Department of Computer Science

Data science for social good

Theory and applications in epidemics, polarization, and fair clustering

Han Xiao

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 118/2020

Data science for social good

Theory and applications in epidemics, polarization, and fair clustering

Han Xiao

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, via remote technology, on 25 September 2020 at 15:00.

Aalto University
School of Science
Department of Computer Science
Data Mining group

Supervising professors
Professor, Aristides Gionis,
KTH Royal Institute of Technology, Sweden.

Professor Aristides Gionis is also an Adjunct Professor at Aalto University, Finland.

Preliminary examiners
Associate Professor, Hanghang Tong,
University of Illinois at Urbana-Champaign, United States.

Assistant Professor, Lorenzo Orecchia,
University of Chicago, United States.

Opponents
Assistant Professor, Danai Koutra,
University of Michigan, United States.

Aalto University publication series
DOCTORAL DISSERTATIONS 118/2020

ISBN 978-952-60-3989-3 (printed)
ISBN 978-952-60-3990-9 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-3990-9

Unigrafia Oy
Helsinki 2020

Finland

Printed matter
4041-0619

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Han Xiao

Name of the doctoral dissertation
Data science for social good

Publisher School of Science

Unit Department of Computer Science

Series Aalto University publication series DOCTORAL DISSERTATIONS 118/2020

Field of research Computer Science

Manuscript submitted 24 April 2020 **Date of the defence** 25 September 2020

Permission for public defence granted (date) 26 June 2020 **Language** English

☐ **Monograph** ☒ **Article dissertation** ☐ **Essay dissertation**

Abstract

Technical innovations have transformed our lives fundamentally, in both positive and negative ways. In this thesis, we look at the negative side. We identify three problems to tackle, namely epidemics, online polarization, and bias in automatic decision-making processes, and approach them using data-driven approaches.

Thanks to globalization, our world is more interconnected than before. While trade and exchange of ideas are happening at an unprecedented rate, the rapid spread of disease is happening globally, as evidenced by the pandemic of COVID-19. To contain epidemics effectively, it is crucial to identify as many infected persons as possible. In practice, however, it is almost impossible to obtain the complete information of who is infected. We study this challenge in the context of social networks, where a disease spreads via network edges. Specifically, we assume only a subset of all infections is observed and we seek to infer who else is infected. Furthermore, we consider two different settings: (1) temporal setting, in which infection time is also observed and, (2) probabilistic setting, in which infection probability of each individual is produced.

Social-media platforms enable people to share and access information easily. Meanwhile, flawed designs in these platforms contribute to the formation of online polarization. As a result, people are unlikely to adopt new ideas that differ from their beliefs, which finally leads to a polarized society. To tackle online polarization, we argue that it is important to discover who is involved in the polarization. We consider a problem setting under social networks, in which the interaction between two persons is either friendly or antagonistic. Furthermore, given some seed nodes that represent different sides of a polarized subgraph, we seek to find the polarized subgraph that is relevant to the seeds. Finding such structures can be used to understand the nature of polarization, and to mitigate the degree of polarization.

Machine-learning algorithms allow the automation of many decision-making processes, for example, deciding whether to grant a loan to a loan applicant. However, unfair results that favor one demographic group (e.g., male) over another (e.g., female) are witnessed. The unfair outcomes may further affect the well-being of the mistreated groups. In this thesis, we focus on the task of data clustering, which has applications in infrastructure design and online social media. We discuss potential fairness issues in existing clustering algorithms that are designed to be fair. As a result, we propose a new fair clustering formulation that captures a novel fairness notion.

For all proposed problems, we study their complexity and design algorithms whose theoretical performance is analyzed. We evaluate all proposed algorithms' efficacy in both synthetic and real-world settings.

Keywords data mining, graph mining, social network analysis, epidemics, fairness, online polarization, algorithm design, approximation algorithm

ISBN (printed) 978-952-60-3989-3 **ISBN (pdf)** 978-952-60-3990-9

ISSN (printed) 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2020

Pages 136 **urn** http://urn.fi/URN:ISBN:978-952-60-3990-9

Preface

Four years of doctoral study was a challenging and fulfilling journey. Despite the hardship, I am a lucky person, for the love and support that I received.

In particular, I owe my graduation to my supervisor, Aris Gionis. Before becoming his student, I had never even imagined being supervised by such a wonderful person. He introduced me to computer science research, which turned out to be so interesting and exciting. Because of his mentoring, I overcame my math phobia developed since I was young and eventually learned to appreciate the beauty of math. He also gave me much freedom in exploring myself and the unknown world. Yet, his constructive feedback is essential for turning my efforts into papers. All in all, I am very honored to be his student!

I am also very grateful to my coauthors, namely, Polina Rozenshtein, Nikolaj Tatti, Çigdem Aslay, and Bruno Ordozgoiti. Working with them was extremely rewarding and delightful. Additionally, I should thank Rohit Babbar, Sujay Khandagale, Kiran Garimella, Fan Yang, works with whom are not included in this thesis, but have definitely expanded my understanding of research. Furthermore, I am grateful to the pre-examiners of this thesis, Hanghang Tong and Lorenzo Orecchia, for their extremely valuable and supportive feedback on the manuscript.

During my doctoral study, I had the fortune to expand my research network outside Aalto University. I thank my mentors and friends that I met during two memorable internships: Francesco Bonchi, Edoardo Galimberti, Lorenzo Severini from ISI Foundation, and Daniel Haimovich, Nimrod Priell from Facebook, UK.

Studying at Aalto University has been an enjoyable experience, mainly because of my colleagues and friends there. These people include but are not limited to: Suhas Muniyappa, Nguyen Tran, Antonis Matakos, Kiran Garimella, Eric Malmi, Guangyi Zhang, Esther Galbrun, Sijing Tu, Seyoung Park, and Michael Mathioudakis. Thank you for the funny chats, random thoughts, foosball games, and many more. Further, I thank my dear friends that I met in Finland: Kai Wang, Yangjun Wang, Joseph

Sakaya, and Yao Lu, for their kindness and help. I should also thank our department and our university for making things so organized and for really caring about doctoral students.

Rolling back further in time, I realize that the people I should thank the most are my parents. I feel blessed for growing up in such a loving and supportive family. Because of your love, I can pursue my dream and live the life that I want. Your kindness, hard-working, and passion for life have affected me so fundamentally. Additionally, I would like to express my gratitude to my parents-in-law, for their love and support.

Finally, to my beloved daughter, Luxi. Probably by the time you can read this, you cannot remember anything about your dad's happy life during his doctoral study. Thank you for being such an adorable girl, and for brightening your dad's life. To my beloved one, Qing, thank you for being another adorable girl and my wonderful wife at the same time. I am grateful that you chose me despite my many imperfections. Your love and understanding raise me up. To Luxi and Qing, my life has new meaning because of you!

Jyväskylä, Finland, August 11, 2020,

Han Xiao

Contents

List of Publications

This thesis consists of an overview and of the following publications, which are referred to in the text by their Roman numerals.

I Xiao, Han and Rozenshtein, Polina and Tatti, Nikolaj and Gionis, Aristides. Reconstructing a cascade from temporal observations. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 666–674, May 2018.

II Xiao, Han and Aslay, Çigdem and Gionis, Aristides. Robust cascade reconstruction by Steiner tree sampling. In *2018 IEEE International Conference on Data Mining*, pages 637–646, November 2018.

III Xiao, Han and Ordozgoiti, Bruno and Gionis, Aristides. Searching for polarization in signed graphs: a local spectral approach. In *The World Wide Web Conference*, pages 362–372, April 2020.

IV Xiao, Han and Ordozgoiti, Bruno and Gionis, Aristides. A distance-based approach to fair clustering. Submitted for publication, July 2020.

Author's Contribution

Publication I: "Reconstructing a cascade from temporal observations"

The author initiated the problem setting, and contributed to the problem formulation and paper writing. He designed one of the proposed algorithms, implemented all proposed algorithms, and conducted the experiments. He was responsible for writing the experiment part in the paper.

Publication II: "Robust cascade reconstruction by Steiner tree sampling"

The author proposed the problem, designed the algorithms, and contributed to the theoretical analysis of the algorithms. He implemented all proposed algorithms and conducted the experimental evaluation. When writing the paper, he was responsible for the experiment part and part of the theoretical analysis. He also contributed to improving the paper's overall quality.

Publication III: "Searching for polarization in signed graphs: a local spectral approach"

The author proposed the problem, designed and implemented the algorithm, and conducted the experiments. He did most of the theoretical analysis and paper writing.

Publication IV: "A distance-based approach to fair clustering"

The author initiated the problem setting and contributed to the problem formulation. He designed one of the two proposed algorithms and analyzed the theoretical performance of both algorithms. He implemented both algorithms and conducted the experimental evaluation. For paper writing, he was responsible for the technical part and the experiment part. He also contributed to improving the paper's overall quality.

1. Introduction

1.1 Background and motivation

What distinguishes us, humans, from many other species on earth is our capacity for complex thinking and reasoning. These abilities have enabled us to develop innovations that improve our life. As such, progress, in a variety of indices, has largely been the main theme in human history. In what is known as "The Great Acceleration" [81], a period from 1750 and onwards, humanity has undergone unprecedented progress in many aspects of our life and our society. For instance, we have a much longer lifespan on average than our prehistoric ancestors, because of the advances in health care and medical science. On the societal level, the idea of racial and gender equality is commonly accepted after many years of reflection and reasoning. Traveling across continents, which was once considered adventurous and could take years, has now become a commonplace service, which is easily accessible and safe.

In parallel to the betterment of our lives, our impact on this planet has become so significant that a new geological epoch known as "Anthropocene" was proposed [80]. While it is still early to assess whether our impacts are benign or not, some of them are already creating new problems. Increasing levels of human activity are placing greater pressure on the global ecological system. Climate change and environmental pollution are threatening the survival of many species that have lived on this planet for millions of years.

To make things worse, our prospective to survive and flourish is facing increasing uncertainty due to the negative consequences made by us. For instance, increasing sea level due to the warming climate will ultimately take away the homes of millions. On our warming planet, large numbers of wild fires with great scale have been reported, which not only hollow the habitats of living creatures, but also undermine the already weak ecosystem. Apart from the problems in the physical world which we

live in, we are experiencing new challenges in the online sphere. Online polarization and misinformation [84] have generated large-scale conflicts and hatred, which have hindered the proper functioning of our societies.

As the eminent psychologist George Miller wrote in 1969 [60], "the most urgent problems of our world today are the problems we have made for ourselves. They are human problems whose solutions will require us to change our behavior and our social institutions." To put it straightforwardly, we messed things up and we are the only ones who can fix the situation.

To address the problems created by us, in this thesis we focus on three categories of problems: *epidemics & infodemics*, *online polarization*, and *bias in automatic decision-making*. All three problems can be viewed as the byproducts from the development of information technology, and are typical challenges in our digitized society. The last problem category is grounded in the physical world, but is still closely related to the online world.

Next we discuss each problem category in greater detail and identify several key issues to address.

Epidemics and infodemics. A part of human history is the long-lasting struggle with epidemics. The Black Death was one of the most devastating pandemics in human history, killing 30% to 60% of Europe's population in the Middle Ages [87]. With the advances in medical and biological science, many infectious diseases are not as deadly as before. However, a few factors, such as increasing population density and higher levels of human mobility, pose new challenges for epidemic prevention in recent days. When no effective vaccination against a new disease is available, prevention becomes critical but challenging. The outbreak of *COVID-19* caused by a novel coronavirus (namely *SARS-CoV-2*) at the end of 2019 has caused 15 581 009 confirmed infections and claimed 635 173 lives until July 26rd, 2020.[1] Yet, there has been no sign of a global turning point by the time of writing this document.

To contain an epidemic effectively, early identification of infected persons and timely quarantine are crucial. However, obtaining relevant information, such as who are infected, is difficult in practice. On the one hand, information is usually noisy. For example, the symptoms of COVID-19 are similar to that of common influenza, thus influenza patients can be mistakenly diagnosed as COVID-19 patients. On the other hand, identifying individuals who are infected but without any symptoms requires testing people at a massive scale, which is often unrealistic.[2]

We should not feel hopeless though. A variety of information sources that can be harnessed bring new opportunities. With the help of the "right" data, we might be able to identify the patients effectively and

[1] https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200725-covid-19-sitrep-187.pdf

[2] https://www.nytimes.com/2020/02/26/health/coronavirus-asymptomatic.html

efficiently. For example, personal mobility records (e.g., collected from mobile devices) provide information on where people have been to. Based on this information, we can further infer who have contacted whom, thus constructing a human-contact network, and further use this network to make the inference.

Important questions remain. Based on the constructed network and the identities of a few infected individuals, how can we identify the remaining infected population? Furthermore, how can we reliably tell the starting point of an epidemic and when each infected individual got infected?

Another phenomenon closely related to epidemics is *infodemics*. An infodemic refers to an epidemic of *misinformation* — false or inaccurate information, especially those that are deliberately intended to deceive. In fact, during the COVID-19 outbreak, a global epidemic of misinformation, which spread rapidly through social-media platforms, is happening. As a consequence, infodemics, which contain misleading and even malicious information, are posing a serious problem for public health [91].

If social-media users who are hoaxed to believe a piece of misinformation and may share the information to his friends, who might also believe the misinformation, it is reasonable to make an analogy of infodemics using epidemics. The misinformation that disseminates can be viewed as a "disease" that spreads among the users, and the users who believe the misinformation can be seen as the individuals that are "infected." Thus, similar questions can be asked in the context of infodemics: *how can we identify the starting points of an infodemic as well as which users are hoaxed to believe the misinformation?*

Online polarization in social media. Our world is increasingly interconnected thanks to globalization and technological advances. The wide adoption of social media has been a driving force that makes such interconnection happen. Social-media platforms have enabled people, with different nationalities and diverse beliefs, to communicate and exchange ideas.

While social media makes it easy for us to publish and access information, it is not necessarily leading to greater cohesion of our world. As evidence, researchers point out that social-media platforms are playing a significant roles in the formulation of *polarization* [25, 35, 65]. These researchers also suggest that a few factors such as social homophily (e.g., social network structure) and algorithmic filtering (e.g., news feed recommendation) have narrowed down the diversity of the information sources that users digest. The lack of diversity accelerates the formation of online filter bubbles, where users only consume content that agrees with their beliefs and ideology. As a consequence, people are divided by opposite views, and thinking from different perspectives becomes hard. This situation ultimately leads to a polarized society.

We are concerned about the harmful effects of polarization, therefore

we ask: *what can we do to ameliorate the extent of polarization?* One way to tackle this question is through a *data-driven approach*. Social-media platforms are typically rich in structural information among the users, such as which user friends whom. Such relational information can be encoded using a mathematical model called *networks* (also known as *graphs*). The model of networks and its associated mathematical tools can help us gain a deeper understanding of the nature of polarization. Furthermore, insights obtained from the mathematical analysis can assist experts in applying justified interventions to reduce polarization.

As a concrete example, consider a scenario in which a polarized debate emerges around a topic discussed in an online social-media platform A natural question to ask is: *can we find groups of users who demonstrate polarization?* That is, we seek to find user groups, in which each group represents supporters of one opinion about a topic and different groups hold contrasting opinions.

Being able to find such structures can be useful for computational social scientists, who are interested in understanding the structure of polarization and the mechanics of their discussions. In addition, finding the polarized groups can be of interest to the social-media platform, if they wish to reduce the degree of polarization, e.g., by recommending a thought-provoking article to the users involved in the argument.

Bias in algorithmic decision-making. It is widely believed that software and algorithms that rely on data are objective.[3] However, human's influence on software and algorithms is unavoidable. For example, software is designed, written, and maintained by humans. The behaviour of automatic decision systems can be adjusted according to human needs. Also, the data they rely on may be curated manually. As a result, algorithms can reflect and even reinforce human bias.

In the United States, for instance, a piece of software is used across the country to predict the future risk of recidivism. However, researchers have found that the software is biased against blacks.[4] This example, which demonstrates the unfair aspect of software systems, is not an isolated one. A study by Datta et al. [24] shows Google's online advertising system presents high-income job ads much more often to men than to women.

With the wide adoption of machine-learning techniques in everyday life, it is crucial to ensure that algorithms are fair. This requirement has been recognized by researchers and the topic of fairness has recently been brought into attention.

An active line of research in fairness in machine learning is about *classification* problems. A classification algorithm automatically classifies

[3] https://www.nytimes.com/2015/07/10/upshot/when-algorithms-discriminate.html

[4] https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

an "object" to a pre-specified set of "labels". An example is determining whether a loan applicant (the object to be classified) should be granted a loan or not (the labels), based on the applicant's background information. One topic in this line of research is to codify what it means for a classification algorithm to be fair. A variety of notions have been proposed and studied. For instance, *statistical parity* states that subjects in different protected groups (e.g., racial groups) should have an equal probability of receiving positive outcomes (e.g., being granted a loan). Another topic in this line of research is to design classification algorithms that achieve a certain fairness notion.

Data clustering, another important topic in machine learning, studies how to group a set of objects in meaningful ways. Data clustering has many important applications such as infrastructure design [26], market research [15], exploratory data analysis [29], and improving the performance of classification algorithms [78]. Despite its importance, it has received less attention in the study of algorithmic fairness, compared to classification.

Most of the existing works in clustering with fairness [1, 5, 21] put their focus on a single type of fairness notion. Failing to study other fairness notions in clustering limits the usability of fair clustering methods. Therefore we ask the following questions: for clustering problems, can we identify the scenarios where current fairness notions are not meaningful? If yes, can we formalize fairness notions that are meaningful in these scenarios? Further, can we design clustering algorithms that achieve such fairness notions?

1.2 Thesis contribution

The main contributions of this thesis are summarized below:

- In Publications I and II, we consider the problem of reconstructing epidemic processes given partial observations over the epidemics. In Publication I, we assume that observations contain *temporal information*, and propose a novel problem formulation. Also, we develop a set of combinatorial approximation algorithms. In Publication II, we study the reconstruction problem in a *probabilistic* setting and resorts to Monte-Carlo simulation to reconstruct cascades. We propose two sampling algorithms with provable theoretical guarantees. For both publications, we conducted extensive experimental evaluations to demonstrate the efficacy of our proposed methods.

- In Publication III, we tackle the problem of polarization detection in social networks. Given some *seed nodes* as the input, we are asked to find a *polarized community* that is relevant to the input seed nodes.

We propose an algorithm that is both effective in finding high-quality communities and efficient in handling large graphs.

- In Submitted Article IV, we consider the issue of algorithmic fairness in the context of data clustering, and propose a new notion of fairness. We develop two families of algorithms, both with approximation guarantees. In our empirical evaluation, we demonstrate that our algorithms achieve the proposed notion of fairness and yield near-optimal solutions.

Most of the source code and datasets (except for the code of Submitted Article IV) are publicly available at `https://github.com/xiaohan2012/`.

1.3 Thesis organization

This thesis follows the format of an article-based dissertation, meaning that it consists of a set of articles, appended at the end of the thesis, and a compilation part (Chapters 1 to 6) that summarizes the studied problems and the main findings of the articles.

The compilation part is organized as follows: In Chapter 2, we equip the readers with the necessary mathematical concepts and notations to understand the subsequent chapters. In Chapter 3 (based on Publication I and Publication II), we consider the problem of reconstructing partially-observed epidemic processes, under two different settings. In the former setting (Publication I), temporal information is available and is incorporated into the inference algorithm. In the latter setting (Publication II), we study probabilistic inference on individuals' infection status. In Chapter 4 (based on Publication III), we present a method to find polarized communities in social networks. In Chapter 5 (based on Submitted Article IV), we discuss fairness issues in data clustering problems, and develop a new fairness notion as well as two novel algorithms that achieve the proposed fairness notion. Finally, we draw conclusions in Chapter 6.

The articles are appended in temporal order of their submission dates and all of them are conference articles.

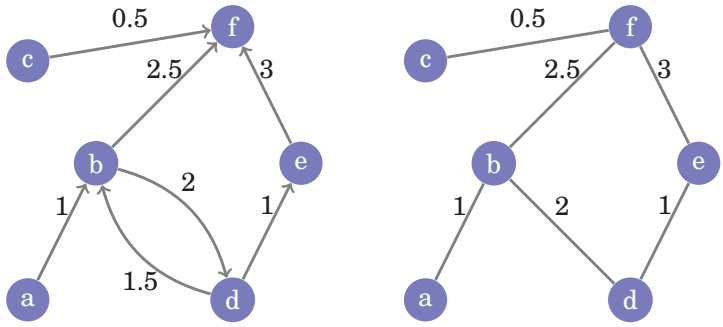
2. Preliminaries

2.1 Graphs

A graph is a data structure that captures pairwise relations between objects. A graph usually consists of two parts: a set of *vertices* and a set of *edges* that connect pairs of vertices. Vertices are also called *nodes*, and we use these two terms interchangeably. Graphs have been used to model a variety of real-world phenomena. For instance, *social networks* are naturally represented as graphs, in which vertices represent people and there is an edge between two vertices if the corresponding people are friends to each other. In road networks, vertices represent different cities and edges represent the roads that connect the corresponding cities. In biology research, a *protein-protein interaction network* models the interaction between protein complexes. Two vertices are connected if the corresponding proteins complexes are involved in the same biochemical process. In computer networks, vertices represent computers and there is an edge between two vertices if the corresponding computers can directly communicate with each other.

In many situations, edges are associated with *weights* (e.g., real-valued numbers). A graph with edge weights is called *weighted*. Otherwise, it is called *unweighted*. Unweighted graphs can be seen as a special case of weighted graphs in which all edge weights are unit (of value 1).

Edge weights usually carry specific meaning. For instance, they are used to indicate the "strength" of the associated edges. In social networks, an edge has a large weight if the corresponding connecting persons are very close friends. In a protein-protein interaction network, the weight of an edge may correspond to the activity level of the biochemical reaction involving the two corresponding protein complexes. In some other cases, the weights indicate the "cost" of edges. In computer networks, the communication speed among different machines may vary and thus, can be represented by edge weights. Similarly, edges in road networks can reflect



(a) directed (b) undirected

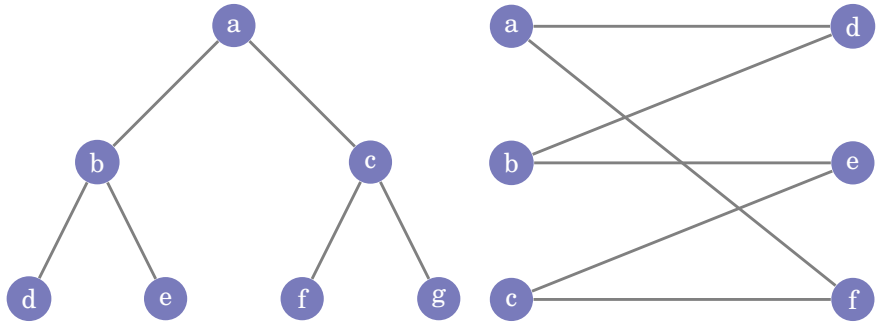
Figure 2.1. Illustration of a directed graph and an undirected one. Both graphs are weighted.

the transportation time between two cities.

Sometimes edge directions are incorporated to capture the asymmetric relations among the vertices. Such edges are called *directed edges*. Graphs that distinguish edge directions are called *directed graphs*. Many natural graphs are directed. For instance, some roads in a road network allow vehicles to travel in only one direction. Similarly in online social media, a user may follow another high-profile user but not the other way around. *Undirected edges*, on the other hand, refer to edges without direction. An undirected edge can be decomposed into two directed edges between the same endpoints but with different directions. From this perspective, directed graphs can be seen as a generalization of the undirected counterparts.

Mathematically, we use the symbol G to denote a graph and a graph is defined as $G = (V, E, w)$, where V and E are the set of vertices and edges in the graph, respectively, and edge weights w maps each element in E to some real number. If G is unweighted, we write $G = (V, E)$ for brevity. We use lower-case v and e to denote a vertex and an edge, respectively. Sometimes we denote an edge by its two endpoints, i.e., $e = (u, v)$, where u and v are the two endpoints of an edge e. For a directed edge, (u, v) represents an edge pointing from u to v. An undirected edge (u, v) is equivalent to (v, u).

We give examples of a directed graph and an undirected graph in Figure 2.1. In both examples, the sets of vertices are the same, $V = \{a, b, c, d, e, f\}$, and edge weights are shown beside the corresponding edges. In Figure 2.1(a) where edges are directed, the edges are $E = \{(a, b), (b, d), (d, b), (d, e), (b, f), (e, f), (c, f)\}$. In Figure 2.1(b) where edges are undirected, the edges are $E = \{(a, b), (b, d), (d, e), (b, f), (e, f), (c, f)\}$.



(a) a tree (b) a bipartite graph

Figure 2.2. Illustration of a directed tree (left) and a directed bipartite graph (right)

2.1.1 Special structures

Several structures in graphs are of particular interest. We consider a few examples next.

Paths. A *path* in a graph is a sequence of vertices $(v_1, v_2, \ldots, v_k)$ such that for $i = 1, \ldots, k-1$, (v_i, v_{i+1}) corresponds to an edge in the graph. For example in Figure 2.1a, in which the graph is directed, (a, b, f) is a path. The sequence (f, b, a), however, is not a path because neither (f, b) nor (b, a) is an edge. In Figure 2.1b, in which the graph is undirected, both (a, b, f) and (f, b, a) are paths. In addition, for any path $(v_1, v_2, \ldots, v_k)$ in which the edges are undirected, we consider the path the same as $(v_k, v_{k-1}, \ldots, v_1)$.

Trees. A tree is an undirected graph in which any two vertices are connected by exactly *one path*. We give an example of trees in Figure 2.2a. We can verify the definition of trees using this example. For instance, e and f are connected only by one path – (e, b, a, c, f). In addition, we define *leaves* as vertices that have only one adjacent edge. In this example, the leaves are $\{d, e, f, g\}$.

Trees can be used to represent many real-world structures. For example, personnel in organizations (companies, universities, etc.) are sometimes organized as trees. Edges represent subordinate relations among the personnel (e.g., a team leader manages a team member). Biologists organize living organisms into a tree structure. At the top level, all living organisms are partitioned into several *kingdoms* (e.g., animals, plants, fungi, etc) and each kingdom is further partitioned into smaller groups, for instance, the animal kingdom consists of vertebrates and invertebrates, etc. In many computer operating systems, files are arranged into directories (also called folders) and directories can be further organized into sub-directories.

Bipartite graphs. A *bipartite graph* contains vertices that can be partitioned into two disjoint sets, such that vertices in each set are disconnected from each other. We give an example in Figure 2.2b, in which the vertices

consists of $V_1 = \{a, b, c\}$ and $V_2 = \{d, e, f\}$, and there is no edge between any pair of nodes in neither V_1 nor V_2.

Bipartite graphs are suitable to model *affiliation relations* where usually two types of vertices are involved. For instance, in a graph of football players and clubs, there is an edge between two vertices if the corresponding player is playing for the corresponding club. The node types (player or club) naturally indicates a vertex bi-partition. As another example, customer purchase records, in which customers and products are involved, can be modeled as bipartite graphs. We can represent all customers as one vertex set and all products as another. If a customer has purchased a product before, we draw an edge between the corresponding vertices.

2.1.2 Beyond structural information

Besides vertex and edge information, other types of information are sometimes present in graph data. For instance, online social networks usually contain meta-information about users, e.g., their posts and demographic background. The edge information can be enriched when there is some interaction between the corresponding users, for example, a user's comments under another user's posts or a video that the two users watched together.

Below we give a few examples of graphs that contain not only vertex-and-edge information.

Temporal graphs. *Temporal graphs*, also known as *time-varying graphs*, dynamically change their structures over time. Each edge is associated with an active period, in which it only exists. This characteristic enables temporal graphs to model many real-world phenomena that plain graphs (without temporal information) cannot. In a phone-call network, for instance, an edge represents a phone call between two persons and the time interval of the edge is the time interval of the phone call. In a road network, roads connecting different cities may be cut off during an earthquake, in which case the corresponding edges are removed. An airline company may add a direct flight between two cities that are not connected before, thus creating new edges between the corresponding vertices.

Signed graphs. In a *signed graph*, every edge is labeled either *positive* or *negative*. In social networks, edge signs indicate whether the interactions between the corresponding users are friendly (positive sign) or antagonistic (negative sign). In protein-protein interaction networks, there exists a positive (or negative) edge if the corresponding protein activates (or inhibits) the functioning of another protein. In Chapter 4, we dive deeper into signed graphs.

2.2 Spectral graph theory

Spectral graph theory studies the characteristics of the eigenvalues and eigenvectors of related matrices of a graph, in relation to the properties of the graph. In this section, we focus on this theory's application in *graph partitioning*.

2.2.1 Laplacian matrix

The *Laplacian matrices* of a graph are central building blocks in the study of spectral graph theory. In particular, Laplacian matrices are closely related to the task of graph partitioning – partitioning all the vertices in a graph into disjoint subsets, such that some quality function over the partition is optimized.

Given an undirected graph $G = (V, E, w)$ with n vertices, m edges and non-negative weights w, we denote $A \in \mathbb{R}^{n \times n}$ as the *adjacency matrix* of G:

$$A_{i,j} = \begin{cases} w(i,j) & \text{if } (i,j) \in E \\ 0 & \text{otherwise,} \end{cases} \tag{2.1}$$

where $A_{i,j}$ is the value of jth entry in the ith row of the matrix A.

Further we use $D \in \mathbb{R}^{n \times n}$ to denote the *degree matrix* of G. The matrix D is a diagonal matrix, with the diagonal entries equal to $D_{i,i} = \sum_{j=1}^{n} A_{i,j}$ for $i = 1, \ldots, n$.

The *unnormalized Laplacian matrix* of G is defined by

$$L = D - A. \tag{2.2}$$

For any vector $x \in \mathbb{R}^n$, the term $x^T L x$ is equivalent to

$$x^T L x = \frac{1}{2} \sum_{i,j=1}^{n} w(i,j)(x_i - x_j)^2. \tag{2.3}$$

It follows that $x^T L x \geq 0$ and by definition, the matrix L is positive semi-definite.

Using the positive semi-definiteness of L, it follows that the eigenvalues of L is non-negative. Let us denote the eigenvalues of L as $\lambda_1, \ldots, \lambda_n$, sorted in ascending order. Let $x_1, \ldots, x_n$ be the corresponding eigenvectors.

According to the Courant-Fischer Theorem, $\lambda_1, \ldots, \lambda_n$ have the following equivalent forms:

$$\lambda_1 = \min_{x \neq 0} \frac{x^T L x}{x^T x},$$

$$\lambda_2 = \min_{\substack{x \neq 0 \\ x^T x_1 = 0}} \frac{x^T L x}{x^T x},$$

$$\lambda_3 = \min_{\substack{x \neq 0 \\ x^T x_1 = 0 \\ x^T x_2 = 0}} \frac{x^T L x}{x^T x},$$

$$\ldots$$

In other words, the kth eigenvector is the non-zero vector that minimizes $\frac{x^T L x}{x^T x}$ and is orthogonal to the previous $k-1$ eigenvectors.

The eigenvalues and eigenvectors of L of a graph can be used to describe many properties of the graph. One result relates to the connected components in G, which is discussed next.

Eigenvectors of L and the connected components in G. Let us first consider the case G is *connected*, that is, there exists a path between any pair of non-identical vertices in the graph. It can be seen from Equation 2.3 that constant vector, i.e., $x = [1,\ldots,1]$ is an eigenvector of L. Furthermore, the associated eigenvalue is zero. In other words, for any connected graph we have that:

$$x_1 = [1,\ldots,1] \text{ and } \lambda_1 = 0.$$

Then consider the case that G has k *connected components*, that is, the graph can be partitioned into k subgraphs such that each subgraph is connected within itself but disconnected with any other subgraphs. Without loss of generality, the unnormalized Laplacian has a *block-diagonal* structure shown below:

$$L = \begin{pmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_k \end{pmatrix}, \tag{2.4}$$

where $L_1,\ldots,L_k$ correspond to the unnormalized Laplacian matrices of the k connected components in G, respectively. For the ith component C_i, we can define a indicator vector $\mathbb{1}_{C_i}$:

$$\mathbb{1}_{C_i}(j) = \begin{cases} 1 & \text{if } j \in C_i \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

It can be shown that, for every $i = 1,\ldots,k$, $\mathbb{1}_{C_i}$ is an eigenvector of L and the associated eigenvalue is zero.

Therefore we have the following result:

Fact 1. *Let G be an undirected graph with non-negative weights. Then the multiplicity k of the eigenvalue 0 of L equals the number of connected components $C_1,\ldots,C_k$ in the graph. The corresponding eigenvectors are k indicator vectors $\mathbb{1}_{C_1},\ldots,\mathbb{1}_{C_k}$, corresponding to $C_1,\ldots,C_k$, respectively.*

The above result can be used to extract the connected components in a graph where each connected component corresponds to a unique eigenvector. In the context of social networks, each connected component can be thought of as the so-called *community* – a group of users that are connected with each other.

However, community structures in real-world graphs are *noisy*, that is, two communities, which are densely connected inside each, can be connected by a few edges in between. Such edges further make the two communities into one connected component. For this reason, it becomes less clear how to use the eigenvectors of L to partition a graph into communities.

Next we show that, with the help of another type of Laplacian matrix, we can address the issue posed by noisy edges and use the eigenvectors of the new Laplacian to partition a graph.

2.2.2 2-way partitioning and Cheeger's inequality

Given an undirected graph $G = (V, E)$, we consider the problem of 2-way graph partitioning, that is, partitioning V into two disjoint sets S and $\bar{S}$, where $\bar{S} = V \setminus S$. For simplicity, we assume G is unweighted.

One reasonable objective function for this purpose is counting the number of edges between S and $\bar{S}$. The problem becomes the *min-cut* problem, which is solvable in polynomial time. A drawback of this formulation is that the partition sizes can be very unbalanced. For instance, assume that G is connected and there exists a vertex v that has only one adjacent edge. An optimal solution is $S = \{v\}$ and $\bar{S} = V - \{v\}$, and the cost is 1. Arguably, such partitions, whose sizes are very imbalanced, convey little useful information about the graph structure.

Using conductance as the objective function is one way to address the issue of unbalanced partitions. First we use *degree* of a vertex i, $\deg(i)$, to denote the number of edges adjacent to the vertex i. Then the *conductance* of a vertex set S is defined as

$$\phi(S) = \frac{cut(S, \bar{S})}{\min\left\{vol(S), vol(\bar{S})\right\}}, \tag{2.6}$$

where $cut(S, \bar{S})$ is the number of edges between S and $\bar{S}$ and $vol(S) = \sum_{i \in S} \deg(i)$. Compared to using only $cut(S, \bar{S})$, which is what the min-cut formulation uses, conductance penalizes imbalanced partition sizes using the denominator in Eq. 2.6.

In our new problem formulation, we seek to find a vertex subset S that minimizes $\phi(S)$. Note that the set S automatically leads to a partition $(S, \bar{S})$.

Further we define the *graph conductance* of G as

$$\phi(G) = \min_{S \subset V} \phi(S), \tag{2.7}$$

that is, the cost of the optimal solution on G. If the graph G has a low conductance value, it roughly means G has a clear community structure, that is, the graph can be partitioned into two communities that are densely-connected within each and sparsely-connected in between.

The conductance of a set S can be related to the Laplacian matrix, which we illustrate next.

Normalized Laplacian. Given a node set S, we use an indicator vector $x = \mathbb{1}_S$ to represent S. By Equation 2.3, we have:

$$\frac{cut(S,\bar{S})}{vol(S)} = \frac{2\,x^T L x}{x^T D x}. \tag{2.8}$$

Note that the left-hand-side term is related to $\phi(S)$ (Equation 2.6) as follows:

$$\phi(S) = \max\left\{\frac{cut(S,\bar{S})}{vol(S)}, \frac{cut(S,\bar{S})}{vol(\bar{S})}\right\}.$$

If we use another vector $y = D^{1/2}x$, we have:

$$\frac{x^T L x}{x^T D x} = \frac{y^T D^{-1/2} L D^{-1/2} y}{y^T y}. \tag{2.9}$$

The matrix in the middle is called the *normalized Laplacian*:

$$\mathscr{L} = D^{-1/2} L D^{-1/2}. \tag{2.10}$$

Next we show how the eigenvector associated with the second smallest eigenvalue of $\mathscr{L}$ can be used to find a good partition. This intuition is formalized by *Cheeger's inequality*.

Cheeger's inequality. The results in Equation 2.8 and Equation2.9 say that the problem of finding a partition $(S,\bar{S})$ that minimizes $\phi(S)$ is equivalent to finding a vector y that minimizes

$$\frac{y^T \mathscr{L} y}{y^T y},$$

subject to:

$$y_i = \begin{cases} \frac{1}{\sqrt{\deg(i)}} & \text{if } i \in S \\ 0 & \text{otherwise.} \end{cases}.$$

Next we relax y to be continuous, i.e., $y \in \mathbb{R}^n$ and this problem is equivalent to finding the eigenvector corresponding to the smallest eigenvalue of $\mathscr{L}$. It can be seen that setting $y = D^{-1/2}\mathbb{1}$ is the solution to the relaxed problem. However, doing so does not give any meaningful partition.

Therefore, we require that y to be orthogonal to $D^{-1/2}\mathbb{1}$, which leads to the problem of finding the eigenvector associated with *second smallest*

eigenvalue of $\mathscr{L}$. Finding this eigenvector can be solved in polynomial time using standard linear algebra software.

Furthermore, denote μ_2 as the second smallest eigenvalue of $\mathscr{L}$ and y_2 as the associated eigenvector, *Cheeger's inequality* [16] states that

$$\frac{\mu_2}{2} \leq \lambda(G) \leq \sqrt{2\mu_2}. \tag{2.11}$$

In other words, μ_2 is an approximation of $\phi(G)$.

Partitioning G by rounding. Last we show how to round y_2, the eigenvector corresponding to the second smallest eigenvalue of $\mathscr{L}$, so that a good partition can be found.

Let $x = D^{-1/2}y_2$, and given some threshold $t \in \mathbb{R}$, define a set $S(t)$ as:

$$S(t) = \{i : x(j) < t\}.$$

Mihail [59] shows that there exists some t such that

$$\phi(S(t)) \leq \sqrt{\frac{y_2^T \mathscr{L} y_2}{y_2^T y_2}}.$$

Note that the right-hand-side term above equals $\sqrt{\mu_2}$. We can find such t by picking the t value that minimizes $\phi(S(t))$.

Combined with the Cheeger's inequality (Relation 2.11), we can find a partition S whose conductance is bounded with respect to the optimal value:

$$\phi(S) \leq 2\sqrt{\phi(G)}.$$

We end our introduction to spectral graph theory here. For a more comprehensive treatment of this topic, one may refer to a book by Chung and Graham [23]. In Chapter 4, we discuss a similar topic, but in the context of signed graphs.

2.3 Data clustering

Data clustering refers to the task of grouping a set of objects, such that objects in every group are similar to each other. Clustering has many real-world applications. In market research, business analysts partition consumers into groups and analyze group-level characteristics [70]. Photos on mobile phones can be clustered so that similar photos appear in the same group [6]. Mobile-phone users may further keep only one of the similar photos to save device storage. Text documents such as news articles are grouped and summarized [66]. Readers can get a quick overview of the documents by selecting a few representative documents from each document group.

To write computer programs that can cluster objects, it is often useful to represent the objects at hand by *numerical values*. An image, for instance, can be represented by pixel matrices, which record color information at each pixel. *Vector representation* is another commonly-used representation form. A common way to represent objects in vector forms is by first selecting a set of *features* related to the objects and then defining functions that compute feature values. A text document, for instance, can be viewed as a list of words accompanied by their frequencies. Each word is a feature and its frequency is the feature value. In health-care systems, human subjects' features may include their demographic information (age, gender, etc.) and personal health records (blood pressure, body weight, etc.).

2.3.1 Distance functions

A common assumption made when designing clustering algorithms is that there exists a distance function that computes how dissimilar *any* two objects are.

Consider a vector $v = [v_1, \ldots, v_n]$ of length n, where v_i is the ith element in the vector and n is called the *vector dimension*. A distance function d takes two vectors u and v that have the same dimension, and outputs a dissimilarity score between u and v. It is often desirable for a distance function d to have the properties below. Let X be the set of all vectors that d can accept. For any $u, v, w \in X$:

1. **Non-negativity**: $d(u,v) \geq 0$,

2. **Identity**: $d(u,v) = 0$ if and only if $u = v$,

3. **symmetry**: $d(u,v) = d(v,u)$, and

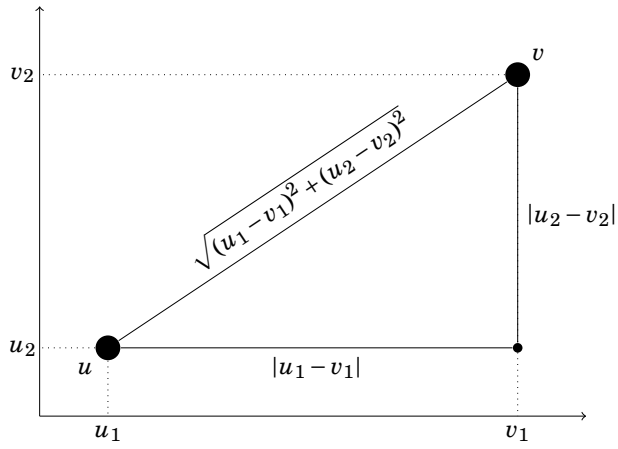
4. **Triangle inequality**: $d(u,v) + d(u,w) \geq d(v,w)$.

A distance function satisfying all these properties is called *a metric*. When designing a distance function, the first three properties are usually easier to achieve than the last one.

Euclidean distance is an example of a metric. Formally it is defined as:

$$\ell_2(u,v) = \sqrt{(v_1 - w_1)^2 + (v_2 - w_2)^2 + \cdots + (v_n - w_n)^2}. \tag{2.12}$$

In the general n-dimensional case, $\ell_2(u,v)$ can be interpreted as the length of the line segment that connects u and v. To see why, consider the special case of $n = 2$ and two vectors u and v shown in the plot below:



In the above plot, the points $u = (u_1, u_2)$ and $v = (v_1, v_2)$ are shown as black solid circles. The hypotenuse is the line segment that connects u and v and its length is equivalent to the Euclidean distance between u and v. It can be verified as follows: the horizontal (vertical) distance between u and v is $|u_1 - v_1|$ ($|u_2 - v_2|$), which corresponds to the horizontal (vertical) solid line segment shown in the plot. Finally, the length of the hypotenuse is $\sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2}$.

2.3.2 Clustering objective functions

Many clustering formulations assume the number of clusters is specified in the input. In the sequel, we use the symbol k to denote this number.

In the general setting, we are given a set of objects C to be clustered, and F, a set of candidate cluster centers that each object in C can be assigned to. A *clustering* (F', ψ) consists of F', which is a subset of F and has size at most k, and an *assignment function* $\psi : C \to F'$, which maps each object to its assigned center in F'. In some applications, the set F represents facilities (e.g., hospitals, bus stations) to be deployed and the set C represents human subjects that use the facilities. In this context, C and F are referred as *clients* and *facilities*, respectively, and the clustering problems are also called *facility-location problems*.

To formalize the goodness of a clustering solution, it is necessary to define an objective function that evaluates the goodness of *any* clustering. We describe two classic objective functions.

The k-median objective. Given a distance function d, the k-median objective over some clustering (F', ψ) measures the *total* distances from all clients to their assigned centers, and it is defined as:

$$\Phi_2(F', \psi) = \sum_{c \in C} d(c, \psi(c)). \tag{2.13}$$

The k-center objective. In contrast, the k-center objective measures the

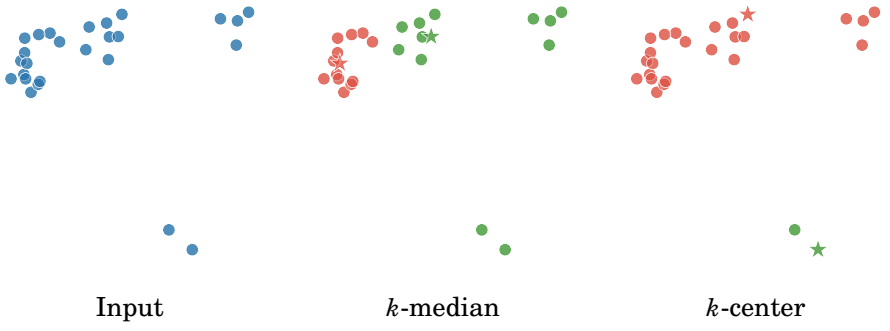


Figure 2.3. Optimal clustering on a toy dataset under different objectives.

maximum distance among all object-center pairs, and it is defined as:

$$\Phi_\infty(F', \psi) = \max_{c \in C} d(c, \psi(c)). \tag{2.14}$$

Once the clustering objective is decided, one then seeks the clustering that achieves the lowest possible objective value.

When there are no other constraints on the problem formulation, both Φ_2 and Φ_∞ can be simplified. That is, ψ is implicitly determined by F' — for all $c \in C$, $\psi(c)$ picks the *closest* center in F' to c. Doing so never increases the solution cost. Then the k-median and k-center objectives depend solely on F':

$$\Phi_2(F') = \sum_{c \in C} \min_{f \in F'} d(c, f), \quad \text{and} \tag{2.15}$$

$$\Phi_\infty(F') = \max_{c \in C} \min_{f \in F'} d(c, f). \tag{2.16}$$

Last we illustrate the difference between these two objectives on a toy dataset. We set $k = 2$ and use the Euclidean distance measure. We consider the special case of $C = F$ (each object is a candidate center).

The optimal clustering under each clustering objective is shown in Figure 2.3. On the left, the input data points are plotted. The plots in the middle and right are the optimal solutions by k-median and k-center, respectively, where the node color indicates the cluster membership. The cluster centers are plotted as stars.

The definition of the k-center objective makes it is sensitive to outliers (for instance, the two points at the bottom), and a center is opened at the bottom. In contrast, the k-median objective, which measures the total distance value, does not have this sensitivity issue.

2.4 Optimization problems and approximation algorithms

Many real-world decision problems can be modeled as *optimization problems*. Companies, for instance, aim for *maximizing profits* when making business plans. When designing structures for buildings and bridges, structural engineers strive for the most endurable and safest design. Logistic systems are designed so that it yields the minimum possible transportation time and cost.

2.4.1 Optimization problems

An optimization problem refers to the problem of finding the *best* solution, according to some specific definition of the goodness of solutions. Clustering problems (described in the previous section) are examples of optimization problems.

Optimization problems can be further categorized by whether the objectives are to be minimized or maximized. If lower objective values are better (e.g., cost, risk), the problem is called a *minimization problem*. Otherwise, it is called a *maximization problem*, where the objective can be some measure of benefit or profit.

An optimization problem usually defines the requirements that a solution should satisfy. A solution that satisfies the requirements is *feasible*. For instance, both the k-median and k-center problems require that: (1) at most k centers are selected; (2) each object is assigned to only one selected center. A clustering satisfying these two requirements is a feasible solution to both the k-median and k-center problems.

Below we describe a classic optimization problem called the *minimum Steiner tree problem*.

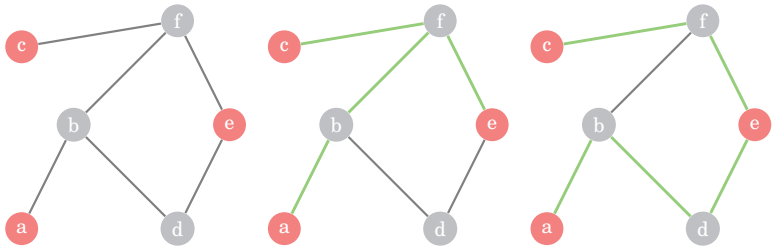
Steiner trees often arise in network designs. Suppose we are given a set of locations that must be connected by wires, the objective is to achieve the lowest possible wiring cost (e.g., measured by the total length of wires).

This problem is formally defined below:

> **Input:** an undirected and weighted graph $G = (V, E, w)$ and a subset of vertices $U \subseteq V$, called *terminals*.
>
> **Objective:** find a tree T that connects U and has the minimum weight, that is, the sum of the edge weights in T is minimized.

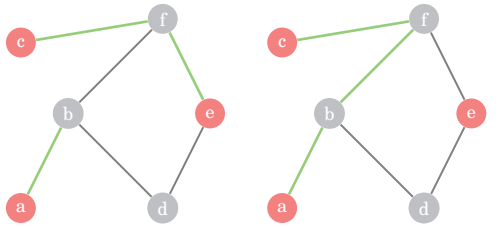
A solution T is feasible if and only if (1) T is a tree and (2) all nodes in U are contained in T. A solution is considered better than another if the former has a smaller weight than the latter. Consider a toy example below, in which the graph is unweighted. The input graph is shown on the left, in which the terminals are drawn in red. Two feasible solutions are plotted next, in which edges colored in green denote the solution. Solution 1 is better than Solution 2 because the former has a small weight (measured

in the number of edges) than the latter.



Input Solution 1 Solution 2

The following solutions achieve smaller objective than Solution 1 above, but are infeasible.



Not a tree Not all terminals are contained

In the following chapters, we will see more examples of optimization problems, for instance, a variant of the minimum Steiner tree problem in Chapter 3.

2.4.2 Approximation algorithms

Before defining approximation algorithms, we define a few related concepts. A *polynomial-time algorithm* is an algorithm whose execution time is either given by a polynomial on the size of the input, or can be upper bounded by such a polynomial. Problems, whose optimal solutions can always be found by at least one polynomial-time algorithms, are called *tractable problems*.

Some optimization problems are *intractable*, that is, no polynomial-time algorithms for them are known to exist. The minimum Steiner tree problem defined previously is such an example. Finding optimal solutions requires brute-force approaches — enumerating all possible solutions and picking the best feasible solution among them.

Sometimes, people design *heuristics* for intractable optimization problems to avoid exhaustive enumeration. The main idea is to trade solution quality for the computation time, in that sub-optimal solutions are acceptable. Heuristics are usually evaluated empirically. They might work well in some cases while failing in others; however, we may not know on which cases.

Approximation algorithms are a special type of heuristics with the key characteristic that their performance is analyzed against the optimal

solutions *theoretically*. Studying approximation algorithms provides a mathematically rigorous basis on which to study heuristics.

Approximation ratio is used to quantify how well an algorithm performs in terms of the solution quality. Given some input instance I of a problem, for instance, a graph G and some terminals U in minimum Steiner tree problem, let $\mathrm{OPT}(I)$ denote the value of the optimal solution (weight of minimum Steiner tree). Suppose an algorithm A produces a feasible solution $A(I)$ given any problem instance I. We want to analyze how well A performs in comparison to the optimal solution. Consider the following ratio

$$\frac{A(I)}{\mathrm{OPT}(I)}.$$

For minimization problems, the ratio is above 1. And for maximization problems, the ratio is below 1. The closer to 1 the ratio is, the better that algorithm A performs on this specific instance I. If this ratio is 1, A produces the optimal solution on the instance I.

Approximation ratio is designed to consider the *worst-case* scenario over *all* possible problem instances. If the problem is a minimization problem, the *approximation ratio of algorithm A* is defined as:

$$\text{approximation-ratio}(A) = \max_I \frac{A(I)}{\mathrm{OPT}(I)}. \tag{2.17}$$

For any maximization problem, the minimum is taken,

$$\text{approximation-ratio}(A) = \min_I \frac{A(I)}{\mathrm{OPT}(I)}. \tag{2.18}$$

Note that A is optimal if and only if $\text{approximation-ratio}(A) = 1$.

In the following chapters, we will present several approximation algorithms developed by us, and re-visit the concept of approximation ratio.

3. Reconstructing propagation processes

In this chapter, we study the problem of reconstructing propagation processes on networks. The word "propagation" is a general term for many real-world phenomena, such as the contagion of disease among humans. We first give an overview of common propagation models. Then we describe important inference tasks related to propagation processes, including the task of reconstructing propagation processes. Next we explain how classic Steiner trees can be used to model and reconstruct propagation processes. In addition, we discuss the limitations of the approach of using Steiner trees. Finally, to address the limitations of some of the proposed approaches, we present two variants of Steiner trees and demonstrate their usage in propagation reconstruction.

3.1 Models for propagation process

The word "propagation" is an umbrella term for many real-world phenomena that involve the spread of a "contagious" state (e.g., disease, opinion) via the connections between objects (e.g., humans). An example is an epidemic in which a disease spreads via human contacts. Another example is the dissemination of a piece of information (e.g., a news article) in online social networks. To study the spreading mechanism of propagation processes, it is often useful to model the process mathematically. Now we describe a few common propagation models. We assume there is an underlying network $G = (V, E)$ in which the propagation occurs.

Susceptible-infected (SI) model [49]. In this model, each node can be in one of the two states: *susceptible* (not infected yet, but can be infected) or *infected*. The whole process breaks down into discrete time steps, $t_0, t_1, t_2, \ldots$. Each step t is represented by two sets: $S(t)$, the set of susceptible nodes at time t, and $I(t)$, the set of infected nodes at time t.

At time t_0, a few nodes, called *seeds*, are infected, and they define $I(t_0)$ and the remaining nodes define $S(t_0)$. Before the next step t_1, the nodes in $I(t_0)$ infect their susceptible neighbours with a certain probability. At

time t_1, the newly-infected nodes are added to $I(t_1)$, together with $I(t_0)$. Accordingly, the newly-infected nodes are removed from $S(t_1)$. Once a node becomes infected, it can never become susceptible. The above process repeats until all nodes become infected.

This model has one parameter p, called *infection probability*. Between time t and $t+1$, for each pair (u,v) such that $u \in I(t)$ and $v \in S(t)$, node u has probability p to infect node v. In addition, these random events happen independently over all pairs of adjacent nodes. Note that a susceptible node will eventually become infected.

A variant of the SI model is called *SIR model*, which incorporates a third state *recovered* (or *removed*). Only infected nodes can become recovered and once they become recovered, they stay recovered. The SIR model has an additional parameter of q, called *recovering probability*. At each step t, each $u \in I(t)$ has probability q to recover. The whole process terminates when all nodes are recovered.

Linear threshold (LT) model [40, 75]. In this model, each node can be either *active* (infected) or *inactive* (uninfected). Different from the SI model where propagation occurs independently over each edge, the LT model assumes that propagation takes into account all edges that share one endpoint. This model has edge-wise weights b and node-wise thresholds θ. Each directed edge (u,v) is associated with weight b_{uv} which indicates its strength. The values of b satisfy $\sum_{(u,v)\in E} b_{uv} \leq 1$ for each node v. Each node v has a threshold value $\theta_v \in [0,1]$, which controls how easily v can be activated.

In this model, all nodes that become active at some point remain active. For each inactive node v, if the sum of weights from its active neighbours exceeds θ_v, node v becomes active. In other words, node v becomes active if

$$\sum_{(u,v)\in E, u \text{ is active}} b_{uv} > \theta_v.$$

Given an initial set of active nodes, the process unfolds deterministically until all nodes are active.

Independent cascade (IC) model. This model has been studied in the context of marketing by Goldenberg et al. [37, 38]. Similar to the LT model, each node is either active or inactive. However, what makes this model different is that the activation attempt over each edge (u,v) occurs at most once during the whole propagation process. Model parameters are edge-wise infection probabilities, p_{uv}, which is the probability for node u to activate node v. At time t, an active node u has probability p_{uv} to activate its inactive neighbor v. If the event succeeds, v becomes active and remains so. Otherwise, u does not have a second chance to activate v. If node v neighbors multiple active nodes, the activation attempts can be ordered arbitrarily.

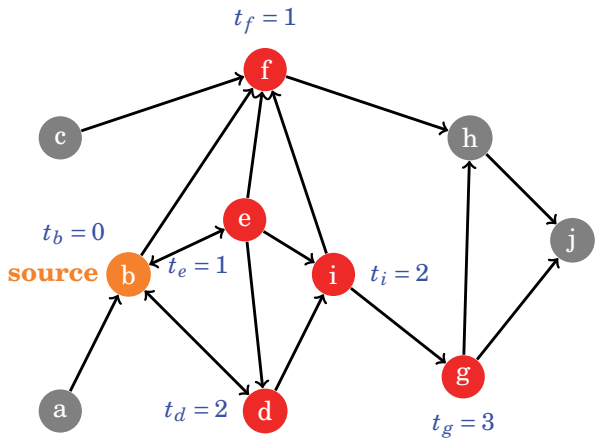
Continuous-time independent cascade (CTIC) model. A known limi-

tation of the IC model is its inability to handle heterogeneous time delays of activation. As a result, a generalization of the IC model called *Continuous-Time Independent Cascade model* is proposed by Gruhl et al. [41]. In this model, node activation can happen at different rates according to edge-specific parameters. Each edge (u,v) is associated with a probability function, which describes the time delay for v to be activated by its neighbor u. Usually the time delay between (u,v) is sampled from some exponential distribution, parametrized by a parameter λ_{uv}. In case a node v is activated by multiple neighbours, the neighbor corresponding to the shortest time delay is chosen as the activator of v. This process terminates when all nodes are active.

3.2 Retrospective tasks on propagation processes

Next we describe common retrospective tasks on propagation processes in networks. We call them retrospective because the goal is to infer network states *in the past*. In contrast, a different line of tasks focuses on *predicting* network state in the future, which we will describe briefly at the end of this section.

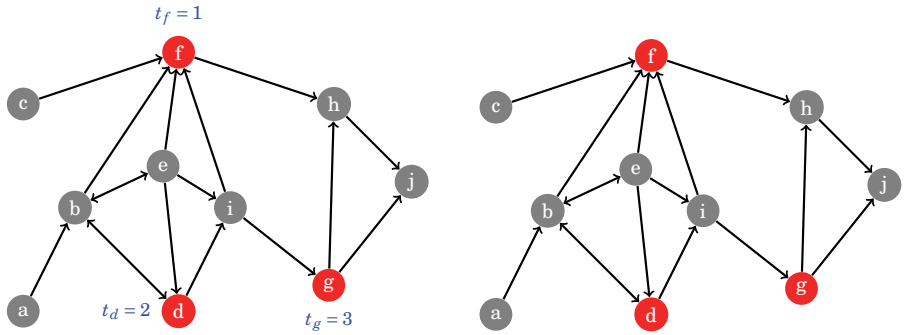
In the sequel, we use the words "infected" and "active" as well as "uninfected" and "inactive" interchangeably.



Cascade: $\{(b,0), (e,1), (f,1), (d,2), (i,2), (g,3)\}$

Figure 3.1. A cascade on a toy graph.

We use the term *cascade* to refer to an instantiation of a propagation process. For an infected node u, let t_u be the time that u become infected. If a node is not infected, we set $t_u = \infty$. A cascade can be represented by the nodes' infection time, for example, using $\{(u,t_u)\}$. By convention, uninfected nodes are omitted in the representation. We give an example



(a) infection time is known (b) infection time is unknown

Figure 3.2. Two types of partial observation: with or without activation time information.

cascade in Figure 3.1. Infected and uninfected nodes are colored in red and grey, respectively, and there is only one source node. We say a cascade is *partially observed* if only a subset of the infected nodes is known. In addition, the infection time of the observed infections may or may not be available and both situations are illustrated in Figure 3.2. Observed infected nodes are colored in red, while both unobserved infected nodes and uninfected nodes are colored in grey.

In practice, partial observation of a cascade is a realistic assumption. For instance, during an epidemic outbreak, it is almost impossible to obtain the complete knowledge of who is infected. But still, we can obtain a partial picture by diagnosing a subset of the population. Similarly, during the spread of misinformation in online social media, we might want to know which users are hoaxed to believe the misinformation. Unfortunately, acquiring this information is also challenging.

Next we describe some common inference tasks on network propagation processes.

Source identification. In case the cascades are only partially observed, can we identify the source of the process and possibly its activation time? We illustrate this task in Figure 3.3, in which the source is drawn in orange. In this example, the source is not observed but in general, it can be.

Identifying the source of a cascade has many applications. In epidemiology, the source refers to the first person that was infected, or usually known as the *patient zero*. Identifying the patient zero may provide useful information about the scale of an epidemic and who else could be infected. In the case of misinformation, the source is also called *"rumor source"*, the first person who started a piece of rumor. Detecting the rumor source can prevent further spread of rumors.

This task has a few problem dimensions, including:

What is the propagation model? The SI model is one common choice [30, 33, 64, 77, 83]. Other models such as the IC model and the CTIC model

are also considered [32, 53]. A recent trend is a model-agnostic approach in which no assumption on the underlying cascade model is made. An example is the work by Rozenshtein et al. [73].

What is observed? Prakash et al. [64] assume that all activated nodes are observed without temporal information. Sundareisan et al. [83] make similar assumptions but only a subset of the infected nodes is known. A different line of work by Rozenshtein et al. [73] and Farajtabar et al. [32] assumes the knowledge of the infection time of observed infections. In addition, a common assumption is that the observation is noise-free. However, the works by Rozenshtein et al. [73] and Sundareisan et al. [83] break this assumption.

How many source nodes are there? Shah and Zaman [77] and Farajtabar et al. [32] assume a single source node in a cascade, while other works [53, 64, 73] can deal with k sources where $k \geq 1$. In addition, the value of k can be either specified in the input [64] or determined automatically by the inference algorithms [73].

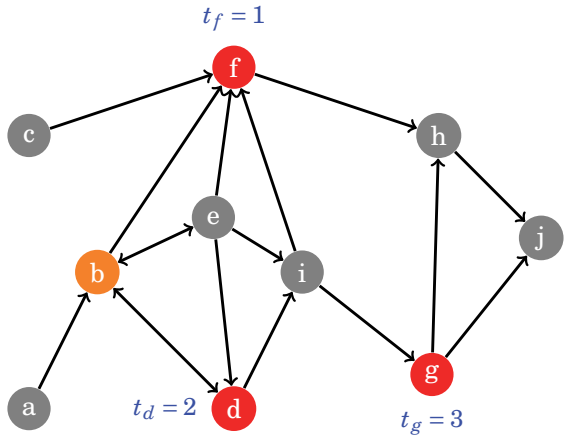


Figure 3.3. Source identification asks to find the source of a cascade that is partially observed. In this example, the cascade has only one source.

Network structure inference. In some situations, we observe cascades without knowing the network structure on which the cascades take place. For instance, a blogger might write an article on their blog after getting inspired by another article from a different blogger, but they do not explicitly refer to the article from which they get inspired. In the case of epidemics, infections are observed at different times, but without knowing the actual physical contact network.

The task of network structure inference is to infer the network structure based on one or more observed cascades. We give an illustration in Figure 3.4. Another related task is about estimating the edge-specific infection parameters such as the activation probabilities in the IC model and the transmission rate parameters in the CTIC model.

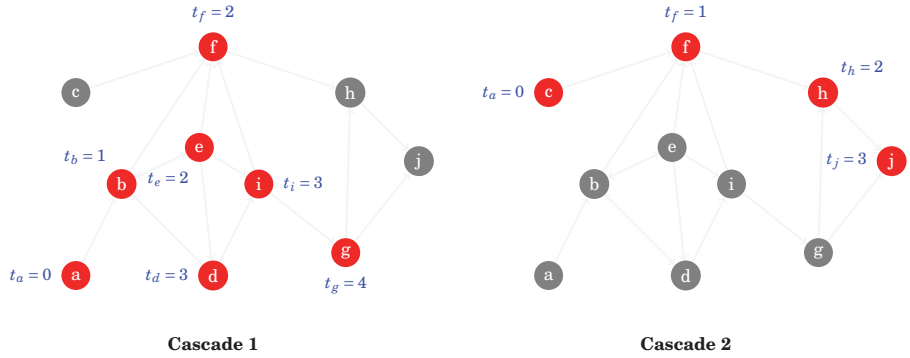


Figure 3.4. Network structure inference asks to uncover the underlying network structure based on one or more observed cascades. In this example, we show two cascades, based on which we want to infer the network edges colored in light gray.

Existing works on network structure inference usually make assumptions on the underlying cascade model, as in the case of source identification. An early work by Myers and Leskovec [62] considers a generalization of the SIR model. A study by Gomez-Rodriguez et al. [39] assumes the IC model and they use a global infection parameter over all edges. Rodriguez et al. [71] assume the CTIC model and attempt to learn edge-wise transmission rate parameters. A recent work by Du et al. [28] generalizes the previous work by taking into account the multi-modal nature of transmission time functions. Specifically, the probability density function of transmission time can have multiple peaks, instead of one.

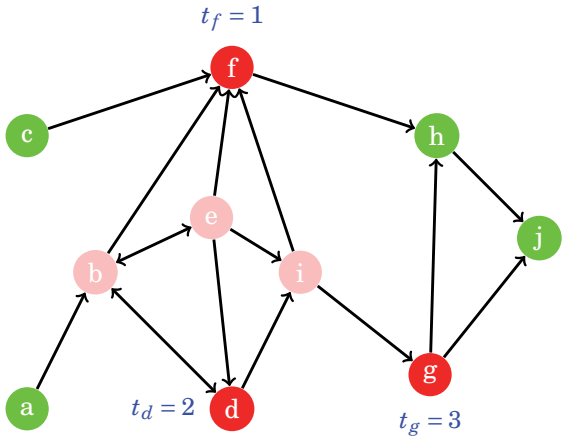


Figure 3.5. The task of cascade reconstruction asks to find out the states of nodes whose states are unobserved, based on partially-observed cascades.

Cascade reconstruction. Given a partially-observed cascade, the cascade-reconstruction task asks to infer the states of nodes whose states are unobserved (we call such nodes *unobserved nodes*). We illustrate this task in Figure 3.5. Infected nodes that are observed are colored in red. Unob-

served infected nodes and uninfected nodes are colored in pink and green, respectively. In the simplest setting, we are asked to determine whether each unobserved node is infected or not. In addition, the infection time can be asked, or the complete propagation trace e.g., who infected whom.

The topic of cascade reconstruction receives relatively little attention. In the simplest setting [68, 76, 83], one assumes the underlying network is static and only node infection states (without temporal information) are observed. The goal is to determine whether any node with unknown status is infected or not. Chen et al. [17] and Sefer and Kingsford [76] generalize the previous setting by considering infection time in the observation.

An interesting work by Rozenshtein et al. [73] deals with *temporal networks*, in which the network structures change over time. In addition, they assume the observed cascade contains infection time information. As the available information becomes richer, they can infer the complete infection trace (who infected whom) as well as the cascade sources.

The above works focus on reconstructing a single cascade. Sun et al. [82] lifts this assumption by reconstructing *multiple* cascades jointly. Their method exploits the synergy relationships between co-evolving cascades.

Predictive tasks. These tasks, in general, ask to predict some state of a propagation process *in the future*. For instance, a popular predictive task is to predict the size of a cascade [18, 56, 86]. A more fine-grained task asks to predict *who* will be infected in the future. Examples include the work by Islam et al. [46] and Wang et al. [85]. These works mainly differ in whether the activation time is part of the prediction or not.

3.3 Using Steiner trees to model propagation processes

Steiner trees are data structures that have many application domains such as infrastructure design [27], molecular biology [61] and VLSI design [45]. In this section, we explain how Steiner trees can be used to model propagation processes.

Tree representations for cascades. Some propagation models such as the SI model and the IC model require that each infected node is infected via a single edge, which we call *a cascade edge*. If we assume there is a single source for each cascade, then the propagation trace of a cascade forms a tree, in which nodes in the tree are infected, the source node is the root of the tree, and tree edges correspond to the cascade edges.

We give an example of a cascade represented as a tree in Figure 3.6. In this example, the graph is directed. The source node is drawn in orange and other infected nodes are drawn in red. The cascade edges are drawn in green thick lines.

Steiner tree representation under partial observation. In practice, complete knowledge of a cascade is often unavailable. Instead, only a

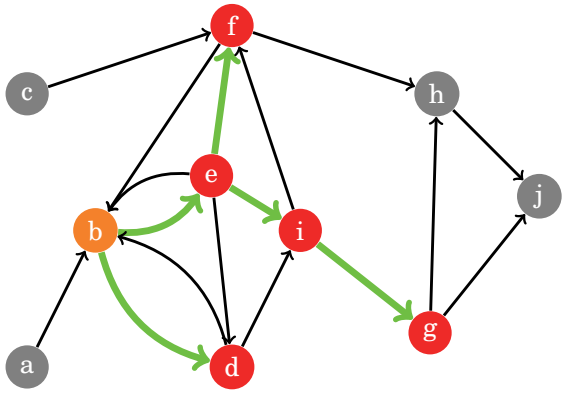


Figure 3.6. A cascade represented as a tree.

fraction of activated nodes is observed. Assume that the underlying cascade is a tree, the task of cascade reconstruction asks the following questions:

Q1: can we find a tree that *explains* the observed infections?

Q2: if there are multiple such trees, which tree is the *best*?

The above questions do not have immediate answers. First, the meaning of "explain" is vague; what does it mean for a tree to explain some observations? Second, what is the criterion for a tree to be the "best"? To make the questions less ambiguous, we may first simplify the problem setting. For instance, we can drop the infection time in the observation. In this view, we can reduce **Q1** to: can we find a tree that *spans* the observed infected nodes? A trivial answer is to return a spanning tree since it spans all observed infections. However, this answer may not be reasonable because all nodes are considered infected, which is not realistic.

Arguably, a more reasonable answer is to use a *Steiner tree*, which spans only a subset of all nodes in the graph. Recall that we introduce Steiner trees in Section 2.4. A Steiner tree T, given a set of *terminals* U, is a tree that spans U, but not necessarily nodes in the entire graph. In our situation, we set the observed infections to be the terminals U, and we say a Steiner tree on U *explains* the observation.

We give two examples of Steiner trees that explain a partially-observed cascade in Figure 3.7. Tree edges are highlighted in green thick lines. Observed infections (the terminals) are colored in red in both Steiner trees. Since the two trees are different, the reconstructed cascade trees are different. In the tree in Figure 3.7(a), nodes $\{b,i\}$ are inferred to be infected, while in the tree in Figure 3.7(b), nodes $\{e,i\}$ are infected.

To answer **Q2**, we ask a different question: can we define a *quality function* on Steiner trees that determines how good a tree is? First we want to mention that it is common to assign edge-specific infection probabilities. For each edge (u,v), let p_{uv} denote the probability for node u to infect node v. Then we can use edge infection probabilities to calculate the probability

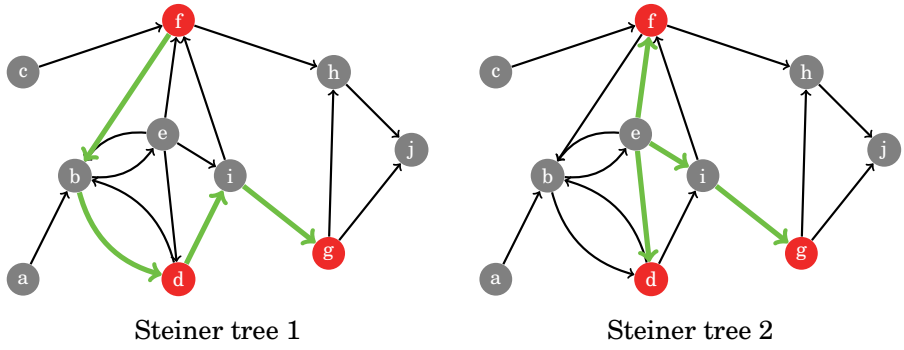


Figure 3.7. Two examples of Steiner trees explaining the same partially-observed cascade.

of any Steiner tree and use the tree probability as a measure for its quality. The probability of a tree T can be defined as follows:

$$Pr(T) = \prod_{(u,v)\in T} p_{uv}. \tag{3.1}$$

The larger probability a tree has, the more likely it is the underlying cascade. In Figure 3.8, we plot two trees with edge probabilities shown beside each edge. The larger probability an edge has, the thicker we plot it. The probability of T_1 is 0.252 according to Eq. 3.1 and the probability of T_2 is 0.0042. Therefore, T_1 is more likely than T_2.

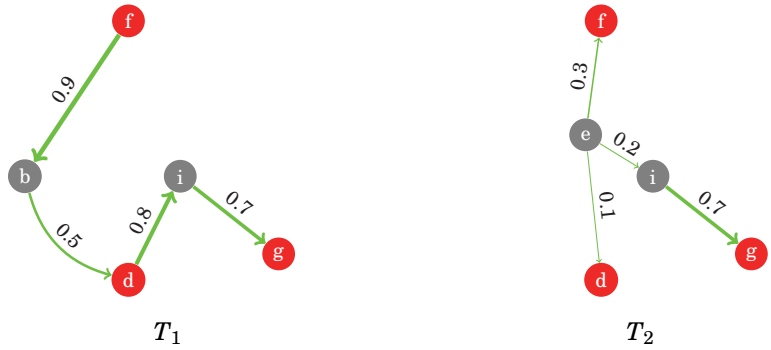


Figure 3.8. Two Steiner trees with different probabilities.

We can use Eq. 3.1 to measure the quality of a tree but there is an equivalent way to do so. If we take negative logarithm on Eq. 3.1, we have the *negative log-likelihood* of the probability of T:

$$-\log(Pr(T)) = -\sum_{(u,v)\in T} \log(p_{uv}) \tag{3.2}$$

Under this transformation, edge weights are transformed by taking the negative logarithm.[1] Accordingly, the best tree is the tree with the

[1]A benefit of taking the negative logarithm is it mitigates the issue of float-point

smallest weight. And **Q2** becomes: *can we find a tree T that minimizes* $-\sum_{(u,v)\in T}\log(p_{uv})$ *?*

Finally, we combine **Q1** and **Q2** into a single problem: finding a Steiner tree that has minimum weight, where edge weights are measured by $-\log(p_{uv})$. This is essentially the *minimum Steiner tree problem*, which we introduce in Section 2.4. Though the problem is an **NP**-hard problem, efficient constant-factor approximation algorithms have been designed [36].

Theoretically sound as this approach seems, it has a few limitations, which we explain next.

Limitations of using minimum Steiner trees. First, the use of Steiner trees assumes that the underlying cascade is a tree and further there is only one source. This assumption does not always hold. For instance, Steiner trees cannot capture cascades generated by Linear Threshold model (Section 3.1), in which a node can be infected by the combined effect of multiple neighbours.

Second, when infection time is also observed, Steiner-tree model could violate the observation. For instance, the best Steiner tree can produce an infection order that is inconsistent with the observed order. We will explain this issue in more detail in the next section.

Third, using a single tree as the reconstruction result is equivalent to making *binary predictions* on nodes' states: a node is considered to be infected if and only if it is in the tree. However, propagation processes are stochastic in nature. Instead of making deterministic predictions, can we predict *the probability* that a node is infected?

In the following sections, we address two of the above limitations.

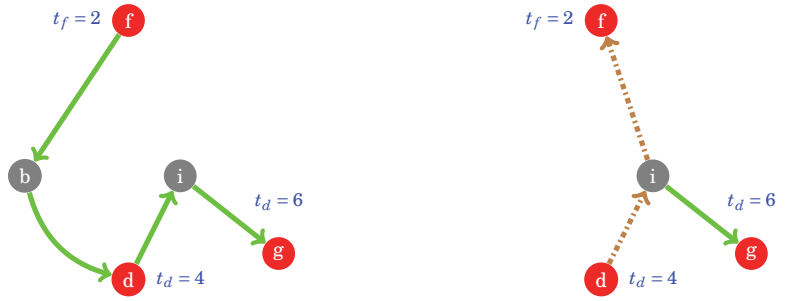
3.4 Thesis contribution 1: temporal extension

In Publication I, we extend the aforementioned Steiner tree approach to the temporal setting. We assume that the observation contains infection time information and the observations are pairs of (u,t), where u is the infected node and t is the infection time of u. In addition, we do not assume any underlying propagation model, which makes it applicable to more than one propagation model.

Our problem formulation is a generalization of the minimum Steiner tree problem. Meanwhile, our problem imposes path-related constraints induced from the temporal information in the observation. Given a tree T and a candidate source node $r \in T$ as the tree root, we consider an observation (u_1,t_1), where u_1 is an infected node and t_1 is its activation time. We say the path p from the root r to node u_1 is an *order-respecting path* if there does not exist any other observed node u_2 (with activation t_2) on p such that $t_2 > t_1$. The intuition of using order-respecting paths is

underflow, which happens when multiplying fractionals.

that an observed infected node that is infected at t cannot appear after (in the tree topology) another infected node which is infected later than t. We say a tree T rooted at r is an *order-respecting tree* if all paths from r are order-respecting paths. We illustrate the notion of order-respecting trees in Figure 3.9. Observed infections are colored in red and the infection time is shown next to each observed infection. The tree in Figure 3.9(a) has only one path (f,b,d,i,g), which is order-respecting because $t_f \le t_d \le t_g$. Therefore, the tree is order-respecting. In contrast, the tree in Figure 3.9(b) is not order-respecting because a path in it, (d,i,f), is not order-respecting. The node f appears after the node d (which is the root) in the tree, but f is infected earlier than d. We highlight the order-violating path in a brown and dashed line.



(a) An order-respecting tree (b) An order-violating tree

Figure 3.9. An order-respecting tree and an order-violating tree.

Our problem is defined as below:

Problem 1. *We are given a weighted undirected graph $G = (V,E,w)$ and a set of observed infected nodes $R = \{(u,t)\}$ with $u \in V$ and $t \in \mathbb{R}$. The goal is to find a seed $s \in V$ and a tree T rooted at s, such that*

(1) T spans all the observed infections,

(2) T rooted at s is order-respecting, and

(3) the weight of T, $\sum_{(u,v)\in T} w(u,v)$ is minimized.

The above problem is a generalization of the minimum Steiner tree problem, thus it is a **NP**-hard problem. We present 3 approximation algorithms. The best algorithm in terms of approximation guarantee gives $\mathcal{O}(\sqrt{k})$-approximation, where k is the number of observed infections.

We use a toy example of a grid graph to illustrate the performance of our method. In Figure 3.10, we show the ground-truth cascade on the left and a solution found by our method on the right. Although the solution does not match the ground truth exactly, all nodes spanned by the solution tree are actually infected. In other words, it achieves high precision.

We also evaluate our methods on real-world graphs and found that our methods usually achieve high precision, that is, nodes spanned by the

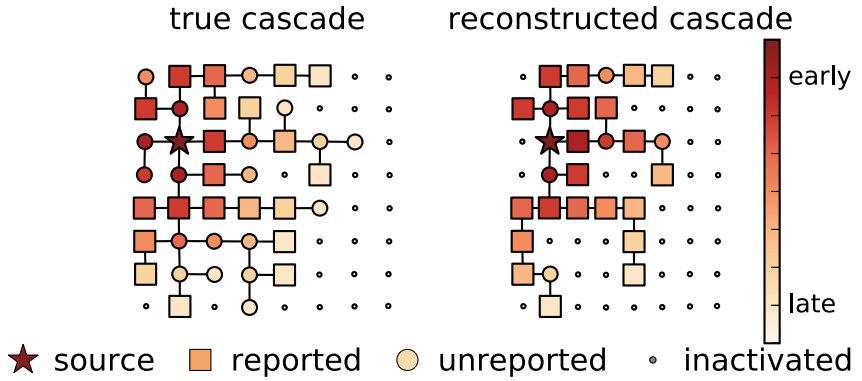


Figure 3.10. Our method aims at reconstructing the underlying cascade given reported infections (left). The reconstructed cascade (right) is parsimonious and respects node infection order induced by infection time. Node infection time is indicated by the colorbar.

solution tree are mostly infected. Meanwhile, our algorithm produces trees that better respect the infection order, compared to the algorithms under the minimum Steiner tree formulation (without infection order constraint). However, low recall is one main drawback of our algorithms. This drawback is expected since our goal is to find trees with minimum weights.

3.5 Thesis contribution 2: probabilistic extension

In Publication II, we consider the task of *probabilistic cascade reconstruction*. We assume only a fraction of infected nodes is observed but without the knowledge of their infection time. Then given a partially-observed cascade, the goal is to infer *infection probability* of the nodes whose states are unobserved. Estimating infection probability makes our approach more suitable in practice, compared to binary predictors, because real-world propagation processes are stochastic in nature. In addition, our formulation does not assume any underlying propagation model, which makes it applicable to more than one propagation model.

We prove that computing nodes' infection probability is a #P-hard problem. Thus, we resort to a Monte-Carlo approach — approximating the infection probability by sampling. We propose the problem of sampling Steiner trees from a target distribution. Then using Steiner tree samples, a node's infection probability is estimated to be the fraction of tree samples in which the node appears. We outline our approach in Figure 3.11.

Our main theoretical contribution is two novel Steiner-tree sampling algorithms with provable guarantees on the probability distribution of tree samples. The problem of sampling Steiner trees with provable guarantees is a generalization of another problem — sampling spanning trees. Though sampling spanning trees has been extensively studied [13, 88], Steiner-tree

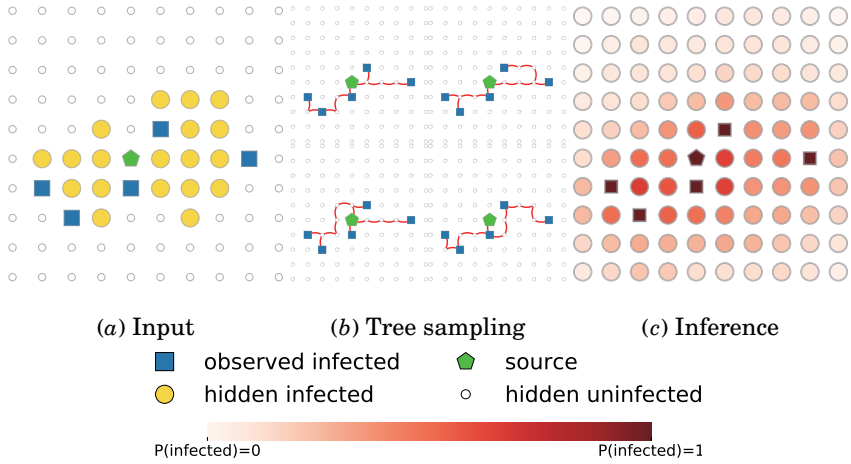


Figure 3.11. Overview of our approach: (*a*) given input observed infections; (*b*) Steiner trees are sampled, each representing a possible cascade explaining the observation. Four samples are shown; (*c*) node infection probabilities are estimated from the samples.

sampling is a novel problem. In general, sampling combinatorial structures (e.g., triangles, trees) is an active research area.

Empirically, our method produces predictions that better align with ground-truth cascades, compared to the baselines. In Figure 3.12, we illustrate the difference between our method and the baselines using a synthetically-generated cascade on a 2D grid graph. The underlying cascade with observations is shown in Figure 3.12(a). Outputs by different methods are shown in the remaining figures. One of the baselines, *NetFill* [83] is a state-of-the-art cascade-reconstruction method designed specifically for the SI model. The other baseline, *Min-Steiner-tree* solves the minimum Steiner tree problem by treating the observed infection as terminals. Both *NetFill* and *Min-Steiner-tree* make binary predictions on nodes' infection states. In contrast, our method *Tree-sampling* makes probabilistic predictions.

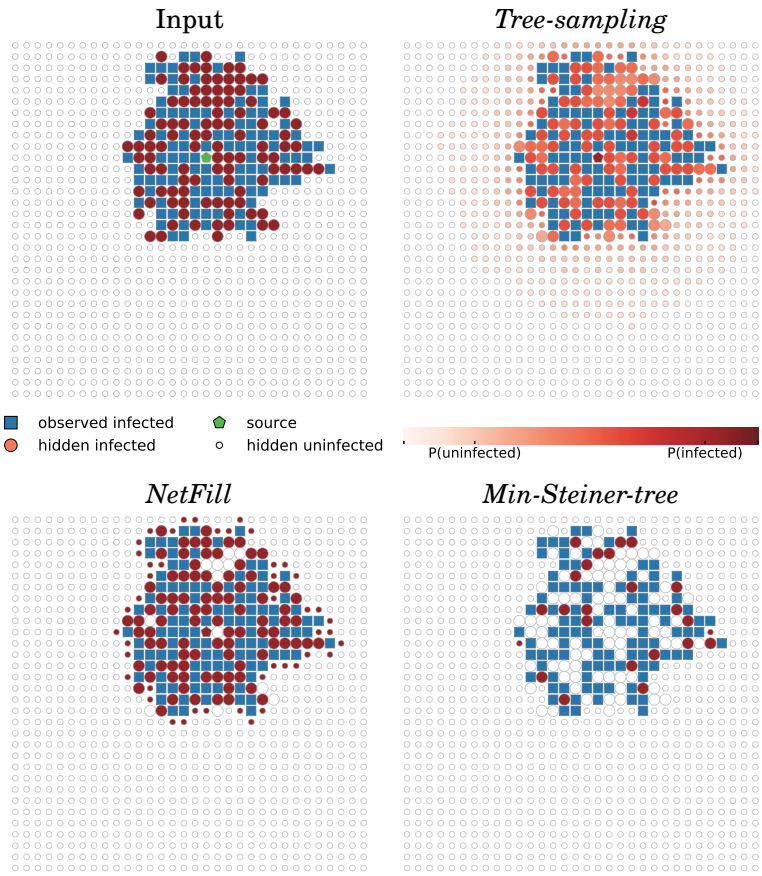


Figure 3.12. Probabilistic cascade reconstruction, by three methods, on a 2D lattice graph. *Tree-sampling* is our proposed method.

4. Searching for polarization in social networks

In this chapter, we start by introducing signed graphs, a powerful model to detect polarization in social networks. Then we discuss three different but related problems about signed graphs, all related to the task of polarization detection. Finally, the last problem leads to the contribution of this thesis and we present our solution to this problem.

4.1 Signed graphs

In many real-world networks, the interactions between two entities can be labeled as either positive or negative. Two people, for instance, can be either friendly or antagonistic to each other. Similarly, a country can be an ally or in conflict with another country. In protein-protein interaction networks, a protein complex can activate or inhibit the functioning of another protein. In human languages, two words may exhibit synonym (e.g., "happy" and "cheerful") or antonym (e.g., "happy" and "sad") relations.

Based on these observations, the concept of *signed graphs* is proposed to model the interactions that are either positive or negative. We provide a real-world example of signed graphs in Figure 4.1. This graph encodes the relations among several tribes in New Guinea. Each tribe is shown as a node, and friendly (antagonistic) relations are represented by solid green (dashed red) lines.

Early works in the study of signed graphs are mainly about observational data from the physical world, for example, the international relations in Europe from 1872 to 1907 [43] and the relations among Allied and Axis powers during World War II [4].

Over the years, with the advent of online social media, increasing focuses have been brought to signed graphs that originate from the online world. Examples of these graphs include Epinions [57] and Slashdot [52]. Compared to the signed graphs studied at the early times, these graphs are usually much larger.

Challenges and opportunities. An unsigned graph can be seen as a

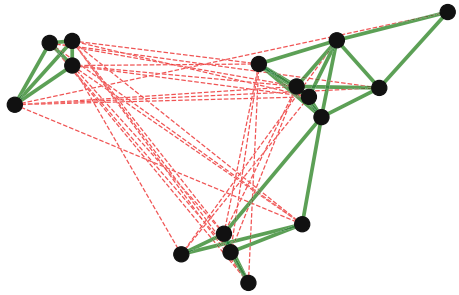


Figure 4.1. *New Guinea Highland Tribes* graph by Read [69], an example of signed graphs. Positive edges are shown in green solid lines, and negative edges are shown in red dashed lines.

graph with only positive edges and thus, it is a special case of signed graphs. Even though studies on unsigned graphs have been undergoing for decades, the existence of negative edges in signed graphs creates both new challenges and new opportunities.

On the one hand, many algorithms for mining unsigned graphs cannot directly handle negative edges. For example, for community detection algorithms on unsigned graphs, such as the Louvain method [10] and the Label Propagation algorithm [67], processing edges with negative signs is not straightforward.

On the other hand, signed graphs facilitate the study of new theory such as social balance theory [43] and status theory [55], which cannot be studied on unsigned graphs. For instance, *social balance theory* captures a commonplace notion that "the friend of a friend is a friend" and "the enemy of an enemy is a friend". Signed graphs naturally capture this intuition, which is illustrated in the first two triads on the left (a triad is a graph with 3 nodes and 3 edges) in Figure 4.2. The two triads on the left in Figure 4.2 are called *balanced*. In contrast, the two triads on the right in the same figure are *unbalanced* because they fail to capture the above notion. Note that unsigned graphs can only model the first triad while signed graphs can model all of them, due to the addition of edge signs.

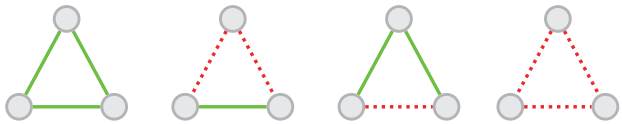


Figure 4.2. The four possible signed triads. Positive edges are drawn in green solid lines and negative edges are drawn in red dashed lines.

In practice, the addition of negative edges brings new opportunities for traditional graph mining tasks on unsigned graphs. Take community detection for example, higher-quality communities can be found by consid-

ering both positive and negative edges, compared to using positive edges only [31]. Moreover, signed graphs enable us to study new problems that unsigned graphs are not suitable for. Bonchi et al. [12], Chu et al. [22] and Xiao et al. [89] use signed graphs to find polarized user groups on social-media platforms. In biology, signed graphs are used to model protein-protein interactions and are further used to discover synergetic protein groups [63].

Constructing signed graphs in practice. Curious readers might wonder how signed graphs are constructed in practice. We describe a few general approaches to construct real-world signed graphs.

If a graph has a reasonable size in terms of the number of edges, edge signs can be inferred manually. An example is the New Guinea Highland Tribes graph [69], which contains 58 undirected edges. Relations between tribes were inferred manually by the author Read [69].

When graphs are too large, for instance, graphs from popular social-media platforms, manually inferring edge signs is costly and time-consuming. A common practice is to leverage auxiliary edge-related information available from the platforms, to serve as a *proxy* to the actual edge signs. The information is a suitable proxy if it contains strong signals of edge signs. For instance in Slashdot.org, an online discussion site, a user can tag another user as either "a friend" or "a foe." In Epinions.com, a consumer review site, a user can tag another user either "trustful" or "distrustful". In Wikipedia.org, community members may vote "support" and "oppose" to admin candidates during administrator elections. In the modeling of human languages, determining if two words are synonyms or antonyms can rely on information from thesauruses such as Merriam-Webster's Dictionary [79].

In the cases when no reliable proxy is available, one may resort to other approaches. Texts, which usually reflect human emotions, are abundant in social-media platforms. To construct signed graphs from social media, one can first construct unsigned backbone networks using user-interaction data (e.g., a user re-tweets/mentions/comments about another user). Then based on the backbone networks, user-generated texts (e.g., comments, mentions) can be analyzed, using sentiment analysis algorithms [58] or crowdsourcing, to determine the edge signs.

A running example of a signed graph constructed using the above approach is the work by Kumar et al. [51]. They construct a *community graph* on Reddit.com, where each node represents a "subreddit" (a community of users interested in a certain topic). There is an edge between two nodes if the corresponding subreddits are linked by at least a hyperlink between the posts in each corresponding subreddit. They further use crowdsourcing to analyze the text content in the posts to infer the edge signs. Another example is from the field of biology, in which a protein may exhibit activating or inhibiting relations with another protein. Ou-Yang et al. [63] leverage

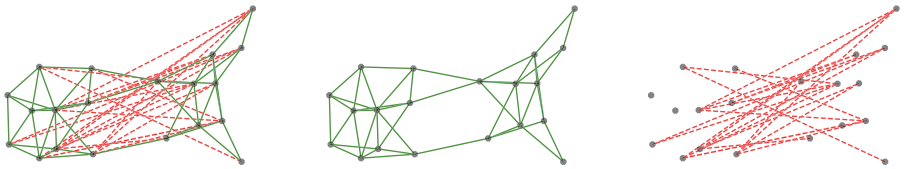
yeast metabolic cycle and gene expression microarrays to infer the "signs" of interactions.

4.2 Polarized structures and the importance of negative edges

To further illustrate the importance of negative edges in signed graphs, we explain how negative edges can help discover polarized structures in signed graphs.

Imagine a network of politicians represented as nodes, and there is an edge between two nodes if the corresponding politicians interact with each other. The sign of an edge is determined by whether the corresponding interaction is supportive (positive sign) or opposive (negative sign). We illustrate this network in Figure 4.3(a). In addition, the politicians in this example hold two different views about a government policy, and are unfortunately involved in a polarized debate. Furthermore, the graph exhibits a *polarized structure*, because politicians with the same view are likely to support each other, and those of different views tend to oppose each other. However, these rules are not strict. For some unknown reasons, politicians with the same view (or different views) may still have a small chance to oppose (or support) one another, thus creating some noisy edges to the polarized structure. Despite so, all politicians can be roughly divided into two groups, corresponding to each side of the polarized structure, and each group of the politicians represents a view regarding the government policy.

Suppose we only know the edges and their signs, can we identify the two groups of politicians representing each view on the policy? In other words, can we partition the nodes in the graph, such that inside each partition, nodes are connected mostly by positive edges, and between two partitions, edges are mostly negative?



(a) Edges of both signs (b) Only positive edges (c) Only negative edges

Figure 4.3. A synthetic signed graph is shown in (a). Subfigures (b) and (c) represent the graph, with only positive and only negative edges, respectively.

Instead of describing a solution to the above question, we first discuss the roles of edges of each sign. We argue that edges of either sign provides *complementary information* in terms of the graph structure. In our example, the graph with only positive edges (shown in Figure 4.3(b))

has a *well-separated* structure, in that it can be partitioned into two subgraphs, such that each subgraph is densely-connected within itself, but is sparsely-connected with the other subgraph. In contrast, the graph with only negative edges (shown in Figure 4.3(c)) has a *near-bipartite* structure, which is *opposite* to the previous case. This graph can be partitioned into two subgraphs, such that each subgraph is sparsely-connected within itself, but is densely-connected with the other subgraph. Both structures in this example indicate a similar node partitioning, but from different perspectives. Therefore, edges of both signs are useful to find a good graph partition.

4.3 Partitioning signed graphs

In this section, we formalize the graph-partitioning problem described in the previous section and present some known theoretical results.

Preliminaries. We first define signed graphs formally and present some notations that are used shortly.

We define a *signed graph* as an undirected graph $G = (V, E^+, E^-)$, where $V = \{1, \ldots, n\}$ is the set of nodes, E^+ is the set of positive edges, and E^- is the set of negative edges. The graph G can be decomposed into the *positive graph* $G^+ = (V, E^+)$ and the *negative graph* $G^- = (V, E^-)$. The set of all edges is $E = E^+ \cup E^-$.

The adjacency matrix for G^+ is denoted A^+. The entry $A^+_{i,j}$ is 1 if $(i,j) \in E^+$ and it is 0 otherwise. We define A^- for the G^- accordingly. The *signed adjacency matrix* of G is defined as $A = A^+ - A^-$. For any node $i \in V$, denote the *positive degree* of i as $deg^+(i) = \sum_{j=1}^{n} \left|A^+_{i,j}\right|$. The *negative degree* of i, $deg^-(i)$, is defined similarly. The degree of i is $deg(i) = deg^+(i) + deg^-(i)$. And the *degree matrix* D is a diagonal matrix, whose diagonal entries are $D_{i,i} = deg(i)$ for $i = 1, \ldots, n$. Last the *volume* of a node set S is defined as $vol(S) = \sum_{i \in S} deg(i)$.

Given a set of edges F and two node sets $X, Y \subseteq V$, we define $F(X,Y) = \{(u,v) \in F \mid u \in X, v \in Y\}$, that is, the edges in F that have one endpoint in X and the other endpoint in Y. For example, $E^-(X)$ is the set of negative edges between X and Y. When $X = Y$, we write $F(X,X) = F(X)$ for brevity. For example, $E^+(X)$ is the set of positive edges having both endpoints in X.

Graph bi-partitioning. We state our first problem next:

Problem 2. *Given a signed graph $G = (V, E^+, E^-)$, find a partition (C_1, C_2) over V such that $C_1 \cap C_2 = \varnothing$, $C_1 \cup C_2 = V$ and (C_1, C_2) minimizes the following:*

$$q(C_1, C_2) = \frac{|E^-(C_1)| + |E^-(C_2)| + 2\left|E^+(C_1, C_2)\right|}{vol(C_1 \cup C_2)}.$$

The term $q(C_1, C_2)$ can be interpreted as the fraction of "noisy" edges, that is, the edges that undermine the informativeness of (C_1, C_2) as a partition.

It is easy to see that $q(C_1,C_2) \in [0,1]$, and the smaller that $q(C_1,C_2)$ is, the better partition that (C_1,C_2) is. If there is some (C_1,C_2) such that $q(C_1,C_2)=0$, the graph G is *balanced*, that is, there is no negative edge in either C_1 or C_2, and no positive edge between C_1 and C_2. If no (C_1,C_2) exists such that $q(C_1,C_2)=0$, the graph G is *unbalanced*.

We give an example of a partition in Figure 4.4, in which the blue dotted line indicates a partition of (C_1,C_2). The partition is not perfect since $q(C_1,C_2)>0$ and there are negative edges inside the subgraph of C_2. And in fact, this graph is unbalanced.

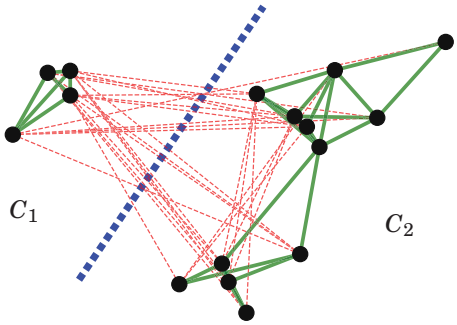


Figure 4.4. A partitioning over the nodes in *New Guinea Highland Tribes* graph.

Spectral graph theory on signed graphs. Attempts to solve Problem 2 contribute to the development of spectral graph theory on signed graphs. In this theory, people study the properties of signed graphs in relation to the eigenvalues and eigenvectors of the matrices associated with the graphs.

Recall that in Section 2.2, we introduce spectral graph theory and how to use the graph Laplacians to partition unsigned graphs. Similarly, Laplacian matrices are defined for signed graphs and these matrices can be used to partition signed graphs as well. Next we explain how.

The *unnormalized Laplacian matrix* of a signed graph is defined as

$$L = D - A, \tag{4.1}$$

and the *normalized Laplacian matrix* is

$$\mathscr{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}. \tag{4.2}$$

Then we can use $\mathscr{L}$ to transform Problem 2 into a matrix form. Consider a vector $\mathbf{x} \in \{-1,1\}^n$, which encodes a partition (C_1,C_2) of the graph, such that $i \in C_1$ if $\mathbf{x}_i = 1$ and $i \in C_2$ if $\mathbf{x}_i = -1$. Then it can be shown that

$$\frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T D \mathbf{x}} = \frac{|4\,E^-(C_1)| + 4\,|E^-(C_2)| + 4\,\left|E^+(C_1,C_2)\right|}{vol(C_1 \cup C_2)}. \tag{4.3}$$

If we replace $\mathbf{x}$ with $\mathbf{y} = D^{\frac{1}{2}}\mathbf{x}$, we have the *Rayleigh quotient* of $\mathbf{y}$ over $\mathscr{L}$:

$$\mathscr{R}_{\mathscr{L}}(\mathbf{y}) = \frac{\mathbf{y}^T \mathscr{L} \mathbf{y}}{\mathbf{y}^T \mathbf{y}} = \frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T D \mathbf{x}}. \tag{4.4}$$

In fact, minimizing $\mathscr{R}_{\mathscr{L}}(\mathbf{y})$ is closely related to minimizing $q(C_1, C_2)$ in Problem 2, since the two terms bound each other by constant factors:

$$q(C_1, C_2) \leq \mathscr{R}_{\mathscr{L}}(\mathbf{y}) \leq 4\, q(C_1, C_2). \tag{4.5}$$

If we relax $\mathbf{y}$ to be continuous, i.e., $\mathbf{y} \in \mathbb{R}^n$, then by Courant-Fischer Theorem, we have that minimizing $\mathscr{R}_{\mathscr{L}}(\mathbf{y})$ is equivalent to finding the eigenvector corresponding to the smallest eigenvalue of $\mathscr{L}$. We denote $(\lambda_1, \mathbf{v}_1)$ as the eigenvalue and eigenvector pair. The pair $(\lambda_1, \mathbf{v}_1)$ reveals important structural information of G. For example, if G is balanced, $\lambda_1 = 0$ and we can infer a good partitioning from the entry signs in $\mathbf{v}_1$. In the general case, the smaller λ_1 is, the closer G is to be balanced.

We end the introduction to signed spectral graph theory now. We will revisit some of the concepts introduced here in later sections. Curious readers may refer to a survey about spectral theory on signed graphs by Gallier [34].

4.4 Finding polarized communities

Real-world signed networks can be huge, up to millions of nodes. It is unrealistic to assume such graphs are polarized, that is, good partitions of the graphs exist. Instead, we ask: can we find a *polarized subgraph*, which contains only a subset of all nodes in the graph? We define our second problem:

Problem 3. *Given a signed graph $G = (V, E^+, E^-)$, find two disjoint sets of nodes (C_1, C_2) such that $C_1 \cup C_2 \subseteq V$ and (C_1, C_2) maximizes some quality function*

$$q(C_1, C_2).$$

We call (C_1, C_2) a *polarized community*, and C_1 and C_2 are the *bands* in this community. Intuitively, the value of $q(C_1, C_2)$ is large if (1) edges in $E(C_1)$ and $E(C_2)$ are mostly positive, and (2) edges in $E(C_1, C_2)$ are mostly negative. In Figure 4.5, the polarized community (C_1, C_2), highlighted in orange circles, is of high quality, because edges in each band are all positive and edges in between are all negative.

Next we present some definitions of the quality function q. One way to measure the degree of polarization is by counting the number of edges that contribute to a polarized structure. Specifically, we count the number of positive edges inside C_1 and C_2, and the number of negative edges between

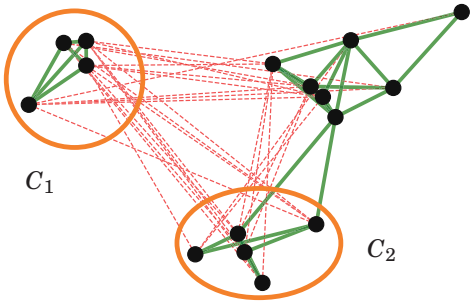


Figure 4.5. A polarized community (C_1, C_2) in *New Guinea Highland Tribes* graph. Each band is highlighted by an orange circle.

C_1 and C_2. Then we have

$$q(C_1, C_2) = \left|E^+(C_1)\right| + \left|E^+(C_2)\right| + 2\,|E^-(C_1, C_2)|. \tag{4.6}$$

However, a drawback of the above definition is that communities with large number of edges are encouraged, since we count the absolute numbers. Larger communities are sometimes considered undesirable than smaller ones, because the former are harder to analyze, visualize or digest by humans. To encourage communities with smaller sizes, we can penalize solutions that are large by normalizing Equation 4.6 with the community size. Then, we have our second definition of q:

$$q(C_1, C_2) = \frac{\left|E^+(C_1)\right| + \left|E^+(C_2)\right| + 2\,|E^-(C_1, C_2)|}{|C_1| + |C_2|}. \tag{4.7}$$

The third definition of q requires that (C_1, C_2) should be isolated from the remaining graph, that is, $|E(C_1 \cup C_2, V \setminus (C_1 \cup C_2))|$ is small. This intuition is a common choice for community definitions in unsigned graphs. And we have:

$$\beta(C_1, C_2) = \frac{2\left|E^+(C_1, C_2)\right| + |E^-(C_1)| + |E^-(C_2)| + |E(C_1 \cup C_2, V \setminus (C_1 \cup C_2))|}{vol(C_1 \cup C_2)}, \tag{4.8}$$

where we count the number of edges that *undermines* the degree of polarization. Thus, we want to *minimize* this measure. Further, this measure is normalized by the total number of edges adjacent to $C_1 \cup C_2$. The measure of β is referred as *signed bipartiteness ratio* by Atay and Liu [3]. The smaller that $\beta(C_1, C_2)$ is, the more polarized that (C_1, C_2) is. Finally, we define q accordingly:

$$q(C_1, C_2) = 1 - \beta(C_1, C_2). \tag{4.9}$$

In fact, finding polarized subgraphs in signed graphs is a nascent topic. Existing works typically focus on optimizing specific definitions of q. For

example, Chu et al. [22] considers optimizing a variant of Equation 4.6. In addition, they allow different bands to overlap in nodes but penalize the degree of overlapping. In contrast, the work by Bonchi et al. [12] considers maximizing Equation 4.7. In contrast, Atay and Liu [3] aims at finding the (C_1, C_2) that minimizes $\beta(C_1, C_2)$.

4.5 Searching for polarized communities

Next we consider the scenario where there are more than one polarized community in a signed graph. And we assume that we know a few nodes that belong to a polarized community (C_1, C_2). We call such nodes *seed nodes*, or *seeds* in short. The question is: can we find a polarized community(C_1, C_2) that is relevant to the seeds? We state our third problem:

Problem 4. *Given a signed graph* $G = (V, E^+, E^-)$ *and two sets of seeds* S_1 *and* S_2*, find two disjoint sets of nodes* (C_1, C_2) *such that* $S_1 \subseteq C_1, S_2 \subseteq C_2$ *and* (C_1, C_2) *maximizes some quality function on the degree of polarization*

$$q(C_1, C_2).$$

We illustrate the problem setting in Figure 4.6. In our example, we show two seeding scenarios. In Figure 4.6(a), setting $S_1 = \{a\}$ and $S_2 = \{b\}$ produces a polarized community, in which the two bands are drawn in red and blue, respectively. In Figure 4.6(b), setting $S_1 = \{a\}$ and $S_2 = \{c\}$ produces a second polarized community, in which the two bands are drawn in red and green, respectively. Note that the two polarized communities in this example are *overlapping* in the red-colored nodes.

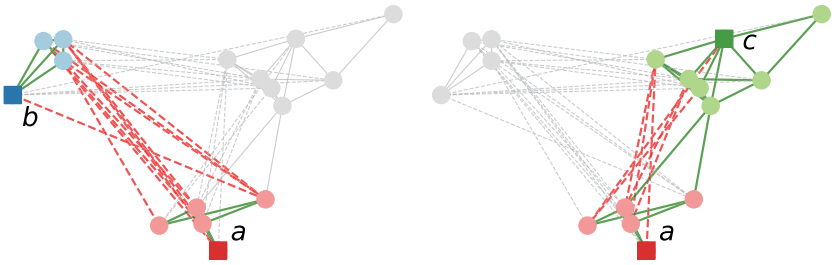
For the definition of q, we can borrow the definitions described in the previous section. However, compared to Problem 3, the additional constraint related to the seeds makes Problem 4 different. In fact, Problem 3 is a special case of Problem 4 if we consider $S_1 = S_2 = \varnothing$.

Note that the above problem falls into a broader class of problems, called *community search problems*. In a typical community search problem, we are given a graph and some seeds (e.g., nodes), and we are asked to find a community (e.g., subgraph) that contains the seeds and optimizes some community quality function.

In the next section, we formalize Problem 4 and describe our solution to it.

4.6 Thesis contribution

In Publication III, we consider a problem that is an instantiation of Problem 4. Specifically, given a pair of node set (S_1, S_2) as seeds, and an integer k, we seek to find a polarized community (C_1, C_2) such that:



(a) Seeds $S_1 = \{a\}$ and $S_2 = \{b\}$ (b) Seeds $S_1 = \{a\}$ and $S_2 = \{c\}$

Figure 4.6. Finding polarized communities under two seeding scenarios, each asking for a different polarized community. The seed nodes are drawn in squares. And each band is drawn in a different color. The *New Guinea Highland Tribes* graph is shown.

1. (C_1, C_2) minimizes $\beta(C_1, C_2)$ (Equation 4.8);

2. $S_1 \subseteq C_1$ and $S_2 \subseteq C_2$;

3. $\frac{vol(C_1 \cup C_2)}{vol(S_1 \cup S_2)} \leq k$;

The last constraint upper bounds the solution size in terms of the number of edges.

Our contribution is related to three topics: (1) finding polarized subgraphs (discussed in Section 4.4); (2) community search problems (mentioned in Section 4.5); and (3) spectral graph theory (introduced in Section 4.3). Our problem definition is relevant to the first and second topics. From the theoretical aspect, our algorithm relies on results from the third topic. In return, our results enrich the studies of all these topics.

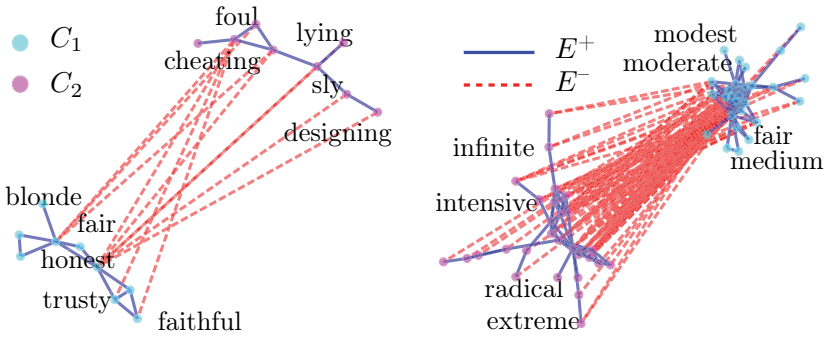
Theoretically, our contribution is two-fold. First, our proposed algorithm has an approximation ratio of $\mathcal{O}(\sqrt{\text{OPT}})$, where OPT is the optimal value. Second, our algorithm runs in time $\mathcal{O}(m \log m)$, which makes it scalable to large graphs.

We evaluate our algorithm on both synthetic and real-world datasets and compare it with a state-of-the-art method due to Chu et al. [22]. We find that our algorithm outperforms the baseline in most of the settings.

In addition, we demonstrate our algorithm's usefulness in finding overlapping communities on real graphs. We consider a graph of English words represented as nodes and the sign of an edge represents either the synonym (e.g., "happy" vs "joyful") or antonym (e.g., "happy" vs "sad") relations between the corresponding words. The edge weights represent the strength of similarity, in which case the edge sign is positive, or the strength of dissimilarity, in which case the edge sign is negative.

Inspired by the phenomena of polysemy (words with more than one meanings) in natural languages, we consider the word "fair" and two of its meanings, "without cheating" and "not excessive". For each meaning,

we seed our algorithm with words of both similar and opposite meanings. The goal is to find antonyms and synonyms of *fair* under each of the two meanings. The result is illustrated in Figure 4.7. It shows that our method finds meaningful antonyms and synonyms for either meaning of fair.



(a) *fair* as *without cheating* e.g., a *fair* game

(b) *fair* as *not excessive* e.g., *fair* amount of time

Figure 4.7. Two polarized communities centered on *fair* in a signed graph of English words. Seed nodes are: (a) $S_1 = \{fair, honest\}, S_2 = \{cheating\}$; (b) $S_1 = \{fair, modest\}, S_2 = \{extreme\}$;

Besides, we test our method's scalability on a graph with 1 million nodes and the algorithm takes 18 seconds on average to find a polarized community.

5. Fair clustering

In this chapter, we present a novel way of clustering data in a fair manner. We first describe how bias circulates through machine learning systems. Then we discuss the question of what fairness is and illustrate the inherent complexity of this question. Next we describe common ways to achieve fairness in classification, which has received considerable attention in the study of algorithmic fairness. Later, we discuss an important topic of machine learning — clustering, summarize existing works on fair clustering and reason about their limitations. Finally, we describe a new way of fair clustering to address the limitations of some of the existing approaches.

5.1 Bias in machine learning systems

We give motivation on the study of bias issues in machine-learning systems in Section 1.1. However, we haven't understood how bias arises in the first place. To see how, we use a diagram from the book by Barocas et al. [7] (Figure 1 in Chapter 1 of the book). The diagram is drawn in Figure 5.1 and depicts the main stages of machine-learning systems. The remaining paragraphs are structured by the stages in the diagram.

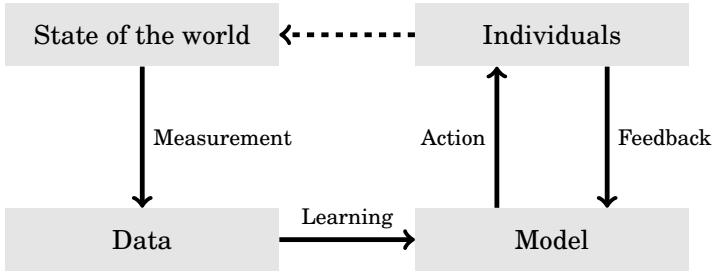


Figure 5.1. Main stages (corresponding to the arrows) in machine-learning systems.

Measurement. The first stage is about creating the datasets to be used by downstream machine-learning applications. This stage essentially reduces the state of the world into a dataset (e.g., texts, images, or a table of values).

Although the term "measurement" indicates an objective process, it can be full of subjective decisions.

First, data that is about humans and is collected by humans naturally encodes human bias. Examples are historical books, which are rich in cultural prejudices and stereotypes.

Second, our world is complicated and measuring it can be both subjective and challenging. Let us consider the task of determining the racial identity of a person. This task can be trickier than imagined. For instance, if the person's parents have different racial backgrounds (e.g., Hispanic and Asian), what racial group should they consider to be? In addition, our perception of racial identity changes over time. The racial categories adopted by the U.S. Census in 1790 contain only three choices: "free white males/females", "all other free person" and "slaves".[1] While in 2020, fine-grained categorization is used, including 12 different racial groups.

Third, data imbalance problem, in which data related to certain groups (e.g., female) is much scarcer compared to that of other groups (e.g., male), may contribute to further bias. Imagine a project that studies the contributing factors of career successes of machine-learning researchers of different genders. The fact that only 12 percent[2] of machine-learning researchers are females makes females under-represented. Further, if only a limited number of samples are obtained and are drawn uniformly randomly, conclusions about female researchers can be inaccurate, even misleading.

Learning. The second stage learns from the data collected at the first stage. Learned models are produced, usually in the form of a set of parameter values. This stage is also called "training".

In general, it is hard to filter out the bias present in the data. On the one hand, data to be used for model training may contain knowledge that is useful for the tasks at hand. On the other hand, bias, which is encoded in the data, can be inherited by the learned models. However, there is no general way for learning algorithms to separate the useful knowledge from the bias.

Buolamwini and Gebru [14] study the issue of gender imbalance in an image classification task. In this task, algorithms are trained to identify the gender of humans based on images of their faces. Buolamwini and Gebru found that darker-skinned females are the most misclassified groups. This conclusion is based on the classification results from three commercial face-recognition systems. One possible reason for the unfair results is that the training datasets used by the face-recognition systems are overwhelmingly composed of lighter-skinned subjects, creating the

[1] https://www.pewresearch.org/fact-tank/2020/02/25/the-changing-categories-the-u-s-has-used-to-measure-race/

[2] https://www.wired.com/story/artificial-intelligence-researchers-gender-imbalance/

problem of data imbalance.

Another example of bias in learning is word embedding. A *word-embedding* model converts words to low-dimensional vectors. Given a trained embedding model, we can measure the association between two words by calculating the similarity between the corresponding vectors. In a study by Bolukbasi et al. [11], the word "she" is most similar to occupations such as "homemaker" and "nurse", while the word "he" is most similar to "maestro" and "skipper" (captain). In other words, gender bias in human languages is inherited by word-embedding models.

Action. Almost all machine-learning systems are designed to produce some impacts on the real world. There is evidence that biased algorithms can potentially create adverse impacts on the life of a certain demographic group.

A study by Datta et al. [24] shows that Google's advertisement recommendations are potentially discriminatory by gender. Google uses recommendation engines to display personalized ads in the search results, based on users' search queries and their demographic information (e.g., gender, age). In their analysis, male users are more likely to see high-paying jobs than female users do. The unfairness in ad-recommendation results may further contribute to gender inequality in income.

Unfairness caused by machine-learning algorithms is also observed in justice systems. An algorithm called COMPAS was used across the United States to assess criminal defendants' likelihood of becoming recidivists – criminals who re-offend. Larson et al. [54] tested whether COMPAS was biased against certain groups. They found that "black defendants were far more likely than white defendants to be incorrectly judged to be at a higher risk of recidivism, while white defendants were more likely than black defendants to be incorrectly flagged as low risk.".

Feedback. Feedbacks are often used by machine learning systems to refine the models. For example, a search engine may collect user clicks on search results and use them to improve the quality of future search results. However, using biased feedbacks for model refinement can exacerbate the bias that already exists, finally creating a negative feedback loop.

Furthermore, interpreting feedbacks correctly is challenging. Take user clicks on search results as an example, do the clicks indicate the clicked pages are interesting or simply because the pages are on top of the search results? If feedbacks are handled inappropriately, bias can accumulate, which can in turn make negative impacts.

To summarize, bias permeates through machine-learning systems. To reduce bias, we should first understand at which stage of the systems bias arises and how it arises.

5.2 Codifying fairness

There are mainly two lines of researches in the topic of algorithmic fairness — (1) codifying what it means for an algorithm to be fair; (2) designing algorithms that achieve certain notions of fairness. In this section, we focus on the former — codifying the notions of fairness.

What is fairness? It is a deep and fundamental question, which has been at the center of debates among the great minds since long ago. It never has easy answers. And we do not make the attempts to answer it here. Instead, we aim to illustrate the complexity of this question and to look closer at it in different contexts, hoping to gain a deeper understanding. To do so, we argue that appropriate definitions of fairness depend on both *our perception of fairness* and the *application domains*.

Our perception of fairness. To illustrate how the notion of fairness varies with our perception of it, we use an example of college-admission processes. Each year, colleges select students for admission, based on the candidates' qualifications such as GPA, scores of standard tests, and their motivation. However, this process is not always fair among different gender groups. To promote gender equality, different colleges may take different approaches. One approach is to require that gender-specific acceptance rates are equal, that is, female candidates have an equal probability of being accepted as male candidates. This notion of fairness is often referred to as *statistic parity*. A drawback of statistic parity is that it ignores the correlation between the candidates' gender and their probability to succeed. For example, it could be the case that being female is more positively *correlated* with future career success than being male.[3] Thus, it is acceptable to have acceptance rates that differ by gender. Nevertheless, using gender information has the risk of abusing it, therefore, the use of it should be *well-justified*. We will formalize the above intuitions of fairness in the next section.

Application domains. Even for the same notion of fairness, its exact formalization can vary by the application domains. The aforementioned college-admission process is an example of a classification problem, in which we seek to assign a *label* (e.g., accepting for admission or rejecting) to an *object* (e.g., a student candidate). However, there are many other problems in machine learning in which the discussion of fairness is important.

For example, in many online platforms (e.g., Amazon, Facebook), users are recommended items (e.g., books, news articles) that they might like, by recommendation algorithms. Researchers have found that many recommendation algorithms can produce biased results for different demographic groups [48, 90]. To make recommendations fairer, Kamishima et al. [48]

[3]Note that we do not say candidates' gender *causes* their success.

propose to use the notion of statistical parity, which we introduced before. Though copying the definition of statistical parity in classification to the recommendation setting is tempting, doing so rarely works due to the different nature of the two problems. In other words, an instantiation of statistical parity that is suitable for recommendation problems is required. To achieve this goal, Kamishima et al. [48] proposes to penalize the differences of predicted item ratings among different demographic groups.

To summarize, for the same problem, the notion of fairness may differ, depending on what we want to achieve. And even the definition of the same fairness notion may differ by the application domains.

5.3 Fairness in classification problems

Classification is a fundamental topic in machine learning. Next we present common fairness notions in classification problems. Though the study of algorithmic fairness is nascent, fairness research in classification problems a received considerable amount of attention. Therefore, we feel the need to discuss this topic in detail. In the following, we first formalize the problem of classification. Then we present three common fairness notions in classification.

Classification problems. In a classification problem, we want to train a *classifier* using observed data and use the trained classifier to make predictions. Suppose we observe n data points $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where x_i is an *input data point* (e.g., text document, image) and y_i is the *label* of x_i (e.g., topic of a document, main object in an image). Each input x_i is usually represented by a vector or a matrix and the label y_i takes value from a finite set. We use random variables X and Y to denote some data point and the label of the data point, respectively, and we assume that the observed data D is drawn randomly from some unknown probability distribution on (X, Y). In other words, an observation (x, y) can be seen as a sample drawn from the probability distribution on (X, Y). In the sequel, we consider the special case that the variable Y is binary and its value is either positive or negative, i.e., $Y \in \{-1, 1\}$. That is to say, we focus on *binary classification problems*. We use the convention that $Y = 1$ is considered more favorable than $Y = -1$, e.g., $Y = 1$ means "being granted a loan" and $Y = -1$ means "not granted a loan".

Essentially, a classifier f is a function, often represented by a set of parameters, whose values are learned using the observed data D. The function f takes in an input data point x and outputs a *prediction* $\hat{y}$, i.e., $\hat{y} = f(x)$. Assume that we know y, which is the true label of x. If $\hat{y} = y$, we say the prediction by f is *correct*. If $\hat{y} \neq y$, the prediction is *incorrect*. We use the variable $\hat{Y}$ to denote the prediction of a classifier. And we express

that $\hat{Y} = f(X)$.

As an example of classification problems, we consider the problem of deciding whether a loan applicant should be granted a loan or not. The variable X encodes the applicants' background information such as age, gender, income level, marital status, and past credit records. The variable Y denotes whether he will default a loan or not in the future, which determines whether he should be granted a loan or not. If the applicant will default a loan, we set $Y = -1$. Otherwise, we set $Y = 1$. Given some classifier f under this setting, the variable $\hat{Y}$ denotes the classifier's prediction of whether a random applicant (represented by X) will default a loan or not, that is, $\hat{Y} = f(X)$. Suppose we know the background information of a person and his loan history, we have an observation, (x, y), where x is a vector representing the background and y indicates whether he has defaulted a loan or not. Further assume that we collected a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$ of historical loan data of multiple persons, we can use it to train a classifier f. During the prediction phase, the classifier takes an input x representing a loan applicant's background and outputs $f(x)$, the prediction of whether he will default a loan or not.

When does unfairness arise? To answer this question, we need to know common ways to measure the performance of a classifier. Intuitively, we want $\hat{Y}$ to match Y as much as possible. To quantify the degree of match, one measure is *accuracy*, which defines the probability of a classifier to make a correct prediction. Formally, we can express it as $Pr\left(Y = \hat{Y}\right)$. A classifier is considered good if $Pr\left(Y = \hat{Y}\right)$ is high. Alternatively, we can use *precision* $Pr\left(Y = 1 \mid \hat{Y} = 1\right)$, the probability of making a correct prediction if the predicted label is positive. Or we can use *recall*, $Pr\left(Y = 1 \mid \hat{Y} = 1\right)$ and $Pr\left(\hat{Y} = 1 \mid Y = 1\right)$, the probability of making a correct prediction if the actual label is positive.

The issue of bias arises when we incorporate *protected attributes* (e.g., gender, race) and distinguish the classifier's performance based on the values of the protected attributes. Consider the loan-applicant example and using race as the protected attribute. If a classifier makes prediction such that the accuracy of different racial groups differ, e.g.,

$$Pr\left(\hat{Y} = Y \mid \text{the applicant is black}\right) < Pr\left(\hat{Y} = Y \mid \text{the applicant is white}\right).$$

One might argue that the classifier is unfair because the accuracy on white applicants is higher than the accuracy on black applicants.

Next we give a few examples of fairness notions for classification problems. We use the random variable A to denote the protected attribute and thus, the data points can be grouped by the values of the associated protected attributes, e.g., black applicants and white applicants. For simplicity, we assume A is binary, i.e., $A \in \{\mathsf{a}, \mathsf{b}\}$.

Statistical parity. A classifier f satisfies *statistic parity* if members in one group have an equal probability of receiving positive prediction as

members in the other group. This notion is also known as *demographic parity* and *independence*. Formally, this fairness notion requires a classifier to satisfy:

$$Pr\left(\hat{Y}=1 \mid A=\mathrm{a}\right)=Pr\left(\hat{Y}=1 \mid A=\mathrm{b}\right).$$

Using the loan-application example and assume that $A \in \{\text{black}, \text{white}\}$, statistical parity says that black applicants should have an equal probability of receiving positive prediction as white applicants.

Statistical parity has two major limitations. First, it rules out the possibility that A is *correlated* with Y. For instance, racial background can be correlated with the tendency to default loans [42]. Ignoring such correlation can harm the performance of the classifier. Second, it promotes *laziness*. One can achieve statistical parity easily by randomly choosing the same fraction of members from each protected group, regardless of other information.

Separation. The notion of *separation* (also called *equal odds*) proposed by Hardt et al. [42] aims at addressing the above limitations of statistical parity.

This notion allows the protected attribute A to be correlated with $\hat{Y}$, only if the correlation is *justified* by Y. Formally, separation requires:

$$Pr\left(\hat{Y}=1 \mid Y=1, A=\mathrm{a}\right)=Pr\left(\hat{Y}=1 \mid Y=1, A=\mathrm{b}\right),$$
$$Pr\left(\hat{Y}=1 \mid Y=0, A=\mathrm{a}\right)=Pr\left(\hat{Y}=1 \mid Y=0, A=\mathrm{b}\right).$$

One way to interpret separation is by considering *true positive rate* and *false positive rate*, defined as $Pr\left(\hat{Y}=1 \mid Y=1\right)$ and $Pr\left(\hat{Y}=1 \mid Y=0\right)$, respectively. This criterion requires the same true positive rates and the same false positive rates among different groups.

Sufficiency A third notion, known as *sufficiency*, states that the prediction subsumes the protected attribute information, for the purpose of prediction. Sufficiency states the following:

$$Pr\left(Y=1 \mid \hat{Y}=1, A=\mathrm{a}\right)=Pr\left(Y=1 \mid \hat{Y}=1, A=\mathrm{b}\right)$$
$$Pr\left(Y=1 \mid \hat{Y}=0, A=\mathrm{a}\right)=Pr\left(Y=1 \mid \hat{Y}=0, A=\mathrm{b}\right)$$

An important question is how to achieve any of the aforementioned fairness criteria. Broadly speaking, there are three approaches: via (1) *preprocessing*, e.g., removing the bias from input data; (2) *training* (or *inprocessing*), e.g., training the classifiers in a specific way such that its predictions are fair; or (3) *post-processing*, e.g., treating any classifier as a black-box and adjusts the predictions according to some fairness criterion.

In general, the study of algorithmic fairness is an interesting and ongoing topic. Interested readers might refer to the book by Barocas et al. [7] for more information.

5.4 Fair clustering

In this section, we consider clustering, which is an important topic in machine learning, and give an overview of fairness study in clustering. In Section 2.3, we give some background information in clustering. For current research on fair clustering, a dominant fairness notion being considered is referred to as *representation-based fairness*.

Representation-based fair clustering. Pioneering work in this direction is by Chierichetti et al. [21]. They consider optimizing standard clustering objectives such as the k-means objective and the k-center objective. In addition, they assume each data point is associated with one of two pre-specified labels. In other words, the data points can be partitioned into 2 disjoint groups, in which each group contains the points of one label. The notion of fairness is incorporated as *representation-based constraints* in the optimization problems, e.g., by requiring that in each output cluster the different data groups should be represented within pre-specified proportions. In Figure 5.2, we illustrate this fairness notion using an example, which is borrowed from Chierichetti et al. [21]. In this example, data points are grouped by the colors (purple and pink), into $\{x, y, z\}$ and $\{a, b, c\}$. An intuitive non-fair clustering ignoring the group information is $\{x, y, z\}$ and $\{a, b, c\}$. Clustering concerning representation-based fairness partitions the points by the dashed line, into $\{x, a, b\}$ and $\{y, z, c\}$. In each cluster of this clustering, the colors of the data points are more balanced, compared to the non-fair clustering.

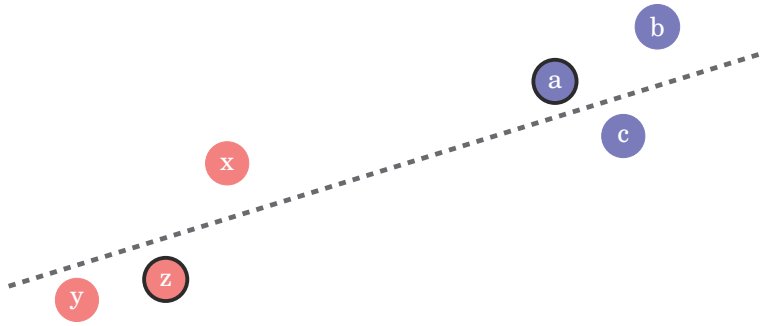


Figure 5.2. A toy example illustrating representation-based fair clustering. The dashed line indicates a clustering that adheres to the representation-based fairness notion. Nodes are grouped by the colors and the cluster centers are highlighted in thick black border.

We use two examples to motivate representation-based fairness. Consider

clustering news articles. In addition, each article has a label indicating its political leaning (e.g., demographic or republic). Representation-based fairness requires that within each cluster, different political views are balanced, that is, no political view is over-represented nor under-represented. A second example is about clustering experts and experts are labeled by their genders. Representation-based fairness translates to gender balance inside each cluster.

Most subsequent works [1, 2, 5, 8, 9, 50, 72] on fair clustering build upon the work by Chierichetti et al. [21]. They propose variant notions of representation-based fairness. These variant notions can be written in a general form as follows.

Let C_f denotes a cluster f and G_i denotes the ith color group, then $G_i \cap C_f$ is the set of points in group G_i that are in cluster C_f. In addition, group-specific parameters α_i and β_i ($0 \leq \alpha_i \leq \beta_i \leq 1$) are given as inputs. Then in general, any representation-based fairness constraints can be written as:

$$\alpha_i \leq \frac{|G_i \cap C_f|}{|C_f|} \leq \beta_i \quad \text{for any cluster } f \text{ and any group } i. \tag{5.1}$$

Intuitively, setting α_i high prevents the group i from being *under-represented*, while setting β_i low prevents group i from being *over-represented*. The values of α_i and β_i can be set to fine-tune the exact notion of fairness. For instance, Rösner and Schmidt [72] set $\alpha_i > 0$ and $\beta_i = 1$, which only prevents under-representation of all groups. In contrast, Ahmadian et al. [1] aim at preventing over-representation by setting $\alpha_i = 0$ and $\beta_i > 0$. Bera et al. [8] studies the more general case, in which both α_i and β_i are adjustable.

A limitation of representation-based fairness. One observation on representation-based fair clustering is that points may not be assigned its closest center. For example in Figure 5.2, a fair clustering assigns point x to a, however z is its closest center.

This property is undesirable in some practical scenarios. Consider a city council that plans to construct a number of facilities (e.g., hospitals) to serve its citizens. The representation-based fair clustering framework would assign citizens to facilities so that all demographic groups are proportionally represented in each cluster. To achieve this constraint, some citizens may not be assigned to their closest facility, causing them an inconvenience. Why would citizens have to go to facilities that are further to the closest ones?

In the next section, we address the above limitation using a different fairness notion.

5.5 Thesis contribution

We propose a different fairness notion for clustering. Consider again the scenario in which a city council plans to construct a number of facilities to serve its citizens. Given a set of opened facilities, our proposed fairness notion assigns any citizen to their closest facilities, therefore addressing the aforementioned limitation of representation-based fair clustering. In our setting, for a demographic group, we measure the *group-level cost* as the average distance of data points in that group to their closest facilities. Our objective seeks to minimize the maximum group-level cost among all groups, that is, no group incurs too high a cost.

We refer to our framework as *distance-based fair clustering*, or in short, D-fair clustering. In contrast, we refer to representation-based fair clustering as R-fair clustering.

Next we formalize our problem definition. The data to be clustered is denoted by a set C, and a metric d is used to measure the distance between any pair of points in C. Data points in C are associated with some protected attribute and can be partitioned into ℓ disjoint groups, $G_1,\ldots,G_\ell$, such that $C = \bigcup_{i=1}^{\ell} G_i$ and $G_i \cap G_j = \varnothing$ for $i \neq j$.

Given some $j \in C$ and some $S \subseteq C$, we define:

$$d(j,S) = \min_{f \in S} d(j,f).$$

Given $S \subseteq F$ as the candidate centers, we define the *average cost* of a group G_i with respect to S as

$$\Phi_{avg}(G_i,S) = \frac{1}{|G_i|} \sum\nolimits_{j \in G_i} d(j,S). \tag{5.2}$$

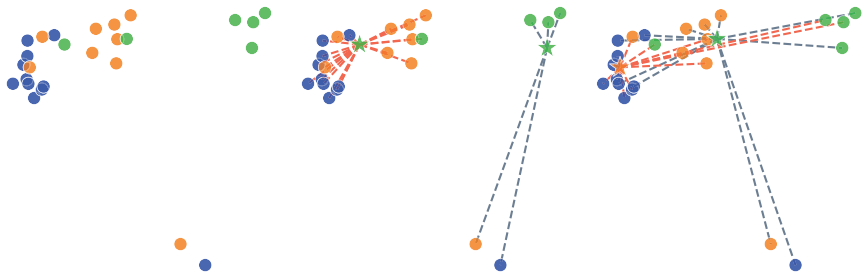
We state our problem below:

Problem 5. *(D-fair k-median) Given a set of n clients C, a set of m facilities F, a distance function d on $C \cup F$, and a partition of C into ℓ groups $\{G_1,\ldots,G_\ell\}$, we seek to find a set $S \subseteq F$ of at most k points so as to minimize the cost*

$$\Phi_{fair}(S) = \max\nolimits_{i \in [\ell]} \Phi_{avg}(G_i,S). \tag{5.3}$$

D-fair k-median is a generalization of k-median and k-center problem, both of which are **NP**-hard problems. Thus, D-fair k-median is an **NP**-hard problem.

In Figure 5.3, we use a toy example to illustrate the difference between our problem formulation and a R-fair formulation. In this example, data points shown in Figure 5.3(a) are partitioned into three groups, indicated by the node color. We consider the case $C = F$ (each data point is a candidate center) and we set $k = 2$. The optimal clustering under D-fair and R-fair formulations are shown in Figure 5.3(b) and Figure 5.3(c), respectively. Cluster centers under each formulation are shown as stars, and points



(*a*) Input (*b*) D-fair clustering (*c*) R-fair clustering

Figure 5.3. Optimal solutions under D-fair and R-fair clustering.

are connected to their centers by either red (1st cluster) or gray lines (2nd cluster). Note that D-fair clustering assigns each node to its closest center. In contrast, R-fair clustering does not necessarily do so because of the representation-based fairness constraints.

To solve the proposed problem, we design two approximation algorithms. The first algorithm uses k-median k-center objectives to bound the objective in D-fair k-median, respectively. Then it employs blackbox k-median solvers and blackbox k-center solvers to approximate D-fair k-median. The algorithm makes further improvement using a local-search subroutine. We name this algorithm LS (the acronym of "local search") We show that the algorithm LS achieves an approximation ratio of $\mathcal{O}(\min\{\ell, s_{\max}\})$, where ℓ is the number of groups and $s_{\max}$ is the maximum group size.

The second algorithm D&C employs a divide-and-conquer approach. The algorithm D&C consists of two phases. First, it divides the problem into ℓ sub-problems, each of which corresponds to a k-median problem instance on points from a different color group. And the algorithm optimizes $\Phi_{avg}(C_i, S)$ for each group i. Second, it aggregates the opened facilities from the previous step into F', solves a k-center problem instance on F' and returns the solution. We show that the algorithm D&C yields an approximation ratio of $\mathcal{O}(k)$.

To get an intuitive feeling of how D-fair methods perform in practice, we use a real-world dataset consisting of local communities in Los Angeles, United States. Each data point corresponds to a local community and the group label of a data point is determined by the income level of the residents in the corresponding community. We end up with 6 groups, representing 6 income groups. We compare our algorithm LS with two standard clustering algorithms under the k-median and k-center formulations, respectively. We name the baselines kM and kC, respectively. We set $k = 3$ and plot the clustering results in Figure 5.4, with the objective value show on top of each subfigure. All data points are assigned to their closest opened facilities. The algorithm kC opens facilities on the periphery because of the definition of the k-center objective. In contrast, the locations

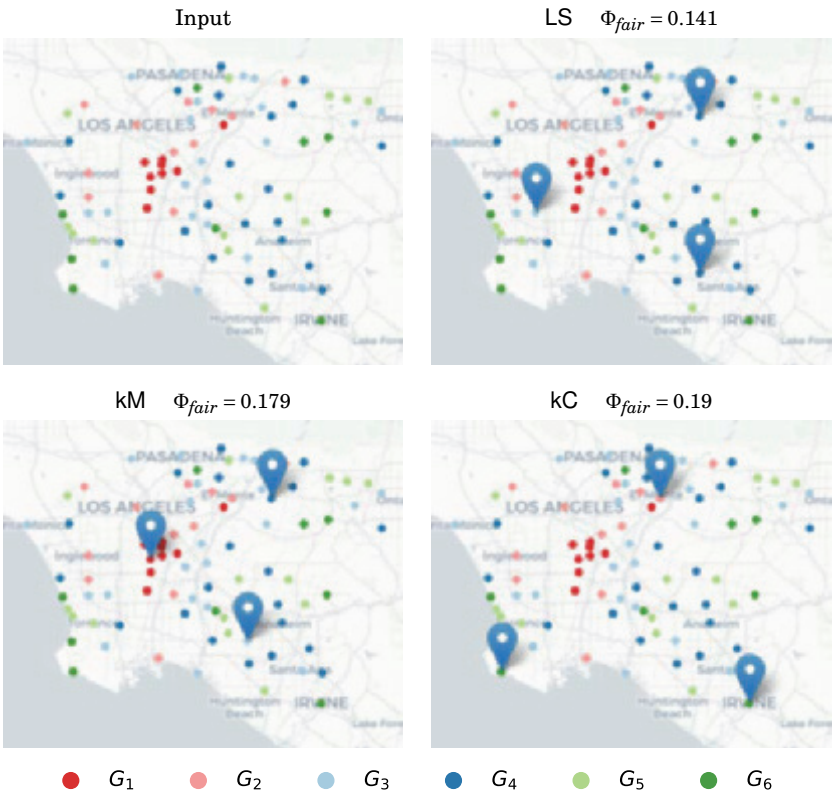


Figure 5.4. Facilities opened by LS, kM and D-kC in Los Angeles ($k = 3$): each data point maps to a local community. Communities are partitioned into 6 groups. Euclidean distance is used. We made this plot for illustration purpose only.

of the facilities opened by kM are more similar to that by LS, compared to kC. Nonetheless, the facilities opened by LS spread out more than those opened by kM. For example, kM opens a facility in the middle of the red points (G_1 and G_2), while LS opens one facility closer to the left, to accommodate the green points on the left (G_5 and G_6). We do not show the results by R-fair methods because clients are not necessarily assigned to their closest facilities and thus, it produces counter-intuitive results.

We also conduct quantitative evaluations to demonstrate our methods' capability in achieving our proposed notion of fairness. We consider three real-world datasets denoted as *EURLex*, *Wiki*, and *Amazon*. In the first two datasets, each data point represents a text document and group labels are determined by the topics of the documents. Both datasets contain 2 400 data points. The third dataset contains review data for Amazon products. Each point represents a review record and the group label corresponds to the product category. The dataset contains 1.2 million data points.

We compare the algorithm D&C with two state-of-the-art R-fair methods [8], which are denoted by R-kM and R-kC and optimize the k-median

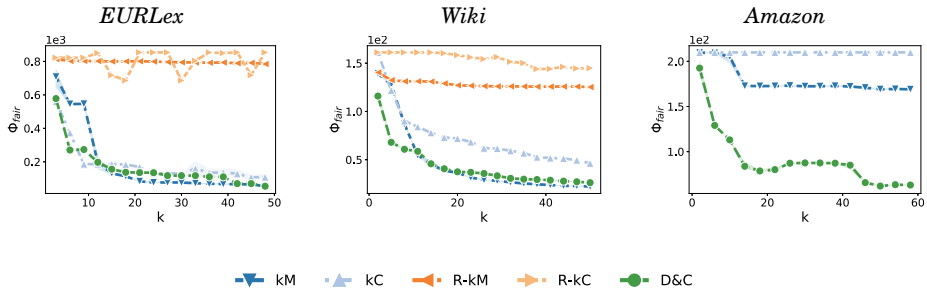


Figure 5.5. Comparison of the algorithm D&C, R-fair methods and vanilla clustering methods on three real-world datasets.

objective and the k-center objective, respectively. We also compare with vanilla k-median and k-center clustering algorithms, named kM and kC, respectively, which do not consider any fairness notions.

The evaluation metric we use is Φ_{fair}, which is the loss function in Equation 5. We consider clustering different k values and the results are plotted in Figure 5.5. It can be seen that the algorithm D&C is consistently among the best, while R-fair methods perform much worse in general. Note that R-kM and R-kC fail to scale up to large data sets such as *Amazon*.

6. Conclusions

Innovations created by us, humans, improved our life and society fundamentally. Meanwhile, these innovations brought about unexpected changes that harm both ourselves and our surroundings. In this thesis, we study three problems among the many, in an attempt to make our society a better one. The problems are: *epidemic & infodemic*, *online polarization* and *bias in automatic decision-making*. All of them have strong connections with the online digital world, while the last problem has its root in the physical world.

To study these problems rigorously, we model them mathematically.

- We study the problem of reconstructing propagation processes on networks, which can facilitate the early detection of epidemic & infodemic outbreak. We formulate two different problems based on Steiner trees, one leverages node infection time (Publication I) and the other makes probabilistic predictions (Publication II).

- We study online polarization in the context of social networks and investigate the problem of searching polarized communities in signed graphs, given input seeds (Publication III);

- To reduce bias in automatic decision-making, we focus on data clustering, propose a novel notion of fairness for clustering and design clustering algorithms that adhere to our proposed fairness notion (Submitted Article IV).

To solve our proposed problems, we design novel algorithms and analyze their theoretical properties. We extensively test the empirical performance of our algorithms on both synthetic and real-world graphs and compare them against strong baselines.

6.1 Challenges and future work

Our proposed problem formulations and algorithms make several assumptions, which are not necessarily true in practice. In this section, we discuss these assumptions, illustrate their limitations, and give directions for future work. We focus on each task separately.

Reconstructing propagation processes. One of the main challenges is data availability. Our proposed methods assume that the underlying contagion networks are available and are free from noise. In practice, high-quality networks are usually difficult to obtain. For example, human contacts often do not leave any traces that can be recorded easily. Auxiliary technology can be used for graph construction purposes. Wireless sensor network technology, for instance, is used by Salathé et al. [74] to track people's locations, which are further used to infer human-contact history. Still, measurement errors on people's location can potentially undermine the quality of the network structure. Furthermore, the usage of people's location is subject to privacy infringement. Can we design robust data construction methods that are both accurate and privacy-preserving?

Another challenge is the lack of data with ground-truth annotations (the true infection states of individuals in an epidemic). Because of this challenge, we have to test our algorithms on synthetically-generated datasets that contain ground truth. However, such datasets cannot faithfully reflect the reality — properties and patterns in real cascades.

Last, all our algorithms assume that a propagation process has only one source node and can be represented as a tree. However, this is assumption is not always true. For instance, multiple users on a social media platform can start sharing the same piece of information if the information is from some external source (e.g., news websites), therefore creating multiple source nodes of a cascade. In fact, there exist real-world cascade datasets that have multiple sources, such as one dataset constructed from Digg.com.[1]

Searching for polarization in signed graphs. Our study deals with signed graphs in which edges have signs. But edge signs are not always free to obtain; in fact, there are no direct means to infer edge signs in many social networks. Therefore, the difficulty to obtain signed graphs limits the applicability of our method.

Because of this challenge, inferring edge signs in social networks is an active research direction. Examples in this direction include the work by Chiang et al. [19, 20], Hsieh et al. [44] and Javari and Jalili [47]. Notably, many of these works assume that the signs of a small set of edges are available beforehand and they rely on this knowledge to infer signs on other edges. However, the assumption may not be true. So can we infer edge

[1] https://www.isi.edu/ lerman/downloads/digg2009.html

signs without any prior knowledge of edge signs? Moreover, online social networks are usually rich in multi-media information, such as text, images, and videos. Can we leverage these information sources to infer edge signs? A work by Kumar et al. [51] achieves this goal by crowdsourcing. They hired human annotators to label the sentiment of user-generated text on Reddit.com. However, the process is manual and can be costly and time-consuming. So an interesting direction is to automate the labeling process, for example, using machine-learning techniques.

Other directions are related to using different definitions of polarized communities. On the one hand, we consider a specific notion of polarity measure called "signed bipartiteness ratio" [3]. Other polarity measures are known, for example, one proposed by Bonchi et al. [12]. So optimizing other polarity measures is worthwhile trying. On the other hand, our definition of polarized community consists of only two groups (2-way polarized community). However, in practice, more than two groups can be in a polarized community. Thus, searching for k-way polarized community is a possible extension of our work.

Fair clustering. We assume each data point belongs to only one group, in other words, there is a disjoint partitioning over all points. This assumption can fail in practice when a data point belongs to multiple groups. For example, a person can belong to multiple racial groups, or a document can be labeled with more than one topic. Our proposed objective function fails to capture this scenario. Thus, defining an appropriate objective function, which captures the above setting and adheres to our proposed fair notion, is an interesting direction.

A second direction is to improve the approximation ratios of the fair clustering problem proposed by us. Recall that we design two approximation algorithms, which achieve approximation ratios $\mathcal{O}(\min\{\ell, s_{\max}\})$ and $\mathcal{O}(k)$, respectively. However, we do not give any tightness analysis on the approximation ratios. Are the ratios tight? If not, can we design constant-factor approximation algorithms?

Another direction is about combining representation-based fairness notions with our fairness notion. These two fairness notions are orthogonal because one is incorporated as constraints, while the other expressed as the problem objective. So can we combine these two fairness notions into one clustering problem? If yes, can we design approximation algorithms for this new problem?

Last, a more general direction is to develop new fairness notions for clustering. Since the study of algorithmic fairness is relatively nascent and the study of fairness in data clustering is even more recent, compared to what is done for supervised machine learning. Can we, for example, adopt some fairness notions in supervised machine learning to the clustering settings?

References

[1] Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. Clustering without over-representation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 267–275, 2019.

[2] Aris Anagnostopoulos, Luca Becchetti, Matteo Böhm, Adriano Fazzone, Stefano Leonardi, Cristina Menghini, and Chris Schwiegelshohn. Principal fairness:\\removing bias via projections. *arXiv preprint arXiv:1905.13651*, 2019.

[3] Fatihcan M Atay and Shiping Liu. Cheeger constants, structural balance, and spectral clustering analysis for signed graphs. *Discrete Mathematics*, 343(1):111616, 2020.

[4] Robert Axelrod and D Scott Bennett. A landscape theory of aggregation. *British journal of political science*, 23(2):211–233, 1993.

[5] Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. Scalable fair clustering. In *International Conference on Machine Learning (ICML)*, pages 405–413, 2019.

[6] Robert M Baldwin, Emily B Grewal, Ashwin Ravindra Bharambe, and Andrew Chung. Photo clustering into moments, August 9 2016. US Patent 9,411,831.

[7] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

[8] Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4955–4966, 2019.

[9] Ioana O Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R Schmidt, and Melanie Schmidt. On the cost of essentially fair clusterings. In *Approximation, Randomization, and Combinatorial Optimization*, 2019.

[10] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[11] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.

[12] Francesco Bonchi, Edoardo Galimberti, Aristides Gionis, Bruno Ordozgoiti, and Giancarlo Ruffo. Discovering polarized communities in signed networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 961–970, 2019.

[13] Andrei Z Broder. Generating random spanning trees. In *FOCS*, volume 89, pages 442–447. Citeseer, 1989.

[14] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91, 2018.

[15] Roger J Calantone and C Anthony Di Benedetto. Clustering product launches by price and launch strategy. *Journal of Business & Industrial Marketing*, 2007.

[16] Jeff Cheeger. A lower bound for the smallest eigenvalue of the laplacian. In *Proceedings of the Princeton conference in honor of Professor S. Bochner*, pages 195–199, 1969.

[17] Zhen Chen, Hanghang Tong, and Lei Ying. Full diffusion history reconstruction in networks. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 707–716. IEEE, 2015.

[18] Justin Cheng, Lada Adamic, P Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, pages 925–936, 2014.

[19] Kai-Yang Chiang, Nagarajan Natarajan, Ambuj Tewari, and Inderjit S Dhillon. Exploiting longer cycles for link prediction in signed networks. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1157–1162, 2011.

[20] Kai-Yang Chiang, Cho-Jui Hsieh, Nagarajan Natarajan, Inderjit S Dhillon, and Ambuj Tewari. Prediction and clustering in signed networks: a local to global perspective. *The Journal of Machine Learning Research*, 15(1): 1177–1213, 2014.

[21] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, pages 5029–5037, 2017.

[22] Lingyang Chu, Zhefeng Wang, Jian Pei, Jiannan Wang, Zijin Zhao, and Enhong Chen. Finding gangs in war from signed networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1505–1514. ACM, 2016.

[23] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[24] Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings. *Proceedings on privacy enhancing technologies*, 2015(1):92–112, 2015.

[25] Michela Del Vicario, Gianna Vivaldo, Alessandro Bessi, Fabiana Zollo, Antonio Scala, Guido Caldarelli, and Walter Quattrociocchi. Echo chambers: Emotional contagion and group polarization on facebook. *Scientific reports*, 6:37825, 2016.

[26] Zvi Drezner and Horst W Hamacher. *Facility location: applications and theory*. Springer Science & Business Media, 2001.

[27] Dingzhu Du and Xiaodong Hu. *Steiner tree problems in computer communication networks*. World Scientific, 2008.

[28] Nan Du, Le Song, Ming Yuan, and Alex J Smola. Learning networks of heterogeneous influence. In *Advances in Neural Information Processing Systems*, pages 2780–2788, 2012.

[29] Richard Dubes and Anil K. Jain. Clustering methodologies in exploratory data analysis. In *Advances in computers*, volume 19, pages 113–228. Elsevier, 1980.

[30] Ritabrata Dutta, Antonietta Mira, and Jukka-Pekka Onnela. Bayesian inference of spreading processes on networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2215): 20180129, 2018.

[31] Pouya Esmailian and Mahdi Jalili. Community detection in signed networks: the role of negative ties in different scales. *Scientific reports*, 5:14339, 2015.

[32] Mehrdad Farajtabar, Manuel Gomez-Rodriguez, Nan Du, Mohammad Zamani, Hongyuan Zha, and Le Song. Back to the past: Source identification in diffusion networks from partially observed cascades. In *Artificial Intelligence and Statistics*, 2015.

[33] Soheil Feizi, Muriel Médard, Gerald Quon, Manolis Kellis, and Ken Duffy. Network infusion to infer information sources in networks. *IEEE Transactions on Network Science and Engineering*, 6(3):402–417, 2018.

[34] Jean Gallier. Spectral theory of unsigned and signed graphs. applications to graph clustering: a survey. *arXiv preprint arXiv:1601.04692*, 2016.

[35] Nabeel Gillani, Ann Yuan, Martin Saveski, Soroush Vosoughi, and Deb Roy. Me, my echo chamber, and i: introspection on social media polarization. In *Proceedings of the 2018 World Wide Web Conference*, pages 823–831, 2018.

[36] Michel X Goemans and David P Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24 (2):296–317, 1995.

[37] Jacob Goldenberg, Barak Libai, and Eitan Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001.

[38] Jacob Goldenberg, Barak Libai, and Eitan Muller. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. *Academy of Marketing Science Review*, 9(3):1–18, 2001.

[39] Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):1–37, 2012.

[40] Mark Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420–1443, 1978.

[41] Daniel Gruhl, Ramanathan Guha, David Liben-Nowell, and Andrew Tomkins. Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web*, pages 491–501, 2004.

[42] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.

[43] Fritz Heider. Attitudes and cognitive organization. *The Journal of psychology*, 21(1):107–112, 1946.

[44] Cho-Jui Hsieh, Kai-Yang Chiang, and Inderjit S Dhillon. Low rank modeling of signed networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 507–515, 2012.

[45] Edmund Ihler, Gabriele Reich, and Peter Widmayer. Class steiner trees and vlsi-design. *Discrete Applied Mathematics*, 90(1-3):173–194, 1999.

[46] Mohammad Raihanul Islam, Sathappan Muthiah, Bijaya Adhikari, B Aditya Prakash, and Naren Ramakrishnan. Deepdiffuse: Predicting the'who'and'when'in cascades. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1055–1060. IEEE, 2018.

[47] Amin Javari and Mahdi Jalili. Cluster-based collaborative filtering for sign prediction in social networks with positive and negative links. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(2):1–19, 2014.

[48] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 643–650. IEEE, 2011.

[49] William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115 (772):700–721, 1927.

[50] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. In *International Conference on Machine Learning (ICML)*, pages 3448–3457, 2019.

[51] Srijan Kumar, William L Hamilton, Jure Leskovec, and Dan Jurafsky. Community interaction and conflict on the web. In *Proceedings of the 2018 World Wide Web Conference*, pages 933–943, 2018.

[52] Jérôme Kunegis, Andreas Lommatzsch, and Christian Bauckhage. The slashdot zoo: mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web*, pages 741–750, 2009.

[53] Theodoros Lappas, Evimaria Terzi, Dimitrios Gunopulos, and Heikki Mannila. Finding effectors in social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1059–1068, 2010.

[54] Jeff Larson, Surya Mattu, and Lauren Kirchner. How We Analyzed the COMPAS Recidivism Algorithm. `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`, 2016. [Online; accessed 24-April-2020].

[55] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1361–1370, 2010.

[56] Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. Deepcas: An end-to-end predictor of information cascades. In *Proceedings of the 26th international conference on World Wide Web*, pages 577–586, 2017.

[57] Paolo Massa and Paolo Avesani. Controversial users demand local trust metrics: An experimental study on epinions. com community. In *AAAI*, pages 121–126, 2005.

[58] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4): 1093–1113, 2014.

[59] Milena Mihail. Conductance and convergence of markov chains-a combinatorial treatment of expanders. In *FOCS*, volume 89, pages 526–531, 1989.

[60] George A Miller. Psychology as a means of promoting human welfare. *American psychologist*, 24(12):1063, 1969.

[61] Rubem P Mondaini. The steiner tree problem and its application to the modelling of biomolecular structures. In *Mathematical Modelling of Biosystems*, pages 199–219. Springer, 2008.

[62] Seth Myers and Jure Leskovec. On the convexity of latent social network inference. In *Advances in neural information processing systems*, pages 1741–1749, 2010.

[63] Le Ou-Yang, Dao-Qing Dai, and Xiao-Fei Zhang. Detecting protein complexes from signed protein-protein interaction networks. *IEEE/ACM transactions on computational biology and bioinformatics*, 12(6):1333–1344, 2015.

[64] B Aditya Prakash, Jilles Vreeken, and Christos Faloutsos. Spotting culprits in epidemics: How many and which ones? In *2012 IEEE 12th International Conference on Data Mining*, pages 11–20. IEEE, 2012.

[65] Walter Quattrociocchi, Antonio Scala, and Cass R Sunstein. Echo chambers on facebook. *Available at SSRN 2795110*, 2016.

[66] Dragomir R Radev, Sasha Blair-Goldensohn, Zhu Zhang, and Revathi Sundara Raghavan. Newsinessence: A system for domain-independent, real-time news clustering and multi-document summarization. In *Proceedings of the first international conference on Human language technology research*, pages 1–4. Association for Computational Linguistics, 2001.

[67] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.

[68] Pavan Rangudu, Bijaya Adhikari, B Aditya Prakash, and Anil Vullikanti. Using partial probes to infer network states. In *13th International Workshop on Mining and Learning with Graphs. MLG*, 2017.

[69] Kenneth E Read. Cultures of the central highlands, new guinea. *Southwestern Journal of Anthropology*, 10(1):1–43, 1954.

[70] James A Roberts. Profiling levels of socially responsible consumer behavior: a cluster analytic approach and its implications for marketing. *Journal of marketing Theory and practice*, 3(4):97–117, 1995.

[71] Manuel Gomez Rodriguez, Jure Leskovec, David Balduzzi, and Bernhard Schölkopf. Uncovering the structure and temporal dynamics of information propagation. *Network Science*, 2(1):26–65, 2014.

[72] Clemens Rösner and Melanie Schmidt. Privacy preserving clustering with constraints. In *45th International Colloquium on Automata, Languages, and Programming (ICALP)*, 2018.

[73] Polina Rozenshtein, Aristides Gionis, B Aditya Prakash, and Jilles Vreeken. Reconstructing an epidemic over time. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1835–1844, 2016.

[74] Marcel Salathé, Maria Kazandjieva, Jung Woo Lee, Philip Levis, Marcus W Feldman, and James H Jones. A high-resolution human contact network for infectious disease transmission. *Proceedings of the National Academy of Sciences*, 107(51):22020–22025, 2010.

[75] Thomas C Schelling. *Micromotives and macrobehavior*. WW Norton & Company, 2006.

[76] Emre Sefer and Carl Kingsford. Diffusion archeology for diffusion progression history reconstruction. *Knowledge and information systems*, 49(2): 403–427, 2016.

[77] Devavrat Shah and Tauhid Zaman. Rumors in a network: Who's the culprit? *IEEE Transactions on information theory*, 57(8):5163–5181, 2011.

[78] So Young Sohn and Sung Ho Lee. Data fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic accidents in korea. *Safety Science*, 41(1):1–14, 2003.

[79] Merriam-Webster Staff. *Merriam-Webster's collegiate dictionary*. Merriam-Webster, 2004.

[80] Will Steffen, Paul J Crutzen, and John R McNeill. The anthropocene: are humans now overwhelming the great forces of nature. *AMBIO: A Journal of the Human Environment*, 36(8):614–621, 2007.

[81] Will Steffen, Wendy Broadgate, Lisa Deutsch, Owen Gaffney, and Cornelia Ludwig. The trajectory of the anthropocene: the great acceleration. *The Anthropocene Review*, 2(1):81–98, 2015.

[82] Yanchao Sun, Cong Qian, Ning Yang, and S Yu Philip. Collaborative inference of coexisting information diffusions. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 1093–1098. IEEE, 2017.

[83] Shashidhar Sundareisan, Jilles Vreeken, and B Aditya Prakash. Hidden hazards: Finding missing nodes in large graph epidemics. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 415–423. SIAM, 2015.

[84] Joshua A Tucker, Andrew Guess, Pablo Barberá, Cristian Vaccari, Alexandra Siegel, Sergey Sanovich, Denis Stukal, and Brendan Nyhan. Social media, political polarization, and political disinformation: A review of the scientific literature. *Political polarization, and political disinformation: a review of the scientific literature (March 19, 2018)*, 2018.

[85] Jia Wang, Vincent W Zheng, Zemin Liu, and Kevin Chen-Chuan Chang. Topological recurrent neural network for diffusion prediction. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 475–484. IEEE, 2017.

[86] Lilian Weng, Filippo Menczer, and Yong-Yeol Ahn. Predicting successful memes using network and community structure. In *Eighth international AAAI conference on weblogs and social media*, 2014.

[87] Wikipedia contributors. Black death — Wikipedia, the free encyclopedia, 2020. URL `https://en.wikipedia.org/w/index.php?title=Black_Death&oldid=943613949`. [Online; accessed 3-March-2020].

[88] David Bruce Wilson. Generating random spanning trees more quickly than the cover time. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 296–303, 1996.

[89] Han Xiao, Bruno Ordozgoiti, and Aristides Gionis. Searching for polarization in signed graphs: a local spectral approach. In *The World Wide Web Conference*, 2020.

[90] Sirui Yao and Bert Huang. Beyond parity: Fairness objectives for collaborative filtering. In *Advances in Neural Information Processing Systems*, pages 2921–2930, 2017.

[91] John Zarocostas. How to fight an infodemic. *The Lancet*, 395(10225):676, 2020.

Aalto-DD 118/2020

ISBN 978-952-60-3989-3 (printed)
ISBN 978-952-60-3990-9 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Computer Science
www.aalto.fi

BUSINESS + ECONOMY

ART + DESIGN + ARCHITECTURE

SCIENCE + TECHNOLOGY

CROSSOVER

DOCTORAL DISSERTATIONS