (2020) and CDIS (2020).

Note:* The terms “country” and “economy” do not always refer to a territorial entity that is a state as understood by international law and practice. Sometimes an economy has a separate physical or legal zone that is under its control, but to which, to some degree, separate laws are applied (e.g., a free trade zone or offshore financial center) (CDIS 2020). Nevertheless, if statistical data for territorial entities that are not states are maintained on a separate basis, such territorial entities are included in the analysis.

4 Results

4.1 Geographical Regionalization

Firstly, for interregional and intraregional network value, network value to GDP and network density change testing over the period 2009–2020, we calculate mean differences between two year groups for each layer and its statistical significance. Calculation results are provided in Table 4.

As revealed in Table 4, intraregional banking as well as interregional debt and banking network density increased significantly comparing 2009–2012 and 2017–2020 period averages. Interregional debt period 2009–2012 average increased the highest, i.e., by + 4.6 percentage points (pp) as compared to period 2017–2020 average, followed by interregional banking increase by + 2 pp. Intraregional network value has increased significantly only for equity layer by + 9.3 mln. of US dollars. Interregional equity and debt layers’ network value also increased significantly comparing 2009–2012 and 2017–2020 period averages with the highest increase in equity layer, i.e., + 11.7 trillion of US dollars. Similar results appear in relation to intraregional and interregional network value as % of world GDP. Intraregional network value as % of world GDP has increased significantly for equity layer by + 7.4 pp. Interregional network value as % of world GDP has increased significantly for equity layer by + 9.9 pp. Overall the lowest increase in intraregional as well as interregional network density, value and value as % of world GDP is observed in banking layer, then in debt layer and the highest increase in equity layer. Regarding Scenario 1 and its criteria testing, since intraregional either network value, network value to GDP or network density has increased significantly during the

Table 4. Interregional and intraregional network value, network value to GDP and network density 2009–2012 and 2017–2020 mean differences by layer

(I) 2009–2016 Mean	Network value, (millions USD)		Network value, (% of world GDP, decimal form)		Network density (% , decimal form)	
	(J) 2017–2020 Mean	P-value	Mean difference (J-I)	P-value	Mean difference (J-I)	P-value
Intraregional equity	9275668*	0.001	0.074*	0.027	0.065	0.188
Intraregional debt	339770	1.261	– 0.014	0.636	0.061	0.123
Intraregional banking	– 340251	3.931	– 0.023	1.078	0.020*	0.003
Interregional equity	11657934*	0.003	0.099*	0.013	0.041	0.109
Interregional debt	2749273*	0.001	0.014	0.140	0.046*	0.026
Interregional banking	1308854	0.095	0.000	6.000	0.020*	0.002

Source: own calculations based on BIS LBS by residence data (2020), CPIS (2020), CDIS (2020) Paired t test significance statistics' p-value shown in the table, * denote significance level of 5%. P-values corrected using Bonferroni correction (Armstrong 2014).

period 2009–2020 for any of the layers, criteria 1 is accepted. Since interregional either network value, network value to GDP or network density has also increased significantly during the period 2009–2020 for any of the layers, criteria 2 is rejected.

Next, we compare intraregional network value, network value to GDP and network density layer and period 2009–2020 mean with interregional mean in order to analyze if they differ significantly (see Table 5).

As shown in Table 5, average intraregional density is significantly higher than interregional density by +11 pp. However, intraregional network value and network value as % of world GDP are significantly lower than interregional – intraregional network value as % of world GDP is lower by –2 pp, and nominal network value lower by –1.6 trillion of US dollars. Thus, even though network has become significantly more connected intraregionally, significant investments in terms of value are made interregionally. Regarding criteria 3 testing, since intraregional network density layer and period 2009–2020 mean is significantly higher than interregional (even though network intraregional value and value to GDP is significantly lower than interregional), criteria 3 is accepted. Thus, overall, since only 2 of 3 criteria are accepted, Scenario 1 is rejected, concluding that the level of multiplex financial network geographical regionalization did not increase during post-crisis period. It is noted that geographical regionalization increase in network density could be observed and a decrease in regionalization based on network value. It is observed that interregional network density is also increasing,

Table 5. Interregional and intraregional network value, network value to GDP and network density layer and period 2009–2020 mean comparison

	Network density (% decimal form)	Network value (% to world GDP, decimal form)	Network value, millions USD
Intraregional 3 layer 2009–2020 mean (I)	0.208	0.134	10249253
Interregional 3 layer 2009–2020 mean (J)	0.098	0.154	11875230
Mean Difference (J-I)	−0.110*	0.020*	1625977*
P-value	0.000	0.000	0.000

Source: own calculations based on BIS LBS by residence data (2020), CPIS (2020), CDIS (2020) Paired t test significance statistics' p-value shown in the table, * denote significance level of 5%.

which suggest that regionalization and globalization are not contradictory processes. In addition, interregional network value is higher than interregional indicating that main investment amounts are invested interregionally. Hence, such increasing trend of interregional investment may suggest a shift towards interregional connections also given the increase in interregional density. As results reveal that not only intraregional connections are increasing, but also interregional, it could not suggest increasing regionalization, but it may indicate new clusters forming, which may include also interregional countries. Analysis of network communities is needed to identify intraregional and interregional investment clusters.

4.2 Bipolar Regionalization

Firstly, we calculate aggregate global financial networks' communities for the period 2009–2020. Communities are calculated using Louvain algorithm created by Blondel et al. (2008) once on the aggregated network, which is one of the most widely used algorithm for community detection. In order to analyze the change of countries' composition within the communities and to check what clusters are forming, whether there were some changes and to check if bipolar segmentation scenario of US & Europe vs. China alliances is likely to happen, we map each country to its respective community (see Fig. 2).

Aggregate financial networks had 3 communities in total in 2009 and 3 as well in 2020. However, countries in these communities have changed. In 2009, Community 1 (light grey) was mainly constituted from Southern, Northern and Eastern Europe (including Russia); Western, South-eastern, Central and Southern Asia (including India); some Caribbean countries and New Zealand. In 2009, Community 1 had been characterized by a close geographical proximity of cluster countries. In 2020, Community 1 European part remained connected, however, Northern America (including US and Canada) has joined. In addition, several countries of South America (including Brazil, Colombia) have joined Community 1. It is also noted, that countries in Community 1 in 2020,

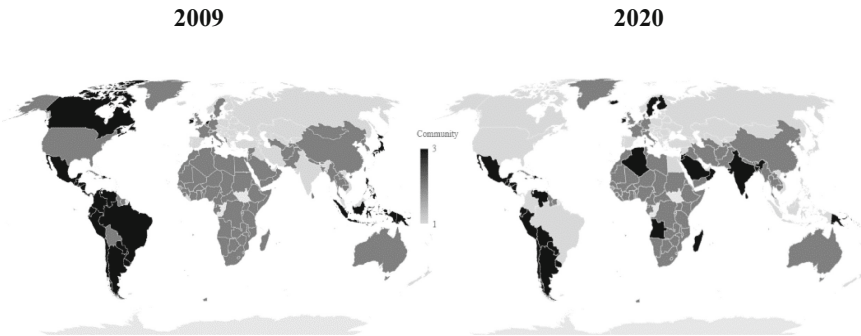


Fig. 2. Aggregate financial networks' communities in 2009 and 2020. Source: own calculations based on BIS LBS by residence data (2020), CPIS (2020), CDIS (2020) Country color depends on its community, calculated using Louvain algorithm created by Blondel et al. (2008).

became much less impacted by geographical proximity as countries from 3 different world regions constituted this community. Thus, if in 2009 Community 1 was mainly led by Eastern Europe, in 2020, US and Canada have joined leaders of this cluster.

Community 2 (dark grey) in 2009 mainly included Western (including France, Germany, Belgium), some Northern Europe countries, US, some south America and Caribbean countries, almost all African countries; Central, South-eastern, Western, Southern and Eastern Asia (including China) and Australia and Oceania region (including Australia). Community 2 in 2009 was not tightly connected geographically. In 2020, Africa, Asia, Western and Northern Europe, Australia and Oceania countries remained in Community 2. However, US has left this cluster. In 2020, Community 2 remained dispersed geographically, however, to a somewhat lesser extent. Thus, if in 2009 Community 1 was mainly led by US, Western Europe and China, in 2020, US have left this cluster, it became mainly led by Western Europe and China.

In 2009, Community 3 (black) mainly included Canada, South America (including Brazil, Mexico, Chile, Colombia), some Eastern Asia countries (including Japan) and South-eastern Asia (including Indonesia). In 2009, Community 3 had been characterized by a close geographical proximity of cluster countries. In 2020, Community 3 countries have changed to a large extent as Northern Europe (including Finland and Sweden), some Asia countries (including India) and some Africa countries have joined the cluster, but Canada, South-eastern Asia (including Indonesia) and some South America countries (including Brazil and Colombia) have left this cluster. In addition, in 2020 Community 3 became much more dispersed geographically. Thus, if in 2009 Community 3 was mainly led by Canada, Brazil, Colombia, Japan and Indonesia, in 2020, Finland, Sweden and India became leaders of this cluster.

Robustness of Louvain communities is checked by comparing no. of countries in Louvain community as percentage of countries in Modularity community, computed using algorithm by Wakita and Tsurumi (2007), see Table 6. Both algorithms give 3 communities, no. of countries overlapping in both algorithms communities' is high, thus, we consider Louvain communities' results robust.

Table 6. Percentage of overlapping countries in Louvain and Modularity communities in 2009 and 2020

Communities	Type	2009	2020	Type	2009	2020
1	Modularity % of	93%	88%	Louvain % of	59%	61%
2	Louvain	88%	74%	Modularity	94%	72%
3		17%	44%		50%	63%

Source: own calculations based on BIS LBS by residence data (2020), CPIS (2020), CDIS (2020).

Concluding Louvain Community 1, 2 and 3 analysis, results show that in 2009 US and China belonged to the same Community 2, however, in 2020, China remained in the same Community 2, but US has joined Community 1. Thus, in 2020, 2 main clusters have formed – one led by China and Western Europe and another by US and Northern and Eastern Europe. Hence, Scenario 2, that multiplex financial network bipolar regionalization has formed during post-crisis period with 2 main clusters - US & Europe vs. China, is accepted. However, it is worth noting that not all European countries belongs to the same cluster, but rather Eastern and Northern Europe, while Western Europe belongs to the same cluster as China.

5 Conclusions and Discussion

Re-globalization is a currently understudied topic and previous research focuses mostly on theoretical discussion of the problem. Empirical re-globalization related research suggests that re-globalization in terms of structural financial network changes did not start recently but appears to be observed after global financial crisis. Global financial system analysis after global financial crisis in 2008 already revealed some regionalization trends based on structural network analysis, which might have been further strengthened by pandemic and recent geopolitical tensions. Among discussed possible future scenarios of re-globalization, most likely are discussed to be geographical regionalization mosaic or ally-based bipolar regionalization.

Multiplex financial network constructed from cross-border capital stock appears to be highly regionalized. Not only intraregional density is increasing, but also interregional. It suggests that globalization as well as regionalization may be not contradictory processes in line with Kim and Shin (2002). Geographical regionalization scenario analysis revealed that intraregional equity network value, network value to GDP or network density has increased significantly and intraregional debt and banking network density has increased significantly. Concerning interregional debt and equity network value, network value to GDP or network density it has also increased significantly as well as interregional banking network value and network density. Intraregional network density is higher than interregional, but intraregional value and value to GDP is lower than interregional. Thus, overall results suggest that geographical regionalization did not increase during post-crisis period. Our results are supported by Altman & Bastian (2023) who show decreasing intraregional share, which does not reveal increase in regionalization. Bipolar regionalization scenario revealed that networks are highly clustered as number

of communities is small. In 2020, 2 main clusters have formed – one led by China and Western Europe and another by US and Northern and Eastern Europe. Hence, bipolar regionalization scenario with 2 main clusters formed - US & Europe vs. China, is accepted. However, not all Europe belongs to the same cluster as US, but rather Eastern and Northern Europe, while Western Europe belongs to the same cluster as China. Thus, further position of Europe in cluster formation could be of change. Our results support Talebian and Kemp-Benedict (2020) who claim that bipolar regionalization could be one of likely scenarios of re-globalization.

As increase in geographical regionalization could not be confirmed, but bipolar regionalization could, it suggests that geographical proximity is no longer a decisive factor considering cross-border capital investment, but region formation is likely to be impacted by other factors such as alliances, familiarity or ideological and political rationale. Hence, this research results support friend shoring rather than nearshoring. Future research could aim to explore in detail what are the main factors affecting ally-familiarity based region formation.

References

- Altman, S.A., Bastian, C.R.: Don't overestimate shifts from globalization to regionalization. Industrial analytics platform, January 2023 (2023)
- Armstrong, R.A.: When to use the Bonferroni correction. *Ophthalmic Physiol. Opt.* **2014**(34), 502–508 (2014). <https://doi.org/10.1111/opo.12131>
- BIS LBS by residence data. Bilateral cross-border claims data (2020). <http://www.bis.org/statistics/bankstats.htm>. Accessed 21 Sept 2023
- Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008** (2008)
- CDIS. CDIS Survey. IMF (2020). <https://data.imf.org/?sk=40313609-F037-48C1-84B1-E1F1CE54D6D5>. Accessed 05 Aug 2023
- CPIS. CPIS Survey. IMF (2020). <https://data.imf.org/?sk=B981B4E3-4E58-467E-9B90-9DE0C3367363>. Accessed 05 Aug 2023
- Del Rio-Chanona, R.M., Korniyenko, Y., Patnam, M., Porter, M.A.: The multiplex nature of global financial contagions. *Appl. Network Sci.* **5**(1), 1–23 (2020)
- Gaigaliene, A., Jurakovaite, O., Legenzova, R.: Assessment of EU banking network regionalization during post-crisis period. *Oeconomia Copernicana.* **9**, 655–675 (2018)
- Grosskurth, P., Karunska, K., Masabathula, S., Zahidi, S.: Four Futures for Economic Globalization: Scenarios and Their Implications. World Economic Forum, White Paper, May (2022)
- Kim, S., Shin, E.H.: A longitudinal analysis of globalization and regionalization in international trade: a social network approach. *Soc. Forces* **81**(2) (2002)
- Korniyenko, Y., Patnam, M., Porter, M., Del Rio-Chanona, R.M.: Evolution of the Global Financial Network and Contagion: A New Approach. IMF Working Paper, 18/113 (2018)
- Lambert, F., Deb, P., Ehrentraud, J., Gonzjlez-Hermosillo, B.: International banking after the crisis: increasingly local and safer? IMF, April 2015 (2015)
- Lund, S., Windhagen, E., Manyika, J., Härle, P., Woetzel, J., Goldshtein, D.: The new dynamics of financial globalization. McKinsey Global Institute (2017)
- Martinez-Jaramillo, S., Alexandrova-Kabadjova, B., Bravo-Benitez, B., Solórzano-Margain, J.P.: An empirical study of the Mexican banking system's network and its implications for systemic risk. *J. Econ. Dyn. Control* **40**, 242–265 (2014)

- Minoiu, C., Reyes, J.A.: A Network Analysis of Global Banking: 1978–2009. *IMF Working Papers* **11**(74), 1 (2012)
- Paul, T.V.: Globalization, deglobalization and reglobalization: adapting liberal international order. *International Affairs* **97**(5) (2021)
- Scott, J., Wilkinson, R.: Reglobalizing trade: progressive global governance in an age of uncertainty. *Globalizations* **18**(1), 55–69 (2021). <https://doi.org/10.1080/14747731.2020.1779965>
- Shein-Chung, C., Jun, S., Hansheng, W.: A note on sample size calculation for mean comparisons based on noncentral T-Statistics. *J. Biopharm. Stat.* **12**(4), 441–456 (2002). <https://doi.org/10.1081/BIP-120016229>
- Talebian, S., Kemp-Benedict, E.: A breakdown in globalization or a world committed to sustainability? Five scenarios for exploring the post-COVID-19 world. *LeadIT Brief.* 7 (2020)
- United Nations world regions classification (2023). *Geographic Regions*, <https://unstats.un.org/unsd/methodology/m49/>. Accessed 14 Jul 2023
- Wakita, K., Tsurumi, T.: Finding community structure in mega-scale social networks (2007). <https://doi.org/10.48550/arXiv.cs/0702048>
- Wray, J., Jones, O., Rickert McCaffrey, C., Krumbmüller, F.: The future of globalization. Prepare now for a new era. EY-Parthenon, September 2022 (2022)



A Modular Network Exploration of Backbone Extraction Techniques

Ali Yassin¹(✉), Hocine Cherifi², Hamida Seba³, and Olivier Togni¹

¹ Laboratoire d'Informatique de Bourgogne - Univ. Bourgogne - Franche-Comté, Dijon, France

aliyassin4@hotmail.com

² ICB UMR 6303 CNRS - Univ. Bourgogne - Franche-Comté, Dijon, France

³ Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, UMR5205, 69622 Villeurbanne, France

Abstract. Network backbone extraction techniques reduce the size of networks while trying to preserve their topological features. The literature reports numerous backbone extraction algorithms. However, there are few works about their ability to highlight the network community structure, although it is an essential property of many real-world networks. This paper presents an experimental comparison of six popular backbone extraction techniques in a typical modular network (Disparity Filter, Locally Adaptive Network Sparsification (LANS), Doubly Stochastic, High Saliency Skeleton, Metric Backbone, globally and locally adaptive network backbone (GLANB)). Investigations on a modular network representing the American elementary school system reveal valuable insights into how each technique influences the network's underlying community structure. Disparity and LANS backbones exhibit multi-component structures. The Doubly Stochastic backbone maintains transitivity. Moreover, it retains a significant proportion of inter-community edges and maintains a balanced ratio of intra and inter-community links. Other methods prioritize intra-community edges. The GLANB method excels in network filtering and accurate representation of the community structure. By shedding light on these nuanced aspects of backbone extraction techniques, our study contributes to a better understanding of their effects on network topology, enabling their practical use in real-world scenarios.

Keywords: Complex Networks · Backbone Extraction · Filtering Techniques · Network Compression · Graph Summarization · Sparsification · Communities · Community Detection · Social Network · Network Analysis

1 Introduction

In recent decades, networks have become a valuable tool for complex systems analysis with multiple applications such as computer vision [1–4] and 3D object modeling [5–9]. They model complex systems, using nodes to denote elements

and edges representing their interactions. Common analytical tasks include community detection [10, 11], identification of influential nodes [12–16], and investigation of network formation [17]. Processing large-scale networks presents substantial challenges. Therefore, several backbone extraction methods have been developed to reduce the network’s size while retaining its essential characteristics. One can classify these methods into two primary categories: structural and statistical approaches.

Structural techniques involve filtering edges or nodes based on specific topological properties such as network modularity distance metrics and identifying overlapping communities [18–21]. Statistical methods such as the disparity filter [22] assess edge significance through statistical tests removing the least significant edges.

In recent years, there has been a notable surge in the study of transportation and urban networks [23]. Network backbone extraction methods have proven instrumental in expediting analysis and enhancing the visualization of these complex networks. They facilitate rapidly identifying vital spatial and topological structures within the network. Prior research has already compared statistical and structural methods in this domain [24–26].

Brattig et al. [27] investigate nine contact networks in a recent study. They show that these networks contain a significant number of redundant links. Interestingly, they show using the metric backbone to remove these redundancies has minimal impact on the community structure and the spread of epidemics. Consequently, they proposed the metric backbone as an optimal subgraph for the sparsification of social contact networks. Building upon this work, we compare six backbone extraction methods, encompassing two statistical, three structural, and one hybrid approach. Our goal is to investigate how well they preserve the original community structure.

The selected methods cover various backbone extraction methodologies. Statistical methods include the popular Disparity filter [22] and the LANS filter [28]. The Disparity filter exploits a uniform null model, while LANS rely on the empirical distribution of weights. Structural techniques incorporate the metric backbone method [19], the high salience skeleton [18], and the doubly stochastic method [29]. The Metric backbone fully preserves the shortest paths of the original graph. The high-salience skeleton removes low-salience links, and the doubly stochastic method relies on normalized edge weights. Additionally, we incorporate a hybrid method, the GLANB [30], combining the Disparity filter and high salience skeleton.

The Experimental comparative evaluation involves several steps. First, we use the Netbone package [31] to extract the backbones of the American Elementary school network. Second, we evaluate these backbones’ fundamental topological properties, including edge and node fractions, component count, and transitivity. Following this preliminary analysis, we quantify the proportion of intra and inter-community edges retained by each method compared to the original network. Finally, we assess their effectiveness in filtering the network while preserving the original community structure.

2 Backbone Extraction Methods

Backbone extraction methods aim to identify essential network components. They fall into two categories: statistical methods, which use hypothesis testing to evaluate edge importance, and structural methods, which consider network topology. Hybrid methods combine both approaches. Table 1 presents briefly the methods under test.

3 Data and Methods

This section introduces the dataset under examination and describes the methodology of the comparative analysis.

3.1 Data

The US-ES contact network depicts student social interactions at an American Elementary School. It spans seven grades, each with three classes. In Fig. 1 (panel A), nodes represent students, color-coded by grade, with varying shades for classes. This visualization highlights community patterns: more interactions within the same class and, alternatively, within the same grade. This dataset [32] was compiled at a suburban elementary school in Utah, USA, over two days, specifically on January 31st and February 2nd, 2013. It was recorded at intervals of approximately 20s. The metadata included information on gender and grades (ranging from Kindergarten to 6th grade), with 21 different classes spanning across seven grades. Table 2 reports the main topological properties of the network.

3.2 Methods

To assess the performance of the backbone extraction techniques, we conducted a series of three experiments. First, using the netbone package [31], we extract the various backbones and analyze their basic topological properties (edge and node fractions, number of components, and transitivity). This analysis allows evaluation of how filtering influences network connectivity and transitivity, as these factors significantly impact the performance of community detection algorithms. Indeed, one can apprehend isolated components as detached communities. Furthermore, higher transitivity typically signifies more robust community structures.

In the second experiment, we plot the extracted backbone using Gephi [33] to visualize the community structure. Then, we investigate the proportion of intra and inter-community edges preserved in each backbone concerning the original network and the corresponding backbone. This examination allows us to determine if the backbone extraction methods are biased toward intra or inter-community edges. This factor directly influences the effectiveness of community detection algorithms in uncovering the communities.

Table 1. Overview of backbone extraction methods characteristics.

Category	Method	Description	Scope	Parameters
Statistical	Disparity [22]	Assumes that the normalized weights of a node’s edges follow a uniform distribution. Then it computes the edge p-values by comparing the observed normalized edge weights to this null model.	Local	α (significance level)
	LANS [28]	It employs the empirical cumulative density function to evaluate the statistical significance of an edge. It calculates the probability of choosing an edge randomly with a weight equal to the observed weight.	Local	α (significance level)
Structural	Doubly Stochastic [29]	It transforms the network’s adjacency matrix into a doubly stochastic matrix by iteratively normalizing the row and column values using their respective sums. Next, it sorts the edges in descending order based on their normalized weight. Finally, it adds the edges to the backbone sequentially until it includes all nodes in the original network as a single connected component.	Local	-
	High Saliency Skeleton [18]	It constructs a shortest path tree for each node by merging all the shortest paths from that node to every other node in the network. Then, the edge saliency is computed as the proportion of shortest-path trees where the edge is present.	Global	β (threshold)
	Metric Backbone [19]	It extracts a subgraph comprising the shortest paths within the network. The shortest path length is defined as the sum of the edge distances	Global	-
Hybrid	GLANB [30]	It defines the involvement of an edge as the fraction of all the shortest paths connecting a node to the rest of the network through this edge. Then, it defines a null hypothesis to determine the statistical significance of each edge based on its involvement. A parameter regulates the influence of the node’s degree on its statistical significance	Local & Global	c (involvement) & α (significance level)

Lastly, we evaluate the ability of the backbone to unveil the community structure. We employ the Louvain algorithm [34] to identify the community structure within the original network and the extracted backbones. We consider the classes in the social network as the ground truth. Subsequently, we compare the composition of nodes within the backbone communities with the ground truth to determine which backbone method best facilitates the identification of the community structure.

Table 2. The Topological features of the American Elementary School social network. N is the number of nodes. E is the number of edges. $\langle k \rangle$ is the average degree. ρ is the density. T is the transitivity.

N	E	$\langle k \rangle$	ρ	T
339	16,546	97.6	0.28	0.44

4 Experimental Results

4.1 Backbones Basic Topological Properties

Table 3 reports the basic topological characteristics of the backbones extracted from the US-ES network. It includes the fraction of edges, the fraction of nodes, the number of components, and the transitivity of each backbone. We rank the methods in descending order of their fraction of edges. The Doubly Stochastic backbone ranks first. It retains approximately 18% of the original social interactions. The worst is the High Saliency Skeleton, with around 3% of the interactions remaining in the backbone. Other methods are between these two extremes, keeping roughly 5% to 6% of the edges from the original network. Importantly, all the backbone extraction methods retain the complete set of nodes within the network.

Interestingly, most methods maintain a single connected component, except for the Disparity and LANS methods. Their backbones split into multiple components of differing sizes.

Transitivity decreases in all backbones except for the Disparity Filter extracted backbone. Indeed, its transitivity increases from 0.44 in the original network to 0.55 in the backbone. The Doubly Stochastic backbone demonstrates the lowest deviation from the original network, with a transitivity value equal to 0.43. The High Saliency Skeleton backbone is not transitive. For other backbones, the transitivity values range from 0.19 to 0.35.

Table 3. The fraction of edges retained, the fraction of nodes preserved, the number of components, and the transitivity of each backbone.

Backbone	Edge Fraction	Node Fraction	Components	Transitivity
Doubly Stochastic	0.182	1.0	1	0.43
Disparity	0.104	1.0	4	0.55
Metric Backbone	0.068	1.0	1	0.26
LANS	0.064	1.0	7	0.35
GLANB	0.054	1.0	1	0.19
High Saliency Skeleton	0.031	1.0	1	0.07

4.2 Qualitative Comparisons of the Backbones Community Structure

We visualize the extracted backbone with the same node colors and positions as the original network. They are arranged in descending order according to the fraction of retained edges, in Fig. 1B–G.

First, all the extracted backbones exhibit greater clarity and sparsity than the original network. Notably, each method maintains community connections,

specifically the edges linking students within the same class. Nonetheless, the density of edges within these communities correlates directly with the number of edges retained in the backbone. For instance, the Doubly Stochastic Backbone in Fig. 1B displays highly interconnected communities, as it keeps the largest fraction of edges. In contrast, the communities exhibit reduced connectivity in the High Saliense Skeleton illustrated in Fig. 1G.

Some methods preserve the edges connecting different communities. For example, in Fig. 1B, the Doubly Stochastic Backbone communities are almost entirely interlinked. Conversely, in Fig. 1E, the communities of the LANS Back-

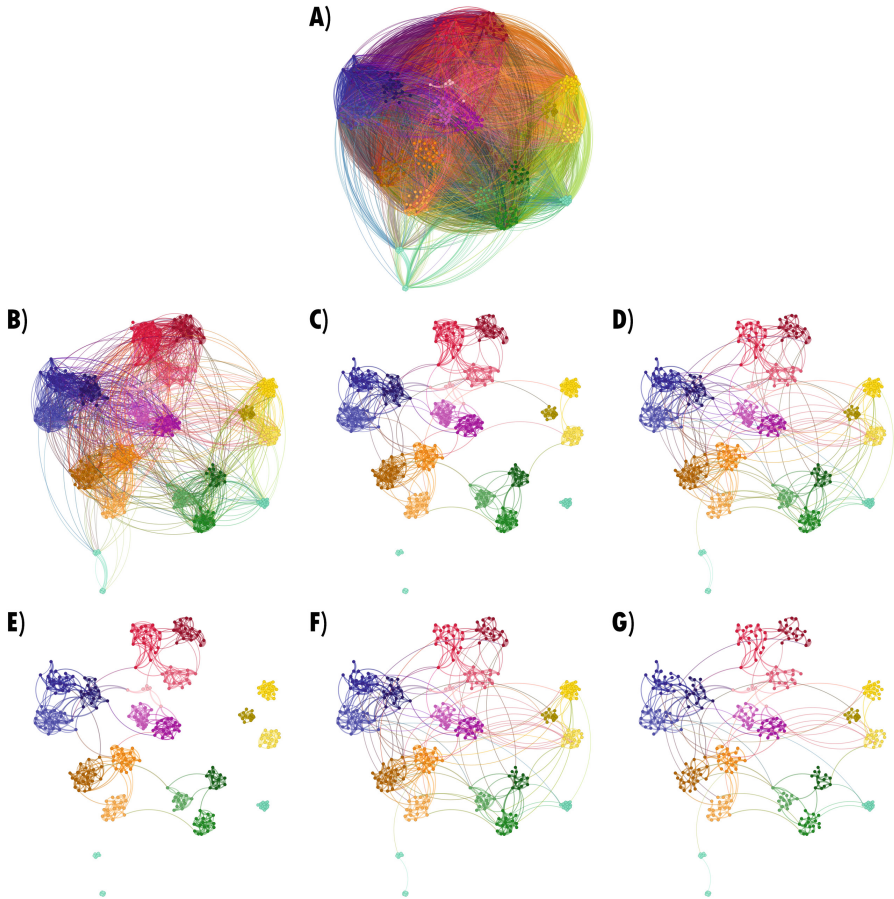


Fig. 1. The American Elementary School Social Network: (A) The original network. (B) Doubly Stochastic Backbone, (C) Disparity Backbone, (D) Metric Backbone, (E) LANS Backbone, (F) GLANB Backbone, and (G) High Saliense Skeleton Backbone. The backbones are arranged in descending order according to the fraction of retained edges. The Colors represent the grades: kindergartners in cyan, 1st grade in yellow, 2nd in green, 3rd in orange, 4th in pink, 5th in blue, and 6th in red. Lighter or darker shades of the same color separate classes within the grade.

bone exhibit minimal interconnections. Meanwhile, the community linkages within the other backbone methods fall between these extremes, featuring varying connections between communities.

4.3 Preserving Inter or Intra-community Edges

To gain deeper insights into the extracted backbones within the framework of community structure, we compare the proportion of intra-community and inter-community edges within the extracted backbones and the original network. As a reminder, intra-community edges refer to connections between students within the same class. In contrast, inter-community edges refer to relations between nodes belonging to different classes in the social network. Table 4 reports the percentage of preserved intra-community and inter-community links in each backbone.

Looking at the retained intra-community and inter-community edges, it appears all methods preserve more intra-community than inter-community edges. The Doubly Stochastic backbone conserves nearly 10% of the original network’s inter-community edges. In stark contrast, the other methods retain less than 2% of these edges. It is roughly one-ninth of what the Doubly Stochastic method preserves.

Shifting focus to analyzing intra and inter-community edges within each backbone, a distinct pattern emerges. The Doubly Stochastic method maintains a nearly equal distribution between intra and inter-community edges. Indeed, 58% are intra-community edges and 42% are inter-community edges. Conversely, the other methods emphasize retaining intra-community edges, wherein at least 80%.

Table 4. The ratio of preserved intra-community and inter-community edges in each backbone relative to both the edges in the original network and the edges in the corresponding backbone.

Backbone	Network		Backbone	
	% Intra Edges	% Inter Edges	% Intra Edges	% Inter Edges
Doubly Stochastic	63.51	9.29	57.63	42.37
Disparity	57.94	1.02	91.86	8.14
Metric	33.43	1.61	80.53	19.47
LANS	36.56	0.49	93.66	6.34
GLANB	26.77	1.27	80.77	19.23
High Saliency Skeleton	15.77	0.69	82.01	17.99

4.4 The Backbone Power in Revealing the US-ES Communities

In the US-ES social network context, we consider that classes represent the ground truth communities. Consequently, the social network contains twenty-one communities, with three corresponding to each grade. To evaluate the ability

of the extracted backbones to unveil these communities, we use the Louvain community detection algorithm [34] to the original network and the extracted backbones. Table 5 summarizes the results.

The community detection algorithm uncovers 13 communities in the original network. It is considerably fewer than the reference ground truth. In contrast, Louvain detect more communities in the extracted backbones, However, this number is still lower than the ground truth. Specifically, the Doubly Stochastic and Metric backbones revealed 16 and 18 communities, respectively. In the Disparity, Metric, LANS, GLANB, and High Saliency Skeleton (HSS) backbones, the Louvain algorithm identify 19 communities, just two communities shy of the ground truth.

Table 5. The number of classes (considered as the ground truth) and the number of communities identified by the Louvain algorithm in the original network and the extracted backbones.

Ground Truth	Original	Doubly Stochastic	Metric	Disparity	LANS	GLANB	HSS
21	13	16	18	19	19	19	19

However, the table merely presents the number of communities in the backbones. We conduct a more comprehensive analysis to understand better which backbone best captures the community structure. This analysis involves comparing the nodes within the communities between the ground truth, the original network, and the extracted backbones. Figure 2 illustrates two layers of Snakey plots for each backbone extraction method. On the left, we observe the communities detected in the original network using the Louvain algorithm. The social network classes (the ground truth) are in the middle. On the far right, we find the communities within the backbones, also identified by the Louvain algorithm. The plots are organized in ascending order according to the number of communities the Louvain algorithm detects.

Upon closer examination, we notice that the Louvain algorithm tends to merge multiple classes when applied to the original network. This merging manifests in two ways: firstly, it combines classes from different grades, such as joining class 20 and a portion of class 12 (belonging to grades 6 and 4, respectively). Secondly, it combines classes within the same grade, with classes 15, 16, and 17 forming grade 5.

Likewise, the Doubly Stochastic backbone merges some classes. Indeed, classes 15, 16, and 17 form a single community. Simultaneously, it disperses nodes from five ground truth classes among backbone communities. For instance, nodes from class 13 are distributed between the communities of class 12 and class 14 in the backbone. We observe a comparable pattern in the High Saliency Skeleton Backbone (bottom right plot).

In contrast, the community structure identified in the other backbones closely resembles the ground truth, except for a few class mergers within the backbones.

For instance, the Metric, Disparity, LANS, and GLANB backbones consider classes 12 and 13 as one community. Classes 15 and 17 are treated as a single community in the Metric, Disparity, and LANS backbones, while the Metric backbone combines classes 9 and 10. Additionally, the GLANB backbone identifies classes 15 and 16 as a single community. These merged classes are in the same grades, such as classes 12 and 13 in grade 4, classes 15 and 17 in grade 5, and classes 9 and 10 in grade 3.

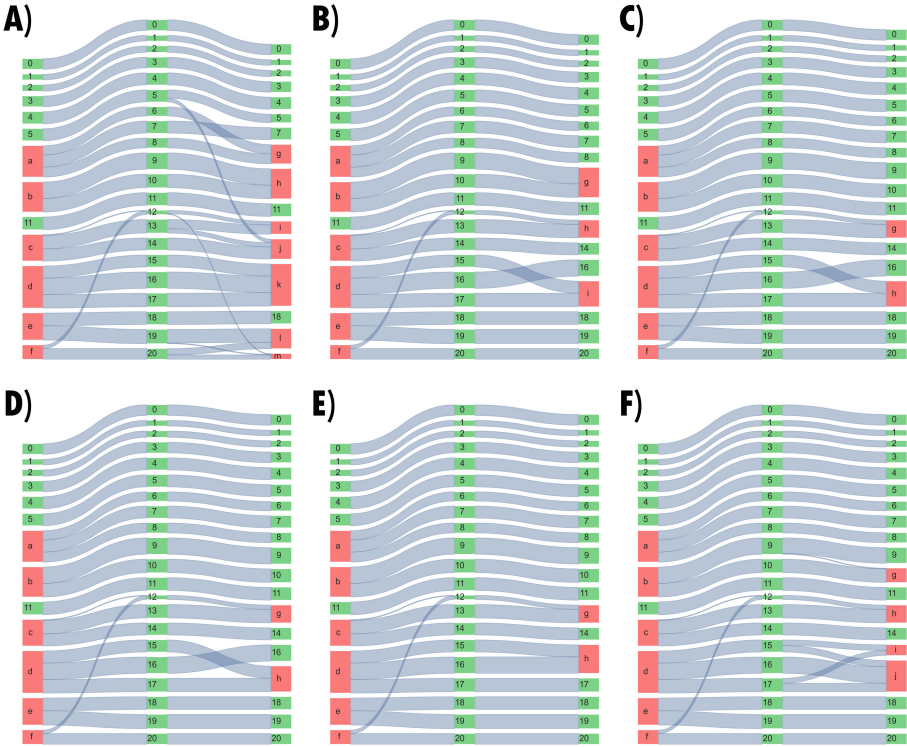


Fig. 2. Two Layers Snakey plots comparing the communities: On the left, the original network communities detected using the Louvain algorithm. In the middle are the ground truth communities representing the school classes. On the right are the backbone communities detected using the Louvain algorithm. **(A)** Doubly Stochastic Backbone, **(B)** Metric Backbone, **(C)** Disparity Backbone, **(D)** LANS Backbone, **(E)** GLANB Backbone, and **(F)** High Saliency Skeleton Backbone. The green labels represent the detected communities that follow the ground truth, while the red one doesn't.

5 Discussion

This investigation involves comparing six distinct methods for extracting backbones of the American Elementary School social network.

Initially, we extract the backbones from the social network and assess their fundamental topological characteristics, including the proportions of retained edges, nodes, the number of components, and transitivity. The findings reveal that the Doubly Stochastic method preserves the largest number of edges among the methods. Furthermore, all methods inherently retain all nodes, except for the High Saliency Skeleton, which we adjusted to maintain all nodes. Concerning connectivity and transitivity, the results indicate that the LANS and Disparity backbones have multiple components, whereas the Doubly Stochastic backbone closely aligns with the original network transitivity.

Subsequently, we evaluate the ratio of intra and inter-community edges within each backbone compared to the original network and the corresponding backbone. The results indicate that all methods retain more intra-community edges from the original network. Notably, the Doubly Stochastic process keeps the highest fraction of inter-community edges. The Doubly Stochastic method includes an equivalent proportion of intra and inter-community edges, while the other methods prioritize intra-community edges.

Lastly, we assess the effectiveness of the backbones in uncovering the community structure by comparing it to the ground truth and the original network. The results demonstrate that all methods outperform the original network in revealing the community structure. However, the Doubly Stochastic method uncovers the fewest ground truth communities, potentially due to its retrieval of many inter-community edges. Conversely, the other techniques reveal more communities. Nevertheless, the High Saliency Skeleton communities mix the communities similarly to the Doubly Stochastic backbone. In contrast, the Metric, Disparity, LANS, and GLANB backbone communities closely align with the ground truth. It is worth noting that the Disparity and LANS backbones consist of multiple components. Thus, among all the methods, the GLANB method is the most effective in filtering the network while revealing the underlying community structure. The metric backbone method follows.

Furthermore, it's important to highlight that the GLANB method falls into the category of hybrid methods, combining both structural and statistical approaches. It proves efficient in this context.

6 Conclusion

This exploration of network backbones extraction techniques uncovers various insights into their behaviors and influence on the network's community structure.

Comparing the basic properties of these backbones shows that most methods effectively preserve network connectivity, while the Disparity and LANS backbones split the backbone into multiple components. The Doubly Stochastic backbone maintains transitivity, whereas other methods either decrease or increase it. These observations have direct implications for the community structure within the extracted backbones.

The balance between intra and inter-community edges shows that the Doubly Stochastic method retains the highest proportion of inter-community edges. Furthermore, concerning the backbone, the Doubly Stochastic algorithm keeps an

almost equal ratio of intra and inter-community edges. Other methods prioritize intra-community edges.

Evaluating the efficacy of these backbone extraction methods

GLANB emerges as the most effective method for revealing the community structure. It excels in both filtering the network effectively and preserving its community structure. The metric backbone method also performs well in this context.

Future investigations will extend this preliminary study to various real-world networks to validate and expand upon these insights.

Acknowledgment. This material is based upon work supported by the Agence Nationale de Recherche under grant ANR-20-CE23-0002.

References

1. Rital, S., Cherifi, H., Miguet, S.: Weighted adaptive neighborhood hypergraph partitioning for image segmentation. In: Singh, S., Singh, M., Apte, C., Perner, P. (eds.) ICAPR 2005. LNCS, vol. 3687, pp. 522–531. Springer, Heidelberg (2005). https://doi.org/10.1007/11552499_58
2. Pastrana-Vidal, R.R., Gicquel, J.-C., Colomes, C., Cherifi, H.: Frame dropping effects on user quality perception. IN: Proceedings of 5th International WIAMIS (2004)
3. Pastrana-Vidal, R.R., Gicquel, J.C, Blin, J.L., Cherifi, H.: Predicting subjective video quality from separated spatial and temporal assessment. In: Human Vision and Electronic Imaging XI, vol. 6057, pp. 276–286. SPIE (2006)
4. Demirkesen, C., Cherifi, H.: A comparison of multiclass SVM methods for real world natural scenes. In: Blanc-Talon, J., Bourennane, S., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2008. LNCS, vol. 5259, pp. 752–763. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88458-3_68
5. Hamidi, M., Chetouani, A., El Haziti, M., El Hassouni, M., Cherifi, H.: Blind robust 3D mesh watermarking based on mesh saliency and wavelet transform for copyright protection. *Information* **10**(2), 67 (2019)
6. Abouelaziz, I., El Hassouni, M., Cherifi, H.: No-reference 3D mesh quality assessment based on dihedral angles model and support vector regression. In: Mansouri, A., Nouboud, F., Chalifour, A., Mammass, D., Meunier, J., ElMoataz, A. (eds.) ICISP 2016. LNCS, vol. 9680, pp. 369–377. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-33618-3_37
7. Abouelaziz, I., El Hassouni, M., Cherifi, H.: A convolutional neural network framework for blind mesh visual quality assessment. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 755–759. IEEE (2017)
8. Abouelaziz, I., Chetouani, A., El Hassouni, M., Jan Latecki, L., Cherifi, H.: Convolutional neural network for blind mesh visual quality assessment using 3D visual saliency. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 3533–3537. IEEE (2018)
9. Abouelaziz, I., Chetouani, A., El Hassouni, M., Latecki, L.J., Cherifi, H.: 3D visual saliency and convolutional neural network for blind mesh quality assessment. *Neural Comput. Appl.* **32**, 16589–16603 (2020)
10. Fortunato, S., Hric, D.: Community detection in networks: a user guide. *Phys. Rep.* **659**, 1–44 (2016)

11. 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)
12. Chakraborty, D., Singh, A., Cherifi, H.: Immunization strategies based on the overlapping nodes in networks with community structure. In: Nguyen, H.T.T., Snasel, V. (eds.) *CSoNet 2016*. LNCS, vol. 9795, pp. 62–73. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42345-6_6
13. 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)
14. 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)
15. Rajeh, S., Savonnet, M., Leclercq, E., Cherifi, H.: Interplay between hierarchy and centrality in complex networks. *IEEE Access* **8**, 129717–129742 (2020)
16. Rajeh, S., Savonnet, M., Leclercq, E., Cherifi, H.: Characterizing the interactions between classical and community-aware centrality measures in complex networks. *Sci. Rep.* **11**(1), 10088 (2021)
17. Orman, G.K., Labatut, V., Cherifi, H.: Towards realistic artificial benchmark for community detection algorithms evaluation. arXiv preprint [arXiv:1308.0577](https://arxiv.org/abs/1308.0577) (2013)
18. Grady, D., Thiemann, C., Brockmann, D.: Robust classification of salient links in complex networks. *Nat. Commun.* **3**, 864 (2012)
19. Simas, T., Correia, R.B., Rocha, L.M.: The distance backbone of complex networks. *J. Compl. Netw.* **9** (2021)
20. Rajeh, S., Savonnet, M., Leclercq, E., Cherifi, H.: Modularity-based backbone extraction in weighted complex networks (2022)
21. Ghalmane, Z., Cherifi, C., Cherifi, H., El Hassouni, M.: Extracting backbones in weighted modular complex networks. *Sci. Rep.* **11**, 12 (2021)
22. Serrano, M.A., Boguna, M., Vespignani, A.: Extracting the multiscale backbone of complex weighted networks. *Proc. Natl. Acad. Sci.* **106**, 6483–6488 (2009)
23. Ducruet, C., Rozenblat, C., Zaidi, F.: Ports in multi-level maritime networks: evidence from the Atlantic (1996–2006). *J. Transp. Geogr.* **18**, 508–518 (2010)
24. Dai, L., Derudder, B., Liu, X.: Transport network backbone extraction: a comparison of techniques. *J. Transp. Geogr.* **69** (2018)
25. Yassin, A., Cherifi, H., Seba, H., Togni, O.: Air transport network: a comparison of statistical backbone filtering techniques. In: Cherifi, H., Mantegna, R.N., Rocha, L.M., Cherifi, C., Micciche, S. (eds.) *COMPLEX NETWORKS 2016 2022*. SCI, vol. 1078, pp. 551–564. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-21131-7_43
26. Yassin, A., Cherifi, H., Seba, H., Togni, O.: Exploring statistical backbone filtering techniques in the air transportation network, pp. 1–8. IEEE, Florence, Italy, July 2022
27. Correia, R.B., Barrat, A., Rocha, L.M.: Contact networks have small metric backbones that maintain community structure and are primary transmission subgraphs. *PLOS Comput. Biol.* **19**(2), e1010854 (2023)
28. Foti, N.J., Hughes, J.M., Rockmore, D.N.: Nonparametric sparsification of complex multiscale networks. *PLoS ONE* **6**, e16431 (2011)
29. Slater, P.B.: A two-stage algorithm for extracting the multiscale backbone of complex weighted networks. *Proc. Natl. Acad. Sci.* **106**(26), E66–E66 (2009)
30. Zhang, X., Zhang, Z., Zhao, H., Wang, Q., Zhu, J.: Extracting the globally and locally adaptive backbone of complex networks. *PLoS ONE* **9**(6), e100428 (2014)

31. Yassin, A., Haidar, A., Cherifi, H., Seba, H., Togni, O.: An evaluation tool for backbone extraction techniques in weighted complex networks, May 2023, preprint
32. Toth, D.J.A., et al.: The role of heterogeneity in contact timing and duration in network models of influenza spread in schools. *J. Roy. Soc. Interface* **12**(108), 20150279 (2015)
33. Bastian, M., Heymann, S., Jacomy, M.: Gephi: an open source software for exploring and manipulating networks. In: *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 3, no. 1, pp. 361–362 (2009)
34. 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)



IS-PEW: Identifying Influential Spreaders Using Potential Edge Weight in Complex Networks

Suman Nandi¹(✉), Mariana Curado Malta², Giridhar Maji³,
and Animesh Dutta¹

¹ National Institute of Technology Durgapur, West Bengal, India
sumaninandi1@gmail.com, animesh@cse.nitdgp.ac.in

² CEOS.PP - Polytechnic University of Porto, Porto, Portugal
mariana@iscap.ipp.pt

³ Asansol Polytechnic, West Bengal, India

Abstract. Identifying the influential spreaders in complex networks has emerged as an important research challenge to control the spread of (mis)information or infectious diseases. Researchers have proposed many centrality measures to identify the influential nodes (spreaders) in the past few years. Still, most of them have not considered the importance of the edges in unweighted networks. To address this issue, we propose a novel centrality measure to identify the spreading ability of the Influential Spreaders using the Potential Edge Weight method (IS-PEW). Considering the connectivity structure, the ability of information exchange, and the importance of neighbouring nodes, we measure the potential edge weight. The ranking similarity of spreaders identified by IS-PEW and the baseline centrality methods are compared with the *Susceptible-Infectious-Recovered* (SIR) epidemic simulator using Kendall's rank correlation. The spreading ability of the top-ranking spreaders is also compared for five different percentages of top-ranking node sets using six different real networks.

Keywords: Complex Networks · Centrality measure · Information exchange · Potential edge weight · Influential spreaders

1 Introduction

Due to the rapid progress of network science in recent decades, many real-world applications are modelled into complex networks. There are some particular nodes in every complex network by which we can explore the structural properties of the entire network [12]. Identifying the specific nodes from the networks is one of the most critical research domains to control the spread of any information or infection. The important nodes (influential spreaders) can help to control network attacks, block rumour spreading, prevent infectious diseases (like COVID-19), advertise new products, and many other fields [20, 23].

In recent years, [9, 12, 21] proposed several centrality measures based on many network properties. Among them, based on the topological location of the nodes, there are four types of centrality measures: local centrality, global centrality, semi-global centrality, and hybrid centrality [23]. In the local centrality measure, the local information is measured considering the nearest neighbouring information. Some examples of local centrality measures are degree [2] and cluster rank [5]. In global centrality, the information of a node is measured globally i.e. considering all other nodes in the networks. Some examples are Betweenness [4], Closeness [18]. Some authors measure semi-global centrality to minimize the disadvantages of local and global centrality based on some level of neighbouring information. Some examples of semi-global centrality are: the neighbourhood coeness method (CNC) [1], Global Local Structure (GLS) [19], and many others [9, 15]. Combining more than one centrality measure, the authors proposed the hybrid centrality methods, e.g. GSM [21] and many others [23].

Apart from the topological information, we observe that distance is also an important parameter to measure the connection strength between the nodes, which is inversely proportional to measuring the importance between the nodes [16]. In almost all existing studies, the distance between the nodes is considered a static parameter [12]. In the case of real-time networks, the strength between the nodes may not always remain the same, like friendship in social networks, protein-protein interaction, and many others [3]. To address this issue, we measure the strength between the nodes in terms of distance, considering the ability of information exchange between them. The maximum capacity of information exchange between nodes implies the shortest distance between nodes, i.e. nodes are closer to each other.

To measure the spreading potentiality of nodes, we also explore the structural connectivity between nodes. Inspired by the cluster rank [5], we feel that whenever the neighbouring nodes are connected, the information (or infection) spreading potentiality increases. In this regard, we observe that whenever a triangle structure is formed with the neighbouring nodes, the information propagation capabilities between them increase accordingly [10]. Hence, to understand the structural connectivity properties of any nodes, the triangle structural percentage between nodes is also considered while measuring node importance.

The information or infection propagation from an influential node is done by its neighbouring node. Hence, the influential ability of a spreader is also dependent on its neighbouring nodes [9, 15]. Whenever a node is connected to more important neighbouring nodes, the influence ability of the nodes is increased accordingly. To efficiently identify the neighbouring importance, we measure the importance of neighbouring nodes by considering their topological information. Where the topological information of the neighbouring nodes is measured locally and globally to identify the neighbouring importance.

This paper proposes a new centrality method to rank the Influential Spreaders using Potential Edge Weight (IP-PEW). The contributions of this paper are as follows:

- A novel centrality measure by calculating potential edge weight between nodes by aggregating the connectivity structure, ability of information exchange, and the neighbouring importance, for the undirected and unweighted networks.
- Introduce a new parameter “ability of information exchange”, to measure the strength between the nodes in terms of distance.
- The novel measure can efficiently identify the spreading ability of every node without any extra tunable parameters.

The rest of the paper is as follows: The following section presents existing centrality methods - considered baseline centrality methods - to compare the efficiency of the IS-PEW method. Section 3 describes the working principle of the IS-PEW method to measure the potential edge weight and subsequently identify the spreading ability of every node. Section 4 presents the experimental setup that aims to measure the performance of IS-PEW. Section 5 presents and discusses the experiments’ results. Finally, the last section concludes the paper.

2 Baseline Centrality Methods

In a graph, $G = (V, E)$, V represents the set of nodes connected through a set of edges E . The connectivity structure between any two nodes says v_a and v_b described by the adjacency matrix is $A_{ab} \in V \times V$. Where $A_{ab}=1$ indicates their exist an edge between two nodes v_a and v_b , $A_{ab}=0$ otherwise.

In this section, We present the benchmark methods for identifying the influence spreaders from the network.

Degree Centrality (DG): The degree of a node v_a is the sum of the number of immediate neighbouring nodes [2]. Due to less complexity, the degree centrality is widely used to measure the local importance of any node.

Betweenness Centrality (BC): For a node v_a , betweenness centrality implies the total number of the shortest path between the pair of nodes via the node v_a [4].

Closeness Centrality (CC): Measures the node influence based on the path information. It is calculated by taking the reciprocal of the sum of minimum path distance from node v_a to the remaining nodes in the network [18].

Kshell Decomposition (KS): Considering the topological structure of all the nodes, the kshell decomposition method assigns the kshell index to the nodes [7]. In the initial phase, all nodes with degree value one are removed and assigned the kshell index as one. The process of removing one-degree nodes continues until there is no more than 1-degree node. After that, the algorithm removes those nodes having degree value two and assigns kshell index as two. The process of removing nodes and assigning the kshell index value accordingly is continued until any node is left in the network.

Global Local Structure (GLS): It is calculated by considering global as well as local structural information, where the global structural information is

measured by the closeness of other nodes whereas local structural information is measured by the degree and the contribution probability [19].

Global Structure Model (GSM): The authors [21] proposed an indexing method to measure the self-influence and global-influence of every node. The GSM calculates the global importance considering the contribution of the neighbouring nodes, where the distance between nodes has been measured using the Dijkstra algorithm.

Aggregating Local Structure Information (ALSI): Utilizing the degree and kshell method, the authors [22] proposed the ALSI method by aggregating the own influence ability of a node and the neighbouring contribution.

2.1 Research Motivation

After studying several existing works, it is shown that the importance of a node is closely related to the spreading ability of connected edges, where the spreading ability of the edge is measured by its potential edge weight. In unweighted and undirected networks, the potential edge weight depends on the connected nodes. Most of the studies have considered all edges to be equal in importance. In real-life applications, the edge weight represents the relationship between the connected edges, which may not be the same for all edges. Inspired by this idea, we measure the potential edge weight between every node pair in the undirected and unweighted networks.

3 Proposed Method: IS-PEW

We present a novel centrality measure to identify the influential spreaders based on the potential edge weight. We calculate the potential edge weight of two adjacent nodes by their connectivity structure, the ability of information exchange, and the neighbouring importance which are discussed in the following subsections.

3.1 Connectivity Structure

In a network, when the nodes form any triangular structure, means the neighbouring nodes are closer to each other [10]. A node with a larger number of triangular structures between its neighbours implies the neighbour nodes can also easily propagate any information or infection to other neighbour nodes. The node having the maximum percentage of the triangular structure between the neighbouring nodes is considered a great influential node [15]. Based on this idea, we identify the percentage of the triangular structure $TP(v_A)$ of node v_A by the equation (1).

$$TP(v_A) = \frac{NTC(v_A)}{TC} \quad (1)$$

where $NTC(v_A)$ is the number of triangular structures of node v_A with its neighbours, and TC is the total number of triangular structures in the network.

3.2 Ability of Information Exchange

Distance is the most important parameter while measuring the interaction between the nodes. Since the information propagation capability of every node is not the same, we consider the strength between the nodes in terms of distance. Concerning real networks, the interaction between the nodes may differ, e.g. friendship in social media, protein-protein interaction in a biological network, and many others. Inspired by this idea, we calculate the distance considering the ability of information exchange between two connected nodes. For an edge (v_A, v_B) , we calculate the information exchange ability $InfE(v_A, v_B)$ between the node v_A with its connected nodes v_B by equation (2).

$$InfE(v_A, v_B) = \frac{DG(v_A) * DG(v_B)}{1 + \sum_{k \in \eta(v_A \cap v_B)} DG(k)} \quad (2)$$

where k denotes the common neighbours of two connected nodes v_A and v_B ($v_A \neq v_B$). The equation (2) signifies that the larger value of the degree centrality of two connected nodes enhances their ability to information exchange. Similarly, for two different connected nodes, if they have more common neighbours with large degree values, then the ability for information exchange between those two nodes is minimized. During information propagation between two nodes, a greater number of common neighbours with large degree values can spread the information to other nodes as well, which decreases their ability to information exchange (i.e. increases their distance). Hence the ability of information exchange between two connected nodes is inversely proportional to their distance i.e. $ED(v_A, v_B) = \frac{1}{InfE(v_A, v_B)}$.

3.3 Importance of Neighbouring Nodes

The spreading ability of nodes mainly depends upon the neighbouring nodes. When a node is connected to more important nodes, then the influence potentiality of that node is increased. We calculate the importance of neighbouring nodes by considering their local importance using degree (DG) and global importance using kshell (KS). The degree and kshell decomposition methods are used due to their lower computational complexity than other centrality methods. We calculate the importance of neighbouring nodes of node v_A by equation (3).

$$NIP_{v_A} = \sum_{v_a \in \eta(v_A)} \sqrt{DG(v_a) * KS(v_a)} \quad (3)$$

where $\eta(v_A)$ is the neighbouring nodes of v_A , denoted by v_a .

3.4 Calculation of the Influential Spreaders

Based on the parameters described in Sect. 3.1, 3.2, and 3.3, we calculate the potential edge weight (described in the following subsection). After that, we measure the influential ability of an arbitrary node based on the incident edges with their calculated potential edge weight.

Calculate Potential Edge Weight. We calculate the potential edge weight between two nodes (v_A, v_B) by aggregating the connectivity structure (explore the hidden topological structure), the ability of information exchange (measure strength between the nodes in terms of distance), and the neighbouring importance (measure the spreading ability) together.

The potential edge weight between two nodes $EW_{(v_A, v_B)}$ is calculated by the equation (4).

$$EW_{(v_A, v_B)} = \left[\frac{KS(v_A) * (1 + TP(v_A))}{ED(v_A, v_B)} * NIP_{v_A} \right] + \left[\frac{KS(v_B) * (1 + TP(v_B))}{ED(v_B, v_A)} * NIP_{v_B} \right] \quad (4)$$

where $KS(v_A)$ and $KS(v_B)$ are the used to measure the coreness of the nodes v_A and v_B using kshell index. The equation (4) implies the edge weight between two adjacent nodes is proportional to their connectivity structural (described in 3.1, and coreness 2), and neighbouring importance (described in 3.3) and also inversely proportional to their distance (described in 3.2). Considering two connected nodes, whenever there is a very few numbers of common neighbours with lesser degree value, then their ability to information exchange is high (i.e. distance between those nodes is low) which enhances the potentiality of edge weight between them.

Identify the Influential Spreaders. Finally, influential spreaders are identified by calculating the spreading ability of every node (IS_{v_A}) considering the calculated potential edge weight of the adjacent edges, shown in equation (5).

$$IS_{v_A} = \sum_{v_B \in \eta(v_A)} EW_{(v_A, v_B)} \quad (5)$$

where the number of adjacent edges of v_A (i.e. directed neighbours of v_A) is denoted by $\eta(v_A)$.

3.5 Algorithm Details

Computational steps of IS-PEW are shown in Algorithm 1.

3.6 Computational Complexity

We first calculate the degree and kshell index of the nodes, so the time complexity is $O(V)$ and $O(E)$, respectively. The complexity of computing the percentage of triangular structure is $O(E)$. Similarly, the computational complexity to measure the ability of information exchange is $O(E) + O(V)$. To calculate the importance of neighbouring nodes, the required time is $O(\langle k \rangle^2)$, where k is the average degree of the nodes. Hence, the time complexity of IS-PEW is $(O(E) + O(E) + O(V) + O(\langle k \rangle^2))$, where for the large networks the average degree $\langle k \rangle$ is very small compared to the number of nodes V . Therefore the computational complexity of the IS-PEW methods is low and can be used to identify the influential spreaders for large networks.

Algorithm 1: Influential Spreaders using Potential Edge Weight

Input: A input graph $G = (V, E)$
Output: G with potential edge weight and the ranking of the nodes

```

begin
  for each edge( $v_A, v_B$ ) in graph  $G$  do
    Measure degree of node  $v_A$  and  $v_B$ 
    Measure kshell of node  $v_A$  and  $v_B$ 
    Measure connectivity structure of node  $v_A$  and  $v_B$  using equation (1)
    Measure ability of information exchange between node  $v_A$  and  $v_B$  using
      equation (2)
    for each neighbouring nodes  $v_a$  of  $v_A$  and  $v_b$  of  $v_B$  do
      Measure importance of neighbouring nodes of  $v_A$  and  $v_B$  using
        equation (3)
    end
    Measure potential edge weight between  $v_A$  and  $v_B$  using equation (4)
  end
  for each node ( $v_A$ ) in graph  $G$  do
    Identify influential spreaders by equation (5)
  end
end
    
```

4 Methodology to Evaluate IS-PEW

To measure the efficiency of IS-PEW, we run two experiments by considering the SIR epidemic simulator, and Kendall’s tau, using six real-time networks (see 4.1). The next sections present details of this evaluation.

4.1 Dataset Description

We use six different real-time networks (see Table 1).

Table 1. Description of the networks, V: count of vertices, E: count of edges, D: network diameter, $\langle k \rangle$: average degree, β_{th} : SIR simulator epidemic threshold

Network	V	E	D	$\langle k \rangle$	β_{th}	Description of networks
Blogs	1224	16718	8	27.32	0.0123	Multiple blogs connected through the hyperlinks [8]
Hamsterster friendships	1858	12534	17	13.49	0.0221	Friendship between the users of Hamsterster.com [8]
Odlis	2900	16382	9	11.30	0.0139	Hypertext reference resources of a library [17]
ca-GrQc	4158	13422	17	6.46	0.0556	Scientific collaboration network [17]
Dmela	7393	25569	11	6.92	0.0422	Biological network of protein-protein interactions [17]
DBLP	12590	49651	10	7.89	0.0228	Authors of a publication database [8]

4.2 Tools

We describe the essential tools to measure the effectiveness of IS-PEW in the following paragraphs.

SIR Epidemic Simulator: The spreading efficiency of identifying influential nodes is measured by the SIR epidemic simulator [14] which is widely adopted as the benchmark in the domain of complex network analysis. Researchers often evaluate different meta-heuristic techniques by assessing their performance in the context of these SIR rankings [13]. In this propagation model, the nodes of a network can be in any one of the three states: S (Susceptible)- which means the nodes are in a healthy state, i.e., not infected yet, I (Infected)- describes the nodes are infected and also can spread the infection to other nodes and R (Recovered or Removed)- means that nodes have completed their infected state and cannot be infected again. At the beginning of the propagation, all the nodes are in the susceptible state except the seed nodes (infected state). After each iteration, the seed nodes may infect the neighbouring susceptible nodes with probability β . Thereafter the previously infected nodes become *recovered* or *removed* with probability λ . The process will run again and again until there remain no nodes in the infected state. The epidemic threshold β_{th} represents in the experiment the division of the average degree by the second-degree average on the network. The λ value is considered one which means the recovered nodes cannot be infected again. As this is a stochastic model, every simulation runs a large number of times (1,000 times for the networks having up to 5,000 nodes, and 100 times for the networks having more than 5,000 nodes), and the average number of recovered nodes is reported as the SIR measure for the nodes.

Kendall's tau (τ) Correlation : To measure the correlation between the two ranks list of the centrality measures, we use Kendall's τ method [6]. The τ value close to one (1) means the ranking lists are very similar and the value close to minus one (-1) means the ranking lists are dissimilar. The τ is calculated by Eq. (6):

$$\tau = \frac{(N_1 - N_2)}{0.5 * N * (N - 1)} \quad (6)$$

where the number of concordant and discordant pairs are represented as N_1 and N_2 , and N represents the number of nodes present on that network.

4.3 Experiments

We perform two experiments to measure the performance of the IS-PEW.

Experiment 1: To calculate the correlation for different infection probabilities: We compare the ranking correlation identified by IS-PEW and the baseline centrality method (described in Sect. 2) with the SIR epidemic simulator (described in Sect. 4.2) under different infection probability (β) using the Kendall tau method (described in Sect. 4.2). We take motivation from [11] and consider the range of infection probability β between β_{th} to $2 * \beta_{th}$ with 5% increment in every step.

Experiment 2: To calculate the correlation for different percentages of node sets: We compare the ranking correlation identified by the IS-PEW

method and the baseline centrality methods (see Sect. 2) with the SIR epidemic simulator (described in Sect. 4.2) under different percentages of top ranking node sets ($P = (0.04, 0.08, 0.12, 0.16, 0.20)$) using the Kendall's τ method (described in Sect. 4.2). Inspired by [23], in this experiment, we consider the β (i.e. infection probability) value as $\beta_{th} + 0.001$.

5 Results and Analysis

The correlation results for different infection probabilities (six different real networks - Experiment 1) are shown in Fig. 1. In each sub-figure, the x-axis indicates the different infection probability (β), and the y-axis indicates the percentage of ranking correlation with the SIR simulator.

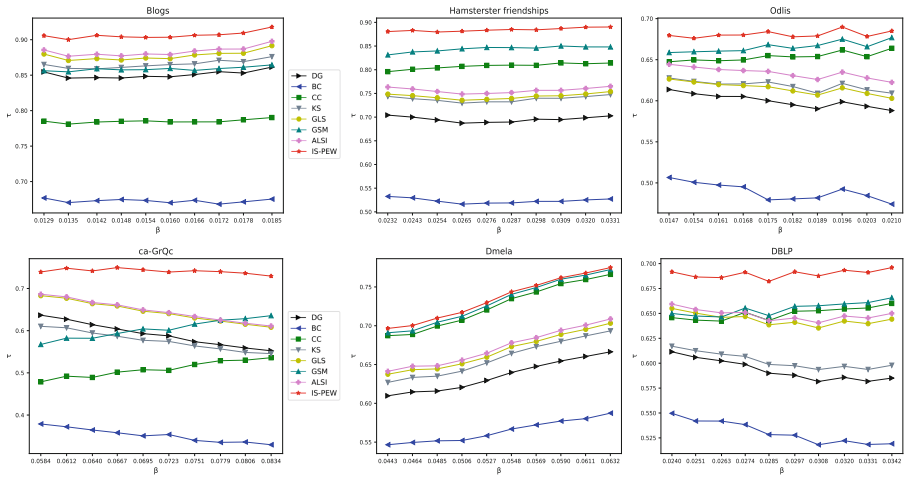


Fig. 1. Kendall's rank correlation (τ) between ranking generated by different centrality measures and SIR ranking

Figure 1 shows that for the Blogs dataset, the correlation between the SIR simulator and the IS-PEW is maximum and has the highest average correlation value of 91.1% compared to the baseline centrality methods. Considering the Hamsterster friendships dataset, the correlation between the SIR simulator and the IS-PEW is maximum and has the highest average correlation value of 88.9% compared to the baseline centrality methods. Considering the Odlis dataset, the correlation between the SIR simulator and the IS-PEW is maximum and the has highest average correlation value of 68.7% compared to the baseline centrality methods. Considering the ca-GrQc dataset, the correlation between the SIR simulator and the IS-PEW is maximum and has the highest average correlation value of 74.8% compared to the baseline centrality methods. Considering the Dmela dataset, the correlation between the SIR simulator and the IS-PEW is

maximum and has the highest average correlation value of 73.9% compared to the baseline centrality methods. Considering the DBLP dataset, the correlation between the SIR simulator and the IS-PEW is maximum and has the highest average correlation value of 69.4% compared to the baseline centrality methods.

The correlation results for different percentages of node-set (for the six real networks - Experiment 2) are summarised in Table 2. The first column is the percentage of top-ranking node sets, and the correlation percentage for different centrality methods with the SIR simulator is in the subsequent columns.

Table 2. Correlation value of centrality measures with SIR simulator considering different percentages of top-ranking node sets

Blogs									Hamsterster friendships									Odlis								
P	DG	BC	CC	KS	GLS	GSM	ALSI	IS-PEW	DG	BC	CC	KS	GLS	GSM	ALSI	IS-PEW	DG	BC	CC	KS	GLS	GSM	ALSI	IS-PEW		
0.04	0.83	0.54	0.6	0.65	0.92	0.85	0.83	0.92	0.78	0.50	0.75	0.66	0.79	0.89	0.79	0.96	0.78	0.71	0.66	0.67	0.78	0.72	0.79	0.84		
0.08	0.78	0.60	0.64	0.66	0.85	0.81	0.81	0.94	0.80	0.85	0.80	0.88	0.81	0.87	0.83	0.92	0.72	0.56	0.66	0.71	0.74	0.74	0.77	0.85		
0.12	0.86	0.63	0.67	0.77	0.92	0.86	0.89	0.97	0.79	0.59	0.82	0.89	0.80	0.87	0.82	0.91	0.68	0.53	0.70	0.72	0.71	0.76	0.73	0.83		
0.16	0.89	0.69	0.71	0.90	0.92	0.85	0.90	0.95	0.82	0.61	0.84	0.90	0.82	0.89	0.84	0.90	0.70	0.49	0.71	0.74	0.70	0.76	0.74	0.80		
0.20	0.89	0.70	0.73	0.95	0.91	0.84	0.91	0.95	0.80	0.62	0.85	0.87	0.80	0.90	0.84	0.93	0.70	0.45	0.75	0.73	0.71	0.78	0.74	0.83		
ca-GrQc									Dmela									DBLP								
P	DG	BC	CC	KS	GLS	GSM	ALSI	IS-PEW	DG	BC	CC	KS	GLS	GSM	ALSI	IS-PEW	DG	BC	CC	KS	GLS	GSM	ALSI	IS-PEW		
0.04	0.80	0.16	0.40	0.74	0.80	0.49	0.82	0.83	0.68	0.56	0.69	0.72	0.68	0.72	0.70	0.79	0.56	0.38	0.61	0.52	0.57	0.63	0.58	0.65		
0.08	0.64	0.21	0.42	0.65	0.67	0.68	0.68	0.83	0.72	0.64	0.75	0.80	0.73	0.78	0.74	0.81	0.59	0.41	0.67	0.63	0.60	0.69	0.61	0.70		
0.12	0.65	0.27	0.44	0.66	0.66	0.70	0.66	0.85	0.75	0.66	0.80	0.80	0.75	0.81	0.76	0.82	0.65	0.47	0.69	0.70	0.66	0.70	0.66	0.74		
0.16	0.65	0.34	0.52	0.66	0.66	0.72	0.68	0.85	0.75	0.68	0.80	0.79	0.75	0.80	0.76	0.81	0.69	0.54	0.71	0.73	0.69	0.72	0.70	0.78		
0.20	0.69	0.41	0.58	0.70	0.69	0.72	0.71	0.84	0.76	0.68	0.80	0.81	0.76	0.79	0.77	0.81	0.72	0.61	0.76	0.75	0.72	0.77	0.73	0.80		

Table 2 shows that for the Blogs dataset, IS-PEW is most correlated with the SIR simulator and also has a maximum correlation value of 97% (by considering $P = 0.12$) with the SIR simulator compared to baseline centrality methods. Considering the Hamsterster friendships dataset, IS-PEW is most correlated with the SIR simulator and also has a maximum correlation value of 96% (by considering $P = 0.04$) with the SIR simulator compared to baseline centrality methods. Considering the Odlis dataset, IS-PEW is most correlated with the SIR simulator and also has a maximum correlation value of 85% (by considering $P = 0.08$) with the SIR simulator compared to baseline centrality methods. Considering the ca-GrQc dataset, IS-PEW is most correlated with the SIR simulator and also has a maximum correlation value of 85% (by considering $P = 0.12$ and $P = 0.16$) with the SIR simulator compared to baseline centrality methods. Considering the Dmela dataset, IS-PEW is most correlated with the SIR simulator and also has a maximum correlation value of 82% (by considering $P = 0.12$) with the SIR simulator compared to baseline centrality methods. Considering the DBLP dataset, IS-PEW is most correlated with the SIR simulator and also has a maximum correlation value of 80% (by considering $P = 0.20$) with the SIR simulator compared to baseline centrality methods.

We proved that the ranking of IS-PEW is more correlated to the SIR simulator considering different infection probability values and different percentages of top-ranking node sets. Since the SIR simulation score is considered a benchmark, more percentage of correction means that IS-PEW is the best centrality measure compared to baseline centrality methods.

6 Conclusion

This paper presents IS-PEW, a method that addresses the issue of not considering the edge importance of unweighted networks. We calculate the potential edge weight of the edges considering the connectivity structure, the ability of information exchange, and the importance of neighbouring nodes. To explore the hidden connectivity structure, we consider the percentage of the triangular structure of the nodes. Since in real-life networks, the distance between the nodes is not the same, we calculate the strength between the nodes in terms of distance considering the ability of information exchange. During the information spreading, the importance of neighbouring nodes plays a significant role, which is also measured by considering the global measure (kshell decomposition) and the local measure (degree). Finally, the influence ability of every node is measured by combining the associated potential edge weight value. The efficiency of IS-PEW is compared with the SIR epidemic simulator with respect to six real-time networks. The experimental results show that IS-PEW performs better compared to the baseline centrality methods. As future work, we will work on detecting the potentiality of the edges for the time series networks.

Acknowledgements. This work is financed by the Erasmus+ICM (International Credit Mobility) program under the project 2020-1-PT01-KA107-078161. This work is financed by Portuguese national funds through FCT - Fundação para a Ciência e Tecnologia, under the project UIDB/05422/2020.

References

1. Bae, J., Kim, S.: Identifying and ranking influential spreaders in complex networks by neighborhood coreness. *Physica A* **395**, 549–559 (2014)
2. Bonacich, P.: Factoring and weighting approaches to status scores and clique identification. *J. Math. Soc.* **2**(1), 113–120 (1972)
3. Daud, N.N., Ab Hamid, S.H., Saadoon, M., Sahran, F., Anuar, N.B.: Applications of link prediction in social networks: a review. *J. Network Comput. Appl.* **166**, 102,716 (2020)
4. Freeman, L.C.: A set of measures of centrality based on betweenness. *Sociometry* 35–41 (1977)
5. Garg, N., Favre, B., Reidhammer, K., Hakkani Tür, D.: Clusterrank: a graph based method for meeting summarization. Technical report, Idiap (2009)
6. Kendall, M.G.: The treatment of ties in ranking problems. *Biometrika* **33**(3), 239–251 (1945)
7. 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), 888–893 (2010)
8. Kunegis, J.: Konect: the koblenz network collection. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 1343–1350 (2013)
9. Li, Z., Ren, T., Ma, X., Liu, S., Zhang, Y., Zhou, T.: Identifying influential spreaders by gravity model. *Sci. Rep.* **9**(1), 1–7 (2019)
10. Ma, X., Ma, Y.: The local triangle structure centrality method to rank nodes in networks. *Complexity* **2019** (2019)

11. Maji, G., Dutta, A., Malta, M.C., Sen, S.: Identifying and ranking super spreaders in real world complex networks without influence overlap. *Expert Syst. Appl.* **179**, 115,061 (2021)
12. Maji, G., Mandal, S., Sen, S.: A systematic survey on influential spreaders identification in complex networks with a focus on k-shell based techniques. *Expert Syst. Appl.* **161**, 113,681 (2020)
13. Maji, G., Namtirtha, A., Dutta, A., Curado Malta, M.: Influential spreaders identification in complex networks with improved k-shell hybrid method. *Expert Syst. Appl.* **144**, 113,092 (2020)
14. Moreno, Y., Pastor-Satorras, R., Vespignani, A.: Epidemic outbreaks in complex heterogeneous networks. *Eur. Phys. J. B-Condensed Matter Complex Syst.* **26**(4), 521–529 (2002)
15. Namtirtha, A., Dutta, B., Dutta, A.: Semi-global triangular centrality measure for identifying the influential spreaders from undirected complex networks. *Expert Syst. Appl.* **206**, 117,791 (2022)
16. Ohara, K., Saito, K., Kimura, M., Motoda, H.: Accelerating computation of distance based centrality measures for spatial networks. In: Calders, T., Ceci, M., Malerba, D. (eds.) *DS 2016. LNCS (LNAI)*, vol. 9956, pp. 376–391. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46307-0_24
17. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI 2015*, pp. 4292–4293. AAAI Press (2015)
18. Sabidussi, G.: The centrality index of a graph. *Psychometrika* **31**(4), 581–603 (1966)
19. Sheng, J., et al.: Identifying influential nodes in complex networks based on global and local structure. *Physica A: Stat. Mech. Appl.* **541**, 123,262 (2020)
20. Tang, Y., Qian, F., Gao, H., Kurths, J.: Synchronization in complex networks and its application - a survey of recent advances and challenges. *Annu. Rev. Control.* **38**(2), 184–198 (2014)
21. Ullah, A., Wang, B., Sheng, J., Long, J., Khan, N., Sun, Z.: Identification of nodes influence based on global structure model in complex networks. *Sci. Rep.* **11**(1), 1–11 (2021)
22. Wang, F., Sun, Z., Gan, Q., Fan, A., Shi, H., Hu, H.: Influential node identification by aggregating local structure information. *Physica A: Stat. Mech. Appl.* **593**, 126,885 (2022)
23. Zhao, Z., Li, D., Sun, Y., Zhang, R., Liu, J.: Ranking influential spreaders based on both node k-shell and structural hole. *Knowl.-Based Syst.* **260**, 110,163 (2023)



Robustness of Centrality Measures Under Incomplete Data

Natalia Meshcheryakova^(✉)  and Sergey Shvydun 

HSE University, Moscow, Russia
natamesc@gmail.com, shvydun@hse.ru

Abstract. Understanding of real systems relies on the identification of its central elements. Over the years, a large number of centrality measures have been proposed to assess the importance of nodes in complex networks. However, most real networks are incomplete and contain incorrect data, resulting in a high sensitivity of centrality indices. In this paper, we examine the robustness of centrality to the presence of errors in the network structure. Our experiments are performed on weighted and unweighted real-world networks ranging from the criminal network to the trade food network. As a result, we discuss a sensitivity of centrality measures to different data imputation techniques.

Keywords: centrality · incomplete data · perturbation analysis · data imputation

1 Introduction

Many real-world systems, such as infrastructural, biological, brain and social, can be represented as networks, where nodes denote the components and links denote relations or interaction between these components. A fundamental issue concerning the complex systems is to understand the impact of individual nodes on the whole system. However, the notion of importance can be defined in different ways depending on the nature of a network or features that a researcher wants to consider while ranking nodes. Therefore, the researchers have introduced more than 400 centrality measures [1], ranging from classical centralities [2] to the measures that take into account specific features of a network [3–5]. These measures have shown a great value in understanding many real networks, including citation networks, computer networks, and biological networks. In general, centrality measures provide different central elements, consequently, the choice of the most appropriate centrality measure depends on the type of a network and the interpretation of important elements.

In most real networks, however, information about the structure is inaccurate due to presence of errors in the data. For instance, Ficara et al. [6] have examined criminal networks that suffer from data incompleteness (due to the nature of the network), data incorrectness (unintentional data collection errors and intentional deception by criminals) and data inconsistency (misleading information

from different sources). Aleskerov et al. [7] have studied the banking foreign claims network, which covers about 94% of total foreign claims as some countries do not report. Meshcheryakova [8] has investigated the trade network under asymmetry as many countries report their own versions of a trade flow between them (up to $7 \cdot 10^7\%$ difference) due to the different commodity classification systems, the different costs calculation (including/excluding transportation and insurance costs) or the time lag. Therefore, the analysis of centrality in these networks requires a careful examination, because many centrality measures are very sensitive to small changes in the graph structure.

The effects of missing or incorrect data in complex networks have been extensively studied in the literature. Most of the studies focus on the sensitivity of centrality in artificial graph structures such as Erdős-Rényi (ER) random graph, Barabási-Albert (scale-free) graph, Watts-Strogatz (small-world) graph and other classical graph structures [9–14]. These studies are mostly limited to the perturbation analysis of classical centrality measures (degree, eigenvector, betweenness, closeness and PageRank) in the case of 1 structural change (edge/node removal/addition). Moreover, all the changes in the structure are performed at random, which might be meaningless for real-world networks.

Some studies are aimed to examine centrality measures in real-world networks. Bolland [15] has examined the performance of 4 classical centrality measures (random changes) on Chillicothe data. Herland et al. [16] consider 3 classical centrality measures and their robustness to random changes in 4 real networks. Niu et al. [17] have examined the stability of 5 centrality measures (degree, betweenness, closeness, eigenvector, k-shell) on 9 real datasets towards random edge addition/removal/rewiring and have evaluated the Spearman correlation between centrality rankings. Segarra and Ribeiro [11] evaluate the effect of random changes on the air traffic network and the network of interactions between sectors of the US economy. These studies are mostly limited to a very small amount of centralities and to the analysis of random changes in a network.

In this paper, we consider two real-world networks that are incomplete or may contain incorrect data. Our goal is to examine how much the set of central nodes is sensitive to the presence of errors in the graph structure. We consider several data imputation strategies, which take into account the nature of the network, and evaluate the robustness of 13 centrality measures.

The paper is organized as follows. Section 2 provides some basic information about the centrality measures and describes our methodology. In Sect. 3, we examine the perturbation analysis of centrality measure on the real-world networks. Section 4 concludes.

2 Methodology

2.1 Preliminaries

We consider a graph $G = (V, L)$, where $V = \{1, \dots, n\}$ is a set of nodes, $|V| = n$, and $L \subseteq V \times V$ is a set of L links. The graph G is described by an $n \times n$ adjacency matrix A whose elements a_{ij} are either one or zero depending on whether there is

a link between nodes i and j or not and $a_{ii} = 0$ for all $i \in V$. The graph is called undirected if $a_{ij} = a_{ji}$ for all $i, j \in V$ and directed, otherwise. Additionally, the graph G can be described by a non-negative weight matrix W , where each element w_{ij} represents the weight of a link between nodes i and j and $w_{ii} = 0$ for all $i \in V$. Given a graph G , a centrality measure $c(\cdot)$ associates a real number $c(i)$ to each node $i \in V$, which is interpreted as follows: the larger $c(i)$, the more central node i should be.

2.2 Centrality Measures

In general, the identification of central elements in a network is an ill-defined problem. Thus, there exist multiple centrality measures that take into account particular aspects of the problem. Table 1 presents a list of centrality measures, which are applied to real networks from Sect. 3. A detailed description of the centrality measures is provided in [2, 5, 23].

Table 1. Centrality Measures

#	Centrality	Description
1	Degree	the number of node neighbours
2	Eigenvector	the importance of a node depends on the importance of its neighbours
3	Katz	the generalization of the eigenvector centrality
4	Betweenness	how often nodes lie on the shortest paths between other nodes
5	Closeness	the inverse of the total distance to other nodes
6	Harmonic	the sum of inverse distances to other nodes
7	Subgraph	the number of closed walks of different length in a graph
8	PageRank	the probability to visit nodes by random walks
9	HITS	<i>hub</i> : the node has a high score if it links to many authorities, <i>authority</i> : the node has a high score if it is pointed by many hubs
10	LRIC	the centrality depends on individual attributes of nodes, their group and indirect influence
11	Laplacian	the drop in the Laplacian energy after deleting a node from the graph
12	k-shell	the nodes in the k -core that are not in the $(k + 1)$ -core
13	Collective Influence	the centrality is proportional to the degree of a node and degrees of its neighbours at a particular distance

We remark that centrality measures, which are based on the paths (betweenness, closeness, harmonic and subgraph centralities), as well as the eigenvector, Katz, Laplacian, k-shell and Collective Influence (CollInf) centralities are computed only for the unweighted network. Similarly, we apply 4 versions of

weighted degree centrality (inDegree, outDegree, Degree = inDegree + outDegree, DegreeDiff = outDegree - inDegree), Hubs and Authorities only to the weighted network.

2.3 Imputation Methods and Performance Analysis

The effects of missing data is hard to estimate as there might be multiple sources of the errors. Therefore, the perturbation analysis depends on the data structure, the nature of a network as well as on the type of errors in the data. Tables 2–3 illustrate the actions, which we have applied to modify the structure of the criminal (unweighted, undirected) and the international food trade (weighted, directed) networks. RAE, RAN and RCE consider random modifications of the graph structure. DAE and DAN perform the addition of links with respect to the configuration model where the expected number of edges between two nodes is proportional to the product of their degrees. Finally, PAE is driven by the idea of similarity between nodes, which can be estimated by the shortest distance between them. The presented list of graph changes is not exhaustive. Some other imputation methods are also discussed in [18, 19].

Table 2. List of modifications in the criminal network (unweighted, undirected).

#	Name	Description
1	RAE	random addition of k new links
2	DAE	addition of k new links with a probability that is proportional to the product of the incident nodes degrees
3	PAE	addition k new links with a probability that is inversely proportional to the shortest path distance between nodes
4	RAN	addition of a new node to d random vertices
5	DAN	addition of a new node to d vertices with probability that is proportional to their degrees

Table 3. List of modifications in the food trade network (weighted, directed).

#	Name	Description
1	RCE	random change of link weights in the range of $[-5\%, 5\%]$
2	RAE	addition of links with the total weight s
3	DAE	addition of links with the total weight s with a probability proportional to the product of the source outdegree and the target indegree

Finally, 5 performance metrics are used to assess the stability of centralities:

1. **Correlation:** the Kendall rank correlation coefficient, which measures the similarity of the orderings of centrality measures¹.

¹ In the case of node addition, we assume that the added node was initially isolated.

2. **TOP1**: the percentage of nodes, which remain TOP-1 after modification.
3. **TOP3**: the percentage of nodes, which remain TOP-3 after modification.
4. **TOP5**: the percentage of nodes, which remain TOP-5 after modification.
5. **TOP10**: the percentage of nodes, which remain TOP-10 after modification.

All the performance measures are averaged over $T = 1,000$ modifications in the graph structure. We perform a pairwise comparison of centrality measures and provide their ranking with respect to various data imputation methods. In particular, the ranking of centralities is constructed using the Copeland score, which is a social choice rule that measures the difference between the cardinality of dominating and dominated² sets [20,21]. Compared to the average value, the Copeland score is more stable for ranking objects and is less sensitive to the outliers.

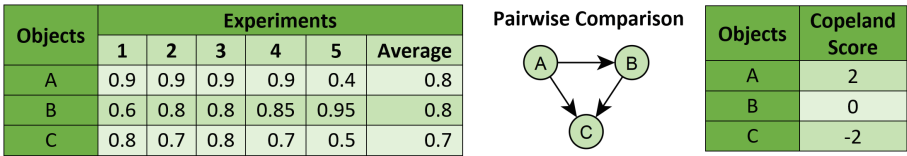


Fig. 1. The difference between the Copeland score and the average value.

Figure 1 shows an example of 3 objects, which are compared by 5 experiments. Objects A and B have the same average value, although A demonstrates a better performance than B in 4 experiments out of 5. On the contrary, A has the highest Copeland score (A is better than B and C) while B is ranked second (B is better than C but B is dominated by A). Since our experiments explore possible realizations of the initial partially-observed graph, we believe that the Copeland score is more reliable for the ranking of centrality measures than the average value.

3 Robustness of Centrality Measures in Real Networks

We consider some real-world networks that suffer from data incompleteness and incorrectness. For each of the networks, we apply various data imputation strategies and evaluate the sensitivity of centrality measures from Sect. 2.2.

3.1 The Analysis of the Criminal Network

The first network refers to Sicilian Mafia interconnections. Ficara et al. [22] collected two datasets of phone calls and personal meetings respectively between members of “criminal Families” in Sicily. The data are derived from Court reports in 2007 based on the results of anti-mafia “Montagna Operation”. Ficara

² The *dominating* set of centrality x includes a list of centrality measures, which are more sensitive than x to the graph modification based on the pairwise comparison. Similarly, the *dominated* set of x contains centralities that are less sensitive than x .

et al. make a remark that such datasets are compiled from judicial documents and suffer from incompleteness. Therefore, the stability analysis of centrality measures toward small graph modifications is reasonable for these networks.

The mafia phone calls network consists of 100 nodes and 124 edges whose weights are integers between 1 and 8. The mafia meetings network consists of 101 nodes and 256 weighted edges with the maximum weight of 10. There are 47 mafia members that are present in both networks. In this paper, we focus on the meetings between mafia members and examine the unweighted mafia network.

The list of graph modifications, which may occur in the Mafia network due to the incompleteness of information, is presented in Table 2. We consider parameter $k = 5\%$ of the total number of edges³ in a graph and assume that parameter d is equal to the average degree in a graph. We do not take into account the deletion of nodes and edges as we suppose that all presented actors and connections have been identified correctly. We consider all the centralities measures from Table 1 except for the HITS. We remark that the collective influence (*CollInf*) is proposed in [23] to analyze influential members in criminal networks.

Table 4. The ranking of centralities by the Copeland score (addition of 5% edges).

Centrality	RAE					DAE					PAE				
	Correlation	TOP1	TOP3	TOP5	TOP10	Correlation	TOP1	TOP3	TOP5	TOP10	Correlation	TOP1	TOP3	TOP5	TOP10
Degree	3	1-10	11	2	5	1	1-4	10	2	4	2	1-8	11	2	5
Eigenvector	6	1-10	7	9	1	7	10	4	9	1	6	9-10	7	11	1
Katz	4	11	2	5	2	5	11	2	5	2	5	11	2	5	2
Betweenness	12	1-10	12	10	12	12	8	11	10	11	12	1-8	12	9	12
Closeness	9	12	6	8	11	11	12	6	7	12	10	12	5	8	11
Harmonic	8	1-10	4	7	8	10	7	5	8	9	7	1-8	4	6	8
Subgraph	2	1-10	3	6	4	4	9	3	6	3	3	9-10	3	7	4
PageRank	10	1-10	8	4	9	6	1-4	8	3	5	8	1-8	8	4	9
LRIC	11	1-10	10	12	10	9	6	9	11	8	11	1-8	9	12	10
Laplacian	5	1-10	5	3	7	3	5	7	4	6	4	1-8	6	3	7
K-shell	1	1-10	1	1	3	2	1-4	1	1	7	1	1-8	1	1	3
CollInf	7	1-10	9	11	6	8	1-4	12	12	10	9	1-8	10	10	6
minValue	.78	.99	.88	.84	.88	.85	.90	.75	.77	.83	.82	.99	.89	.85	.89
maxValue	.96	1	1	1	.99	.97	1	1	1	.99	.96	1	1	1	0.99

The results for RAE, DAE and PAE graph modifications are presented in Table 4. First, most centrality measures are equally stable in the context of TOP1 nodes within RAE and PAE graph modifications. We also observe that

³ We have also performed the analysis for $k = 10\%$ and the overall results are highly agreed with $k = 5\%$, even though the centrality measures are less stable.

the relative position of centrality measures by the TOP10 metric is identical for RAE and PAE scenarios with a little difference for the DAE scenario. Second, the k -shell centrality demonstrates the most stable results for all edge addition scenarios. For instance, TOP1 includes 7 nodes, that are not changed for all considered modifications. On the contrary, the degree, the Katz and the subgraph centralities are the most sensitive indices in case of the missing edges in the network. Next, the degree centrality provides more stable results within DAE than for RAE and PAE. Under the DAE scenario, we are more likely to add new edge between nodes that have high degree scores, which only strengthen their degree centrality. Similarly, the harmonic centrality is more stable within PAE than in RAE and DAE scenarios as the shortest path distances do not change dramatically. Interestingly, the closeness centrality is more sensitive to the edge addition than the closeness centrality. Overall, the betweenness, the closeness and the LRIC are the most sensitive to the edge addition.

The results for the node addition scenarios are provided in Table 5. In general, we observe a high correlation coefficient (> 0.92) for all the centrality measures. Hence, all the centralities are relatively stable toward the addition of a new node. Similarly, the node addition also does not change the TOP1 in the graph for all centrality measures. As for the relative position of centrality measures, the subgraph centrality is the most stable according to the correlation coefficient. The k -shell centrality is stable in the context of all discussed TOP metrics. On the other hand, the LRIC, the collective influence, the betweenness and the closeness centralities are the most sensitive measures compared to other indices.

Table 5. The ranking of centralities by the Copeland score (node addition).

Centrality	RAN					DAN				
	Correlation	TOP1	TOP3	TOP5	TOP10	Correlation	TOP1	TOP3	TOP5	TOP10
Degree	6	1-12	12	1-3	6	3	1-12	12	1-3	5
Eigenvector	2	1-12	10	8	1-3	5	1-12	11	10	1-3
Katz	3	1-12	1-5	5	1-3	6	1-12	1-5	7	1-3
Betweenness	11	1-12	8	10	12	9	1-12	8	9	11
Closeness	10	1-12	6	9	11	11	1-12	1-5	8	12
Harmonic	8	1-12	1-5	7	9	10	1-12	1-5	6	8
Subgraph	1	1-12	1-5	6	4-5	1	1-12	6	5	6
PageRank	9	1-12	7	4	7	7	1-12	7	4	7
LRIC	12	1-12	9	11	10	12	1-12	9	11	10
Laplacian	5	1-12	1-5	1-3	8	4	1-12	1-5	1-3	9
Kshell	4	1-12	1-5	1-3	1-3	2	1-12	1-5	1-3	1-3
CollInf	7	1-12	11	12	4-5	8	1-12	10	12	4
minValue	.92	1	.97	.90	.95	.93	1	.93	.89	.92
maxValue	.98	1	1	1	1	.97	1	1	1	1

3.2 The Analysis of Food Trade Network

The global trade process is a major part of international relations. We study the network of trade between countries. In particular, we consider the international trade of cereals.

In order to construct a directed weighted network, we address to the World Integrated Trade Solution (WITS) database [24], where bilateral trade statistics are provided. We use data that are reported by importers only. Still, some information is lost as, first, not every country reports its statistics and, second, export and import statistics for a particular flow may differ in many times [8]. Overall, we obtain a directed weighted network that represents 222 countries and 9,384 trade flows between them in 2020. The largest value of a flow is equal to 5,426,071.14 thousand dollars from Canada to USA.

We consider some reasonable graph modifications in order to evaluate the stability of centrality measures (see Table 3). We assume that the initial graph covers around 99% of the total trade, consequently, 1% of trade is added with respect to RAE and DAE scenarios.

Table 6. The ranking of centralities by the Copeland score for the trade network.

Centrality	RCE					RAE					DAE				
	Correlation	TOP1	TOP3	TOP5	TOP10	Correlation	TOP1	TOP3	TOP5	TOP10	Correlation	TOP1	TOP3	TOP5	TOP10
outDegree	2	1-7	1-6	1-5	1-6	4	1-4	2	2	1	1	5	3	2	1
inDegree	4	1-7	1-6	8	7	7	8	8	7	7	6	8	8	8	8
Degree	1	1-7	1-6	1-5	1-6	3	1-4	5	5	5	4	4	5	4	4
DegreeDiff	8	8	8	1-5	1-6	5	1-4	3	1	3	8	3	2	1	2
PageRank	3	1-7	1-6	6	1-6	8	7	7	8	8	5	7	7	6	6
Hubs	6	1-7	1-6	1-5	1-6	1	1-4	1	3	4	2	2	1	3	3
Authorities	7	1-7	7	1-5	1-6	6	6	6	6	6	7	6	6	7	7
LRIC	5	1-7	1-6	7	8	2	5	4	4	2	3	1	4	5	5
minValue	.99	.96	.98	.95	.98	-.12	.07	.07	.07	.09	.14	.07	.07	.09	.22
maxValue	.99	1	1	1	1	.54	1	.93	1	.91	.79	1	.93	1	.92

The Copeland scores of eight centrality measures are provided in Table 6. First, the random change of edge weights (RCE, $\pm 5\%$) does not significantly change the centrality of nodes in the network. Second, the random addition of new links (RAE, 1%) highly influences the set of central elements. The highest stability is observed for the hub score: TOP1 and TOP3 nodes remain the same in more than 93% of cases while the Kendall rank correlation is moderate (0.54). As to the TOP5 and TOP10 nodes, the highest stability is for the outDegree

centrality ($\geq 84\%$). Overall, outDegree, Degree, Hubs and LRIC measures are the only centralities with a correlation coefficient, which is greater than 0.37, and a higher sustainability of the most central nodes. Finally, inDegree, PageRank and Authorities are the most sensitive to the presence of missing links.

The addition of new links with respect to the node degree (DAE, 1%) also affects all the centralities. OutDegree and Hubs provide the most stable results. These measures provide a strong correlation coefficient (≥ 0.78) and a stable set of central elements (TOP1-TOP10 $\geq 70\%$). In fact, the most central node remains the same under the DAE scenario for all centrality measures except for the inDegree, the Authorities and the PageRank. Interestingly, DegreeDiff is stable for TOP1-TOP10 nodes ($\geq 70\%$) but has a very weak correlation coefficient (≈ 0.14). On the contrary, the overall ranking of nodes with respect to the LRIC score is stable (correlation ≈ 0.71 , TOP1 $\approx 100\%$), however, the TOP3-TOP10 nodes remain the same only in 40–50% of cases. We also remark that inDegree and Authorities are the least stable with respect to the DAE scenario.

4 Discussion

Data incompleteness and incorrectness is a serious challenge to the analysis of real systems. In this regard, the set of the most central elements in the system requires a careful examination. In this paper, we have examined how the presence of missing or incorrect links affect the results of 13 existing centrality measures in the criminal (unweighted) and the food trade (weighted) networks. Our main observation for the unweighted network is that the addition of new edges influences the stability of centrality measures more than the addition of a new node. Overall, there is no evidence of considerable changes in most centrality scores (except for the betweenness, closeness and LRIC indices) under all discussed modifications in the criminal network. For the weighted network, we observe that 5% inaccuracy in edge weights does not significantly affect the centrality of the nodes while the presence of missing links may dramatically influence the results of some centrality measures (e.g.: inDegree, PageRank and Authorities).

The results on the robustness of centrality measures are only valid for the criminal network and the trade food network. To draw meaningful and robust conclusions for partially-observed networks, more experiments on a large set of different benchmark network topologies should be performed.

We would like to emphasize that our work is not intended to demonstrate the deficiency of some centrality indices but to show that some centralities require a cautious interpretation in the presence of missing or incorrect data.

Acknowledgment. The article was prepared within the framework of the HSE University Basic Research Program and funded by the Russian Academic Excellence Project ‘5–100’. The analysis of centrality measures (Sects. 2–3) was supported by grant No. MK-3867.2022.1.6.

References

1. Centiserver: The most comprehensive centrality resource and web application for centrality measures calculation (2023). <https://www.centiserver.org/centrality/list/>. Accessed 1 Jul 2023
2. Newman, M.E.J.: *Networks: An Introduction*. Oxford University Press, Oxford (2010). <https://doi.org/10.1093/acprof:oso/9780199206650.001.0001>
3. Myerson, R.B.: Graphs and cooperation games. *Math. Oper. Res.* **2**, 225–229 (1977). <https://doi.org/10.1287/moor.2.3.225>
4. Kang, C., Molinaro, C., Kraus, S., Shavitt, Y., Subrahmanian, V.S.: Diffusion centrality in social networks. In: *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 558–664, Istanbul (2012). <https://doi.org/10.1109/ASONAM.2012.95>
5. Aleskerov, F., Shvydun, S., Meshcheryakova, N.: *New Centrality Measures in Networks: How to Take into Account the Parameters of the Nodes and Group Influence of Nodes to Nodes (1st ed.)*. Chapman and Hall/CRC (2021). <https://doi.org/10.1201/9781003203421>
6. Ficara, A., et al.: Criminal networks analysis in missing data scenarios through graph distances. *PLoS ONE* **16**(8), e0255067 (2021). <https://doi.org/10.1371/journal.pone.0255067>
7. Aleskerov, F., Andrievskaya, I., Nikitina, A., Shvydun, S.: Key Borrowers Detected by the Intensities of Their Interactions. *Handbook of Financial Econometrics, Mathematics, Statistics, and Machine Learning (In 4 Volumes)*, 355–389 World Scientific: Singapore Volume 1, Chapter 9 (2020). https://doi.org/10.1142/9789811202391_0009
8. Meshcheryakova, N.: Network analysis of bilateral trade data under asymmetry. In: *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, The Hague, Netherlands, pp. 379–383 (2020). <https://doi.org/10.1109/ASONAM49781.2020.9381408>
9. Borgatti, S.P., Carley, K.M., Krackhardt, D.: On the robustness of centrality measures under conditions of imperfect data. *Soc. Networks* **28**(2), 124–136 (2006). <https://doi.org/10.1016/j.socnet.2005.05.001>
10. Frantz, T.L., Cataldo, M., Carley, K.M.: Robustness of centrality measures under uncertainty: Examining the role of network topology. *Comput. Math. Organ. Theory* **15**(4), 303–328 (2009). <https://doi.org/10.1007/s10588-009-9063-5>
11. Segarra, S., Ribeiro, A.: Stability and continuity of centrality measures in weighted graphs. *IEEE Trans. Signal Process.* **64**(3), 543–555 (2016). <https://doi.org/10.1109/ICASSP.2015.7178599>
12. Martin, C., Niemeyer, P.: Influence of measurement errors on networks: estimating the robustness of centrality measures. *Network Sci.* **7**(2), 180–195 (2019). <https://doi.org/10.1017/nws.2019.12>
13. Murai, S., Yoshida, Y.: Sensitivity analysis of centralities on unweighted networks. In: *The World Wide Web Conference on - WWW 2019*, pp. 1332–1342 (2019). <https://doi.org/10.1145/3308558.3313422>
14. Meshcheryakova, N., Shvydun S.: Perturbation analysis of centrality measures. In: *2023 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE (2023). <https://doi.org/10.1145/3625007.3627590>
15. Bolland, J.M.: Sorting out centrality: an analysis of the performance of four centrality models in real and simulated networks. *Soc. Networks* **10**(3), 233–253 (1988). [https://doi.org/10.1016/0378-8733\(88\)90014-7](https://doi.org/10.1016/0378-8733(88)90014-7)

16. Herland, M., Pastran, P., Zhu, X.: An empirical study of robustness of network centrality scores in various networks and conditions. In: 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, pp. 221–228 (2013). <https://doi.org/10.1109/ICTAI.2013.42>
17. Niu, Q., Zeng, A., Fan, Y., Di, Z.: Robustness of centrality measures against network manipulation. *Physica A* **438**, 124–131 (2015). <https://doi.org/10.1016/j.physa.2015.06.031>
18. Krause, R.W., Huisman, M., Steglich, C., Snijders, T.A.B.: Missing network data a comparison of different imputation methods. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona, Spain, pp. 159–163 (2018). <https://doi.org/10.1109/ASONAM.2018.8508716>
19. Kossinets, G.: Effects of missing data in social networks. *Soc. Networks* **28**(3), 247–268 (2006). <https://doi.org/10.1016/j.socnet.2005.07.002>
20. Saari, D.G., Merlin, V.R.: The Copeland method: I: relationships and the dictionary. *Econ. Theory* **8**(1), 51–76 (1996)
21. Shvydun, S.: Normative properties of multi-criteria choice procedures and their superpositions: I. Working paper WP7/2015/07 (Part 1). Moscow: HSE Publishing House (2015). <https://doi.org/10.48550/arXiv.1611.00524>
22. Ficara, A., et al.: Social network analysis of Sicilian mafia interconnections. In: Cherifi, H., Gaito, S., Mendes, J., Moro, E., Rocha, L. (eds.) *Complex Networks and Their Applications VIII. COMPLEX NETWORKS 2019*. Studies in Computational Intelligence, vol. 882. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-36683-4_36
23. Cavallaro, L., Ficara, A., De Meo, P., Fiumara, G., Catanese, S., et al.: Disrupting resilient criminal networks through data analysis: the case of Sicilian Mafia. *PLoS ONE* **15**(8), e0236476 (2020). <https://doi.org/10.1371/journal.pone.0236476>
24. The World Integrated Trade Solution (2023). <https://wits.worldbank.org/>. Accessed 1 Sept 2023



ATEM: A Topic Evolution Model for the Detection of Emerging Topics in Scientific Archives

Hamed Rahimi^(✉), Hubert Naacke, Camelia Constantin, and Bernd Amann

Sorbonne University, CNRS, LIP6, Paris, France
{hamed.rahimi,hubert.naacke,
camelia.constantin,bernd.amann}@sorbonne-universite.fr

Abstract. This paper presents ATEM, a novel framework for studying topic evolution in scientific archives. ATEM employs dynamic topic modeling and dynamic graph embedding to explore the dynamics of content and citations within a scientific corpus. ATEM explores a new notion of citation context that uncovers emerging topics by analyzing the dynamics of citation links between evolving topics. Our experiments demonstrate that ATEM can efficiently detect emerging cross-disciplinary topics within the DBLP archive of over five million computer science articles.

Keywords: Evolution Model · Topic Emergence · Science Evolution

1 Introduction

The evolution of science is a continuous process that examines the development of new theories shaped by the collective efforts of scientists through research, experimentation, and analysis [1]. Understanding the evolution of science possesses the capacity to revolutionize the research landscape, as it has significant implications for research funding and public policy decisions in academic and industrial environments [2]. One of the most useful analyses of the evolution of science is the detection of topic emergence, which involves the identification of new areas of research and study within scientific disciplines [3]. Emerging topics are ideas or issues that gain attention or become more prominent in a particular field or area of interest. Detecting emerging topics has far-reaching implications for society, as it provides a way to track the progression of scientific fields and shape future research and technological development [4]. This task has been described by various communities using different terminologies such as trend analysis [5] and knowledge flow patterns [6]. Existing approaches suffer from certain limitations: Some of these approaches can identify trends for specific terms or phrases [7], but may not capture the broader context and relationships between scientific concepts. On the other hand, some can capture the relationships between scientific articles [8] but are less effective at identifying trends for specific terms or phrases. These limitations highlight the need for a

more holistic and versatile approach to provide a deeper understanding of topic evolution while preserving the nuanced relationships between scientific topics.

In this paper, we aim to discover emerging topics by proposing a framework called ATEM that discovers the evolution of science with different analyses. ATEM is driven by the recognition that citation links serve a dual purpose: they not only signify semantic connections among various subjects but also suggest the potential emergence of new topics within the cited interdisciplinary domains. Dynamic graph embedding allows ATEM to detect emerging topics and discover new interdisciplinary topics of the future.

2 Evolution Analysis in Scientific Archives

Several approaches have been proposed in the literature to analyze science evolution in scientific archives [9,10]. We categorize these approaches into two classes based on different factors that contribute to the advancement of science: Single-Domain Evolution Analysis and Cross-Domain Evolution Analysis. Single-Domain (SD) evolution refers to the development of scientific knowledge and methods within a particular discipline independent of external factors, while Cross-Domain (CD) evolution refers to the interaction and cross-fertilization of different scientific disciplines leading to new insights and discoveries.

Single-Domain Evolution Analysis attempts to describe the change in conceptual characteristics of scientific topics over time. This analysis is widely studied by dynamic topic models [11,12], which reflect the evolution by discovering latent semantic structures of the documents published in different time periods. A more recent family of topic models uses novel word embedding and language models to analyze content evolution. For example, Leap2Trend [13] relies on temporal word embeddings to track the dynamics of similarities between pairs of keywords and their rankings over time.

Cross-Domain Evolution Analysis focuses on observing the relational evolution of topics over time [14]. By comparing topic representations in different time periods, it is possible to build structured topic evolution networks and identify evolution patterns like topic merge and split [15]. The predictive power of evolutionary topic networks is also validated through the use of community detection algorithms to identify emerging topic correlations [16]. Document citation networks contain additional information about other semantic relationships between topics, such as topic influence and information flow [17,18]. By analyzing the structure of these networks over time, it is possible to study various more complex trends in the interaction between different topics [3,19] and to identify novel topic in its embryonic stage [20].

SD-based approaches are generally able to identify trends in the use of specific terms or phrases [7], but may not capture the broader context and relationships between scientific concepts. On the other hand, CD-based approaches can capture the relationships between scientific articles [8], but may be less effective at identifying trends in the usage of specific terms or phrases.

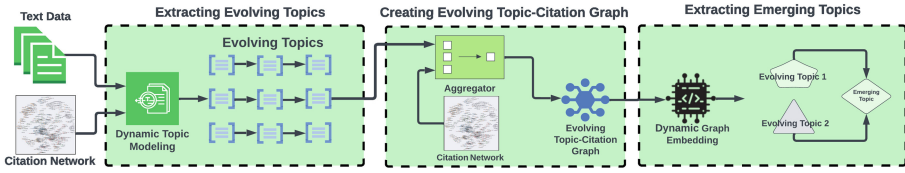


Fig. 1. The Architecture of ATEM

3 ATEM Framework

ATEM is a general-purpose framework for modeling and analyzing the evolution of topics generated from scientific archives. ATEM extracts *evolving topics* using dynamic topic models and builds *evolving topic-citation graphs* that connect topics through temporal citation links. One of the main goals of ATEM is to identify emerging research topics based on the topic citation graph. Our hypothesis is that citations in documents indicate a relationship between the topics discussed in those documents. ANTM explores the *emergence of evolving topics* by defining the notion of *citation context* using dynamic graph embedding techniques. This framework is consisting of 3 layers as illustrated in Fig. 1.

3.1 Extracting Evolving Topics

ATEM extracts evolving topics from a corpus of documents using a dynamic topic model. We use the following abstract definition of topics extracted from a scientific archive \mathcal{A} .

Definition 1 (Topics). A topic $t \in \mathcal{T}$ is a couple $(D(t), R(t))$ where $D(t) \subseteq \mathcal{A}$ denotes a subset of semantically similar documents, called the document cluster of t , and $R(t)$ is a weighted vector of terms in some vocabulary V , called the representation of topic t .

Definition 2 (Evolving Topics). Given an ordered sequence of possibly overlapping time periods $P = [p^0, \dots, p^n]$ and a scientific archive \mathcal{A} , an evolving topic is a sequence of topics $t = [t^0, \dots, t^n]$ such that all documents in $D(t^i)$ have been published during the time window p^i and $D(t) = \cup_{t^i \in t} D(t^i)$, is a cluster of semantically similar documents.

An example of evolving topics with its temporal representations is illustrated in Table 1. This layer allows to analyze the change within the word representation of a single evolving topic. Observing this change allows us to explore the semantic transformation in our understanding of a single topic by examining the words and phrases that are commonly associated with that topic. Besides, one can identify changes in the way people conceptualize a single topic for discussion and research. This kind of information is useful for researchers and companies seeking to stay up-to-date on the way people think about and discuss a particular topic.

Table 1. The Temporal Word Representation of The Evolving Topic ID T680C6.

Year	Label
2004	['knn', 'linear classifier', 'nearest neighbors', 'nearest neighbor', 'distributional', 'neighbor classifier']
2005	['knn', 'nearest neighbors', 'euclidian', 'pearson', 'instance based', 'neighbor nn']
2006	['knn', 'instance based', 'neighbor classifier', 'nearest neighbors', 'neighbor nn', 'neighbor knn']
2007	['knn', 'relief', 'nn classifier', 'neighbor nn', 'membership values', 'nearest neighbors']
2008	['knn', 'neighbor nn', 'nearest neighbour', 'nn algorithm', 'nearest neighbors', 'instance based']
2009	['knn', 'neighbor knn', 'nn classifier', 'nearest neighbors', 'neighbor nn', 'text classification']
2010	['nearest neighbors', 'neighbor classification', 'knn', 'metric learning', 'knn classifier', 'neighbor classifier']
2011	['instance selection', 'knn', 'neighbor classifier', 'neighbor classification', 'nearest neighbors', 'instance based']
2012	['knn', 'nearest neighbors', 'neighbor knn', 'test sample', 'instance based', 'nn classifier']
2013	['knn', 'nearest neighbors', 'based nearest', 'knn classifier', 'decision boundary', 'neighbor classifier']
2014	['knn', 'metric learning', 'nearest neighbors', 'nn classifier', 'knn classifier', 'neighbor nn']
2015	['nn classifier', 'knn classifier', 'knn', 'instance based', 'pmc', 'class label']
2016	['knn', 'instance selection', 'knn algorithm', 'knn classification', 'dpc', 'nn classification']
2017	['knn classifier', 'local mean', 'harmonic mean', 'nearest neighbors', 'knn', 'based nearest']
2018	['cent', 'knn classifier', 'knn', 'neighbor method', 'instance selection', 'nearest neighbors']

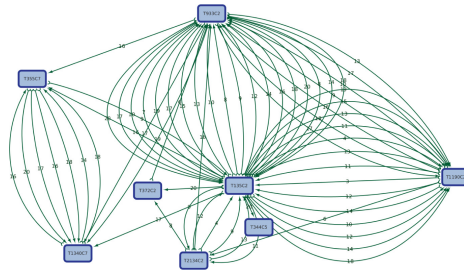
3.2 Creating Evolving Topic-Citation Graph

ATEM aims to discover citation relationships among the evolving topics as indicators of cross-domain evolution. This layer projects the structure of the citation network into evolving topics extracted in the previous layer and creates an evolving topic-citation graph. This graph is defined as follows.

Definition 3 (Evolving Topic Citations Graph). Let \mathcal{T} be a set of evolving topics defined over a scientific archive \mathcal{A} and $E_D(\mathcal{A}) \subseteq \mathcal{A} \times \mathcal{A}$ by a set of citation links defined on \mathcal{A} . Then, the topic clusters $D(t_i)$, all topics $t_i \in \mathcal{T}$ and the document citation edges E_D define a set of edges $(t_x, t_y, j) \in E_{\mathcal{T}}$ from an evolving topic t_x to evolving topic t_y if there exists at least one citation from some document in $D(t_x^j)$ to a document in $D(t_y^k)$ where $0 \leq k \leq j$:

$$E_{\mathcal{T}} = \{(t_x, t_y, j) \mid d \in D(t_x^j), d' \in D(t_y^k), 0 \leq k \leq j : E_D(d, d')\} \quad (1)$$

We add to each edge (t_x, t_y, i) in $E_{\mathcal{T}}$ a weight w which corresponds, to the number of citations which exist between the documents in $D(t_x^j)$ and $D(t_y^k)$, $0 \leq k \leq i$. Figure 2 is an example of Evolving Topic-Citation Graphs.

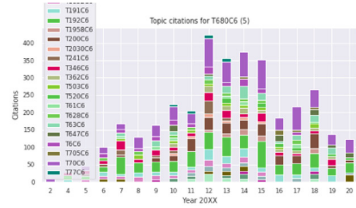
**Fig. 2.** Evolving Topic Citation Graph

This layer allows ATEM to observe the cross-domain evolution of evolving topics. Figures 3(b) and 3(d) show the evolution of citations to topics T680C6

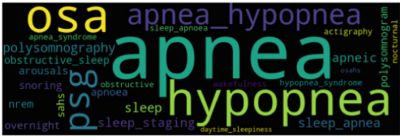
and T485C6 (with at least 5 citations) illustrated respectively by the word clouds in Figs. 3(b) and 3(c). ATEM investigates factors that contribute to the growth or decline of topics by observing the evolution of citation links between topics. Any variation in the existence and number of citation links between topics is a signal of change for the discovery of new knowledge, and in particular the emergence of new research topics. Figure 3(f) shows the evolution of co-citations to evolving topic T680C6 and T485C6. We can see that both topics have been cited by topic T70C6 (Fig. 3(e)) in 2009 and from 2012 to 2020. The number of co-citing topics increases in 2012 which might be considered as a first indication that both concepts are the origin of a new emerging research topic about nearest neighbor classifiers and apnea analysis.



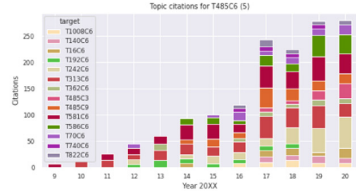
(a) Evolving topic T680C6.



(b) Citations to evolving topic T680C6.



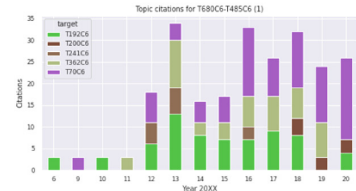
(c) Evolving topic T485C6.



(d) Citations to evolving topic T485C6



(e) Evolving topic T70C6



(f) Co-citations of T680C6 and T485C6

Fig. 3. Co-citation Analysis with ATEM

3.3 Extracting Emerging Topics

This layer applies a dynamic graph embedding method on the evolving topic-citation graph and defines the notion of citation context for evolving topics. Using this notion, Emerging Topics are defined as the couples or the sets of evolving topics with similar citation contexts.

Citation context refers to the ways in which documents in a given topic cite and are cited by documents in other topics. The citation context similarity between two topics can be seen as a measure of the likelihood of discovering new interdisciplinary topics in the future by merging these topics. We assume that *two evolving topics t_i and t_j with a highly similar citation context at a given time period produce an emerging evolving topic $t_{i,j}$* . Based on this assumption, we need to define a similarity measure that allows us to compare the context of two nodes defined by the topic citation graph $E_{\mathcal{T}}$. A graph embedding [21] of a graph G is a mapping function $emb : G \mapsto 2^{\mathbb{R}^d}$, which aims to represent nodes, edges, sub-graphs, or even the entire graph by low-dimensional feature vectors $v \in \mathbb{R}^d$ that preserve the topological and other contextual information about the encoded entity. The embedding dimension d is expected to be much smaller than the size of the graph $d \ll n$, where n is the number of nodes in G , which allows nodes to be efficiently compared by the encoded properties.

Table 2. Common documents for topic (T680C6,T661C6) emerging in 2013

Year	Title
2020.0	Performance evaluation of classification methods with PCA and PSO for diabetes.
2020.0	An Empirical Evaluation of Machine Learning Techniques for Chronic Kidney Disease Prophecy
2020.0	Using Machine Learning to Predict the Future Development of Disease
2019.0	Performance Analysis of Machine Learning Techniques to Predict Diabetes Mellitus.
2018.0	Accurate Diabetes Risk Stratification Using Machine Learning: Role of Missing Value and Outliers.
2017.0	Automatic Diagnosis Metabolic Syndrome via a k- Nearest Neighbour Classifier.
2016.0	Predicting risk of suicide using resting state heart rate.
2015.0	Computer-aided diagnosis of diabetic subjects by heart rate variability signals using discrete wavelet transform method
2013.0	Automated detection of diabetes using higher order spectral features extracted from heart rate signals

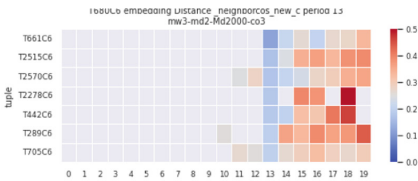
There are several dynamic representation learning methods capable of embedding nodes in a low-dimensional vector space which captures the evolution of the network structure. We use dynamic node embeddings [22], which project each node v in a sequence of graphs into a *sequence $emb(v)$ of low-dimensional vectors*. By projecting the topic citation links $E_{\mathcal{T}}$ defined in previous layer on each time period $p^i \in P$ a set of topic edges $E_{\mathcal{T}}^i = \{(t_x, t_y) \mid (t_x, t_y, i) \in E_{\mathcal{T}}\}$, we can produce a *sequence of graphs $\mathcal{G}(P) = [(T^i, E_{\mathcal{T}}^i) \mid p^i \in P]$* ordered by periods p^i that reflects the distribution of citations between documents of all topics for all time periods. Our hypothesis is that the dynamic topic embedding vector $emb(t)$ of a topic t in a dynamic topic citation graph represents the evolution of the citation context of t , and that two topics with similar embedding (citation context) at period p^i are likely to generate new emerging topics. More formally, we can now provide a more precise definition of emerging topics:

Definition 4 (Emerging Topics). *Two evolving topics t_i and t_j define an evolving topic $t_{i,j}$ emerging at time period p^k , if the context distance $dist(t_i^k, t_j^k)$ at period p^k is above a given threshold ϕ and below this threshold before p^k .*

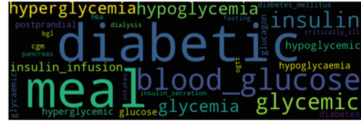
To compute the similarity between the citation context of two evolving topics, we use *cosine-similarity* on the dynamic vector representations of evolving topics. Using this definition, we can now detect emerging topics in two ways:

1. K-nearest neighbors of a given topic t : we generate for each evolving topic t and period p^i a set of nearest neighbors with minimal embedding distance higher than a given threshold.
2. Cluster the embeddings of each period: we apply a clustering algorithm on the topic embeddings of each period. Each cluster represents an emerging topic defined by a set of similar topics.

Figure 4(a) shows the evolution of the embedding distance in the neighborhood of topic T680C6 (nearest neighbor classifiers) in 2013. We can see, for example, topic T661C6 describing documents on diabetes appears in 2013 as a near embedding neighbor of evolving topic T680C6 (with a maximal distance of 0.2). Table 2 shows the documents that are common to T680C6 and T661C6. These documents are obtained by taking the intersection of the results of two queries $R(T680C6) = [\text{'nearest neighbors'}, \text{'knn'}, \text{'nearest neighbor'}]$ and $R(T661C6) = [\text{'glycemic'}, \text{'hypoglycemia'}, \text{'hyperglycemia'}]$ ranked by the average search score. The result shows that most of the top relevant documents for emerging topic (T680C6, T661C6) have been published after its emergence period 2013.



(a) Evolution of embedding distance for topic T680C6 at time period 2013



(b) Evolving topic T661C6

Fig. 4. Extracting emerging topic T661C6 with citation context

4 Implementation

ATEM has been applied to the DBLP dataset of 5M scientific articles published between 2000 and 2020. The evolving topics are extracted through the customized architecture of BERTopic [23] and Top2Vec [24]. This customization includes a combined method for document clustering, which is based on Doc2Vec [25] on document content and Leiden community detection [26] on the citation graph. These clusters are aggregated into a set of new clusters that regroup semantically similar documents published and cited within a scientific community. The topic document clusters $D \in \mathcal{T}$ are then divided into $n = |P|$ time frames denoted by $D = (D^1, \dots, D^n)$ where each D^i , is a cluster of documents in period p^i . In this regard, we adopt the dynamic document integration of clusters upon using static time windows. We only keep clusters with a minimal number of 3 documents. Each of these topic clusters is represented in two manners:

- Nearest Words: we compute for each document cluster a centroid vector by averaging over the embeddings of its vectors. The cluster representation is defined by the top- n words corresponding to the n nearest embedding neighbors of the centroid vector.
- Class-based TF-IDF: similar to [23], we regroup the documents of each topic cluster $D(t^i)$ in all time periods p^i and apply TF-IDF to each group to find the top- n word representation for each group.

We then create the evolving topic-citation graph as explained in the previous section. To compute the temporal node embeddings on the topic citation graph, we used `OnlineNode2Vec` [27], which is based on `StreamWalk` and online second-order similarity. The result is a temporal embedding for each topic, which can be used to compare evolving topics by their citation context. In particular, it allows us to identify evolving topics (t_x, t_y) when the distance between two evolving topics t_x and t_y is less than a given threshold.

5 Proof of Concept

The objective of this section is to demonstrate the effectiveness of the proposed framework in identifying emerging topics as compared to co-citation analysis. To achieve this, we compare the emerging topics within two groups: one based on the embedding representations of the topic-citation graph (referred to as *EmbeddingContext*), and the other based on the shortest citation paths defined between topics (referred to as *CitationContext*). To facilitate this comparison, we generate a set of emerging topics from both *EmbeddingContext* and *CitationContext*. We assess the validity of these emerging topics by examining the presence of related documents in the past and future of their discovery. To quantify this, we employ a predictability metric that evaluates the distribution of related documents over time. By scoring the emerging topics based on this metric, we can effectively evaluate their predictive power and performance.

We first generate a random sample of 200 evolving topics T . For each evolving topic $t \in T$, we generate two sets of $n = 10$ topics in each time period p^i :

- *EmbeddingContext*(t^i) contains n topics t_x that are *new* nearest embedding neighbors of t^i at period p^i with a given maximum distance threshold of 0.2 and minimum embedding norm equal to 0.22 to remove noisy embedding vectors (t_x was not a neighbor before period p^i).
- *CitationContext*(t^i) contains a random set of n topics t_x connected for the first time to the evolving topic t at period p^i by a citation path of maximal length 3.

In both sets, each pair $t_e = (t, t_x)$ generated by $t_x \in \text{CitationContext}(t^i)$ and $t_x \in \text{EmbeddingContext}(t^i)$ is expected to form an emerging topic at period p^i . To explore this expectation, we consider all pairs t_e as emerging in time period p^i with representation $R(t_e) = [R(t^j) \cup R(t_x^j) \mid p^j \in P]$ and a document cluster $D(t_e) = [D(t^j) \cap D(t_x^j) \mid p^j \in P]$ over all periods p^j in P .

We then look at each of these new topics and investigate their emergence predictability based on the year their papers get published. Therefore, we partition $D(t_e)$ into two subsets: $D_{past}(t_e)$ of documents published before the emergence period of t_e , and $D_{future}(t_e)$ of documents in $D(D)$ published after the emergence period of t_e .

Finally, we quantify the *emergence predictability* \mathcal{E} of each topic pair t_e by defining the following function that measures the distribution of its documents before and after its emergence period:

$$\mathcal{E}(t_e) : \frac{|D_{future}(t_e)| - |D_{past}(t_e)|}{|D(t_e)|} \tag{2}$$

meaning (i) when $\mathcal{E}(t_e) = 1$, all documents are published at emergence period of (t, t_e) or afterwards, (ii) when $\mathcal{E}(t_e) = 0$, the same number of documents are published before and after the emergence period and (iii) when $\mathcal{E}(t_e) = -1$, all documents are published before period p .

Figure 5 compares the predictability values for emerging topics of *EmbeddingContext* and *CitationContext*. We find that random pairs from *EmbeddingContext* have higher predictability compared to *CitationContext*. Figures 6(a) and 6(b) show the box-plot and violin distribution of predictability values. By Eq. (3), we can observe that in average (i) 75% of emerging topics generated by *EmbeddingContext* have $1.25/0.75 = 1.66$ times more publications after emergence than before and (ii) 50% of emerging topics in *EmbeddingContext*, have $2.6/0.4 = 6.2$ more publications after emergence than

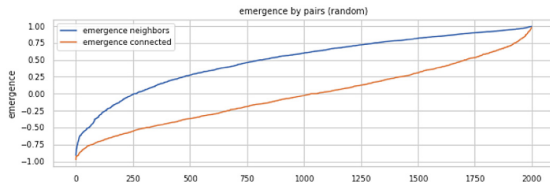
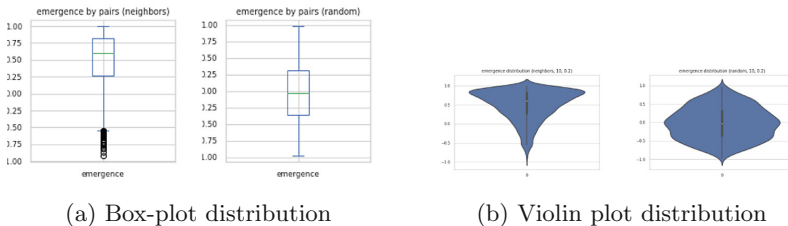


Fig. 5. Average emergence values of *EmbeddingContext* and *CitationContext*.



(a) Box-plot distribution

(b) Violin plot distribution

Fig. 6. Emergence predictability distribution.

before, whereas the ratio is 1 for topics generated by *CitationContext*.

$$\frac{|D_{future}(t_e)|}{|D_{past}(t_e)|} = \frac{\mathcal{E}(t_e) + 1}{1 - \mathcal{E}(t_e)} \tag{3}$$

Emerging topic properties. Fig. 8(b) shows the correlation between various parameters that shape the dynamics of emerging topics. We can see that the average embedding distance (*dist*) per period increases in time (*year*). This signifies that the applied dynamic embedding method estimates that the analyzed topics get more and more diverse. However, this conclusion has to be confirmed by a deeper analysis of the bias introduced by the dynamic computation algorithm. Second, the predictability (*emergence*) decreases with increasing distance and *strongly decreases* in time (see also Fig. 7(a)). This is a natural consequence of the definition of emergence which compares the number of relevant documents before and after the emergence period. This number is also influenced by the “relative length” of the past and the future covered by the archive (as shown in Fig. 7(b), the average emergence of topic pairs is positive before 2016 and becomes negative afterward). Finally, we can see that the average cluster size (*all*) of emerging topics is independent of the period, the average predictability, and the average embedding distance.

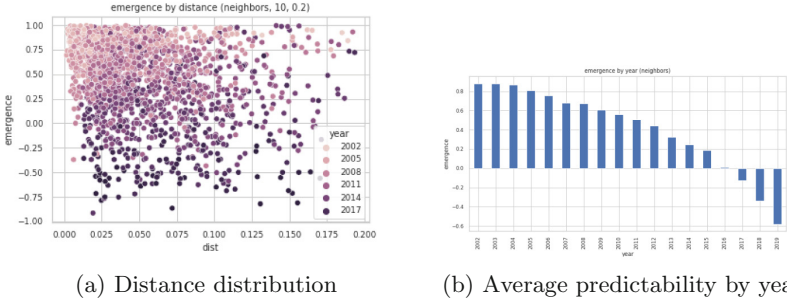


Fig. 7. Distance and predictability of emerging topics.

Figure 8(a) shows the correlations between the average number of new emerging topics (*EmbeddingContext*) by period (*n*), the average number of connected pairs (*CitationContext*) by period (*c*), and the average number of connected emerging topics (intersection of *EmbeddingContext* and *CitationContext*) by period (*cn*). We can see that the average number of embedding neighbors decreases with time, which is consistent with the observation that the embedding distance increases. The fraction of connected neighbors is independent of the number of neighbors, but increases with the number of connected topics.

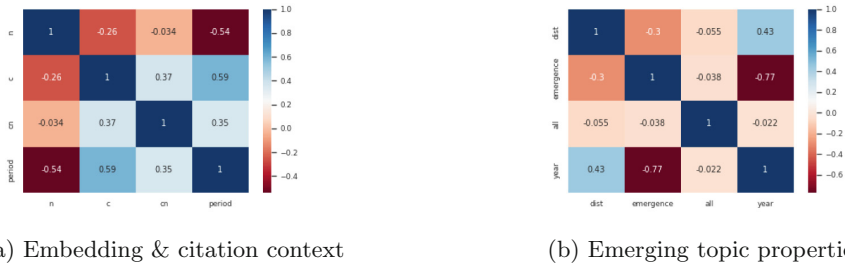


Fig. 8. The correlation analysis

6 Conclusion

This article presents a new framework for studying the evolution and emergence of topics over time. The analysis framework is based on the notions of single-domain and cross-domain evolution, aiming to distinguish between the evolution of individual topics and the evolution of relationships between topics. This framework is then used to detect emergent topics by using recent graph embedding techniques on topic citation graphs to analyze the evolution of citation context at the topic level and to detect similar topic pairs as new emergent topics. We have implemented this framework, and our experiments show that citation context-based topic similarity is efficient for detecting emerging topics.

References

1. Luhmann, N.: Evolution of science. *Epistemol. Philos. Sci.* **52**(2), 215–233 (2017)
2. Moghissi, A.A., Straja, S.R., Love, B.R., Bride, D.K., Stough, R.R.: Innovation in regulatory science: evolution of a new scientific discipline. *Technol Innov* **16**(2), 155–165 (2014)
3. Jung, S., Segev, A.: Identifying a common pattern within ancestors of emerging topics for pan-domain topic emergence prediction. *Knowl.-Based Syst.* **258**, 110020 (2022)
4. Ohniwa, R.L., Hibino, A.: Generating process of emerging topics in the life sciences. *Scientometrics* **121**(3), 1549–1561 (2019)
5. An, Y., Han, M., Park, Y.: Identifying dynamic knowledge flow patterns of business method patents with a hidden Markov model. *Scientometrics* **113**(2), 783–802 (2017)
6. Sharma, S., Swayne, D.A., Obimbo, C.: Trend analysis and change point techniques: a survey. *Energy, Ecol. Environ.* **1**, 123–130 (2016)
7. Rahimi, H., Naacke, H., Constantin, C., Amann, B.: ANTM: an aligned neural topic model for exploring evolving topics. arXiv preprint [arXiv:2302.01501](https://arxiv.org/abs/2302.01501) (2023)
8. Cordeiro, M., Sarmiento, R.P., Brazdil, P., Gama, J.: Evolving networks and social network analysis methods and techniques. *Social media and journalism-trends, connections, implications*, pp. 101–134 (2018)

9. Rossetto, D.E., Bernardes, R.C., Borini, F.M., Gattaz, C.C.: Structure and evolution of innovation research in the last 60 years: Review and future trends in the field of business through the citations and co-citations analysis. *Scientometrics* **115**(3), 1329–1363 (2018)
10. Balili, C., Lee, U., Segev, A., Kim, J., Ko, M.: TermBall: tracking and predicting evolution types of research topics by using knowledge structures in scholarly big data. *IEEE Access* **8**, 108514–108529 (2020)
11. Blei, D.M., Lafferty, J.D.: Dynamic Topic Models. In: Proceedings of the 23rd International Conference on Machine Learning, ICML '06, (New York, NY, USA), pp. 113–120, ACM (2006)
12. Abdelrazek, A., Eid, Y., Gawish, E., Medhat, W., Hassan, A.: Topic modeling algorithms and applications: a survey. *Inf. Syst.* **112**, 102131 (2023)
13. Dridi, A., Gaber, M.M., Azad, R.M.A., Bhogal, J.: Leap2trend: a temporal word embedding approach for instant detection of emerging scientific trends. *IEEE Access* **7**, 176414–176428 (2019)
14. Liu, H., Chen, Z., Tang, J., Zhou, Y., Liu, S.: Mapping the technology evolution path: a novel model for dynamic topic detection and tracking. *Scientometrics* **125**(3), 2043–2090 (2020)
15. Chavalarias, D., Cointet, J.-P.P.: Phylomemetic patterns in science evolution—the rise and fall of scientific fields. *PLoS ONE* **8**(2), e54847 (2013)
16. Salatino, A.A., Osborne, F., Motta, E.: AUGUR: forecasting the emergence of new research topics. In: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, JCDL '18, (New York, NY, USA), pp. 303–312, ACM (2018)
17. Andrei, V., Arandjelović, O.: Complex temporal topic evolution modelling using the Kullback-Leibler divergence and the Bhattacharyya distance. *EURASIP J. Bioinform. Syst. Biol.* **2016**(1), 1–11 (2016). <https://doi.org/10.1186/s13637-016-0050-0>
18. Beykikhoshk, A., Arandjelović, O., Phung, D., Venkatesh, S.: Discovering topic structures of a temporally evolving document corpus. *Knowl. Inf. Syst.* **55**, 599–632 (2018)
19. Jung, S., Segev, A.: DAC: descendant-aware clustering algorithm for network-based topic emergence prediction. *J. Informet.* **16**, 101320 (2022)
20. Salatino, A.A., Osborne, F., Motta, E.: How are topics born? Understanding the research dynamics preceding the emergence of new areas. *PeerJ Computer Science* **3**, e119 (2017)
21. Cai, H., Zheng, V.W., Chang, K.C.-C.: A comprehensive survey of graph embedding: problems, techniques, and applications. *IEEE Trans. Knowl. Data Eng.* **30**(9), 1616–1637 (2018)
22. Mahdavi, S., Khoshraftar, S., An, A.: Dynnode2vec: Scalable Dynamic Network Embedding. [arXiv:1812.02356](https://arxiv.org/abs/1812.02356) Feb. (2019)
23. Grootendorst, M.: Bertopic: Neural topic modeling with a class-based tf-idf procedure. arXiv preprint [arXiv:2203.05794](https://arxiv.org/abs/2203.05794) (2022)
24. Angelov, D.: Top2vec: Distributed representations of topics. arXiv preprint [arXiv:2008.09470](https://arxiv.org/abs/2008.09470) (2020)
25. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems*, vol. 26, (2013)
26. Traag, V.A., Waltman, L., Van Eck, N.J.: From louvain to leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**(1), 1–12 (2019)
27. Béres, F., Kelen, D.M., Pálovics, R., Benczúr, A.A.: Node embeddings in dynamic graphs. *Appl. Netw. Sci.* **4**, 64 (2019)



Analysis and Characterization of ERC-20 Token Network Topologies

Matteo Loporchio^(✉), Damiano Di Francesco Maesa, Anna Bernasconi,
and Laura Ricci

Department of Computer Science, University of Pisa, Largo Bruno Pontecorvo 3,
56127 Pisa, Italy

`matteo.loporchio@phd.unipi.it`,

`{damiano.difrancesco,anna.bernasconi,laura.ricci}@unipi.it`

Abstract. The transparent nature of public blockchain systems allows for unprecedented access to economic community data. Examples of such communities are the fungible token networks created by the ERC-20 standard on the Ethereum protocol. In this paper we study ERC-20 token networks, where nodes represent users and edges represent fungible token transfers between them. We focus our analysis on the top 100 largest networks, including a total of about 160 million edges and 60 million nodes. After a global analysis of the size and temporal evolution of such networks, we define and study seven features describing their main topological properties. In an attempt to characterize the networks by their topologies, we use the introduced features to cluster the networks together. To evaluate our results, we manually classify each network depending on the application domain of the corresponding contract and measure the homogeneity of the obtained clusterings. Overall, the results appear to indicate a lack of relationship between the scope of a contract and the topological features of the induced networks.

Keywords: Blockchain · Ethereum · Fungible Token · Network Analysis

1 Introduction

The advent of blockchain technology has disrupted traditional paradigms across multiple sectors, including financial systems, intellectual property, decentralized identity and supply chain management. Indeed, blockchains have the ability to provide secure, transparent, and decentralized record-keeping, eliminating the need for trusted intermediaries in transactions. Within this ever-evolving landscape, Ethereum – ranking as the second largest blockchain by market capitalization – has stood out for its innovations, foremost among them being the capability to store and execute code, in the form of *smart contracts* [13]. A smart contract is a piece of arbitrary code whose execution is validated by consensus, i.e., replicated by all participants of the blockchain network. Smart contracts have enabled the development of a wide range of *decentralized applications*

(DApps) running on the blockchain. Nowadays, DApps serve a variety of purposes, including decentralized finance, gaming, and social networking. Moreover, many DApps utilize the concept of *token*, namely a transferable asset that can be either *fungible* or *non-fungible*. Fungible tokens are interchangeable and identical, like traditional currencies. For instance, in the context of gaming, fungible tokens may represent reputation or player skills, while in the field of finance they can be used to represent assets or fiat currencies. Conversely, non-fungible tokens (NFTs) are unique digital assets with distinct properties, each with a distinct value. NFTs are often used to represent ownership of digital or physical items (e.g., works of art, collectibles, and more).

To enforce interoperability among fungible tokens on Ethereum, the ERC-20 standard was introduced. This standard defines rules for smart contracts implementing such tokens, facilitating token integration and exchange across various decentralized applications. In addition to this, each ERC-20 token creates a unique economy within the Ethereum ecosystem, where participants hold and trade tokens of the same kind. From a more theoretical perspective, we can say that each economy can be modeled as a *token network*, i.e., a graph whose nodes correspond to participants and edges represent token exchanges. Therefore, the analysis of ERC-20 token networks provides useful insights on the corresponding token economies. Indeed, it allows us to understand the evolution of transfers and how users tend to interact within these economies, e.g., whether they form communities, or if certain users hold more central roles with respect to others.

Motivated by these reasons, in this paper we study the properties of the top 100 ERC-20 token networks by total number of transfers. To gather information about transfers, we use data from the first 15 million Ethereum blocks, covering the time period between July 30th, 2015 and June 21st, 2022. Specifically, we exploit Ethereum transaction receipts, which include information about ERC-20 Transfer events. Indeed, such events serve as the main mechanism for notifying participants of token transfers, recording the sender, recipient, and the amount of tokens transferred. Our main contribution is articulated as follows. First, we study the historical evolution of transfer events within the analyzed data set. Then, we analyze the topological properties of token networks by associating each network with a set of seven features describing its connectivity, degree distribution, transitivity, density, diameter, and average shortest path length. Subsequently, we use such features to conduct further analysis based on clustering techniques, aiming at identifying groups of networks sharing similar topological properties. Finally, we classify token networks based on the application domain of the corresponding token. We use this classification to investigate possible connections between the topology of a network and the semantics of the corresponding ERC-20 token.

Related Work. Ethereum token networks have already been studied in the literature. The authors of [9] analyzed the global ERC-20 token network, i.e. the union of all ERC-20 token networks, between February 2016 and February 2018. They found out that the degree distribution follows a power-law and the token popularity among buyers and sellers also follows a power law model. Similarly,

the analysis in [11] revealed that many ERC-20 token networks exhibit either a star or hub-and-spoke topology. Additionally, such networks tend to have low clustering coefficients and are disassortative. Instead, the authors of [4] found out that, despite the high number of ERC-20 tokens, only a few are active and valuable. Moreover, few accounts hold a large number of tokens, while many accounts only hold a small number of tokens. Lastly, the authors discovered that some addresses create a large number of tokens to attack the Ethereum network.

If compared to prior works, our analysis is based on a broader time period and focuses on the top 100 networks with the highest number of token transfers. Moreover, our contribution is not solely focused on analyzing networks but also on comparing them with each other by associating each network with a set of numerical features capturing its topological properties. Lastly, our analysis also introduces a semantic classification of token contracts obtained by manually retrieving information from the internet.

2 Background

Blockchain. A blockchain is a shared, immutable, and decentralized ledger organized in blocks, each containing ledger state updates and managed through a *distributed consensus algorithm*. Ethereum [13] has been the first blockchain project implementing a Turing-complete virtual machine, called *Ethereum Virtual Machine* (EVM). This means that, besides monetary transactions, the Ethereum blockchain is also capable of storing and executing pieces of arbitrarily complex code, called *smart contracts* [10]. Smart contracts are written in a high-level language (e.g., Solidity) and then compiled to bytecode. Their execution is validated by distributed consensus and replicated by all participants. Specifically, each call to a function of a smart contract is executed sequentially in the current block state, and the final state is updated accordingly.

Decentralized Applications and Fungible Tokens. As stated in Sect. 1, smart contracts enable the development of *decentralized applications* (DApps), which may serve a wide range of purposes (e.g., finance, gaming, social networking). Many DApps adopted the concept of fungible token to represent interchangeable assets that can be transferred between participants. The ERC-20 implementation proposal [12] introduces a standard for fungible tokens. Specifically, it defines a consistent set of methods for creating and interacting with tokens. Also, it ensures token interoperability, meaning that all compliant tokens can be easily integrated into different decentralized applications. For the purposes of this paper, we remark that, whenever an ERC-20 contract transfers tokens between two addresses, an *event* must be raised. In Ethereum, events are a mechanism adopted to notify a state update or a particular condition being met during the execution of a smart contract. This facilitates the communication between contracts and off-chain applications. In Solidity, events are identified by a signature specifying the type and number of their parameters. The signatures of the Transfer and Approval events defined by the ERC-20 standard are:

```

event Transfer(address, address, uint256)
event Approval(address, address, uint256)

```

The Transfer event is emitted every time a token transfer occurs between two addresses. Its signature consists of three parameters: the sender address, the recipient address, and the amount of tokens transferred. Conversely, the Approval event is triggered when a user allows another participant to transfer a certain number of tokens on their behalf. We observe that, according to the ERC-20 standard definition, after the issuance of an Approval event, a Transfer event notifying the actual transfer of tokens must necessarily follow. Thus, for the remainder of this paper, we will only consider Transfer events to study token transfers among participants.

3 Transfer Event Graph

Transfer events represent redistributions of tokens between two users. By gathering information about the occurrences of such events, it is therefore possible to analyze the evolution of a token economy. To this aim, in this section we formalize the concept of Transfer event graph, i.e., the graph where nodes represent users and edges represent Transfer event occurrences. In the following, we denote by \mathcal{A} the set of all Ethereum addresses, which are used to identify network participants. An occurrence of a Transfer event can be represented as a tuple $e = (t, from, to, v)$, where $t \in \mathbb{N}$ is a numeric timestamp, $from \in \mathcal{A}$ is the address of the sender, $to \in \mathcal{A}$ is the receiver address and $v \in \mathbb{N}$ is the amount of tokens transferred. In the following, given a contract C , we denote by $\mathcal{T}(C)$ the set of ERC-20 Transfer events triggered by C . We can then define the *Transfer event graph* of C as a simple undirected graph $G_C = (V_C, E_C)$. Here, the set of vertices $V_C = \{a \in \mathcal{A} \mid \exists (t, from, to, v) \in \mathcal{T}(C) \text{ s.t. } a = from \vee a = to\}$ contains all addresses induced by the events in $\mathcal{T}(C)$, while the set of edges $E_C = \{(a, b) \mid \exists (t, from, to, v) \in \mathcal{T}(C) \text{ s.t. } a = from \wedge b = to\}$ includes one edge between two nodes a and b if and only if there exists at least one token transfer between them.

4 Experimental Results

In this section we present the experimental results of our analysis of token networks. First, we study the evolution of Transfer events over time. Then, we compare the topological properties of Transfer event networks and examine possible connections between such properties and the semantics of the corresponding smart contracts. For our experiments, we downloaded the first 15 million blocks of the Ethereum blockchain along with the corresponding transaction receipts, which include all necessary information about triggered events. The time period covered by our data set ranges from July 30th, 2015 03:26:13 PM UTC, to June 21st, 2022 02:28:10 AM UTC. The code for the experiments and data analysis has been written in C++, Java and Python and is publicly available at <https://github.com/mlporchio/EthTokenAnalysis>. In particular, the Transfer event graph analysis was conducted using igraph [6] and WebGraph [1].

4.1 Global Analysis

By analyzing the transaction receipts in our data set, we were able to collect $N_e = 961\,603\,795$ occurrences of the Transfer event, raised by $N_c = 386\,615$ different smart contracts. The plots of Fig. 1 provide further insight into the occurrences of Transfer events. In particular, Fig. 1a illustrates the frequency of ERC-20 Transfer events within the analyzed blocks. It appears that a significant number of blocks (i.e., above 10^6) do not contain any occurrence of such events. Also, we can notice that blocks with a large quantity of transfers are less frequent. Instead, Fig. 1b illustrates the total number of Transfer events on a monthly basis starting from 2015 (i.e., the year of the Ethereum blockchain inception) until June 2022. Using a logarithmic scale on the y -axis, the plot highlights how the number of such events experienced a rapid growth in 2016 and 2017, before stabilizing at around 10^7 transfers per month starting from 2018.

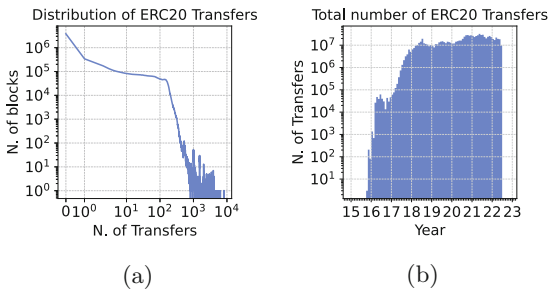


Fig. 1. Frequency distribution of ERC-20 transfers (left) and monthly number of raised Transfer events (right).

4.2 Graph Construction

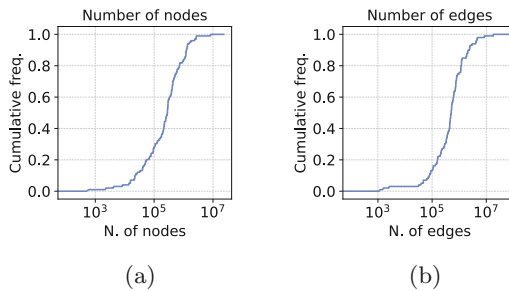
To gain insight on the trading volume of each token economy, we first ranked the ERC-20 contracts based on the number of raised Transfer events. Table 1 displays the first ten positions of our ranking. As the reader may notice, these contracts alone include approximately 357 million occurrences, thus covering about 37% of the total number of events N_e despite being less than the 0.012% of the number of contracts N_c . Moreover, we can also observe that eight tokens out of ten are related to the field of decentralized finance, as they are associated with stablecoins or wrapped tokens. The only exceptions are represented by the tokens of ChainLink [3], i.e., a decentralized oracle network, and Livepeer, a framework for decentralized video streaming applications.

We selected the top 100 contracts from our ranking and constructed, for each of them, the corresponding Transfer event graph, as discussed in Sect. 3. We then computed the number of nodes and edges of each graph and noticed that, on average, Transfer event graphs have about 759 004 nodes and 1 701 879 edges. We remark that the number of nodes coincides with the number of participants in the

Table 1. Top 10 ERC-20 token contracts by triggered Transfer events.

Contract address	Token name	N. of Transfers	Percentage
dac17f958d2ee523a2206206994597c13d831ec7	Tether USD (USDT)	149 408 698	15.537
c02aaa39b223fe8d0a0e5c4f27ead9083c756cc2	Wrapped Ether (WETH)	104 183 120	10.834
a0b86991c6218b36c1d19d4a2e9eb0ce3606eb48	USD Coin (USDC)	42 601 224	4.430
6b175474e89094c44da98b954eedeac495271d0f	Dai Stablecoin (DAI)	14 387 573	1.496
514910771af9ca656af840dff83e8264ecf986ca	ChainLink Token (LINK)	11 388 177	1.184
174bfa6600bf90c885c7c01c7031389ed1461ab9	More Gold Coin (MGC)	8 947 669	0.930
95ad61b0a150d79219dcf64e1e6cc01f0b64c4ce	SHIBA INU (SHIB)	7 781 424	0.809
990f341946a3fdb507ae7e52d17851b87168017c	Strong (STRONG)	6 964 935	0.724
58b6a8a3302369dac383334672404ee733ab239	Livepeer Token (LPT)	6 025 932	0.627
03cb0021808442ad5efb61197966aef72a1def96	coToken (coToken)	5 370 855	0.559
Total		357 059 607	37.130

corresponding token economy. For a more detailed insight, Fig. 2a summarizes the cumulative frequency of the number of nodes among all graphs. From the plot, it is possible to notice that the majority of all graphs has between 10^4 and 10^6 nodes. Specifically, we can notice that 80 graphs out of 100 have less than 1 million nodes. Similarly, Fig. 2b illustrates the cumulative distribution function for the number of edges, highlighting that approximately 80% of all graphs have less than 1 million edges. Speaking of graph sizes, we observe that the graph with the lowest number of nodes, amounting to 691, corresponds to the ‘‘Bancor Network’’ token, which is related to the field of decentralized finance. Instead, the graph with the highest number of nodes, namely 23 176 194, is that of ‘‘Tether USD’’ token, the stablecoin holding the first position in Table 1. To give a sense of our data set, we note that, if all 100 graphs were combined into a single graph describing all participants and transfer events of the corresponding 100 economies, the resulting graph would comprise 59 120 625 unique nodes and 160 259 567 unique edges.


Fig. 2. Cumulative distributions for number of nodes (left) and edges (right) of the considered Transfer event graphs.

4.3 Graph Analysis

We then analyzed the constructed Transfer event graphs. To this aim, we associated each graph with seven numerical features capturing their topological

properties. To deal with disconnected graphs, we have chosen to always compute such measures on the largest connected component for consistency. As such, all features we describe from now on always refer to the subgraph induced by the nodes and edges of the largest component. In particular, given a Transfer event graph G with largest connected component G_{LCC} , we computed the following features. (1) *Coverage*, namely the percentage of nodes of G included in G_{LCC} . (2) *Alpha*, which represents the exponent of the power law distribution best fitting the degree distribution of G_{LCC} . (3) *Fitting error*, which corresponds to the error obtained during the fitting process to obtain the previously described alpha. (4) *Relative diameter*, which represents the ratio between the diameter of G_{LCC} and the natural logarithm of the number of nodes. (5) *Relative average shortest path length*, which is computed as the average shortest path length of G_{LCC} divided by the natural logarithm of the number of nodes. (6) *Transitivity* coincides with the global clustering coefficient of G_{LCC} , namely the ratio between the number of triangles and connected triples in the graph. (7) *Density*, as the ratio between the actual number of edges and the maximum possible number of edges in G_{LCC} . To fit a power law curve on the degree distribution of each graph, we used the procedure detailed in [5]. In accordance with such method, we use the Kolmogorov-Smirnov statistic to quantify the fitting error as the distance between the two distributions. Moreover, we remark that the average shortest path lengths have been computed using the HyperBall algorithm, which provides an approximate but reasonably accurate result [2]. Indeed, due to the sizes of the analyzed graphs, obtaining the exact value for the lengths turned out to be too computationally expensive.

Figure 3 summarizes the distributions of the features among all graphs. In particular, the histogram of Fig. 3a illustrates the coverage distribution and provides information about the connected components of the examined graphs. We can observe that, for 98% of the graphs, the largest connected component covers a percentage of nodes ranging from 90% to 100%. This means that, in most cases, as the token economy evolves, token transfers tend to create a single, large community of users, with only a few nodes remaining isolated. There are, however, two graphs where the coverage percentage falls between 10% and 20%. A further analysis revealed that these two outliers correspond to the “Etheal Promo” and “INS Promo” tokens, whose largest connected components cover around 18% and 14% of all nodes, respectively. Both tokens were launched on the market through *airdropping*, a marketing strategy where tokens are sent to existing users’ wallets, typically as a free giveaway.

Our analysis of node degrees is summarized by Figs. 3b and 3c, which illustrate the distributions of the fitted power law exponents and fitting errors, respectively. More than half of the tokens have a power law exponent between 2.5 and 3.75, while the majority of graphs have a fitting error below 0.05. Indeed, we observed that the mean fitting error over all graphs is 0.02. Interestingly enough, the graph with maximum fitting error (i.e., approximately 0.15) corresponds to the “More Gold Coin” token. As discussed in [7], the associated contract address is known for its spamming campaign, which took place in July 2019. During this

massive campaign, small quantities of tokens were airdropped to many users causing a sudden congestion on the entire Ethereum network.

For what concerns the relative diameter, we observe a mean value of approximately 1.55. Indeed, Fig. 3d shows that, for more than 70% of all graphs, this feature is below 2. So the diameter is within a low linear factor of the logarithm of number of nodes, a classical behaviour in small world networks. Similarly, for the relative average path length, Fig. 3e shows how the values for this feature are concentrated between 0.2 and 0.3 for most graphs, with a mean of 0.28.

The histograms of Figs. 3f and 3g describe the transitivity and density distributions, respectively, using a logarithmic scale on the y -axis. As the reader may notice, in both cases the distributions are positively skewed, with a mean value of about 3.55×10^{-4} for transitivity and 2.07×10^{-4} for density. This suggests that interactions among participants tend to be sparse. Moreover, it leads us to believe that token networks have a weak community structure and participants are not likely to form well-connected groups, in contrast with the small world behavior observed when looking at the diameter.

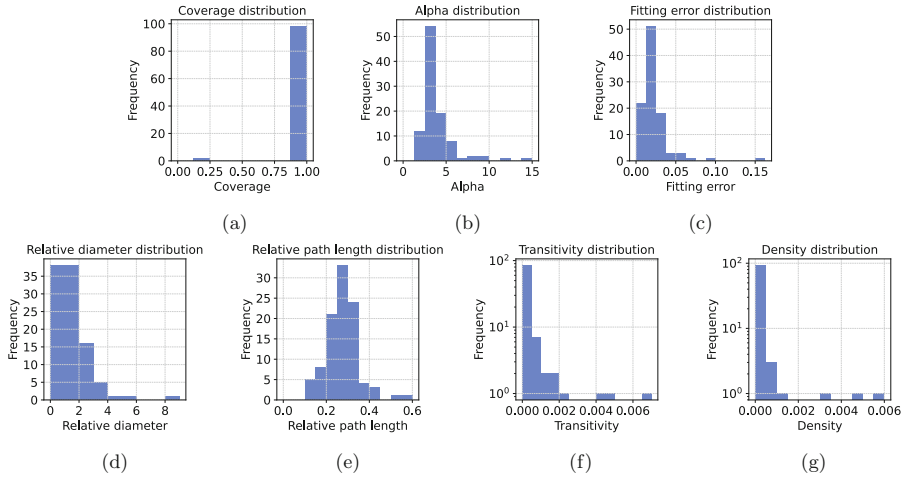


Fig. 3. Distributions of the selected features.

4.4 Clustering

After examining the features of each graph individually, we conducted another analysis employing clustering techniques. The goal of this analysis is to identify groups of contracts with similar topological properties. For our initial experiment, we attempted to identify which subset of the features described in Sect. 4.3 yields the best clustering. To achieve this aim, we employed the K-means algorithm, testing all possible feature subsets while varying the number of clusters k from a minimum of 2 to a maximum of 20. For each subset, we then selected

the value of k maximizing the silhouette coefficient. We note that, with 7 different features, the number of valid subsets is equal to 127. Each subset then generates 19 possibilities, resulting in a total of 2 413 combinations. Figures 4a, 4b and 4c illustrate, respectively, the top three clusterings obtained with this approach, namely those with the highest silhouette scores. As the reader may notice, all three configurations comprise $k = 2$ clusters. The first configuration, with a silhouette of 0.945, was obtained using only the coverage feature. The second configuration, which returned a score of 0.834, was obtained using only the density feature. Finally, the third configuration was obtained by combining both features together, yielding a silhouette of 0.785. We observe that, in all three cases, the obtained clusterings are highly imbalanced. Indeed, we can always find a small cluster, containing no more than 20 elements, and a large cluster, with more than 80 elements.

To attempt a different clustering approach, we also conducted further analysis based on dimensionality reduction. In particular, we used principal component analysis to reduce the number of features and then executed the K-means algorithm on this reduced data set. Before applying the dimensionality reduction, however, we used the explained variance ratio method to determine the optimal number of components. More precisely, we set a threshold of 0.8 (to keep 80% of the total variance of the original data) and selected the minimum number of principal components such that the explained variance ratio is above the threshold. In this regard, the plot of Fig. 4d illustrates the total explained variance ratio as the number of components varies. As the reader may notice, it appears that the optimal number of features is equal to 4. We then applied the K-means algorithm again to the reduced data set, trying values of k ranging from 2 to 20. As before, among the 19 configurations tested, we chose the one that maximized the silhouette score. As shown in Fig. 4e, the maximum silhouette value (slightly above 0.7) is achieved, once again, for $k = 2$ clusters. The corresponding clustering for this configuration is described by the plot of Fig. 4f: it can be observed that this partitioning is highly unbalanced, with 97 contracts assigned to the first cluster and only 3 elements to the second one.

Considering the difficulty encountered in separating contracts according to the associated features, we introduced a new classification based on contract semantics. Specifically, we manually assigned to each contract a categorical label describing its main application domain. The ultimate goal of this analysis was to study the composition of the obtained clusters, in order to determine whether similar graphs correspond to contracts with similar purposes. In this regard, we identified nine token categories: (1) *defi* comprises all tokens related to decentralized finance (e.g., stablecoins, wrapped tokens, tokens issued by exchanges and automated market makers, etc.); (2) *games* includes all token related to games; (3) *blockchain* denotes all tokens related to independent blockchain projects; (4) *layer-2* contains tokens related to layer-2 solutions aimed at improving the scalability of Ethereum; (5) *content* includes reward tokens related to content creation platforms; (6) *storage* represents all tokens related to decentralized storage solutions; (7) *mining* indicates tokens associated with cryptocurrency mining

services; (8) *multimedia* comprises all tokens related to multimedia content (e.g., music, video streaming services, etc.); (9) *other* comprises all tokens whose application domain is not included into any of the previous categories. Table 2 illustrates the number of contracts for each application domain. We can notice that the most numerous category is that of tokens related to decentralized finance, comprising 54 contracts out of 100. Furthermore, 15 contracts did not fall into any of the application domains and were therefore labeled as “other”.

Table 2. Contract classification based on their application domain.

Category	Count
defi	54
other	15
games	9
blockchain	5
layer-2	4
content	4
storage	4
mining	3
multimedia	2
Total	100

We then used this labeling to measure clustering *homogeneity*. Homogeneity quantifies, on a scale from 0 to 1, how much each cluster predominantly contains elements belonging to a certain category of contracts [8]. We assigned a score to each clustering by comparing the labels returned by the K-means algorithm with our manually-assigned categories. To better understand how the clustering reflects such categories, we have focused on clustering results with $k = 8$, i.e., one cluster per category excluding the heterogeneous “other” category. In this regard, Fig. 4g reports the clustering result with $k = 8$, colored by category, yielding the maximum silhouette among all possible combinations of features. Moreover, to also illustrate the best possible division of the categories among clusters, we show in Fig. 4h the result with maximum homogeneity. Finally, in Fig. 4i we report the coloring for $k = 8$ considering the principal component analysis clustering. In all cases we can see how the semantic categories are spread among different clusters. Indeed, in Fig. 4g, despite the high silhouette score indicating a good level of cohesion among the elements within each cluster, the homogeneity of the clusters is rather low. Conversely, the configuration of Fig. 4h exhibits a higher homogeneity, but a lower silhouette score. This suggests that, while the graphs have similar topological properties, their similarity does not reflect on the application domain of the respective contracts. In other words, the topology of Transfer event graphs is not a good indicator of the semantics of the corresponding contracts.

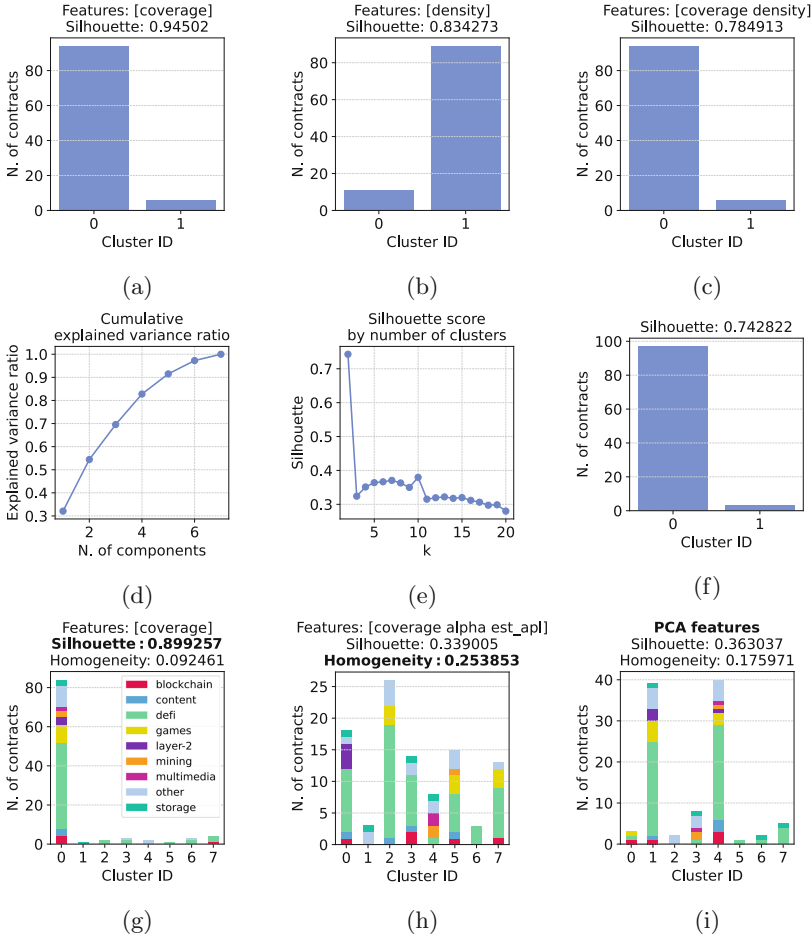


Fig. 4. Clustering analysis results (*est_apl* in figure (h) represents the relative average shortest path feature).

5 Conclusions and Future Work

In this paper we have analyzed the top 100 ERC-20 token networks by number of transfers. The study of the topological properties has revealed that – despite their diameter being of the order of the logarithm of the number of nodes – all networks exhibit a low clustering coefficient. This leads us to believe that such graphs are not small-world networks. Moreover, by analyzing the structure of the largest connected components and their degree distributions, we identified three networks that are associated with promotional tokens. Such tokens were launched through airdropping campaigns and one of them is regarded as an attempt at spamming the Ethereum network by the user community. To identify groups of networks with similar topological characteristics, we conducted a clustering

analysis and compared the results with manually-assigned labels describing the application domains of the contracts. Results suggest that a token network topology does not effectively reflect the semantics of the associated contract, meaning that contracts with similar applications can induce different network structures, and vice versa. Concerning future work, we plan to further explore the relation between contract semantics and network topology by considering additional features and different clustering methods. It could also be possible to enrich the graph with edge weights (e.g., transfer timestamp or amount). The data set might also be expanded by considering more contracts, including non-fungible ones.

References

1. Boldi, P., Vigna, S.: The webgraph framework I: compression techniques. In: Proceedings of the 13th International Conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004, pp. 595–602. ACM (2004)
2. Boldi, P., Vigna, S.: In-core computation of geometric centralities with hyperball: a hundred billion nodes and beyond. In: 2013 IEEE 13th International Conference on Data Mining Workshops, pp. 621–628. IEEE (2013)
3. Breidenbach, L., et al.: Chainlink 2.0: next steps in the evolution of decentralized oracle networks. Chainlink Labs 1, 1–136 (2021)
4. Chen, W., Zhang, T., Chen, Z., Zheng, Z., Lu, Y.: Traveling the token world: a graph analysis of Ethereum ERC20 token ecosystem. In: WWW '20: The Web Conference 2020, Taipei, Taiwan, pp. 1411–1421. ACM / IW3C2 (2020)
5. Clauset, A., Shalizi, C.R., Newman, M.E.: Power-law distributions in empirical data. *SIAM Rev.* **51**(4), 661–703 (2009)
6. Csardi, G., Nepusz, T., et al.: The Igraph software package for complex network research. *Int. J. complex syst.* **1695**(5), 1–9 (2006)
7. Lucian, A.: Scam ‘More Gold Coin’ Clogs Ethereum Network Causing Gas Price to Spike. <https://beincrypto.com/scam-more-gold-coin-clogs-ethereum-network-causing-gas-price-to-spike>, Accessed 2 Sep 2023
8. Rosenberg, A., Hirschberg, J.: V-measure: a conditional entropy-based external cluster evaluation measure. In: Proceedings of the 2007 joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 410–420 (2007)
9. Somin, S., Gordon, G., Altshuler, Y.: Network analysis of ERC20 tokens trading on Ethereum blockchain. In: Unifying Themes in Complex Systems IX: Proceedings of the Ninth International Conference on Complex Systems 9, pp. 439–450 (2018)
10. Szabo, N.: Smart contracts: building blocks for digital markets. *EXTROPY: The J. Transhumanist Thought*,(16) **18**(2), 28 (1996)
11. Victor, F., Lüders, B.K.: Measuring Ethereum-Based ERC20 Token Networks. In: Goldberg, I., Moore, T. (eds.) FC 2019. LNCS, vol. 11598, pp. 113–129. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32101-7_8
12. Vogelsteller, F., Buterin, V.: ERC-20: Token Standard (2015)
13. Wood, G.: Ethereum: a secure decentralised generalised transaction ledger (2014)

Network Geometry



Modeling the Invisible Internet

Jacques Bou Abdo^{1(✉)} and Liaquat Hossain²

¹ University of Cincinnati, Cincinnati, OH, USA
bouabdjs@ucmail.uc.edu

² University of Nebraska, Kearney, USA

Abstract. Understanding the properties of anonymity networks such as the Invisible Internet Project (Garlic router) and Tor (Onion router) is critical for the future of cybersecurity, cyberwarfare and Internet freedom. In this paper, we theoretically model the Invisible Internet and provide the preliminary components for developing a theoretical lens that can be used to address its open questions. Additionally, this work lays the theoretical foundation for studying I2P's key network properties such as resilience, anonymity and minimum attackers/routers ratio to exploit the network. The model was validated against a simulated I2P network.

Keywords: Invisible Internet Project · Garlic Router · Onion Router · Tor Network · Random graph

1 Introduction

Peer-to-peer overlay networks are becoming extremely important as they constitute the backbone that supports blockchain [7, 14], cryptocurrencies [20, 27] and the dark web [8]. The Invisible Internet Project (Garlic Router) [5] and Tor (Onion Router) [11], two overlay anonymity networks, are at the forefront of fighting censorship [15, 16], facilitating whistleblowing [10, 22] and supporting anonymous communication [9, 13].

It is becoming extremely important to study anonymity networks such as Invisible Internet Project (I2P) and Tor for multiple reasons. Firstly, social causes such as whistleblowing and censorship evading, which are very important to the society [21], rely on hidden services and anonymous communication. Secondly, cyberattackers leverage anonymity networks to obfuscate and anonymize bot-master and master-attacker connections [23]. Accordingly, formalizing cyber attribution requires understanding anonymity networks. Lastly, I2P and Tor have unique network characteristics [19]. This deprives the community from using existing theoretical lenses [6] to understand I2P and Tor's network properties.

One of I2P's open research questions is as follows "Is there a way that I2P could perform peer selection more efficiently or securely?" [3]. The peer selection process is the seed that emerges into I2P's network structure. This structure governs key network properties such as resilience and obfuscation, which are

essential to both censorship and attribution evasion. Before proposing a more efficient peer selection mechanism, we should measure the efficiency of the current one, which is still an open question especially due to the lack of a suitable theoretical framework. In this paper, we model the I2P network and develop a basic theoretical framework for predicting I2P's network structure. The motivation behind this work is modeling the I2P and laying the theoretical foundation for studying its network's properties such as resilience, obfuscation and minimum attackers/routers ratio to exploit the network. The main contributions of this work are as follows:

1. Develop a simple recursive description of weighted node selection probability with replacement. This description facilitates modeling anonymity networks with weighted peer selection mechanism.
2. Model I2P's degree distribution, which allows us to predict the number of active links as a function of network parameters.
3. Validate the correctness of the proposed model against simulated I2P network.

The remaining of this paper is organized as follows, Sect. 2 introduces I2P's peer selection mechanism. Section 3 develops a simple recursive description of node selection probability under weighted random sampling. Section 4 builds on top of Sect. 3 to develop node's probability of joining a tunnel. Section 5 builds the I2P's theoretical network structure model. Section 6 validates the model against simulated I2P and Sect. 7 concludes the work.

2 Background: Descriptive Explanation of Garlic Routing and I2P Tunneling

I2P is a full mesh overlay P2P network composed of nodes, called routers. Even though a router is theoretically connected to all other routers in the network, only a small number of connections are active at any point in time. Figure 1(a) shows a full mesh network which is the very abstract representation of I2P. The blue circles represent the I2P routers and the light blue lines represent the inactive neighboring connections.

Generally speaking, a router establishes a multi-router tunnel to act as a multi-layer Mix-Net for ensuring anonymization [11]. Every I2P router establishes multiple tunnels, each for a different purpose as will be discussed later. Let's consider for now, that a router wants to establish one tunnel, an outbound one. This router will be the first node in the tunnel and will be called *Outbound Gateway (OG)* [5]. The tunnel will then include a certain number of routers called *Outbound Participants (OP)*. A tunnel supports between 0 and 5 *Outbound Participants*, but is defaulted to 1 [1]. The tunnel ends with a router called *Outbound Endpoint (OE)*. Figure 1(b) shows a full mesh I2P network. The blue circles represent the I2P routers and the light blue lines represent the inactive neighboring connections. The yellow router represents the *Outbound Gateway*, the green router represents the *Outbound Participant* and the red router represents the *Outbound Endpoint*. The solid blue lines are the

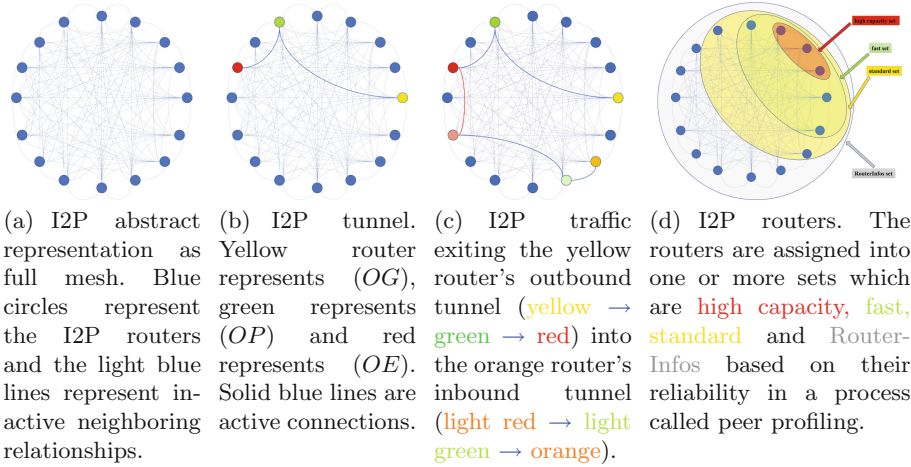


Fig. 1. I2P network representation (Color figure online)

active connections. Accordingly, a router's active connections are due to his tunnels and the tunnels he is assisting. In addition to the outbound tunnel, I2P routers establish inbound tunnels; since tunnels are unidirectional. The inbound tunnel starts with a router called *Inbound Gateway*, followed by 0 to 5 (default is 1) routers called *Inbound Participant* and ends with the router establishing the tunnel called *Inbound Endpoint*. Figure 1(c) shows a full mesh I2P network where the yellow I2P router sending information to the orange router. To ensure anonymity, the information is not sent directly to the orange router, but exits from yellow router's outbound tunnel and then enters through the orange router's inbound tunnel. The orange router is thus the *Inbound Endpoint*. The remaining routers in the inbound tunnel are *Inbound Gateway* indicated in light red and *Inbound Participant* indicated in light green. The traffic exiting the outbound tunnel and entering the inbound tunnel is colored in red. It is worth noting that for the orange router to respond back, he has to use another tunnel (orange router's outbound tunnel) and send it to the yellow router's inbound tunnel. A two-way communication needs two pairs of tunnels. It is also worth noting that the same outbound tunnel can be used to communicate with multiple inbound tunnels, until the outbound tunnel is dismantled. Not all routers have the same chance of joining a tunnel [4]. Routers are profiled based on their performance over the last 7 days and assigned into one of three overlapping sets which are:

- **high capacity**: this is a set of the 30 most reliable routers.
- **fast**: this is a set of the 75 most reliable routers (including those in “high capacity”).
- **standard**: this is a set of least reliable or recently joined routers. This set does not have a limit but contains typically around 500 routers (including those in “fast”).

All routers, including “standard set” are tracked using “RouterInfos”, which is stored in the local network database. We can consider “RouterInfos” as the universe of all active routers in I2P network at any moment [4]. Figure 1(d) shows a full mesh I2P network with the routers assigned to one or more sets which are high capacity (Red), fast (green), standard (yellow) and RouterInfos (gray). Any router, in the RouterInfos set, creates two types of tunnels, communication and exploratory. The communication tunnels are made of inbound and outbound tunnels. Each of those tunnels survive for a short period of time (usually 1 minute), before being dismantled and replaced by another tunnel. The routers participating in a tunnel are randomly selected from the ”high capacity” set. There are some restrictions on the selection, but it is generally random. The exploratory tunnels are created by selecting routers from the ”standard set”. In case the initiating router was not able to find reliable routers for his tunnel, routers from the ”fast” set are also selected using a weighted random selection process.

3 Weighted Random Sampling Without Replacement

Weighted random sampling is still an open question [26] for statisticians [18], computer scientists [12,17,24], mathematicians and network scientists [25]. Weighted random sampling with replacement is trivially easy, while weighted random sampling without replacement is still, to the best of our knowledge, not yet solved analytically and approached computationally as shown in [26]. This section aims at developing a simple recursive description of node selection probability, which is I2P’s aspect of interest in weighted random sampling without replacement.

Consider two sets A and B , with sizes N_a and N_b respectively. The elements in A are equiprobable and the elements in B are equiprobable as well, but the chance of selecting an element from A and B is a and b respectively. Let $a, b \in \mathbb{N}^+$.

Corollary 1. *The probability of selecting node n_a from A , n_b from B , a node from A and from B is respectively:*

$$P(n = n_a, a, b, N_a, N_b) = a/(aN_a + bN_b), \quad P(n = n_b, a, b, N_a, N_b) = b/(aN_a + bN_b)$$

$$P(n \in A, a, b, N_a, N_b) = aN_a/(aN_a + bN_b), \quad P(n \in B, a, b, N_a, N_b) = bN_b/(aN_a + bN_b)$$

Theorem 1. *The probability of selecting node n_a in a sample out of x weighted random selections without replacement, where $n_a \in A$, is the x^{th} unit of geometric recursive sequence:*

$$P(n_a \in S_x) = P_r(x, a, b, N_a, N_b) = P(n \in B, a, b, N_a, N_b) \times P_r(x - 1, a, b, N_a, N_b - 1) + \frac{a}{aN_a + bN_b} + \frac{(N_a - 1)}{N_a} \times P(n \in A, a, b, N_a, N_b) \times P_r(x - 1, a, b, N_a - 1, N_b) \quad (1)$$

where $P_r(0, a, b, N_a, N_b) = 0$, $x \leq N_a$ and $x \leq N_b$

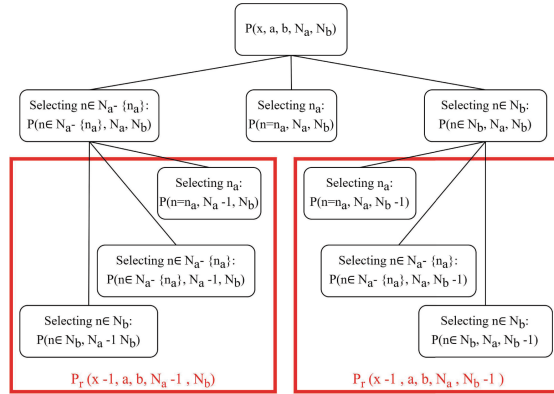


Fig. 2. Graphical description of weighted sampling without replacement.

Let S_x be a weighted randomly selected sample from $A \cup B$ with replacement. Let n be any element in $A \cup B$ other than n_a i.e. $(n \in A \cup B - \{n_a\})$. Then

$$P(n_a \in S_x) = P(n_a, n, \dots, n) \dots + P(n, n, \dots, n_a) = P(n_a, n, \dots, n) + P(n) \times P(n_a \in S_{x-1}) \\ = P(n = n_a) + P(n) \times P(n_a \in S_{x-1})$$

Let $P(n_a \in S_x)$ be defined as $= P_r(x, a, b, N_a, N_b)$. Then

$$P(n_a \in S_{x-1}) = \begin{cases} P_r(x-1, a, b, N_a, N_b-1), & n \in B \\ P_r(x-1, a, b, N_a-1, N_b), & n \in A' \end{cases}$$

where $A' = A - \{n_a\}$. Since S_x is a weighted random sample, then

$$P(n_a \in S_x) = P(n = n_a) + P(n \in B, N_a, N_b) \times P_r(x, a, b, N_a, N_b-1) \\ + P(n \in A - \{n_a\}, N_a, N_b) \times P_r(x, a, b, N_a-1, N_b)$$

Accordingly, $P(n_a \in S_x) = P(x, a, b, N_a, N_b) = \frac{1}{aN_a + bN_b} \times (a$
 $+ bN_b \times P_r(x-1, a, b, N_a, N_b-1) + a(N_a-1) \times P_r(x-1, a, b, N_a-1, N_b))$

The proof of Theorem 1 is visualized in Fig. 2.

4 Tunnel Formation in Complete Graph

Considering two complete homogeneous graphs Γ and Ω of sizes N_γ and N_ω respectively, where $N_\gamma > 2$ and $N_\omega > 2$. Let γ and ω be the set of nodes in Γ and Ω respectively, defined as $\gamma = \{n_1, n_2, \dots, n_{N_\gamma}\}$ and $\omega = \{n_1, n_2, \dots, n_{N_\omega}\}$ respectively. Let $\gamma \subset \omega$ and thus Γ a subgraph of Ω . Let, for the sake of simplicity, $\omega \cap \bar{\gamma}$ be denoted as $\bar{\gamma}$, which is considering ω to be the universe.

Let T be a tunnel, which is a linear chain graph, starting with node n_s , where $n_s \in \omega$, and followed by H unique nodes randomly selected from γ . Let τ be the set of nodes in T .

Corollary 2. *The cardinality of γ , ω and τ are respectively:*

$$Card(\gamma) = N_\gamma; \quad Card(\omega) = N_\omega; \quad Card(\tau) = H + 1$$

Lemma 1. *The probability of a node n_i to be part of τ , knowing that $n_i \in \gamma$ is:*

$$P(n_i \in \tau \mid n_i \in \gamma) = \frac{H \times N_\omega + N_\gamma}{N_\omega \times N_\gamma} \tag{2}$$

$$\begin{aligned} P(n_i \in \tau \mid n_i \in \gamma) &= P(n_i \in \tau \mid n_i \in \gamma \ \& \ n_s \in \bar{\gamma}) \times P(n_s \in \bar{\gamma}) \\ &+ P(n_i \in \tau \mid n_i \in \gamma \ \& \ n_s \in \gamma \ \& \ n_s = n_i) \times P(n_s \in \gamma \ \& \ n_s = n_i) \\ &+ P(n_i \in \tau \mid n_i \in \gamma \ \& \ n_s \in \gamma \ \& \ n_s \neq n_i) \times P(n_s \in \gamma \ \& \ n_s \neq n_i) \\ &= \frac{H}{N_\gamma} \times \frac{N_\omega - N_\gamma}{N_\omega} + 1 \times \frac{1}{N_\omega} + \frac{H}{N_\gamma - 1} \times \frac{N_\gamma - 1}{N_\omega} = \frac{H \times N_\omega + N_\gamma}{N_\omega \times N_\gamma} \end{aligned}$$

Each of the tunnel’s terminal nodes (*Gateway* and *Endpoint*) engage in 1 tunnel-related link, but the intermediate nodes (*Participant*) engage in 2 tunnel-related links. Accordingly, the sum of degrees resulting from one tunnel is $2 \times (H)$.

Corollary 3. *The expected tunnel-related degree of a node n_i to be part of τ , knowing that $n_i \in \gamma$ is:*

$$Deg_{link}(n_i \in \tau \mid n_i \in \gamma) = \frac{H_l \times N_\omega + N_\gamma}{N_\omega \times N_\gamma} \tag{3}$$

where $H_l = 2 \times H - 1$. Following Lemma 1,

$$Deg_{link}(n_i \in \tau \mid n_i \in \gamma) = \frac{2 \times H - 1}{H} \times \left(\frac{H}{N_\gamma} \times \frac{N_\omega - N_\gamma}{N_\omega} + \frac{H}{N_\omega} \right) + 1 \times \frac{1}{N_\omega}$$

Lemma 2. *The probability of a node n_i to be part of τ , knowing that $n_i \in \bar{\gamma}$ is:*

$$P(n_i \in \tau \mid n_i \in \bar{\gamma}) = \frac{1}{N_\omega} \tag{4}$$

$$\begin{aligned} P(n_i \in \tau \mid n_i \in \bar{\gamma}) &= P(n_i \in \tau \mid n_i \in \bar{\gamma} \ \& \ n_s \in \bar{\gamma}) \times P(n_s \in \bar{\gamma}) \\ &+ P(n_i \in \tau \mid n_i \in \bar{\gamma} \ \& \ n_s \in \gamma \ \& \ n_s = n_i) \times P(n_s \in \gamma \ \& \ n_s = n_i) \\ &+ P(n_i \in \tau \mid n_i \in \bar{\gamma} \ \& \ n_s \in \gamma \ \& \ n_s \neq n_i) \times P(n_s \in \gamma \ \& \ n_s \neq n_i) \\ &= 0 \times \frac{N_\gamma}{N_\omega} + 1 \times \frac{1}{N_\omega} + 0 \times \frac{N_\omega - N_\gamma - 1}{N_\omega} = \frac{1}{N_\omega} \end{aligned}$$

Considering two complete homogeneous graphs G_1 and G_2 of sizes N_{g_1} and N_{g_2} respectively, where $N_{g_1} > 2$ and $N_{g_2} > 2$. Let g_1 and g_2 be the set of nodes of G_1 and G_2 respectively. Let G_1 be a subgraph of G_2 which is in turn a subgraph of Ω and thus $g_1 \subset g_2 \subset \omega$.

Let T_e be a tunnel, which is a linear chain graph, starting with node n_s , where $n_s \in \omega$, and followed by H unique nodes randomly selected from g_1 and g_2 with weight α for selecting from g_1 and β for selecting from $g_2 \cap \bar{g}_1$. Let τ_e be the set of nodes in T_e .

Corollary 4. *Probability of randomly selecting a node from g_1 and g_2 with weight α for selecting from g_1 and β for selecting from $g_2 \cap \bar{g}_1$ is:*

$$P(n_i \in g_2) = \begin{cases} \frac{\alpha}{\alpha \times N_{g_1} + \beta \times (N_{g_2} - N_{g_1})} & , n_i \in g_1 \\ \frac{\beta}{\alpha \times N_{g_1} + \beta \times (N_{g_2} - N_{g_1})} & , n_i \in g_2 \cap \bar{g}_1 \end{cases} \quad (5)$$

following Corollary 1 and replacing S_x with τ_e , x with $H + 1$, a with α , b with β , N_a with N_{g_1} and N_b with $N_{g_2} - N_{g_1}$.

Following the definition of τ_e , we can conclude that:

Corollary 5. *Probability of node n_i to be part of τ_e , knowing that $n_i \in \omega \cap \bar{g}_2$ is:*

$$P(n_i \in \tau_e \mid n_i \in \omega \cap \bar{g}_2) = \frac{1}{N_\omega} \quad (6)$$

Similar to Lemma 2. τ_e formation is a weighted random sampling without replacement, as described in Sect. 2. We can conclude, following Theorem 1 and Lemma 1, that:

Lemma 3. *Probability of node n_i to be part of τ_e , knowing that $n_i \in g_1$ is:*

$$P(n_i \in \tau_e \mid n_i \in g_1) = \frac{1}{N_\omega} (1 + (N_{g_1} - 1) \times \Delta_{g_1}(-1, 0)) \\ + (N_{g_2} - N_{g_1}) \times \Delta_{g_1}(0, -1) + (N_\omega - N_{g_2}) \times \Delta_{g_1}(0, 0) \quad (7)$$

where $\Delta_{g_1}(i, j) = P_r(H, \alpha, \beta, N_{g_1} - i, N_{g_2} - N_{g_1} - j)$

Corollary 6. *The expected tunnel-related degree of a node n_i to be part of τ_e , knowing that $n_i \in g_1$ is:*

$$Deg_{link}(n_i \in \tau_e \mid n_i \in g_1) = \left(P(n_i \in \tau_e \mid n_i \in g_1) - \frac{1}{N_\omega} \right) \times \left(\frac{2H - 1}{H} \right) + \frac{1}{N_\omega} \quad (8)$$

Similar to Corollary 3

Symmetrically, this discussion can be extended to $g_2 \cap \bar{g}_1$ by switching α with β and N_{g_1} with $N_{g_2} - N_{g_1}$. Accordingly, we can conclude that:

Lemma 4. *Probability of node n_i to be part of τ_e , knowing that $n_i \in g_2 \cap \bar{g}_1$ is:*

$$P(n_i \in \tau_e \mid n_i \in g_2 \cap \bar{g}_1) = \frac{1}{N_\omega} (1 + (N_\omega - N_{g_2}) \times \Delta_{g_2}(0, 0)) \\ + (N_{g_1}) \times \Delta_{g_2}(0, -1) + (N_{g_2} - N_{g_1} - 1) \times \Delta_{g_2}(-1, 0) \quad (9)$$

where $\Delta_{g_2}(i, j) = P_r(H, \beta, \alpha, N_{g_2} - N_{g_1} - i, N_{g_1} - i)$

Corollary 7. *The expected tunnel-related degree of a node n_i to be part of τ_e , knowing that $n_i \in g_2 \cap \bar{g}_1$ is:*

$$Deg_{link}(n_i \in \tau_e \mid n_i \in g_2 \cap \bar{g}_1) = \left(P(n_i \in \tau_e \mid n_i \in g_2 \cap \bar{g}_1) - \frac{1}{N_\omega} \right) \times \frac{2H - 1}{H} + \frac{1}{N_\omega} \quad (10)$$

Similar to Corollary 3

5 I2P Network Structure

Each router is theoretically connected to all other routers, but maintains a smaller set of active connections. Those connections are due to the tunnels formed by the router and those tunnels the router is invited to join, as discussed in Sect. 2. It was shown in Sect. 4 that a router is invited to join a tunnel with a probability proportional to the set it belongs to. We can thus predict that a router’s degree (number of active connections) is proportional to the set it belongs to. I2P assigns each router to one of its 4 sets (and indirectly to the larger sets) which are: high capacity \subset fast \subset standard \subset RouterInfos, as discussed in Sect. 2. Those sets were resembled in Sect. 4 as $\gamma \subset g_1 \subset g_2 \subset \omega$.

Each router creates two types of tunnels, as presented in Sect. 2, where each type has at least one inbound and one outbound tunnels. Routers manage tunnels using tunnel pools, but let’s consider for now that each router has four different tunnels: inbound communication, outbound communication, inbound exploratory and outbound exploratory. Accordingly, the I2P network has $2 \times N_\omega$ communication tunnels and $2 \times N_\omega$ exploratory tunnels. We can thus calculate the expected degree of each router.

Lemma 5. *The degree of a router in set “high capacity” ($n_i \in \gamma$) is expected to be:*

$$\begin{aligned}
 Deg(n_i \in \gamma) = & 4 + 2HN_\omega + \frac{4H - 2}{H}((N_{g1} - 1) \times \Delta_{g1}(-1, 0) \\
 & + (N_{g2} - N_{g1}) \times \Delta_{g1}(0, -1) + (N_\omega - N_{g2}) \times \Delta_{g1}(0, 0)) \quad (11)
 \end{aligned}$$

The degree of node n_i (number of active connections), where $n_i \in \gamma$, is:

$$\begin{aligned}
 Deg(n_i \in \gamma) = & Card(\tau) \times Deg_{link}(n_i \in \tau \mid n_i \in \gamma) \\
 & + Card(\tau_e) \times Deg_{link}(n_i \in \tau_e \mid n_i \in \gamma)
 \end{aligned}$$

$Card(\tau) = Card(\tau_e) = 2 \times N_\omega$.

$Deg_{link}(n_i \in \tau \mid n_i \in \gamma)$ from Corollary 3, Eq. 3.

$Deg_{link}(n_i \in \tau_e \mid n_i \in \gamma) = Deg_{link}(n_i \in \tau_e \mid n_i \in g_1)$ defined in Corollary 6, Eq. 8.

Lemma 6. *The degree of a router in set “fast” ($n_i \in g_1 \cap \bar{\gamma}$) is expected to be:*

$$\begin{aligned}
 Deg(n_i \in g_1 \cap \bar{\gamma}) = & 4 + \frac{4H - 2}{H}((N_{g1} - 1) \times \Delta_{g1}(-1, 0) \\
 & + (N_{g2} - N_{g1}) \times \Delta_{g1}(0, -1) + (N_\omega - N_{g2}) \times \Delta_{g1}(0, 0)) \quad (12)
 \end{aligned}$$

The degree of node n_i (number of active connections), where $n_i \in g_1 \cap \bar{\gamma}$, is:

$$\begin{aligned}
 Deg(n_i \in g_1 \cap \bar{\gamma}) = & Card(\tau) \times Deg_{link}(n_i \in \tau \mid n_i \in g_1 \cap \bar{\gamma}) \\
 & + Card(\tau_e) \times Deg_{link}(n_i \in \tau_e \mid n_i \in g_1 \cap \bar{\gamma})
 \end{aligned}$$

$Card(\tau) = Card(\tau_e) = 2 \times N_\omega$.

$Deg_{link}(n_i \in \tau \mid n_i \in g_1 \cap \bar{\gamma}) = P(n_i \in \tau \mid n_i \in \bar{\gamma})$ defined in Lemma 2, Eq. 4.

$Deg_{link}(n_i \in \tau_e \mid n_i \in g_1 \cap \bar{\gamma}) = Deg_{link}(n_i \in \tau_e \mid n_i \in g_1)$ defined in Corollary 6, Eq. 8. Accordingly,

$$Deg(n_i \in g_1 \cap \bar{\gamma}) = (2 \times N_\omega) \times \left(\frac{1}{N_\omega} + \left(P(n_i \in \tau_e \mid n_i \in g_1) - \frac{1}{N_\omega} \right) \times \left(\frac{2 \times H - 1}{H} \right) + \frac{1}{N_\omega} \right)$$

Lemma 7. *The degree of a router in set "standard" ($n_i \in g_2 \cap \bar{g}_1$) is expected to be:*

$$Deg(n_i \in g_2 \cap \bar{g}_1) = 4 + \frac{4H - 2}{H} ((N_\omega - N_{g_2}) \times \Delta_{g_2}(0, 0) + (N_{g_1}) \times \Delta_{g_2}(0, -1) + (N_{g_2} - N_{g_1} - 1) \times \Delta_{g_2}(-1, 0)) \quad (13)$$

The degree of node n_i (number of active connections), where $n_i \in g_2 \cap \bar{g}_1$, is:

$$Deg(n_i \in g_2 \cap \bar{g}_1) = Card(\tau) \times Deg_{link}(n_i \in \tau \mid n_i \in g_2 \cap \bar{g}_1) + Card(\tau_e) \times Deg_{link}(n_i \in \tau_e \mid n_i \in g_2 \cap \bar{g}_1)$$

$Card(\tau) = Card(\tau_e) = 2 \times N_\omega$.

$Deg_{link}(n_i \in \tau \mid n_i \in g_2 \cap \bar{g}_1) = P(n_i \in \tau \mid n_i \in \bar{\gamma})$ defined in Lemma 2, Eq. 4.

$Deg_{link}(n_i \in \tau_e \mid n_i \in g_2 \cap \bar{g}_1)$ defined in Corollary 7, Eq. 10. Accordingly,

$$Deg(n_i \in g_2 \cap \bar{g}_1) = (2 \times N_\omega) \times \left(\frac{1}{N_\omega} + \left(P(n_i \in \tau_e \mid n_i \in g_2 \cap \bar{g}_1) - \frac{1}{N_\omega} \right) \times \left(\frac{2 \times H - 1}{H} \right) + \frac{1}{N_\omega} \right)$$

The degree of node n_i (number of active connections), where $n_i \in \omega \cap \bar{g}_2$, is:

$$Deg(n_i \in \omega \cap \bar{g}_2) = Card(\tau) \times P(n_i \in \tau \mid n_i \in \omega \cap \bar{g}_2) + Card(\tau_e) \times P(n_i \in \tau_e \mid n_i \in \omega \cap \bar{g}_2)$$

$Card(\tau) = Card(\tau_e) = 2 \times N_\omega$.

$P(n_i \in \tau \mid n_i \in \omega \cap \bar{g}_2) = P(n_i \in \tau \mid n_i \in \bar{\gamma})$ defined in Lemma 2, Eq. 4.

$P(n_i \in \tau_e \mid n_i \in \omega \cap \bar{g}_2)$ defined in Corollary 5, Eq. 6. Accordingly,

$$Deg(n_i \in \omega \cap \bar{g}_2) = (2 \times N_\omega) \times \frac{1}{N_\omega} + (2 \times N_\omega) \times \frac{1}{N_\omega} = 4$$

Lemma 8. *The degree of a router in set "RouterInfos" ($n_i \in \omega \cap \bar{g}_2$) is expected to be:*

$$Deg(n_i \in \omega \cap \bar{g}_2) = 4 \quad (14)$$

Theorem 2. *The degree of a router is expected to be:*

$$Deg(n_i) = \begin{cases} Deg(n_i \in \gamma), & \text{if } n_i \in \gamma \\ Deg(n_i \in g_1 \cap \bar{\gamma}), & \text{if } n_i \in g_1 \cap \bar{\gamma} \\ Deg(n_i \in g_2 \cap \bar{g}_1), & \text{if } n_i \in g_2 \cap \bar{g}_1 \\ Deg(n_i \in \omega \cap \bar{g}_2), & \text{if } n_i \in \omega \cap \bar{g}_2 \end{cases} \quad (15)$$

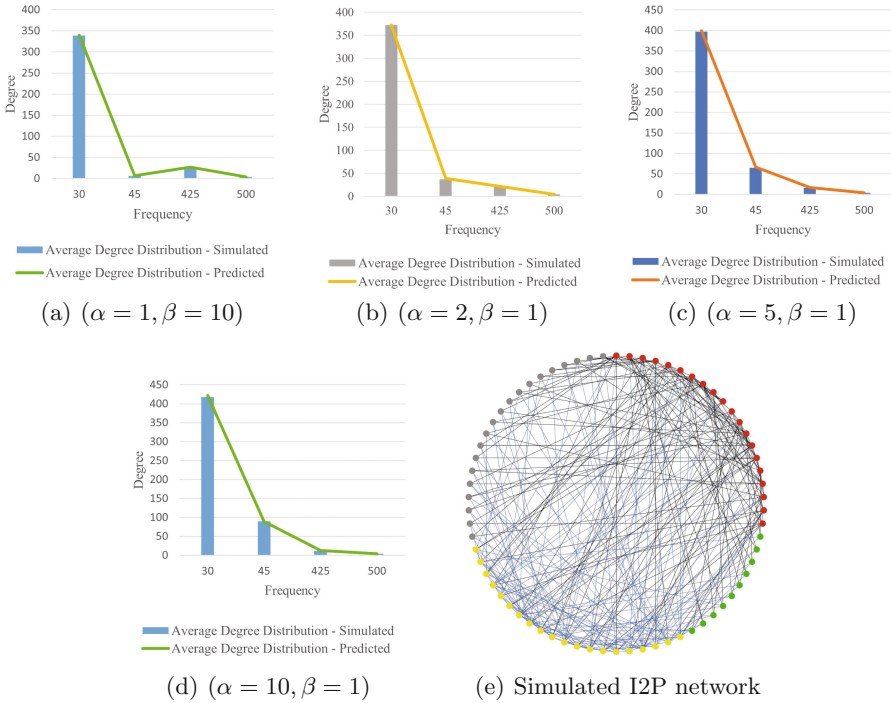


Fig. 3. Validating I2P Model vs. Simulation. Figures (a)-(d) represent simulated average degree distribution (bar), per node type, vs. predicted (line). Network configuration: $(N_\omega = 1000, N_{g2} = 500, N_{g1} = 75, N_\gamma = 30, H = 3)$. Figure (e) illustrates the simulated I2P network where red, green, yellow and gray represent high capacity, fast, standard and RouterInfo sets respectively. Black and blue links represent communication and exploratory respectively.

6 Validation

The aim of this section is to validate the developed model and its ability to describe the I2P network that emerges from the current peer selection mechanism. First, we simulated I2P’s peer selection mechanism and its emerging network structure using Netlogo, where the code is shared [2] for result reproducibility and to help researchers visualize I2P’s expected structure. Figure 3(e) shows a simulated I2P network that emerged from the current peer selection mechanism. The routers are colored following Fig. 1(d) where red, green, yellow and gray represent high capacity, fast, standard and RouterInfo sets respectively. It is worth noting that this network was configured as follows: $(N_\omega = 70, N_{g2} = 50, N_{g1} = 30, N_\gamma = 20, H = 1, \alpha = 1, \beta = 1)$. This network does not represent the real network sizes presented in Sect. 2, but has been modified for visibility. It is also worth noting that the black links represent communication tunnels while the blue links represent the exploratory tunnels.

Figure 3 shows the simulated average degree distribution (per node type) vs. the I2P model described in Theorem 2. The bars represent averaged degree distribution of the simulated I2P network, while the lines present that predicted by the I2P model. It can be seen that the model was able to accurately predict the I2P network, which emerged from the current peer selection mechanism. The model has been tested (against the simulated I2P network) for a wide range of configurations and was able, in each time, to accurately predict the network structure. The remaining results were not added due to lack of space.

7 Conclusion

This work has laid the theoretical foundation for studying the I2P network through modeling its degree distribution and validating the results. This work should help in understanding I2P's key network characteristics such as resilience, obfuscation and minimum attackers/routers ratio to exploit the network. This work should also help in measuring current peer selection mechanism's accuracy as a prerequisite for answering I2P's major open research questions.

References

1. Client protocol - i2p. <https://geti2p.net/en/docs/how/tunnel-routing>
2. I2p netlogo simulation - i2p. <https://github.com/Bouabdoj/I2P-modeling>
3. Open research questions - i2p. <https://geti2p.net/en/research/questions>
4. Peer profiling and selection - i2p. <https://geti2p.net/en/docs/how/peer-selection>
5. Tunnel routing - i2p. <https://geti2p.net/en/docs/how/tunnel-routing>
6. Barabási, A.L.: Scale-free networks: a decade and beyond. *science* **325**(5939), 412–413 (2009)
7. Bou Abdo, J., El Sibai, R., Demerjian, J.: Permissionless proof-of-reputation-x: a hybrid reputation-based consensus algorithm for permissionless blockchains. *Trans. Emerg. Telecommun. Technol.* **32**(1), e4148 (2021)
8. Bradbury, D.: Unveiling dark web. *Netw. secur.* **2014**(4), 14–17 (2014)
9. Cilleruelo, C., De-Marcos, L., Junquera-Sánchez, J., Martínez-Herraiz, J.J.: Interconnection between darknets. *IEEE Internet Comput.* **25**(3), 61–70 (2020)
10. Dingleline, R.: Tor hidden services. *Proc. What the Hack* (2005)
11. Dingleline, R., Mathewson, N., Syverson, P.: Tor: the second-generation onion router. *Tech. rep.*, Naval Research Lab Washington DC (2004)
12. Efrimidis, P.S., Spirakis, P.G.: Weighted random sampling with a reservoir. *Information processing letters* **97**(5) (2006)
13. Gehl, R.W.: *Weaving the dark web: legitimacy on freenet, Tor, and I2P*. MIT Press (2018)
14. Hegde, P., de Veciana, G.: Performance and efficiency tradeoffs in blockchain overlay networks. In: *Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pp. 221–230 (2022)
15. Hoang, N.P., Doreen, S., Polychronakis, M.: Measuring i2p censorship at a global scale. In: *9th USENIX Workshop on Free and Open Communications on the Internet (FOCI 19)*. USENIX Association, Santa Clara, CA (2019). <https://www.usenix.org/conference/foci19/presentation/hoang>

16. Hoang, N.P., Kintis, P., Antonakakis, M., Polychronakis, M.: An empirical study of the i2p anonymity network and its censorship resistance. In: Proceedings of the Internet Measurement Conference 2018, pp. 379–392 (2018)
17. Hübschle-Schneider, L., Sanders, P.: Parallel weighted random sampling. *ACM Trans. Math. Softw. (TOMS)* **48**(3), 1–40 (2022)
18. Koyuncu, N., Kadilar, C.: Calibration weighting in stratified random sampling. *Commun. Stat.-Simul. Comput.* **45**(7), 2267–2275 (2016)
19. Li, M., Lee, W.C., Sivasubramaniam, A.: Semantic small world: an overlay network for peer-to-peer search. In: Proceedings of the 12th IEEE International Conference on Network Protocols, 2004. ICNP 2004, pp. 228–238. IEEE (2004)
20. Li, Z., Xia, W., Cui, M., Fu, P., Gou, G., Xiong, G.: Mining the characteristics of the ethereum p2p network. In: Proceedings of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure, pp. 20–30 (2020)
21. Marx, G.T.: Censorship and secrecy, social and legal perspectives. *Int. Encycl. Soc. Behav. Sci.* (2001)
22. Owen, G., Savage, N.: Empirical analysis of tor hidden services. *IET Inf. Secur.* **10**(3), 113–118 (2016)
23. Sanatinia, A., Noubir, G.: Onionbots: subverting privacy infrastructure for cyber attacks. In: 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. IEEE (2015)
24. Shekelyan, M., Cormode, G., Triantafillou, P., Shanghooshabad, A., Ma, Q.: Weighted random sampling over joins. *arXiv preprint. [arXiv:2201.02670](https://arxiv.org/abs/2201.02670)* (2022)
25. Stamatelatos, G., Efraimidis, P.S.: About weighted random sampling in preferential attachment models. *arXiv preprint. [arXiv:2102.08173](https://arxiv.org/abs/2102.08173)* (2021)
26. Tillé, Y.: Remarks on some misconceptions about unequal probability sampling without replacement. *Comput. Sci. Rev.* **47**, 100,533 (2023)
27. Zeadally, S., Abdo, J.B.: Blockchain: trends and future opportunities. *Internet Technol. Lett.* **2**(6), e130 (2019)



Modeling the Dynamics of Bitcoin Overlay Network

Jacques Bou Abdo¹(✉), Shuvalaxmi Dass², Basheer Qolomany³,
and Liaquat Hossain³

¹ University of Cincinnati, Cincinnati, OH, USA
bouabdjs@ucmail.uc.edu

² University of Louisiana at Lafayette, Lafayette, LA, USA

³ University of Nebraska at Kearney, Kearney, NE, USA

Abstract. The world economy is experiencing the novel adoption of distributed currencies that are free from the control of central banks. Distributed currencies suffer from extreme volatility, and this can lead to catastrophic implications during future economic crisis. Understanding the dynamics of this new type of currencies is vital for empowering supervisory bodies to behave proactively as well-informed planners rather than reactively as incident responders. Bitcoin, the first and dominant distributed cryptocurrency, is still notoriously vague, especially for a financial instrument with market value exceeding \$1 trillion. Modeling the Bitcoin Overlay Network poses a number of important theoretical and methodological challenges. This drastically undermines the ability to predict key features such as network's resilience. In this work, we developed Evolutionary Random Graph, a theoretical model that describes the network of bitcoin miners. The correctness of this model has been validated using real and simulated bitcoin data.

Keywords: Bitcoin · Blockchain · Random graph · Scale-free networks · Evolutionary random graph · Scaling laws

1 Introduction

Scale-free networks [2, 3, 6] evolved attempting to describe the dynamics of real networks that other graphs failed to describe, mainly Erdős and Rényi's random graph [15]. New real-world information networks such as BitOverNet (Bitcoin Overlay Network), the network of miners supporting all bitcoin transactions, do not show scale-free properties and fail to be described by the graphs existing in literature [17, 22]. As indicated earlier, theorizing the network's dynamics yields important benefits and BitOverNet is not an exception. Additionally, our lack of understanding of bitcoin overlay's network dynamics is resulting in numerous problems such as forking and valuation.

To deal with those shortcomings, two approaches can be followed namely physical network measurement (probing) and network modeling. The first method only measures a snapshot of the network, fails to predict its dynamics and accordingly fails to predict the network's resilience, tipping points and other

important properties. Although being trivial, all existing probing techniques fail to measure and map the BitOverNet as shown in Sect. 2. The community does not agree on a theoretical model for BitOverNet, which is the second method, where some researchers describe it as Mandala [25] while others disapprove [18]. This work is the first, to the best of our knowledge, to successfully build a grounded mathematical model for BitOverNet. It is worth noting that this work focuses on BitOverNet not “Bitcoin Transaction Network” which was studied in [24, 26].

The motivation behind this work is modeling the BitOverNet and identifying its key properties and thus laying the theoretical foundation for studying the network’s properties such as resilience and valuation. The main contributions of this work are as follows:

1. Show that scale-free networks and other network models (graphs) found in literature are not able to describe the BitOverNet.
2. Propose evolutionary random graph, show that it describes the BitOverNet and show that it predicts the properties of the BitOverNet. This is the first model to describe and predict the properties of BitOverNet.
3. Show the correctness of proposed model in predicting the current properties of the BitOverNet. The model was tested against real collected data.
4. Deduce BitOverNet’s key graph properties and lay the theoretical foundation for studying the network’s key properties.

In Sect. 2 we will survey existing physical network measurement techniques (probing) developed for bitcoin. We will model the probability density function followed by the BitOverNet in Sect. 3. We will also propose the graph that describes the BitOverNet and name it “Evolutionary Random Graph”. Based on “Evolutionary Random Graph”, we will calculate key graph properties in Sect. 4 and validate its adequacy in predicting the BitOverNet in Sect. 5. Finally, Sect. 6 concludes this work.

2 Related Work

Lischke and Fabian [19] analyzed the public transaction network history of the first four years of Bitcoin with respect to economic and network aspects by introducing benchmark data. Their analysis showed that large portions of the network follow a power law distribution and can be considered as scale-free networks. It is worth noting that their study focuses on Bitcoin transaction network, which is a different layer than BitOverNet. Deshpande et al. [11] developed a framework named BTCmap to discover and map the Bitcoin network topology. The framework includes two modules, sniffer that communicate with real peers; and Bitcoin peer emulator for outbound neighbors’ selection and generate the topology. Their analysis showed that the online peers list remains valid during 56 min 40 s. Within this duration, BTCmap requested more than 8200 reachable peers to map the visible part of the real Bitcoin network topology. Neudecker and Hartenstein [21] reviewed different attacks on the network layer of permissionless blockchains [7, 28]. They showed that there is a lack of models that

analyze and formalize the tradeoffs of most network design decisions of permissionless blockchains. They emphasized that simulation-based approaches could cope with these limitations and are suited for the analysis of the network layer of permissionless blockchains.

Delgado-Segura et al. [10] presented TxProbe, a technique for reconstructing the Bitcoin network topology. They validated their reconstructing topology technique on Bitcoin testnet and showed that the precision and recall surpassing 90%. Essaid et al. [16] proposed a real-time Bitcoin-based topology discovery system for Bitcoin P2P links with the use of a customized version of the Page-Rank algorithm that can determine in real-time which nodes require deeper graph analysis. Ben Mariem et al. [4] presented a Bitcoin crawler that is able to discover and track all the active nodes of the BTC P2P network and use it to analyze and characterize the BTC network topology and main properties from a purely network measurements-based approach. Eisenbarth et al. [14] presented an open measurement dataset on the Bitcoin p2p network. They assessed their crawler soundness and made it available with the used scripts to perform analysis to provide facilities to reproduce and extend the study.

Donet et al. [13] presented an analysis of the collected data of the decentralized P2P network identifying more than 872000 different Bitcoin nodes. The analysis showed that the Bitcoin P2P network is homogeneously spread all over the world, with some exceptions on very low populated areas and underdeveloped countries. Park et al. [23] presented a comparative measurement study of nodes in the Bitcoin network by scanning the live Bitcoin network for 37 d in 2018 and compare them with the data reported by prior work in 2013–2016. Their measurements showed that there are approximately 1 million users in the Bitcoin networks, but only around 8500–23000 are full node peers that participate in information propagation. Miller et al. [20] introduced AddressProbe, a technique that discovers peer-to-peer links in Bitcoin, and apply this to the live topology, within the discovered topology, they found “influential” nodes that appear to directly interface with a hidden topology that consists of mining pools that are otherwise not connected to the public Bitcoin network.

The surveyed research fails in mapping the BitOverNet because of the reliance on physical probing techniques which are limited to the visible part of the network. Additionally, even if physical mapping is successful, the resultant empirical measurement lacks generalizability and is limited to one instance of the network (captured at one moment in time).

3 Evolutionary Random Graph

BitOverNet can be modeled as having N connected nodes, when node $(N + 1)$ gets introduced, it gets connected to each other node with the same probability $p_{(N+1)} = \frac{p}{N}$, as shown in Algorithm 1. It is similar to random networks in the way that a new node connects equiprobably to all existing nodes, but as more nodes join the network, the probability of connection decreases. In bitcoin, each new node has to establish a fixed number of outgoing connections called m . It is

Algorithm 1. Node addition in Evolutionary Random Graph

```

1: procedure ADD-NODE-EVOL-RANDOM( $N+1$ )
2:
3:   for  $i \in [1, n]$  do
4:     if ( $\text{random}(0, 1) < p/N$ ) then
5:        $\text{link}(i, N + 1)$  ▷ link node i to the new node
6:     end if
7:   end for
8: end procedure

```

worth noting that for implementation purposes, every released “Bitcoin Core” version has a hard-coded list of IP addresses that were available during the time the specific version was released [12]. This list does not create any guarantees about the availability of those addresses, else this centrality results in a fatal flaw. We propose “Evolutionary Random” graph as a theoretical model that describes how BitOverNet behaves, as shown in Algorithm 1.

Lemma 1. *After the N_{th} node join the network, node i should have received in average Λ_i links, where:*

$$\Lambda_i = \max(m - i, 0) + m \times (H_{N-1} - H_{\max(m-1, i-1)}) \tag{1}$$

where H is the harmonic number. Let $m, x, i, N \in \mathbb{N}$ where:

- m is the fixed number of neighboring connections initiated by a new node
- x is the network size (number of nodes already in the network)
- i is the node’s arrival index (number of nodes in the network before node i)
- N is the total number of nodes to join the network (final network size)

For every new node joining the network, an existing node will receive a neighboring request from the new node with probability equals to:

$$P = \begin{cases} 1 & x \leq m, \\ \frac{m}{x-1} & x > m \end{cases}$$

If node i arrived, where $i \geq m$, and then two more nodes arrived after it, then node i will receive in average $\frac{m}{i-1} + \frac{m}{i}$ incoming neighboring requests. The number of incoming neighboring requests node i will receive, in average, when all the N nodes arrive is:

$$\Lambda_i = \begin{cases} \sum_{x=i+1}^m 1 + \sum_{x=m+1}^N \frac{m}{x-1} & i < m, \\ \sum_{x=i}^N \frac{m}{x-1} & i \geq m \end{cases}$$

$$\implies \Lambda_i = \max(m - i, 0) + \begin{cases} m \times \sum_{x=m+1}^N \frac{1}{x-1} & i < m, \\ m \times \sum_{x=i+1}^N \frac{1}{x-1} & i \geq m \end{cases}$$

$$\begin{aligned} \implies \Lambda_i &= \max(m - i, 0) + \begin{cases} m \times (H_{N-1} - H_{m-1}) & i < m, \\ m \times (H_{N-1} - H_{i-1}) & i \geq m \end{cases} \\ \implies \Lambda_i &= \max(m - i, 0) + m \times (H_{N-1} - H_{\max(m-1, i-1)}) \end{aligned}$$

Corollary 1. *The number of incoming neighboring requests, node i receives, is a random variable with probability mass function:*

$$P(x = k, i) = \frac{\Lambda_i^k}{k! \times e^{\Lambda_i}} \tag{2}$$

The incoming neighboring requests, a node receives, are discrete consecutive random events and thus the number of incoming neighboring requests is a discrete random variable following Poisson distribution.

Theorem 1. *Evolutionary Random Graph’s degree distribution is:*

$$P(k) = \sum_{i=1}^N \frac{\Lambda_i^k}{k! \times e^{\Lambda_i}} \tag{3}$$

The probability a node receives k incoming neighboring requests depends on i , which is the node’s arrival index. Each node’s probability mass function is a shifted version of the others and thus the graph’s probability mass function is the average of the shifted probability mass functions (shifted by i). Accordingly, the number of incoming neighboring requests (incoming links) a node receives is a random variable with probability mass function:

$$P_k = \frac{1}{N} \times \sum_{i=1}^N P(x = k, i) = \frac{1}{N} \times \sum_{i=1}^N \frac{\Lambda_i^k}{k! \times e^{\Lambda_i}}$$

Accordingly, the degree distribution is: $P(k) = \sum_{i=1}^N \frac{\Lambda_i^k}{k! \times e^{\Lambda_i}}$

To measure the accuracy of our bitcoin model, we simulated a BitOverNet (of 1000 miners) and measured the number of peers having k links (degree distribution). Figure 1(a) shows the degree distribution of the bitcoin network (green bars) in addition to our predicted bitcoin model (red line) and a fitted power-law distribution (black line). It can be easily seen that BitOverNet does not show scale-free properties and that evolutionary-random graph nicely predicts the dynamics of the bitcoin network. We can now utilize evolutionary-random graph as a foundation for understanding the BitOverNet, predicting forks and considering its implications.

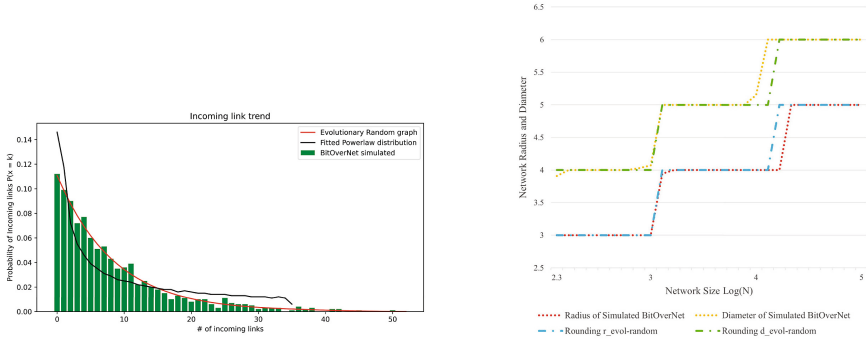
4 Key Graph Properties

Network diameter has been defined by [1] as “the maximal distance between any pair of its nodes.” Random graphs have, in average, the diameter [1, 8]:

$$d_{random} = \frac{\ln(N)}{\ln(pN)} = \frac{\ln(N)}{\ln(\langle k \rangle)} \tag{4}$$

where k is the number of links per node and $\langle k \rangle$ is the average number of links. Scale-free networks scale better than random graphs when it comes to diameter which can be represented as [1]:

$$d_{scale-free} = A \times \ln(N - B) + C \tag{5}$$



(a) BitOverNet’s incoming link distribution. The green bars represent the BitOverNet’s incoming link distribution, which are considerably different from scale-free’s powerlaw distribution (black line). On the contrary, it is nicely predicted by evolutionary random graph (red line). (b) Radius and diameter verification: simulated BitOverNet and evolutionary random graph. The x-axis represents the network size and the y-axis represents the radius and diameter

Fig. 1. Coverage verification

where A , B and C are fitting configuration parameters. Its diameter tends asymptotically to:

$$d_{scale-free} \propto \begin{cases} \frac{\ln(N)}{\ln(\ln(N))} & \lambda = 3, \text{ (Bollobas \& Riordan [6])} \\ \ln(N) & \lambda > 3, \text{ (Cohen and Havlin [9])} \end{cases}$$

Theorem 2. *Evolutionary Random Graph’s diameter and radius are:*

$$d_{er} = \frac{\ln(N - 2 \times m)}{\ln(\sum_{k=1}^{\infty} (k + m) \times P_k)} + 2 \tag{6}$$

$$r_{er} = \frac{\ln(N - m)}{\ln(\sum_{k=1}^{\infty} (k + m) \times P_k)} + 1 \tag{7}$$

Evolutionary Random Graph’s diameter can be calculated theoretically, using the same method used in [1, 8], but following evolutionary random graph (Eq. 3), as: $(\sum_{k=1}^{\infty} (k + m) \times P_k)^{(d_{er}-2)} = N - 2 \times m$ where the left side of the equation calculates the average number of nodes reached in (diameter - 2) steps. The -2 is used to reserve the first and last steps only for peripheral nodes which are only connected through the m outgoing links without incoming links. This concept is not available in graphs that do not distinguish between outgoing and

incoming neighbor requests such as random graph and scale-free. The right side of the equation is total number of nodes, while excluding the direct neighbors of the two periphery nodes reserved on the left side. Solving for d_{er} results in:
$$d_{er} = \frac{\ln(N-2 \times m)}{\ln(\sum_{k=1}^{\infty} (k+m) \times P_k)} + 2$$

In the same context, the graph's radius can be calculated, following the radius definition in [9], as: $(\sum_{k=1}^{\infty} (k+m) \times P_k)^{(r_{er}-1)} = N - m$

where the left side of the equation calculates the average number of nodes reached in (radius) steps, assuming that we are starting from a highly connected node (high centrality score). For calculating the radius, the differentiation between the incoming and outgoing neighbor requests is not needed, for the initiator, and thus the calculation is similar to [9]. The terminal node is peripheral and accounts for the 1 in $(r_{er} - 1)$. The right side of the equation is total number of nodes to be reached, while excluding the direct neighbors of the periphery node reserved on the left side. Solving for r_{er} results in:
$$r_{er} = \frac{\ln(N-m)}{\ln(\sum_{k=1}^{\infty} (k+m) \times P_k)} + 1$$

Corollary 2. Assuming constant processing time at each node and constant propagation time at each link, convergence, the time needed for a message to be broadcasted from one node to all the nodes in the network, is:

$$conver_r \leq conver \leq conver_d \tag{8}$$

where $conver_r = r_{er} \times Shd$, $conver_d = d_{er} \times Shd$ and Shd is the single hop delay i.e. the time needed for a message to be transmitted over a link, received and processed by the receiving node and have it ready for broadcasting (constant processing time at a node + constant propagation time at a link).

It is beneficial, in many cases, to calculate the *conver* time of a block, but it is also very insightful to study the propagation of the block and calculate the number of reached nodes as a function of time.

Lemma 2. Assuming constant processing time at each node and constant propagation time at each link, coverage, the number of nodes that have received the broadcasted block, is:

$$cover_r(t) \geq cover(t) \geq cover_d(t) \tag{9}$$

where $cover(t)$ is the coverage of the generated block, $cover_r(t)$ is the coverage lower bound representing block generated by a central node and $cover_d(t)$ is the coverage upper bound representing block generated by a periphery node.

Following the definition, we can deduce that at time 0 only the initial issuer has the block: $cover_r(0) = cover(0) = cover_d(0) = 1$

We can also deduce that at respective *conver* time, the block reached all the nodes: $cover_r(conver_r) = cover(conver) = cover_d(conver_d) = N$

Theorem 3. Evolutionary Random Graph's coverage bounds are:

$$cover_r(t) = \begin{cases} 1 & t < Shd \\ (\sum_{k=1}^{\infty} (k+m) \times P_k)^{(t')} & Shd \leq t < conver_r \\ N & t \geq conver_r \end{cases} \tag{10}$$

$$cover_d(t) = \begin{cases} 1 & t < Shd \\ m + (\sum_{k=1}^{\infty} (k + m) \times P_k)^{(t-1)} & Shd \leq t < conver_d \\ N & t \geq conver_d \end{cases} \quad (11)$$

where $t' = \frac{\min(t,conver_r)}{Shd}$ in Eq. 10 and $t' = \frac{\min(t,conver_d)}{Shd}$ in Eq. 11.

Following the assumptions in Lemma 2, block propagation events happen, theoretically, in multiples of Shd and thus it makes sense to convert the continuous time t into discrete hops. It is worth reminding that $conver$ is $radius \times Shd$ (taking lower bound as example), following Corollary 2, and thus $\frac{t}{Shd}$ is a coverage hop using the same unit as $radius$ which is hops. It is also worth noting that any time beyond $conver$ will not result in any coverage events and thus the discrete representation of time, we will use, is $\frac{\min(t,conver)}{Shd}$, where $conver$ depends on the initial broadcaster of the block, following Theorem 2.

For the lower bound, the first and third cases are trivial following Lemma 2. The second case is based on Eq. 7. The upper bound of t' is $(r_{er} - 1)$ and thus the second case's upper bound is $N - m$ following Theorem 2. The last hop is m , also following Theorem 2 which specifies that the last node is peripheral. This makes coverage reaches N precisely. For the upper bound, The first and third cases are trivial following Lemma 2. The second case is based on Eq. 6. The upper bound of t' is $(d_{er} - 1)$ and thus the upper bound of $t' - 1$ is $(d_{er} - 2)$. The second case's upper bound is $N - 2 \times m$ following Theorem 2. The last hop is m , also following Theorem 2 which specifies that the last node is peripheral. This makes coverage reaches N precisely. It is worth noting that second case's m is due to the first peripheral hop.

In Sects. 3 and 4, we deduced mathematically the graph that describes BitOverNet and we calculated some of its key properties namely, radius, diameter, convergence and coverage. To validate the adequacy of “Evolutionary Random graph” in predicting BitOverNet, Sect. 5 is spent on validating the predictions using real data collected from the BitOverNet.

5 Verification of Key Graph Properties

The correctness of the proposed distribution, Eq. 3, and its ability to describe BitOverNet’s link degree has been validated in Sect. 3, especially in Fig. 1(a). In this section, we validate the correctness of the key graph properties deduced in Sect. 4.

5.1 Verification: Diameter and Radius

In this section we verify the correctness of Theorem 2, namely d_{er} of Eq. 6 and r_{er} of Eq. 7. The correctness of d_{er} and r_{er} is verified against simulated BitOverNet. Verifying against the real bitcoin network is not possible for two reasons:

1. As discussed in Sect. 2, the existing probing techniques are only able to measure the reachable BitOverNet and thus unable to accurately measure the whole BitOverNet. Verifying against inaccurate measurements defy its purpose.
2. Even if accurate measurements were possible, we can only verify the instantaneous network size, but using simulated BitOverNet, the accuracy of the deduced properties has been verified over 4 orders of magnitude.

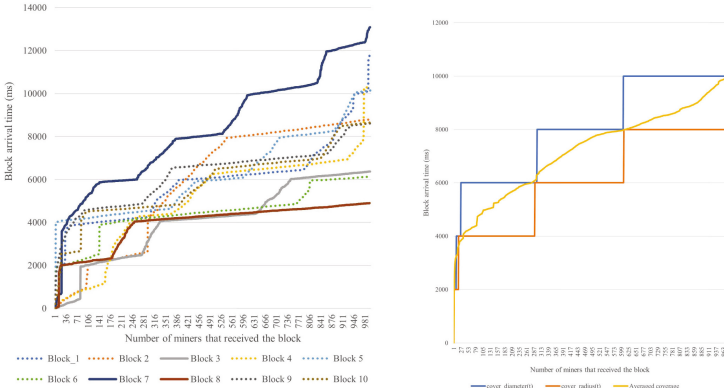
To measure the accuracy of the deduced radius and diameter, we simulated the BitOverNet with different sizes (values of N between 200 and 100000) and measured its network radius and diameter (averaged over 100 iterations). The measurements of the simulated BitOverNet and the deduced radius and diameter can be seen in Fig. 1(b) where the x-axis represents the network size and the y-axis represents the radius and diameter. As can be seen in Fig. 1(b), the deduced radius and diameter are able to predict the simulated BitOverNet for network sizes spanning over 4 orders of magnitude.

5.2 Verification: Coverage

As discussed in Sect. 5.1, the measurements generated from existing probing techniques are not complete, and thus cannot be used to verify the correctness of diameter and radius. However, the measurements reported by [5, 27] include timestamps of the first 1000 miners receiving a newly generated block. Those measurements are enough to verify the correctness of the coverage function (Theorem 3) for the initial 1000 nodes. We extracted from [5, 27] 1000 bitcoin blocks generated by the BitOverNet between Wednesday, December 22, 2021 5:57:12 AM (GMT) and Wednesday, December 22, 2021 4:12:01.884 PM (GMT). Out of the 1000 blocks we list, in Table 1, 10 blocks ranging from the block with fastest coverage, out of the 1000 observed blocks, to that with slowest coverage. Those 10 blocks span the coverage spectrum. In Table 1, the first column includes index, which is a number we give to each of the 10 short-listed blocks. The second column includes time, which is the generation time of the block. The third column includes hash, which is the PoW hash. It is worth noting that time and hash are enough to uniquely identify a bitcoin block.

Those 10 blocks are plotted in Fig. 2(a), where the x-axis represents the number of miners reporting the reception of the newly generated block and the y-axis represents the arrival time. This figure illustrates the coverage, for the first 1000 miners, for each of the ten plotted blocks. Block 3 has the fastest start, which resembles a block generated by a miner with high centrality (central). This coverage is the closest to $cover_r$. Block 7 resembles a block generated by a miner with low centrality (peripheral). This coverage is the closest to $cover_d$. It is worth noting that neither block 3 nor block 7 are the exact bounds defined in Theorem 2. The solid lines represent the empirical coverage bounds (blocks 3, 7 and 8) and the dotted lines represent the coverage spanning between the empirical coverage bounds. We can also observe that the coverage function is behaving like a step function, where the steps are 2000 ms. This step is in fact the

Shd and thus we can conclude that the Shd is 2000 ms. To verify the correctness of coverage function (Theorem 3) for the initial 1000 nodes, we averaged the coverage of each of the 10 blocks listed in Table 1 and compared it to the coverage function (Theorem 3) as shown in Fig. 2(b). As can be seen in Fig. 2(b), the coverage function ($cover_d(t)$ and $cover_r(t)$) envelopes the averaged coverage of real bitcoin blocks. This verifies the correctness of the Theorem 3.



(a) Coverage of blocks 3, 7 and 8. Blocks 3 and 8 resemble blocks generated by a central node. Block 7 resembles a block generated by a peripheral node. (b) Coverage function enveloping the average coverage of real bitcoin blocks, showing its ability to predict block propagation and coverage

Fig. 2. Coverage verification

Table 1. List of 10 sample blocks that span the coverage spectrum

Block Information		
Index	Time (12/22/2021)	Hash
1	6:13:50.016	0048aff673f37d5c5020246b630a00cfc4b2004c29afe097a0c0fa4043abc1a
2	6:17:45.978	3ea038dfc7083e96030d18e757f17022d985ffbeaa93eedaa18c3656d0996b74
3	6:15:15.931	15bf68a078a25b7b2f0f53830d8fb98478f5ba54cc533e22412eb3ef55dd79d
4	6:17:53.707	da0128876c568a1ccea5f54f7e99e7a74b062e8e683566d9f56aff20544de6807
5	6:14:25.960	5f8aa6c0bb36c5746e5779aff54e9109ab771799c6b7542f84027c3651d967a3
6	6:12:51.964	846a4be7a74d5e2bde2fdde423def3f38b785425904918d12c09c43aee65a5c1
7	6:12:12.008	6e64e3aa92e871c6ae3ce59390c1e6e3f5710c5c1f9511235833c96462ae7561
8	6:15:57.937	a39b5ce4b641f437453bd0f24c64f184c3b3eba62896eb75b55cfd9f4e9e0b08
9	6:13:15.377	fb78c02bab5091e442279616fe5060bd69315b05838e0adb036e5b39faa5b224
10	6:13:41.417	403f8a3d23d56c446e032f18627fe9c58c9fa17b60432d84ab444186f2cf8f79

6 Conclusion

Peer-to-peer overlay networks including BitOverNet, blockchain, Tor anonymization network (onion router) and Invisible Internet Project (I2P Garlic Routing) are free from spatial-temporal restrictions and thus unexplainable using existing theoretical lenses. The lack of theoretical model to understand those peer-to-peer overlay networks has significant implications such as:

- Inability to predict the dynamics and thus manage the characteristics of Peer-to-peer overlay networks.
- Inability to model network's tipping points, disintegration and other network failures, which can be the result of malicious activities.
- Inability to determine whether the network is efficiently operating.

In this work, we aimed at developing a theoretical model for explaining the behavior of BitOverNet. To the best of our knowledge, this work is the first in this direction. The contributions of this work are multi-folded. First, we showed that BitOverNet does not follow scale-free networks or other network models (graphs) found in literature. Second, we proposed evolutionary random graph and showed, using real data, its ability to describe and predict the current properties of the BitOverNet. Third, we developed key graph properties, which are important for proactively managing the BitOverNet. As a future work, we plan on building on top of this theoretical layer to investigate key bitcoin properties such as resilience and valuation. We are also planning on investigating other peer-to-peer overlay networks by generalizing evolutionary random graph to overcome its specificity to BitOverNet.

References

1. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**(1), 47–97 (2002). <https://doi.org/10.1103/RevModPhys.74.47>. Publisher: American Physical Society
2. Albert, R., Jeong, H., Barabási, A.L.: Diameter of the world-wide web. *Nature* **401**(6749), 130–131 (1999). <https://doi.org/10.1038/43601>. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 6749 Primary_atype: Research Publisher: Nature Publishing Group
3. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999). <https://doi.org/10.1126/science.286.5439.509>
4. Ben Mariem, S., Casas, P., Donnet, B.: Vivisecting blockchain P2P networks: unveiling the bitcoin IP network. In: ACM CoNEXT Student Workshop (2018)
5. Blockchain.com: Blockchain charts: The most trusted source for data on the bitcoin blockchain (2022). <https://www.blockchain.com/charts>
6. Bollobas, B., Riordan, O.: The diameter of a scale-free random graph. *Combinatorica* **24**(1), 5–34 (2004). <https://doi.org/10.1007/s00493-004-0002-2>
7. Bou Abdo, J., El Sibai, R., Demerjian, J.: Permissionless proof-of-reputation-X: a hybrid reputation-based consensus algorithm for permissionless blockchains. *Trans. Emerg. Telecommun. Technol.* **32**(1), e4148 (2021)

8. Chung, F., Lu, L.: The diameter of sparse random graphs. *Adv. Appl. Math.* **26**(4), 257–279 (2001)
9. Cohen, R., Havlin, S.: Scale-free networks are ultrasmall. *Phys. Rev. Lett.* **90**(5), 058,701 (2003). <https://doi.org/10.1103/PhysRevLett.90.058701>. Publisher: American Physical Society
10. Delgado-Segura, S., et al.: TxProbe: discovering bitcoin's network topology using orphan transactions. In: Goldberg, I., Moore, T. (eds.) *FC 2019*. LNCS, vol. 11598, pp. 550–566. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32101-7_32
11. Deshpande, V., Badis, H., George, L.: BTCmap: mapping bitcoin peer-to-peer network topology. In: 2018 IFIP/IEEE International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), pp. 1–6. IEEE (2018)
12. Bitcoin Developer: P2P network (2022). https://developer.bitcoin.org/devguide/p2p_network.html
13. Donet Donet, J.A., Pérez-Solà, C., Herrera-Joancomartí, J.: The bitcoin P2P network. In: Böhme, R., Brenner, M., Moore, T., Smith, M. (eds.) *FC 2014*. LNCS, vol. 8438, pp. 87–102. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44774-1_7
14. Eisenbarth, J.P., Cholez, T., Perrin, O.: An open measurement dataset on the bitcoin p2p network. In: 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 643–647 (2021). ISSN 1573-0077
15. Erdős, P.: On random graphs I. *Publ. Math. Debrecen*, pp. 290–297 (1959)
16. Essaid, M., Park, S., Ju, H.: Visualising bitcoin's dynamic P2P network topology and performance. In: 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp. 141–145 (2019). <https://doi.org/10.1109/BLOC.2019.8751305>
17. Jiang, S., Wu, J.: Approaching an optimal bitcoin mining overlay. *IEEE/ACM Trans. Networking* **31**, 2013–2026 (2023)
18. Li, J., et al.: MANDALA: a scalable blockchain model with mesh-and-spoke network and H-PBFT consensus algorithm. *Peer-to-Peer Netw. Appl.* **16**, 226–244 (2022)
19. Lischke, M., Fabian, B.: Analyzing the bitcoin network: the first four years. *Future Internet* **8**(1), 7 (2016). <https://doi.org/10.3390/fi8010007>. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute
20. Miller, A.K., et al.: Discovering bitcoin's public topology and influential nodes (2015)
21. Neudecker, T., Hartenstein, H.: Network layer aspects of permissionless blockchains. *IEEE Commun. Surv. Tutor.* **21**(1), 838–857 (2019). <https://doi.org/10.1109/COMST.2018.2852480>. Conference Name: IEEE Communications Surveys Tutorials
22. Paphitis, A., Kourtellis, N., Sirivianos, M.: Graph analysis of blockchain P2P overlays and their security implications. In: Arief, B., Monreale, A., Sirivianos, M., Li, S. (eds.) *SocialSec 2023*. LNCS, vol. 14097, pp. 167–186. Springer, Singapore (2023). https://doi.org/10.1007/978-981-99-5177-2_10
23. Park, S., Im, S., Seol, Y., Paek, J.: Nodes in the bitcoin network: comparative measurement study and survey. *IEEE Access* **7**, 57009–57022 (2019)
24. Serena, L., Ferretti, S., D'Angelo, G.: Cryptocurrencies activity as a complex network: analysis of transactions graphs. *Peer-to-Peer Netw. Appl.* **15**(2), 839–853 (2022)

25. Sgantzos, K., Grigg, I., Al Hemaury, M.: Multiple neighborhood cellular automata as a mechanism for creating an AGI on a blockchain. *J. Risk Financ. Manag.* **15**(8), 360 (2022)
26. Tao, B., Ho, I.W.H., Dai, H.N.: Complex network analysis of the bitcoin blockchain network. In: 2021 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5. IEEE (2021)
27. Yeow, A.: Bitnodes (2022). <https://bitnodes.io/>
28. Zeadally, S., Abdo, J.B.: Blockchain: trends and future opportunities. *Internet Technol. Lett.* **2**(6), e130 (2019)



Graph Based Approach for Galaxy Filament Extraction

Louis Hauseux^(✉), Konstantin Avrachenkov, and Josiane Zerubia

Inria, Université Côte d'Azur, Sophia-Antipolis, France
{louis.hauseux,konstantin.avrachenkov,josiane.zerubia}@inria.fr

Abstract. We propose an original density estimator built from a cloud of points $\mathcal{X} \subset \mathbb{R}^d$. To do this, we consider geometric graphs $\mathcal{G}(\mathcal{X}, r)$ on the cloud. These graphs depend on a radius r . By varying the radius, we see the emergence of large components around certain critical radii, which is the phenomenon of *continuum percolation*. Percolation allows us to have both a local view of the data (through local constraints on the radius r) and a global one (the emergence of macro-structures). With this tool, we address the problem of galaxy filament extraction. The density estimator gives us a relevant graph on galaxies. With an algorithm sharing the ideas of the Fréchet mean, we extract a subgraph from this graph, the galaxy filaments.

Keywords: geometric graphs · continuum percolation · Fréchet mean · galaxy filaments

1 Introduction

At scales of billions light-years, the observable universe—matter and light—does not follow a uniform distribution but forms what are known as ‘large-scale structures’ [3, 13]. These structures seem arranged hierarchically: 1° super-clusters of galaxies (hyper-dense small volumes, sometimes called ‘knots’ or ‘nodes’); 2° ‘sheets’ or ‘walls’ of galaxies; 3° ‘filaments’ of galaxies. These different clusters delimit large “voids” regions that are virtually empty of galaxies: they shape the “cosmic web”, like a giant sponge or a spider’s web.

Astronomical surveys [1, 2] now contain millions of galaxies, making it impossible to extract these structures with the naked eye. Various types of algorithms have been proposed to extract automatically these clusters, and particularly the galaxy filaments. (*Cf.* the survey “Tracing the cosmic web” [22]). Most are based on density estimators¹ (two comparative studies: [12, 14]). The density-based

The first author would like to thank the Université Côte d’Azur (UCA) DS4H Investments in the Future project managed by the National Research Agency (ANR, reference number ANR-17-EURE-0004) and 3IA Côte d’Azur for partial funding of his PhD thesis. All the authors acknowledge a partial support by Nokia Bell Labs “Distributed Learning and Control for Network Analysis” and Bpifrance in collaboration with Airbus D&S (LiChIE contract, 2020–2024).

¹ Those ‘filaments’ are relatively thick and have a non-negligible width; a real 1D-manifold would not have a density w.r.t. the Lebesgue measure.

methods are often based on the Delaunay density estimator DTFE [28], estimator derived from Delaunay triangulation; the estimated density being inversely proportional to the area of the neighbouring triangles (the analogous exists with Voronoï tiling). We will look at another classical density estimator used: The K -Nearest Neighbours (K -NN) algorithm [6]. This very simple algorithm can produce—with some refinements—impressive results. For example, the HDB-SCAN hierarchical clustering algorithm [10] is based on the High-Density Levels of the K -Nearest Neighbours density estimator.

Obtaining a filamentary structure naturally led to introduce graphs on the galaxies (considered as points in space). As early as 1985, the pruned minimal spanning tree [5] was proposed as a filamentary model.

More interesting is the idea proposed by Colberg [11] who also pruned minimal spanning tree and studied what happens at the percolation stages. The percolation thresholds are directly linked to the types of structure that appear. This is the key point to observe.

The Delaunay and K -Nearest Neighbours estimators have only a local view of the data. K -NN estimator has good properties (consistency, calculation speed, see monograph by Biau & Devroye [6]). However, obtaining consistency requires that k tends to infinity. In practice, k is taken smaller than 10 and the number of galaxies is insufficient too much hope in this estimator.

Percolation is a phenomenon which, under local constraints, can be observed macroscopically: It is the precise moment when macro-structures appear. Percolation allows us to have both a local view of the data (through local constraints on the graph) and a global one (through the emergence of macro-structures).

If we assume galaxies are IID points plotted in space by an unknown measure of density f , thanks to the hierarchical structures, we could identify galaxy clusters with the highest density clusters [18, 25], *i.e.* $f^{-1}([h; +\infty)) = \{x \in \mathbb{R}^d \mid f(x) \geq h\}$.

In this article we propose a new estimator for *density levels* and filament extraction using geometric graphs [25]. If \mathcal{X} denotes the cloud points and $\mathcal{G}(\mathcal{X}, r)$ the geometric graph of radius r built on these points, we vary r from 0 until the percolation phases. At each radius r , we associate a cluster $\Sigma_r \subset \mathbb{R}^d$, the *density level* for radius r . Σ_r increases with r (like $\mathcal{G}(\mathcal{X}, r)$).

An intuitive idea for extracting filaments from the cluster Σ_r is to take its medial axis [4, 8]. However, we would not take advantage of the persistent information (r can vary), nor the fact that we have a graph (with an induced distance). This is why we prefer to proceed as follows: increasing the radius r until big components in $\mathcal{G}(\mathcal{X}, r)$ appear. At this moment, we initialise a new filament within the big component by its Fréchet mean [15]. As the component grows with r , we add points to the filament so that the augmented filament satisfies a minimum condition (similar to Fréchet's mean minimization).

We show an example of such filament extraction on a synthetic 2D-image of galaxies. At a glance, we compare our results with a stochastic method [29].

For a quantified comparison, we compare our density level estimator with conventional density estimators for this type of problem (Delaunay estimator,

K -Nearest Neighbours) on point cloud generated by a known density function f . Our estimator is already showing very good results, especially for high-density clusters.

2 Preliminaries

In this section, we introduce the mathematical background.

Geometric Graphs. Given a set \mathcal{X} of points in $R \subset \mathbb{R}^d$ and a radius r , the geometric graph $\mathcal{G}(\mathcal{X}, r)$ is the undirected graph whose nodes are the points in \mathcal{X} , and whose edges join all the nodes that are at a distance less than r .

Percolation Phenomenon [9, 23, 25]. Let \mathcal{H}_λ be a Poisson point process on \mathbb{R}^d of intensity λ and $\mathcal{H}_{\lambda,0} := \mathcal{H}_\lambda \cup \{0\}$. Denote $p_1(\lambda), p_2(\lambda), \dots$ the probabilities that the component containing origin in $\mathcal{G}(\mathcal{H}_{\lambda,0}, 1)$ has exactly 1, 2, \dots nodes. And $p_\infty(\lambda)$ the *percolation probability* (this component is of infinite size):

$$p_\infty(\lambda) := 1 - \sum_{k=1}^{\infty} p_k(\lambda).$$

$p_\infty(\cdot)$ is an increasing function of λ ; there exists a *critical value* λ_c (which depends on the dimension d of the space) below which $p_\infty(\lambda) = 0$ (for $\lambda < \lambda_c$) and above which $p_\infty(\lambda) > 0$ (for $\lambda > \lambda_c$). In the latter case, $p_\infty(\lambda)$ can be seen as the proportion of points that fall into the giant component (the second component being of negligible size compared to the first).

This phenomenon of percolation, *i.e.* the appearance of a giant connected component, is very interesting for modelling and studying numerous problems. For example, the spread of a forest fire (the nodes being the trees, the neighbourhood radius r the threshold below which a tree devoured by flames sets fire to its neighbours). We can then deduce from the density of the forest whether an outbreak of fire is likely to be naturally confined to a limited area or not.

2.1 Density Estimator and Density Levels

Various indicators exist for comparing probability measures, such as the Kullback-Leibler divergence [14, 21] or Wasserstein distance [24, 32]. But what do these tools mean when the provided density estimator is not integrable, like the K -Nearest Neighbours estimator? There is a much stronger objection: The Kullback-Leibler divergence and the Wasserstein distance do not take into account the specific features of our problem: galaxy clusters to be identified are highly hierarchical. We are asking for good *relative* accuracy (preserving density hierarchy), not necessary *absolute*.

To have a tool that conforms to the hierarchical structure of clusters, we are going to define a notion of density level inspired by the “High-Density Clusters” introduced by Hartigan [18], *cf.* also the introduction of Penrose’s book [25].

The High-Density Clusters of level h are the different connected components of $f^{-1}([h; +\infty))$. Hartigan [19] showed that the connected components of geometric graphs is a consistent estimator of these clusters in dimension 1.

Let $P \in [0; 1]$ be a parameter representing the proportion of classified points (those of highest density). To this proportion P , we can associate the density-height h_P defined as follows:

$$h_P = \inf \left\{ h \mid \int_{f \geq h} f(x) dx \leq P \right\}.$$

Now, given a point $x \in \mathbb{R}^d$, we attribute to x the first P such that x lies into one of the clusters of level h_P :

$$\mathcal{P} : x \in \mathbb{R}^d \mapsto \mathcal{P}(x) := \inf \{ P \in [0; 1] \mid x \in f^{-1}([h_P; +\infty)) \}.$$

Intuitively, $\mathcal{P}(x)$ represents the proportion of points that must be taken in the cloud points for x to appear in one of the High-Density Clusters.

For convenience, we will consider the function $1 - \mathcal{P}$ instead, which is thus an increasing function of the density. It is this $1 - \mathcal{P}$ function that we call the *map of density levels*.

This hierarchical classification is perfectly suitable if we have a good estimate of the proportion of galaxies which lie in each kind of clusters (superclusters, walls, filaments [20]). If this knowledge is lacking, the proportion $P(r)$ of classified points as a function of r can still be used to highlight percolation phases.

Comparison for the Identification of a Specific Cluster. We may wish to compare two estimators for the correct identification of a particular cluster. We can then use the following protocol, inspired by the Precision/Recall method: Let \mathcal{C}_P be a cluster of level P (a connected component of level h_P for the true density function f) with volume $|\mathcal{C}_P|$. Let $\hat{\mathcal{C}}_{P'}$ be the corresponding empirical cluster and $|\hat{\mathcal{C}}_{P'}|$ its volume. We can then define *Precision* and *Recall* :

$$\text{Precision}(P, P') = \frac{|\mathcal{C}_P \cap \hat{\mathcal{C}}_{P'}|}{|\hat{\mathcal{C}}_{P'}|}, \quad \text{Recall}(P, P') = \frac{|\mathcal{C}_P \cap \hat{\mathcal{C}}_{P'}|}{|\mathcal{C}_P|}.$$

Comparison of Density Level Maps. Now suppose that we have a complete density level map of a region $\mathcal{R} \subset \mathbb{R}^d$ observed:

$$1 - \hat{\mathcal{P}} : x \in \mathbb{R}^d \mapsto 1 - \hat{\mathcal{P}}(x) \in [0; 1]$$

which is an estimator of the ground truth function $1 - \mathcal{P}$. We can then take the p norms (from the L^p space) to compare our estimators with the original function.

3 Our Method

Let us describe more accurately our method in this section. In three main steps:

- Starting with a radius r equal to 0 and increasing it. For each radius r , we construct $\mathcal{G}(\mathcal{X}, r)$. From this graph, we retain only the connected components with more nodes than a certain *percolation threshold*. We then look at the proportion P of points lying into one of these major connected components.
- The associated estimator of the High-Density Clusters of level h_P is a set $\Sigma_P \subset \mathbb{R}^d$ containing the points of the major connected components.
- Each time a large component appears, a new filament is created. (Filaments are modelled by sub-graphs of connected components). Filaments are initialized with the Fréchet mean of the component (for the distance induced on the graph). Then, they grow progressively with the High-Density cluster.

Persistent Ingredients. By analogy with persistent homology (see the survey by Bobrowski & Kahle [7]), we call ‘persistent’ the variational method of observing what happens when the radius r varies.

3.1 Theoretical Advantage: The Percolation Rate

Percolation is a ‘fast’ phenomenon. Let \mathcal{H}_1 be a Poisson point process on \mathbb{R}^2 with fixed intensity $\lambda = 1$: We only vary the radius r of the geometric graph $\mathcal{G}(\mathcal{H}_1, r)$.

Starting with $r = 1$, percolation has not yet taken place. The largest component therefore contains a proportion $p_\infty(1) = 0$ of the points. For $r := r_c = \sqrt{\lambda_c} \approx 1.2$ [27, 34], percolation occurs: A giant component appears².

As soon as the giant component appears (for $r \geq r_c$), if we increase radius r slightly, the probability of percolation $p_\infty(r)$ approaches 1 very quickly. At this point, the giant component includes almost all the points (a proportion $p_\infty(r)$).

The plot below³ shows a simulation of $p_\infty(r)$ on \mathbb{R}^2 .

How can we measure the speed of percolation? From a certain radius, r_{\min} , the giant component becomes non-negligible in size compared with the cloud of points. Suppose it contains $\varepsilon \leftarrow 5\%$ of the points. That is:

$$r_{\min} := p_\infty^{-1}(\varepsilon).$$

For another larger radius $r_{\max} \geq r_{\min}$, the giant component encompass almost all the points (a proportion $1 - \varepsilon$). We can consider the percolation to be complete:

$$r_{\max} := p_\infty^{-1}(1 - \varepsilon).$$

The quantity of interest is

$$\frac{r_{\max}}{r_{\min}}.$$

² But still $p_\infty(r_c) = 0$. Although it is widely accepted that for any $d \geq 2$, $p_\infty(r_c) = 0$, this has only been proved for $d = 2$ (cf. theorem 4.5 by Meester & Roy [23] and by Tanemura for d sufficiently large [30]).

³ Thanks to Vinay Kumar [33] for sharing the data.

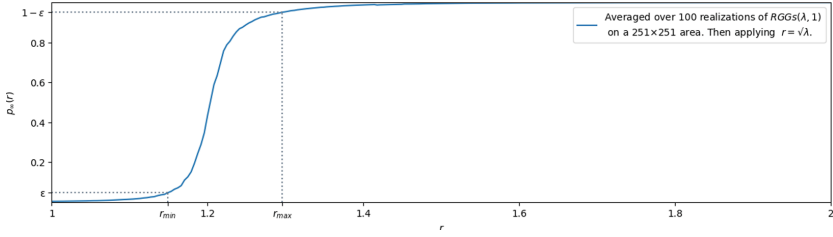


Fig. 1. Estimation of percolation probability $p_\infty(r)$ in \mathbb{R}^2 by simulation on giant Random Geometric Graphs. © [33]. With $\varepsilon = 0.05$, it gives the following results: $r_{\min} = 1.15$ and $r_{\max} = 1.30$. Note that on this curve, $p_\infty(r)$ is positive even if $r \lesssim r_c \approx 1.2$; this is due to the approximation of \mathbb{R}^2 by a finite square 251×251 .

Suppose there are two large contiguous regions, of intensity λ_1 and λ_2 with $\lambda_2 < \lambda_1$. From a certain radius $r_{\min}^{(1)}$, percolation begins in the first region with the highest λ_1 intensity. To identify this region correctly without confusing it with the neighbouring region of lower intensity λ_2 , the first percolation phase must be ‘completed’ before percolation begins in the second region. In other words, we want to have :

$$r_{\max}^{(1)} < r_{\min}^{(2)}.$$

Now, in \mathbb{R}^d , $r_{\min}^{(2)} = \left(\frac{\lambda_2}{\lambda_1}\right)^{\frac{1}{d}} \times r_{\min}^{(1)}$ and $r_{\max}^{(2)} = \left(\frac{\lambda_2}{\lambda_1}\right)^{\frac{1}{d}} \times r_{\max}^{(1)}$, the two regions can therefore be correctly and distinctly identified if and only if :

$$\frac{r_{\max}^{(1)}}{r_{\min}^{(1)}} < \left(\frac{\lambda_2}{\lambda_1}\right)^{\frac{1}{d}}.$$

(The quantity $\frac{r_{\max}}{r_{\min}}$ being independent of the intensity λ of the region).

Percolation will be all the faster as the ratio $\frac{r_{\max}}{r_{\min}}$ becomes close to 1. On Fig. 1, for $\varepsilon = 0.05$, we can see that this ratio is indeed close to one: $r_{\min} = 1.15$ and $r_{\max} = 1.30$. Thus: $\frac{r_{\max}}{r_{\min}} \approx \frac{1.30}{1.15} \approx 1.13$ in \mathbb{R}^2 .

The ‘Percolation’-Graph. Introducing percolation ingredients into the geometric graph is made in a very simple way: We set a robust percolation threshold (e.g. $PercolThreshold \leftarrow 50$), and consider only connected components with more than $PercolThreshold$ nodes. We denote $\mathcal{G}_1(\mathcal{X}, r)$ the graph pruned of the small connected components.

3.2 Filament Extraction from a Graph

In this sub-section, we fix the radius r of the pruned geometric graph $\mathcal{G}_1(\mathcal{X}, r)$ and look at one of the big connected components, which is a sub-graph $G(V, E)$ with

vertices V and edges E . A filament *Filament* may already have been drawn on this component (for previous radii). We have a distance d on this graph induced by Euclidean distance between two neighbouring points. If $x, y \in V$ are two vertices, $ShortPath(x, y)$ denotes the set of vertices of the shortest path (for the distance d) from x to y in G .

The variable *Centres* denotes the set of vertices which are chosen to represent *Filament*. The first centre is the Fréchet mean of G . *FilNodes* are the vertices of *Filament* which join the *Centres* such that *Filament* is the Minimal Tree spanning *Centres* nodes.

Algorithm 1. Filament extraction of a connected component $G(V, E)$

```

Centres                                ▷ The centres of the pre-existing filament
FilNodes                                ▷ Nodes of Filament
PercolThreshold ← 50                    ▷ The percolation threshold
while |Centres| < int(| $V$ |/PercolThreshold) do    ▷ We search for a new centre
     $D \leftarrow \{\}$                                 ▷ The sums of the distances to minimise
    for  $x \in V$  do                                ▷  $x$  is the hypothetical new centre
        NodesFilament ← copy(FilNodes)
        Branchx ← ShortPath( $x, NodesFilament$ )    ▷ Hypothetical new branch
        NodesFilament ← NodesFilament ∪ Branchx
         $D[x] \leftarrow 0$ 
        for  $y \in V$  do
             $D[x] \leftarrow D[x] + d(y, NodesFilament)^2$ 
        end for
    end for
     $x \leftarrow \text{argmin}(D)$                                 ▷ The new centre chosen
    Centres ← Centres ∪ { $x$ }
    FilNodes ← FilNodes ∪ Branchx
end while
Filament ← MinimalSpanningTree(FilNodes)
Returns Filament

```

Note that the first centre chosen with the Algorithm 1 is the Fréchet mean [15] of the graph $G(V, E)$. Moreover, the *Filament* result is a tree.

In some cases, we might want to ‘close’ the tree by inserting a loop and introduce a closing post-processing algorithm.

A final post-processing consists of pruning the filamentary network obtained of branches that are too small (e.g. those shorter than the radius r of the geometric graph).

3.3 The Density Level Estimator

Thanks to all the concepts and tools defined above, we are now able to provide an estimator of density levels. Let $P \in [0; 1]$ be the proportion of ‘classified’

points (lying in one of the great components). The associated radius is

$$r_P = \inf \left\{ r \mid \frac{| \text{Vertices}(\mathcal{G}_1(\mathcal{X}, r)) |}{| \text{Vertices}(\mathcal{G}(\mathcal{X}, r)) |} \geq P \right\}.$$

At radius r_P , the connected components of $\mathcal{G}(\mathcal{X}, r_P)$ of size $\geq \text{PercolThreshold}$ represent a proportion P of the cloud point \mathcal{X} . We have now to find a volume of the space $\Sigma_P \subset \mathbb{R}^d$ that fits best the classified points. An intuitive solution, inspired by percolation theory (*Boolean model*), would be to take the union of balls centred on these classified points:

$$\Sigma_P := \bigcup_{x \in \text{Vertices}(\mathcal{G}_1(\mathcal{X}, r_P))} B(x, R).$$

The radius R needs to be chosen. Usually, in continuum percolation theory, we consider $R = r_P/2$. But this radius is too small for our purpose: If we consider the volume on the entire Boolean model at percolation stage:

$$\Sigma := \bigcup_{x \in \text{Vertices}(\mathcal{G}(\mathcal{X}, r_c))} B(x, r_c/2)$$

(also $\Sigma \supset \Sigma_P$), in \mathbb{R}^2 , Σ occupies only a proportion $\phi_c \approx 0,676$ [27,34] of the space. (ϕ_c is called the *space coverage* [17]). That is, Σ will not recover a proportion $1 - \phi_c$ of the space, equals by ergodicity to $e^{-\lambda_c \theta_d / 2^d}$ (θ_d being the volume of the unit ball in \mathbb{R}^d). With a radius twice as large (our choice: $R = r_P$; see on Fig. 2 an example), this un-recovered proportion of space is reduced to: $e^{-\lambda_c \theta_2} = (1 - \phi_c)^4 \approx 0.01$. Our experiments show that taking a larger radius R (up to $1.5 \times r_P$) produces better results. Increasing R increases the *Recall*. Taking R too large, however, can end up lowering *Precision*.

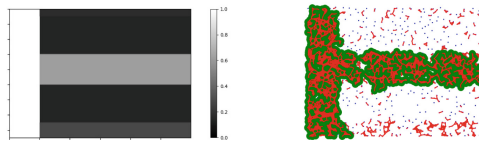


Fig. 2. Left: Density levels of f . Function support is a rectangle 24×17 subdivided into sub-rectangles. A left high density ‘blob’ ($f \propto 4$); At the center, a thick ‘Filament’ ($f \propto 3$); Below a thinner one ($f \propto 2$) and upper a very thin one ($f \propto 1$). Between ‘Filaments’, some ‘voids’ ($f \propto \frac{1}{2}$). Right: 2000 IID points generated by f . The $\mathcal{G}(\mathcal{X}, r \leftarrow 0.4)$ graph edges and the density level volume associated Σ_r , with $\text{PercolThreshold} \leftarrow 50$.

The Density Level Map. The definition of the empirical density level map $1 - \hat{\mathcal{P}}$ follows naturally: let $x \in \mathbb{R}^d$, $\hat{\mathcal{P}}(x)$ is the first P for which x lies in Σ_P , *i.e.*

$$\hat{\mathcal{P}}(x) := \inf \{ P \in [0; 1] \mid x \in \Sigma_P \} \quad (\text{with the convention } \inf(\emptyset) = 0).$$

On Fig. 4, the reader can see three examples of empirical density level maps estimated on a cloud of points IID generated with density f (see Fig. 2).

4 Results

In this section, we first see an example of filament extractions on a synthetic 2D-image of galaxies⁴ and compare visually with a stochastic method [29]. Second, for a more quantified comparison, we compare density level estimators (ours, Delaunay estimator, 10-Nearest Neighbours) on point cloud \mathcal{X} generated by a known density function f plotted on Fig. 2: A rectangular-shaped density map (‘blob’ modelled by a large and high-density rectangle, ‘filament’ by a thin one).

Visual Comparison with Stochastic Method. Stochastic geometry methods have been proposed for extracting galaxy filaments [16, 29, 31]. A sheet of *Filament* is represented by a rectangular box. Geometric priors are then introduced on its shape, its density, its connectivity (or alignment) with the other boxes, ... In the end, using techniques such as simulated annealing, the configuration that best fits the data is obtained. Figure 3 shows the result of such an algorithm.

We apply Algorithm 1 (see below) to \mathcal{X} with r varying from 0 to 5.2. As r grows, components increase in size, merge, and *Filaments* grow with the radius. In Fig. 3, the result (= *Filaments* drawn) for a very small stopping radius and a larger one.

Note that, thanks to persistent ingredients, *Filaments* are robust to the choice of stopping radius: once the majority of points appear in a large component, few new centres are added. So the result is the same except for a few ‘connection-bridges’. As there is no ground truth, comparison is difficult. Our filaments, which are drawn without geometric constraints, are more irregular. However, they form a genuine network and are less prone to over-detection. Our method is also much less computationally intensive. What is more, it can easily be applied to three-dimensional images.

Comparison with Classical Density Estimators. In order to obtain quantified results, let us now work on clouds of points IID generated according to a density function f that we know (*cf.* Fig. 2 with a 2000-points cloud \mathcal{X} scattered).

The results of the three density level estimators (ours, Delaunay [28] and K -Nearest Neighbours [6] with $K \leftarrow 10$) can be seen in Fig. 4. Visually, ours is more homogenous and less prone to local overestimates in low-density zone.

We are now numerically able to compare the estimated density level maps (Fig. 4) with the original one (Fig. 2; density plateaus between two density levels have been replaced by the half-sum). Table 1 shows an advantage for our estimator.

⁴ Thanks to the authors of the article “Detection of cosmic filaments using the Candy model” [29] for the generation of the data used herein. These data were kindly supplied by Radu Stoica, Enn Saar and Vicent Martínez.

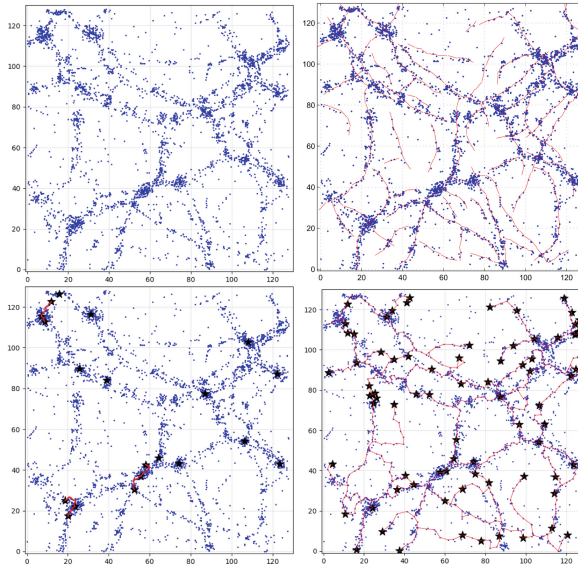


Fig. 3. Top Left: Mock cloud point $\mathcal{X} \subset \mathbb{R}^2$ generated by Stoica *et al.* [29]. Top Right: Results of the stochastic ‘Candy Model’ algorithm [29]. Bottom: The persistent extracted Filaments on \mathcal{X} with $PercolThreshold \leftarrow 50$. Left, for r varying from 0 to 1. First Centres (blacks stars \star) and Filaments appear. Right, $0 \leq r \leq 5.2$.

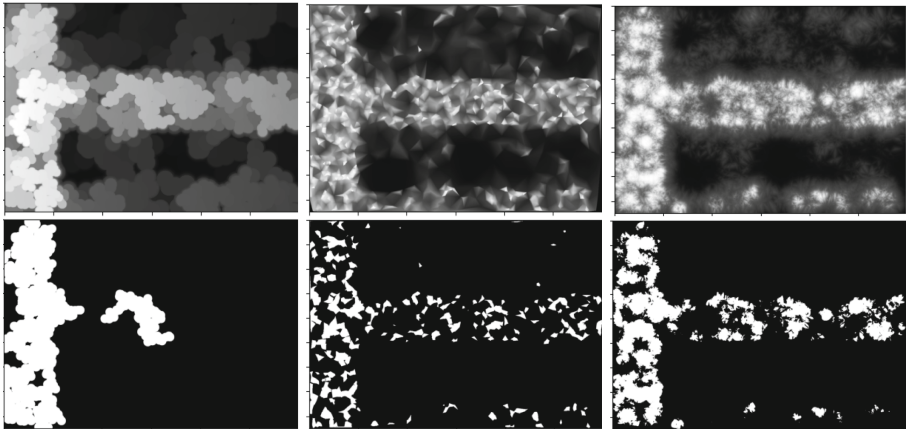


Fig. 4. Top: Three density level maps estimated on the cloud point of Fig. 2. From left to right: Our Percol-Graph estimator, the Delaunay estimator [28], the K -Nearest Neighbours estimator [6] with $K \leftarrow 10$. Bottom: The cluster (white) associated to the density level $1 - \hat{\mathcal{P}} > 0.617$. Theoretically, this cluster is the rectangle on the left.

Table 1. Distance between the estimated density level map and the original one

Algorithm	$L^1 := \sum_x \frac{1}{N} \hat{\mathcal{P}}(x) - \mathcal{P}(x) $	$L^2 := \sqrt{\sum_x \frac{1}{N} (\hat{\mathcal{P}}(x) - \mathcal{P}(x))^2}$
Graph-Percol	0.095	0.142
Delaunay	0.123	0.187
K -Nearest Neighbours	0.114	0.166

Let us take a closer look. Four levels are interesting, corresponding to the successive appearance of the ‘filaments’: $1^\circ 1 - \mathcal{P} = 0.617$ (highest-density left rectangle). $2^\circ 1 - \mathcal{P} = 0.280$ (middle thick filament). $3^\circ 1 - \mathcal{P} = 0.169$ (bottom filament). $4^\circ 1 - \mathcal{P} = 0.140$; (Only “voids” are not in this level). We compute *Precision* and *Recall* for these levels. Results are listed in Table 2.

Table 2. *Precision* and *Recall* on density levels of filament apparitions.

Algorithme	$1 - \hat{\mathcal{P}} > 0.617$		$1 - \hat{\mathcal{P}} > 0.280$		$1 - \hat{\mathcal{P}} > 0.169$		$1 - \hat{\mathcal{P}} > 0.140$	
	<i>Precision</i>	<i>Recall</i>	<i>Precision</i>	<i>Recall</i>	<i>Precision</i>	<i>Recall</i>	<i>Precision</i>	<i>Recall</i>
Graph-Percol	0.741	0.827	0.881	0.970	0.814	0.945	0.783	0.896
Delaunay	0.573	0.315	0.857	0.799	0.892	0.899	0.881	0.886
K -NN	0.600	0.528	0.899	0.864	0.861	0.945	0.802	0.893

Our estimator outperforms the other ones for the highest level of density, *i.e.* for the correct detection of the high-density left rectangle. Percolation is in fact a fast enough phenomenon to occur in this zone at density $f \propto 4$ before taking place in the medium filament with close density $f \propto 3$. See Fig. 4: Almost the entire cluster is detected and only one (small) connected component of the thick filament appears. The other estimators have a density level much more uniformly distributed over the main clusters of close densities.

5 Conclusion and Perspectives

In this paper we propose a new estimator of density levels based on geometric graphs. Looking at what happens persistently allows us to observe percolation phases. Since continuum percolation is a very fast phenomenon, our estimator is able to identify two neighbouring levels of close density.

This estimator of density levels could find a natural application to the problem of identifying galaxy clusters, which are highly hierarchical. In addition, the availability of a graph makes it fairly easy to extract galaxy filaments without having to resort to methods such as calculating the median axis.

Compared with conventional density estimators for this type of problem, it is already showing very good results, especially for high-density clusters.

In the future, we will try to further improve these results focusing on four principal research directions: 1° We mainly looked at one type of clusters, ‘filaments’. Having an estimator of density levels allows us to look at other types, such as ‘super-clusters’, ‘walls’ and ‘voids’. 2° A galaxy was represented only by a point. Its mass (= its luminosity) could be taken into account using different radii, depending on galaxies. 3° The question of Σ_P for density levels was briefly considered in this paper. There are certainly wiser choices to be made (*e.g.* inspired by Penrose’s works [26]) to approach strong-consistency. 4° We worked with graphs. We could look at other notions of connectivity (*e.g.* connectivity of simplicial complexes).

References

1. 2df galaxy redshift survey. 2dF Galaxy Redshift Survey 2dF Galaxy Redshift Survey
2. Sloan digital sky survey. <http://www.sdss.org>
3. Longair, M., Einasto, J. (eds.). The Large Scale Structure of the Universe, International Astronomical Union Symposia, vol. 79, p. 464. Springer, Tallinn (1978). <https://doi.org/10.1007/978-94-009-9843-8>
4. Attali, D., Boissonnat, J.D., Edelsbrunner, H.: Stability and Computation of Medial Axes - a State-of-the-Art Report, pp. 109–125. Springer (2009). https://doi.org/10.1007/b106657_6
5. Barrow, J.D., Bhavsar, S.P., Sonoda, D.H.: Minimal spanning trees, filaments and galaxy clustering. MNRAS **216**(1), 17–35 (1985). <https://doi.org/10.1093/mnras/216.1.17>
6. Biau, G., Devroye, L.: Lectures on the Nearest Neighbor Method, vol. 246. Springer (2015). <https://doi.org/10.1007/978-3-319-25388-6>
7. Bobrowski, O., Kahle, M.: Topology of rand. geom. complexes: a survey. J. Appl. Comput. Top. **1**, 331–364 (2018). <https://doi.org/10.1007/s41468-017-0010-0>
8. Boissonnat, J.D., Wintraecken, M.: The reach of subsets of manifolds. J. Appl. Comput. Top. 1–23 (2023). <https://doi.org/10.1007/s41468-023-00116-x>
9. Bollobás, B., Riordan, O.: Percolation. Cambridge University Press (2006). <https://doi.org/10.1017/CBO9781139167383>
10. Campello, R.J.G.B., Moulavi, D., Sander, J.: Density-based clustering based on hierarchical density estimates. In: Advances in Knowledge Discovery and Data Mining, pp. 160–172. Springer, Berlin (2013). https://doi.org/10.1007/978-3-642-37456-2_14
11. Colberg, J.M.: Quantifying cosmic superstructures. MNRAS **375**(1), 337–347 (2007). <https://doi.org/10.1111/j.1365-2966.2006.11312.x>
12. Darvish, B., Mobasher, B., Sobral, D., Scoville, N., Aragon-Calvo, M.: A comparative study of density field estimation for galaxies. Astrophys. J. **805**(2), 121 (2015). <https://doi.org/10.1088/0004-637X/805/2/121>
13. Einasto, J.: Large scale structure of the Universe. AIP Conf. Proc. **1205**(1), 72–81 (2010). <https://doi.org/10.1063/1.3382336>
14. Ferdosi, B.J., Buddelmeijer, H., Trager, S.C., Wilkinson, M.H.F., Roerdink, J.B.T.M.: Comparison of density estimation methods for astronomical datasets. Astron. Astrophys. **531**, A114 (2011). <https://doi.org/10.1051/0004-6361/201116878>

15. Fréchet, M.: L'intégrale abstraite d'une fonction abstraite d'une variable abstraite et son application à la moyenne d'un élément aléatoire de nature quelconque. *La Revue Scientifique* (1944)
16. Gernez, P., Descombes, X., Zerubia, J., Slezak, E., Bijaoui, A.: Galaxy filament detection using the quality candy model. In: *IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 2 (2006). <https://doi.org/10.1109/ICASSP.2006.1660447>
17. Hall, P.: *Introduction to the Theory of Coverage Processes*. Probability and Mathematical Statistics. Wiley, Hoboken (1988)
18. Hartigan, J.A.: *Clustering Algorithms*. Wiley, Hoboken (1975)
19. Hartigan, J.A.: Consistency of single linkage for high-density clusters. *J. Am. Stat. Ass.* **76**(374), 388–394 (1981). <https://doi.org/10.1080/01621459.1981.10477658>
20. Kuchner, et al.: An inventory of galaxies in cosmic filaments feeding galaxy clusters. *MNRAS* **510**(1), 581–592 (2021). <https://doi.org/10.1093/mnras/stab3419>
21. Kullback, S., Leibler, R.A.: On Information and Sufficiency. *Ann. Math. Stat.* **22**(1), 79–86 (1951). <https://doi.org/10.1214/aoms/1177729694>
22. Libeskind, et al.: Tracing the cosmic web. *MNRAS* **473**(1), 1195–1217 (2017). <https://doi.org/10.1093/mnras/stx1976>
23. Meester, R., Roy, R.: *Continuum Percolation*. Cambridge Tracts in Mathematics. Cambridge University Press, Cambridge (1996). <https://doi.org/10.1017/CBO9780511895357>
24. Olkin, I., Pukelsheim, F.: The distance between two random vectors with given dispersion matrices. *Linear Algebra Appl.* **48**, 257–263 (1982). [https://doi.org/10.1016/0024-3795\(82\)90112-4](https://doi.org/10.1016/0024-3795(82)90112-4)
25. Penrose, M.: *Random Geometric Graphs*, vol. 5. Oxford University Press, Oxford (2003). <https://doi.org/10.1093/acprof:oso/9780198506263.001.0001>
26. Penrose, M.: Random Euclidean coverage from within. *Probab. Theory Relat. Fields* **185**(3–4), 747–814 (2023). <https://doi.org/10.1007/s00440-022-01182-5>
27. Quintanilla, J., Torquato, S., Ziff, R.: Efficient measurement of the percolation threshold for fully penetrable discs. *J. Phys. A* **33**(42), L399–L407 (2000). <https://doi.org/10.1088/0305-4470/33/42/104>
28. Schaap, W.E.: *Dtf : the delaunay tessellation field estimator*. Ph.D. thesis, Proefschrift Rijksuniversiteit Groningen (2007)
29. Stoica, R., Martínez, V., Mateu, J., Saar, E.: Detection of cosmic filaments using the candy model. *Astron. Astrophys.* **434**(2), 423–432 (2005). <https://doi.org/10.1051/0004-6361:20042409>
30. Tanemura, H.: Critical behavior for a continuum percolation model. *Probability Theory and Mathematical Statistics*, pp. 485–495 (1996)
31. Tempel, E., Stoica, R., Kipper, R., Saar, E.: Bisous model. *detect. filam. Patterns in p.p. A & C* **16**, 17–25 (2016). <https://doi.org/10.1016/j.ascom.2016.03.004>
32. Vaserstein, L.N.: Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredači Informacii* **5**(3), 64–72 (1969)
33. Vinay Kumar, B., Kashyap, N., Yogeshwaran, D.: An analysis of probabilistic forwarding of coded packets on random geometric graphs. *Perform. Eval.* **160**, 102,343 (2023). <https://doi.org/10.1016/j.peva.2023.102343>
34. Xu, W., Wang, J., Hu, H., Deng, Y.: Critical polyn. in the nonplanar and cont. Percol. Models. *Phys. Rev.* **103**, 022,127 (2021). <https://doi.org/10.1103/PhysRevE.103.022127>



Metric Invariants for Networks’ Classification

Eldad Kronfeld and Emil Saucan^(✉)

Department of Applied Mathematics, Braude College of Engineering, Snunit 57,
Karmiel, Israel
semil@braude.ac.il

Abstract. We suggest an approach to the shape DNA of data based on a number of metric invariants introduced by Grove and Markvorsen that encode its essential global geometry of the given structure. First experiments on real life networks and on natural images are given to demonstrate the feasibility of this approach. Even this incipient test clearly demonstrate the efficiency of the proposed invariants in the classification and understanding of stochastic textures as opposed to man-made ones.

Keywords: metric invariants · packing radii · packers · excess · natural textures

1 Introduction

Shape recognition, even under deformations, represents a problem of deep and continuing interest in Imaging, Graphics and, of course, Pattern Recognition. However, even in this rather mild conditions, where, even if distorted and noisy, the character/type of data – and, even more important – dimensionality – is supposed to be known (usually perceived as a smooth surface/manifold represented/approximated by a polyhedral mesh), the quest for the so called *shape DNA* is far from trivial and represents an open problem. The difficulty of the proposed task is proportionally much higher when one deals with unstructured (or weakly structured) data, such as clouds of points, where no manifold structure is supposed and certainly no smoothness assumptions are even remotely realistic. Given, however, that the one has to deal with basically the same issue, but in different settings – after all, in these fields one is supposed to infer the shape of the given object from the sample points available – one should begin where the Graphics/Imaging stops. Indeed, a quite general method was proposed in the “classical” context, that is easily not just adaptable, but also extensible to the more general context.

Motivated by the celebrated work of Gromov [1], metric methods, first proposed in [2], have become main stream tools in Graphics and Imaging – see, e.g. [3–5, 7, 7]. Such methods consist in *isometric embeddings* (usually into some Euclidean space, but other ambient spaces have also been considered) and *metric curvatures* (a natural connection existing between these two). Indeed, Gromov’s

K-curvature classes [1] are similar in spirit to the metric curvature approach mentioned above, as well as to the metric invariants we propose herein. However, there are serious problems, both theoretical as well as computational, in applying these ideas in practice – see [7–9].

We propose therefore a series of other purely metric invariants for shape recognition. These might complement, rather than supplement curvature and other classical, basic invariants, and we suggest, as the best method of employing the new invariants we propose is by creating a “dictionary” of metric “feature invariants”, and filter, so to say, the shape through by all these invariants. Another important feature that we emphasize is that some of the invariants are *local* (a typical classical example being Gauss curvature), whereas others are *global* (well known examples being diameter and volume). It is precisely through the combination of local and global *shape invariants* that one can identify the shape DNA of the data. Note that, in its common usage in Manifold Learning, the term “shape DNA” – see [10,11] – is not a geometric, but rather a spectral/Laplacian based approach. Clearly, geometric invariants are better described by this appellation and, moreover, they are far more intuitive and easily to represent visually. Furthermore, eigenvalues are global invariants, thus they cannot be used to understand the local (or even semi-local) structure of data. Thus to understand data at all its scales, one should use local and global invariants in tandem. In fact, a unifying approach to this problem, that allows the natural exploration in parallel of both these types of invariants via Forman’s Bochner-Laplacians and curvature measures was already proposed by the authors in [12]. In Differential Geometry this problem was attacked in [13] (see also [14] for a brief account of the problem and of the results). The basic idea is to extend the basic idea of metric geometry, that is looking at *pairs* of points (i.e. distances between points), to looking at *triples* (i.e. triangles), *quadruples* (which are closely related to curvature – see [1]), etc.

The organization of the remainder of this paper is quite simple: In Sect. 2 we introduce the proposed metric invariants; this being followed in Sect. 3 with some first experimental results, which are further discussed in the overview Sect. 4.

2 The Invariants

The new geometric *shape invariants* considered therein – and that we propose here for our own specific goals – are the following:

1. *Extent* We first bring the basic definition, namely

Definition 1. *Let (X, d) be a compact metric space. The q -extent of X , $\text{xt}_q X$, is the maximal average distance between q -tuples of points in X :*

$$\text{xt}_q X = \frac{1}{\binom{q}{2}} \max_{(x_1, \dots, x_q) \in X^q} \sum d(x_i, x_j).$$

A configuration of q points that realizes $\text{xt}_q X$ is called a q -extender. Moreover, $\text{xt} X = \lim_q \text{xt}_q X$ is called the extent of X .

Obviously, xt_2X is the *diameter* of X , $xt_2X = \text{diam}X$, and Grove and Markvorsen call the higher order extents similarly: xt_3X is the *triameter*, $xt_3X = \text{triam}X$, xt_4X is the *quadrameter*, etc.

The following inequalities hold:

$$xt_2X \geq xt_3X \geq \dots \geq xt_qX \geq xt_{q+1}X \geq \dots \geq xtX; \tag{1}$$

where $xtX = \lim_q xt_qX$ is the *extent* of X . Note also that $\frac{1}{2}\text{diam}X \leq xtX \leq \text{diam}X$. An important feature of this family of invariants resides in the fact that extents are sensitive to the asymmetries of X . They should be viewed, therefore, as *global shape invariants*. Thus they complement very well with such local invariants as the various *metric curvatures*.

2. *Excess* We begin with the following basic (and classical)

Definition 2. *Given a (geodesic) triangle $T = \Delta(pxq)$ in a metric space (X, d) , the excess of T is defined as*

$$\text{exc}(T) = d(p, x) + d(x, q) - d(p, q).$$

Sometimes the excess is also denoted more concisely as $e(T)$. The excess of X itself is a global version of the definition above: Given a metric space (X, d) , the excess of X is defined as

$$\text{exc}X = \min_{(p,q)} \max_x (e(\Delta(pxq))).$$

We should also note that both global and local variations of this quantity have been considered by Otsu [15].

A variation of this quantity has also been considered, namely the so called (after [15]) *global big excess*:

$$E(X) = \max_q \min_p \max_x (e(\Delta(pxq))).$$

A local version of the notion of excess – introduced, it seems by Otsu [15] – also exist, namely the *local excess* (or, more precisely, the *local d-excess*):

$$e(x) = \max_p \max_{x \in B(p,d)} \min_{q \in S(p,d)} (e(\Delta(pxq))),$$

where $d \leq \text{rad}(X) = \min_p \max_q d(p, q)$, (and where $B(p, d), S(p, d)$ stand – as they commonly do – for the ball and respectively sphere of center p and radius d).

Obviously, one has the following inequalities: $0 \leq \text{exc}M \leq \text{diam}M$. More importantly, there is a connection between the extent xtX and the excess $\text{exc}X$ of X , more precisely *small extent implies small excess*. This statement can be formulated in a precise, *quantitative* form, as follows:

Proposition 1 ([13], **Proposition 1.12**). *Let X be a compact metric space. Then, for any $\varepsilon > 0$, there exists $\delta > 0$, such that $\text{exc}X < \varepsilon \text{diam}X$, if $xtX \leq (\frac{1}{2} + \delta)\text{diam}X$.*

While the reciprocal of the assertion above is not true, for odd q (and not for even q 's), the following strong result holds: If $\text{xt}_q X$ is minimal, that is if $\text{xt}X = \frac{1}{2}\text{diam}X$, then $\text{exc}X$ (for details see [13]).

The geometric “content” of the notion of local excess is that, for any $x \in B(p, d)$, there exists a (minimal) geodesic γ from p to $S(p, d)$ such that γ is close to x . Moreover, it is intuitively clear that (local) excess and curvature are closely related concepts since the geometric “content” of the notion of local excess resides in the fact that, for any $x \in B(p, d)$, there exists a (minimal) geodesic γ from p to $S(p, d)$ such that γ is close to x . (See also [16] for a different approach to metric curvature via a 3 points condition.) The type of curvature specifically connected to excess is the so called *Haantjes curvature* or *Finsler-Haantjes curvature* (which was introduced by Haantjes [17], who extended to metric spaces an idea proposed by Finsler in his PhD Thesis.) As we shall see, it represents a simple and direct alternative – at least for many applications – of more involved and fashionable concepts.

Definition 3 (Haantjes curvature). *Let (M, d) be a metric space and let $c : I = [0, 1] \xrightarrow{\sim} c(I) \subset M$ be a homeomorphism, and let $p, q, r \in c(I)$, $q, r \neq p$. Denote by \widehat{qr} the arc of $c(I)$ between q and r , and by qr line segment from q to r .*

We say that c has Haantjes curvature $\kappa_H(p)$ at the point p iff:

$$\kappa_H^2(p) = 24 \lim_{q,r \rightarrow p} \frac{l(\widehat{qr}) - d(q, r)}{(l(\widehat{qr}))^3}; \tag{2}$$

where “ $l(\widehat{qr})$ ” denotes the length – in intrinsic metric induced by d – of \widehat{qr} .

Alternatively, since for points where Haantjes curvature exists, $\frac{l(\widehat{qr})}{d(q,r)} \rightarrow 1$, as $d(q, r) \rightarrow 0$ (see [17]), κ_H can be defined (see, e.g. [18]) by

$$\kappa_H^2(p) = 24 \lim_{q,r \rightarrow p} \frac{l(\widehat{qr}) - d(q, r)}{(d(q, r))^3}; \tag{3}$$

In applications it is this alternative form of the definition of Haantjes curvature that will prove to be more malleable, as we shall illustrate shortly.

In any of its versions, the intuition behind the notion of Haantjes curvature is quite transparent: The longer is the arc as compared to the chord, the more “curved” it is. (The longer the bow is in comparison to its string, the more “bowed”, i.e. curved it is.) However, its complicated form is far less intuitive. For now, let us observe that it is proportional to $1/l$ (or $1/d$), which hints to the radius of curvature (and to Menger curvature). Less transparent and definitely more cumbersome is, however, the factor of “24” appearing in the definition. However, the two are interrelated and that the “24” factor arises naturally, in the course of the essential theorem below (due to Haantjes):

Theorem 1. *Let $\gamma \in \mathcal{C}^3$ be smooth curve in \mathbb{R}^3 and let $p \in \gamma$ be a regular point. Then the metric curvature $\kappa_H(p)$ exists and equals the classical curvature of γ at p .*

Simply put, for smooth curves in the Euclidean plane (or space), Haantjes curvature coincides with the standard (differential) notion, proving that, it represents, indeed, a proper generalization of the classical concept of curvature.

Due to its intuitive definition, as well as the simplicity of its computation, Haantjes curvature has proven to be very malleable and useful in a variety of tasks in networks' practice, such as clustering for DNA microarray analysis [19], wavelets intelligence and texture segmentation [20], financial markets understanding [21], deep learning [21], neural [23] and semantic [24] networks. However, as we have already observed above, while the idea behind Haantjes curvature is quite simple, the notion itself appears somewhat cumbersome. Therefore, it is only natural to try and simplify it – even at the price of discarding a dimensionality condition – so long as the essential geometric motivation is preserved. This motivation, as well as the expression of Haantjes curvature, brings us the connect it to the notion of *excess*. More precisely, we have the following relation between the two notions:

$$\kappa_H^2(T) = \frac{e}{d^3}, \tag{4}$$

where by the curvature of a triangle $T = T(pxq)$ we mean the curvature of the path \widehat{pxq} . Here and below we have used a simplified notation and discarded (for sake of simplicity and clarity) the normalizing constant “24”. Thus Haantjes curvature can be viewed as a *scaled* version of excess. Keeping this in mind, one can define also a global version of this type of metric curvature, namely by defining, for instance:

$$\kappa_H^2(X) = \frac{E(X)}{\text{diam}^3(X)}, \tag{5}$$

or

$$\kappa_H^2(X) = \frac{e(X)}{\text{diam}^3(X)}, \tag{6}$$

as preferred. To be sure, one can proceed in the opposite direction and express the proper (i.e. point-wise) Haantjes curvature by means of the definition (2) of local excess, as

$$\kappa_H^2(x) = \lim_{d \rightarrow 0} e(x). \tag{7}$$

3. *Packing Radius* Another useful family of (geo-)metric invariants is that of *packing radii*:

Definition 4. *The q -th packing radius of X , $\text{pack}_q X$, is the largest r for which X contains q disjoint open balls of radius r , i.e.*

$$\text{pack}_q X = \frac{1}{2} \min_{(x_1, \dots, x_q)} \max_{1 \leq i < j \leq q} d(x_i, x_j), \quad x_1, \dots, x_q \in X^q.$$

A configuration of q points that realizes $\text{pack}_q X$ is called a q -packer.

We have the following sequence of inequalities, akin to (1):

$$\frac{1}{2} \text{diam} X = \text{pack}_2 X \geq \text{pack}_3 X \geq \dots \geq \text{pack}_q X \geq \dots \geq \lim_q \text{pack}_q X = 0. \tag{8}$$

While generally assuring just the recognition of the *topology type* of a space, in certain instances all the invariants above provide a solution for the recognition problem from the *metric* viewpoint as well. In particular, since these metric invariants can be used to completely identify/characterize certain metric spaces, e.g. spheres, [13, 14], the closeness/departure of their counterparts would/could be used, for instance, as a measure of closeness of networks to sphericity.

3 Experiments

We experiment with the main of the proposed invariants on real life complex networks as they are discrete enough to represent good test-cases and, moreover, they are convenient, easy storable, efficient and an increasingly popular way of representing (as graphs) and storing of data sets that do encode complex structures and phenomena. Moreover, they can be effectively analyzed with data mining methods. Furthermore, networks represent the middle ground, in regard of complexity and abstractiveness between the relatively tame meshes common in Graphics and Imaging, and clouds of points, which are more difficult to handle due, in part, to a lack of geometric intuitiveness (and which, to be dealt with in practice, are in many cases transformed to graphs, for this very reason).

In addition, we experimented with the square grids that arise naturally in Imaging. The reason we considered such simple graphs is that they retain much of the geometric, visual content of the original image, thus they allow us the better comprehend the intuition behind the considered invariants and thus better estimate their effectiveness.

Remark 1. Note that we do not include here experiments with Haantjes curvature, both because we explored it extensively elsewhere – see the previously cited articles, and also due to its strong correlation with the notion of excess, which is far more relevant to our present study, and which we shall also explore in a sequel study.

3.1 The Data Sets

We applied our invariants to five small and four medium standard networks whose source and further details can be found in [25].

We also considered square grids corresponding to 11×11 patches (Fig. 1, above) from a natural image (Fig. 1, below). (The choice of the size of the patches is a compromise between practicality and interest, as it lies, on the one hand, at the upper end of neighborhoods dimensions that encode local, geometric properties of an image, rather than statistical ones, and our desire to consider networks of at least medium size.) The width of the grid's edges is proportional to the length of the edge, i.e. the distance between the centers of the pixels in the standard stick model [26].

We examined the efficiency of the q -extents in the understanding of the geometry of networks by testing them, for $q = 3$ and $q = 4$, on the real life networks

Table 1. The q -extents, $q = 3$, $q = 4$ and $q = 5$, of five small networks. Note that xt_qX increases with q and it is inverse proportional to the size of the network.

xt_qX	$q = 3$	$q = 3$	$q = 5$
Enzymes	$3.42E - 03$	$5.42E - 03$	0.007419354839
Ecological	$3.69E - 04$	0.0005536417323	0.0007381889764
Infections	$3.95E - 04$	$6.32E - 04$	0.0008691529709
Emails	0.0006894513937	0.001034177091	0.001378902787
Chemical	0.0002801120448	0.0004201680672	0.0005602240896

Table 2. The q -th packing radius, for $q = 3$, $q = 4$ and $q = 5$ of the same networks as in Table 1. Note that, at least for these small sized networks, the packing radii convey little information and, moreover, they are quite stationary in most cases.

$pack_qX$	$q = 3$	$q = 3$	$q = 5$
Enzymes	0.75	2.25	4.25
Ecological	0.5	0.5	0.5
Infections	0.5	0.5	0.5
Emails	0.75	0.75	0.75
Chemical	0.5	0.5	0.5

in Table 1. We observe that xt_qX increases as a function of q and, moreover, that it is larger on smaller networks.

We also experimented with the q -extents and q -th packing radii, for the same q -s as before, as well as for the extent, on the four 11×11 patches of natural images in Fig. 2.

The results for a typical patch are systemized in Table 3, and they clearly show how efficient the simple metric invariants proposed in the present paper are in distinguishing stochastic textures from repetitive, man-made ones. Indeed, both xt_3 and xt_4 equal the same number for all the stochastic textures considered, while for the (essentially) periodic one the values are different both from the ones obtained for the other textures, as well as one from each other.

Table 3. The q -extents, $q = 3$ and $q = 4$, of four medium sized networks. Note that xt_qX increases with q and inverse proportionally to the size of the network.

xt_qX	$q = 3$	$q = 3$
Biological	$4.95E - 06$	$7.43E - 06$
Mouse brain	$8.86E - 05$	0.0001328727079
Dolphins	0.0001303471975	0.0002014456689
Economical	$5.70E - 06$	$8.55E - 06$

Table 4. The q -th packing radius, for $q = 3$ and $q = 4$ of the medium sized networks in Table 3. Note that, at least for these small sized networks, the packing radii convey little information and, moreover, they are quite stationary in most cases.

$\text{pack}_q X$	$q = 3$	$q = 4$
Biological	0.25	0.25
Mouse brain	0.5	0.5
Dolphins	1.25	1.25
Economical	0.074999973	0.074999973

The histograms of these invariants for exhaustion of these textures by 11×11 blocks also clearly demonstrate both the difference between the natural versus man-made textures, as well as the capacity of the suggested metric invariants to clearly distinguish between them. To wit the histogram of artificial texture presents a “double hump” distribution, as opposed to the gaussian-type of the natural ones, while, in contrast, in the case of the excess, the natural textures display almost a δ -function distribution, in contrasted with the multiple peaks for the artificial one.

Computation of the considered invariants is quite efficient. To wit, computing them on one of the small networks ($|V| = 125$) necessitates, on a Intel i9-13900K, 32GB RAM machine, between $0.02s$ for xt_3 and $0.06s$ for the excess, to $1.61s$ for xt_4 . For a larger one ($|V| = 636$), times are, respectively $1.16s$, $4.3s$ and $826.63s$. However, it should be noted that, due to the fact that the computation complexity of xt_q and pack_q is $O(|V|^q)$ computing them for high q -s on large networks might prove challenging.

The code residing behind the results above is accessible at the following GitHub link: <https://github.com/Eldadkro/ShapeOfData>.

Table 5. The $\text{exc}X$ of the considered networks.

Network	$\text{exc}X$
Enzymes	62
Ecological	5
Infections	5
Emails	15
Chemical	3
Biological	19
Mouse brain	3
Dolphins	11
Economical	1.199999684

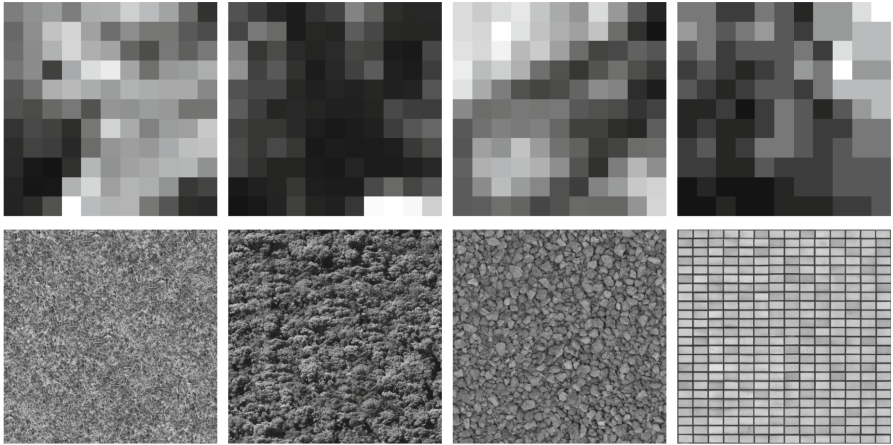


Fig. 1. The considered 11×11 patches (above) from natural images (below). From left to right: Two patches of a natural landscape textures (grass and forest, respectively); gravel; and a man made texture (bricks).

Table 6. The q -extents, q -th packing radii for $q = 3$ and $q = 4$, and excess, of the considered textures. Note that for the stochastic textures the extents convey little information and, moreover, they are quite stationary in most cases.

	pack ₃	pack ₄	exc	xt ₃	xt ₄
Grass	6.25	12.5	622	0.00833333333333	0.00833333333333
Forest	6	8.5	384	0.00833333333333	0.00833333333333
Gravel	2.5	3.5	298	0.00833333333333	0.00833333333333
Bricks	1.5	1.25	54	0.0002066115702	0.0003443526171

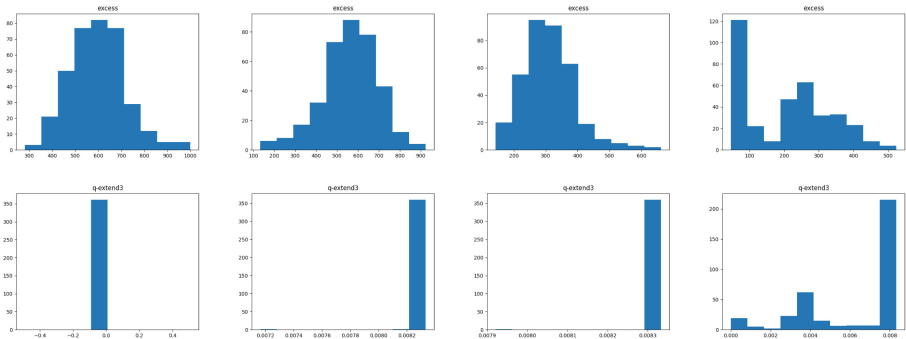


Fig. 2. The histograms of excess (above) and xt_3 of the textures in Fig. 1. Note that the histogram of artificial texture presents a “double hump” distribution, as opposed to the gaussian-type of the natural ones. The difference between natural (stochastic) and artificial textures is quite striking in the histograms of $pack_3$.

4 Conclusions and Future Study

The experiments presented above are clearly only incipient ones and meant to function as capability proof. In particular, we concentrated only on three main invariants and postponed the systematic exploration of the others, which we discussed in another context previously, for later study. Therefore, the first and foremost future task is to extend the experiments on larger and more diverse data sets, as well as including more invariants and using them jointly to understand and classify data sets.

Another extension that naturally imposes itself is passing from the relatively tame setting of networks to the even more intriguing, but harder to handle, setting of clouds of points. However, a restriction in scope, rather than its expansion, would represent yet another interesting direction of study. Namely, one should apply the metric invariants proposed in this paper in the context of understanding and classification of textures not just of natural images, but also in the more important and challenging setting of 3D CT and MRI medical images. Indeed, given the manifest efficiency of the suggested invariants in the understanding of natural images, their use in the classification of 3D textures might prove as a new useful tool in computer assisted diagnosis.

Furthermore, the estimation of the efficiency in geometric data analysis of the new invariants should also be done by comparing their performance with that of more established methods, such as various types of graph curvatures and Persistent Homology.

Yet one more augmentation of the present study is by experimenting on more of the suggested invariants, as well as by considering more involved ones, first and foremost among those being Gromov's *filling radius* and Urysohn's *intermediate diameters* (see [1] for the now classical exposition of these notions). However, we have to admit that, at this stage, we are not capable of performing meaningful experiments with these more difficult to handle, deep invariants, and we must therefore satisfy ourselves to suggesting them and to postpone experiments for a non-immediate future. For future research we also leave the exploration of the connections between our approach based upon the works of Grove and Markvorsen and the ideas in [28].

Acknowledgments. Research partially supported by the GIF Research Grant No. I-1514-304.6/2019.

The authors would like to thank Anthea Monod and the anonymous reviewers for the help in improving our exposition.

References

1. Gromov, M.: Metric Structures for Riemannian and Non-Riemannian Spaces. Birkhauser, Second printing (2001)
2. Saucan, E.: Surface triangulation – the metric approach. [arxiv:cs.GR/0401023](https://arxiv.org/abs/cs/0401023) (2004)

3. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Isometric embedding of facial surfaces into \mathbb{S}^3 . In: Kimmel, R., Sochen, N.A., Weickert, J. (eds.) *Scale-Space 2005*. LNCS, vol. 3459, pp. 622–631. Springer, Heidelberg (2005). https://doi.org/10.1007/11408031_53
4. Bronstein, A., Bronstein, M., Kimmel, R.: Three-dimensional face recognition. *Int. J. Comput. Vision* **64**(1), 5–30 (2005)
5. Memoli, F.: On the use of gromov-hausdorff distances for shape comparison. In: *Proceedings of Symposium on Point Based Graphics, Prague (2007)*
7. Saucan, E., Appleboim, E.: Metric methods in surface triangulation. In: Hancock, E.R., Martin, R.R., Sabin, M.A. (eds.) *Mathematics of Surfaces 2009*. LNCS, vol. 5654, pp. 335–355. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03596-8_20
7. Saucan, E.: A metric Ricci flow for surfaces and its applications. *Geom. Imaging Comput.* **1**(2), 259–301 (2016)
8. Saucan, E.: Isometric embeddings in imaging and vision: facts and fiction. *J. Math. Imaging Vis.* **43**(2), 43–155 (2012)
9. Memoli, F.: The Gromov-Hausdorff distance: a brief tutorial on some of its quantitative aspects. *Actes des rencontres du CIRM* **3**(3), 335–341 (2014)
10. Reuter, M., Wolter, F.-E., Peinecke, N.: Laplace-spectra as fingerprints for shape matching. In: *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling*, pp. 101–106 (2005)
11. Reuter, M., Wolter, F.-E., Peinecke, N.: Laplace-Beltrami spectra as ‘Shape-DNA’ of surfaces and solids. *Comput. Aided Des.* **38**, 342–366 (2006)
12. Saucan, E., Jost, J.: Network topology vs. geometry: from persistent homology to curvature. In: *Proceedings of NIPS LHDS (2016)*. <http://www.cs.utexas.edu/~rofyu/lhds-nips16/papers/11.pdf>
13. Grove, K., Markvorsen, S.: New extremal problems for the Riemannian recognition program via Alexandrov geometry. *J. American Math. Soc.* **8**, 1–28 (1995)
14. Grove, K., Markvorsen, S.: Curvature, Triameter and beyond. *Bull. (New Ser.) Am. Math. Soc.* **27**(2), 261–265 (1992)
15. Otsu, Y.: On manifolds with small excess. *Amer. J. Math.* **115**, 1229–1280 (1993)
16. Bačák, M., Hua, B.B., Jost, J., Kell, M., Schikorra, A.: A notion of nonpositive curvature for general metric spaces. *Diff. Geom. Appl.* **38**, 22–32 (2015)
17. Haantjes, J.: Distance geometry. Curvature in abstract metric spaces, *Proc. Kon. Ned. Akad. v. Wetenseh. Amsterdam* **50**, 496–508 (1947)
18. Kay, D.C.: Arc curvature in metric spaces. *Geom. Dedicata.* **9**(1), 91–105 (1980)
19. Saucan, E., Appleboim, E.: Curvature based clustering for DNA microarray data analysis. LNCS **3523**, 405–412 (2005)
20. Appleboim, E., Hyams, Y., Krakovski, S., Sageev, C., Saucan, E.: The scale-curvature connection and its application to texture segmentation. *Theor. Appl. Math. Comput. Sci.* **3**(1), 38–54 (2013)
21. Samal, A., Pharasi, H.K., Ramaia, S.J., Saucan, E., Jost, J., Chakraborti, A.: Network geometry and market instability. *R. Soc. Open Sci.* **8**, 201734 (2021)
22. Saucan, E., Samal, A., Jost, J.: A simple differential geometry for complex networks, network science. *R. Soc. Open Sci.* **8**, 201734 (2021)
23. Elumalai, P., Yadav, Y., Williams, N., Saucan, E., Jost, J., Samal, A.: Graph Ricci curvatures reveal atypical functional connectivity in autism spectrum disorder. *Sci. Rep.* **10**, 10819 (2022)
24. Cohen, H., Nachshon, Y., Maril, A., Naim, P.M., Saucan, E.: A path-curvature measure for word-based strategy searches in semantic networks. *Symmetry* **14**(10), 1737 (2022)

25. Rossi, R., Ahmed, N.K.: Network repository. <https://networkrepository.com/> (2012)
26. Pratt, W.K.: Digital Image Processing. Wiley, New York (2001)
27. Barkanass, V., Chen, W., Lei, N., Saucan, E.: Geometric graph measures for textures classification, preprint. <https://doi.org/10.13140/RG.2.2.19509.14566> (2023)
28. Joharinad, P., Jost, J.: Topology and curvature of metric spaces. *Adv. Math.* **356**, 106813 (2019)



The Hidden-Degree Geometric Block Model

Stefano Guarino¹, Enrico Mastrostefano¹, and Davide Torre^{1,2,3(✉)}

¹ Institute for Applied Mathematics “Mauro Picone” (IAC), CNR, Rome, Italy
d.torre@iac.cnr.it

² Libera Università Internazionale degli Studi Sociali “Guido Carli” (LUISS),
Rome, Italy

³ Campus Bio Medico University of Rome (Università Campus Bio-Medico
di Roma), Rome, Italy

Abstract. Defining accurate models for real-world social networks is essential across various research fields including sociology, epidemiology, and marketing. Such models serve as indispensable tools to capture the dynamics of phenomena ranging from disease spread to rumor dissemination, encapsulating intricate patterns of interactions among individuals within a population. To this end, a latent geometry and/or hidden degrees can be used to obtain networks that are small-world, highly clustered, and have a scale-free degree distributions.

This study aims to integrate group mixing within the framework of latent geometry models. Our approach is based on conceptualizing a graph with a planted partition as the union of different mono- and bipartite subgraphs, for intra- and inter-block edges, respectively. We highlight that the hidden degree – the analogous of the radial coordinate in purely geometric hyperbolic models – must be replaced by a hidden fitness, and that all latent features must be assigned to the nodes once and for all, rather than once for each subgraph.

Through extensive simulations, we show that the proposed model generates networks with a unique combination of features, that cannot be obtained with standard geometric models nor with maximum entropy degree-corrected block models.

1 Introduction

Accurate models for real-world social networks are decisive tools to capture the dynamics of various phenomena across diverse research domains, ranging from the spread of diseases to the dissemination of rumors, by enclosing complex patterns of interactions among individuals within a population.

The deep impact of a network’s structure on dynamic processes is well-documented [1]. Urban social networks, with their distinct topologies, sizes, and demographic compositions, can significantly influence the transmission of diseases [2] and other phenomena within urban environments [3]. As researchers seek to gain deeper insights into the basic mechanisms governing real-world network formation, a variety of network models have emerged, each aiming to reflect

and potentially unfold specific observed features of complex networks [4]. In particular, three key properties are extensively encountered in real-world networks: a heavy-tailed degree distribution, high transitivity and some meso-scale community structure. Formulating sound and general models that encompass all of these characteristics turned out to be a formidable challenge.

Entropy maximization [5] and latent geometry [6] are two popular approaches to random network modeling that allow to generate networks with desirable topological properties relying on an elegant formulation. However, both families of models have limitations that have not been thoroughly addressed in the literature. Maximum entropy models are generally used to understand whether other structural features emerge naturally from local constraints such as vertex degree [7]. This is not the case for high clustering, a property that must be explicitly imposed, with constraints that significantly increase the complexity of the model. The \mathbb{S}^1 model and its quasi-isomorphic \mathbb{H}^2 model address this issue introducing a latent similarity space [8–10]. Through this notion of vertex homophily, these models allow to generate networks characterized by a high clustering coefficient and a scale-free degree distribution [11], but they lack parameters to control the existence of blocks – or communities – with specific mixing patterns, another common feature in real-world networks. Recent work highlighted that the dimensionality of the latent space impacts on the meso-scale structural properties that can be imposed to the network [12], yet no previous work ever clarified how arbitrary mixing patterns could be enforced in latent geometric models.

In this paper, we make a step towards filling this gap by presenting the Hidden-degree Geometric Block Model (HGBM), a generalization of the \mathbb{S}^1 model for networks with a planted partition. The rationale of our model is splitting the graph into a set of mono- and bipartite subgraphs – for intra- and inter-block edges, respectively – and separately modeling each of these subgraphs as a \mathbb{S}^1 network. We provide two main insights: (i) the concept of hidden degree must be replaced with a hidden fitness, which is to be normalized based on the connectivity of the pertaining block; (ii) the latent features of each node must be extracted once and for all, so that the centrality and homophily patterns are preserved across different subgraphs. Through extensive simulations, we show that our HGBM can be used to obtain all three properties together – the desired block mixing and degree distribution, and high clustering – differently from both the maximum-entropy Fitness-Corrected Block-Model and the \mathbb{S}^1 model.

1.1 Related Work

Various random graph models have been proposed in computational social sciences to capture the key features of real-world social networks, such as heavy-tailed degree distribution, high transitivity, positive degree and type assortativity, summarized in [13].

Many real-world social networks often display a form of group mixing, driven by individuals’ natural inclination to socialize with others who share similar interests or attributes, as highlighted by McPherson et al. [14]. This characteristic can be effectively replicated using the Stochastic Block Model (SBM), which

has gained prominence as a means of generating networks with predefined community structures [15, 16]. In the original SBM formulation, nodes within the same block are considered indistinguishable, resulting in a degree distribution that tends to resemble a Poisson distribution for larger graphs [15]. To create networks that better capture real-world characteristics, such as heavy tailed degree distributions, several model extensions have been suggested in the literature. These extensions include the Degree-Corrected Block Model (DCBM) [15] and its maximum-entropy variant [17]. In a way, these models can be considered as the Stochastic Block Model's counterparts to the widely recognized configuration model. They are designed to simultaneously accommodate the desired group mixing patterns and a specified degree sequence, either exactly or on average. These models, however, struggle to replicate the high local connectivity, i.e. high clustering, observed in real-world networks.

On the other hand latent space models offer a promising approach to generating random networks with a heavy-tailed degree distribution and high transitivity, drawing on the idea that network centrality and homophily can be captured by hidden metric spaces. In [6] the authors show that clustering can naturally emerge as a consequence of the triangular inequality within a hidden metric space. They introduced the S^1 class of network models, embedding nodes in a metric space and establishing connections based on a gravity-law-like probability that balances node distance and degrees. This definition incorporates both similarity distance and node significance, enabling the model to generate scale-free, small-world, and clustered graphs resembling real complex networks. In a subsequent work [8], the theory of random geometric graphs in hyperbolic geometry was presented. Remarkably, this formalism naturally produces scale-free graphs, suggesting hyperbolic geometry as an ideal framework for modeling complex networks. Incorporating group mixing in a hyperbolic setting is not straightforward due to the relationship between the rules for distributing vertices in the hyperbolic space and the rules for connecting vertices based on their distance.

Finally, entropy-based models have been extensively used in the last 20 years as null-models for – or randomized versions of – real complex networks [5]. One fundamental paper in this area is that of Park and Newman [18], who interpreted the general framework of Exponential Random Graphs (ERGs) [19] in terms of maximum entropy models. The entropy-based model was later tailored on observed networks [7, 20]. This approach has been widely used to study structural patterns in various systems, including financial and trade networks, biological systems, and online social networks. The general framework can be extended to different kinds of networks, including undirected, directed, weighted, directed and weighted, bipartite, bipartite weighted, and degree corrected block models.

2 Methods

Let \mathcal{G} be the ensemble of all simple graphs of N vertices. All graphs in \mathcal{G} have vertex set $V = \{v_i\}_{i=0}^{N-1}$, which we assume to be partitioned into n blocks $\{B_I\}_{I=0}^{n-1}$. From here on, we shall use the lowercase indices (e.g., i, j) for vertices, and

uppercase indices (e.g., I, J) for blocks. The size of block I is $N_I = |B_I|$ and, for each $v_i \in V$, I_i denotes the index of the block to which v_i belongs. For the sake of simplicity, we will often refer a vertex/block by its index, e.g., $i \in I$ means $v_i \in B_I$.

Let $A(G) = \{a_{ij}(G)\}_{i,j=0}^{N-1}$ be the adjacency matrix of G , i.e., $a_{ij}(G) = 1$ if edge $(i, j) \in E(G)$ and $a_{ij}(G) = 0$ otherwise. The degree of vertex v_i in G is $\deg_i(G) = \sum_j a_{ij}(G)$. The total degree of block I is $\deg_I(G) = \sum_{i \in I} \deg_i(G)$ and, for all I, J , $L_{IJ}(G) = \sum_{i \in I} \sum_{j \in J} a_{ij}(G)$ is the number of edges between B_I and B_J in G , or, if $I = J$, twice that number. Using the definitions, $\deg_I(G) = \sum_J L_{IJ}(G)$. For the sake of simplicity, the dependence of these quantities on the specific graph G will be often omitted in the following.

A random model for networks of fixed size $N > 0$ can be thought of as a probability distribution P over \mathcal{G} , so that $P(G)$ is the probability that the model produces the graph $G \in \mathcal{G}$. For given P , $\langle \cdot \rangle_P$ denotes the expectation with respect to P .

2.1 Hyperbolic Geometric Models

To obtain random networks that are scale-free, small-world, and clustered, the \mathbb{S}^1 model [6, 10] assigns to each node both random angular coordinates in a metric similarity space and a *hidden* degree. The edge probability takes the form of a gravity law, with hidden degrees playing the role of masses. The clustering of the graph is controlled by the temperature of the model – or, as often done in the literature, by the inverse temperature β . For instance, when $\beta \rightarrow +\infty$, an edge is drawn between v_i and v_j if and only if $d_{ij} < \mu \kappa_i \kappa_j$, where d_{ij} is their distance on the circle, κ_i, κ_j are their hidden degrees, and μ controls the average degree of the network. It can be proven that $\langle \deg_i \rangle_{\mathbb{S}^1} = \kappa_i$, where the mean $\langle \cdot \rangle_{\mathbb{S}^1}$ is taken with respect to the choice of the coordinates of the nodes on the circle.

Network with similar desirable properties can also be generated with a purely geometric model. In the \mathbb{H}^2 model, the nodes are distributed within the Poincaré disk and the dissimilarity between nodes is measured by their hyperbolic distance. In the limit $\beta \rightarrow +\infty$ – which, again, maximizes clustering – nodes are connected by edges if their hyperbolic distances are less than a threshold that depends on the network density. The properties of the hyperbolic space guarantee that points near the disk center have a higher expected degree. In particular, the \mathbb{H}^2 model is quasi-isomorphic to the \mathbb{S}^1 model with hidden degrees κ_i drawn from a power-law distribution of exponent $\gamma > 2$ [8]. For the sake of simplicity, the model proposed in this paper is based on the \mathbb{S}^1 model, but it could be (almost) equivalently reformulated in a hyperbolic geometry framework.

In the \mathbb{S}^1 model, the only way to generate assortative communities is placing the nodes of the same group close to each other on the circle, and farther away from the other groups [12]. With communities defined by their position on the circle, in fact, edges occur mainly within groups or between “consecutive” groups in a linear ordering. This limitation can be addressed using a higher-dimensional latent space [12], but, to the best of our knowledge, there is no known generalization of the \mathbb{S}^1 model that allows enforcing arbitrary community structures.

2.2 The HGBM Model

Say that we want to generate random networks with the following properties:

$$\langle L_{IJ} \rangle_P = K_{IJ} \quad \text{for all } I, J \tag{1}$$

$$\langle \text{deg}_i \rangle_P = \kappa_i \quad \text{for all } i \tag{2}$$

In words, (1) specifies the network’s expected block mixing patterns, while (2) specifies the network’s expected degree distribution. Conditions (1) and (2) are only consistent if $\sum_J K_{IJ} = \sum_{i \in I} \kappa_i$ for all I , which is false in general if the K_{IJ} are given (e.g., data-driven), while the hidden degrees κ_i are drawn from a probability distribution (e.g., a power-law).

A way to guarantee the internal consistency of the model is drawing a vertex-intrinsic *fitness* f_i rather than directly drawing the hidden degree κ_i . We then set

$$\kappa_i = \frac{f_i}{\sum_{h \in I} f_h} \sum_J K_{IJ} \tag{3}$$

where f_i says how well connected v_i is with respect to the other nodes in its block. If the f_i ’s are drawn from a power-law distribution with exponent γ , then the tail of the distribution of the κ_i ’s approximately follows the same distribution [21].

Now, let us consider the decomposition of G into the subgraphs $G_{IJ} = (V_{IJ}, E_{IJ})$, such that $V_{IJ} = B_I \cup B_J$ and $E_{IJ} = \{(i, j) \in E : i \in I, j \in J\}$. Our HGBM model works by generating each G_{IJ} separately, to then merge them together to obtain G . Using (3), (1) and (2) can be rewritten as

$$2\langle E_{II} \rangle_P = K_{II} \tag{4}$$

$$\langle \text{deg}_i(G_{II}) \rangle_P = \frac{f_i}{\sum_{h \in I} f_h} K_{II} \quad \text{for all } i \in I \tag{5}$$

for all I , and

$$\langle E_{IJ} \rangle_P = K_{IJ} \tag{6}$$

$$\langle \text{deg}_i(G_{IJ}) \rangle_P = \frac{f_i}{\sum_{h \in I} f_h} K_{IJ} \quad \text{for all } i \in I \tag{7}$$

$$\langle \text{deg}_j(G_{IJ}) \rangle_P = \frac{f_j}{\sum_{h \in J} f_h} K_{IJ} \quad \text{for all } j \in J \tag{8}$$

for all $I \neq J$. Conditions (4) and (5) can be used to generate G_{II} with the standard \mathbb{S}^1 model, while conditions (6), (7) and (8) can be used to generate G_{IJ} with the bipartite version of \mathbb{S}^1 introduced in [22].

It is worth noting that both the fitness f_i and the angular coordinate $\theta_i \sim \mathcal{U}(0, 2\pi)$ are drawn just once and for all – i.e., not re-drawn for each subgraph. The fact that v_i has the same f_i in all subgraphs guarantees that (3) holds and that G has, in good approximation, the desired degree distribution. On the other hand, the fact that v_i has the same θ_i in all subgraphs preserves the transitivity of the vertex similarity across different subgraphs, so as to guarantees the desired high clustering for the entire graph.


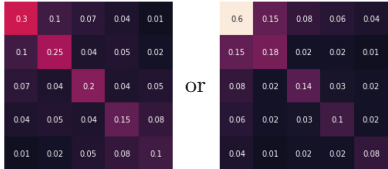
3 Simulation-Based Analysis

In this section, we experimentally evaluate the main properties of the HGBM model. To do so, we compare our model with two benchmark models:

- the Fitness-Corrected Block-Model [21], defined as the maximum-entropy model satisfying (1) and (2), with κ_i given by (3);
- the \mathbb{S}^1 model, with random angular coordinates but with the hidden degrees given by (3).

For the experiments, we considered all 16 combinations of the parameter values reported in Table 1. In words: we fixed the number of blocks to $n = 5$, the power-law exponent to $\gamma = 2.5$ and the inverse temperature of the model to $\beta = 10$; we let vary the size of the network ($N = 10\text{K}$ and $N = 100\text{K}$), the density of the network ($\mu = 10$ and $\mu = 100$), the size of the 5 blocks (perfectly balanced and very unbalanced), and the mixing matrix (relatively balanced and very unbalanced). In the following, however, we will only report the results obtained for specific configurations. All other configurations are analogous and are omitted due to space limitations.

Table 1. The parameters used in the experimental analysis.

N	μ	n	N_α/N	$K/\langle E \rangle$	γ	β
10000 or 100000	10 or 100	5			2.5	10

N is the network size; μ is the network average degree; n is the number of blocks; N_α/N is the relative size of each block; $K/\langle E \rangle$ is the normalized mixing matrix; γ is the power-law exponent; β is the inverse temperature of the model.

We consider all 16 combinations of the above values.

We compare the obtained graphs focusing in particular on three aspects: (i) their ability to reproduce the desired block mixing patterns; (ii) their ability to reproduce the desired degree distribution; (iii) the clustering (i.e., transitivity) of the resulting graph. All reported values are averaged over 20 graph instances per model. In Table 2 we summarize our findings. While all three models guarantee the desired degree distribution, graphs produced with the FCBM show low clustering, and graphs produced with the \mathbb{S}^1 model fail to reproduce the imposed block-mixing patterns. Our HGBM is the only model that provides all three properties at once.

Table 2. Summary of the simulation-based analysis.

	<i>HGBM</i>	<i>FCBM</i>	\mathbb{S}^1
$\langle \epsilon_{IJ} \rangle$	[-0.044, 0.006]	[-0.025, 0.028]	[-0.664, 5.667]
α	[2.56, 2.61]	[2.38, 2.61]	[2.54, 2.57]
$\langle c_{loc} \rangle$	[0.659, 0.682]	[0.066, 0.073]	[0.722, 0.744]

$\langle \epsilon_{IJ} \rangle$ is the average relative error of entry IJ of the block mixing matrix, as defined in (9); α is the exponent of the best possible powerlaw fit of the degree distribution; $\langle c_{loc} \rangle$ is the average local clustering coefficient.

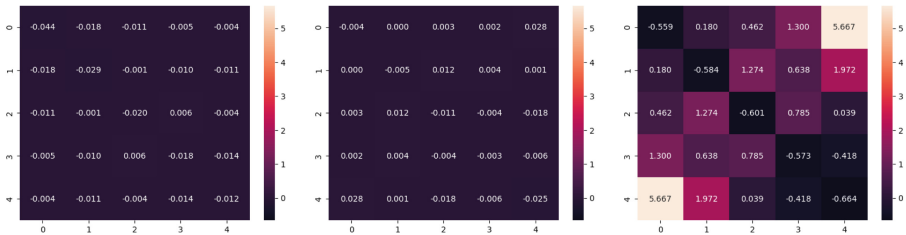
For each quantity, the reported values are the minimum and maximum value observed in the simulations.

3.1 Block Mixing

In Fig. 1, we show the relative error of the block-mixing matrix obtained in the experiments with respect to the expected one. The relative error for entry IJ is defined as the difference between K_{IJ} and the average of L_{IJ} over the 20 simulated network, divided by K_{IJ} :

$$\epsilon_{IJ} = \frac{K_{IJ} - \sum_{t=1}^{20} L_{IJ}^{(t)}/20}{K_{IJ}} \tag{9}$$

While the errors in the simulated block-mixing matrix are always less than 4% for both *HGBM* and *FCBM*, the \mathbb{S}^1 model fails to reconstruct the desired block-mixing matrix. This is unsurprising, as the block-matrix is not imposed in the \mathbb{S}^1 model in any way.

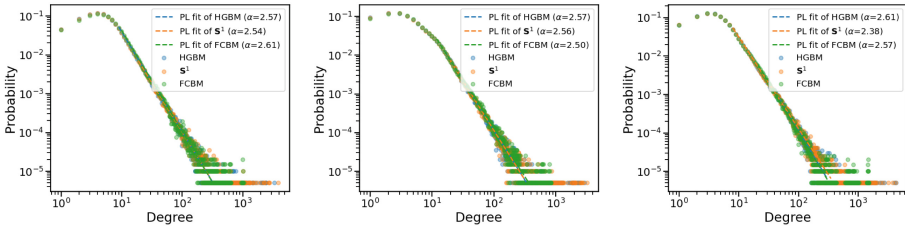


(a) Relative error matrix for the HGBM. (b) Relative error matrix for the FCBM. (c) Relative error matrix for the \mathbb{S}^1 model.

Fig. 1. Comparison of the relative error matrix, as defined in (9), for networks with $N = 10K$, $\mu = 10$, balanced groups and balanced mixing. The results show that the HGBM preserves the imposed mixing, as the FCBM, while the \mathbb{S}^1 model does not.

3.2 Degree Distribution

In Fig. 2, we show the degree distribution of the simulated graphs, together with a power-law fit obtained using the POWERLAW Python package [23]. Regardless of how balanced are the blocks and their mixing patterns, the HGBM preserves the imposed degree sequence comparably to the FCBM and \mathbb{S}^1 model: all models succeed at creating networks with the desired power-law degree distribution.



(a) Balanced blocks and balanced mixing matrix. (b) Balanced blocks and unbalanced mixing matrix. (c) Unbalanced blocks and unbalanced mixing matrix.

Fig. 2. Comparison of the degree distribution for networks with $N = 10K$, $\mu = 10$, and with different combinations of blocks and mixing matrices. The plot includes the exponent of the best power-law fit – whereas the hidden degrees were drawn from a power-law with exponent $\gamma = 2.5$. The results show that the HGBM preserves the imposed degree sequence comparably to the FCBM and \mathbb{S}^1 model.

3.3 Local Clustering Distribution

In Fig. 3, we show the distribution of the local clustering coefficient for the simulated graphs. Regardless of how balanced are the blocks and their mixing patterns, the HGBM guarantees high clustering, comparable to the one obtained with the \mathbb{S}^1 model, and significantly larger than the one obtained with the FCBM. The FCBM, in facts, lacks any element of vertex homophily, which is key to obtain high transitivity. Let us underline that the fluctuations in Fig. 3 are due to the plots showing the entire distribution of the local clustering over the 20 simulated graphs. Even in the HGBM and the \mathbb{S}^1 model, in fact, a few peripheral vertices may exist having near-zero clustering. Yet, as shown in Table 2, the average local clustering in these networks is one order of magnitude larger than in the FCBM.

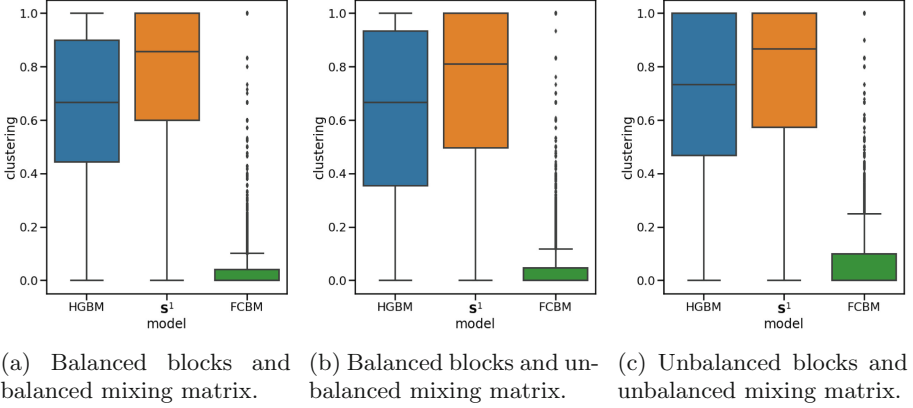


Fig. 3. Comparison of the distribution of the local clustering coefficient for networks with $N = 10K$, $\mu = 10$, and with different combinations of blocks and mixing matrices. The results show that the HGBM generates networks with high clustering, comparable to the \mathbb{S}^1 model and much higher than the one obtained with the FCBM.

4 Conclusions

In this paper, we introduced the Hidden-degree Geometric Block Model (HGBM), an extension of the \mathbb{S}^1 model that effectively incorporates group mixing. The HGBM builds on the intuition that any partition of the vertex set into blocks induces a partition of the graph into a set of mono- and bipartite subgraphs, corresponding to intra- and inter- block edges, respectively. These subgraphs can be individually modeled as \mathbb{S}^1 networks, provided that all latent features are extracted once and for all, so that the centrality and homophily patterns are preserved across different subgraphs. We showed that this requires replacing the hidden degree used in the \mathbb{S}^1 model with a hidden fitness, thus leading to a formulation that resembles that of the recent Fitness-Corrected Block-Model [21]. In a sense, then, our HGBM can be interpreted as a way to introduce a latent geometry into the framework of maximum-entropy degree-corrected block models.

To verify that the HGBM serves the purpose it was designed for, we performed a comparative analysis between our model, the \mathbb{S}^1 model and the FCBM. By conducting a series of simulations, we showed that the HGBM is the only one of these three models that generates synthetic networks that concurrently exhibit three main features often found in real-world networks: a specific group mixing, a heavy-tailed degree distribution, and a high transitivity.

We believe that our work paves the way to the integration of hyperbolic latent spaces into data-driven network models, offering the potential to generate more realistic social networks while preserving data-driven and empirical features, such as age-based and distance-based mixing. From a more theoretical point of view, we envisage at least two relevant directions for future work. First, we aim

to study latent geometric models in higher dimensions, to understand whether a more elegant formulation exists that encodes the block-mixing patterns into geometric properties of the network. Second, we aim to investigate whether our or other existing models reproduce additional properties of real-world social networks, such as degree assortativity – and, if not, how the models can be modified to achieve that goal.

A Python implementation of the HGBM, as well as all software used for our experimental analysis, is freely available as open-source under the GPLv3.

References

1. Keeling, M.: The implications of network structure for epidemic dynamics. *Theor. Popul. Biol.* **67**(1), 1–8 (2005)
2. Ribeiro, H.V., Sunahara, A.S., Sutton, J., Perc, M., Hanley, Q.S.: City size and the spreading of COVID-19 in Brazil. *PLoS ONE* **15**(9), e0239699 (2020)
3. Colizza, V., Barrat, A., Barthélemy, M., Vespignani, A.: The role of the airline transportation network in the prediction and predictability of global epidemics. *Proc. Natl. Acad. Sci.* **103**(7), 2015–2020 (2006)
4. Newman, M.: *Networks*. Oxford University Press, Oxford (2018)
5. Cimini, G., Squartini, T., Saracco, F., Garlaschelli, D., Gabrielli, A., Caldarelli, G.: The statistical physics of real-world networks. *Nat. Rev. Phys.* **1**(1), 58–71 (2019)
6. Serrano, M.A., Krioukov, D., Boguná, M.: Self-similarity of complex networks and hidden metric spaces. *Phys. Rev. Lett.* **100**(7), 078701 (2008)
7. Squartini, T., Garlaschelli, D.: Analytical maximum-likelihood method to detect patterns in real networks. *New J. Phys.* **13**(8), 083001 (2011)
8. Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., Boguná, M.: Hyperbolic geometry of complex networks. *Phys. Rev. E* **82**(3), 036106 (2010)
9. Krioukov, D.: Clustering implies geometry in networks. *Phys. Rev. Lett.* **116**, 208302 (2016)
10. Serrano, M.A., Boguná, M.: The shortest path to network geometry: a practical guide to basic models and applications, elements in structure and dynamics of complex networks. *Camb. Univ. Press Camb. Engl.* **10**, 9781108865791 (2022)
11. Boguna, M., Bonamassa, I., De Domenico, M., Havlin, S., Krioukov, D., Serrano, M.: Network geometry. *Nat. Rev. Phys.* **3**(2), 114–135 (2021)
12. Désy, B., Desrosiers, P., Allard, A.: Dimension matters when modeling network communities in hyperbolic spaces. *PNAS Nexus* **2**(5), pgad136 (2023)
13. Kertész, J., Török, J., Murase, Y., Jo, H.-H., Kaski, K.: Modeling the complex network of social interactions. In: Rudas, T., Péli, G. (eds.) *Pathways Between Social Science and Computational Social Science*. CSS, pp. 3–19. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-54936-7_1
14. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: homophily in social networks. *Annu. Rev. Sociol.* **27**(1), 415–444 (2001)
15. Karrer, B., Newman, M.E.J.: Stochastic blockmodels and community structure in networks. *Phys. Rev. E* **83**, 016107 (2011)
16. Peixoto, T.P.: Hierarchical block structures and high-resolution model selection in large networks. *Phys. Rev. X* **4**, 011047 (2014)
17. Fronczak, P., Fronczak, A., Bujok, M.: Exponential random graph models for networks with community structure. *Phys. Rev. E* **88**, 032810 (2013)

18. Park, J., Newman, M.E.J.: Statistical mechanics of networks. *Phys. Rev. E* **70**(6), 066117 (2004)
19. Robins, G., Snijders, T., Wang, P., Handcock, M., Pattison, P.: Recent developments in exponential random graph (p^*) models for social networks. *Soc. Netw.* **29**(2), 192–215 (2007)
20. Garlaschelli, D., Loffredo, M.I.: Fitness-dependent topological properties of the world trade web. *Phys. Rev. Lett.* **93**(18), 188701 (2004)
21. Bernaschi, M., Celestini, A., Guarino, S., Mastrostefano, E., Saracco, F.: The fitness-corrected block model, or how to create maximum-entropy data-driven spatial social networks. *Sci. Rep.* **12**(1), 18206 (2022)
22. Kitsak, M., Papadopoulos, F., Krioukov, D.: Latent geometry of bipartite networks. *Phys. Rev. E* **95**, 032309 (2017)
23. Alstott, J., Bullmore, E., Plenz, D.: PowerLaw: a python package for analysis of heavy-tailed distributions. *PLoS ONE* **9**(1), e85777 (2014)

Networks in Finance and Economics



Interactions Within Complex Economic System

Daniela Cialfi^{1,2}(✉)

¹ Institute for Complex Systems (Council of National Research of Italy) UoS La Sapienza University, 00185 Rome, Italy

² Enrico Fermi Research Center, 00184 Rome, Italy
daniela.cialfi@uniroma1.it, danielacialfi@gmail.com

Abstract. The emergence of a complex economic system is an interesting issue that has been addressed by many economists. This paper suggests that the processes that develop network formation within economic agents could be assimilated with the same procedures used by neurons in the human brain. Furthermore, the present paper presents a heuristic proof that suggests that the previous assumption is possible since the complex economic system, as a biological one, is a self-organization that has the same properties as any ergodic random dynamical chaotically system. In particular, it has been found that both systems possess a Markov Blanket or a Markov Decision Process, economically speaking. Furthermore, the demonstration in the present paper is restricted to how coupled dynamical systems organize themselves over time. In conclusion, the present work focuses on a simple but key aspect of complex economic system self-organization, providing a behavior metaphor in a different time-scale.

Keywords: Network formation · complex economic system · coupled random system · bifurcation · Lorenz system

1 Introduction

Physics of non-equilibrium systems or systems far from equilibrium represents one of the most interesting and ongoing theoretical research question for physicist, biologists and economists. For this reason, nowadays, economic science is making extensive use of mechanical models based on statistical inference, and in particular network theory and its assumptions, as the key tool able to describe the relationship and the interaction among individuals within different components of the economy and the following consequences of their interaction on the system. In conclusion, the aim of the present research is to focus on if the network formation process between economic agents could follow the same steps as neurons do in the brain. This will show that network models can introduce complex phenomena in economic systems by allowing for the endogenous evolution of networks. Thus, to shed a light on this aspect, the present paper is structured as follows. Section 2 and Sect. 3 presents in details how complex economic system

In case of contact please send your email to danielacialfi@gmail.com

possesses features typical of physical systems, the Lorenz systems specifically, which represent the basis of random dynamical chaotic systems. Section 4 then provides further remarks and discusses possible future challenges.

2 Economic System as Random Dynamic Chaotic One

From what affirmed in the Introduction, the interaction within the system assumption could be summarized in the following question: *is a complex economic system similar to a biological one?* A possible answer, according to [19], is throughout the reformulation of the second law of thermodynamics.¹

So, what is the link between thermodynamics, complex systems, and evolution and, consequently, the common characteristics of these systems? According to previous works, such as [18] and [1], both biological and economic systems could never be considered near-classic thermodynamic equilibrium systems due to the existence of the self-organization feature. In particular, this feature shapes the structure of the system through the use of free energy which can resist the thermodynamic gradient and the structural development. Consequently, what emerges are complex structures through the creation of endogenous feedback processes to solve their energy problems² affirming that they constitute channels enabling transitions from one meta-stable state to another meta-stable higher entropy state (see [11] and [23]).³ Consequently, adding energy cost to the objective function leads to the emergence of computational properties like stochastic and heterogeneity of the representative agent of the complex systems, such as the economic agent and neurons. Therefore, having shed light on how both systems are equipped with a dissipate structures feature, the next section will show that, by possessing this property, the complex economic system goes through a stochastic pathway, which represents one of the key attributes of a random complex system, such as a Lorenz-system.

As stated in the previous part of the section, both complex economic and biological systems have dissipative structures which provides a powerful account

¹ In particular, this reformulation is used to conceptualize the relationship between evolution, complexity and ecosystem (for more details see [21]). Moreover, this principle offers the possibility of formalizing economic evolution in terms of structural complexity development or, in other words, the available environmental energy transformation into the adverse degradation gradients.

² With this last assumption, complex systems, such as the biological and economic ones, could be considered a sub-class of dissipate structures, since their formation is statistically favored by the generalization of the second law of thermodynamics. So, this happens because these structures do not enable the dissipation of accessible reservoirs of free energy. Furthermore, they facility, at the same time, the irreversible relaxation of the associated disequilibrium.

³ It is important to stress that this line of thinking is based on the two works of [14] and [15], who tried to link natural selection to a physical principle of maximum energy transformation. Thus, to minimize the created dissipate heat during the extraction work process, the system must develop an efficient and predictive representation of the driving environment dynamics.

of how highly ordered non-equilibrium systems could emerge from essential thermodynamic principles or, in other words, from fluctuations present in stochastic thermodynamic chaotic processes. Having said that, this section will be divided into two parts: the first part provides mathematical proof of how these two systems possess stochastic chaotic processes, considered one of the key features of random dynamic chaotic systems. as in Lorenz-systems. Continuing this line of research, the second part will demonstrate in more detail how a complex economic system could be assimilated with a Lorenz-system through the existence of specific conditions and parameterization of the economic system itself—the bifurcation phenomenon that arises when heterogeneous economic agents with rational and behavioral expectations interact.

2.1 Complex Economic Systems as Lorenz-System

This sub-section will describe how complex economic and biological systems possess a stochastic chaotic process using Lorenz-systems [13], considered the foundation of random complex dynamical systems able to exhibit stochastic chaotic processes. This happens because Lorenz-systems posses pullback attractors which, according to [5]), are themselves random variables or, in other words, exhibit a probability density over the states referred to in this research context as non-equilibrium steady-state densities (or conservative components of the flow which, according to [12]), underwrite stochastic chaos). Consequently, what follows is the mathematical proof of the former statement. In particular, this proof will show the application of the Helmholtz decomposition as a solution of the Fokker Plank equation, which represents the density dynamics of any random complex system. Here what is important to stress out is the use of the Helmholtz decomposition enable us to, according to [22] and [2], to represent the expected flow information as the product of a flow operator and the gradient of a scalar field which corresponds on one side to the self-information (information theory) and on the other to the potential fluctuation (stochastic dynamics theory).

To achieve the previous, it is necessary to express the flow of a Lorenz-systems by the generalization of the Helmholtz decomposition into dissipative and conservative components, as follows:

$$\begin{aligned}
 \dot{p}(x) = 0 &\iff & (1) \\
 f(x) = \mathcal{Q}\nabla\mathfrak{S}(x) - \Gamma\nabla\mathfrak{S}(x) - \Lambda(x) &= \Omega\nabla\mathfrak{S}(x) - \Lambda(x) \\
 \mathfrak{S}(x) &= -\ln p(x) \\
 \Omega &= \mathcal{Q} - \Gamma(x) \\
 \mathcal{Q}(x) &= -\mathcal{Q}(x)^T \implies \nabla \cdot \mathcal{Q}\nabla\mathfrak{S}(x) \\
 \Lambda(x)_i &= \sum_j \frac{\partial \Omega_{ij}}{\partial x_j}
 \end{aligned}$$

where $\mathfrak{S}(x) = -\ln p(x)$ represents the self-information of any given state or potential function, $\Delta(x)$ is the correction term or, in other words, the house-keeping term able to activate changes within the flow operator $\Omega(x)$ over the state-space (i.e., changing the amplitude of random fluctuations). Furthermore,

the next step requires solving the density dynamical part of the problem in terms of time-dependent surprisal through the application of the reformulated Fokker-Planck equation derived in terms of time-dependent potential, $\mathfrak{S}(x) = -\ln p_\tau$ (Eq. 2).

$$\begin{aligned} \dot{p}_\tau &= \nabla \cdot \Gamma \nabla p_\tau - p_\tau \cdot f(x) - f(x) \cdot \nabla p_\tau & (2) \\ \dot{\mathfrak{S}}_\tau &= (\nabla - \nabla \mathfrak{S}_\tau) \cdot \Gamma \nabla \mathfrak{S}_\tau + \nabla \cdot f(x) - f(x) \cdot \nabla \mathfrak{S}_\tau \\ \dot{p}_\tau &= -p_\tau \dot{\mathfrak{S}}_\tau, \nabla p_\tau = -p_\tau \nabla \mathfrak{S}_\tau, \nabla^2 p_\tau = \\ & \quad -p_\tau \nabla^2 \mathfrak{S}_\tau + p_\tau \nabla \mathfrak{S}_\tau \nabla \mathfrak{S}_\tau \end{aligned}$$

but, being the flow at each point in the state-space time-invariant, it is possible to express the above time-dependent surprisal in terms of the steady-state potential of any Laplacian system as follows:

$$\begin{aligned} f(x) = (x, q, h) = (\mathcal{Q} - \Gamma) \nabla \mathfrak{S} - \Lambda \implies & (3) \\ \dot{\mathfrak{S}}_\tau &= (\nabla - \nabla \mathfrak{S}_\tau) \cdot \Gamma - \nabla (\mathfrak{S}_\tau - \mathfrak{S}) - \\ & \quad + \nabla \mathfrak{S}_\tau \cdot \mathcal{Q} \nabla \mathfrak{S} + \nabla (\mathfrak{S}_\tau - \mathfrak{S}) \cdot \Lambda \end{aligned}$$

Consequently, when $\mathfrak{S}_\tau - \mathfrak{S}$ is equal to no fluctuation changes in the surprisal, the density converges to steady-state, which exactly corresponds to the flow in a Lorenz-system where the non-equilibrium steady-state possesses a simple dependency structure.

2.2 Lorenz-Systems as Economic Bifurcation

This section shows that as nonlinear systems with a dimension larger than 1, the physical Lorenz-systems could be assimilated with the notion of economic bifurcation and, in particular, with strange attractors when in the presence of two-dimensional (2-D) systems.

Let us consider a 2-D discrete dynamical system:

$$(x_{t+1}, y_{t+1}) = F_\lambda(x_t, y_t) \tag{4}$$

where F_λ is a nonlinear 2-D differentiable map and λ its parameter. Furthermore, it is possible to define an orbit with an initial state (x_0, y_0) as follows:

$$\{(x_0, y_0), (x_1, y_1), (x_2, y_2) \dots\} = \{(x_0, y_0), F_\lambda(x_0, y_0), F_\lambda^2(x_0, y_0)\} \tag{5}$$

or in other words, a countable set in $x - y$ plane. Furthermore, [9] introduced the following simple 2-D quadratic map:

$$\begin{aligned} x_{t+1} &= 1 - ax_t^2 + y_t & (6) \\ y_{t+1} &= by_t \end{aligned}$$

where a and b are parameters. Thus, it is possible to further represent this map as the following differential equation:

$$(x_{t+1}, y_{t+1}) = H_{a,b}(x_t, y_t) \tag{7}$$

where $H_{a,b}$ is equal to $H_{a,b} = (1 - ax^2 + y, bx)$. In this system, an important characteristic is present: the attractor. It could be defined as a set of points $x_{t+1} = F(x_t)$ representing the long-term dynamical behavior of the system with the following characteristics:

1. Set A is invariant.
2. There exists an open neighborhood U of A such that all initial states $x_0 \in U$ converge to the attractor.
3. There exists an internal state $x_0 \in A$ for which the orbit is dense in A .

It is important to stress that this generic attractor definition is valid for a steady-state. Although, what is necessary here is what happens when we are in the presence of nonlinear systems. In this condition, the above type of attractor is called a *stranger attractor*, defined as follows:

Proposition 1. *An attractor A is called a stranger attractor of the N -dimensional dynamical system $x_{t+1} = F(x_t)$ if the map F has sensitive dependence with the set of the internal states that converge to A .*

From the previous definition, in each economic nonlinear system, according to [10], we have to consider the existence of *two-dimensional* cobweb model with rational versus naive expectations. Under mathematical point of view, let us consider the notion of the *stable manifold* and the *unstable manifold* of the steady-state defined as

$$W^S(S) = \{(x, m) \mid \lim_{x \rightarrow \infty} F_\beta^n(x, m) = S\} \tag{8}$$

$$W^u(S) = \{(x, m) \mid \lim_{x \rightarrow \infty} F_\beta^n(x, m) = S\}$$

Consequently, the stable manifold is the set of points which converge to the saddle point steady state and on the other hand the unstable one represents the set of points that move away from the steady state or in other words the set of points converging to the steady state backward in time. It is proper this geometrical explanation of the dynamical complexities of the evolutionary switching dynamics, that drives a complex economic system toward the steady-state by a *far from equilibrium* in order to stabilize force when most of the economic agents switch to rational expectations (Fig. 1A and Fig. 1B). This section has shown that a Lorenz-system and a complex economic system are equal to Lorenz-systems with the Helmholtz decomposition, polynomial expansions, and the complex economy system via the substantial equivalence between the Lorenz-systems and the bifurcation principle, since they both possess stochastic chaotic processes throughout.

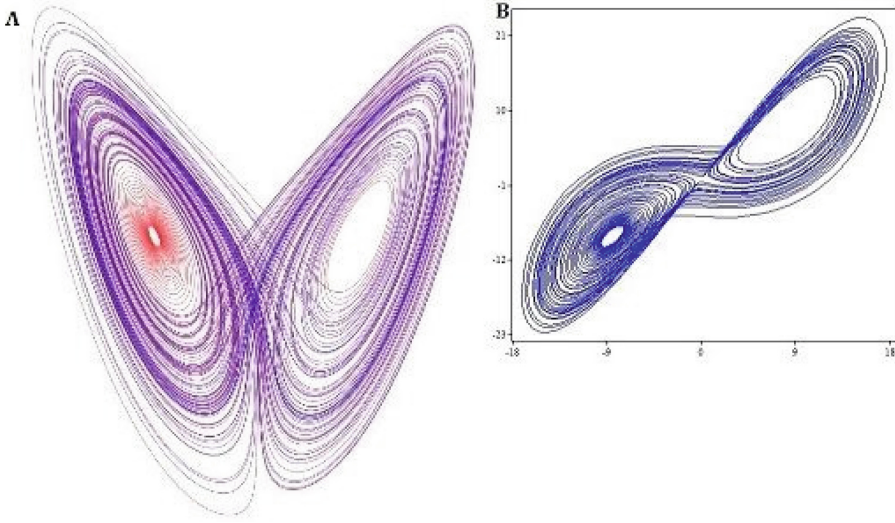


Fig. 1. Lorenz-systems and economic bifurcation. **(A)** Lorenz-systems in physics. **(B)** Lorenz-systems in economics elaborated with E & F Chaos software.

3 Economic Network Formation Similar to Neuron Formation

This section demonstrates how a complex economic system could be assimilated with a brain. Furthermore, it discusses the possible existence of common characteristics, which would affirm that the network formation between economic agents would be substantially equal to what happens in neurons’ network development. Let us start with the first step of the analysis.

As stated in Sect. 2, a complex economic system could be assimilated with a Lorenz-system, which possesses a conditional independence structure able to identify independent internal and external states, which themselves are conditioned upon blanket states. Thus, this assumption leads to accepting the existence of a particular partitioned state which, according to [3] and [20], interprets the generalized synchrony as a conditional expectation of the internal states. In more depth, this last assumption means that it could be parameterized as probability or, in other words, as Bayesian beliefs about external states. Furthermore, as will be seen in this section, this principle could be formalized as a notion of the variational free energy functional [6] since it can be used to separate states of a partition (i.e. internal and active states) from the remaining (i.e. external states). In more detail, it is necessary to use a new version of the Helmholtz decomposition seen in Sect. 3. This implies that the Jacobian matrix could be expressed in terms of the Hessian:

$$\begin{aligned}
 f(x) &= \Omega \nabla \mathfrak{S} - A & (9) \\
 J &= \Omega H + \nabla \mathfrak{S} \cdot \nabla \mathfrak{S} \nabla A
 \end{aligned}$$

$$J_{uv} = \frac{\partial f_u}{\partial x_v} = \sum_i \Omega_{ui} H_{iv} + \sum_i \frac{\partial \Omega_{ui}}{\partial x_v} \frac{\partial \mathfrak{S}}{\partial x_i} - \sum_i \frac{\partial^2 \Omega_{ui}}{\partial x_i \partial x_v}$$

The previous equation is used to introduce and motivate the existence of *sparse coupling conjecture*, defined as an absence of coupling between two states. Mathematically speaking, this means that the Jacobian coupling between states u and v is rare (Eq. 10)

$$J_{uv} = \frac{\partial f_u}{\partial x_v} = \sum_i \Omega_{ui} H_{iv} + \sum_i \frac{\partial \Omega_{ui}}{\partial x_v} \frac{\partial \mathfrak{S}}{\partial x_i} - \sum_i \frac{\partial^2 \Omega_{ui}}{\partial x_i \partial x_v} \tag{10}$$

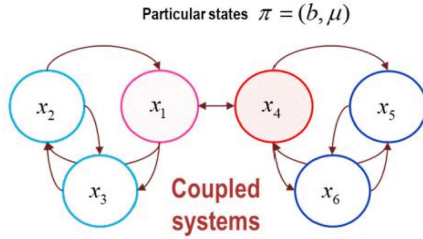


Fig. 2. Source [7]. Markov Blanket. Partition applied to six Lorenz-systems

So, this means that only two states are conditionally independent *if one state does not influence the other*. From this, it follows that this conjecture is not based on a Gaussian hypothesis for non-equilibrium steady-state density. Thus, being independent means that it is possible to build a partition via the following three rules (Fig. 2)

- 1 The Markov boundary $a \subset X$ is a set of internal states $\mu \subset x$ that represents the minimal set of states such that exists a non-zero Hessian submatrix.
 - a Internal states are independent of the remaining states called *active states*, and a combination of active and internal states is referred to as *autonomous states*
- 2 The Markov boundary $s \subset X$ is a set of internal states is a set of autonomous states and the minimal set of states such that exists a non-zero Hessian submatrix.
 - a Autonomous states are independent of the remaining states called *sensory states*, and a combination of active and sensor states originates the *blanket state* $b = \{a, b\}$
- 3 The remaining states constitute the *external states*.

In other words, a Markov Decision Process could be considered a form of the generative model in a discrete state-space, since it was introduced into the Bayesian network by [17]). Consequently, the existence of a Markov Blanket implies any state is not coupled with another (see Fig. 3) Before moving on to the second part of the section, it could be useful to summarize what has been discovered so far with the following proposition.

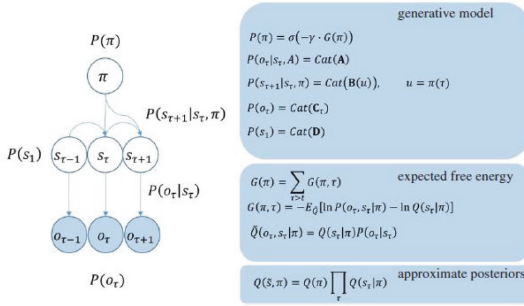


Fig. 3. Source [16]

Proposition 2. Any ergodic random dynamical system that possesses a Markov Blanket will appear to actively maintain its structural and dynamical integrity, which implies that:

- 1 A complex economic system is ergodic in the sense that the average of any measure of their states converges over a sufficient time.
- 2 A complex economic system is equipped with a Markov Blanket. This implies the existence of a partition of states into external and internal ones.

So, the last assumption affirms that a complex economic system could be optimized by Bayesian Inference. As a consequence, a complex economic system could be assimilated with the brain, since the latter is considered a statistical organ or, in other words, an agent able to infer the causes of its sensorium by its sensory and internal models provided by the continuous changes of the external world.

Consequently, the key elements for developing an economic network follow the same procedures as neurons do. Let us start the discussion on neurons with the use of predictive coding. This schema can compare the conditioned expectations with top-down predictors to elaborate the prediction error itself. Furthermore, the same prediction error is forwarded to the below level that encodes the conditional expectations (Fig. 4). So, the neural architecture optimizes the conditional expectations of causes in its hierarchical model of sensory input. This structure allows the link levels to be forwarded with reciprocal backward connections. Consequently, as it has been shown in Sect. 3, it is possible to find the corresponding network formation steps within a complex economic system: how knowledge, as in neurons, is diffused and created within the network. According to [4], it is necessary to assume that *i* the knowledge diffusion is useful only when the economic agent itself broadcasts its knowledge to whomever is directly connected, as happens between neurons of the same hierarchical level, and (ii) the creation of knowledge is present only when an economic agent receives new knowledge. Let us take $v_{i,k}^t$ in which the agent's *i* knowledge related to $k \in \{1, \dots, K\}$ at time *t*, $j \in \Gamma(i)$ is the agent who receives the agent's *i* knowledge. Thus, *i* and

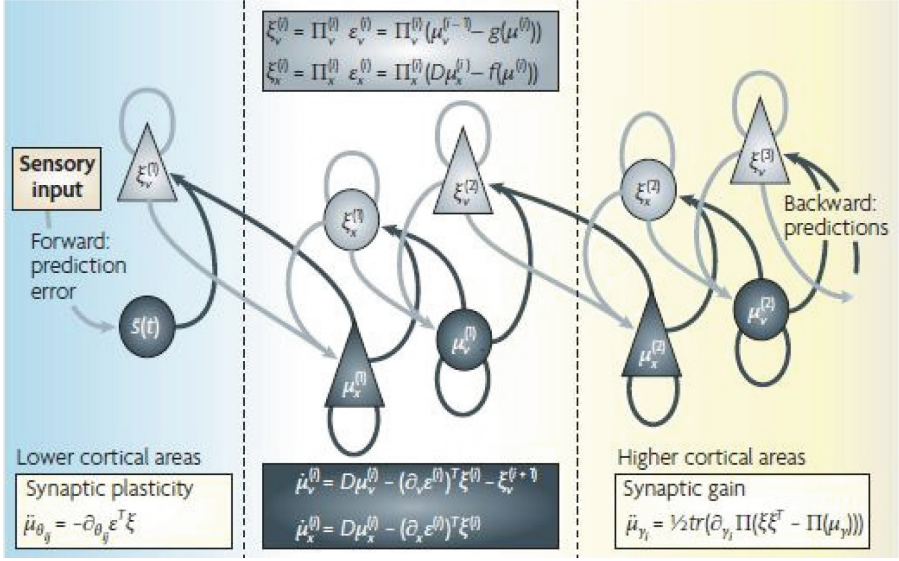


Fig. 4. Source [8]

j are connected by a non-directed graph such that their dissimilarity is not low enough.

$$\Delta(i, j) < 0 \in (0, \infty) \quad (11)$$

and in this context, the knowledge category can increase according to

$$v_{j,k}^{t+1} = v_{i,k}^t + \alpha \cdot \max\{0, v_{i,k}^t - v_{j,k}^t\} \quad (12)$$

where α represents how much knowledge will be transferred or diffused. At this point, we are interested in what happens when N economic agents act in an undirected graph. To discover their behavior [4] employed a heuristic approach. As happens where we had shed a light on how message passes between neurons during the network formation, also here in a economic context, it is necessary to create a lattice and subsequently assigning a probability p in order to re-write each edge of the graph, which represents a single economic agent or neuron. Consequently, this network has the following structure:

$$G(I, n, p) \quad (13)$$

where I represents the lattice on which the network is built, n explains the number of vertices of the graph or in economic term the economic agent, and p is the degree of randomness $p \in [0, 1]$. So, with these assumptions, knowledge diffusion and creation phenomena have the following determinants:

- 1 Agent's i knowledge diffusion is $w_i^t = \frac{1}{K} \sum_K w_{i,k}^t$.
- 2 The average knowledge level in a complex economic system at time t is $w^t = \frac{1}{N} \sum_{i \in I} (w_i^t)$.

3 Variations in the knowledge level are equal to $\frac{1}{N} \sum_{i \in I} (w_i^t)^2 - (w^t)^2$.

4 The degree of the spatial local system to interconnect economic agents is

$$S = \frac{1}{\sigma^2} \sum_{i \in I} \sum_{i \neq j} w_i (w_i - w)(w_j - w), \text{ where } w_{ij} = \frac{\frac{1}{d(i,j)}}{\sum_{i \in I} \sum_{j \neq i} \frac{1}{d(i,j)}}.$$

In conclusion of this section, it is possible to affirm that knowledge diffuses via specific contacts between agents, as happens in neurons' hierarchical structure within the brain. This latter assumption affirms that the knowledge is easy to spread within a small word or in mathematical terms in a locally connected graph.

4 Conclusions

In conclusion, this paper has rehearsed, from a generic point of view, if steps used to develop a network within a complex economic system could be assimilated with the same procedures that neurons follow within their hierarchical structure in the human brain. In other words, this paper's aim is to shed a light on the substantial equivalence between, from a mathematical point of view, the message-passing mechanism in the brain and the knowledge creation and diffusion between economic agents during network formation within a complex economic system. This equivalence has been achieved by comparing and treating the complex economic system as a random dynamical chaotic system and, in particular, as a Lorenz-system. During this comparison, the existence of conditional independence at a non-equilibrium steady-state has been identified: the Lorenz-system properties could be assimilated with the notion of bifurcation in economics. Furthermore, this conditional independence has generated a particular partition of states where internal states are statistically secluded from external ones using a blanket state, the so-called Markov Blanket. Consequently, the complex economic system is subjected to a Markov Decision Process. It has been discovered that it is possible to establish an equivalence between the complex economic system and the human brain. In particular, this was accomplished through a comparison between the message-passing phenomenon and the knowledge creation and diffusion phenomena. Finally, it is possible to affirm that this kind of comparison could represent the first mathematical basis for understanding even more how some economic phenomena derive from autonomous or active sensors in biology.

References

1. Allen, P.: Evolving complexity in social science. In: Altman, G., Koch, W. (eds.) *Systems: New Perspectives for the Human Sciences*, pp. 3–8. Walter de Gruyter (1998). <https://doi.org/10.1515/9783110801194.3>
2. Ao, P.: Emerging of stochastic dynamical equalities and steady-state thermodynamics from Darwinian dynamics. *Commun. Theor. Phys.* **49**(5), 1073 (2008). <https://doi.org/10.1088/0253-6102/49/5/01>

3. Barreto, E., Josie, K., Morales, C., Sander, E., So, P.: The geometry of class synchronisation, Chap. 13, pp. 151–164 (2003). <https://doi.org/10.1063/1.1512927>
4. Cowan, R., Jordan, N.: Knowledge creation, knowledge diffusion and network structure. In: Kirman, A., Zimmermann, J.B. (eds.) *Economics with Heterogeneous Interacting Agents*. LNEMS, vol. 503, pp. 327–343. Springer, Heidelberg (2001). https://doi.org/10.1007/978-3-642-56472-7_20
5. Crauler, H., Flandoli, F.: Attractors for random dynamical systems. *Probab. Theory Relat. Fields* **23**(9), 365–393 (1994). <https://doi.org/10.1007/BF01193705>
6. Friston, K., Ao, P.: Free energy, value, and attractors. *Comput. Math. Methods Med.* 937860 (2012). <https://doi.org/10.1155/2012/937860>
7. Friston, K., Heins, C., Uelzhoffer, K., DaCosta, L., Parr, T.: Stochastic chaos and Markov Blanket. *Entropy* **23**, 12–20 (2021). <https://doi.org/10.3390/e23091220>
8. Friston, K.: Free energy principle: a unified brain theory? *Nat. Rev. Neurol.* 937860 (2010). <https://doi.org/10.1038/nrn2787>
9. Hénon, M.: A two-systems dimensional mapping with a strange attractors. *Commun. Math. Phys.* **50**, 69–77 (1976). <https://doi.org/10.1007/BF01608556>
10. Hommes, C.: *Behavioral Rationality and Heterogeneous Expectations in Complex Economic Systems*. Cambridge University Press, Cambridge (2015). 9781107019294
11. Jeffreys, K., Pollack, D., Ravelli, C.: On the statistical mechanisms of life: schrodinger revised. *Ent.* **21**(12), 1211 (2019). <https://doi.org/10.3390/e21121211>
12. Ma, Y., Tan, Q., Yuan, B., Ao, P.: Potential function in a continuous dissipative chaotic system: decomposition schema and role of stranger attractors. *Int. J. Bif. Ch.* **24**(2), 1450015 (2014). <https://doi.org/10.1142/S0218127414500151>
13. Lorenz, E.N.: Deterministic non-periodic flow. *J. Atmos. Sci.* **23**(9), 130–141 (1963). [doi.org/10.1175/1520-0469\(1963\)020<0130:DNF>2.0.CO;2](https://doi.org/10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2)
14. Lotka, A.: Natural selection on as physical principle. In: National Academy of Sciences (eds.) *Proceedings of National Academy of Science of the United States of America*, pp. 151–154. National Academy of Sciences (1922). <https://www.jstor.org/stable/84175>
15. Lotka, A.: Contribution to the energetics of evolution. In: National Academy of Sciences (eds.) *Proceedings of National Academy of Science of the United States of America*, pp. 147–151. National Academy of Sciences (1922). <https://doi.org/10.1073/pnas.8.6.147>
16. Parr, T., Friston, K.: Uncertainty, epistemics and active inference. *J. Roy. Soc. Interface* **14**, 20170376 (2017). <https://doi.org/10.1098/rsif.2017.0376>
17. Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Nature of Plausible Inference*. Morgan Kaufman, San Francisco (2013). 9781558604797
18. Prigogine, J.: Time, structure and fluctuations. *Science* **201**, 777–784 (1978). <https://doi.org/10.1126/science.201.4358.777>
19. Raine, A., Foster, J., Potts, J.: The new entropy law and the economic process. *Ecol. Complexity* **3**, 354–360 (2006). <https://doi.org/10.1016/j.ecocom.2007.02.009>
20. Rulkov, N.F., Sushchik, M.M., Tsimring, L.S., Abarbanel, H.D.I.: Generalized synchronization of chaos in directionally coupled chaotic systems. *Phy. Rev. E* **51**(2), 980 (1995). <https://doi.org/10.1103/PhysRevE.51.980>
21. Schneider, E., Kay, J.: Life as a manifestation of the second law of thermodynamics. *Math. Comput. Modoll.* **9**, 25–48 (1994). [https://doi.org/10.1016/0895-7177\(94\)90188-0](https://doi.org/10.1016/0895-7177(94)90188-0)

22. Shi, J., Chan, T., Yuan, R., Yuan, B., Ao, P.: Relation of a new interpretation of stochastic differential equations to Ito process. *J. Stat. Phys.* **148**(3), 579 (2012). <https://arxiv.org/pdf/1111.2987.pdf>
23. Uelzhoffer, K., Da Costa, L., Cialfi, D., Friston, K.: A drive towards thermodynamic efficiency for dissipative structures in chemical reaction networks. *Entropy* **20**(2), 1115 (2021). <https://doi.org/10.3390/e23091115>



Demand Shocks and Export Surges in Trade Networks

John Schoeneman¹(✉), Marten Brienen¹, Lixia Lambert², Dayton Lambert²,
and Violet Rebek¹

¹ Department of Global Studies, Oklahoma State University,
Stillwater, OK 74078, USA
john.schoeneman@okstate.edu

² Department of Agricultural Economics, Oklahoma State University,
Stillwater, OK 74078, USA

Abstract. Using network analysis, this paper analyzes export surges of commodities from U.S. states to examine their connectedness along industrial and inter-state ties, and how they propagate through the trade network along these ties. Our findings suggest that export surges tend to be highly skewed towards particular goods, that these surges spread according to higher-order structural dependencies in the supply chain, and that increases of export surges were defined by preferential attachment.

Keywords: supply chains · network analysis · shocks · trade

1 Introduction

The effects of the COVID-19 pandemic on global markets provide an opportunity to more closely examine how the composition and structure of trade networks either positively or negatively affected export rebound. During various stages of the pandemic, the latter caused several important shifts in global trade as well as in the terms of trade for many countries. These shifts resulted in shocks in demand for intermediate and final products, which were experienced unequally in different countries and regions [14]. During the early stages of the pandemic, prices for intermediate inputs rapidly increased as demand outpaced the ability of supply chains to adapt. The uncertainty regarding the scope and duration of the pandemic also compounded upward pressure on prices. This paper analyzes the export surges of commodities from U.S. states to examine how these surges propagate through trade networks along industrial and inter-state ties.

During the recovery phase, and as some supply chains found new routes to deliver goods, retail demand surged and was further fueled by stimulus policies enacted by some countries, while demand for services, particularly in the restaurant and travel sectors, experienced a sharp decline. Seemingly paradoxically, global trade volume reached record levels as firms adapted to the changing market conditions caused by the pandemic. We use network theory to explain

why exports for some goods surged in some states but not in others. We also expand on the literature on bipartite trade networks, which has tended towards product specificity. In contrast to that literature, we are the first to examine the trade network in this manner and for all goods. Our analysis finds that the export surges tend to be highly skewed towards particular goods, that these surges spread according to higher-order structural dependencies in the supply chain network, and that increase of export surges were defined by preferential attachment.

2 Theory

Global trade dynamics underwent a significant transition from 2022 and 2023 as supply chains rebounded following the COVID-19 pandemic (United Nations Conference on Trade and Development, 2023). World trade was already trending downwards prior to the pandemic as a result of tensions between major economies. However, the 2020 pandemic brought a decline in trade volumes even more profound than that experienced during the 2008 financial crisis. Many factors, including widespread economic slowdowns, border closures, travel restrictions, and supply chain disruptions, conspired to produce this precipitous decline. Amid these challenges, the year 2021 inspired some hope as trade volumes rebounded, soaring to a record USD\$28.5 trillion of world trade in goods, representing a 13% increase over pre-pandemic levels [19].

This resurgence, however, was characterized by uneven growth across countries and sectors. While trade in goods surged to unprecedented heights, recovery in the service sector was notably more sluggish. Global trade surged even further to reach an impressive \$32 trillion by the end of 2022 [19]. However, pent-up demand for travel, services, retail goods, along with record-high savings drove inflation to levels not experienced since the 1990s. We apply network analysis methodology to explain why exports surged in some states and goods but not in others. Previous literature has already established the usefulness of using network analysis to examine the impact of the COVID-19 pandemic on global trade, establishing that ties between states and industries lead to shared negative trade shocks being clustered [8].

The relationship between exports, business performance, and economic growth has long been a subject of interest for researchers in international trade and industrial economics at both the micro and macro scales. The importance of international trade to the economy has been very well established, as increased export volumes have been shown to be important drivers for economic growth [16]. Disruptions to international trade consequently pose a significant threat to economic stability, as shocks experienced by industries are likely the primary drivers of fluctuations in GDP, shaping overall economic performance [1]. Further underscoring the importance of international trade, Bernard and Jensen [5] showed that higher export volumes boost employment and wages in the U.S. in the short term. Disruption of international trade, on the other hand, undermines these benefits, as trade shocks can propagate and amplify, and ultimately cause what Schumpeter called the ‘creative destruction’ of an economy [18].

Carvalho and Tahbaz-Salehi [10] stressed that while production networks are often analyzed at the industry level, accounting for firm-level influences produces richer theoretical and empirical insights. Indeed, given the importance of international trade for economic growth and the threat posed by trade shocks traveling along industry networks, decision making at the firm level is an important factor in countering that threat. Firms enter markets to replace inefficient firms and capture industry profits, creating a complex interplay between input-output network effects and market dynamics[2]. Bernard and Jensen [4] found that established exporting firms respond strongly to export shocks, underscoring the role of sunk costs in shaping decision-making processes in exporting firms based on their sensitivity to market conditions. Berthou et al. [7] investigated the relationship between export and import expansion and firm productivity and found that export and import expansion increased average firm productivity. However, export expansion also shifted activity towards more productive and efficient firms. Import expansion operates in the opposite direction. In addition, the heterogeneous nature of firms across industrial sectors and locations should be emphasized, given its implications for evaluating trade gains [6].

These economic properties for firms often lead to preferential attachment in the establishment of new economic ties, where firms with more connections are more likely to form new connections as the network grows. Another study on firm-level exports in Columbia found that firms that had already been well-positioned in the network prior to the pandemic experienced greater increases in export volume [9]. Furthermore, these economic properties are cyclically reinforcing, as more efficient firms export more, while increased exports make firms more efficient. This should apply at the aggregated state-level as well, as past work has shown that the power-law distribution that results from preferential attachment applies to international trade networks at the product category level at the country-level [17]. *Therefore, we expect increased exports at the state-level to follow the same preferential and skewed attachment pattern as the formation of trade relationships at the country-level.*

The importance of industry ties and the networks they form has also been well established. Upstream and downstream factors often hold different levels of importance for firm productivity, according to Bernard et al.'s production network model [3]. In this model, larger, more efficient firms have more geographically dispersed ties. Indeed, it has been shown that the costs of firms that export to more countries appear to be less affected by geographical distance [11]. If this is true, then the presence of ties with one state are indicative of an advantage that makes ties across multiple states more likely. Given that exporting firms are generally the most efficient and the largest, commodity export surges will spread along the geographically dispersed ties of these large, efficient firms. *Given firm-level economic ties between states and industries, we expect export networks to be characterized by higher-order structural dependence. Furthermore, increases in upstream and downstream demand lead to increased demand throughout the supply chain and can have pronounced bull-whip patterns on increases; consequently, we expect the structural dependence of the network to increase over time.*

Finally, *given the expected combination of high levels of clustering and preferential attachment, we also expect to observe highly connected networks where distinctions between communities decrease as export surges increase.*

3 Data

Tintelnot et al. [13] stressed the importance of modeling domestic production networks to better comprehend the behavior of different types of firms engaged in international trade, but employing a single approach to elucidate the complete economic repercussions of trade shocks remains a significant challenge [15]. Because of this, we employ a multi-pronged approach to network analysis to build upon existing literature.

Monthly U.S. state-level commodity import and export data used were collected by the U.S. Census using the U.S. Customs' Automated Commercial System [20]. We collected data for 1,227 commodities for the period spanning from December 2018 to November 2021. This time frame allows us to investigate several important aspects of the export surges. First, it provides a baseline for disruptions prior to the outbreak of the COVID-19 pandemic. Second, it allows us to investigate longer trends as the shifts in demand resulting from the initial shock of the pandemic persisted. Import and export data are reported in total inflation-adjusted value, in 2020 U.S. Dollars. All 50 states and the District of Columbia are included in the analysis.

4 Research Design

For the analysis, we construct a bipartite network, which consists of two different types of nodes and where edges can only be shared between the two different types. In this application we use states as the first mode and exports at the four-digit level commodity code of the Harmonized System (HS-4) as the second mode. The edges in the bipartite graph are a measure of export disruption, comparing the export value of the current month to a three-month window centered on the same month of the previous year. When the current month's value was more than 25% of the maximum value in the window for the previous year, it was coded as a one for disruption. Beginning in 2021, however, we look at the window for two years prior so that disruptions were based on values preceding the COVID-19 pandemic.

Standard inferential models for bipartite networks, such as exponential random graph models (ERGM), stochastic actor-oriented models (SAOM), or conditional uniform graph (CUG) tests would ordinarily be preferred for testing hypotheses about structural dependence. However, given our network's size and highly unbalanced structure, model convergence was impossible or computationally prohibitive with the currently available methodology and software. Therefore, we use three methods to analyze the structure of the network and the manner in which ties spread.

The first method compared bipartite networks with simulated random bipartite networks of the same size and density. This follows the same intuition as CUG tests but is not as computationally demanding as CUG tests for large networks. We use this method to compare the number of four- and six-cycles in the network. Four-cycles are the bipartite equivalent of transitivity or clustering in monopartite networks. Four-cycles are defined as two states sharing ties with two different commodities and six-cycles are defined as three states sharing ties with three different commodities.

The second method is ordinary least squares (OLS) to test for preferential attachment both for states (activity) and HS-4 commodities (popularity). For these models, we calculate the number of ties per node for a given node and use OLS to find the relationship between the number of ties a node has and the number of new ties it forms in the next period. In order to take advantage of time and unit fixed effects, while preserving statistical power and the ability to examine change over time, we use a six-month moving window for the OLS models. For example, the sample for the first model is December 2018-May 2019, and the sample for the second model is January 2019-June 2019. The use of fixed effects here is important, because the economies of the states vary along several important variables such as a location, size, policy, and economic complexity, and thus some export a wider variety of commodities and have more opportunities to experience export surges than others. The reported models that included unit and time effects increased R-squared measures of the models from about ten percent to over fifty percent. Given that 25% is a somewhat arbitrary definition for disruption, we include a sensitivity check of the same process using a 50% minimum value threshold to define a trade surge.

The last method is community detection. In addition to our main network, we also examine collapsed monopartite graphs to analyze higher level community formation among states and commodities. In these graphs, shared trade surges result in weighted edges between states or two-digit commodities. For the bipartite network, we used the Leiden algorithm with a metric multidimensional scaling layout, which helps to visually separate states from the commodities while still showing clusters, unlike the standard bipartite layout with a fast-greedy algorithm. The fast-greedy algorithm worked best for the monopartite networks to capture dense clusters. It worked better than with the bipartite network, given fewer nodes. We used a special layout for these networks, which reduces overlap for a more meaningful visual representation of the communities. All community detection and plotting were done using the *igraph* package for R [12].

5 Results

In Fig. 1, graph *a* shows that the export surge network's density drops at the beginning of the pandemic, as demand for goods drops amid uncertainty due to lay-offs and lock-downs, but then rapidly increases far above pre-pandemic levels until it begins to plateau in mid-2021. We also see in graph *b* that the network

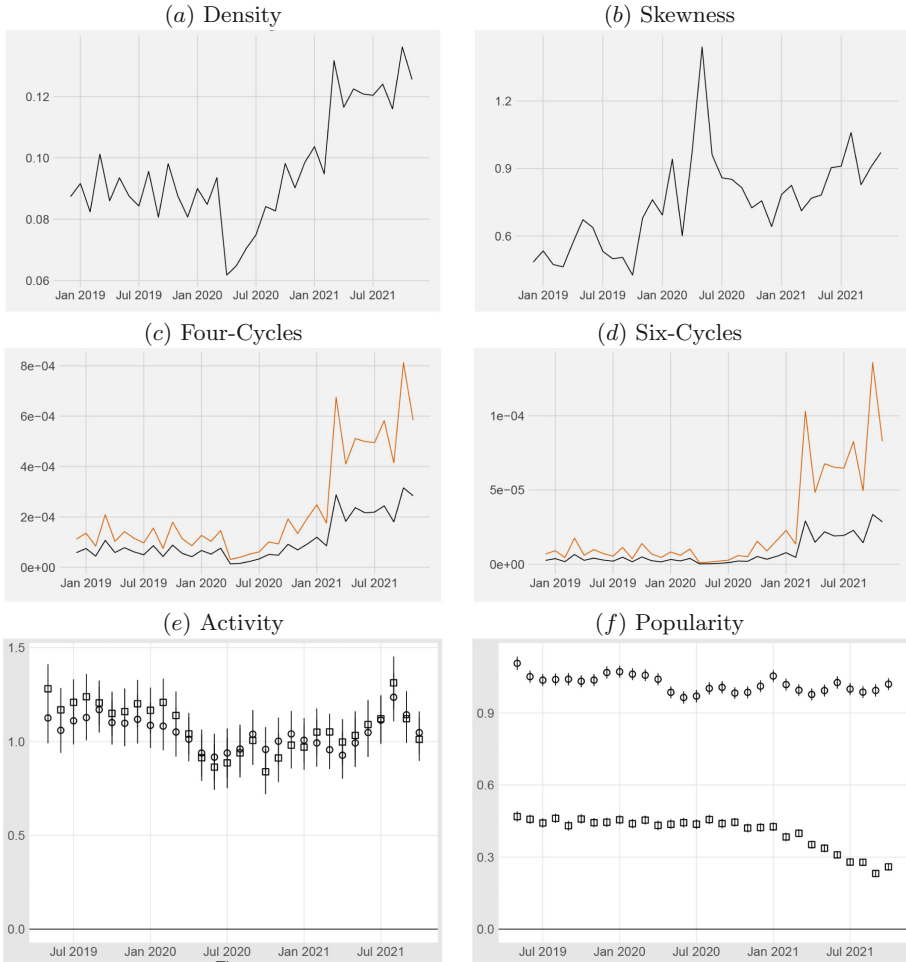


Fig. 1. Descriptive Statistics and Modified Statistical Tests. Line plots *a* and *b* are statistics of the observed bipartite networks. Line plots *c* and *d* are cycle statistics of the observed and simulated bipartite networks. The orange line is the observed network and the black line is the simulated network. Rope ladder plots *e* and *f* are OLS coefficients, with unit and time fixed effects. Circles are for models with 50% increases and squares are for models with 25% increases.

is always positively skewed, with certain commodities having far more export surges than the median. However, we see that this was especially pronounced at the beginning of the pandemic and that even after dropping, the skewness remained above pre-pandemic levels. This indicates a concentration of demand in certain certain industries that drove export surges and that these industries have continued to benefit from the pandemic-driven growth.

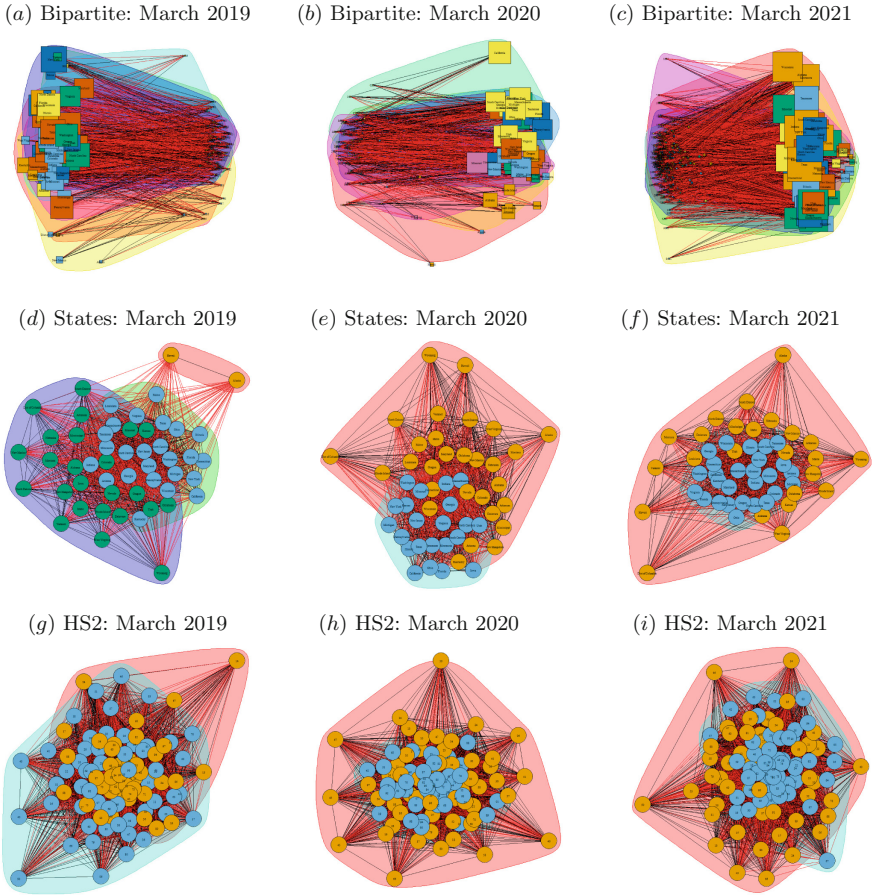


Fig. 2. Community Detection Plots. Plots *a:c* are the full networks with node determined by degree (squares are states, circles are commodities), *d:f* are collapsed state networks with fixed node size, and *g:i* are collapsed HS two-digit code networks with fixed node size.

Graphs *c* and *d* show that cycles are more prominent in the observed network than in a network of the same density with ties randomly assigned. This supports our hypothesis that industry and state ties form economic clusters through which network ties spread. While six cycles occur less frequently, they follow the same, albeit delayed, pattern as four cycles. This indicates that four-cycles expanded to six-cycles over time, which in turn means that the structural dependence was driving the formation of new ties.

Graphs *e* and *f* show that preferential attachment (i.e. the more ties a node has, the more likely it is to obtain yet more ties) is statistically significant for states and commodities. This effect is relatively consistent, apart from a small drop at the beginning of the pandemic. The effect is slightly more pronounced

for states, and coefficients for commodities have smaller confidence intervals, which can be explained by the fact that there are only 50+1 states while there are 1,227 four-digit commodities.

For the bipartite network (Fig. 2, graphs *a:c*), the most prominent observation over time is that the heterogeneity of degree distribution increases, with certain states continuing to experience far more export surges than other states, while the network itself becomes more connected and assigned communities of nodes exhibit more overlap. This indicates that certain states benefited more than others from the pandemic-driven export boom. However, all states and industries reaped some benefits. This pattern of increasing community overlap and increased median node centrality is also shown in graphs *d:i*. However, these increases occur more quickly and are more prominent in the commodity network than in the state network.

6 Conclusion

The COVID-19 pandemic laid bare insufficiencies in the global supply chains, resulting in shortages of specific commodities and economic dislocation. As the pandemic continued and governments worldwide formulated policy responses, the global economy returned to a new normal that was characterized by surges as a long period of social and economic disruption resulted in sudden and explosive shifts in demand for specific commodities, causing new waves of disruptions as industries struggled to cope with supply chains buckling under the pressure to meet shifts in demand. The ability to track the spread of supply chain disruptions occasioned by these surges in demand for specific goods will be an indispensable tool in efforts to restore stability and predictability to global supply chains. It is in this context that network analysis has become an invaluable tool, as it allows precisely for the modeling of networks and movement within them.

Our novel approach to network analysis allowed us to avoid collapsing the network into a monopartite configuration and retain all the information contained in the bipartite model. This in turn allowed us to track not only the spread of disruptions through industries, but also to determine more clearly which states and commodities were affected and to model additional structural dependencies through which disruptions spread. This deeper understanding of disruption spread increases the predictive value of the models for determining where surges will occur, and which goods will be affected. This should prove invaluable to decision-makers in the affected industries as well as to policy makers, as it gives them the tools to see which critical goods are likely to become the subject of export surges and to put measures in place to ensure the reliable supply of critical and strategic goods.

Notes and Comments. All data used are from publicly available sources. For replication code, please email the authors.

References

1. Atalay, E.: How important are sectoral shocks? *Am. Econ. J. Macroecon.* **9**(4), 254–280 (2017)
2. Baqaee, D.R., Farhi, E.: Macroeconomics with heterogeneous agents and input-output networks. Technical report, National Bureau of Economic Research (2018)
3. Bernard, A.B., Dhyne, E., Magerman, G., Manova, K., Moxnes, A.: The origins of firm heterogeneity: a production network approach. *J. Polit. Econ.* **130**(7), 1765–1804 (2022)
4. Bernard, A.B., Jensen, J.B.: Why some firms export. *Rev. Econ. Stat.* **86**(2), 561–569 (2004)
5. Bernard, A.B., Jensen, J.B., Lawrence, R.Z.: Exporters, jobs, and wages in us manufacturing: 1976–1987. *Brookings papers on economic activity. Microeconomics* **1995**, 67–119 (1995)
6. Bernard, A.B., Jensen, J.B., Redding, S.J., Schott, P.K.: Firms in international trade. *J. Econ. Perspect.* **21**(3), 105–130 (2007)
7. Berthou, A., Chung, J.J.H., Manova, K., Sandoz Dit Bragard, C.: Trade, productivity and (MIS) allocation. Available at SSRN 3502471 (2019)
8. Brienen, M., Lambert, L.H., Lambert, D.M., Schoeneman, J.: A social network analysis approach to estimate export disruption spread in the us during the COVID-19 pandemic: how policy response and industry ties relate. *J. Ind. Bus. Econ.* 1–19 (2023)
9. Campi, M., Dueñas, M.: Clusters and resilience during the COVID-19 crisis: evidence from Colombian exporting firms. Technical report, IDB Working Papers Series 1375 (2022)
10. Carvalho, V.M., Tahbaz-Salehi, A.: Production networks: a primer. *Annu. Rev. Econ.* **11**, 635–663 (2019)
11. Chaney, T.: The network structure of international trade. *Am. Econ. Rev.* **104**(11), 3600–3634 (2014)
12. Csardi, G., Nepusz, T., et al.: The Igraph software package for complex network research. *InterJ. Complex Syst.* **1695**(5), 1–9 (2006)
13. Dhyne, E., Kikkawa, A.K., Mogstad, M., Tintelnot, F.: Trade and domestic production networks. *Rev. Econ. Stud.* **88**(2), 643–668 (2021)
14. Di Giovanni, J., Kalemli-Özcan, e., Silva, A., Yildirim, M.A.: Global supply chain pressures, international trade, and inflation. Technical report, National Bureau of Economic Research (2022)
15. Gopinath, G., Neiman, B.: Trade adjustment and productivity in large crises. *Am. Econ. Rev.* **104**(3), 793–831 (2014)
16. Hausmann, R., Pritchett, L., Rodrik, D.: Growth accelerations. *J. Econ. Growth* **10**, 303–329 (2005)
17. Liu, L., Shen, M., Tan, C.: Scale free is not rare in international trade networks. *Sci. Rep.* **11**(1), 13359 (2021)
18. Schumpeter, J.A.: *Capitalism, Socialism and Democracy*. Routledge (2013)
19. UNCTAD: *Global Trade Update*. Division on International Trade and Commodities (2022)
20. U.S. Census Bureau: *U.S. Import and Export Merchandise trade statistics*. Economic Indicators Division USA Trade Online (2022)



Properties of B2B Invoice Graphs and Detection of Structures

Joannès Guichon^{1,2}(✉), Nazim Fatès^{1,2}, Sylvain Contassot-Vivier¹,
and Massimo Amato²

¹ Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France

joannes.guichon@inria.fr

² Bocconi University, Milan, Italy

<https://members.loria.fr/JGuichon/>

Abstract. In economy, a major issue is the potential lack of liquidity for settling the debts generated by payment delays among companies. Since this lack may trigger cascading failures, we analyse the interconnection of debts. Settling debts means lowering the systemic risks. We analyse the data of a large economic network from an Italian invoice operator on a one-year span. We compare different methods to detect structures or communities that could be helpful for debt netting algorithms. The structure of such networks is not currently well known. We give hints on how to sort and identify the type of B2B invoice graphs. In particular, we address the possibility to identify relevant communities in such networks.

Keywords: Graph analysis · community detection · B2B invoice networks

1 Introduction

The intricate nature of debt arises from financial obligations, among companies triggered by the payment system, fostering a web of interconnected relationships [11]. As companies generally pay their *invoices* with a delay, debts accumulate and intertwine. A potential chain reaction of defaults emerges, posing a significant threat to the stability of the entire financial system. Debt settlement emerges as a vital mechanism to uphold commitments, maintain credibility, and bolster trust in the corporate world.

To address the challenges posed by trade debts, the concept of *netting* may come into play [9]. Such treatment consists in reducing the global amount of debts between the companies by using mutual compensations, thus lowering the need for immediate liquidity. This liquidity saving mechanism could streamline the debt settlement process, minimize credit risks, and enhance the efficiency and resilience of the corporate ecosystem to ensure a smoother and more secure financial landscape. Different types of debt netting techniques exist, mainly focused on partial [12] or complete [2] settlement. Partial netting is based on the possibility to split invoices in order to partially settle the debt. On the opposite way,

complete netting is restricted to cancel entire invoices. Today, research on this type of netting is not well developed and this technique needs a finer knowledge about the graphs structure. Globally, all those kinds of processes mainly exist between banks [14, 15] and our aim is to extend them to B2B exchanges.

However, in order to implement such a method in algorithmic form, it would be useful to identify specific properties of invoice graphs between companies. These networks are quite specific as they correspond to weighted, directed, multi-edge and time-varying graphs. In Sect. 2, we determine the main characteristics of exchange networks. Then, in Sect. 3, we try to identify particular structures or communities. The aim is to split up our graphs in order to facilitate their processing by netting algorithms. This work should be considered a preparatory work before more research on netting algorithms. As a first step, we focus on three methods to understand the structure of these graphs.

2 B2B Invoice Graphs: Definition and Properties

This study is based on a set of 27,445,353 invoices emitted by companies of North Italy over the span of the 2019 year. This dataset was provided by Infocert, an Italian electronic invoices operator. Each invoice is composed of the following data:

- unique identifier to register the invoice;
- unique identifier of the debtor company;
- unique identifier of the creditor company;
- due debt in euros;
- date of the invoice emission.

It is worth noticing that this dataset is anonymized and obviously it is partial as it includes only the invoices given to the operator by subscribers of its service. However, although we have non-exhaustive information, we think that the size of the dataset is in itself significant and could provide some hints on the structure of the national activity.

We filtered the original dataset (removal of incomplete data) and divided it into subsets of different sizes, according to the span of time considered (day, week, month). In particular, studying different time granularities is useful to determine which time span is best suited to the netting procedures [2].

We use **weighted directed multi-edges graphs**, often referred to as a **weighted directed multigraphs**. This type of graphs is represented by a tuple $G = (V, E, w)$, where V is the set of nodes, E is the set of directed edges and w is the weight function over the edges. The edge with index i is defined as $e_i = (v_j, v_k)$, where v_j and v_k are distinct vertices in V . The weight function returns a positive value for a given edge index.

The particularity of multigraphs is that the same couple of distinct nodes may be in different edges. However, they are distinguished by the edge index.

In the economic context, we call our multigraphs *invoice graphs*. They are oriented as follows: the source of an arc is the receiver (debtor) while the destination is the emitter (creditor).

2.1 Comparative Study of Our Datasets

As mentioned above, we analyse our data on a monthly, weekly and daily basis in order to identify specific properties according to time granularity.

Aggregation per Month. The monthly decomposition offers several advantages. It is commonly used for financial reporting purposes, providing a concise overview of companies’ revenue and expenses over time. Furthermore, it facilitates the evaluation of overall performance, offering a holistic view of revenues and expenses by comparing the trends from month to month. This method enables long-term decision-making, including activities like setting budgets or identifying long-term growth trends when comparing year-on-year performance. Particularly, if one possesses data spanning more than a year, these trends become crucial for prioritization within a netting process.

When we compare each month, Fig. 1 shows that the start of year and the month of August are lighter in terms of transactions. It seems quite logical as the month of January is a month of slowdown for companies that allocate their budget, plan their decision-making strategy and have a decline in sales after the end of year rush. On the other hand, August registers a lower number of invoices because it is the preferred month for companies’ Summer holiday in Italy. We observe similar behaviours for nodes, edges and weights variations.

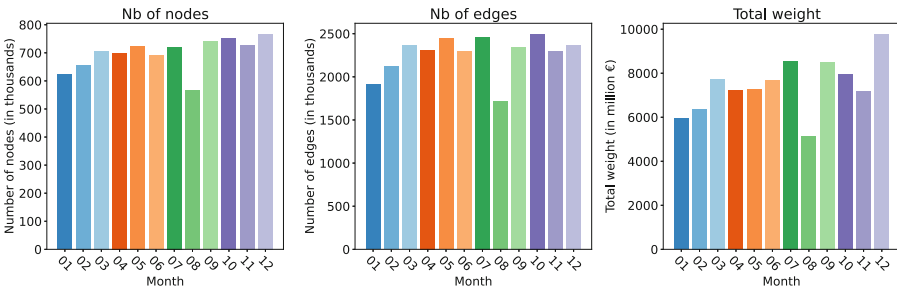


Fig. 1. Variations of nodes, edges and weights on monthly graphs. Colour tones depend on the trimester.

Aggregation per Week. The advantages of making weekly cuts include the ability to detect short-term trends. This approach is beneficial for reporting and mid-term planning as it offers more detailed insights than broader monthly cuts. Additionally, this type of cut smoothens the daily fluctuations and highlights cyclical activity patterns. Indeed, weekly cuts allow for a closer view on the working capital turnover.

Looking at the invoice distribution on a weekly basis, we can see in Fig. 2, that companies operate mostly on a monthly billing cycle, sending their invoices

grouped on the last week of each month. This highlights the need in terms of efficiency for companies to send invoices in bulk. Also, it allows for payment alignment with their clients’ schedule. From our perspective, it could be interesting to tackle invoices by performing the netting algorithm twice a month. According to the edge distribution over a month, half of the transactions happen on the first three weeks and the second half occurs during the last week. So, it appears reasonable to apply the netting on the first three weeks separately from the last one. As for monthly cuts, variations on each graph are similar.

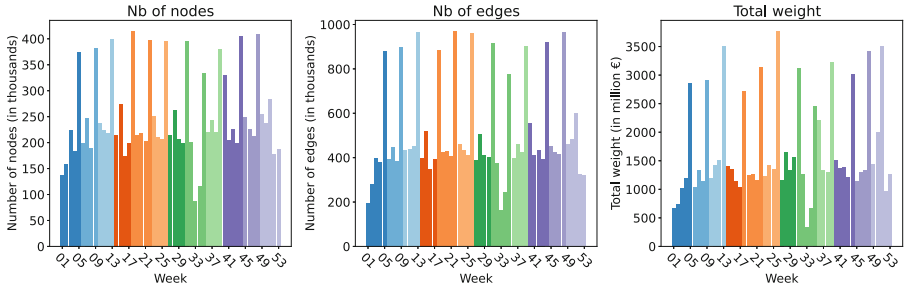


Fig. 2. Variations of nodes, edges and weights on weekly graphs. Colour depends on the trimester and the month of the first day of the week.

Aggregation per Day. From an economic point of view, cutting our data by day provides a high level of granularity for a detailed view of transactions and cash flow patterns. It is useful to identify short-term trends or irregularities and for real-time monitoring, in order to promptly respond to issues of delayed payments for example. Moreover, it gives an insight for operational decision-making, improving the management of inventories and resources allocation of the different companies.

From the graph analysis point of view, this aggregation gives us a fine-grain decomposition into small sub-graphs that may be easier to process by a netting algorithm. Figure 3 gives the total weights of incomes for each daily sub-graphs. General periodic patterns can be observed for weeks and months with only one exception in June with an extremely high concentration of volume on the last day of each month. As before, the total number of nodes and edges are strongly correlated to the weights, so we do not include them here.

2.2 Statistical Study on Monthly Graphs

We now focus on monthly cuts as they provide graphs that are less subject to variations and they are more densely connected than weekly and daily cuts.

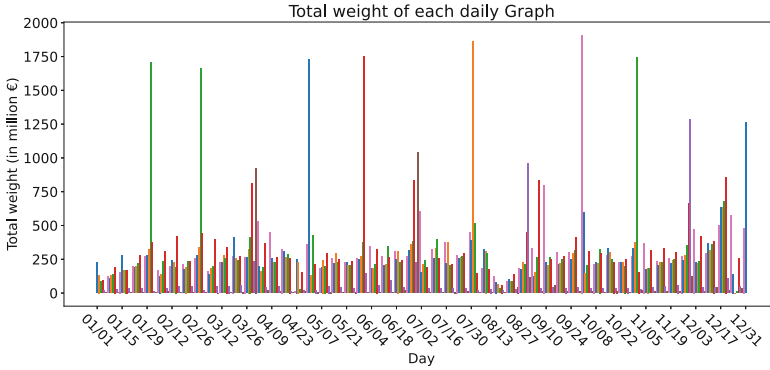


Fig. 3. Total weights of daily graphs. Colours depend on the day of the week.

Table 1 gathers a set of statistics exhibiting particular behaviours. Concerning the diameters, they are obtained after removing all nodes of degree one (about a third of the nodes). Average path lengths are obtained by computing shortest paths between ten thousand random pairs of nodes.

Looking at our data from a correlation point of view¹, we see that the diameter and the clustering coefficient do not seem to be especially correlated with the size of our network and are quite stable from one month to another. The low values of clustering coefficient indicate that we are far from having a Small-World network in which the clustering coefficient is much higher [10].

Also, we observe that the diameter and the average path length are quite small compared to the number of nodes and strongly correlated.

For the month of January, only thirty nodes have a degree higher than a thousand. These nodes can be considered as mega hubs in the network since they interact with more than 0.1% of the rest of the nodes. Coupled with the low diameter and average path length, it means that no node is far from a hub, linking it to the rest of the graph.

2.3 Comparison with Well-Known Families of Graphs

We now compare B2B invoice networks with Small-World and Scale-Free networks as they are the most commonly used to analyse real world networks. Such a comparison is relevant as in the case of similarities with a given family, we could take advantage of the identified structure to design efficient netting algorithms.

Small-world graphs were famously described by social psychologists Milgram and Travers in their “six degrees of separation” experiment [18], in which they found that individuals are, on average, separated by only a few acquaintances. This type of networks was formalized by Watts and Strogatz [19] and exhibits a high clustering coefficient and a short average path length.

¹ More information are available in an [extended version](#) on the French preprint server HAL.

Table 1. Statistical values describing our twelve monthly and the annual graphs. GWCC and GSCC respectively stand for Giant Weakly and Giant Strongly Connected Component.

Month	01	02	03	04	05	06	07	08	09	10	11	12	Annual
Nodes (10^3)	623	657	707	699	723	691	721	566	740	754	729	767	2057
Edges (10^3)	1910	2120	2362	2311	2443	2296	2462	1723	2348	2496	2301	2364	27138
Mean Degree	3.07	3.23	3.34	3.31	3.38	3.22	3.42	3.05	3.17	3.31	3.16	3.08	13.19
Clustering Coefficient	0.010	0.007	0.010	0.007	0.007	0.013	0.007	0.008	0.011	0.011	0.007	0.016	0.028
Diameter	29	29	28	28	31	25	27	32	28	33	31	27	34
Average Path Length	9.57	9.14	9.26	9.52	9.34	9.51	9.24	10.25	9.64	9.31	9.20	9.42	9.37
Nodes GWCC (%)	99.19	99.20	99.34	99.32	99.36	99.29	99.40	99.07	99.40	99.42	99.40	99.45	99.95
Edges GWCC (%)	99.80	99.20	99.84	99.84	99.85	99.74	99.85	99.75	99.84	99.85	99.84	99.85	99.97
Nodes GSCC (%)	7.70	8.58	9.09	8.88	9.35	9.15	9.50	7.01	8.69	9.26	8.77	8.99	11.83
Edges GSCC (%)	34.87	34.75	36.86	36.42	38.08	36.53	37.87	32.07	33.64	37.14	36.18	36.1	47.34

Scale-free networks, characterized by hubs with many connections, resemble many real-world systems like social networks and the internet [1], as highlighted by Faloutsos and al. [6]. Their structure affects how information spreads. In these graphs, edges’ growth is generally faster than nodes’ growth due to the property of preferential attachment. The presence of hubs enhances communication efficiency but also makes networks vulnerable to targeted attacks. Barabási et al. showed the signature power-law degree distribution of these networks [3]. Their main feature is the existence of power-law distributions with the conservation of properties at different scales.

2.4 Distribution of Degrees and Weights

The link between nodes and edges suggests a power-law relationship (Table 1). This suggests that our graphs follow the usual scale-free property of economic networks which is the typical behaviour of scale-invariant graphs. The degree distribution resembles a power-law but presents a large tail. Indeed, the plot of degree distribution on log-log graphs shows good agreement with straight lines. These lines are the mark of a power-law distribution but other distributions such as the negative binomial distribution can be considered, as raised by Fricke and Lux in their article on interbank networks [8].

In order to find the power-law coefficients that fit our graphs, the method of the Maximum Likelihood Estimator (MLE) has been used on our entire dataset. The different gamma values obtained with MLE are provided in Table 2. As strong differences have been observed between In and Out degrees in our graphs, it is better suited to distinguish their power-law analysis. Figure 4 provides the different histograms as well as the overall degree rank plot. These plots are made using density in order to make the comparison easier visually and to have normalized data.

To judge the fitting quality of those power-law, the generalized r^2 proposed by Cox and Snell [5] was used for each month on the whole data interval. For the In degree fitting, the minimal value found was $r^2 = 0.95$ in December which means that a power-law fits quite well with the In degree distribution. The results for Out degree laws are lower with the best being $r^2 = 0.72$, meaning that this distribution does not precisely correspond to a power-law. Concerning the rank plot, its global linear aspect tends to empower the idea of a construction following power-laws. It is an open problem to understand this law asymmetry between In and Out degrees.

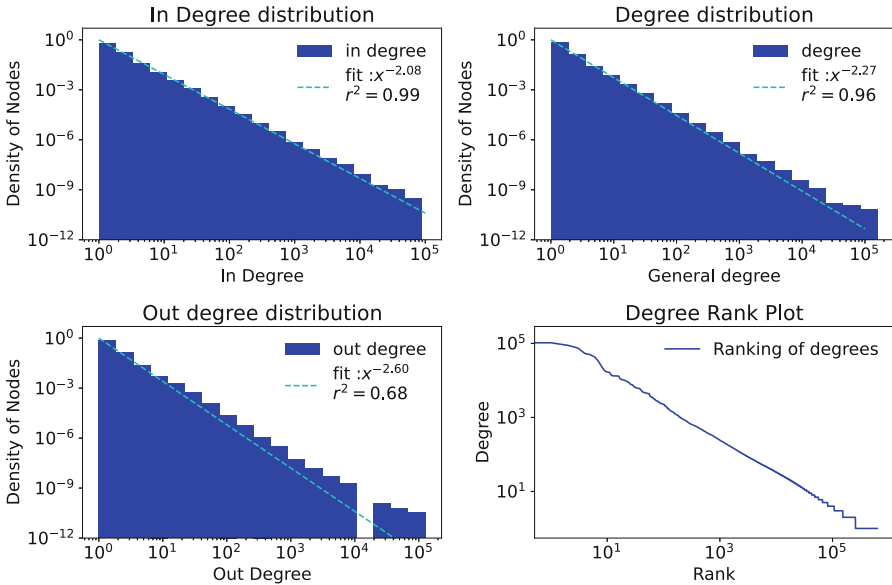


Fig. 4. Distibution of degrees for the month of January

Table 2. Values of our different gammas found by MLE. SD is the Standard Deviation between months.

Month	01	02	03	04	05	06	07	08	09	10	11	12	Mean	SD
$\gamma_{General}$	2.27	2.25	2.21	2.22	2.23	2.22	2.22	2.19	2.23	2.21	2.23	2.23	2.23	0.02
γ_{Out}	2.60	2.62	2.61	2.59	2.54	2.55	2.52	2.51	2.58	2.53	2.55	2.61	2.57	0.04
γ_{In}	2.08	2.06	2.08	2.05	2.05	2.04	2.05	2.01	2.04	2.04	2.05	2.05	2.05	0.02

We also compare the gamma obtained for different intervals on the month of January since the tail of our data can be not representative because of variations.

We also cut the nodes of degree one in our intervals since they may be strongly biased by our dataset that is limited to Infocert clients’ invoices. The results in Table 3 show the variability of the gamma depending on the data considered. The interval that fits best all type of degree is the [2–50] range with a r^2 of 0.91 for In Degree distribution, 0.87 for general degree and 0.86 for Out degree. When taking all the data into account, a reason we don’t observe a high value of r^2 for the Out degree distribution probably comes from the fact that statistically speaking, the Out degree is smaller than the In degree, which generates a narrower distribution of Out degree and less reliable statistical results.

Table 3. Values of our different gammas found by MLE for the month of January depending on the range for data.

Range	Global	[2-50]	[2-100]	[2-500]	[2-1000]	[2-10000]	[2-∞[
γ_{General}	2.27	2.60	2.51	2.43	2.42	2.41	2.41
γ_{Out}	2.60	2.80	2.71	2.67	2.66	2.66	2.66
γ_{In}	2.08	2.40	2.31	2.22	2.20	2.19	2.18

As stated before, our aim in finding a classic graph family close to monthly cut graphs is to use their specific properties to improve netting algorithms. As seen above, our networks present potentially two distinct laws on the In and Out degree distributions. These laws allow us to consider the possibility of efficient graphs partitioning to apply netting algorithms.

3 Communities Detection in B2B Invoice Graphs

In this section we tackle the problem of detecting communities. As mentioned before, this would be useful to apply a decomposition of the netting problem. Indeed, since the netting problem we are interested in is NP-complete [2], this divide-and-conquer approach allows us to apply our algorithms on large graphs in reasonable time.

The main obstacle in community detection is the multiplicity of definitions of what a community is. In particular, in economy there are many ways to define communities, it could be actors of the same sector, of the same geographical perimeter, etc. In our case it is more relevant to define communities as groups of highly connected actors.

3.1 Two Direct Methods for Communities Detection

As a first step, we use two different approaches that are not the most popular ones for communities detection nowadays. However, they are still used due to their simplicity and acceptable results quality.

Matrix Reordering Methods. In some cases, adjacency matrices can allow one to visualize the structure of graphs. However, the difficulty is to find a reordering of the nodes that spacially separates the communities inside the matrix. Different methods exist and we applied some of those exposed by Behrish et al. in their survey [4]. We were surprised to notice that none of these approaches could reveal the presence of communities. After analysing the problem, we found out that it is not really surprising as even with strongly structured graphs such as Caveman graphs, this method does not produce good results as soon as some noise is added to the structuration of the graph (random links between communities)².

k -core Method. A simple idea to detect communities is described by Fortunato [7] (see Sect. 4). The idea is to get rid of nodes that have less than a certain degree, to update the degree of the other nodes and to repeat until there is no change in the graph anymore. This method named k -cores (k is the degree threshold) was first used by Seidman when working on social network graphs [17]. After trying this method, we found out that it often generates either one giant component or a myriad of small communities with high degree nodes. As shown in Table 4, each decomposition of the January graph presents a main weakly connected component (WCC) that concentrates most of the nodes and edges.

The mean degree is not linearly correlated to the core number. This comes from the fact that invoice graphs present some nodes of extremely high degree heavily influencing the mean degree when the size of the graph decreases. Concerning the diameter, we observe an expected decrease according to the increase of k . However, it is interesting to see that the decrease is not strictly monotonous. Finally, we can see that the strongly connected components are smaller than the weakly connected ones and their evolution is not monotonous. But, the higher the k , the higher the concentration inside the GWCC, highlighting our hyper-connected parts of the main graph. Indeed, suppressing small degree nodes tends to remove small connected components that are not highly connected making the SCC ratios increase. However, the removal of higher degree nodes tend to divide the SCCs so that their ratio decreases (which is not what we apply here).

These decompositions help us to find groups of extremely highly connected components we could use as a starting point for netting algorithms.

3.2 Communities Detection Using Modularity

The most usual method in community detection on big networks is the graph partitioning according to the measure of modularity. This parameter measures the degree of segregation or clustering of nodes within a network compared to what would be expected by chance (random network of the same size). It quantifies the extent to which a network can be divided into distinct, densely connected groups or communities. Readers that want more details about modularity can refer to the survey of Newman [16]. The classic way to create communities using

² Different visuals are available in an [extended version](#) on HAL.

Table 4. Statistical values describing different k -cores for the month of January

k -core	Complete	3-core	5-core	10-core	20-core	50-core	100-core	200-core
Nodes	622 941	147 521	74 386	28 066	9 434	2 180	876	355
Edges (10^3)	1 910	1 334	1 092	803	567	367	282	217
Mean Degree	3.07	9.04	14.30	29.01	60.00	169.20	323.42	612.00
Diameter	24	21	20	17	14	10	9	9
Nodes GWCC (%)	99.19	99.73	99.75	99.78	99.84	99.82	100.00	99.44
Edges GWCC (%)	99.80	99.93	99.93	99.95	99.95	99.96	100.00	99.89
Nodes GSCC (%)	7.70	27.32	37.25	47.09	57.01	73.30	78.88	82.25
Edges GSCC (%)	34.87	48.55	54.98	62.95	72.24	81.72	85.56	87.04

modularity to is the Louvain method, described by Lambiotte et al. in their article on communities for large networks [13].

This method takes a resolution parameter that guides the size of extracted communities. It allows one either to create numerous small groups of weakly connected nodes while providing good inter-connections between groups, or to produce a set of larger groups with less inter-connections. Having little groups is useful for computing algorithms but it may miss a part of the information contained in the inter-connection. Moreover, although the number of communities does not change much with the resolution, there is an increase of the number of large communities (see Table 5). This means that the largest communities merge and that the smaller ones stay on their own.

Table 5. Repartition of the nodes according to modularity

Resolution	0.5	1.0	1.5
Modularity	0.747	0.749	0.741
Number of communities	2170	2082	2058
Number of community with more than 5% of the nodes	4	6	9
Percentage of nodes in the biggest community	6.95	7.43	14.74
Percentage of edges in the biggest community	3.32	4.92	10.70
Percentage of inter-community edges	21.5	19.65	17.93

This method allows for the control of the sizes of the largest communities. It is efficient to find well-connected nodes but it does not affect smaller groups. These small groups are either in different connected components or not connected enough to bigger ones.

Another limitation of this modularity approach as well as previous methods is that they are based on grouping nodes and not on grouping edges. Contrary

to classic approaches, the netting problem requires to group invoices by similar amounts instead of grouping them with a node-based approach. Hence, an open problem is the research of new algorithms that would group edges by similarities rather than nodes.

4 Conclusion

We have proposed a two-fold study on B2B invoice graphs from Infocert. On the one hand, we focused on identifying distinctive characteristics of such graphs. On the other hand, we compared different methods to extract communities from these graphs.

Concerning the first aspect, we observe that B2B graphs do not correspond to small-world graphs, displaying lower clustering coefficients. They closely align with scale-free graphs, demonstrated by power-law degree distributions. These findings emphasize the presence of influential nodes in the network and global robustness to random failures. This also tends to align with the study of other real networks in economy that present a tendency to preferential attachment and so power-laws. Although not a perfect fit, this study advances our understanding of invoice graphs and their properties, crucial for our application on B2B networks.

Concerning the communities aspect, the different approaches for communities successfully identify densely connected node groups. Indeed, we observe the presence of a giant strongly connected component in our network. However, the obtained results show that further refinement is necessary to be fully suited for edge-centric applications such as netting.

This research not only advances our understanding of invoice graphs but also serves as a stepping stone for future work on B2B netting procedures. The imbalance between In and Out degree may have to be taken into account when working on debt netting algorithms.

Acknowledgement. The authors are grateful to Infocert for their trust and support.

References

1. Akella, A., Chawla, S., Kannan, A., Seshan, S.: Scaling properties of the Internet graph. Association for Computing Machinery, New York, NY, USA, pp. 337–346 (2003). <https://doi.org/10.1145/872035.872087>
2. Amato, M., Fatès, N., Gobbi, L.: The economics and algorithmics of an integral settlement procedure on B2B networks, 1 September 2021. <http://dx.doi.org/10.2139/ssrn.3915380>
3. Barabasi, A.-L., Bonabeau, E: Scale-free networks. *Sci. Am.* **288**(5), 60–69 (2003). <http://www.jstor.org/stable/26060284>
4. Behrisch, M., Bach, B., Henry Riche, N., Schreck, T., Fekete, J.D.: Matrix reordering methods for table and network visualization. *Comput. Graph. Forum* **35**, 24. Wiley (2016). <https://inria.hal.science/hal-01326759/document>

5. Cox, D., Snell, E.: Special Logistic Analyses. Analysis of Binary Data, pp. 26–105, 2nd edn. Chapman and Hall, London (1989)
6. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the Internet topology. SIGCOMM Comput. Commun. Rev. **29**(4), 251–262 (1999). <https://doi.org/10.1145/316194.316229>
7. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3-5), 75–174 (2010). ISSN 0370-1573, <https://doi.org/10.1016/j.physrep.2009.11.002>
8. Fricke, D., Lux, T.: On the Distribution of Links in the Interbank Network: Evidence from the e-Mid Overnight Money Market, January 2013. <https://EconPapers.repec.org/RePEc:zbw:ifwkwp:1819>
9. Gaffeo, E., Gobbi, L., Molinari, M.: The economics of netting in financial networks. J. Econ. Interact. Coord. **14**, 595–622 (2019). <https://doi.org/10.1007/s11403-018-0229-4>
10. Gu, L., Huang, H.L., Zhang, X.D.: The clustering coefficient and the diameter of small-world networks. Acta Math. Sin.-Engl. Ser. **29**, 199–208 (2013). <https://doi.org/10.1007/s10114-012-0387-6>
11. Iosifidis, G., Charette, Y., Airoidi, E.M., Littera, G., Tassioulas, L., Christakis, N.A.: Cyclic motifs in the Sardex monetary network. Nat. Hum. Behav. **2**, 822–829 (2018). <https://doi.org/10.1038/s41562-018-0450-0>
12. Klein, M.: A primal method for minimal cost flows with applications to the assignment and transportation problems. Manag. Sci. **14**(3), 205–220 (1967). <https://www.jstor.org/stable/2627433>
13. Lambiotte, R., Lefebvre, E., Blondel, V., Guillaume, J.L.: Fast unfolding of communities in large networks, July 2008. <https://doi.org/10.1088/1742-5468/2008/10/P10008>
14. Leinonen, H., Soramäki, K.: Optimizing liquidity usage and settlement speed in payment systems. Discussion Papers 16/1999, Bank of Finland, Helsinki (1999). <https://dx.doi.org/10.2139/ssrn.228249>
15. Guntzer, M.M., Jungnickel, D., Leclerc, M.: Efficient algorithms for the clearing of interbank payments, pp. 212–219 (1998). [https://doi.org/10.1016/S0377-2217\(97\)00265-8](https://doi.org/10.1016/S0377-2217(97)00265-8)
16. Newman, M.: Communities, modules and large-scale structure in networks. Nat. Phys. (2011). <https://doi.org/10.1038/nphys2162>
17. Seidman, S.: Network structure and minimum degree. Soc. Netw. **5**(3), 269–287 (1983). ISSN 0378-8733, [https://doi.org/10.1016/0378-8733\(83\)90028-X](https://doi.org/10.1016/0378-8733(83)90028-X)
18. Travers, J., Milgram, S.: An experimental study of the small world problem. Sociometry **32**(4), 425–443 (1969). <http://www.jstor.org/stable/2786545>
19. Watts, D., Strogatz, S.: Collective dynamics of small-world networks. Nature **393**, 440–442 (1998). <https://doi.org/10.1038/30918>



A Model and Structural Analysis of Networked Bitcoin Transaction Flows

Min-Hsueh Chiu and Mayank Kejriwal^(✉)

University of Southern California, Los Angeles, CA 90292, USA
{minhsueh,kejriwal}@isi.edu
<https://usc-isi-i2.github.io/kejriwal/>

Abstract. Despite its unconventional origins, Bitcoin has emerged as the predominant cryptocurrency in the modern era and has entered mainstream discourse. Transactions in the Bitcoin ecosystem are different from those of ordinary finance, because of the way in which the cryptocurrency was designed. A proper structural understanding, and model, of Bitcoin transaction flows has largely been lacking. In this paper, we propose a model (based on directed acyclic graphs) that enables us to conduct structural analysis of networked Bitcoin transaction flows. Our model includes ‘activity’ measures, analogous to liquidity measures in ordinary financial markets, that are inspired by intuitions from thermodynamics. We apply the model and the activity measures to conduct structural analysis on a large transaction dataset from the early days of Bitcoin (first five years) when it was most in flux, and its future was still uncertain. Among other findings, our structural analysis suggests that the activity measure is correlated with major news events that affected Bitcoin. Our model could potentially be used to study other cryptocurrencies for which transaction data is available, as well as more recent Bitcoin transaction data.

Keywords: Bitcoin · transaction flows · directed acyclic graphs · economic complexity

1 Background and Motivation

Bitcoin was proposed in a white paper in 2008 [16], along with the underlying concept of *blockchain* [22], and has since entered mainstream discourse [4]. This unique technology purports to address several shortcomings in the current fiat monetary system through features such as decentralization, immutability, transparency, limited supply, and global accessibility. Bitcoin depends on an underlying blockchain, which serves as the ledger on which transactions are recorded. Bitcoin ‘miners’ validate these transactions, which can be conducted by any individual (or digital entity) with the required computational tools. This mining process, also called *proof-of-work*, is crucial for realizing the decentralization that is a cornerstone of Bitcoin. The miners are incentivized with block rewards and transaction fees. While the transaction fee is not a necessity for validating the transaction, it influences the speed at which the transaction gets

validated. The transactions utilize anonymous tokens to identify the owner, providing rigorous privacy. No overwrite is allowed once the transaction is recorded on the blockchain, which provides transparency and data integrity. Bitcoin has no denomination; in other words, it can be divided into any value with the smallest amount to eight decimal places (the so-called *satoshi*).

We schematically illustrate the structure of typical Bitcoin transactions in Fig. 1. Each color box with a solid outline represents a transaction. A transaction allows multiple inputs and multiple outputs. For example, transaction TX3 has one input and two outputs; input3 (฿10) is divided into output3 (฿6) and output4 (฿4). The transaction input is referenced from the previous transaction outputs, i.e., TX3's input3 is referenced from TX1's output1, which is (฿10). The spendable Bitcoin is identical to the unspent transaction outputs (UTXO). In this example, TX4 and TX6 are the UTXOs since they have yet to be spent. Once UTXO is spent and validated by miners, the UTXO will no longer be usable. The comprehensive explanations of Bitcoin and transaction details, such as address, wallet, and UTXO unlocking, fall beyond the scope of this work but we refer the interested reader to [1, 3] (for introductory reading) and to [7] for a more advanced and recent treatment, especially in the context of modern finance.

As the figure and this description suggest, the set of Bitcoin transactions is amenable to being modeled and studied as a complex system. Given the extensive research into user activity and transaction anonymity [17, 18, 21], our primary goal in this paper is to conduct a study focusing on the characteristics of transaction networks: first, by presenting a graph-based model for representing networked Bitcoin transaction flows; second, by proposing 'activity measures' for conducting structural analysis of networked transaction flows. In ordinary financial markets, like the stock market, an intuitive example of an activity measure is 'liquidity' i.e., do proposed buy/sell trades 'clear' (or are completed) near-instantaneously or is there a long lag and pricing friction? Liquidity is related to, but not necessarily equivalent to, the volume of transactions and trades occurring in unit time. In the Bitcoin ecosystem, developing an appropriate measure of 'liquidity' is an important and under-studied problem. We propose a measure that is inspired by thermodynamics, and use it to study the graphs that we construct.

While this work is related to other work on economic complexity [5, 6, 9, 10, 13], it is also different in that we are less interested in studying Bitcoin 'users' than transactions. Our work is more related to modeling attempts in other domains, such as illicit finance and human trafficking, to gain deeper insights into these phenomena using the tools of complex systems research [8, 11, 12, 14, 15]. There have been no studies, to our knowledge, at the intersection of Bitcoin and economic complexity. One important caveat here is that we focus on the early stages of Bitcoin, as we are interested in understanding Bitcoin transactions in the earliest days of the cryptocurrency's growth. There is an argument to be made that current Bitcoin markets are large enough that ordinary principles of economic theory and models can be applied to them. However, this was not

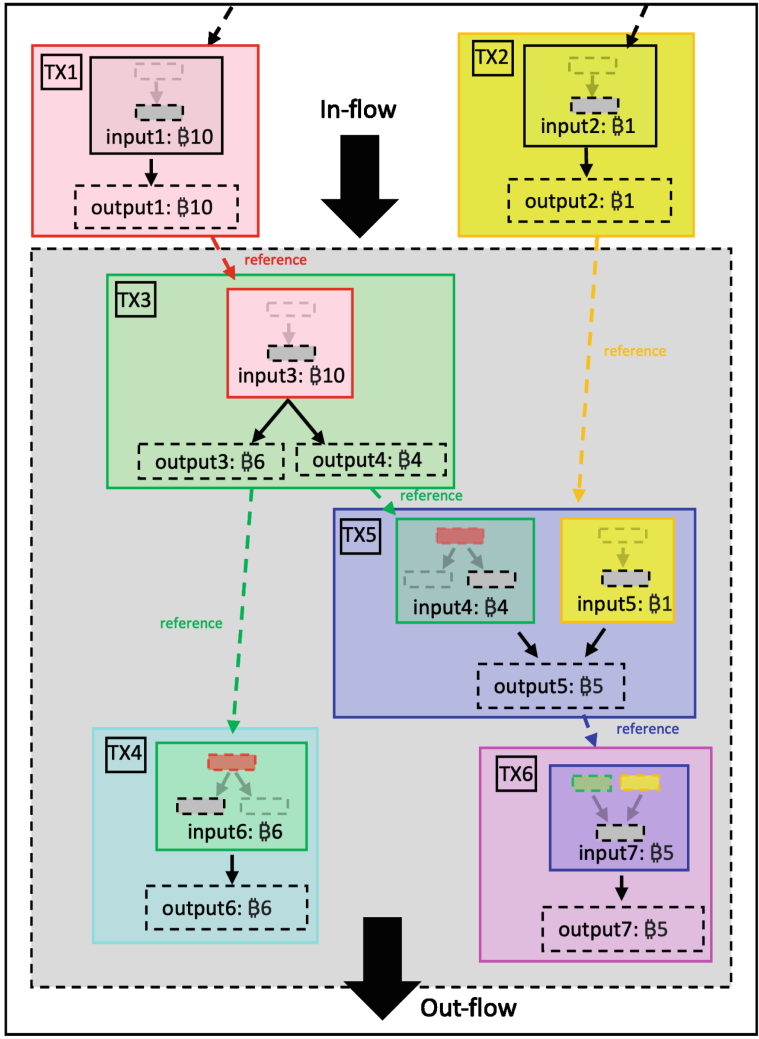


Fig. 1. Example of Bitcoin transaction

necessarily true in the first five years when Bitcoin price was extremely volatile (hence, it could not remotely be considered as a store of value), and since it could not be used for useful physical transactions, its future was uncertain. We posit that the methods of complex systems research are more appropriate for studying such a system. Nevertheless, applying some of our methods to more recent Bitcoin data is an important line of future work.

The rest of this paper is structured as follows. First, we describe a model for understanding the structural properties of networked transaction flows by constructing Directed Acyclic Graphs (DAGs). We then describe specific activity

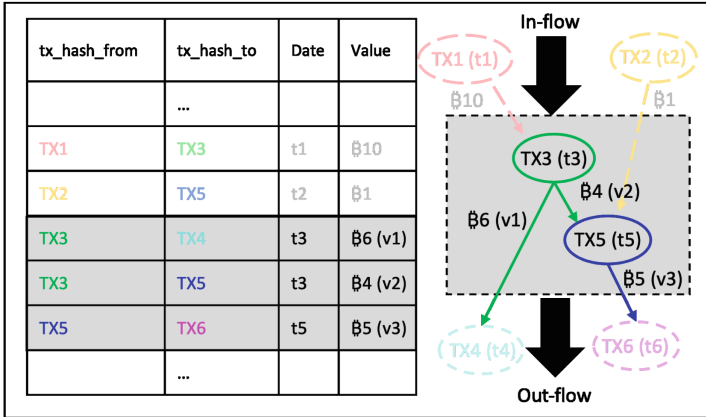
measures used for characterizing the transaction flows in this paper. We follow this with materials and methods, including details on the dataset we use for conducting the experiments. Next, we report the results of the study. Finally, we conclude the paper with a brief summary and some guidance on future research avenues.

2 Construction of Networked Transaction Flows as Directed Acyclic Graphs (DAGs)

Bitcoin is characterized by the unique transaction mechanism described in Sect. 1: it is sequential and cyclic, and each transaction is a one-time use. To capture both the Bitcoin amount involved in the transaction and the directionality of the trade, we propose Directed Acyclic Graphs (DAGs) as an appropriate model. A DAG is constructed for transactions in a given time period T . Within the DAG, the set of TXIDs is modeled as the node-set V . An edge is constructed to represent the actual Bitcoin flow between two TXIDs, which was validated in T . Formally, the edge-set E contains elements of the form (s, r, w) where s and r (representing the sending and receiving TXID) are both in V ; and w is the (always positive) weight on the edge representing the number of Bitcoins transacted. For example, $s \rightarrow r$ illustrates the directed edge and generates chronologically from the existing TXID s to the new TXID r . Transitivity and acyclicity are effectively encapsulated within the DAG structure.

Figure 2 shows an example of DAG construction; this is our constructed model to describe the identical transaction scenario in Fig. 1. The symbolic table on the left side includes five transaction records, with the constructed D visualized on the right side. Note that the attribute *Date* corresponds to `tx.hash.from`; it is not a time identifier for the edge. For example, in the third and fourth records, both Date values represent the transaction date of TX3. We further extract a portion of the whole D with limited T as D^T . In this example, T includes the three grey background records in the table, and the corresponding D^T is shown in the dashed outline box.

Importantly, the DAG is constructed by a given period T . We used a rule of thumb for determining T ; namely, a verified transaction is considered as confirmed after six more transactions are added after the transaction. However, the time interval between two transactions varies. Because we want T to be long enough to illustrate and measure transitivity, we determine its value by randomly sampling a date t in 2013, and building the corresponding DAG with T ranging from 1 to 14 days, where $T = 1$ includes all the transactions in $[t : (t + 1))$. This process was repeated ten times to mitigate the effect of the outlier. The results are shown in Fig. 3; each data point denotes the average of the size ratio $(\frac{|Node(D_{lc}^T)|}{|Node(D)|}, \frac{|Edge(D_{lc}^T)|}{|Edge(D)|})$ between the largest connected component (LCC) and all transactions in T . In both curves, the standard deviation gradually saturates after $T = 7$, and it covers most of the transactions ($>98\%$). Therefore, a value of 7 is chosen for parameter T to characterize the transitivity of the system.



Not used

Fig. 2. An illustration of the constructed DAG using sample Bitcoin transactions.

3 Activity Measures

This work considers the DAG as an ‘activated’ system, which can be analogized to a thermodynamics process. In the first law of thermodynamics, the internal energy can be described as $\Delta U = Q - W$, where Q is the heat added to the system, and W is the work done by the system. Similarly, we define activity (A) as the Bitcoin amount difference between the in-flow and the out-flow in the largest connected component of the DAG. Formally,

$$A = \sum_i u_i - \sum_j v_j$$

$$\forall u_i \in D_{lc}^T, N^-(u_i) \in \emptyset$$

$$\forall v_j \in D_{lc}^T, N^+(v_j) \in \emptyset$$

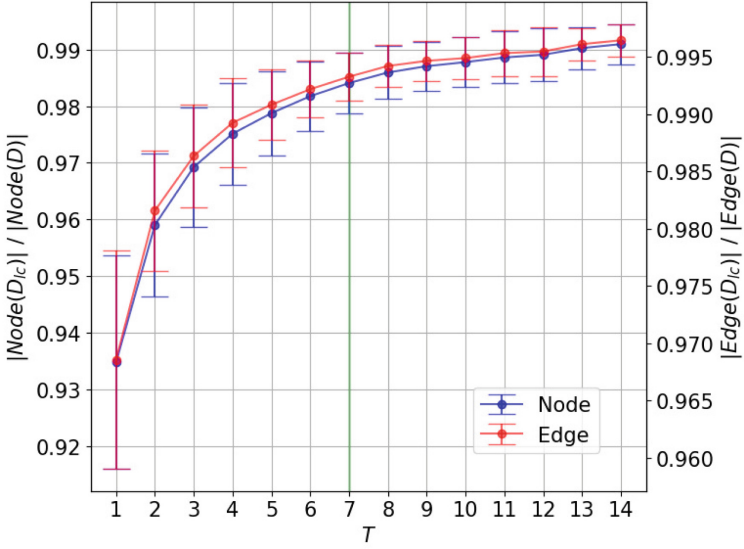


Fig. 3. Ratio of node and edge sizes.

where D_{lc}^T is the LCC in the DAG in a given period T . $N^+(\cdot)$ and $N^-(\cdot)$ is the out-neighbor and in-neighbor function, respectively. The remaining connected components are neglected due to the small volume (<2%) and lack of connectivity. In Fig. 2, $A = (v1 + v2) - (v1 + v3) = -1$. A is not necessarily 0 since we limited T ; the transactions' inputs might not be contained in $[t : (t + 7)]$, leading to a negative A . In the previous example, TX5 have two inputs TX3 and TX2, where $t2 < t$; as a result, $v2 < v3$. We consider TX2 as an external in-flow. In other words, the system needs additional in-flow (energy) outside the T , indicating the system is less liquid. This quantity describes how much Bitcoin activated during T remains in the system. A is also correlated to volatility, bid-ask spread, and stability.

4 Materials and Methods

4.1 Data

In this work, we used the Bitcoin Transactions dataset [20] released publicly on the IEEE DataPort platform in 2019. The dataset contains records from Bitcoin's inception (2009) until 2014 (but with 2014 only partially covered). However, the first official Bitcoin transaction (in this dataset) occurred on May 22, 2010. For our studies and subsequent modeling of transaction flows, we consider the transaction data from 2011 till the end of 2013, as not all transactions in 2014 are reported in this dataset.

The dataset is a structured table with four fields of interest for this work: two transaction identifiers (TXIDs) that uniquely identify the initiated transaction (`tx_hash_from`) and received (`tx_hash_to`) transaction, the `tx_hash_from`'s date, and the number of Bitcoins involved between the two transactions. We found that the transaction fee is not included in this dataset, i.e., in $\text{TX1} \rightarrow \text{TX2}$, TX1 's Bitcoin \geq TX2 's Bitcoin. This extends A 's upper bound and is positive when $T \rightarrow \infty$.

Some parts of the study also rely on historical market statistics that we downloaded (as CSV-formatted daily data) from [2]. The data includes price, open, close, high, low, volume, and percentage of change.

5 Experiments

5.1 Growing and Saturating Stages

The degree distribution from 2011 to 2013 is shown in Fig. 4a for investigating structural differences in DAGs in the few years after Bitcoin launched. Each color represents the degree distribution of the weekly DAG with a given starting date. As shown therein, each weekly DAG exhibits similar scale-free behavior. However, different stages are shown as the first half (purple) and the second half (magenta) of 2013 overlap. α fitted to the power-law distribution was found to be around 1.7 to 2.4, slightly smaller than that found in [19]. We suppose that the variation is arising from distinct time periods, as [19] constructed DAGs with all transactions spanning from 2009 to July 2011. The heavy-tailed structure closely resembles that found in [18], which utilized a more similar time frame as ours (2009–2013). The linear regression results on a log-log plot show the intercept gradually increases from 10^6 (January 2011) to $10^{12.5}$ (June 2013) and again decreases to 10^{11} (December 2013).

We define these two stages as the growing and saturating stages, which are separated on June 11th, 2013 as the maximum intercept ($10^{12.566}$) occurs on that date. The variable $\frac{|Node(D_{lc})|}{|Node(D)|}$ is shown in Fig. 4b. The edge size ratio has the similar trend as the node ratio, so we do not repeat the result for redundancy. During the growing stage, the average $\frac{|Node(D_{lc})|}{|Node(D)|}$ of 2011, 2012, and 2013 is 0.811, 0.965, 0.988; and $\frac{|Edge(D_{lc})|}{|Edge(D)|}$ is 0.872, 0.983, and 0.995, respectively; eventually, they saturate to 0.980 and 0.992 in the saturating stage. Both the small intercept and $\frac{|D_{lc}|}{D}$ in the growing stage indicate that the transactions remain relatively sparse; the average number of transactions per month in 2011 was one-twelfth of that in 2013 (saturating phase). This fact is also reflected in the price. The average price in 2011, 2012, 2013 (growing), and 2013 (saturating) was \$5.645, \$8.292, \$73.420, and \$298.785, respectively.

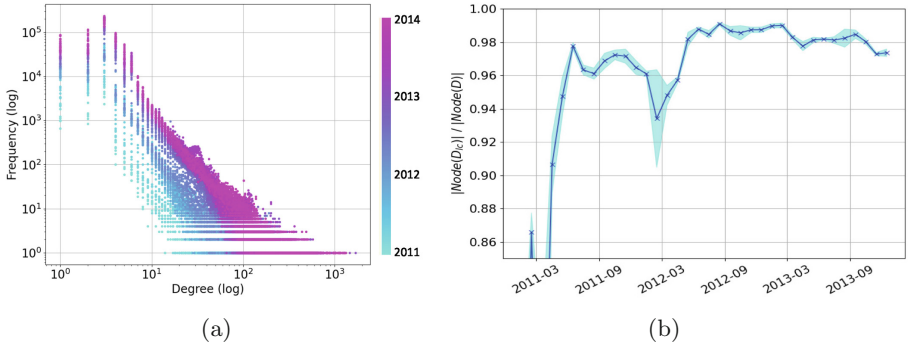


Fig. 4. (a) Degree distribution across various time intervals (b) variation in node sizes across diverse time intervals

5.2 Analysis in 2013

We look closer at 2013 as it contains both the growing and the saturating stages. The eight indicators are shown in Fig. 5: average price per week, average volume per week, activity, number of nodes, number of edges, average degree, and average cluster coefficient. We also show four important events (denoted by the red vertical lines) in the Bitcoin market in 2013. The first event was the *blockchain fork and split* on March 12th, caused by software bugs. The second event was the *closing of the Bank of Cyprus* on March 15th. The third event was the *Silk Road shutdown*. Silk Road is the dark web market that uses Bitcoin as the primary currency of illegal transactions. The Federal Bureau of Investigation (FBI) arrested the organizer Ross William Ulbricht and seized his laptop on October 1st. The final event was the *banning of Bitcoin by China*. The People’s Bank of China prohibited any transaction related to Bitcoin on December 5st. The green vertical line on June 11th in each subplot denotes the boundary between the growing and saturating stages.

We first investigate general observations between the two stages. The price shows to be relatively stable in the growing stage except for the minor rally in April. However, the price surged after October. The stages are defined differently regarding DAG’s degree distribution, but the daily rate of change (DROC) also shows consistency. In the early growing stage, it remained at around 0.02%; however, the DROC in the saturating stage oscillated about 0. The node and edge numbers show invariance to stages and vary in the identical distribution the whole time. However, distinctions were demonstrated in the other three higher-level statistics. The activity (A) remained negative but gradually increased, with a positive value shown in October, indicating the market is much more active and liquid, leading to a price rise. The degree and cluster coefficient shows a slightly lower value (on average) in the saturating stage, indicating the lower density and connectivity in DAG.

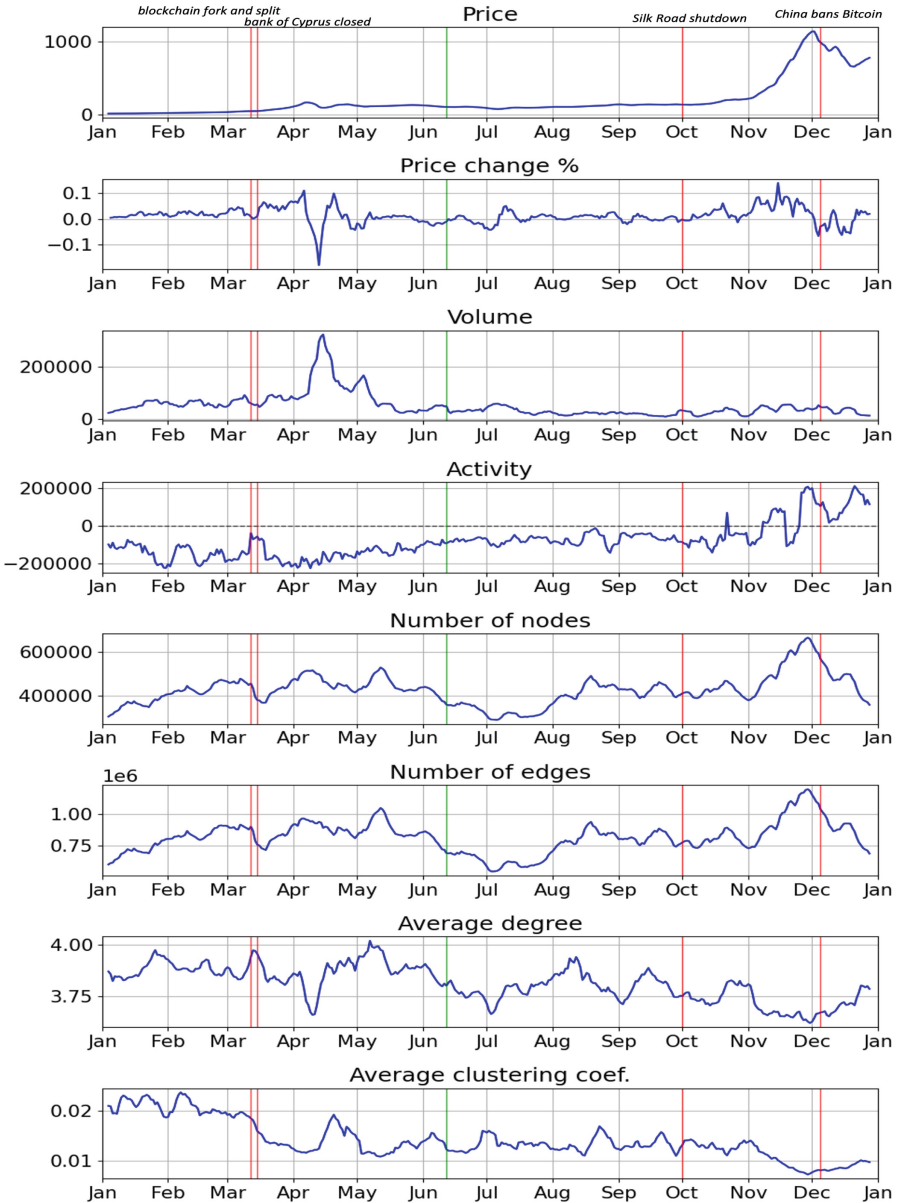


Fig. 5. Detailed analysis of Bitcoin market and activity variables in 2013.

The stable growth was disrupted with the occurrence of the blockchain fork and split event, which raised concerns about this novel technology’s feasibility. This event is immediately reflected on most indicators (except average degree). However, the significant drop in node and edge numbers might also come from

splitting the blockchain, making orphan blocks and transactions neglected in this dataset. This drop lasted briefly as the second event (closing of the Bank of Cyprus) happened in three days. The traditional banking system's financial collapse led to significant safety concerns. The novel and decentralized cryptocurrency attracted people's attention, leading to the first surging in 2013. The DROC reached up to 10.985% on April 6th. However, the sharp drop on April 13th (DROC = -17.910%) followed in a week, accompanied by the highest volume (up to \$300K) in 2013. This rapid price fluctuation highlighted Bitcoin's volatility and speculative nature. Interestingly, the activity A remains negative in this bubble period.

In the early saturating state, the node and edge numbers slightly decreased. Still, the average degree and clustering coefficient remained, which is consistent with the degree distribution's discovery. This status remained until the FBI shut down Silk Road in October. The attention of regulatory and security agencies, and possible legitimacy positively affected the Bitcoin price. This trend had lasted about two months; it reached the highest price at \$1144.62 on December 2nd the highest DROC up to 13.915% on November 15th. In the meantime, the node numbers and edge numbers also reached the highest in 2013. Also, the activity started to rise above 0. However, the degree and the cluster coefficient show opposite trends. The sparsity observed in the DAG may be associated with either the accumulation or distribution of Bitcoin, a phenomenon also observed during April. In cases of accumulation, where individuals are gathering Bitcoin, the output degree tends to be 1. Conversely, during instances of large-scale Bitcoin selling, the input degree tends to be 1. The bull market immediately turned into a bear market due to the fourth event, *China Bans Bitcoin*. This news reduced market enthusiasm, with the price halving. Intriguingly, the volume remained stable when these two events occurred.

6 Conclusion

This paper proposed a model based on DAGs for modeling networked Bitcoin transaction flows. We also proposed activity measures inspired by thermodynamics for capturing a natural intuition of 'liquidity' in this complex system. When applied to transaction data leading up to 2013, the model reveals some interesting structural properties of the Bitcoin ecosystem. We also show that the activity measure can allow us to reflect the effects of real world events on Bitcoin metrics, although the correlation is not perfect or always directionally predictable.

There are many avenues for future research. Bitcoin is still a relatively novel ecosystem, and many questions still remain about its growth and dynamics. Other cryptocurrencies have been even less studied. We believe that the framework presented in this paper offers a valuable way to conduct such studies. Conducting a study of the cryptocurrency market as a whole would be an ambitious agenda but well worth considering given its growing imprint on the financial markets.

References

1. <https://en.bitcoin.it/wiki/Transaction>
2. <https://www.investing.com/crypto/bitcoin/historical-data>
3. Antonopoulos, A.M.: *Mastering Bitcoin*, 2nd edn. O'Reilly Media, Sebastopol (2017)
4. Badea, L., Mungiu-Pupazan, M.C.: The economic and environmental impact of bitcoin. *IEEE Access* **9**, 48091–48104 (2021)
5. Hidalgo, C.A.: Economic complexity theory and applications. *Nat. Rev. Phys.* **3**(2), 92–113 (2021)
6. Hidalgo, C.A., Hausmann, R.: The building blocks of economic complexity. *Proc. Natl. Acad. Sci.* **106**(26), 10570–10575 (2009)
7. John, K., O'Hara, M., Saleh, F.: Bitcoin and beyond. *Annu. Rev. Financ. Econ.* **14**, 95–115 (2022)
8. Kejrival, M.: Domain-specific search engines for investigating human trafficking and other illicit activities. In: *Encyclopedia of Criminal Activities and the Deep Web*, pp. 478–496. IGI Global (2020)
9. Kejrival, M.: On using centrality to understand importance of entities in the panama papers. *PLoS ONE* **16**(3), e0248573 (2021)
10. Kejrival, M., Dang, A.: Structural studies of the global networks exposed in the panama papers. *Appl. Netw. Sci.* **5**(1), 1–24 (2020)
11. Kejrival, M., Gu, Y.: Network-theoretic modeling of complex activity using UK online sex advertisements. *Appl. Netw. Sci.* **5**, 1–23 (2020)
12. Kejrival, M., Kapoor, R.: Network-theoretic information extraction quality assessment in the human trafficking domain. *Appl. Netw. Sci.* **4**(1), 1–26 (2019)
13. Kejrival, M., Luo, Y.: On the empirical association between trade network complexity and global gross domestic product. In: Cherifi, H., Mantegna, R.N., Rocha, L.M., Cherifi, C., Micciche, S. (eds.) *Complex Networks and Their Applications XI. COMPLEX NETWORKS 2016 2022. SCI*, vol. 1077, pp. 456–466. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-21127-0_37
14. Kejrival, M., Szekely, P.: An investigative search engine for the human trafficking domain. In: d'Amato, C., et al. (eds.) *The Semantic Web – ISWC 2017. ISWC 2017. LNCS*, vol. 10588, pp. 247–262. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68204-4_25
15. Kejrival, M., Szekely, P., Knoblock, C.: Investigative knowledge discovery for combating illicit activities. *IEEE Intell. Syst.* **33**(1), 53–63 (2018)
16. Nakamoto, S.: *Bitcoin: a peer-to-peer electronic cash system*. Decentralized business review (2008)
17. Ober, M., Katzenbeisser, S., Hamacher, K.: Structure and anonymity of the bitcoin transaction graph. *Future Internet* **5**(2), 237–250 (2013)
18. Pham, T., Lee, S.: Anomaly detection in the bitcoin system—a network perspective. arXiv preprint [arXiv:1611.03942](https://arxiv.org/abs/1611.03942) (2016)
19. Reid, F., Harrigan, M.: An analysis of anonymity in the bitcoin system. In: Alshuler, Y., Elovici, Y., Cremers, A., Aharony, N., Pentland, A. (eds.) *Security and Privacy in Social Networks*. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-4139-7_10
20. Shafiq, O.: Bitcoin transactions data 2011–2013 (2019). <https://dx.doi.org/10.21227/8dfs-0261>

21. Yang, S.Y., Kim, J.: Bitcoin market return and volatility forecasting using transaction network flow properties. In: 2015 IEEE Symposium Series on Computational Intelligence, pp. 1778–1785. IEEE (2015)
22. Zheng, Z., Xie, S., Dai, H.N., Chen, X., Wang, H.: Blockchain challenges and opportunities: a survey. *Int. J. Web Grid Serv.* **14**(4), 352–375 (2018)



Rank Is All You Need: Robust Estimation of Complex Causal Networks

Cameron Cornell^(✉), Lewis Mitchell, and Matthew Roughan

The University of Adelaide, Adelaide, Australia
{cameron.cornell, lewis.mitchell, matthew.roughan}@adelaide.edu.au

Abstract. Financial networks can be constructed using statistical dependencies found within price series of speculative assets. Inference generally involves multivariate predictive modelling to reveal causal and correlational structures within the time series data, but difficulties frequently arise due to the highly unstable nature of these markets. The complex interplay of social and economic factors results in erratic behaviour, producing data that rarely adheres to theoretical assumptions. It remains unclear if these violations impact the constructed networks, and if so, whether robust alternatives produce more informative results. This study introduces the Rank-Vector-Autoregression model, demonstrating its capacity to produce robust cryptocurrency networks aligned with economic rationale. Our rank method achieves superior classification compared to the standard approach for various types of simulated data, particularly when including adversarial abnormalities. When applied to a dataset of 261 cryptocurrency return series, our method produces a network containing fewer, but more strongly market-correlated links, and increased connectivity within the mean-reversion subset. Applying our method to the squared deviations produces a comparatively dense volatility network, suggesting that significant price coupling occurs in higher order moments. Our results demonstrate the use of a robust and scalable technique for obtaining accurate causality networks in finance.

Keywords: Vector Autoregression · Rank Regression · Causality Networks · Financial Networks · Cryptocurrency

1 Introduction

Understanding the causes of price fluctuations in complex assets is beneficial. It aids investors and policymakers in understanding risk structures or generating out-of-sample forecasts capable of yielding profitable trading strategies. An important aspect of causal structure are networks that model the interdependencies between assets. In this paper we apply forecasting methodologies through an inferential lens, aiming to generate such causal networks.

The emergence of cryptocurrency markets offers a promising environment for the application of these techniques. These digital assets have attracted substantial attention due to their spectacular growth, but are mired in complexities: they

are highly volatile [14], arising from an intricate interplay of social and economic factors [14] — including market sentiment, news events, technological advancements, regulatory changes and the erratic behaviour of market participants. In response to this complexity, the analysis and modelling of cryptocurrency prices has emerged as a vital area of research. Here, we utilise causality networks to reveal the interdependencies within the cryptocurrency market.

A common and widely-accepted approach for multivariate forecasting is the Vector Autoregression (VAR) framework [24, 25]. This methodology extends univariate autoregression to account for the cross-dependencies among multiple time series. In this paper we develop an extension to the standard VAR model, aiming to increase the robustness of our estimation—a need that arises because the conventional assumptions of VAR are often violated in cryptocurrency data. The proposed model, Rank-VAR, employs a rank representation of the original series, reducing the effects of outliers and highly-leveraged observations.

We validate our method through simulations that illustrate its advantages, and then compare the standard and Rank-VAR methods on a dataset of 261 cryptocurrencies over a one-year period. We develop networks to represent both the mean response dependency, as well as the dependence structure for the volatility of returns. Our results indicate that robust techniques identify fewer links than the standard methodology (1.57% vs 3.78%), but the corresponding node degrees have stronger correlations to market capitalisation. We observe a marked propensity towards negatively-signed self-dependence, with the rank method finding an elevated number of these mean-reversion links (90% vs 85%). When applied to volatility modelling, the rank method again finds fewer links than the standard method, however both networks are substantially denser than their mean response counterparts (8% and 4%).

The primary contributions of this paper are:

- A robust extension to the VAR model, exploring its theoretical connections to copula modelling and framing its advantages through the lens of a multivariate extension to the Spearman correlation.
- A simulation study demonstrating the advantages of the Rank-VAR method when modelling VAR processes with structural violations. The results show a clear advantage in non-standard cases: in the best case, it improves the Area-Under the Curve (AUC) from 0.562 to 0.707, while never significantly underperforming the standard VAR method.
- An empirical application of Rank-VAR to one year of hourly return data from 261 cryptocurrencies, yielding robust causality networks that further our understanding of causality and risk in cryptocurrency markets.

Our analysis aims to not only provide more accurate networks, but also to determine the potential causes of dissimilarity between the robust and standard methods, which provides additional insight into the nature of causal structures in cryptocurrency markets.

2 Related Work

Networks based on correlational structures in financial asset time series data have been explored in many papers. Often, researchers investigate simultaneous correlations between prices, seeking to model the joint structure of these observations as a network [8]. Several studies analyze the temporal evolution of these networks using 'sliding window' methods [3, 17], with others investigating evolution by incrementally adding nodes based on their correlations, developing what is known as an "asset graph" [21].

Causal networks can be generated from cross-correlational effects. These types of networks have been constructed before [7] and are frequently used to model the joint causal dynamics of price and sentiment [5, 23]. Cross-correlational effects are typically modelled in a partial-effects framework to control for confounding variables. The VAR model is a common approach for this purpose, and is equivalent to partial effects of the cross-correlation matrix. Variations to this model have been developed to incorporate specific features, such as long range dependency [16] and restrictions to acyclic graphs [1].

Cryptocurrency causality network research has generally focused on the interactions between cryptocurrencies and other data (sentiment, traditional financial assets, *etc.*) [4–6, 13, 19]. While the incorporation of sentiment data has been explored through a VAR model [2], most analyses of cryptocurrency networks focus on bivariate analysis, rather than full partial-effects VAR models.

Various techniques have been developed for the robust estimation of VAR models [9, 12, 20]; however these methods generally focus on enhancing forecast performance while retaining interpretability and keeping coefficients within their original scale. When the focus is solely on the network, detailed interpretability of coefficients is not required. Our primary objective is to ascertain whether the time series are interdependent, reducing the problem to a binary outcome. This simplifies the technical elements and allows for less-complex robust estimators, such as the Rank-VAR we introduce. This study demonstrates the application of our robust technique to the development of cryptocurrency causality networks.

3 Methodology/Constructing Networks

The aim of causal network analysis is to construct a directed graph (digraph) denoted by $G = (V, E)$, where V is a set of nodes and E is a set of edges indicating causal dependencies. Each edge $e_{ij} \in E$ indicates that the next observation of asset j depends on the previous values of i . The set of edges E may be represented as an $N \times N$ adjacency matrix W (where $N=|V|$ is the number of series under study), with elements $W_{i,j} = 1$ if there is a link e_{ij} , and $W_{i,j} = 0$ otherwise.

The primary methodological choice when developing these networks from empirical data is the selection of a suitable model to test the interrelations within the time series, *i.e.*, to infer E . This section provides an overview of the common methodology for this estimation problem, as well our proposed extension.

3.1 Pearson and Spearman Correlation

The Pearson correlation $r(X, Y)$ has many advantages: including intuitive simplicity, straightforward calculation, and strong theoretical ties to \mathcal{L}_2 statistical analyses. However, this metric is sensitive to the input distributions $F_X(\cdot)$ and $F_Y(\cdot)$. Because it is based on squared deviation terms, data that contains outliers or fat tails may distort $r(X, Y)$. This sensitivity is particularly relevant to our causal analysis, as we establish links e_{ij} from a binary classification of test statistics tied to these correlation values r . Significant bias or variance in the metric will lower link accuracy in the causal network.

The Spearman rank correlation coefficient $\rho(X, Y)$ generalises $r(X, Y)$. It measures the degree to which two series are monotonically related, and is defined as the Pearson correlation of the variables after a rank transformation:

$$R(X_i) = |\{X_j : X_j < X_i\}| + 0.5|\{X_j : X_j = X_i\}| + 1 \quad (1)$$

Spearman's ρ is typically utilised as a robust alternative to the Pearson correlation, with less sensitivity to data outliers and fewer assumptions about the relational form. This is particularly valuable when analysing data with substantial skewness or kurtosis. More generally, it's emphasis on monotonicity assists in identifying nonlinear relationships, proving beneficial in scenarios where the co-linear assumptions of the Pearson coefficient are inappropriate. These features make it particularly suitable for analyzing cryptocurrency data, whose complex volatility means it varies significantly from traditional Gaussian behaviour [11].

3.2 Vector Autoregression

Vector autoregression (VAR) is a popular statistical model introduced by the macroeconomerician Christopher Sims [22] to model the joint dynamics and causal relations among a collection of time series. It is the natural multivariate extension of the univariate autoregression (AR) model frequently used to analyse the inter-temporal dependency of a sequence of observations. Under the VAR(p) formulation the expectation of the data vector \mathbf{y}_t at the next observation is a linear function of p previous observations. Equations 1 and 2 below show the relationship for order-1 and -p lagged variants:

$$\text{Order-1: } \mathbf{y}_t = A_1 \mathbf{y}_{t-1} + \mathbf{c} + \boldsymbol{\epsilon}_t, \quad (2)$$

$$\text{Order-}p: \mathbf{y}_t = A_1 \mathbf{y}_{t-1} + A_2 \mathbf{y}_{t-2} + \dots + A_{t-p} \mathbf{y}_{t-p} + \mathbf{c} + \boldsymbol{\epsilon}_t, \quad (3)$$

where \mathbf{y}_t is a $N \times 1$ vector of observations at time t , \mathbf{c} is a constant vector, the A_k are $N \times N$ coefficient matrices for lags $k = 1, \dots, p$, and $\boldsymbol{\epsilon}_t$ is a $N \times 1$ vector of error terms with zero mean and covariance matrix Σ_ϵ . The vector $\boldsymbol{\epsilon}_t$ usually originates from a Gaussian distribution. The VAR model assumes that the current value of each variable depends on its past values as well as the past values of all other variables in the system (full conditioning).

The estimation of a VAR model comprises estimating the coefficient matrices A_k and the error covariance matrix Σ_ϵ . As we are interested in the causal

influence structure within our dataset, we primarily require estimates of A_k , as they fully characterise the causal relations. This is often accomplished by the multivariate least squares (MLS) approach under which estimating the VAR is viewed as a general multivariate regression problem, with closed-form solutions generated via orthogonal projection [18].

We may conduct hypothesis tests for the statistical significance of the elements of the coefficient matrices by noting that our estimates \hat{A}_k are asymptotically normally distributed under finite variance assumptions, *i.e.*,

$$\sqrt{N} \text{Vec}(\hat{A}_k - A_k) \xrightarrow{d} \mathcal{N}(0, \Gamma^{-1} \otimes \Sigma_\epsilon), \tag{4}$$

where $\Gamma = YY'/N$, \otimes indicates the Kronecker product and $\text{Vec}(\cdot)$ denotes casting a matrix into vector form. For the case of a VAR(1) model the term Y is the matrix representation of our response data \mathbf{y}_t , implying that Γ is an estimate of the covariance matrix of returns. For generalised VAR(p) the complexity of this matrix increases, however Eq. 4 is still valid. To establish the existence of link e_{ij} we construct t values associated with the null hypothesis $A_{k,i,j} = 0$ as $t_{i,j} = \hat{A}_{k,i,j} / \hat{s}_{i,j}$, where $\hat{s}_{i,j}$ is the relevant term from $\Gamma^{-1} \otimes \Sigma_\epsilon$. These t vales can be used to generate a binary link classification with a false positive probability α as $e_{ij} = |t_{ij}| > \Phi(1 - \alpha/2)$, where Φ is the inverse normal CDF.

To streamline our discussion and estimation of causal networks, we limit our analysis to VAR(1) processes and omit the index p from our discussion, with $A = A_1$ unless otherwise specified. For simulations, we note that VAR(p) processes can be transformed into a VAR(1) form [18], implying that our VAR(1) simulation results should generalise as the fitting routines remain unchanged. For the empirical networks in Sect. 6.2 we operate under the assumption that any causal link $i \rightarrow j$ will first manifest in order-1 effects, and that lag $p > 1$ effects will not occur independently of a $p = 1$ dependence. This assumption is intuitive and the scenarios where it doesn't hold are expected to be relatively rare.

3.3 Rank Vector Autoregression

Rank-VAR is an extension to the VAR model, with the model variables transformed into their rank representation. This generates the multivariate time series analogue of the Spearman rank correlation, which can be expressed as:

$$\text{Order-1: } R(\mathbf{y}_t) = A_1 R(\mathbf{y}_{t-1}) + \mathbf{c} + \boldsymbol{\epsilon}_t, \tag{5}$$

$$\text{Order-}p: R(\mathbf{y}_t) = A_1 R(\mathbf{y}_{t-1}) + \dots + A_{t-p} R(\mathbf{y}_{t-p}) + \mathbf{c} + \boldsymbol{\epsilon}_t, \tag{6}$$

where the rank transform $R(\mathbf{y}_t)$ is applied elementwise. The advantages of this model are inherited from ρ , notably the heightened resilience to violations of the baseline model assumptions. The primary drawback is a reduction in interpretability, as coefficients are now related to ranks rather than precise values. This shift is also expected to diminish forecast precision in the original scale. However, the intended applications for Rank-VAR primarily encompass identification problems, particularly, deriving causal networks. For these binary outcomes, parameter interpretability and forecasting precision is irrelevant. These

concerns, however, do explain why this approach has not been adopted in the broader time series literature, and why non-temporal rank regression approaches have seen limited application beyond introductory studies [10, 15]. Before delving into simulations, it's essential to elucidate the theoretical ties of our methodology to other time series approaches. By noting that correlation/linear models are invariant to scalar multiplication, we can view rank imputation as functionally equivalent to a CDF transform. Specifically, the rank of an observation $R(X_i)$ simply counts how many X values are either equal to or less than X_i . Dividing these ranks by N , the number of observation gives us an approximation to the empirical distribution function:

$$\hat{F}_X(X_i) = \frac{1}{N} \sum_{j=1}^N 1_{X_j \leq X_i} \approx \frac{R(X_i)}{N}, \quad (7)$$

with the differences arising from $R(X_i)$ assigning only half a count to observations of an equal value (we take 'mid ranks'). From this we identify parallels between Rank-VAR and copula modelling. Using copulas, sets of series such as $\{X, Y, \dots, Z\}$ have their values transformed by their marginal CDF functions:

$$\{U_X, U_Y, \dots, U_Z\} = \{F_X(X), F_Y(Y), \dots, F_Z(Z)\},$$

where the U_i are uniformly distributed. The joint cumulative distribution function $C(u_x, u_y, \dots, u_z) = P(U_X < u_x, U_Y < u_y, \dots, U_Z < u_z)$ is then defined as the copula of $\{X, Y, \dots, Z\}$. This approach separates the contributions of the marginal distributions and their copula dependence structures. Sklar's theorem underpins this approach: every multivariate joint distribution can be decomposed into univariate marginals and a copula describing the dependencies.

Hence, by emphasizing ranks we are using a copula-like technique to isolate the challenges of joint dependence identification and the characterisation of the marginals. Yet, it's important to note that these parallels are approximate. We use empirical, rather than true marginal distributions, and our approach diverges from traditional copula modelling as we assess joint dependencies via conditional expectations, not direct modelling of the joint cumulative distribution function.

4 Simulation Methodology

4.1 Simulating VAR Processes

A VAR process is defined by the constant \mathbf{c} , recurrence matrices, A_i , and the distribution of the error process: $F_{\mathcal{E}}(\boldsymbol{\epsilon}_t)$ (with expectation 0 by design). The common simplifying assumption is that $\boldsymbol{\epsilon}_t$ is Gaussian, and hence the whole process is characterised by the covariance matrix Σ_{ϵ} . However, real data for complex systems commonly exhibit non-Gaussian distributions. We therefore seek to test VAR and Rank-VAR under such conditions, starting with the Gaussian case.

Generating Σ_{ϵ} : The first step is the construction of a random covariance matrix, whose validity requires both symmetry and positive definiteness. To

achieve this, we randomly rotate a set of positive eigenvalues $\{\lambda_i\}$, which are sampled as the absolute value of normal observations: $\{\lambda_i\} \sim |\mathcal{N}(0, 1)|$. Our random rotation matrix is obtained by taking the Q term from a QR decomposition of a randomly generated normal matrix. Our final covariance matrix is then:

$$\Sigma_\epsilon = QD(\lambda_i)Q^T, \tag{8}$$

where $D(\lambda_i)$ is the diagonal matrix of our eigenvalues.

Generating A: The second step is to generate valid coefficient matrices, ensuring the stationarity of the process. For this, the eigenvalues of A_i must lie within the unit circle. If they occupy the perimeter ($|\lambda_i| = 1$), the process will drift, while $|\lambda_i| > 1$ indicates a divergent process. Reusing the technique for generating Σ_t would result in non-sparse recurrence matrices, which poorly approximate our intended causal identification. Instead, we create a sparse Erdős-Rényi graph (with probability p that a link is selected) and assign Gaussian values to each edge to obtain A^* . The final matrix is normalised by \sim the absolute value of the largest eigenvalue: $A = A^*/\max(|\lambda_i| \times 1.05)$ in order to ensure that the process is stationary (the 1.05 ensures sufficient distance from drift).

4.2 Variations on VAR Processes

Fat-Tailed Distribution: While drawing the error series ϵ_t from a fat-tailed distribution is not explicitly a ‘violation’ of the VAR model, it is unconventional because the typical (often implicit) assumptions of narrow tails facilitate the use of \mathcal{L}_2 estimators, such as least squares projection. These estimators typically struggle with outliers, meaning their application to high-kurtosis distributions may result in large estimation errors. Various fat-tailed distributions exist, but a commonly used form is the mixed normal distribution. Here, the desired variable Z is conceptualised as a probabilistic mixture of two normal variables: $X \sim \mathcal{N}(0, \sigma_x)$ and $Y \sim \mathcal{N}(0, \sigma_Y)$, each having distinct variances. To generate observations of Z we draw from $F_X(x)$ with probability q or from $F_Y(y)$ with probability $(1 - q)$. Consequently, the cumulative density function for Z is then:

$$F_Z(z) = qF_X(x) + (1 - q)F_Y(y). \tag{9}$$

For fat-tailed behaviour, the parameters q and σ_X are selected to induce kurtosis. Typically, this is accomplished by taking a relatively small value for q , and $\sigma_X \gg \sigma_Y$. Given a fat-tailed error series ϵ_t , the production of a VAR simulation remains unchanged. We simply apply the linear filter A to the now fat-tailed error series, with the fat tails percolating through the system.

Post-Recurrence Spiking: We introduce a temporary, non-auto-regressive error component \mathbf{s}_t , which is sparse, but has a large ‘spike’ magnitude relative to typical values of \mathbf{y}_t . The resulting 1st-order process is:

$$\mathbf{y}_t = A_1(\mathbf{y}_{t-1} - \mathbf{s}_{t-1}) + \mathbf{c} + \epsilon_t + \mathbf{s}_t, \tag{10}$$

Contextual examples of such phenomena could include measurement artifacts or temporary economic shocks that are not expected to persist or propagate to other variables in the system. Real-world datasets often include rare but large artifacts that can distort analysis unduly. In simulations, we generate \mathbf{s}_t using the previously described mixture distribution, with $\sigma_x \gg \sqrt{\text{Var}(\epsilon)} \gg \sigma_y$.

Conditional Heteroskedasticity (VARCH): Here we consider serial correlation in the squared residual series ϵ_t^2 . This can be modeled univariately using the Generalised Autoregressive Conditional Heteroskedasticity (GARCH) framework, where the variance of the residual series is a function of past squared deviations, ϵ_t^2 , and the preceding variances σ_t^2 *i.e.*, in the univariate case:

$$\sigma_t^2 = w + \sum_{i=1}^q \alpha_i \epsilon_{t-i}^2 + \sum_{i=1}^q \beta_i \sigma_{t-i}^2 \tag{11}$$

for recurrence parameters α_i and β_i . The multivariate analogue to GARCH introduces substantial complexity, stemming from the potential to model each covariance $\sigma_i \sigma_j$ as having autoregressive relations involving the autoregression of entire matrices. We employ a simplified form where covariances respond solely to changing variances, without independent dynamic behaviour. This is referred to as the constant conditional correlation form (CCC-GARCH). Here, the covariance matrix of the residual series Σ_t can always be expressed in a diagonalised form, paired with a time-invariant correlation matrix R , *i.e.*, $\Sigma_t = D(\sigma_t)RD(\sigma_t)$, where only the σ_t term varies over time. The recurrent form of σ_t^2 then reuses the VAR structure, with squared errors in place of the original values:

$$\sigma_t^2 = A_1 \epsilon_{t-1}^2 + A_2 \epsilon_{t-2}^2 + \dots + A_p \epsilon_{t-p}^2 + \mathbf{c}. \tag{12}$$

This excludes the lagged σ_t^2 terms, implying that the volatility follows a VAR, not VARMA process. Our simulations follow a simplified scenario with $p = 1$, leading to short term volatility trends. We derive A as the elementwise absolute value of a matrix generated according to Sect. 4.1. The process is then generated by recursively forming σ_t^2 from (12) and diagonalising to obtain Σ_t .

Nonlinear Recurrence Functions: Another avenue for modification lies in altering the nature of recurrence itself, by allowing a nonlinear recurrence function. Specifically, an elementwise monotonic activation of the recurrence input. Instead of the recurrence in Eq. 2, we introduce the transformed recurrence:

$$\mathbf{y}_t = A_1 \phi(\mathbf{y}_{t-1}) + \mathbf{c} + \epsilon_t, \tag{13}$$

for some monotonic function $\phi(\cdot)$, applied elementwise, with an added restriction of sub-linearity to ensure stationarity of the resulting process. We select both centered Sigmoid (logistic) and Rectified Linear Unit function (ReLU) functions:

$$\text{Sig}(x) = \frac{1}{1 + e^{-x}} - \frac{1}{2}, \quad \text{and} \quad \text{ReLU}(x) = \max(0, x). \tag{14}$$

Scenarios exist where such relations are plausible. For instance, in the financial domain, one may observe *draw-down correlation*, where returns are correlated in the negative region, but remain largely independent for positive returns.

5 Simulation-Based Validation

Simulation results for 100 processes (10,000 links) of 1,000 observations with parameters: $N = 100$, $c = 0$, $\sigma_\epsilon = 1$ (standard error), $\sigma_{fat \epsilon} = 10$ and $q=3\%$. s_t has spike $\sigma_x = 10$, and non-spike $\sigma_y = 0.1$. These settings correspond to ~ 3 years of daily data, featuring monthly spikes or fat-tailed events.

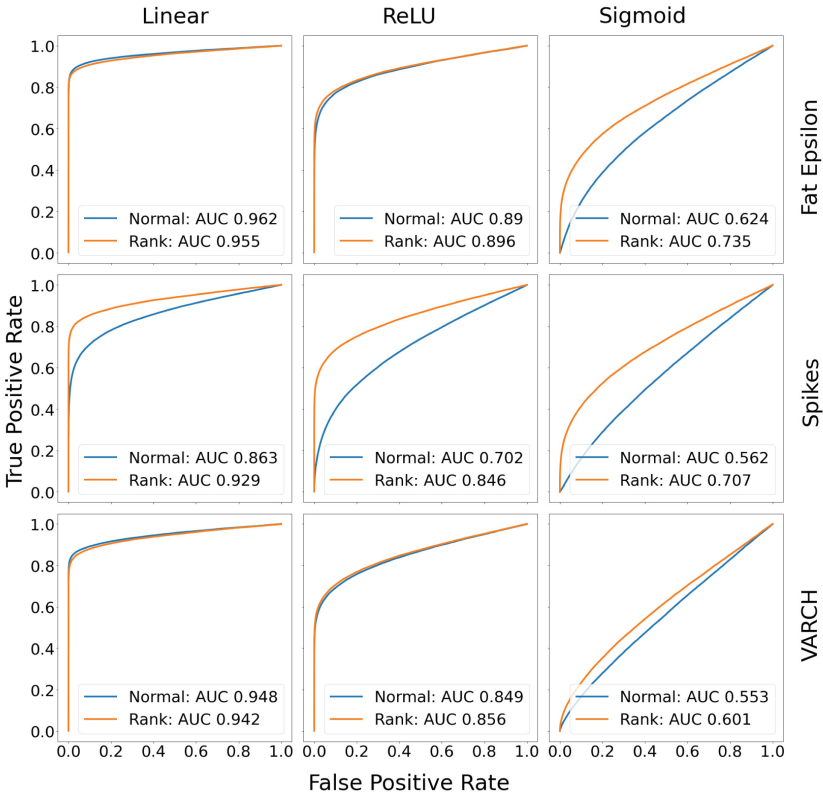


Fig. 1. ROC curves of VAR and Rank-Var for all combinations of noise model and recurrence function. AUC indicates the ‘overall’ performance of each model.

5.1 ROC Analysis

A key advantage of simulation is that we know the true correlation structure. This allows us to derive Receiver Operating Characteristic (ROC) curves, as

shown in Fig. 1. We show results for each combination of noise model and linkage-recurrence, with the exception of the standard error case, where the difference in performance was negligible. In each scenario, $|t|$ values for all potential links are estimated and thresholded at increasing values to generate a curve that demonstrates a model’s capacity to trade true positives for false positives. The ROC curves in Fig. 1 and associated Area Under Curve (AUC) values show:

1. Rank-VAR is never significantly worse than standard VAR.
2. Rank-VAR performs much better in some cases, notably:
 - in the post-recurrence spiking model (labelled “Spikes”), regardless of recurrence linking function, and
 - when there is a sigmoid recurrence linkage (though the improvement is dampened in the GARCH case).

In the most detrimental cases (Sigmoid spike or VARCH models), VAR loses almost all discriminative power, essentially choosing links at random. The most extreme improvement, Sigmoid recurrence with Spiking, shows a 25.8% increase in AUC, improving it from 0.562 to 0.707. The “Spike” plots show consistent performance improvement for Rank-VAR, with an average AUC increase of 18%. Understandably, this temporary spiking appears highly detrimental to standard link identification: spikes contain no cross-asset correlations, but two random, approximately contemporaneous spikes can easily pollute estimates.

5.2 Classification Analysis

To construct causal networks we create binary classifications for each potential link e_{ij} , based on whether the associated t_{ij} exceeds the threshold t^* corresponding to our target false positive rate α . Here we evaluate both standard and Rank-VAR using classification metrics on our simulated data.

Table 1a displays the binary classification results for simulated VAR processes, targeting a false positive rate, α , of 1%. The results for the combined

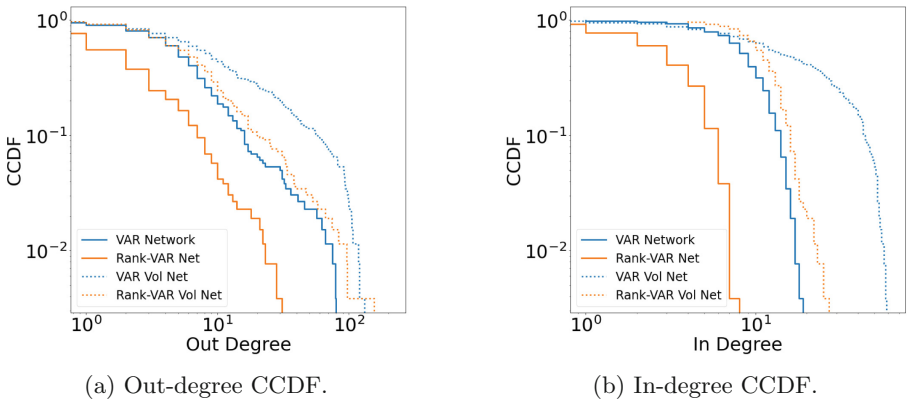


Fig. 2. Degree distribution CCDF plots. Results show mean response and volatility network curves for both standard and Rank-VAR.

Table 1. Rank network Results. Figure 1a displays Rank (ρ), and standard VAR (r) classification metrics for the simulation study, while Figs. 1 and 1c show the empirical Network Metrics: mean, median, standard deviation σ , Spearman’s rank correlation (ρ) with market capitalisation and associated \mathbf{p} value for capitalisation independence.

	Error	Type	$\hat{\alpha}$	Precision	Recall	F1		Attribute	mean	median	$\sigma_{attribute}$	ρ	\mathbf{p}	
Linear ϕ	None	r	0.010	0.82	0.86	0.84	VAR	Out-Deg.	9.79	6	13.6	0.193	1.75e-3	
		ρ	0.012	0.78	0.84	0.81		In-Deg.	9.79	10	3.68	0.0896	0.140	
	ϵ_t	r	0.015	0.75	0.86	0.81		Clust.	0.0972	0.0855	0.0603	0.129	0.0368	
		ρ	0.013	0.78	0.86	0.82		Central.	0.324	0.326	0.0964	0.207	7.53e-4	
	S_t	r	0.032	0.51	0.62	0.56		Rank-VAR	Out-Deg.	4.13	3	5.02	0.238	1.05e-4
		ρ	0.018	0.69	0.80	0.74			In-Deg.	4.13	4	1.92	-0.216	4.31e-4
	VARCH	r	0.031	0.59	0.86	0.70			Clust.	0.0283	0.0064	0.0570	0.0109	0.862
		ρ	0.020	0.69	0.83	0.75			Central.	0.0307	0.0106	0.0538	0.225	2.48e-4

(b) Mean response Network

	Attribute	mean	median	$\sigma_{attribute}$	ρ	\mathbf{p}
VAR	Out-Deg.	20.9	10	27.6	0.306	4.47e-7
	In-Deg.	20.9	16	15.7	0.394	3.7e-11
	Clust.	0.239	0.233	0.134	0.304	5.65e-7
	Central.	0.0338	0.00889	0.0519	0.296	1.09e-6
Rank VAR	Out-Deg.	12.19	7	20.2	-0.00384	0.95
	In-Deg.	12.19	12	3.97	0.214	5.08e-4
	Clust.	0.146	0.141	0.060	0.093	0.135
	Central.	0.0332	0.0173	0.0523	0.0159	0.799

(c) Volatility Network

(a) Simulation Results

metric (F1 score - The harmonic mean of precision and recall) are generally commensurate with the AUC behaviour discussed previously:

1. Rank-VAR minutely underperforms the standard method for Linear and Relu models with standard errors.
2. Rank-VAR substantially outperforms standard VAR for several scenarios:
 - again in all post-recurrence spiking simulations,
 - all scenarios involving the VARCH type errors, and
 - in both nonlinear, fat-tailed ϵ scenarios.

All other scenarios displayed functionally equivalent performance.

The empirical false positive rate $\hat{\alpha}$ provides a potential explanation for the differences in AUC and F1 performance. Since AUC is agnostic to the actual t values (it only considers their ordering), the effect of elevated $\hat{\alpha}$ is only visible

under binary classifications. The VARCH scenarios generate $\hat{\alpha}$ that are nearly double their target for standard VAR, thereby diminishing its F1 score. This discrepancy is most pronounced in the Linear VARCH, ReLU VARCH, and fat-tailed ReLU simulations, where we observe near-identical ROC performance, but superior classification for the rank-method. Here, Rank-VAR doesn't offer a meaningful advantage in distinguishing between true and false positives, rather, it demonstrates more consistent parameter convergence, such that the asymptotic normality of t_{ij} is better realised.

6 Empirical Data Analysis

6.1 Data

Our dataset contains hourly prices from 261 cryptocurrencies from 1/1/2021 to 1/1/2022, and market capitalisation (June 2022 figures) for each cryptocurrency. The selected currencies were derived from the 750 coins with highest capitalization at time of collection (June 2022). However, a significant number of these, mainly those with lower capitalization, had incomplete or missing price histories and had to be excluded. The resulting data contains 79.6% (836B of 1.05T) of the total capitalisation of the market. The returns y_t are generated by taking the logged ratio of subsequent observations in the original price series p_t , quoted in terms of the Coin/USD relation: *i.e.*, $y_t = \log(p_t/p_{t-1})$. For comprehensive details on this dataset see [11]. For our purposes, the key takeaway is that our dataset shows high levels of capitalization dependent a-normality, which may posit the use of robust methods such as Rank-VAR.

6.2 Empirical Networks

Using the data from Sect. 6.1, we construct causality networks with both VAR and Rank-VAR methods (Mean-response networks). We also construct two forms of volatility networks, by running the two aforementioned methods on the squared-centered series $(y_t - \bar{y})^2$. Note that these networks are not derived from the residual series of the first networks, and hence correspond to assumptions of mean-response efficiency in the cryptocurrency market (*i.e.*, a complete absence of statistically significant mean-response causality).

We first examine several structural network attributes, such as node degrees and their distributions. The node degree log-log complementary cumulative density functions (CCDFs) displayed in Figs. 2a and 2b show:

1. Out-degree distributions show less curvature compared to their In-degree counterparts, suggesting that outgoing influence within the network is more concentrated than incoming influence.
2. Both in-degree and out-degree distributions in the VAR volatility network show significant curvature. This pattern is potentially indicative of a high false positive rate, as a system dominated by noise would produce significantly curved binomial distributions.

We also explore the link between market capitalization and node attributes. Key network statistics, along with Spearman rank coefficients and p values, are in Table 1b. Standard VAR shows statistically significant correlations between market capitalisation and out degree, clustering and centrality. While in degree shows no significant correlation with market capitalisation, it does correlate with out degree ($\rho = 0.343$, $\mathbf{p} = 1.29\text{e-}8$). Rank-VAR exhibits similar correlation between out degree and centrality against market capitalisation. However, it lacks a significant correlation for clustering, and in degree is negatively correlated. Most strikingly, the Rank-VAR net contains an average of 4.13 (1.6%) links per node, compared to the 9.79 (3.7%) found using standard VAR.

Table 1b displays corresponding information for the Volatility networks. We observe a similar pattern for node degree counts, with the Rank network having an average of 12.19 (4.66%) edges per node, compared to the 20.9 (8%) found using standard VAR. The observed correlation structures differ significantly, with the VAR network having all metrics being correlated against capitalisation. Comparatively, the Rank-VAR has statistically significant correlations only between in degree and market capitalisation. Across both networks the overall edge count is significantly higher for the Volatility networks, suggesting that a significant proportion of the causal dependencies occur in the higher order moments.

The presence of self-edges indicates either autocorrelation of returns or self-excitatory behavior. For the VAR network, 85.4% of potential self-edges are statistically significant. In contrast to the previous section, Rank-VAR now reports an elevated detection rate, with 90.0% being statistically significant. Of these self-edges, 98.4% in VAR and 99.6% in Rank-VAR were negatively signed. Despite the general sparsity of our networks, this reveals a densely connected subset associated with the mean reversion of individual cryptocurrency prices. Equivalent behaviour is observed in the volatility networks, with Rank-VAR detecting 93.8% of self-links, compared to 88.8% for standard VAR. Once again there is strong asymmetry in effect, with 100% and 97.4% of self links being positively signed, indicating self-excitatory behaviour.

7 Conclusion

This study introduces a technique for robust link identification in multivariate causality networks. The Rank-VAR model is validated on simulated data featuring a range of recurrence and error abnormalities, achieving superior classification for the most detrimental variations. In our empirical study on 261 cryptocurrencies, the Rank-VAR network contained fewer but more meaningful links (1.7% vs 3.7%), with increased correlation to market capitalisation. Rank-VAR identified a marginally increased proportion of edges in the highly connected, negatively signed self-dependence subset of links. Combined, these findings suggest that the cryptocurrency market has relatively sparse cross-causality, with a significant proportion of the identified causal elements relating to mean-reversion of individual coin prices. When shifting our focus to volatility networks we observed substantially denser connections (4% Rank-VAR vs 8% VAR), indicating that asset cross-coupling may predominantly occur in higher-order or

symmetric moments. This has implications for both investors and policymakers, as it highlights that the relatively independent action of standard returns may not imply isolated behaviour during extreme market conditions. This study serves as a baseline for further research into robust financial causality networks, particularly for exploring the role of higher-order moments in asset coupling.

References

1. Ahelegbey, D.F., Billio, M., Casarin, R.: Bayesian graphical models for structural vector autoregressive processes. *J. Appl. Economet.* **31**(2), 357–386 (2016)
2. Ahelegbey, D.F., Cerchiello, P., Scaramozzino, R.: Network based evidence of the financial impact of COVID-19 pandemic. *Int. Rev. Financ. Anal.* **81**, 102,101–102,101 (2021)
3. Almog, A., Shmueli, E.: Structural entropy: monitoring correlation-based networks over time with application to financial markets. *Sci. Rep.* **9**, 10,832 (2019)
4. Amirzadeh, R., Nazari, A., Thiruvady, D., Ee, M.S.: Modelling determinants of cryptocurrency prices: a Bayesian network approach (2023). <https://ssrn.com/abstract=4403923>. Working paper, available at SSRN
5. Aste, T.: Cryptocurrency market structure: connecting emotions and economics. *Digit. Financ.* **1** (2019)
6. Azqueta-Gavaldón, A.: Causal inference between cryptocurrency narratives and prices: evidence from a complex dynamic ecosystem. *Phys. A Stat. Mech. Appl.* **537**, 122,574 (2020)
7. Billio, M., Lo, A., Sherman, M., Pelizzon, L.: Econometric measures of connectedness and systemic risk in the finance and insurance sectors. *J. Financ. Econ.* **104** (2011)
8. Boginski, V., Butenko, S., Pardalos, P.: Statistical analysis of financial networks. *Comput. Stat. Data Anal.* **48**, 431–443 (2005)
9. Chang, L., Shi, Y.: A discussion on the robust vector autoregressive models: novel evidence from safe haven assets. *Ann. Oper. Res.* (2022)
10. Chen, T., Tang, W., Lu, Y., Tu, X.: Rank regression: an alternative regression approach for data with outliers. *Shanghai Arch. Psychiatry* **26**(5), 310–315 (2014)
11. Cornell, C., Mitchell, L., Roughan, M.: Vector autoregression in cryptocurrency markets: unraveling complex causal networks (2023). ArXiv preprint [arXiv:2308.15769](https://arxiv.org/abs/2308.15769)
12. Croux, C., Joossens, K.: Robust estimation of the vector autoregressive model by a least trimmed squares procedure. In: Brito, P. (ed.) *COMPSTAT 2008*, pp. 489–501. Physica-Verlag HD, Heidelberg (2008). https://doi.org/10.1007/978-3-7908-2084-3_40
13. Elsayed, A.H., Gozgor, G., Lau, C.K.M.: Causality and dynamic spillovers among cryptocurrencies and currency markets. *Int. J. Financ. Econ.* (2020)
14. Giudici, G., Milne, A., Vinogradov, D.: Cryptocurrencies: market analysis and perspectives. *J. Ind. Bus. Econ.* **47**(1), 1–18 (2020)
15. Iman, R.L., Conover, W.J.: The use of the rank transform in regression. *Technometrics* **21**(4), 499–509 (1979)
16. Johansen, S.: A representation theory for a class of vector autoregressive models for fractional processes. *Economet. Theor.* **24**, 651–676 (2008)
17. Kenett, D., Tumminello, M., Madi, A., Gershgoren, G., Mantegna, R., Ben-Jacob, E.: Dominating clasp of the financial sector revealed by partial correlation analysis of the stock market. *PloS one* **5**, e15,032 (2010)

18. Luetkepohl, H.: *The New Introduction to Multiple Time Series Analysis* (2005)
19. Milunovich, G.: Cryptocurrencies, mainstream asset classes and risk factors - a study of connectedness (2018)
20. Muler, N., Yohai, V.J.: Robust estimation for vector autoregressive models. *Comput. Stat. Data Anal.* **65**, 68–79 (2013). Special issue on Robust Analysis of Complex Data
21. Onnela, J.P., Kaski, K., Kertész, J.: Clustering and information in correlation based financial networks. *Eur. Phys. J. B Condens. Matter* **38** (2003)
22. Sims, C.: Macroeconomics and reality. *Econometrica* **48**(1), 1–48 (1980)
23. Souza, T., Aste, T.: Predicting future stock market structure by combining social and financial network information. *Phys. A Stat. Mech. Appl.* **535**, 122,343 (2019)
24. Stock, J.H., Watson, M.W.: Vector autoregressions. *J. Econ. Perspect.* **15**(4), 101–115 (2001)
25. Toda, H.Y., Phillips, P.C.B.: Vector autoregression and causality: a theoretical overview and simulation study. *Economet. Rev.* **13**(2), 259–285 (1994)

Author Index

A

Abdo, Jacques Bou 359
Agarwal, Nitin 15, 208
Akinnubi, Abiola 15
Alassad, Mustafa 15, 208
Amann, Bernd 332
Amato, Massimo 444
Amure, Ridwan 15
Avrachenkov, Konstantin 384

B

Behrouz, Ali 49
Bernasconi, Anna 344
Biró, József 271
Boldi, Paolo 102
Bou Abdo, Jacques 371
Bougiatiotis, Konstantinos 75
Brienen, Marten 435

C

Chattopadhyay, Chiranjoy 234
Cherifi, Hocine 296
Chiu, Min-Hsueh 456
Cialfi, Daniela 423
Constantin, Camelia 332
Contassot-Vivier, Sylvain 444
Cordova, Giulio 183
Cornell, Cameron 468
Cortese, Francesca 114
Costas, Rodrigo 147

D

D'Ascenzo, Davide 102
Dass, Shuvalaxmi 371
Dutta, Animesh 309

F

Fatès, Nazim 444
Ficzere, Dániel 271
Frankó, Attila 271

Frost, Hildreth Robert 3
Furia, Flavio 102
Fushimi, Takayasu 37

G

Gaigaliene, Asta 283
Guarino, Stefano 409
Guichon, Joannès 444
Guzzi, Pietro Hiram 114

H

Hashemi, Farnoosh 49
Hauseux, Louis 384
Henke, Johnathon 222
Hollósi, Gergely 271
Hossain, Liaquat 359, 371

J

Jabeen, Fakhra 245
Jiang, Jiaojiao 62
Jurakovaite, Otilija 283

K

Kejriwal, Mayank 456
Khanna, Sandeep 234
Kimura, Masahiro 171
Koyutürk, Mehmet 258
Kronfeld, Eldad 397
Kumano, Masahito 171
Kundu, Sukhamay 134
Kundu, Suman 234

L

Lambert, Dayton 435
Lambert, Lixia 435
Larson, Jennifer M. 28
Lestas, Ioannis 159
Lewis, Janet I. 28
Li, Mengzhen 258

Loporchio, Matteo 344

Lu, Zehao 147

M

Ma, Boqian 62

Maesa, Damiano Di Francesco 344

Maji, Giridhar 309

Malta, Mariana Curado 309

Mastrostefano, Enrico 409

Meara, Ellen R. 194

Mehta, Dinesh 222

Meshcheryakova, Natalia 321

Metze, Tamara 147

Mitchell, Lewis 468

Miyazaki, Takumi 37

Moen, Erika L. 194

Morden, Nancy E. 194

N

Naacke, Hubert 332

Nanavati, Amit Anil 124, 134

Nanavati, Praharsh 124

Nandi, Suman 309

O

O'Malley, A. James 194

P

Paliouras, Georgios 75

Palla, Luca 183

Papastaikoudis, Ioannis 159

Pozo, Roldan 89

Q

Qolomany, Basheer 371

R

Rahimi, Hamed 332

Ran, Xin 194

Rebek, Violet 435

Ren, Hao 62

Ren, Xiao-Long 147

Ricci, Laura 344

Rockmore, Daniel N. 194

Roelofsma, Peter H. M. P. 245

Rossetti, Giulio 183

Roughan, Matthew 468

Roy, Arkaprava 114

S

Saucan, Emil 397

Schoeneman, John 435

Seba, Hamida 296

Sebahi, Yassine 245

Shvydun, Sergey 321

Sustrico, Martina 183

T

Taal, H. Rob 245

Togni, Olivier 296

Torre, Davide 409

Treur, Jan 245

U

Uga, Keisuke 171

V

Varga, Pál 271

Veltri, Pierangelo 114

Vigna, Sebastiano 102

W

Wang, Shihan 147

Y

Yassin, Ali 296

Z

Zerubia, Josiane 384