# ABSTRACT

Title of dissertation: ADVERTISEMENT ALLOCATION AND TRUST MECHANISMS DESIGN IN SOCIAL NETWORKS

Peixin Gao

Dissertation directed by: Professor John S. Baras
Dept. of Electrical & Computer Engineering

Social network sites (SNS), such as Facebook, Google+ and Twitter, have attracted hundreds of millions of users daily since their appearance. Within SNS, users connect to each other, express their identity, disseminate information and form cooperation by interacting with their connected peers. The increasing popularity and ubiquity of SNS usage and the invaluable user behaviors and connections give birth to many applications and business models. We look into several important problems within the social network ecosystem. The first one is the SNS advertisement allocation problem. The other two are related to trust mechanisms design in social network setting, including local trust inference and global trust evaluation.

In SNS advertising, we study the problem of advertisement allocation from the ad platform's angle, and discuss its differences with the advertising model in the search engine setting. By leveraging the connection between social networks and hyperbolic geometry, we propose to solve the problem via approximation using hyperbolic embedding and convex optimization. A hyperbolic embedding method,

HyperCubeMap, is designed for the SNS ad allocation problem, and several components are introduced to realize the optimization formulation. We show the advantages of our new approach in solving the problem compared to the baseline integer programming (IP) formulation.

In studying the problem of trust mechanisms in social networks, we consider the existence of distrust (i.e. negative trust) relationships, and differentiate between the concept of local trust and global trust in social network setting. In the problem of local trust inference, we propose a 2-D trust model. Based on the model, we develop a semiring-based trust inference framework. In global trust evaluation, we consider a general setting with conflicting opinions, and propose a consensus-based approach to solve the complex problem in signed trust networks.

# ADVERTISEMENT ALLOCATION AND TRUST MECHANISMS DESIGN IN SOCIAL NETWORKS

by

Peixin Gao

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2016

Advisory Committee:
Professor John S. Baras, Chair/Advisor
Professor Jennifer A. Golbeck, Co-Advisor
Professor Gang Qu
Professor William Rand
Professor Charalampos Papamanthou

# Acknowledgments

This thesis to me is a milestone in my five years of work at University of Maryland and specifically within the SEIL Laboratory. My experience at UMD has been nothing short of amazing. I have been given unique opportunities and taken advantage of them. Throughout these years I have learned that there are so many areas under exploration that interest me and my research work has so much influence on solving many real-world problems. This thesis presents the research work that I have conducted during the five years of PhD study at UMD, namely advertisement allocation and trust mechanisms design in social network scenario. The research work discussed in this thesis will not be finished successfully without the help and support from lots of remarkable individuals, who I wish to acknowledge and express my sincere gratitude to.

First and foremost I wish to express my special appreciation and thanks to my advisor, Professor Dr. John Baras, who has been an extraordinary mentor with tremendous support and guidance during these five years of studying at University of Maryland as a PhD student. He has offered me the invaluable opportunity to work on challenging and extremely interesting projects over the past five years. I am also very much grateful for his encouragement on my research work and allowing me to grow as a researcher. His advice on both research as well as on my career have been very precious and beneficial. He has been very supportive since the days I began my studying at University of Maryland, not only in the aspect of research assistantship but also academically and emotionally through the rough road to finish this thesis.

Thanks to his guidance, I had the opportunity to explore various domains and learn knowledges from multiple disciplines. It has been a great pleasure to work with and learn from such an extraordinary individual.

I would also like to thank my co-advisor, Professor Jennifer Golbeck. She has always made herself available for help and advice with great patience. Without her extraordinary ideas and expertise on my research topic, this thesis would have been a dream of distance. Meanwhile, I am very thankful to Professor Gang Qu, Professor Charalampos Papamanthou and Professor William Rand for their guidance on my research and serving as my thesis committee members with invaluable advices.

My time at UMD has been enjoyable in large part due to the accompanying of many friends and groups that became a part of my life. It was my luck to meet many interesting genius who started their graduate study with me: Chau-Wai Wang, Dakang Ma, Sikai Qu, Yunlong Huang and many other people, with whom I have spent a lot of time and accumulated so many great memories. I would like to acknowledge Hui Miao who worked together with me on formulating the optimization problem in SNS advertising. My research group has always been a source of friendships as well as good advice and collaboration, there are so many names in the group that I cannot enumerate. I am also grateful for the help from Dr. Zhixin Liu, who I have had the pleasure to work with during her staying at our group as a visiting scholar. My colleagues at the SEIL lab deserve a special mention here, as they have not only helped me with graduate studies but also offered a great support for me. During the five years, I have received a lot of caring and assistance from Mrs. Edwards, the "manager" of our group, which makes my PhD life smooth

and efficient.

Lastly, I want to express special thanks to my family. Words cannot express how grateful I am to my parents for all of the sacrifices that they have made as well as the unconditional love and care they have provides for me. They are role models of my life, and have taught me about hard work and self-respect, about persistence and about how to be independent.who have always stood by me and guided me through my career, and have pulled me through against impossible odds at times. I would not have made it this far without their love and encouragement. I know I always have my family to count on when times are rough. Of course I owe a deep appreciation to my beloved beautiful wife, Yao Guo, who has offered me endless help, support, encouragement, and patience. She has unconditionally loved and cared me during my good and bad times. These past several years have not been an easy ride, both academically and personally. I truly thank Yao for staying by my side, even when I was irritable and depressed. I can not finish my PhD study so smoothly without her fully support.

This dissertation was partially supported by National Security Agency grant H98230-14-C-0137, and US Air Force Office of Scientific Research (AFOSR) MURI grants FA9550-09-1-0538, FA9550-10-1-0573.

Thank you all for making this thesis possible and my PhD study experience at University of Maryland invaluable and enjoyable! I will cherish these five years forever!

# Table of Contents

# List of Figures

# List of Tables

# CHAPTER 1: SOCIAL NETWORK MODELING

Social network sites (SNS) such as Facebook, Google+ and Twitter have attracted hundreds of millions of daily users since their appearance [1]. Within SNS, users connect to each other, express their identity, disseminate information and form discussions and cooperations by interacting with their connected peers. SNS browsing constitutes a large amount of time that people spent on the Web [2], dominating other kinds of online activities, meanwhile generating a huge amount of information on user behaviors and relationships. The increasing popularity of SNS has been generating data, demands, as well as new markets. Several social network-based applications have been built by leveraging the characteristics of social networks.

## 1.1 Characteristics of Social Networks

Social networks are playing an increasingly significant role in the information transmission, critical to the commerce and society of the current age. Due to such importance in economy and social life, a considerable amount of attention has been devoted to the computational analysis of social network structures.

A social network can be denoted as a graph $G(E, V)$, where $V$ is the set of vertices (nodes) representing the entities in the network, and $E$ is the set of edges

connecting vertices in $V$. An element $v_{ij}$ in $V$ represents a connection (possibly weighted and/or directed) between node $i$ and node $j$, where $i, j \in V$. Without loss of generality, the default setting in our work is that the network is undirected.

The large dimension of social networks is a key aspect. In fact, they represent an incredible source of information on a large-scale. These large-scale complex networks, however different in terms of application scenarios, share several common topological features on the graphs which represent these networks [3, 4].

There are several characteristics that commonly appears in most social networks, which are discussed and applied in modeling social networks.

### 1.1.1 The Small-World Effect

The "Small-World"effect on Social Networks was proposed by Milgram et al [5, 6]. Despite their big dimension, social networks usually show a common feature of small diameter: there exists a relatively short path connecting any pair of nodes within the network. Most nodes are not neighbors of each other, but could reach each other by a small number of hops.

The diameter $l$, that determines the "Small World"property, scales proportionally to the logarithm of the dimension of the network, which is formalized as

$$l \propto \log |V| \tag{1.1}$$

where $|V|$ means the cardinality of $V$. Many complex networks other than SNS also have the property of "Small World", such as Internet.

### 1.1.2 Scale-Free Degree Distributions

Another important feature of social networks is degree distribution of nodes. Recent empirical results shows that the degree distribution follows a power-law, i.e.

$$P(k) \propto k^{-\gamma} \tag{1.2}$$

where $k$ is the degree, and $\gamma$ is the power-law coefficient determined by the network topology which mostly ranges between 2 and 3.

Scale-free networks are a class of power-law networks where the high-degree nodes tend to be connected to other high-degree nodes, which have been discussed in detail by Li et al. [7]. Mislove et al. [3] showed that common online social networks display power-law distributions thus present scale-free properties as well.

### 1.1.3 Community Structure

Social networks contain community structure. The extent of such a "clustering"effect varies among different situations. The more this structural characteristic is evident, the higher a network tends to divide into groups of nodes whose connections are denser among entities belonging to the given group and sparser otherwise. The emergence of community structures in social networks has been supported by theoretical analysis [8] and verified by empirical experiment [3].

### 1.1.4 Social Influence

Everyone in a social network has their own influence over the network due to their connections with other nodes in the network. A user's behavior or perspective has explicit or implicit effect on her neighbors connecting to her in the network and there is a probability that her neighbors would imitate her behavior. Due to the mutual-influence nature, social network plays a fundamental role as a medium for the spread of information, ideas, and influence among its members. Two basic diffusion models, i.e. the *Linear Threshold Model* [9] and the *Independent Cascading Model* [10] are proposed to describe the influence diffusion process in a social network. How to leverage social influence of people in digital marketing led to the influence maximization problem [11–13], which is an optimization problem.

In order to build applications within social network, it is necessary to capture the influence effect embedded in the network.

## 1.2 Statistical Models

A series of models have been proposed in order to capture the features of social networks, including the *Poisson Random Graph Model*, the *Exponential Random Graph Model*, the *Small World Model*, and the *Preference Attachment Model*.

Poisson Random Graph model (Erdos-Renyi model [14]) and Exponential Random Graph models [15] have been widely adopted in social network analysis due to their simplicity. However, the series of models using random graph do not properly reflect the structure of real-world large scale networks because of un-power-law

degree distribution and lack of community structures in the network [4]. Small World model proposed by Watts and Strogatz [16] also fails in capture the power-law degree distribution within social networks even if the community structure is well represented [17]. The Preferential Attachment Model proposed by Barabasi and Albert [18], unlike the random graph models and small world model, offers the power-law like distribution of degrees, however, it does not support a meaningful community structure of the network [4].

## 1.3 Social Network and Hyperbolic Geometry

Hyperbolic space is a non-Euclidean geometrically smooth space that generalizes the idea of Riemannian manifolds with negative curvature without the parallel postulate [19]. Basic properties of the hyperbolic space include negative curvature, infinite number of parallel lines, thin triangles and the smoothness of the space [20, 21].

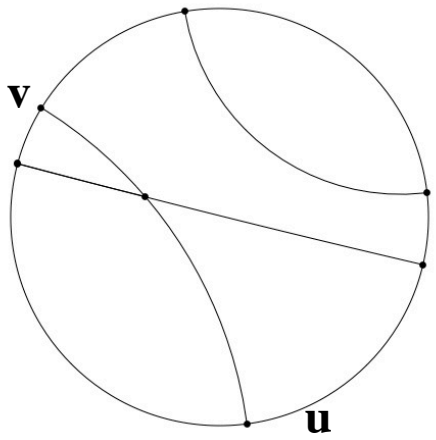



Figure 1.1: Poincaré disc model

There are several hyperbolic space models which are related by transformations. In our formulation, we use Poincaré disc model shown in Fig. 1.1. The disc $\mathbb{D} = \{z \in \mathbb{C} \mid |z| < 1\}$ is called the Poincaré disc. The circle $\partial\mathbb{D} = \{z \in \mathbb{C} \mid \|z\| = 1\}$ is called the circle at $\infty$ or boundary of $\mathbb{D}$. The advantage of using the Poincaré disc model is the hyperbolic distance $d(u, v)$ between two points $u, v \in \mathbb{D}$ can be calculated with the following equation easily:

$$d(u, v) = arcosh(1 + \frac{2\|u - v\|^2}{(1 - \|u\|^2)(1 - \|v\|^2)}) \tag{1.3}$$

Hyperbolic geometry rests in negative curvature, which has a generalization in the context of metric space. A metric space can be defined as follows:

**Definition.** *Metric Space*: A metric space $(X; d)$ is a set $X$ of points with a distance function $d : X2 \to \mathbb{R}_+$ satisfying the following conditions:

for $\forall u, v, w \in X$:

$$\begin{aligned} d(u; v) &= d(v; u) \\ d(u; v) &= 0 \Longleftrightarrow u = v \\ d(u; v) &\leq d(u; w) + d(w; v) \end{aligned} \tag{1.4}$$

A metric space $(X; d)$ is (Gromov) $\delta$-hyperbolic if for $\forall u, v, w, x \in X$:

$$d(u, v) + d(w, x) \geq d(u, x) + d(w, v) \geq d(u, w) + d(x, v) \tag{1.5}$$

and $\exists\ \delta$, such that:

$$D(u, v, w, x) = d(u, v) + d(w, x) - d(u, x) - d(w, v) \leq 2\delta \tag{1.6}$$

then we call $\delta_{X,d} = \sup\{D(u, v, w, x)/2, \forall u, v, w, x \in X\}$ the hyperbolicity of the metric space $(X; d)$.

Euclidean plane is not hyperbolic (i.e. the corresponding $\delta \to \infty$), while the hyperbolic plane (e.g Poincaré disc) is constantly hyperbolic.

### 1.3.1 Hyperbolicity of Complex Networks

As a geometric space that generalizes the idea of Riemannian manifolds with negative curvature, hyperbolic space has raised increasing attention due to its application in network modeling and analysis [22–25].

The connection between complex network and hyperbolic space can be shown via Gromov's $\delta$-hyperbolicity of a metric space.

In terms of a complex network, the undirected connected graph $G = (V, E)$ can be viewed as a metric space $(V, d_G)$ with standard graph distance metric $d_G$, e.g. shortest path length between $u, v \in V$ [26]. Similarly, $\delta(G) = \delta_{V,d_G}$ can be used to describe the hyperbolicity of the network. The $\delta$-hyperbolicity of a graph is shown as a measure of its "tree-likeness" [27, 28]. For example, $\delta(G)$ of a tree is 0, whereas a cycle of size $n$ is $\mathcal{O}(n)$-hyperbolic. Montgolfier et al. [27] shows that the $\delta(G)$ of the Internet is very small which reflects it is a tree-like structure, and that small hyperbolicity is a natural property of power law random graphs, which is a simple complex network model capturing many structural properties of complex networks.

### 1.3.2 Application of Network Hyperbolicity

Among the related work, there are two major branches in applying hyperbolic geometry to model complex networks [26]. One, proposed by R. Kleinberg [22], is

based on the concept of hyperbolicity of complex networks (e.g. social networks). The other is proposed by Krioukov et al. [25] based on the connection between characteristics of complex networks and hyperbolic geometry.

Kleinburg's model [22] is related to the tree-likeness of the networks. Tree structure is showed to have hyperbolicity [22,26,29] and can perfectly be embedding into a hyperbolic space. To embed a complex network (e.g. WSN, social network) in hyperbolic space of 2-D Poincaré disc, a spanning tree is generated from the network and used to map into the space. With such embedding, each node has a virtual coordinate in hyperbolic space and a greedy routing algorithm can be conducted in the network [22, 24].

Krioukov et al. [25, 30] showed that hyperbolic geometry underlies complex networks (e.g. social networks), and the heterogeneous degree distributions and strong clustering in complex networks emerge naturally as simple reflection of the negative curvature and metric property of hyperbolic geometry. Correspondingly, a network would have an effective hyperbolic geometry underneath if the network has some metric structure and degree distribution is heterogeneous within the network. By utilizing the relation between hyperbolic geometry and properties of complex networks, Krioukov et al. further designed a hyperbolic embedding scheme that maps a complex network into the hyperbolic space of 2-D Poincaré disc according to degrees of nodes in the network [23, 31]. The node density and expected degree distribution are both well-defined after mapping into the Poincaré disc.

These two different ideas are both useful in different applications. Currently hyperbolic embedding has been applied in navigability analysis [32], routing algo-

rithms design [24, 33] and link prediction [23, 30, 31] in complex networks.

## 1.4 Contributions of The Dissertation

With the increasing popularity and ubiquity of online social networks (SNS) usage and the invaluable data on user behaviors and connections, the problem of social network modeling and social-aware applications has received increasing attention from both industry and academia.

In this dissertation, we work on three major areas that are related to social network, namely SNS advertising, local trust metric design, and global trust evaluation. We build models and develop solutions to these problems discussed in this dissertation.

### 1.4.1 Social Network Advertising

In the problem of SNS advertising, we first study the social network advertising problem with pay-per-mille impression model and show its difference with the AdWords model [34] in Chapter 2. With the connection between social networks and hyperbolic geometry, we propose to solve the problem of social network advertisement allocation via hyperbolic embedding (HyperCubeMap) and convex optimization, with the aim of reducing dimensionality and solving the problem efficiently. We analyze the advantages of our formulation compared to the baseline approach. In Chapter 3, we further discuss several possible extensions based upon our current optimization framework, and evaluate the performance of our so-

lution via experiments, illustrating the optimality of our approach and the runtime improvement in multiple orders of magnitude over the baseline IP formulation.

### 1.4.2 Trust Evaluation in Social Networks

Regarding the problem of trust mechanisms design in social networks, we start from a investigation on research literature in the domain of trust metric evaluation and trust management in Chapter 4. We differentiate between the concept of local trust and global trust. Considering both trust and distrust (i.e. positive and negative trust) relationships within social networks, we then introduce an approach for local trust inference based on a semiring framework (distrust-semiring) in Chapter 5. In Chapter 6, we consider global trust evaluation in a general case where communities of conflicting opinions exist in the trust network. In the signed trust network, we formulate global trust evaluation as a bipartite consensus problem. We reach some results based on the property of structural balance in signed networks.

# CHAPTER 2: SNS AD ALLOCATION BASED ON HYPERCUBE-MAP

With the increasing popularity and ubiquity of online social networks (SNS), a large number of advertisers choose to post their advertisements within social networks and SNS advertising has grown rapidly in the past few years and is now a billion-dollar business.

In the SNS advertising model, advertisers participating in SNS advertisement (ad) campaigns benefit from the effects of viral marketing and network diffusion. Modern SNS serve as advertising agents, and take the advantage of network diffusion to attract advertisers and charge for the cascading impressions. The optimal ad allocation task is the problem of choosing the ad allocation plan that maximizes the revenue for the SNS.

Considering that users in SNS have diffusion abilities and limited daily impressions, and advertisers have various bidding prices and budget concerns, a feasible plan that obeys the constraints is difficult to find. The solution to this problem lies in the space of $\mathbb{N}_0^{|Ads|\times|User|}$, which makes direct optimization unattractive.

In this chapter, we study SNS advertising business models and formulate the SNS ad allocation problem. We show its connection with hyperbolic embedding,

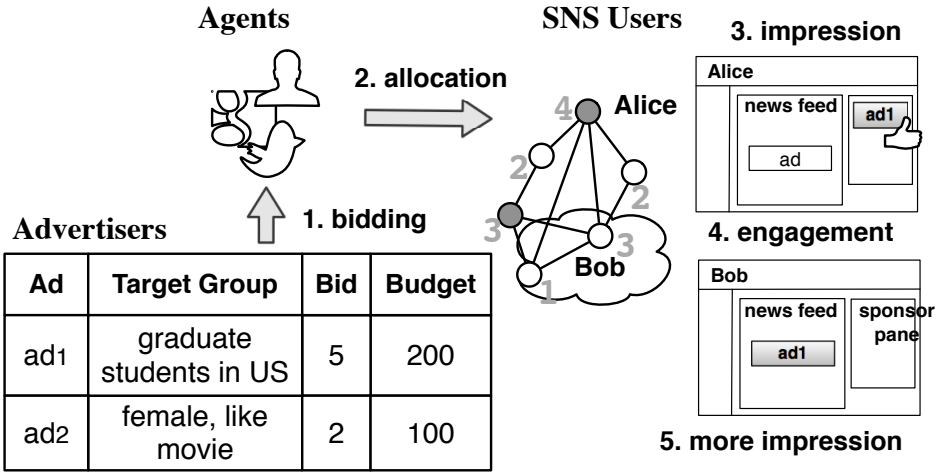
and discuss how the SNS ad allocation problem can be formulated as a linear program on region allocation in hyperbolic space based on hyperbolic embedding of the social network. We consider a general and complex setting with multiple target groups for different advertisers. Accordingly, we develop a new embedding algorithm HyperCubeMap along with the optimization framework that allow for dimension reduction. Our proposed method is able to reduce the dimensionality of the original problem significantly.

## 2.1 Advertising in SNS

Social network sites (SNS) such as Facebook, Google+ and Twitter have attracted hundreds of millions of daily users since their appearance. SNS browsing constitutes a large amount of the time that people spend on the Web, dominating other online activities [2]. In modern SNS, users expose many personal behaviors and connect to each other based on real world relationships, which makes SNS ideal for targeted advertising [35]. SNS advertising has grown rapidly in the past years. For example, Facebook has more than 1 million advertisers and 100 billion hits per day [36, 37], which contributes 90% of its revenue.

Ad agents (e.g. Facebook) allocate each ad to user impressions (i.e the behavior of user visiting the site). The advertising mechanism used by online advertising platforms, including social network websites, is essentially large-scale auctions where advertisers place bids on user impressions, and specify their daily or total budget [34].

As shown in Fig. 2.1, to perform a marketing campaign in an SNS such as Facebook or Twitter, advertisers first find an agent (which typically is the SNS site itself), choose target audience by specifying desirable user profiles (for instance, graduate students in US, or female, like movie), and provide their advertisements (ads) with bidding prices and budgets. Then the ad agent allocates the ads to the set of users whose profiles match the advertisers' targeting requests. For each impression (page view) of a user, the agent chooses one or several ads whose target audience include the user. Now the user can see and engage with the ad, for example 'like' in Facebook, '+1' in Google+, and 'retweet' in Twitter, and then her friends may see the ad and further engage. For example, in Fig. 2.1, Alice is a graduate student in US and is allocated for ad1. Alice and Bob are friends. The ad that Alice liked may be shown as a sponsored story in Bob's news feed in Facebook setting.




| Ad | Target Group | Bid | Budget |
|---|---|---|---|
| ad1 | graduate students in US | 5 | 200 |
| ad2 | female, like movie | 2 | 100 |

Figure 2.1: SNS ad campaign and allocation

For SNS advertising campaigns, instead of keywords, the advertisers bid for a target group of users' actions, which can be mille impressions (often referred as cost

per thousand impressions or CPM), engagements (e.g. click, retweet, comment), or actions (e.g. mobile application installation, product purchase). The agents run large auctions using the bids and charge advertisers by the user actions. There are associated billing policies, such as pay-per-mille, pay-per-click, pay-per-action, pay-per-engagement [38]. The pay-per-mille model is the default and most popular policy in Facebook, where the ad agent receives commission for one thousand user impressions displaying the ad. We will assume this policy throughout the paper.

The SNS ad allocation problem, to maximize the agent's revenue by allocating ads to user impressions while respecting the advertisers' requirements (targeting criteria, bidding method, and budget constraint etc.), is a central problem for advertising agents. An important component of the problem is the concept of *paid social influence*, which distinguishes it from standard ad allocation problem (AdWords [34]) and influence maximization problem [12] in complex networks.

Previous research mainly focused on search engine settings, where the impression is ad-hoc and associated with search queries. As a state of the art approach, AdWords [34, 39, 40] was proposed by Mehta et al. to solve the advertising allocation problem in the search engine setting of Google. The ad platform allocates impressions resulting from search queries to advertisers, with each advertiser having a budget constraint on the total spend. Each bidder puts in a set of bids for different keywords relevant to the ad. Once an advertiser's budget is exhausted, it cannot be allocated any more queries. The objective is to maximize the total efficiency of the matching, which is equivalent to maximizing the total amount of money (budget) spent by the advertisers. The offline algorithm is formulated as a bipartite matching

problem and solved via integer programming (IP) [39]. Due to incomplete information, and more importantly the size of the problem, the AdWords problem is solved as an online optimization problem in practice. For the online problem, Mehta et al. [34, 39, 40] achieved near 1-1/e optimality for the worst case.

In the SNS setting, each advertiser bids for a target group of users instead of keywords, each advertiser $A_i \in A = \{A_1, ..., A_{|A|}\}$ bids $p_i$ for all users in the target group. The agent assigns user impressions to $A_i$ before exhausting its budget $b_i$, with the allocation problem being to maximize the total amount of money (budgets) spent by advertisers. The impression of the user in this case is by no means of unit value, and is no longer ad-hoc. When a user is engaging with an ad (e.g. "like"in Facebook), her friends in the ego-network can see the ad and potentially engage with it as well.

Comparing with AdWords, although ad allocation in SNS have similar objectives, there are several key differences. First, in the AdWords problem, advertisers bid on a set of ad-hoc search query keywords, whereas in the SNS ad allocation problem, advertisers bid on active users. Moreover, as the substantial role of information diffusion in SNS [41], in SNS ad, the users allocated to a particular ad is allowed to engage with the ad and diffuse it to her neighbors, generating more impressions of the ad. The advertisers pay all the impressions. When using the AdWords approach directly without considering the paid social influence in the optimization, an advertiser's budget will be easily run out hence valuable user impressions will be wasted. Furthermore, the advertising agent (i.e. SNS provider) may need to define and consider the domain constraints of the ad allocation. For example, there may

be a constraint on fairness (i.e. users allocated to advertisers have similar influence distributions).

On the other hand, note that since the advertisers need to pay for all the impressions, the problem differs from the influence maximization problem in which one hopes to pay the best fixed size seed set of users to maximize the final number of influenced users she can reach by cascading. In our setting, each advertiser is interested in a user category satisfying certain search criteria, and she pays equally for all users' impressions including the ones via user engagement. In our work, we focus on the exact optimization of the SNS ad allocation problem, which is often conducted offline.

## 2.2 The SNS Ad Allocation Problem

How to allocate user impressions to a set of advertisers with bidding constraints while considering the social influence and possibly even the domain constraints is not well-studied. In this work we focus on offline optimization of advertisement allocation in SNS, which is approachable as the daily impression of users assigned to ads can be derived from their usage history. The offline optimization result is also meaningful for the online algorithm, since it suggests how to leverage social connections to improve revenue, and gives guidance in designing online optimization algorithms. It also serves as reference for advertisers on which groups of users they should choose, how big their target groups should be, and how much they should bid over the target groups.

To formulate this problem, let $A$ denote the set of advertisers, and $U$ be the set of users. Each user $u \in U$ has a daily impression $I_u$, 1-hop friends (neighbors) set $F_u$, and a social influence function $P(u)$ which we define in Eq. 2.2. Each advertiser $A_i \in A$ has a target user group $x_i \subseteq U$, a budget $b_i$ and bidding price $p_i$.

**Example 1.** In Fig. 2.1, the impression of *Alice*, $I_{u='Alice'}$, is 4, which shows how many times she views a refreshed Facebook page per day, while her social influence, $P('Alice')$, is when an ad is shown to her, how many other users will see that ad eventually via her engagement and diffusion. On the other hand, *ad1*'s target user group $x_{ad1}$ are all graduate students in *US*, $p_{ad1} = \$5$, and $b_{ad1} = \$200$.

In Facebook's case, there are seven major user attributes to define the target group $x_i$ of ad $A_i$ [42], including "location", "age", "gender", "language", "interests", and "more categories"(e.g. family status), forming roughly $10^6$ combinations. When an advertiser bids a selected target group, Facebook shows how many users satisfying that criteria in its network. If the number of users for a bid is too small, it is discouraged and a warning is generated.

Without loss of generality, we assume each advertiser has only one ad, and on a user's one impression, one ad is allowed to be displayed in the sponsor pane. Our proposed method can be extended by adding scalars in practice case. Notice that in a user's impression, her friends' ad engagement (e.g. liked, retweet, +1) is treated as common friends' updates which are displayed anyway with other updates in the news feed.

The solution of the allocation problem decides for each ad what is the initial

set of users to be displayed by considering their influence ability. Let the decision variable be $I \in \mathbb{N}_0^{|A|\times|U|}$. For each user $u$ and ad $A_i$, one dimension in the decision variable $I_{u,i} \in \mathbb{N}_0$ represents how many impressions of $u$ to be assigned to $A_i$ (i.e. the impression allocation strategy in the network). The optimization problem is to find the allocation that maximizes the ad agent's total revenue, which can be formulated as an integer program (IP):

$$\begin{aligned}
\max_{I} \quad & \sum_{A_i \in A} p_i \sum_{u \in x_i} I_{u,i}(1 + P(u)) && && \text{(revenue)} \\
\text{subject to} \quad & p_i \sum_{u \in x_i} I_{u,i}(1 + P(u)) \leq b_i && \forall A_i \in A && \text{(budget)} \\
& \sum_{A_i \in A} I_{u,i} \leq I_u && \forall u \in U && \text{(impression)} \\
& I_{u,i} \in \mathbb{N}^+, && \forall u \in U, A_i \in A && \text{(nonegative integer)}
\end{aligned} \tag{2.1}$$

where $A_i \in A$ is an ad with bid $p_i$ and $u$ is a node (i.e. user) in the social network with daily impression value $I_u$. The social influence function based on 1-hop neighborhood information about user $u$ is:

$$P(u) = \sum_{\nu \in F_u} w \min\{I_u, I_\nu\} \tag{2.2}$$

with $w$ the expected engagement probability (click-through rate). The $\min\{I_u, I_\nu\}$ in Eq. 2.2 means regardless the engagement probability, the user $u$'s engagement can be seen by her friend $\nu \in F_u$, $\min\{I_u, I_\nu\}$ times. If $I_u > I_\nu$, it is bounded by the daily impression her friend $\nu$ has. If $I_u \leq I_\nu$, the user $u$ at most engages $I_u$ times. As mentioned earlier, in practice, the engaged ad will be shown in her friends' news

feed, we assume her friend can always see the engagement if it happens when visiting her news feed. The reasons are two folds, a) recent Facebook study [43] shows that a user's normal post can be read by 35% of her friends, and 61% of them over a month, b) the news feed is ordered by proprietary ranking algorithm [44], which may treat ads and posts differently. We also assume 1-hop influence, as $w$ is often small (0.3%) in real SNS and network cascading is known to be shallow in general [45, 46]. With more conditions being considered, $P(u)$ can be adjusted. We discuss more in Sec. 3.2.

In the baseline formulation for SNS ad allocation described in Eq. 2.1, the decision variable $I \in \mathbb{N}_0^{|A|\times|U|}$ lies in high dimension as much as $10^{16}$ when considering 1 million advertisers, and billion users daily in Facebook. This makes the direct optimization impractical.

In order to make the offline optimization more tractable, we propose an approximation scheme. The key idea of our method exploits the concept of target group by using hyperbolic embedding for complex networks [23, 25]. Notice that in an advertiser $A_i$'s target group $x_i$, all users are considered the same by the advertiser. The only difference of choosing users in a target group is their different influence capability. If we can represent the user level impression allocation results and calculate the corresponding revenue with influence considered on target group level rather than on user level, we will be able to eliminate several orders of magnitude dimensions for the problem. Considering $10^9$ users and $10^{3\sim5}$ categories in a real world SNS, we can reduce the dimension around $10^{4\sim6}$, which makes the optimization much more attractive.

## 2.3 Our Approach and Techniques

We develop the optimization process by using hyperbolic embedding and a decomposition method that divide the problem into a series of smaller and simpler ones without introducing strong assumptions.

Hyperbolic embedding is a geometric mapping from a complex network $G(V, E)$ to a set of points and segments in a hyperbolic space $\mathbb{D} = \{z \in \mathbb{C} | \, |z| < 1\}$ (Poincaré disc model). The hyperbolic space is continuous and hyperbolic embedding is able to map arbitrary size complex networks into a bounded area where each node is assigned a coordinate. If we embed the social network and target group properly, we can use one or multiple regions in the hyperbolic space to express the allocation of a subset of users for an ad $A_i \in A$, in other words, we could approximate the revenue from $A_i$ as calculating integral of the user's influence function over a certain region $R_i \subset \mathbb{D}$:

$$\sum_{A_i \in A} p_i \sum_{u \in x_i} I_{u,i}(1 + P(u)) \cong \sum_{A_i \in A} p_i \int_r \int_\theta I(R_i(r, \theta)) d\theta dr \tag{2.3}$$

As we can use very few variables to describe the set of users assigned to $A_i$ with regular shapes such as fan and round, we can reduce dimensions of the original problem significantly (in the order of $O(U)$). On the other hand, region-wise advertisement allocation on hyperbolic embedding is a convenient framework for representing and visualizing domain constraints.

**Example 2.** In Fig. 2.2, we show the idea using data in Fig. 2.1. Two shapes on the base area represent two target groups: the left fan for *'graduate students in US'*,

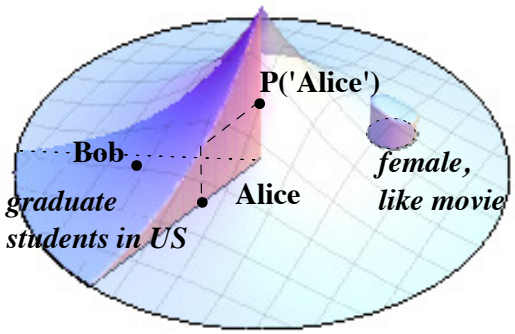



Figure 2.2: Hyperbolic embedding example

while the right round for *'female, like movie'*. Each user is assigned to a coordinate in her group, e.g. *Alice* is assigned to the left fan. The influence function defines the top surface. Note the impressions defines a different surface which is not shown.

To derive an embedding algorithm that fits the problem and allows integration is challenging, especially for the multiple target group setting where one needs to map different groups properly onto the hyperbolic space with smooth node density and influence function. Moreover, geometric shape design, region overlapping and impression distribution further make it even more difficult.

In order to make the idea possible, we develop an hyperbolic embedding algorithm, HyperCubeMap, for solving the SNS ad allocation problem under multiple user target group setting. Based on HyperCubeMap, we discuss approximating the SNS ad allocation problem as a hyperbolic space region allocation problem, and propose the corresponding optimization framework. The novel formulation is able to reduce the dimensionality of the problem to a large scale.

## 2.4 Preliminaries on Hyperbolic Embedding

Due to the hyperbolicity of complex networks (e.g. social networks), several research work has been done on network analysis in the framework of hyperbolic geometry [22, 24–26, 30]. Among related works, there are two major directions [26]: one proposed by Kleinberg [22] is for geometric routing by embedding the minimum spanning tree of the graph into the hyperbolic space, the other one is proposed by Krioukov et al. [23, 25], in which they use hyperbolic embedding to describe the topology and characteristics of complex networks. Our method is closely connected to the later line of work.

By assuming a hyperbolic geometry underlies complex networks (e.g. social networks), Krioukov et al. [25, 30] studied the connection between topology of hyperbolic geometry and the characteristics of complex networks. They showed that power law degree distributions and strong clustering in complex networks can be understood as reflection of the negative curvature property of the underlying hyperbolic geometry. They also showed that the hyperbolic geometry is able to accommodate complex networks of arbitrary size. By utilizing the connection, they further design a mapping scheme between hyperbolic space geometric framework and statistical mechanics of complex networks via the Poincaré disc mode, which successfully captures most important features in the complex network, i.e. small world effect, power law degree distribution (scale-free) and community structure. Hyperbolic embedding has been applied in in several areas, including navigability analysis [32], routing algorithms design [24, 33] and link prediction [30] in complex

networks.

## 2.5 HyperCubeMap: Hyperbolic Embedding for SNS Ad Allocation

The hyperbolic space is continuous and hyperbolic embedding on the Poincaré disc $\mathbb{D}$ is able to map arbitrary size social network into a fixed area by assigning each node to a coordinate $(r, \theta)$. Our approximation idea on the embedding (shown earlier in Sec. 2.3 and Fig. 2.2) essentially proposes to approximate the sum over the set of users to an integral function over a column assigning to an advertiser $A_i$.

In SNS ad allocation, the advertisers bid on heterogeneous user groups customized for their campaigns, and the users have different impressions and influence capability. Thus the hyperbolic embedding of a social network onto the Poincaré disc should have the following properties in order to apply it for dimension reduction of the problem:

- Both node density and degree distribution should be well-defined along angular and radial axises to support integrals in areas within the Poincaré circle.
- The social influence function defined on a point $(r, \theta)$ in $\mathbb{D}$ should be continuous on the Poincaré disc. As the cascading effect only happens when a user's friend engages with an allocated ad campaign, we can describe the influence as a function of users' degree accordingly.
- The embedding method should embed users within the same targeting group

onto connected regions, otherwise an allocation strategy for an advertiser targeting at particular user group have to be described by a collection of discrete points and the dimension reduction would not be achieved.

To the best of our knowledge, the existing embedding methods [22, 23, 25] do not obey all these prerequisites. In [23], Papadopoulos et al. propose an embedding scheme to map a complex network onto hyperbolic plane, where the node density $P_n(r) \propto e^r$ and the expected value of degree $E[d(r)] \propto e^{-\frac{r}{2}}$ along the radial coordinate are well-defined. The method satisfies the first prerequisite, however, their model does not help SNS ad allocation as the target groups, impression, and influence are not taken into considerations. The MLE step used in [23] to arrange angular coordinates is also computational expensive, where embedding a hundred thousand node network takes days to finish, which should be avoid when embedding real world SNS.

For SNS ad allocation in multiple target group scenario, we propose HyperCubeMap embedding algorithm. By extending [23], it first ensures the node density and degree are well-defined along radial axis:

$$\begin{aligned} \rho(r) &= ae^r && \text{(node desnity)} \\ \omega(r) &= ce^{-r/2} && \text{(degree distribution)} \end{aligned} \tag{2.4}$$

where $a$ and $c$ are constants derived from embedding.

Our embedding strategy further organizes the Poincaré disc into dimension reduction feasible grids (Sec. 2.5.1), calculates the minimal number of groups to boost dimensions reduction effectiveness (Sec. 2.5.2), and gives a hyperbolic embedding

method (Sec. 2.5.3) that satisfies all three prerequisites. To improve precision, we also propose uniform node density embedding which is easier to tune the performance (Sec. 2.5.4).

### 2.5.1 Isolated Cubes and Degree Spectrum

As mentioned in Sec. 2.1, an advertiser specifies the *target user group* for a particular campaign and sets bidding price and budget. The agent (i.e. Facebook) can only allocate and charge for the qualified users' impressions.

To enable this, the agent often provides a set of categorical filters, each of which has fixed number of options, for example, location, gender (M/F), age (0/20/30/40+), language and interests. The target user group of a campaign is defined by a selection of some or all of the given options, for instance, male and adult (20/30/40+) users in all states. The cardinality of option profiles are not very large,

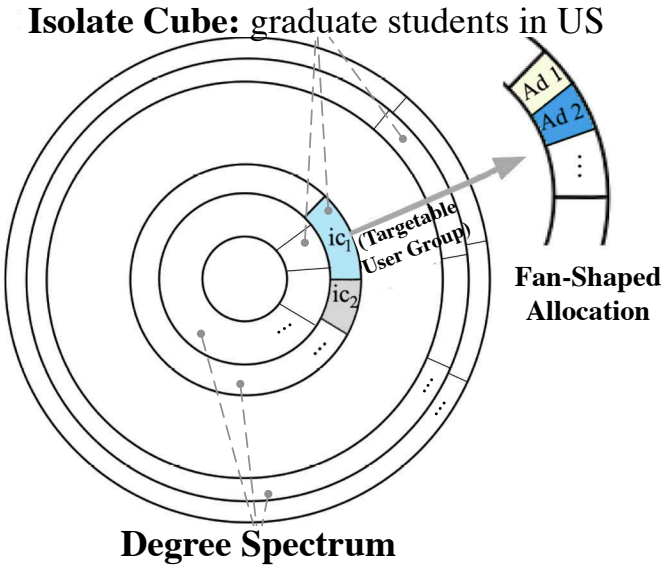



Figure 2.3: Isolated cubes and degree spectrum

e.g. Facebook has common option profiles upper bounded by $10^6$ and discourages advertisers from using too fine-grained filters by warnings during bidding [42]. Furthermore, users targeted by the different campaigns can be grouped together and result in many fewer groups for the matching process.

When embedding, the allocation rule should be considered, otherwise the same group of users will separate apart, so one needs to sum up over all discrete points representing the qualified users, which does not lead to dimension reduction. On the other hand, allocating the users together on the Poincaré disc may break the requirement on node density mentioned above or leave complicated shapes which are difficult to calculate in the optimization.

To capture these aspects of our problem, we propose the concept of *isolated cube* to express user similarities and groupings, and *degree spectrum* to divide Poincaré disc into finer and more regular shapes which eases the calculation and improves the precision, as shown in Fig. 2.3.

**Definition.** *Isolated Cube*: An isolated cube is a set of unit targetable user groups having the same set of campaigns.

In other words, users in the same isolated cube are shared by the same set of campaigns. Any two users in an isolated cube are interchangeable in an allocation solution to the advertiser. Note the opposite is not true: the campaigns sharing an isolated cube are not interchangeable in the allocation solution, as one campaign can target many isolated cubes. For instance, assuming we only have two advertisers, $\{a, b\}$, each of which starts a campaign, campaign $c_a$ targets on 'graduate

student living in US', $c_b$ targets on 'all male graduate student'. By definition 'male graduate student in US' is an isolated cube, $ic_0$, and within this cube, $a$ and $b$ share the impressions, which introduces necessary decision dimensions in the unknown. However, 'female graduate student lives in US' ($ic_1$), 'all male graduate student not living in US' ($ic_2$) are other isolated cubes targeted by each campaign exclusively, which implies one dimension is enough in the allocation.

As the isolated cubes are related to dimension reduction performance, the fewer isolated cubes, the better potential performance we can benefit from the embedding. The following lemma gives the worst number of isolated cubes:

**Lemma 2.5.1.** Considering that the ad agent defines $F$ categorical filters with each $f \in F$ has $v_f$ distinct options, there are at most $\prod_{f \in F} v_f$ isolated cubes.

As one can envision, the population in each user group may vary a lot, not to mention the degree distributions in each of them, which means different isolated cube can result in very different shapes on the Poincaré disc. To make the embedding useful and ensure the accuracy, we introduce the concept of *degree spectrum* to regularize the embedding shape.

**Definition.** *Degree Spectrum:* A degree spectrum $\Lambda$ is a series of annuli centered at $(0, 0)$ on 2-D Poincaré disc. Each annulus $\lambda \in \Lambda$ has a range $(r_s, r_e)$, representing all the users with degrees in the range of $(\omega(r_s), \omega(r_e)]$.

Recall in Eq. 2.4, $\omega(r)$ is the degree distribution function which maps radius $r$ to corresponding degree $d$.

As shown in Fig. 2.3, all the annuli constitute the degree spectrum. Within each annulus, isolated cubes are allocated in grids with areas proportional to their sizes. Each ad $A_i$ targets at a set of isolated cubes, $IC_i$, each of which has locations in some or all annuli in the spectrum $\Lambda$, thus the allocation can be represented by at most $|IC_i| \cdot |\Lambda|$ dimensions for $A_i$ comparing with $|\{u|u \in x_i\}|$ dimensions in the baseline formulation (recall $x_i \subseteq U$ is $A_i$'s target user group in Eq. 2.1).

Due to Lemma 2.5.1, $|IC_i|$ in practice is no more than $10^6$. Moreover, user groups of different criteria can be combined together when they are targeted by the same set of ads, making the dimensions be reduced even lower. On the other hand, the cardinality of the spectrum $|\Lambda|$ is an independent tuning parameter of our method, which can be tuned by fixing the degree range $d$. In the extreme case, $d = 1$, each annulus only contains the users with the same degree. $|\Lambda|$ is very small compared to the user set size.

### 2.5.2 Optimal Isolated Cubes

As the size of isolated cubes is important in the dimension reduction performance, we show how to get the minimal set of isolated cubes given all the advertisers' target user groups.

Assume the ad agent designs a set of filters $F$, where each $f \in F$ has a set of possible values, $v$. Each advertiser $A_i$ selects targeting values $(f, v_i)$ for each filter, denoted by $O_i = \{(f, v_i)|f \in F\}$, which defines a set of target users $T_i = \{u|(f, v_i) \in O_i, u[f] \in v_i\}$. Given all advertisers $A$ and their targeting profiles $O$, we can cluster

**Algorithm 1** Optimal Isolated Cube

Let $uc$ be an map of isolated cube and its bidding advertisers

**for** each advertiser $(A_i, (f, v_i)) \in O$ **do**

$uc(f, v_i) \cup = \{A_i\}$

**end for**

Let *opt_ic* be an map of hash key and clustered $uc$

**for** each $((f, v_i), A') \in uc$ **do**

$opt\_ic(\text{hash}(A')) \cup = \{(f, v_i)\}$

**end for**

**return** all targetable user group sets in *opt_ic*

targeted users together and derive the *optimal isolated cubes*, which gives the best dimension reduction performance in the hyperbolic embedding.

**Definition.** *Optimal Isolated Cubes:* The optimal isolated cubes is the smallest set of isolated cubes, s.t. all targeted users by the same set of advertisers are clustered together.

In Alg. 1, we give the clustering method to calculate optimal isolated cubes in $\mathcal{O}(O)$ time. It first goes through all possible unit targetable user groups in $O$ bid by the advertisers $A$, and groups advertisers by targetable user group. Then, it clusters targetable user groups together into individual sets if they share the same set of advertisers.

**Lemma 2.5.2.** Given filter $F$, advertisers $A$ and their targeting profiles $O$, Alg. 1 gives the optimal isolated cubes.

*Proof.* First, Alg. 1 outputs valid isolated cubes by definition, all targetable user groups with the same hash key share the same set of advertisers. Second, since any subset of different hash key targetable user groups cannot be merged to be a valid isolated cube, its output is the smallest set of isolated cubes. □

### 2.5.3 The HyperCubeMap Algorithm

After generating the optimal isolated cubes, we embed them onto the Poincaré disc in a proper way to satisfy the three prerequisites mentioned in the beginning of Sec 2.5. Here we introduce HyperCubeMap, a hyperbolic embedding algorithm for SNS ad allocation, which is given in Alg. 2. Given a social network $G(U, E)$, advertisers $A$, targeting profile $O$ and a spectrum design $\Lambda$, HyperCubeMap places each user $u \in U$ to $(r_u, \theta_u)$.

In Alg. 2, it first generates the optimal isolated cubes *opt_ic*, and then for each spectrum annulus $\lambda(r_s, r_e)$, it assign each $ic \in$ *opt_ic* a range of angular coordinate $(\theta_s, \theta_e)$. To ensure the uniform node density along angular axis, the range assignment is proportional to the $ic$'s target user size portion in this spectrum annulus. Then, Alg. 2 begins to assign users virtual coordinates on the hyperbolic plane. In order to let the node density and degree distribution be well-defined and in accordance with the requirements, we modify the method proposed by Papadoupolous et al. [23] on the assignment of angular coordinate. To ensure the same targetable user groups are allocated together, we assign the angular of each user according to its associated isolated cube $ic$. It is worth noting that in Alg. 2, $\zeta$ is the parameter related to

the curvature $K$ of the hyperbolic plane, with $\zeta = \sqrt{-K}$. We set $\zeta$ to be 1 in our experiment. $\beta$ is a mitigating factor determined by the power law exponent $\gamma$ with $\beta = \frac{1}{\gamma}$; $\gamma$ can be found for a given social network. The complexity of this algorithm is linear given a user set sorted by degree.

The embedding algorithm HyperCubeMap produces an embedding that satisfies our prerequisites. On the Poincaré disc, the degree distribution and node density are well-defined along angular and radial axises, with node density $\rho(r) = ae^r$, and correspondingly expected degree $\omega(r) = ce^{-r/2}$, $r \in [0, R)$. On the other hand, the same targeting group users are embedded into connected regions. By using a continuous social influence function definition, we can use the output of this algorithm to reformulate the SNS ad allocation problem as an region allocation problem with much reduced dimensions for the decision variable.

### 2.5.4 Uniform Node Density Embedding

From Alg. 2 we can notice that the inner area of the Poincaré disc is very sparse due to the exponential node density along the radius, which effects the optimality of approximation and makes it difficult for parameter tuning and allocation scheme design. Thus, corresponding to our application scenario, we propose to adjust the node density and make it uniform along the radius by moving nodes inside. The degree distribution is also adjusted accordingly.

Suppose for a previous point at $(r, \theta)$ in the Poincaré disc by HyperCubeMap, its new coordinate is $(r', \theta)$, then the uniform density function $\rho'(r')$ can be

**Algorithm 2** HyperCubeMap

Let *opt_ic* be the Optimal Isolated Cube output by Alg. 1

Let each annulus $\lambda(r_s, r_e) \in \Lambda$ and its user size be $N_a$

$\theta_s = 0$

**for** each $\lambda(r_s, r_e) \in \Lambda$ **do**

  **for** each $ic \in opt\_ic$ **do**

    Let isolated cube $ic$'s user size be $ic.n_a$

    $\theta_e = \theta_s + 2\pi \cdot ic.n_a / N_a$

    Let $ic$'s angular range $ic_{ang}[\lambda] = (\theta_s, \theta_e)$

    $\theta_s = \theta_e$

  **end for**

**end for**

Sort $U$ by degree in descending order $d_1 > d_2 > \cdots > d_n$ and break ties arbitrarily.

Let $u$'s degree be $d_u$

Let $r_1 = 0$, and $\theta_1$ is chosen randomly in $[0, 2\pi]$

**for** $u$ from 1 to $n - 1$ **do**

  Let $r_u = \beta \frac{2}{\zeta} \log u + (1 - \beta) \frac{2}{\zeta} \log n$

  Find spectrum $\lambda'(r_s, r_e)$, satisfying $r_u \in \lambda'(r_s, r_e)$

  Find isolated cube $ic$ satisfying $u[f] \in v_{ic}, \forall (f, v_{ic}) \in ic$

  Let $u$'s angular coordinate $\theta_u$ be chosen randomly from $ic_{ang}[\lambda']$

**end for**

derived as follows:

$$\rho'(r') = \frac{\int_0^R \int_0^{2\pi} \rho(\tau) d\theta d\tau}{\pi R^2} = \frac{2a(e^R - 1)}{R^2} \qquad (constant) \tag{2.5}$$

and the cumulative distribution function:

$$\text{CDF}(r') = \frac{\pi r'^2}{\pi R^2} = \frac{\int_0^r \int_0^{2\pi} a e^{\tau} d\theta d\tau}{\int_0^R \int_0^{2\pi} a e^{\tau} d\theta d\tau} = \frac{e^r - 1}{e^R - 1} \tag{2.6}$$

thus the mapping between $r$ and $r'$ is:

$$r' = \psi(r) = \sqrt{R^2 \cdot \frac{(e^r - 1)}{(e^R - 1)}} = R\sqrt{\frac{e^r - 1}{e^R - 1}} \tag{2.7}$$

and

$$r = \psi^{-1}(r') = \ln(r'^2 e^R - r'^2 + R^2) - 2\ln(R) \tag{2.8}$$

Then the expected degree at new coordinate $(r', \theta)$ is:

$$\omega'(r') = \omega(r)|_{r=\psi(r')} = \frac{cR}{\sqrt{r'^2 e^R - r'^2 + R^2}} \tag{2.9}$$

With the uniform node density, the expected degree along the radius is still well-defined. We can use the new node density and corresponding degree distribution to formulate the region allocation problem mentioned above similarly.

## 2.6 SNS Ad Allocation as Region Allocation in Hyperbolic Space

Using the hyperbolic embedding of a social network by HyperCubeMap, we can reformulate the ad allocation problem as a region allocation problem on the 2-D Poincaré disc. Each ad $A_i \in A$, bids on a set of isolated cubes denoted as $T_i = \{ic_1, ic_2, ..., ic_n\}$, where $|T_i|$ is the number of $A_i$'s isolated cubes. $T = \cup T_i$ are

the optimal isolated cubes (*opt_ic*) generated by Alg. 1. Given the degree spectrum $\Lambda$, the allocation profile for $A_i$ is defined as $S_i = \{S_i^{\lambda,c} | \lambda \in \Lambda, c \in T\}$, where $\lambda$ denotes the annulus in the degree spectrum, $c$ specifies the isolated cube. In other words, each $S_i^{\lambda,c}$ in the allocation profile $S_i$ for $A_i$ specifies a set of users embedded in the particular cube $(\lambda, c)$. By specifying the allocation region shape as fan shape as shown in Fig. 2.3, $S_i^{\lambda,c}$ can be determined by the start angle $\theta_{i,s}^{\lambda,c}$ and the end angle $\theta_{i,e}^{\lambda,c}$, with $S_i \in \mathbb{R}^{2|\Lambda| \times |T|}$. We can then cast the optimal SNS ad allocation problem as follows:

**Problem.** *Optimal Region Allocation:* On the 2-D Poincaré disc with interior representing the users embedded by HyperCubeMap algorithm, each user $u \in U$ is placed at $(r_u, \theta_u)$ in polar coordinates, with expected degree $d_u = \omega(r_u) = ce^{-r_u/2}$ and impression $I_u \in I$. Given a set of ads $A$, where $A_i \in A$ has a budget $b_i$ and bidding price $p_i$ on target users (its isolated cubes $T_i$).

The ad agent designs an allocation profile $S_i$ for each advertiser $A_i$. $S_i$ is a set of fans $\{(\theta_{i,s}^{\lambda,c}, \theta_{i,e}^{\lambda,c})\}$, each of which describes how to allocate users in an isolated cube $c$ in a degree spectrum annulus $\lambda$. The optimal region allocation is to derive an allocation profile $S$ for $A$ to maximize the revenue of the agent and, at the same

time, respect the budget and impression constraints:

$$
\begin{aligned}
&\max_{S} \sum_{A_i \in A} p_i f_i(S, I) \\
&\text{subject to } S_i \subset T_i && \forall A_i \in A \\
&\quad f_i(S, I) \geq 0 && \forall A_i \in A \\
&\quad p_i f_i(S, I) \leq b_i && \forall A_i \in A \\
&\quad \phi_u(S, I) = \sum_{S_i \in S} \phi_u(S_i, I) \leq I_u && \forall u \in U \\
&\quad \theta_{i,s}^{\lambda,c} \geq \theta_s^{\lambda,c} && \forall c \in T, \lambda \in \Lambda \\
&\quad \theta_{i,e}^{\lambda,c} \leq \theta_e^{\lambda,c} && \forall c \in T, \lambda \in \Lambda
\end{aligned}
\tag{2.10}
$$

where $f_i(S, I)$ is $A_i$'s actual sum of impressions considering social influence. $\phi_u(S_i, I)$ is the amount user $u$'s impressions assigned to $A_i$. $T$ is the set of optimal isolated cubes, and $T_i$ is the target set of $A_i$. $\Lambda$ is the degree spectrum.

As we can see, comparing with the original optimization problem, now a set of users is assembled as a fan-shaped region on the Poincaré circle, which reduces the dimensions significantly. On the other hand, angular coordinates are continuous value instead of discrete value as before. If we can give closed forms for each advertiser $A_i$'s assigned impression $f_i(S, I)$ and each user $u$'s allocated impression $\phi_u(S, I)$, then we can solve the problem directly. In order to do so, we need to specify how to incorporate with social influence, and address two major challenges: a) the impression distribution may not be well-defined and may be uncorrelated with degree and density, b) the overlapping fans. The first issue prevents us to apply integral, while the second issue makes the optimization problem much more

complicated.

### 2.6.1 Incorporating Social Influence

As is defined above, $f_i(S, I)$ is the sum of actual impressions assigned to advertiser $A_i$. The actual impressions resulted from user $u$ is different from $I_u$ due to her social influence in the network. All exposed qualified impressions will have a cost, thus actual profit that the ad agent can get from allocating the advertisement $A_i$ at a user $u$ is:

$$p_i \cdot I_u \cdot (1 + P(u)) \tag{2.11}$$

where $P(u)$ is the function describing the influence of user $u$ due to her engagement of the campaign (Sec. 2.2). Assuming $P(u)$ is proportional to her 1-hop degree, then the Eq. 2.11 can be rewritten as:

$$p_i \cdot I_u \cdot (1 + w \cdot d_u) \tag{2.12}$$

where $d_u$ is the degree of node $u$, and $w$ is a constant presenting the engagement rate. After embedding the SNS using HyperCubeMap, its expected degree at $(r_u, \theta_u)$ is:

$$d_u = \omega(r_u) = ce^{-r_u/2} \tag{2.13}$$

Thus, the influence function of user $u$ is:

$$P(u) = P(r_u, \theta_u) = w \cdot ce^{-r_u/2} \tag{2.14}$$

under uniform node density, the influence function can be denoted as:

$$P'(u) = P(r'_u, \theta'_u) = w \cdot \frac{cR}{\sqrt{r'^2 e^R - r'^2 + R^2}} \tag{2.15}$$

which are both continuous functions, and can be used in integral to express $f_i(S, I)$ over the Poincaré disc.

In Eq. 2.12, we essentially assume that the cascading is up to 1-hop of $u$. As mention in Sec. 2.2, diffusion is shallow [45, 46] and driven by simple contagion via social influence [41], and the engagement rate $w$ is small in practice (0.3%) [47], so the effect of multi-hop cascading is negligible and we argue it is a reasonable assumption. In general, as long as the approximate influence function is continuous about $(r_u, \theta_u)$, our method can be used. We discuss it in detail in Sec. 3.2.

### 2.6.2 Challenges

*Uncorrelated Impressions*: Function $f_i(S, I)$ not only depends on $S$ but also depends on user impressions $I$. It is important to note that in a social network, users have different impressions and degrees. After hyperbolic embedding, although the degree distribution is well-defined, the distribution of user impressions is not known, and may not be correlated with the influence. For example, President Obama who has about 40 million followers on Facebook may visit SNS seldomly, while a teenager having moderate number of friends uses SNS heavily everyday. As $I$ has unknown distribution in the mapped hyperbolic space, without introducing strong assumptions, it requires optimizing over the combinations of users' impressions in the corresponding isolated cube in $T$, which significantly increases the dimensions.

*Overlapping Fans*: The allocation areas of different ads may overlap, as a user could be assigned to different ads. Due to multiple impressions that a user may

have, fans assigned to different advertisers inevitably have intersection regions in the Poincaré disc, which makes the $f_i(S, I)$ calculation and the overall optimization problem more difficult.

To address these issues, we propose a novel unit-impression based decomposition method which preserves the advantages of the hyperbolic space mapping, and at the same time derives an optimal solution without introducing strong assumptions or high complexity constraints.

### 2.6.3 Unit Impression Decomposition

The unknown user impression distribution after the hyperbolic embedding of the SNS significantly affects our formulation. Complex region intersection may not have an analytical expression or convexity. Also the unknown impression distribution forces us to discretize $f_i$ and inevitably increase the complexity. We'd like to avoid this in order to maintain the attractiveness of our formulation.

We introduce a novel decomposition called *Unit Impression Decomposition* to avoid these two issues without introducing strong assumptions (e.g. disallow overlapping, enforce well-defined impression distributions). We first introduce the unit impression graph, and then use it to develop our optimization algorithm.

**Definition.** *Unit Impression Graph*: Given a social network graph $G(U, E)$, where $U$ represents users, $E$ shows relationships between users. Each $u \in U$ has an impression value $I_u$. $G$ is called a Unit Impression Graph if $I_u = 1,\ \forall u \in U$.

Given an SNS, we can induce a set of unit impression graphs. For example

if a user visits SNS 3 times a day (i.e. 3 impressions, or 3 chances to engage with a campaign), she can appear in 3 different graphs of unit impression, which means her impressions can be potentially assigned to 3 ads. The number of impressions in each unit impression graph now is 1, and there cannot be any intersections (i.e. one impression cannot be shared by advertisers). A sub step optimization problem can be conducted with a unit impression graph by adding a non-overlap constraint, and more importantly $f_i(S, I)$ can be formulated as $f_i(S_i)$, as the volume (impressions) assigned to $A_i$ is independent from others.

**Example 3.** Following up the example shown in Fig. 2.1, each graph in Fig. 2.4 is a *Unit Impression Graph* decomposed step by step from the original graph. The number in red beside the vertex shows the residual impressions of a user. As $G^{(1)}$ is a *Unit Impression Graph*, all its node have impression of 1. Then an optimization is performed in $G^{(1)}$. Assuming all users are allocated, $G^{(2)}$ shows the next unit impression graph. The residual impressions are updated by subtracting 1 from each node. Notice that $v_5$ has zero impression, it is not included in the graph $G^{(2)}$. In other words, if a user does not have impressions anymore, her friend's engaged campaign will not influence her any more on that day. In $G^{(3)}$, $v_3$ and $v_6$ are removed. The decomposition and optimization process ends at $G^{(4)}$, as no one in the network has impressions any more.

Corresponding to Fig. 2.4, the general algorithm for unit impression decomposition is shown in Alg. 3.

With the *Unit Impression Decomposition*, we can solve the original problem

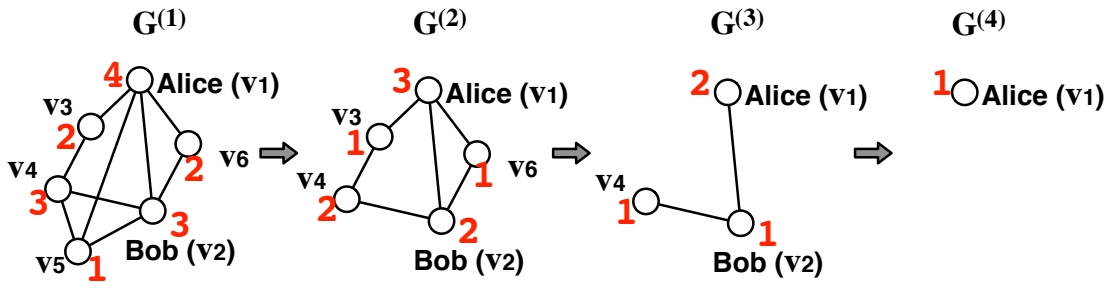



Figure 2.4: Unit impression graph transformation

using a multi-stage optimization process. It finishes when all impressions are allocated or all budgets are used. In the $m$th stage, given the unit impression graph $G^{(m)}$, we apply HyperCubeMap to embed $G^{(m)}$ in the hyperbolic space. The optimization problem regarding a unit impression graph $G^{(m)}$ only considers ads with nonzero budgets $A^{(m)} = \{A_i \mid b_i^{(m)} > 0\}$,and is given in Eq. 2.16.

$$
\begin{aligned}
\max_{S^{(m)}} \quad & \sum_{A_i \in A^{(m)}} p_i f_i(S_i^{(m)}) \\
\text{subject to} \quad & S_i^{(m)} \subset T_i^{(m)} && \forall A_i \in A^{(m)} \\
& f_i(S_i^{(m)}) \geq 0 && \forall A_i \in A^{(m)} \\
& p_i f_i(S_i^{(m)}) \leq b_i^{(m)} && \forall A_i \in A^{(m)} \\
& S_i^{(m)} \cap S_j^{(m)} = \emptyset && \forall A_i, A_j \in A^{(m)} \wedge i \neq j \\
& \bigcup_{A_i}^{A^{(m)}} S_i^{\lambda, c(m)} \leq S^{\lambda, c(m)} && \forall c \in T^{(m)}, \lambda \in \Lambda^{(m)}
\end{aligned}
\tag{2.16}
$$

We then solve the non-overlapping problem stated in Eq. 2.16, and record its optimal solution $S^{(m)*}$ and optimal value $\sum_{A_i \in A^{(m)}} f_i(S_i^{(m)*})$. If all ads' budgets are reached, the whole optimization ends. Otherwise the budget vector is updated as

**Algorithm 3** Unit Impression Decomposition

Put users with impressions into the first graph $G^{(1)}$.

Let the maximum impression in the network be $I_{max}$.

**for** $k$ from 2 to $I_{max}$ **do**

Decrement impressions of all users in $G^{(k-1)}$ by one.

**for** each $u \in U^{(k-1)}$ **do**

**if** the impression of $u$, $I_u > 0$ **then**

Add $u$ into $G^{(k)}$

**end if**

**end for**

**end for**

**return** $\{G^{(k)} | k \in \{1, ..., I_{max}\}\}$

$b_i^{(m+1)} = b_i^{(m)} - p_i \cdot f_i(S_i^{(m)*})$. Then the $(m+1)$th graph is generated with residual impressions, with zero impression users removed from the graph. The whole process ends when all advertisers' budgets are used, or all the impressions are exploited.

The unit impression decomposition process largely simplifies the optimization problem in each stage. The original problem of solving multi-location ad allocation with overlapping can be transformed to a multi-stage ad allocation problem with no overlapping.

### 2.6.4 Optimal Ad Allocation Strategy

As mentioned above, the area $S_i^{\lambda,c}$ that is assigned to ad $A_i$ in an isolated cube $c$ in a degree spectrum annulus $\lambda$ can be described by $(\theta_{i,s}^{\lambda,c}, \theta_{i,e}^{\lambda,c})$. With well-defined node density and degree distribution, the allocation $f_i(S_i^{\lambda,c})$ can be calculated as:

$$\begin{aligned} f_i(S_i^{\lambda,c}) &= f_i(\theta_{i,s}^{\lambda,c}, \theta_{i,e}^{\lambda,c}) \\ &= \int_{r_s^\lambda}^{r_e^\lambda} \int_{\theta_{i,s}^{\lambda,c}}^{\theta_{i,e}^{\lambda,c}} \rho(\tau)(1 + P(\tau, \theta)) d\theta d\tau \\ &= a \int_{r_s^\lambda}^{r_e^\lambda} e^\tau (1 + wce^{-\frac{\tau}{2}}) \int_{\theta_{i,s}^{\lambda,c}}^{\theta_{i,e}^{\lambda,c}} d\theta d\tau \\ &= a(\theta_{i,e}^{\lambda,c} - \theta_{i,s}^{\lambda,c})(2wce^{\frac{r_e^\lambda}{2}} - 2wce^{\frac{r_s^\lambda}{2}} + e^{r_e^\lambda} - e^{r_s^\lambda}) \\ &= \Delta_\lambda \theta_i^{\lambda,c} \end{aligned} \tag{2.17}$$

where $\Delta_\lambda = a(2wce^{\frac{r_e^\lambda}{2}} - 2wce^{\frac{r_s^\lambda}{2}} + e^{r_e^\lambda} - e^{r_s^\lambda})$ is a constant related to the annulus $\lambda$ on the degree spectrum $\Lambda$, and $\theta_i^{\lambda,c} = \theta_{i,e}^{\lambda,c} - \theta_{i,s}^{\lambda,c}$ is the angle range of the region $S_i^{\lambda,c}$. From Eq. 2.17, we notice that

If we apply the uniform node density transform, then $f_i$ can be calculated with

a different boundary $(r'^{\lambda}_s, r'^{\lambda}_e)$, and the expression for $f_i(S_i^{\lambda,c})$ is:

$$
\begin{aligned}
f_i(S_i^{\lambda,c}) &= f_i(\theta_{i,s}^{\lambda,c}, \theta_{i,e}^{\lambda,c}) \\
&= \int_{r'^{\lambda}_s}^{r'^{\lambda}_e} \int_{\theta_{i,s}^{\lambda,c}}^{\theta_{i,e}^{\lambda,c}} \rho'(\tau')(1+P'(u))d\theta d\tau' \\
&\quad \text{(variable substitution with } \tau = \psi(\tau') \text{ using Eq. 2.8)} \\
&= a \int_{\psi^{-1}(r'^{\lambda}_s)}^{\psi^{-1}(r'^{\lambda}_e)} e^{\tau}(1+wce^{-\frac{\tau}{2}}) \int_{\theta_{i,s}^{\lambda,c}}^{\theta_{i,e}^{\lambda,c}} d\theta d\tau \\
&= a(\theta_{i,e}^{\lambda,c} - \theta_{i,s}^{\lambda,c})(2wce^{\frac{\psi^{-1}(r'^{\lambda}_e)}{2}} - 2wce^{\frac{\psi^{-1}(r'^{\lambda}_s)}{2}} + \\
& e^{\psi^{-1}(r'^{\lambda}_e)} - e^{\psi^{-1}(r'^{\lambda}_s)}) \\
&= \Delta'_{\lambda}\theta_i^{\lambda,c}
\end{aligned}
\tag{2.18}
$$

where $\Delta'_{\lambda} = a(2wce^{\frac{\psi^{-1}(r'^{\lambda}_e)}{2}} - 2wce^{\frac{\psi^{-1}(r'^{\lambda}_s)}{2}} + e^{\psi^{-1}(r'^{\lambda}_e)} - e^{\psi^{-1}(r'^{\lambda}_s)})$, which is also a constant related to the annulus $\lambda$ on the degree spectrum $\Lambda$.

Combining the newly introduced impression decomposition and fan-shaped allocation strategy with HyperCubeMap, we can elaborate the optimal region allocation problem stated in Eq. 2.16 as follows:

$$
\begin{aligned}
\max_{\Theta^{(m)}} \quad & \sum_{A_i \in A^{(m)}} p_i \sum_{\lambda \in \Lambda^{(m)}} \Delta_{\lambda} \sum_{c \in T_i^{(m)}} \theta_i^{\lambda,c(m)} \\
\text{subject to} \quad & \theta_i^{\lambda,c(m)} \geq 0 \\
& p_i \sum_{\lambda \in \Lambda^{(m)}} \Delta_{\lambda} \sum_{c \in T} \theta_i^{\lambda,c(m)} \leq b_i^{(m)} \\
& \sum_{A_i \in A^{(m)}} \theta_i^{\lambda,c(m)} \leq \theta_e^{\lambda,c(m)} - \theta_s^{\lambda,c(m)} \\
& \qquad \forall A_i \in A^{(m)}, c \in T_i^{(m)}, \lambda \in \Lambda^{(m)}
\end{aligned}
\tag{2.19}
$$

where the decision variable $\Theta \in \mathbb{R}^{|A|\times|\Lambda|\times|T|}$, $p_i$ is the bidding price and $b_i^{(m)}$ is the budget of $A_i$ at stage $m$. $\Delta_{\lambda}$ is the constant related to annulus $a$ in Eq. 2.17. The

expression for uniform node density setting can be derived accordingly by replacing $\Delta_\lambda$ to $\Delta'_\lambda$ (shown in Eq. 2.18).

With the unit impression decomposition under fan-based allocation strategy, the optimization problem can be solved by a series of linear programs like Eq. 2.19. If the optimization is stopped after $n$ stages, then the allocation of advertiser $A_i$ is the aggregation of optimal solutions: $\cup_{m=1}^{n} S_i^{(m)*}$. It's worth pointing out while in one iteration there are no overlaps, the finally aggregated regions combining results from all $n$ stages do have overlaps among ads, as each iteration is conducted on a different unit impression graph embedded on the Poincaré disc. This framework simplifies the optimization without strong assumptions.

The decomposition is easy to produce and finite, because the expected impression $I_u$ of each user $u$ is known beforehand and $I_u$ is upper-bounded (due to limited time a normal user can spend on social websites). It is also computationally efficient as HyperCubeMap is linear to the size of users (Sec 2.5.3).

Comparing with the original formulation introduced in Sec 2.1, the dimensionality of unknown $\Theta$ in our formulation in the worst case is $|A| \times |\Lambda| \times |T|$, which is the number of campaigns multiplied by the degree spectrum and the optimal isolated cubes. In comparison, the original optimization problem has $|A| \times |U|$ dimensions, i.e. the product of ad campaigns and users. This improvement is significant as $|A|$ is around one million [48], but $|U|$ is in billions.

## 2.7 Summary

In this chapter, we develop HyperCubeMap, a novel formulation for the SNS ad allocation problem via hyperbolic embedding. We introduce HyperCubeMap, a new hyperbolic embedding method which extends previous methods and addresses the requirements of SNS ad allocation in multiple target group setting. We introduce components like unit impression decomposition in order to handle the challenges such as uncorrelated impression distribution and region overlapping issues in the embedding. With the hyperbolic embedding and unit impression decomposition process over the social graph, the original integer program can be approximated by a series of linear programs for hyperbolic space region allocation, which successfully and largely reduces the dimensionality and complexity of the optimization problem, enabling its application in real-world SNS with billion users.

# CHAPTER 3: EXTENSIONS & EVALUATION OF THE HYPERCUBEMAP-BASED SNS AD ALLOCATION APPROACH

To incorporate more real-world requirements and demonstrate the generality of our framework, we discuss two important extensions of our method. We first discuss the implications of different shape designs and compare with the fan shape that we use in HYPERCUBEMAP. We show how to formulate the fairness constraints using different shapes in the Poincaré circle. Next, we discuss how to handle more complex social influence requirements with respect to $P(u)$. We extend it to multi-hop cases, as well as handling selectivity constraints on the users according to real world billing policies.

In order to evaluate the performance of our approach, we conduct a series of experiments and compare our HYPERCUBEMAP-based LP formulation with the baseline IP approach. Compared to the baseline approach, our proposed method runs two to four orders of magnitude faster, and reaches 95% of the optimal solution.

## 3.1 Accommodating Domain Constraints via Shape Design

Within the ad platform, there may be additional domain constraints besides the basic ones introduced in Eq. 2.19. Among all these domain constraints, fairness

is an important one [49,50]. In SNS, the ad agent may want to distinguish between different advertisers, and assign set of users with different impression quality. The impression quality of a user set can be represented by its degree demographics. Intuitively in an allocation strategy, assigning an ad to one user with 1M friends is different from assigning it to 1M users all with one friend. When the ad platform wants to keep the game fair, the user sets assigned to different ads can admit similar distribution of user impression quality.

To formulate the concept of fairness, we classify the fairness-related domain constraints into three major categories, namely fairness model, priority model as well as partial fairness model based on the differences of allocated user influence demographics among ads.

1. *Fairness model*: Fairness model requires the user influences (degree) demographics among advertisers to be similar. Formally, the constraint for fairness model over the allocation strategy $S$ in the optimization problem can be expressed as:

$$var(\phi(S)) \leq \eta \tag{3.1}$$

where $\phi(S) = (\phi(S_1), \cdots, \phi(S_{|A|}))$ is the fairness measure of user demographics over the vector of optimal allocation; greater $\phi(\cdot)$ corresponds to higher ratio of influential users. Here we use variance to reflect the demographics difference, with $\eta$ as the threshold.

2. *Priority model*: Contrary to the fairness model, the priority model requires more influential users allocated to advertisers of higher priority (e.g. with

higher bids). The allocation constraint for the priority model can be described as:

$$\phi(S_i) \leq \phi(S_j) \qquad \forall A_i, A_j \in A, \rho_i \leq \rho_j \tag{3.2}$$

where $\rho_j$ is ad $A_j$'s priority, and greater value represents higher priority.

3. *Partial Fairness model*: Partial Fairness model (Hybrid model) is between the two extremes mentioned above. If we want both fairness and priority to co-exist in advertisement allocation (i.e. an ad with a low bid is allowed to have some highly influential users), then the allocation strategy should consider both sides:

$$\begin{aligned} &var(\phi(S)) \in [\underline{\eta}, \overline{\eta}] \\ &\phi(S_i) \leq \phi(S_j) \qquad \forall A_i, A_j \in A, \rho_i \leq \rho_j \end{aligned} \tag{3.3}$$

where $\underline{\eta}$ and $\overline{\eta}$ are the lower and upper bounds for the variance.

In the HyperCubeMapbased optimization framework discussed in Sec. 2.6, there is no restriction on which annulus, or isolated cube that an ad should be assigned, so the optimal solution admits no difference between areas in the hyperbolic space that correspond to isolated cubes in different annuli. Thus fairness constraints are not reflected in such setting.

Here we show the connection between the three fairness models and different shape designs. We discuss *Fan*, *Ring*, *Circle*, and other general shapes. We analyze the impact of different allocation strategies w.r.t. convexity, efficiency, and fairness. For ease of discussion and without loss of generality, we look at a simplified setting where the ads all target the whole social network (i.e. single target group), and

illustrate how to incorporate fairness constraints. we consider this problem on a unit-impression graph reached via impression decomposition, thus there is no intersection between regions allocated to different ads.

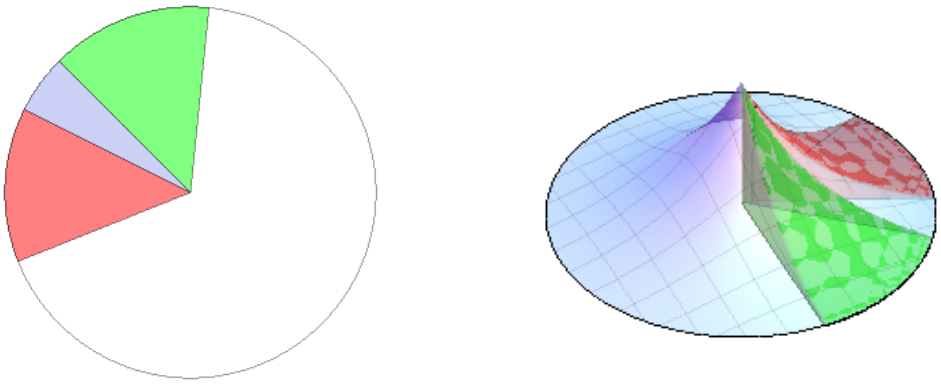
### 3.1.1 Fan

We propose the *Fan*-shaped ad allocation strategy for the *fairness model* discussed above, where each ad is assigned a fan area, as shown in Fig. 5. The allocation area $S_i$ for ad $A_i$ is a fan (or pie) of angle $\theta_i$ in the Poincaré circle of network. Such allocation strategy reflects the fairness model described in Eq. 3.1, as the user degree distribution (i.e. influence demographics) are similar among fan areas assigned to ads due to uniform expected degree distribution in the Poincaré circle along the angular axis. Similar to Eq. 2.17, the corresponding volume function $f_i(S_i)$ is:

$$\begin{aligned} f_i(S_i) = f_i(\theta_i) &= a \int_0^R e^{\tau} \int_0^{\theta_i} (1 + w \cdot \delta(\tau))) d\alpha d\tau \\ &= a \cdot \theta_i (2wc(e^{\frac{R}{2}} - 1) + e^R - 1) = \alpha \cdot \theta_i \end{aligned} \tag{3.4}$$

with $\alpha = a(2wc(e^{R/2} - 1) + e^R - 1)$ a constant. Integrating the unit impression decomposition and fan-shaped allocation strategy, the optimization problem in Eq. 2.16 can be reformulated as a linear programming problem like Eq. 2.19.

Via the *Fan*-shaped allocation strategy, the *fairness model* is well-supported, since areas allocated to different ads have similar demographics due to well-defined node density and expected degree distribution. In Eq. 3.4, $f_i(\cdot)$ is a linear function about $\theta_i$, and leads the optimization problem a linear programming (LP) one, which is another advantage of such allocation strategy. Additionally, fans of different ads



(a) *Fan* allocation 2D (b) *Fan* allocation 3D

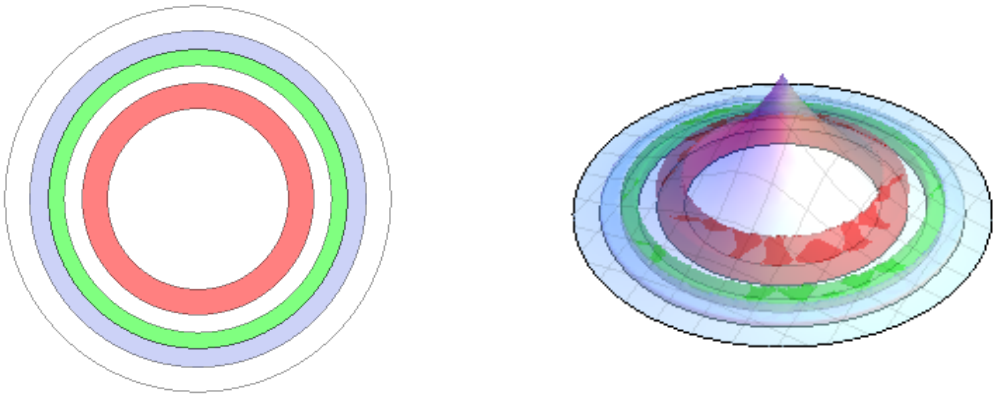
Figure 3.1: Fan allocation strategy

can be arranged tightly close to each other and impressions can be completely utilized in each round of optimization with enough budgets, thus number of iterations are minimized. Furthermore, residual graphs can be generated independently, thus all iterations can run in parallel with careful budget arrangement.

### 3.1.2 Ring

When considering the *priority model* (i.e. Eq. 3.2) by assigning more influential users to higher priority bidders, we propose to use the *Ring* shaped allocation strategy (Fig. 6). Let $r_{i,s}$ and $r_{i,e}$ be the starting and ending radius, and $\rho_i$ be the priority value of $A_i$, the expression for $f_i$ over the ring $[r_{i,s}, r_{i,e})$ in Poincaré disc is a function of $r_{i,s}$ and $r_{i,e}$:

$$\begin{aligned} f_i(S_i) = f_j(r_{i,s}, r_{i,e}) &= a \int_{r_{i,s}}^{r_{i,e}} e^{\tau} \int_0^{2\pi} (1 + w \cdot \delta(\tau))) d\alpha d\tau \\ &= 2\pi a \left(2wc \cdot e^{\frac{r_{i,e}}{2}} - 2wc \cdot e^{\frac{r_{i,s}}{2}} - e^{r_{i,s}} + e^{r_{i,e}}\right) \end{aligned} \tag{3.5}$$



(a) *Ring* allocation 2D (b) *Ring* allocation 3D

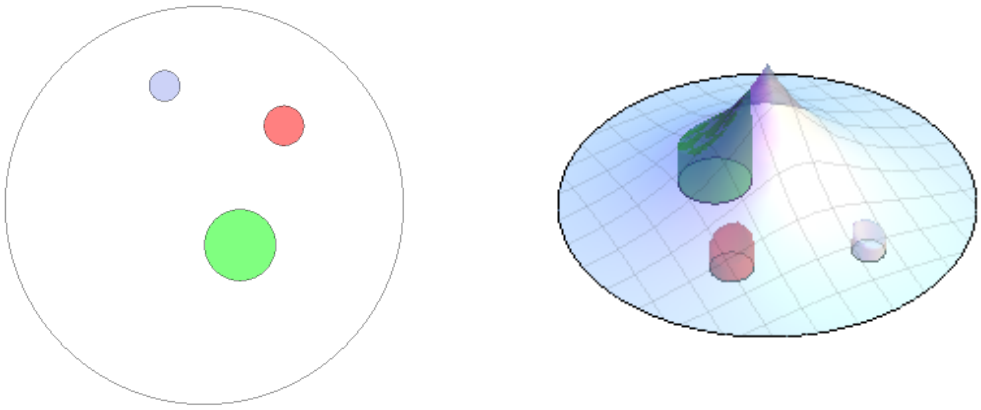
Figure 3.2: Ring allocation strategy

The optimization can be formulated as:

$$\begin{aligned} \max_{S} \quad & \sum_{i=1}^{|A|} p_i f_i(r_{i,s}, r_{i,e}) \\ \text{s.t.} \quad & p_i f_i(r_{i,s}, r_{i,e}) \leq b_i \qquad \forall i \in \{1, 2, ..., |A|\} \\ & 0 \leq r_{i,s} \leq r_{i,e} \leq R \qquad \forall i \in \{1, 2, ..., |A|\} \\ & r_{i,e} \leq r_{j,s} \qquad\qquad\qquad \forall \rho_j \leq \rho_i \end{aligned} \tag{3.6}$$

where the last constraint abstracts the priority model, that ads of higher priority are arranged in inner area. Decision variable $((r_{1,s}, r_{1,e}), \cdots, (r_{|A|,s}, r_{|A|,e})) \in \mathbb{R}^{2|A|}$.

*Ring*-shaped allocation strategy represents the *priority model*, in the sense that ads of different priorities are assigned different demographical population in terms of social influence. As we can see in Eq. 3.5, the volume function $f_i(r_{i,s}, r_{i,e})$ for the *ring*-shaped allocation strategy is nonlinear in $r_{i,e}$ and $r_{i,s}$. Similar to the fan shape case, rings of different ads can be arranged tightly and impressions can be completely utilized in each round of optimization, thus sub-step iterations are



(a) *Circle* allocation 2D (b) *Circle* allocation 3D

Figure 3.3: Circle ad allocation strategy

minimized.

### 3.1.3 Circle

Using circle as the allocation region, as shown in Fig. 7, is a potential solution to incorporate the *partial fairness model* mentioned in Sec. 2.4. The *circle* allocation strategy for $A_i$ can be represented using center position and radius $(x_i, \theta_i, r_i)$. $f_i(S_i)$ therefore is:

$$f_i(S_i) = f_i(x_i, \theta_i, r_i) = a \int_0^{r_i} e^{\tau} \int_0^{2\pi} (1 + w \cdot \delta(dis(x_i, \tau, \alpha))) d\alpha d\tau \tag{3.7}$$

where $dis(x_i, \tau, \alpha) = \sqrt{{x_i}^2 + \tau^2 - 2x_i \tau cos(\alpha)}$ is the distance between a point $(\tau, \alpha)$ from $x_i$ and the disc center.

Using such allocation strategy, the optimization problem can be written as

$$
\begin{aligned}
&\max_{x,\theta,r} \sum_{i=1}^{|A|} f_i(x_i, \theta_i, r_i) \\
&\text{s.t.} \quad f_i(x_i, \theta_i, r_i) \leq b_i, \qquad \forall i \in \{1, 2, ..., |A|\} \\
&\qquad 0 \leq x_i,\ r_j \leq R, \qquad \forall i \in \{1, 2, ..., |A|\} \\
&\qquad x_i + r_i \leq R, \qquad \forall i \in \{1, 2, ..., |A|\} \\
&\qquad (x_i)^2 + (x_j)^2 - 2x_i x_j cos(\theta_j - \theta_i) \\
&\qquad \geq (r_i + r_j)^2 \qquad \forall i, j \in \{1, 2, ..., |A|\}
\end{aligned}
\tag{3.8}
$$

The *circle* allocation strategy can reflect the *partial fairness model*, since circles of similar sizes and similar distances to the center have similar influence demographics, while circles at different positions with different radii have different demographics. It can be tuned by adding size and position constraints. From Eq. 3.7 we can see that $f_i$ is not convex in $\theta_i$. As for efficiency, impressions cannot be fully utilized in each iterations thus more iterations are needed.

### 3.1.4 General Allocation Strategies

As shown in previous sections, shape design is a powerful and intuitive way to represent domain constraints, such as the fairness model. Table. 3.1 summarizes the characteristics w.r.t. convexity, efficiency and corresponding fairness constraint, of different shape-based ad allocation strategies. In addition, it's worth discussing the general allocation strategy to incorporate other domain constraints and show the limitation of our method.

| Shape | Convexity | Efficiency | Fairness Constraints |
|---|---|---|---|
| Fan | Linear | Full space utilization | Fairness model |
| Ring | Nonlinear | Full space utilization | Priority model |
| Circle | Nonconvex | White space between circles | Partial fairness model |

Table 3.1: Features of the three shapes discussed

a) *Convexity:* Convex problems have prominent advantages in solvability, reliability and efficiency. To have convexity, we can design shapes of convex volumes about radial coordinate $r$ and angular coordinate $\theta$. Non-convex volume expressions have many local optima, and require advanced optimization frameworks.

b) *Efficiency:* Another important factor of runtime is the number of unit impression graphs, which implies the number of sub-step optimization routines. The less unallocated area in one iteration, the fewer iterations needed. In a shape design, if all areas can be allocated in each iteration, then we can generate all unit impression graphs regardless of the optimization result, which makes it possible to execute in parallel with careful budget arrangement.

c) *Domain constraint:* As we showed above, the fairness constraint is defined over the user influence demographics. Because it is well-defined over the Poincaré disc, we are allowed to use fan, ring or circle to specify different fairness models. Other business rules that have well-defined metrics over the graph's degree also have the potential to apply in our framework.

### 3.1.5 Extension to Multiple Target Groups

If we extend the idea discussed above to the general setting where there are multiple target groups within the SNS, the domain constraints can be imposed into the optimization framework by introducing additional constraint functions in a similar way.

For the fairness model, we can combine the areas that correspond to the same isolated cube at all annuli:

$$\Delta_c = \sum_{\lambda} \Delta_\lambda (\theta_e^{\lambda,c} - \theta_s^{\lambda,c}) \tag{3.9}$$

where $\Delta_\lambda$ is same as the one in Eq. 2.17. Then the optimization can be re-formulated as:

$$\begin{aligned} \max_{\Gamma} \quad & \sum_{A_i \in A} p_i \sum_{c \in T_i} \Delta_c \gamma_i^c \\ \text{subject to} \quad & \gamma_i^c \geq 0 \\ & p_i \sum_{c \in T_i} \Delta_c \gamma_i^c \leq b_i \\ & \sum_{A_i \in A} \gamma_i^c \leq 1 \\ & \forall A_i \in A, c \in T_i \end{aligned} \tag{3.10}$$

where the optimization variable $\gamma_i^c$ is the proportion of area on the isolated cube $c$ assigned to ad $A_i$.

For priority model, similar to the last constraint in Eq. 3.6, we can add one more constraint function in Eq. 2.19 requiring that the annuli assigned to ads of

higher priority should be inner than those of lower priority:

$$\begin{aligned} min\ \{\lambda|\theta_i^{\lambda,c} > 0\} \leq min\ \{\lambda|\theta_j^{\lambda,c} > 0\} \\ \forall \rho_i \geq \rho_j, A_i \in A, c \in T_i \end{aligned} \tag{3.11}$$

## 3.2 Social Influence Models

In previous sections, we mainly consider 1-hop neighbors and model it as a linear function of engagement rate and degree, by assuming the influence is shallow [49]. In real world, different ad format may have different influence impact and more complex influence functions may be needed. For example, recent work [51] shows the cascading of popular photos in SNS may not be shallow. If applying HyperCubeMap on ads with non-shallow ad format, our model need to be adjusted. When the click-through rate is too high to neglect the influence of a user's activities (e.g. clicking an ad) over her multiple-hop friends (e.g. 2-hop ones), the social influence function $P(u)$ is required to be modified to reflection the multi-hop influence. Meanwhile, in real-world SNS ad campaign, the users reached via social diffusion may not lie in ads' target groups, which is related to the concept of effectiveness in users' social influences.

### 3.2.1 Multi-Hop Influence

For a user $u$ of degree $d_u$ in the network, if we consider the $k$-hop influence within the network in the IP formulation, the expression of her social influence can

be written as:

$$
\begin{aligned}
P(u) &= w \sum_{v_1 \in F_u} (min\{I_u, I_{v_1}\} + P(u, v_1)) \\
P(u, v_1) &= w \sum_{v_2 \in F_{v_1}} (min\{I_u, I_{v_1}, I_{v_2}\} + P(u, v_1, v_2)) \\
&\cdots \\
P(u, v_1, \cdots, v_{k-1}) &= w \sum_{v_k \in F_{v_{k-1}}} min\{I_u, I_{v_1}, \cdots, I_{v_k}\}
\end{aligned}
\tag{3.12}
$$

As an approximation, in the LP formulation, we can use the expected multi-hop neighbor set size of a isolated cube in each annulus to represent the value of a certain user in the cube to make the social influence expression integrable.

$$
P(u) = P(r_u, \theta_u) = w \cdot ce^{-r_u/2} + \sum_{l=2}^{k} (w^l n_{\lambda,l}) \tag{3.13}
$$

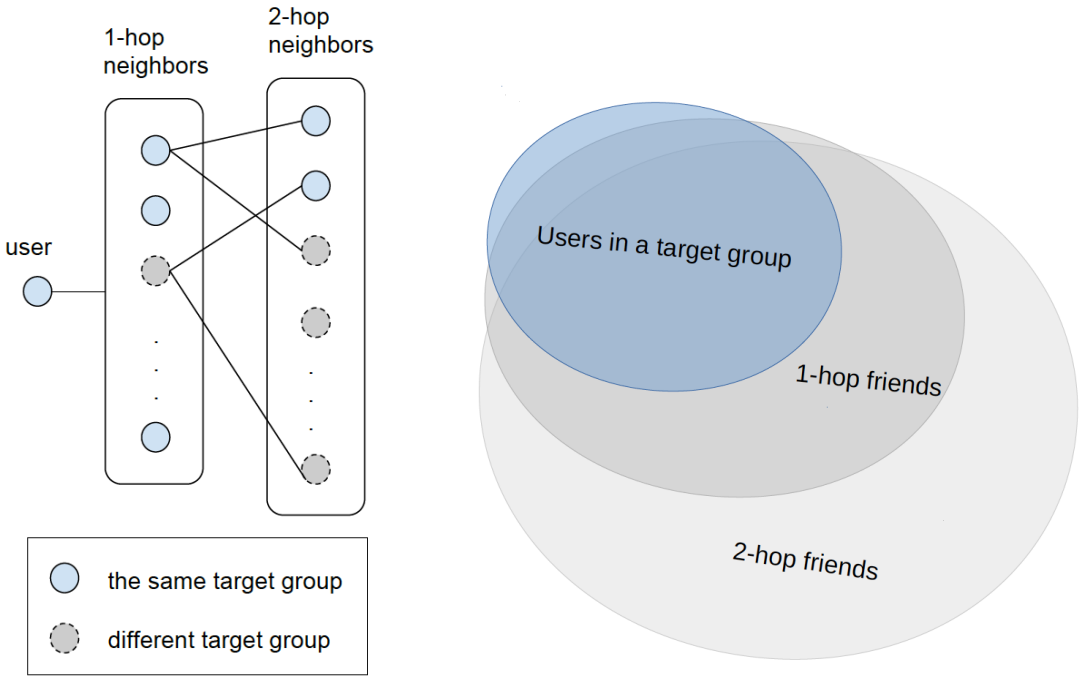
where $n_{\lambda,l}$ is the expected $l$-hop neighbor size of the isolated cube in annulus $\lambda$ that $u$ belongs to:

$$
n_{\lambda,l} = E_{v \in ic_\lambda}[n_{v,l}]\Big|_{u \in ic_\lambda} \approx \frac{\sum_{v \in ic_\lambda} n_{v,l}}{|ic_\lambda|}\Big|_{u \in ic_\lambda} \tag{3.14}
$$

Note that since the second part in Eq. 3.13 is a constant, putting Eq. 3.13 into Eq. 2.19, the optimization is still a linear program.

### 3.2.2 Effectiveness in Billing Models

In real world SNS ad billing models, there is a concept of effectiveness in users' social influences. For example, for a user in one isolated cube, her neighbors are not necessary having the same target group w.r.t. the current isolated cube. If a billing policy enforces to charge only for the influences over the same target group, the way




(a) Selectivity for a single user

(b) Selectivity for a target group

Figure 3.4: Selectivity of multi-hop friends

to calculate the budget in both formulations need to be updated. Our framework can easily be extended to handle this case. To incorporate this, we introduce the term *selectivity* as a measure of effectiveness within an isolated cube, which can be defined as the probability that users an ad reaches via social influences are still in the same target group. Fig. 3.4 is an illustration of selectivity of multi-hop neighbors (friends) in the SNS graph.

In order to introduce selectivity into current optimization framework, we can use $\psi_{\lambda,c,k}$ to denote the $k$-hop selectivity of the isolated cube $c$ in annulus $\lambda$, which can be calculated via the proportion of $k$-hop neighbors of the isolated cube that is still in the cube. By such definition, the linearity of the social influence function is

well-kept:

$$P(u) = P(r_u, \theta_u) = w \cdot ce^{-r_u/2}\psi_{\lambda,c} + \sum_{l=2}^{k}(w^l n_{\lambda,l}\psi_{\lambda,c,l}) \tag{3.15}$$

where $\psi_{\lambda,c} = \psi_{\lambda,c,1}$ the selectivity of 1-hop neighbors.

Placing Eq. 3.15 into Eq. 2.19, we can adjust our LP formulation to handle this type of billing constraints, and the modified formulation is still a LP.

## 3.3 Evaluation

In order to show advantages of our formulation over the original IP formulation, we conduct experiments on our hyperbolic embedding-based approach and the baseline IP (SnsIP) on synthetic data using IBM CPLEX optimizer (v12.6). We implement HyperCubeMap, the hyperbolic embedding algorithm mentioned in Sec. 2.4, and the unit graph impression optimization routine in Alg. 3. We discuss the experimental results on the optimization framework that is developed based on HyperCubeMap, followed by the results regarding shape design in a simplified setting of single target user group.

### 3.3.1 Dataset Description

We construct our dataset using distributions observed from public available real world advertising datasets. On the advertiser side, we look at keyword bidding and budget distributions from the Yahoo! Webscope dataset A1 [52] and open advertising dataset collected from Google AdWords used in [53]. We find that campaign bidding prices fit well with the lognormal distribution, and the advertiser

budget follow Pareto distribution approximately. On SNS activity side, we use the Stanford Network Analysis Platform (SNAP) 2.2 [54] to generate power law networks by setting $\alpha = 2.2$ [55] and varying the network size. In addition to degree distribution, we also need to assign daily impression to each node. The real impression distribution of well-known SNS is not available to the public to the best of our knowledge. To generate it, we argue a real user's SNS usage is bounded by her daily time, thus we model user impressions using a Poisson distribution, which is also reported in real advertising network study [56]. To cluster users with different profiles into targetable user groups of different sizes, we use $|GR|$ to represent the group/user ratio, and use a Dirichlet prior to generate a multinomial distribution over group size. We then apply HyperCubeMap to embed the generated network with default spectrum width $d = 10$. Finally, to generate bidding from campaigns to users, we use $|AR|$ as the advertiser/user ratio and use bipartite preferential attachment with two Zipfian distributions to represent the nodes popularity. The list of parameters and the default values are shown in Table 3.2. Then we vary the number of users from 10K to 100M, apply Alg. 1 to derive the optimal isolated cubes, and summarize the data in Table 3.3. All data and codes are available online[1].

### 3.3.2 Performance of HyperCubeMap

In the following, we refer HembExp to the linear program of exponential node density distribution in Eq. 2.19, and HembUni to the one using uniform node density distribution in Eq. 2.18. As hyperbolic embedding is essentially an approx-

[1]http://www.cs.umd.edu/~hui/code/hypercubemap

| Name | Default | Description |
|---|---|---|
| $\|NU\|$ | 10,000,000 | Number of User |
| $\|AR\|$ | 0.001 | Advertiser/User ratio |
| $\|GR\|$ | 0.0005 | Isolated Cube/User ratio |
| $d$ | 10 | Spectrum degree width |
| $w$ | 0.003 | Click through rate |
| $\lambda$ | 10 | Poisson for user impression |
| $(K, \vec{\alpha})$ | $(\|NU\| \cdot \|GR\|, \vec{1})$ | Dirichlet prior for isolated cube size |
| $(\mu, \sigma)$ | $(-1, 1)$ | Lognormal for advertiser bid |
| $\alpha$ | 1.2 | Pareto I for advertiser budget |

Table 3.2: Parameters of dataset generation

| $\|NU\|$ | edge | $\|A\|$ | $ic$ | $opt_{ic}$ | $\sum_u I_u$ |
|---|---|---|---|---|---|
| 5M | 13.1M | 5000 | 2500 | 1462 | 50M |
| 10M | 26.8M | 10K | 5000 | 2722 | 100M |
| 50M | 137.0M | 50K | 25K | 11308 | 500M |
| 100M | 276.2M | 100K | 50K | 21063 | 1B |

Table 3.3: Summary of Dataset

imation algorithm through dimension reduction, our experiments aim at showing the advantages of hyperbolic embedding over the original IP formulation in terms of runtime, scalability and optimality. We also show the degree spectrum parameter $d$ tuning to trade-off between runtime and optimality. All experiments run on a linux

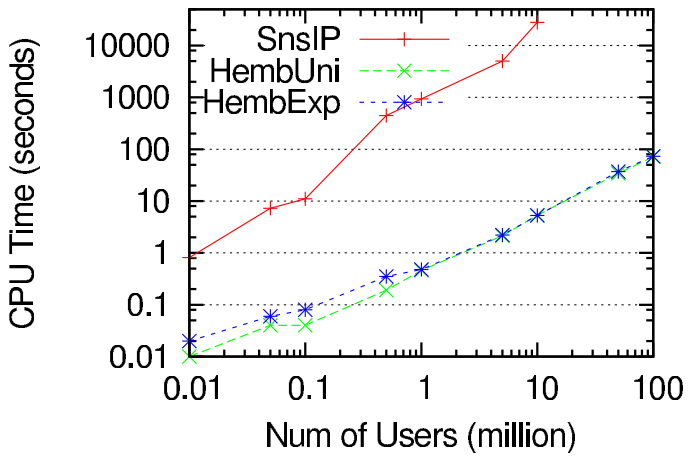



Figure 3.5: Run time by varying problem size

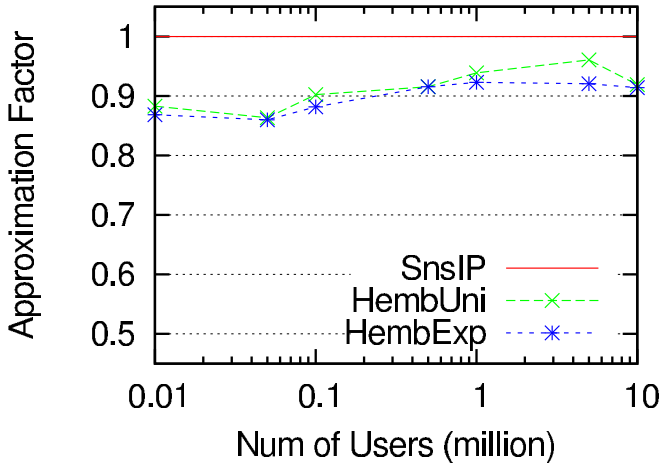



Figure 3.6: Approximation factor

server with two 2.66 GHz 6-core Xeon X5650 CPUs and 128G memory. The CPLEX optimizer is configured to utilize all 24 threads; for the IP, we fix the *MIPSearch* parameter to use the branch and cut. The time metric are in seconds and collected via CPLEX timer representing actual CPU time used in the optimization.

We first show the runtime performance by varying network size in Fig. 3.5. In general, hyperbolic embedding methods HembExp and HembUni finish the optimization process two to four orders of magnitude faster than the baseline SnsIP. In

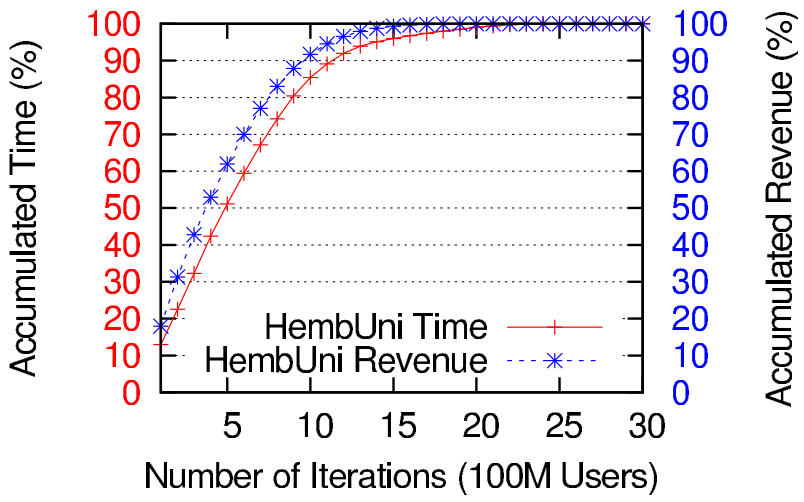



Figure 3.7: Accumulated time and revenue

the 10M networks, SnsIP takes 7 to 8 CPU hours on average to finish, while HembExp and HembUni use less than 100 CPU seconds in the optimization. HembExp and HembUni runtime performance are similar, even in real world size networks (100M). Besides runtime, the HyperCubeMap-based methods require much less memory than the baseline IP. Networks of size 50M and 100M cannot run under SnsIP, as they run out of memory. In comparison, HembExp and HembUni only use 2G memory for network 100M, due to the dimension reduction in our approach.

Next in Fig. 3.6, we show the optimality result using approximation factor P, for instance, in HembExp case:

$$\mathrm{P}_{\mathrm{HembExp}} = \frac{\sum_{i=1}^{\max\_iter} OPT_{\mathrm{HembExp}}}{OPT_{\mathrm{SnsIP}}} \tag{3.16}$$

As IP cannot run on 50M and 100M network, we omit those SnsIP data points. The solution of HembExp and HembUni reach about 90% of the original IP solution on average, and when network size increases, hyperbolic embedding methods have better solutions. In our experiments, the minimum value of P is

85.97% while the maximum is 96.07%. Also HembUni always performs better than HembExp with little additional cost. The exponential node density distribution makes the embedding coefficient less accurate in the center regions, where the users have higher social influences. If the engagement rate $w$ becomes larger, the difference between $P_{\text{HembExp}}$ and $P_{\text{HembUni}}$ will become larger as well.

In Fig. 3.7, we show the accumulated revenue and time in the unit decomposition optimization process in the HembUni experiment on 100M network; HembExp has very similar performance. The left y axis in red is accumulated time percentage, and the right y axis in blue is accumulated optimal objective value. Our optimization process spends most time on the early iterations which also contribute similar percentages in revenue. This observation has practical meanings when the advertiser demand is high: we can decompose the whole graph into small number of unit impression graphs without deducting the budgets in the optimization sequence. These early iteration graphs can be prepared and run in parallel, and the aggregates are good enough to use as an estimate of optimal solution.

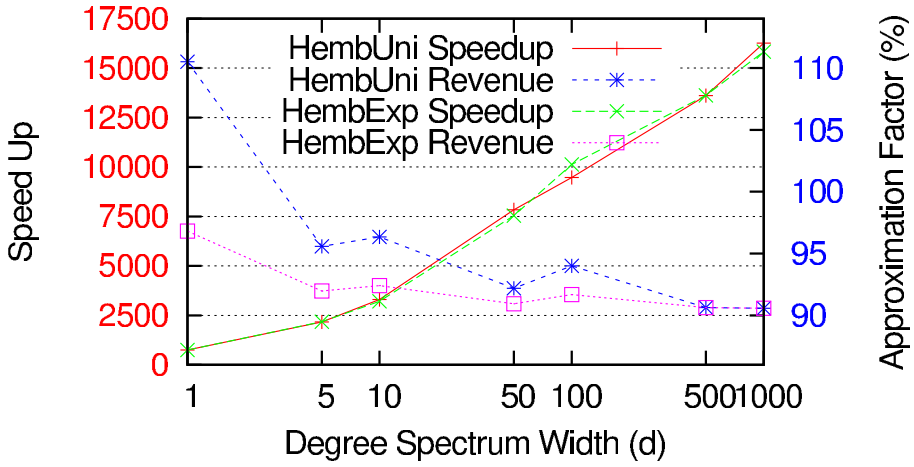



Figure 3.8: Effect of tuning degree spectrum width $d$

Next we show the parameter tuning of our hyperbolic approach. The degree spectrum width $d$ affects dimensions reduction directly and is independent from the SNS itself. We vary $d$ in $\{1, 5, 10, 50, 100, 500, 1000\}$ to see its impact with respect to runtime speedup and the approximation factor. In the extreme case, $d = 1$, each annulus only contains the users with the same degree. As shown in Fig. 3.8, increasing $d$ reduces more dimensions, thus the speedup (left y axis) increases, and HEMBEXP and HEMBUNI have similar benefit. On the other hand, the approximation becomes less accurate, and the approximation factor decreases. As expected, it is easier to tune $d$ in HEMBUNI than HEMBEXP, and it's worth pointing out when $d = 1$, $\mathrm{P}_{\text{HEMBUNI}}$ is greater than 1. From speedup and approximation factor aspects, we suggest to set $d$ around 10, which is where the two curves intersect.

As shown in the experimental results, HYPERCUBEMAP has prominent advantage in efficiency while reaching solutions close to the optimal values.

### 3.3.3 Experimental Results on Fairness Constraints

In this section, we evaluate the performance of our extensions on fairness constraints, aiming to compare the impacts of additional fairness constraints towards the optimization results.

We used SNAP 2.2 to generated graphs of power-law degree distribution and sizes to be $1,000$, $10,000$ and $100,000$. User impressions follow a Poisson distribution with mean $\lambda = 10$. As shown in Table. 3.4, we fixed the number of ads to be 10, each $a_j$ bids $p_j \sim \mathcal{N}(0.1,\ 0.01)$. For the three graphs, we generated the bud-

| $\|NU\|$ | $\|A\|$ | ad bids distr. | ad budgets distr. |
|---|---|---|---|
| 1k | 10 | $\mathcal{N}(0.1,\ 0.01)$ | $\mathcal{N}(15,\ 25)$ |
| 10k | 10 | $\mathcal{N}(0.1,\ 0.01)$ | $\mathcal{N}(150,\ 2500)$ |
| 100k | 10 | $\mathcal{N}(0.1,\ 0.01)$ | $\mathcal{N}(1,500,\ 250k)$ |

Table 3.4: Summary of the dataset

gets of ads from normal distributions $\mathcal{N}(15,\ 25), \mathcal{N}(150,\ 2500), \mathcal{N}(1,500,\ 2.5\times10^5)$ accordingly. Both baseline IP and our novel approach are based on the same impression decomposition procedure. Without loss of generality, we compare both models via the optimization over the first graph $G^{(1)}$ after the impression decomposition operation.

To model the fairness constraints in the IP formulation, we added the following constraints in Eq. 2.1:

1. *Fairness* model: We define the linear constraint as:

$$|\frac{\sum_{u\in S_i} d_u}{|S_i|} - d_V| \leq \eta, \qquad \forall A_i \in A \tag{3.17}$$

where $d_u$ is the degree of $u$, $d_V$ is the average degree of the whole network graph, $\eta$ is the threshold to measure the deviation of the user influence demographics.

2. *Priority* model: We define the linear constraint as:

$$d_u \leq d_v, \qquad \forall u \in S_i, v \in S_j,\ A_i, A_j \in A,\ s.t.\ \rho_i \leq \rho_j \tag{3.18}$$

where $\rho_i$ is the quantized priority of ad $A_i$. The constraint enforces advertisers

| | Network size | 1,000 | 10,000 | 100,000 |
|---|---|---|---|---|
| Revenue | Baseline IP | 108 | 1157 | 11703 |
| | Baseline IP (Priority) | 108 | 1157 | 11700 |
| | Baseline IP (fairness) | 108 | 1157 | 11703 |
| | Fan shape allocation | 108 | 1156 | 11669 |

Table 3.5: Optimal values reached via different approaches

with higher bid have the users with higher influence (i.e. larger degree) in the model.

3. *Partial fairness* model: The partial fairness model can be formulated as a combination of the first two models described in Eq. 3.17 and Eq. 3.18.

$$\begin{aligned} &\sum_{u \in S_i} d_u \leq \sum_{v \in S_j} d_v, \qquad \forall A_i, A_j \in A,\ s.t.\ \rho_i \leq \rho_j \\ &|\frac{\sum_{u \in S_i} d_u}{|S_i|} - d_V| \leq \eta, \qquad \forall A_i \in A \end{aligned} \tag{3.19}$$

Other models can formulate constraints accordingly.

In order to explore the influence of additional fairness constraints in optimization, we compare the optimal values reached by the baseline IP formulations with or without additional fairness constraints and the fan-shape allocation under various network sizes. The results are shown in Table 3.5, from which we notice the followings:

- Different fairness constraints lead to different optimal solutions, but they reach similar optimal values, i.e. the maximum profit that the ad agent can earn via

ad allocation. This result is corresponding to the pay-per-mille model applied in the optimization setting, where impressions are charged instead of clicks.

- The new approach has good performance in approaching the optimal value. This is consistent with the results shown in Sec. 3.3.2, that HyperCubeMap can reaching solutions close to the optimal values.

## 3.4 Summary and Future Direction

In this chapter, we propose extensions to the HyperCubeMap-based optimization framework. First we discuss how to incorporate the fairness constraints using different shape designs and their algorithmic complexities. As in HyperCubeMap, the influence surface is uniform along angular axis, different influence domain constraints, such as fair or prioritized assignment strategies among advertisers can be represented using different shapes on the isolated cubes of different annuli (layers) to the ads. Furthermore, in addition to the 1-hop model applied in the formulation, we show multi-hop models for the social influence function $P(u)$ and possible approximations, in order to incorporate non-shallow cascading ad format in real world applications. In general, HyperCubeMap works well with minor modifications. We also conducts a series of experiments to evaluate the performance of the new hyperbolic embedding-based approach. Experimental results show that the new approach largely reduces the runtime and simplifies the optimization problem, enabling finding the optimal solution fast and robustly.

Though our algorithm is for offline purpose, it has the potential to be applied

online. For online usage, we need to solve the issue of network and bidding updates. Users with attribute updates can be simply assigned to different isolated cubes, while users with friends change or new users will be embedded using the current algorithm. For bids updates, the optimal isolated cubes need to be split or merged smartly without re-calculating. Periodical adjustments need to be performed as well in order to ensure the performance. We will leave it as future work.

# CHAPTER 4: TRUST IN SOCIAL NETWORKS

With the fast development of Internet and IT technologies, online social network services (SNS) such as Facebook and Twitter are gaining tremendous popularity. Via exposing personal behaviors and connecting to each other, hundreds of millions of users interact and exchange information over these platforms [57]. The sharply increasing amount of information flowing in SNS brings a significant benefit to the users in SNS as they could make decisions via collecting and combining information from different sources (i.e. other users) in the network. Such a phenomenon opens a promising market for SNS-based applications. For example, in recommender systems, the preference of a user towards a product/service can be predicted based on the information about users with similar tastes [58], and the recommendations can be further personalized based on the social context [59–61].

Trust in social network setting can be seen as the preference of a user towards her neighbors (e.g. friends) in the social network and provides a guideline for the user to interact and make decisions [62]. Information on trust relationships in social networks is very important for the success of many SNS-based applications.

## 4.1 Concept of Trust

The concept of trust has been noticed and raised attention for a long time, recognized as an relationship aligned with human interaction. Trust, as defined in the Merriam-Webster dictionary, is the "belief that someone or something is reliable, good, honest, effective, etc" [63]. In modern age, with the fast development and wide application of information and communication technologies, trust has been largely extended from physical interactions between humans to a much wider range of domains, from e-commerce to smart grids, from recommender system to sensor networks. Research on trust now stands at the crossroads of several distinct research communities, including sociology, economics, computer science, and so on.

The wide application of trust makes it an umbrella term with multiple interpretations in different contexts. Trust can be defined in a quantifiable way so that it can be measured, evaluated and applied in computational tasks. In [64], Jøsang et al. discussed two common definitions of trust used among various scenarios, which are *reliability* and *decision* trust. Reliability trust can be interpreted as the probability held by the evaluating individual $a$ (truster) that the target individual $b$ (trustee) would perform a given action on which $a$'s welfare depends. On the other hand, decision trust links to the extent to which an agent (i.e. the truster) is willing to depend on another one (trustee) in a given situation for decision making, with a feeling of similarity, closeness or security.

In the Public Key Infrastructure (PKI), trust is used for authentication and secure transactions [65, 66]. In P2P network, a global trust is evaluated to regulate

the interactions among users in the network [67]. Trust is also an important concept for security in Ad Hoc Networks, influencing processes like intrusion detection and access control [68, 69]. In these settings, trust is in the flavor of the reliable trust discussed above, as it is seen as a measure of integrity or level of confidence about other entities in cooperation, from past experiences, knowledge about the entities and/or recommendations from trusted entities. Trust applied in these scenarios can be categorized as reliability trust.

In the case of social network scenario, trust is interpreted in a more subjective way, and the notion of decision trust is applied. Here trust is directed and is a compound of integrity, preference/taste similarity, and social closeness (subjective similarity). It can be defined as, in a certainty domain (context), the extent to which a truster will consider the trustee's opinions in such domain. Such a definition makes trust a domain-specific concept that is quantifiable and tailored for social network scenarios. In discussing trust, there are several dimensions [70], including :

- *Trust measure*: The range of the trust measure can be from complete distrust over a neutral trust measure to full trust. The more a trustee is trusted, the higher the trust measure should be.

- *Trust certainty*: The confidence of the truster in her estimated trust value about the trustee.

- *Trust context*: The context of domain within which people reach their trust statements, when defining trust in a fine-grained manner. A context can be different categories.

- *Trust directions*: Based on different interpretations of the reciprocity in trust, trust relationships can be defined as direct and indirect ones.

- *Trust dynamics*: A trust relationship may change dynamically along time with increasing amount of evidence and change of trustee's personality.

## 4.2 Modeling Trust in Social Networks

The trust relationship between users in SNS is very important in decision making as well as the success of many SNS-based applications like recommender systems. When modeling trust in social networks, it is natural to think of the trust relationships as potentially asymmetric. For example, in a social network, Alice likes Bob but Bob may not like Alice in the same extent. The network of trust can be formed based on the trust relationships that connect people in the social network.

### 4.2.1 Trust and Distrust Relationships

In social network scenario, trust relationship is based on the social connection between truster and trustee.

In describing the trust relationship between the evaluating agent (truster) and the target agent (trustee), there are three different types of trust, namely real, direct and evaluated trust, that we need to distinguish.

- *Real trust*: the real state of the target node's trust value (reputation).

- *Direct trust*: the trust opinion that the truster holds about the trustee that is reached via direct interactions.

- *Evaluated trust*: the trust value derived from the pre-defined evaluation rule. Most of the time, the evaluated trust values are derived through trust propagation and aggregation.

Direct trust can be either expressed explicitly (e.g. ratings provided by users in an SNS), or derived from interactions with the target node based on certain definition of policies, either using deterministic models or stochastic approaches. In the rest of this thesis, we assume that direct trust are provided explicitly.

There are various ways to quantitatively represent trust [64, 71–73]. The trust value domain $\mathcal{T}$ is application dependent. For instance, it can be a set of discrete labels $\{0, 1\}$ or a continuous range like $[0, 1]$. It can have both positive values and negative values, and can even be a multi-dimensional vector space. When $\mathcal{T}$ contains negative trust values, it can be used to differentiate unknown users (e.g. of trust value 0) from ones that are not trusted (with negative trust values). Various representations of trust exist in related literature. Some work define trust values as real numbers in certain intervals like $[1, 1]$ (e.g. [74, 75]), or probabilities in $[0, 1]$ as [70, 76, 77]. Others use discrete values, like the binary representation used by Jiang et al. [78] and the four discrete-valued setting introduced in [79].

The negative trust relationship is also called distrust. While theories on trust is increasingly established, the use and modeling of distrust remains relatively unexplored. Although recent research work [72, 73, 80] shows an emerging interest in modeling the notion of distrust, models that take into account both trust and distrust are still limited. Most approaches completely ignore distrust, or consider

trust and distrust as opposite ends of the same continuous scale. However, there is a growing body of opinion that distrust cannot be seen as the equivalent of lack of trust [62, 81]. Introducing distrust makes the trust model complex as the non-negativity of trust values no long exists, and the linearity of trust aggregation is hard to argue. This can be especially challenging for many trust-based approaches such as matrix factorization [82].

Trust relationships that connect people in the social network form a trust network.

### 4.2.2 Trust Network

Based on the trust relationships among people in a social network (SNS), a trust network can be established. The main aim in setting up the trust network is to allow agents to form trust opinions on unknown agents or sources by asking for trust opinions from acquainted agents in the social network. The trust network is usually described as a directed graph, with each vertex as a person and each edge denoting the directed trust relationship. A weight can be attached to the edge to represent the level of trust placed upon the trustee by the truster. In such setting, the symmetric situation can be seen as having two reciprocal and equally weighted edges on both directions between the two vertices (nodes).

Referring to previous work [59,62,72,73,81,83,84], we define the trust network in social network setting as follows.

**Definition.** *Trust Network*: A trust network $\mathcal{G}(V, E, t_e)$ is a directed and weighted

graph based on the trust relationships in a social network, where $V$ is the set of nodes (i.e. users) with $|V| = N$ size of the graph. $E$ is the set of connections (i.e. trust links) denoting the directed trust relationships, $\forall$ directed edge $e_{ij} = (v_i, v_j) \in E, v_i, v_j \in V$, is a directed trust link from node $v_i$ towards $v_j$. $t_e : E \mapsto \mathcal{T}$ is a mapping from an edge to the trust value placed on the edge. Trust links are not necessarily symmetric. $N_i = \{v_j | e_{ij} \in E\}$ is the neighbor set of node $v_i$.

In the defined trust network, a directed path of length $k$ is a sequence of distinct nodes, $\{v_1, v_2 \cdots, v_{m+1}\}$, such that $\{v_i, v_{i+1}\} \in E, \forall i \in 1, 2, \cdots, m$. Between two nodes in the network, there might be multiple distinct paths.

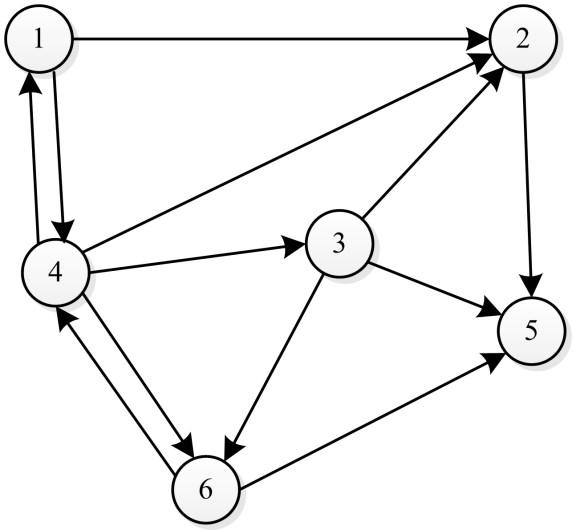



Figure 4.1: An example of trust network

Fig. 4.1 gives an example of trust network, where $V = \{v_1, v_2, \cdots, v_6\}$. There are 12 directed edges, each representing the trust relationship between truster (tail) and trustee (head). For node $v_3$, its neighbor set is $N_3 = \{v_2, v_5, v_6\}$.

Trust networks are challenged by two major issues regarding trust opinion

formation towards unknown nodes. Firstly, the trust network is naturally sparse; in a large network it is likely that many agents do not know each other, hence there is an abundance of ignorance. Secondly, because of the lack of a central authority, different agents might provide different and even contradictory opinions, hence inconsistency may occur.

### 4.2.3 Transitivity in Trust

Trust transitivity describes how a trust rating can be passed through a chain of people, and is the foundation of most trust inference models. It allows the truster to acquire information about the trustee from her friends and theirs' ("word of mouth") [85]. For example, as shown in Fig. 4.2, if node $v_1$ and $v_3$ are not directed connected, but if $v_1$ trusts $v_2$, and $v_2$ trusts $v_3$, then $v_1$ can use such trust evidence to infer its opinion about $v_3$ using (possibly partial) transitivity.

As trust is a mental phenomenon, it is barely possible to reach an objective definition for trust transitivity, which leads to different interpretations in different trust models. Four types of trust transitivity models were mentioned in [81], namely direct propagation, co-citation, transpose trust and trust coupling. Among these four categories, direct propagation is mostly considered and can be seen as the classical transitivity. It can be defined as:

$$\begin{aligned} \langle v_i, v_j \rangle_{\text{trust}} \wedge \langle v_j, v_k \rangle_{\text{trust}} \Rightarrow \langle v_i, v_k \rangle_{\text{trust}} \\ \forall v_i, v_j, v_k \in V \end{aligned} \tag{4.1}$$

Based on the assumption of transitivity (may be partial), trust can propagate along the paths between two nodes in the network and their trust relationship can

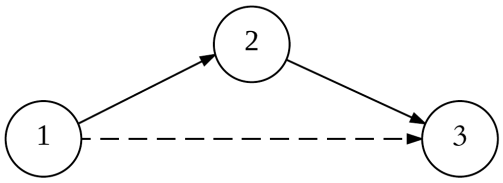



Figure 4.2: Transitivity of trust relationships

be inferred. When multiple paths exist, a combination scheme is needed to derive the trust value. Due to the subjective nature, conflicts may happen in such occasion and need to be resolved.

Transitivity becomes more complex when considering distrust relationships [86]. For example, the trust relation between node $v_1$ and $v_3$ is not obvious if $v_1$ distrusts $v_2$ and $v_2$ distrusts $v_3$. The conflict resolution is also much more difficult with distrust.

### 4.2.4 Reciprocity

Trust reciprocity describes the extent of symmetry in directed trust relationships between two users (i.e. nodes) in the social network. Similar to Eq. 4.1, we can write the following expression for reciprocity.

$$\begin{aligned} &\langle v_i, v_j \rangle_{\text{trust}} \Rightarrow \langle v_j, v_i \rangle_{\text{trust}} && \text{(trust reciprocity)} \\ &\langle v_i, v_j \rangle_{\text{distrust}} \Rightarrow \langle v_j, v_i \rangle_{\text{distrust}} && \text{(distrust reciprocity)} \\ & && \forall v_i, v_j \in V \end{aligned} \tag{4.2}$$

As shown in [83, 87], trust relationships are asymmetric in terms of values (especially magnitude) in social network scenarios, thus most trust evaluation ap-

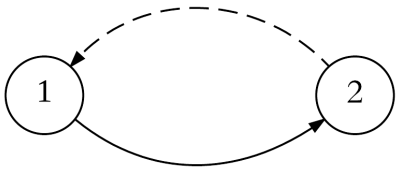



Figure 4.3: Reciprocity in trust relationships

proaches don't consider reciprocity. However, by relaxing the condition in Eq. 4.2 to only sign agreement of positive trust relationships, partial reciprocity can be defined. As we will show in Sec. 5.10.1, the partial reciprocity exists in the Epinions trust network dataset. The partial reciprocity between nodes may be useful in inferring indirect trust in some circumstances.

## 4.3 Evaluating Trust in Social Networks

In literature, two classes of trust metrics have been studied, which are local trust metrics [83, 88–90] and global ones [67, 91–93]. Local trust (AKA personalized trust value) refers to the subjective trust opinions of each agent about her neighbors in the network, which provides a personalized trust value that depends on the point of view of the evaluating agent (e.g. node). Global trust of a node in the social network is a unique trust value assigned to her, independently of the evaluating agent. The ground truth for global trust corresponds to the *real trust* mentioned in Sec. 4.2.1, and is also called as reputation in many cases. Global trust is a quantity derived from the underlying social network, and is globally visible to and generally accepted by members of the network.

### 4.3.1 Local Trust Evaluation

Evaluation of local trust depends on past interactions between the truster and the trustee, and on the transitivity ("word of mouth") model that allow the truster to acquire information about the trustee from her friends and theirs [85]. There have been a series of research work on the evaluation of local trust.

Golbeck proposed TidalTrust [83, 94] for inferring trust relationships between people with no direct connection based on shortest trust paths between them within the trust network. The algorithm aggregates the weighted trust values between neighbors to reach indirect trust. In the following work [95], trust paths of different length are considered. MoleTrust [96, 97] proposed by Avesani et al. is similar to TidalTrust, but considers all raters up to a fixed maximum-depth given as an input.

Jøsang et al. [89] used a subjective logic framework for local trust reasoning. Trust and distrust are treated as two separate concepts, and several probabilistic trust aggregation operators are proposed according to different transitivity models.

DuBois et al. [98] designed a probabilistic approach to infer the trust relationship between users in social networks and applied it in network clustering. In their following work [86], they further considered distrust in the network and introduced a modified spring-embedded algorithm for trust inference. Kuter et al. [99] developed the SUNNY algorithm for trust inference based on probabilistic confidence models. Huang et al. [100] used a probabilistic soft logic framework for trust prediction.

### 4.3.2 Global Trust

The global trust of a node in the trust network is a general proposition about the node's integrity, reliability, and quality, which is based on and comes from the reputation of the trustee within the community. An individual's global trust is estimated using the information from the complete trust network [71]. The converged global trust can be reached via combination of local trust opinions gathered from all the agents within the network.

The most basic approach is to take average over all trust opinions about the target node in the trust network. Such a straightforward approach has been applied in many SNS as well as e-commerce platforms with minor modifications. Despite the efficiency of the method, a lot of information contains in the trust network may be lost due to the low-pass filtering effect of the averaging operation.

The game theory-based research introduced in [101, 102] lays the foundation for research on online reputation systems and provides interesting insight into the complex behavioral dynamics. Mui et al. [103] also give a review summarizing existing works on reputation across diverse disciplines, including distributed artificial intelligence, economics, and evolutionary biology.

Many global trust metrics applied the idea of the PageRank algorithm [104] used for web page ranking. For example, Kamvar et al. [67] developed a PageRank-like algorithm to evaluate global trust of peers in P2P network via aggregating local trust values. The issue of malicious nodes is addressed by computing trust value on other peers and majority voting. Richardson et al. [88] proposed an algorithm similar

to PageRank for global trust evaluation in the semantic web. These approaches imply that nodes should be ranked higher when pointed by nodes of better ranks.

## 4.4 Study Objectives

In the following chapters, we will discuss our work regarding trust evaluation and application in social network environment in two directions, covering both local and global trust evaluation.

In Chapter 5, we discuss local trust inference in social network setting. We model the trust relationships among users in SNS as a 2-dimensional vector, in order to present the information contained in trust relationships. In the trust model, both trust and distrust (i.e. positive and negative trust) are considered. Based on the trust model, we introduce distrust semiring, which is a semiring structure for trust propagation and aggregation, and develop a trust inference framework in social network scenario. We evaluate the performance of the semiring-based trust inference method with a real-world dataset.

In Chapter 6, we consider the scenario of opinion divergence within social networks, and propose a method to reach different global trust values between groups of users with controversial opinions in the network. We model the global trust opinion formation in discrete-time dynamics and introduce bipartite consensus as the approach to establish global trust in such circumstances. We first discuss such approach under the condition of structural balanced trust network, where non-trivial global trust can be reached within the network. We then extend the results to more

general situations where eventual positivity applies. The global trust reached via our scheme can guide users' social behaviors, and support many SNS-based applications.

# CHAPTER 5: SEMIRING-BASED LOCAL TRUST INFERENCE IN SOCIAL NETWORK SCENARIO

In this chapter, we model the trust relationship between users in SNS as a 2-dimensional vector containing both trust and certainty information, and propose a semiring-based framework to combine trust evidences for inferring trust relationships in social network setting. In our approach, both trust and distrust (i.e., positive and negative trust) are considered, and opinion conflict resolution is supported by the trust inference framework. We evaluate the proposed approach on a real-world dataset. Experimental results show that our trust inference framework has high accuracy and is capable of handling ;ocal trust inference in large networks.

## 5.1 Introduction

Local trust inference in social network scenario faces several challenges:

- **Sparse social connections**, due to the scale-free property of social networks
- **Inconsistency and conflicts in trust opinions**, because of the subjective nature of local trust
- **Trust data availability**, as online users tend not to expose trust to others

explicitly

To address these challenges, there has been a line of study regarding trust inference in social network setting, which can generally be categorized in two groups. One group focuses on theoretical studies regarding trust metrics and their properties [64, 73, 81, 89, 105]. The other group proposes effective data-driven approaches for inferring trust, such as graph-theoretic models [59, 60, 83, 88, 94, 96] and machine learning approaches [106].

Graph-theoretic models generally have good performance in terms of efficiency and reasoning. However, as most work applies linear weighted averaging over trust opinions from neighbors, nonlinearity in human decision making and interaction between trust and distrust may not be fully captured. The path-based design is also sensitive to the connection sparsity in the trust network, since the trust path will not exist if some edges in between are missing. These defects restrain the accuracy of the models. On the other hand, machine learning methods can solve the data sparsity, sometimes with better label prediction accuracy. However, machine learning methods are usually slower in speed and have weak interpretation on how trust is inferred.

It is ideal if there exist a method which addresses most challenges at the same time. In this chapter, we propose a trust inference method based on a semiring model. The basic idea is to build up a nonlinear trust aggregation rule that can handle both trust and distrust information, and offer conflict resolution in the opinion combination process.

We model the trust opinion between two nodes (i.e. truster and trustee) as a 2-dimensional vector, containing both trust and certainty information. By making trust level take values in the range of $[-1, 1]$, distrust (i.e. negative trust) is considered. Based on the trust model, we propose distrust semiring and accordingly develop a semiring-based trust inference framework. We propose certainty models, and discuss several properties of the trust inference framework. In order to further improve the coverage and accuracy of our proposed trust metric, we introduce trust iteration in the framework and exploit the information from partial reciprocity.

Apart from the flexibility in modeling, this approach has the advantages of efficiency inherited from graph-theoretic models and better accuracy due to an iterative component and optimization over parameters. We evaluate our model using a real-world dataset, and show that our approach can achieve the accuracy to about 95%.

In the following sections, we first introduce semiring structure. We then give our model of 2-D trust opinion vector, and develop distrust semiring accordingly. Based on the propagation and aggregation rules defined upon semiring operations, we propose a trust inference framework with certainty models. We evaluate our approach using the Epinions dataset.

## 5.2 Semiring Structure

A semiring is an algebraic structure, consisting of a set and two binary operations, addition ($\oplus$) and multiplication ($\otimes$), with several conditions over the oper-

ations. It can be defined as follows.

**Definition.** *Semiring*: A semiring is a tuple $\langle A,\ \oplus,\ \otimes,\ \tau,\ \mathbf{1}\rangle$ such that

- $\mathbf{A}$ is a (possibly infinite) set with two special elements $\mathbf{0}, \mathbf{1} \in \mathbf{A}$
- $\oplus$, called the additive operation, is commutative and associative, with $\mathbf{0}$ as the unit element, such that

$$a \oplus b = b \oplus a$$

$$a \oplus (b \oplus c) = (a \oplus b) \oplus c$$

$$a \oplus \mathbf{0} = a = \mathbf{0} \oplus a$$

$$\forall a, b, c \in \mathbf{A}$$

- $\otimes$, the multiplicative operation, is associative, with $\mathbf{1}$ as the unit element and $\mathbf{0}$ as absorbing element, such that

$$a \otimes (b \otimes c) = (a \otimes b) \otimes c$$

$$a \otimes \mathbf{1} = a = \mathbf{1} \otimes a$$

$$a \otimes \mathbf{0} = \mathbf{0} = \mathbf{0} \otimes a$$

$$\forall a, b, c \in \mathbf{A}$$

- $\otimes$ distributes over $\oplus$, i.e.

$$a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$$

$$\forall a, b, c \in \mathbf{A}$$

A semiring $\langle A,\ \oplus,\ \otimes,\ \mathbf{0},\ \mathbf{1}\rangle$ is called an ordered semiring if there exists a partial order relation "$\preceq$" that is monotone with both operators:

$$a \preceq b \text{ and } a' \preceq b' \quad \Rightarrow \quad a \oplus a' \preceq b \oplus b' \text{ and } a \otimes a' \preceq b \otimes b'$$

Semiring is similar to ring algebra structure, but it doesn't require an additive inverse for each element (e.g. $a$ and $-a$). An example of a semiring is the set of nonnegative integers $\mathbb{N}$, with the usual addition ($+$) and multiplication ($\times$).

The operations used in a semiring structure can be seen as a generalization of addition and multiplication. Different realizations of semirings can be designed and applied in different application scenarios. With careful design, The semiring operations are possible to capture the nonlinearity in trust evaluation. Theodorakopoulos et al. [68, 107] modeled trust opinion in an Ad Hoc network as a 2-D vector of (trust, confidence), and proposed two semiring frameworks, namely path semiring and distance semiring, as trust metrics. [108] uses a semiring model similar to path semiring for multi-trust evaluation within a trust network.

However, despise its potential in modeling trust, the set $\mathbf{A}$ in most semiring models for trust evaluation only have nonnegative elements (e.g. the range of $[0, 1]$) [68]. In order to accommodate negative trust values (i.e. distrust), a modification on the semiring model is needed.

## 5.3 Trust Model

The main aim in setting up trust networks is to allow agents to form trust opinions on unknown agents or sources by asking for trust opinions from acquainted

agents. While trust has been studied for a long time, the use and modeling of distrust remains relatively unexplored. Although recent research works [72, 73] show an emerging interest in modeling the notion of distrust, models that take into account both trust and distrust are still scarce. Most approaches completely ignore distrust, or consider trust and distrust as opposite ends of the same continuous scale. However, there is a growing body of opinion that distrust cannot be seen as the equivalent of lack of trust [62, 81].

### 5.3.1 Trust Opinion Vector

As discussed in Sec. 4.1, trust is domain-specific in SNS. People hold different levels of trust towards others in different domains (contexts); it's common that people trust others in some domains instead of others. Here for simplicity, we consider the case of unified context, without considering the differences of trust opinions in multiple domains.

As pointed out in [83], in social network setting, there are two dimensions in trust. One describes the opinion of the truster on trustee's quality in providing messages of integrity, which can be used in evaluating confidence on the trustee's introducing other people to establish trust relationship (i.e. certainty). The other describes the weight that the truster puts on trustee's opinion in domain-specific decision making (i.e. trust). In order to capture both certainty and trust information, we define the trust opinion as a 2-dimensional vector.

**Definition.** *Opinion Vector*: In our trust model, trust is defined as a 2-dimensional

opinion vector $\tau_{ij} \in \mathcal{T} = [-1, 1] \times [0, 1]$ from truster $i$ towards trustee $j$:

$$\tau_{ij} = (t_{ij}, c_{ij}) \tag{5.1}$$

where $t_{ij} \in [-1, 1]$ is the trust level representing how much $i$ trusts (likes)/distrusts (dislikes) the opinions (taste) of $j$ in the current domain. $c_{ij} \in [0, 1]$ is the certainty level which shows how much $i$ believes in the integrity of $j$.

By making trust levels take values in the range of $[-1, 1]$, distrust is considered along with trust. Trust level $t_{ij} < 0$ corresponds to a distrust relation between truster and trustee, and that there is to some extent a disagreement/opposition in preference/taste. $t_{ij} = 1$ means "totally agree"or "like", while $t_{ij} = -1$ means "totally disagree"or "dislike". Certainty is orthogonal to trust value, it denotes the quality and accuracy of the trustee's opinion. $c_{ij} = 1$ shows an extreme certainty on $j$'s integrity. While a high trust value may be because of similarity in taste or preference, a high certainty value may be due to direct connection with the truster or large number of connections (i.e. high degree). Certainty determines if the opinion will be considered, and opinions with a high certainty value are more useful in making trust inference. As both trust level and certainty level about the trustee are considered in the opinion vector, more complicated situations can be modeled and analyzed.

The initial trust relationship can be established based on the information in the social network. For instance, both explicit ratings, or extraction from user interactions (e.g. 'like' and 'dislike') can be sources of directed trust opinions.

Note that though defining trust as a 2-D vector is similar to [68,107], which ap-

ply trust in Ad Hoc networks, the interpretation of trust and the way trust is applied are different. Here in a social network setting, trust is used as a measure of preference and certainty is a measure of propagation credibility, instead of a representation of identity in the Ad Hoc network case. In our model, the trust component takes values in $[-1, 1]$ where a value of $-1$ means opposite taste/preferences, while in [68] trust is in the range of $[0, 1]$ and a value of 0 is used to denote zero-trustworthiness.

### 5.3.2 Trust Network

As discussed in Sec. 4.2.2, based on the trust relationships among people in SNS, a trust network can be established, with each edge representing a directed trust opinion.

**Definition.** *Trust Network*: A trust network based on the set $\mathcal{T}$ of trust opinions is a directed and weighted graph $\mathcal{G}(V, E, t_e),\ t_e : E \mapsto \mathcal{T}$, where $V$ is the set of users. $E$ is the set of trust relationships. For $\forall e_{ij} = (v_i, v_j) \in E,\ v_i, v_j \in V$, $t_e$ associates it with an opinion vector $\tau_{ij} = (t_{ij}, c_{ij}) \in \mathcal{T}$, indicating the trust and certainty that node $v_i$ holds on $v_j$. Trust links are directed. $N_i = \{v_j | e_{ij} \in E\}$ is the neighbor set of node $v_i$.

## 5.4 Distrust Semiring

In order to tackle the issue of sparse trust relationships and opinion conflict, in our trust model, we propose a local trust metric based on a semiring that can handle both trust and distrust in opinion propagation and aggregation. Specifically,

propagation uses a multiplication operator ($\otimes$) and aggregation process is conducted via an addition operator ($\oplus$).

According to our application scenario where both trust and distrust are needed to be considered in trust evaluation, we propose a novel semiring structure, which is called *Distrust Semiring*.

**Definition.** *Distrust Semiring*: A distrust semiring is a 2-dimensional semring defined on the trust opinion set $\mathcal{T}$, such that

- $\mathbf{A} = \mathcal{T} = [-1, 1] \times [0, 1]$ the set of trust opinion vectors, with two special elements $\mathbf{0} = (0, 0), \mathbf{1} = (1, 1)$

- The additive operation $\oplus$ is defined as

$$(t_a, c_a) \oplus (t_b, c_b) = (t, c) \tag{5.2}$$

  with $c = max\{c_a, c_b\}$, and

$$t = \begin{cases} t_a & c_a > c_b \\ t_b & c_b > c_a \\ sign(t_a + t_b) \cdot max\{|t_a|, |t_b|\} & c_a = c_b \end{cases} \tag{5.3}$$

- The multiplicative operation $\otimes$ is defined as

$$(t_a, c_a) \otimes (t_b, c_b) = (t, c) \tag{5.4}$$

  where $c = c_a c_b$, and

$$t = \begin{cases} 0 & t_a < 0,\ t_b < 0 \\ t_a t_b & \text{otherwise} \end{cases} \tag{5.5}$$

The additive operation ($\oplus$) depends on the certainty level of the two opinions vectors. When the two vectors have the same certainty level, the trust level $t$ after operation equals to the trust level of the opinion that has the larger magnitude. This is an optimistic definition since trust opinions of higher magnitude will be selected in such setting. In the multiplicative operation ($\otimes$), if the two opinions both have negative trust values (which corresponds to distrust relationships), then the trust after operation goes to 0, meaning that the transitivity is cut in this case.

### 5.4.1 Trust Properties Reflected in Distrust Semiring

As its basic properties in social network setting, trust information diminishes and becomes noisy in propagation, and increases when aggregating neighborhood values [89]. Distrust semiring essentially defines an algebraic way to calculate trust, thus we first need to show whether the basic operations, addition used in trust aggregation and multiplication used in trust propagation, satisfies the commonsense.

By defining a partial ordering, we show that the semiring framework is intuitive and consistent with requirements for trust propagation and aggregation.

**Definition.** A partial order relation $\preceq$ can be defined upon two 2-D opinion vectors $\tau_a = (t_a, c_a)$ and $\tau_b = (t_b, c_b)$:

$$\tau_a \preceq \tau_b \tag{5.6}$$

if and only if

$$c_a \leq c_b, \qquad \text{or} \qquad |t_a| \leq |t_b| \ \text{ and } \ c_a = c_b \qquad \forall\ \tau_a, \tau_b \in \mathcal{T}$$

**Theorem 5.4.1.** The distrust semiring structure satisfies the following two conditions,

1. the multiplication operation is non-increasing:

$$\forall \tau_a, \tau_b \in \mathcal{T}, \quad \tau_a \otimes \tau_b \preceq \tau_a \ \wedge \ \tau_a \otimes \tau_b \preceq \tau_b \tag{5.7}$$

2. the addition operation is non-decreasing

$$\forall \tau_a, \tau_b \in \mathcal{T}, \quad \tau_a \preceq \tau_a \oplus \tau_b \ \wedge \ \tau_b \preceq \tau_a \oplus \tau_b \tag{5.8}$$

*Proof.* Let $\tau_a \otimes \tau_b = (t, c)$. As $c_a,\ c_b\ \in [0, 1]$, and $t_a,\ t_b\ \in [-1, 1]$, based on the definition of the multiplication operation, it is easy to see that $c = c_a c_b \leq c_a$, $c = c_a c_b \leq c_b$. For the trust value, if both $t_a$ and $t_b$ are negative, then $t = 0 < |t_a|, |t_b|$. Otherwise, $|t| = |t_a t_b| = |t_a||t_b| \leq |t_a|$, and $|t| = |t_a t_b| \leq |t_b|$. Thus Eq. 5.7 holds, and the non-increasing property of the multiplication operation is proved.

The non-decreasing property of trust aggregation can be shown in a similar way. In aggregation, $\tau = (t, c) = \tau_a \oplus \tau_b$. based on definition, $c = max\{c_a, c_b\} \geq c_a, c_b$. Thus the first condition is satisfied, and the non-decreasing property of the additive operation is proved. □

The non-increasing property of the multiplication operation is in accordance with the requirement for trust propagation process, whereas the non-decreasing property of the addition operation connects to the trust aggregation process. Thus the two operations in distrust semiring can be used in defining trust propagation and aggregation rules.

## 5.5 Certainty Models

In the context of social networks, though trust data may be available, the certainty information is generally implicit and contained in user interactions. Without certainty data, it is infeasible to apply the 2-D semiring model for trust inference. In practice, certainty may be derived by sentiment analysis or other natural language processing pipelines. However, NLP toolboxes are not light-weight and require rich text data in the SNS. Here we consider the situation when only SNS connections and trust data are available, and propose two ways to model certainty. One is based on length of the trust path, the other one is degree-oriented.

*Path-based certainty models*: Based on the fact that neighbors which are reachable via a longer trust path carry less valuable trust information [60], we come up with a path-based certainty model. We model the certainty value of user $v_s$ about $v_t$ as a function of hops (i.e. the length of the shortest path) between the two.

$$c_{st} = g(\text{dist}_{s,t}) = \alpha^{\text{dist}_{s,t}} \tag{5.9}$$

where $a \in (0, 1]$ is a hyperparameter and can be seen as the decay factor, and $\text{dist}_{s,t}$ represents the shortest trust path length between $v_s$ and $v_t$. When $\alpha = 1$, the decay disappears and nodes of all distances are considered equally. Such definition is equivalent to introducing a 1-hop decay of magnitude $\alpha$ at each hop. Instead of calculating $c_{st}$, we consider the decay at node $v_i$ in the middle:

$$c_i = g(v_i) = \alpha \qquad \text{(the decay factor)} \tag{5.10}$$

then along $\text{path}_{st}$ from $v_s$ to $v_t$, $c_{st} = \prod_{v_i \in \text{path}_{st}} c_i$ In this simple model, the relative

certainty between two paths is stable for $\forall a \in (0, 1)$, i.e. the trust evaluation result is $\alpha$-invariant.

*Degree-based certainty models*: Another way to model certainty in social network setting is based on node degree, with the hypothesis that nodes of higher degree are more reliable and their trust opinions have more certainty. The certainty function of a node $v_s$ can be accordingly denoted as

$$g(v_s) = g(d_s) \tag{5.11}$$

where $d_s$ is the degree of $v_s$.

We consider two realizations for $g(d_s)$, a linear model and an exponential model. The linear model can be written as

$$g(d_s) = min(\beta + \gamma d_s, 1) \tag{5.12}$$

and the exponential model can be described by

$$g(d_s) = 1 - \eta^{d_s} \tag{5.13}$$

The coefficients $\beta$, $\gamma$, and $\eta$ are tunable. Given the trust metric, an optimization problem over the parameters can be formulated accordingly to maximize its performance.

## 5.6 Trust Inference Based on Distrust Semiring

With the corresponding certainty models discussed above, a trust inference framework can be developed. Trust propagation and aggregation, as the two building blocks of the framework, can be defined using the distrust semiring structure.

### 5.6.1 Trust Propagation

Propagation of trust opinion is based on transitivity, and is defined using the $\otimes$ operator. The transitivity in trust and distrust are handled differently.

**Definition.** *Trust Propagation*: In the trust metric, the trust relationship between two nodes $v_s$ and $v_t$ with no direct connection can be estimated via the multiplicative operation $\otimes$ of trust values on edges along each path $\text{path}_{st}$ between the two nodes.

$$(t_{st}, c_{st}) = \prod_{\otimes, e_{ij} \in \text{path}_{st}} (t_{ij}, c_{ij}) \tag{5.14}$$

When there are multiple paths between the two nodes, the trust opinions reached along all paths should be aggregated together.

Considering the decay of influence along the path, a maximum hop value $\lambda$ can be introduced in order to stop early and accelerate the calculation. As shown later in Sec. 5.10, a 4-hop setting has already reached about 95% accuracy.

### 5.6.2 Trust Aggregation

The trust aggregation component in the trust metric is to combine trust information from different sources (i.e. paths). It can be defined based on the additive operator $\oplus$ in distrust semiring:

$$(t, c) = \sum_{\oplus, \{a | \text{path}_a \in P\}} (t_a, c_a) \tag{5.15}$$

where $P$ is the set of trust paths considered in aggregation, and each $(t_a, c_a)$ with $\text{path}_a \in P$ is the trust vector along that trust path $a$.

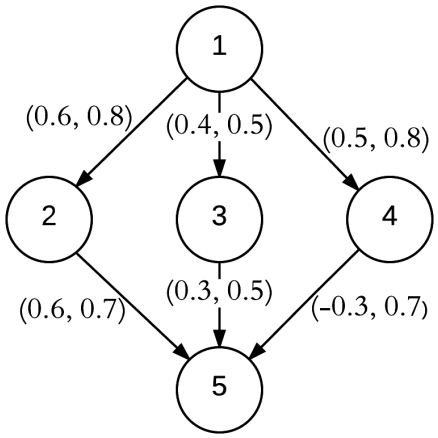



Figure 5.1: An example trust network for trust propagation and aggregation

As an example, for the trust network shown in Fig. 5.1, there are three paths from node $v_1$ to $v_5$. For the left path, when multiply $\tau_{12}$ and $\tau_{25}$ using $\otimes$, we have $\tau_1 = \tau_{12} \otimes \tau_{25} = (0.36, 0.56)$. Similarly, the middle path has $\tau_2 = (0.12, 0.25)$ and the right one $\tau_3 = (-0.15, 0.56)$. When combining $\tau_1, \tau_2, \tau_3$, we can use $\oplus$ defined in distrust semiring and have combined opinion $\tau = \tau_1 \oplus \tau_2 \oplus \tau_3 = (0.36, 0.56)$.

### 5.6.3 Overall Trust Inference Framework

A trust metric can be developed based on the trust propagation and aggregation rules defined above. Given a pair of users $v_s$ (truster) and $v_t$ (trustee), the proposed trust inference method is a function $f : V \times V \mapsto \mathcal{T}$, such that

$$(t_{st}, c_{st}) = f(v_s, v_t) = \sum_{\oplus, v_j \in N_s} (t_{sj}, c_{sj}) \otimes (t_{jt}, c_{jt}) \tag{5.16}$$

where $N_s$ is the neighbor set of $v_s$. In order to save computation resources, a threshold for trust value ($\sigma_t$) and certainty value ($\sigma_c$) can be introduced. When

below the thresholds, the trust opinion will not be considered in aggregation, i.e.

$$(t_{st}, c_{st}) = \sum_{\oplus, v_j \in N_s, (\sigma_t, \sigma_c) \preceq \tau_{sj} \otimes \tau_{jt}} (t_{sj}, c_{sj}) \otimes (t_{jt}, c_{jt}) \tag{5.17}$$

The way that the trust metric is applied to trust inference is shown in Alg. 4. The algorithm is to evaluate the trust opinion $(t_{st}, c_{st})$ of $v_s$ (truster) about $v_t$ (trustee), with $\lambda$ the maximum hop number, $\sigma_t$ and $\sigma_c$ the lower bounds for trust and certainty value respectively. One can interpret the algorithm as follows: $v_s$ asks her neighbors for their trust opinions about $v_t$. Each neighbor $v_i \in N_s$ provides her opinion about $v_t$ (i.e. $t_{it}$ and $c_{it}$), either directly or estimated using the trust inference algorithm recursively. At each hop forward, $\lambda$, the maximum hop number, will decrease by 1 until reaching 0. Then $v_s$ aggregates all the evidence and reach $(t_{st}, c_{st})$ about $v_t$.

In such a trust inference framework, both trust and distrust (i.e. negative trust) are taken into consideration for trust inference. As paths above the thresholds are all integrated into the calculation, trust information are fully exploited in this approach for better coverage. When aggregating trust opinions along different propagation paths, we apply *First Aggregate Then Propagate* (FATP) scheme. At each node, the trust information along different paths is first aggregated together and then propagation to the node as a unified trust opinion. Such 1-hop propagation before aggregation scheme can reduce the noise during trust propagation process and improve the inference quality. The possible conflicts among opinions are handled in a non-trivial way with the introduction of certainty value and nonlinear addition operation ($\oplus$).

**Algorithm 4** Semiring-based trust inference algorithm, $f(v_s, v_t, \lambda, \sigma_t, \sigma_c)$

Mark $v_s$ as `visited`

**if** $\lambda = 0$ **then**

return $(0, 0)$

**end if**

$c_s \leftarrow g(v_s)$

**if** $t_{st}$ exists **then**

return $(t_{st}, c_s)$

**end if**

$t_{st} \leftarrow 0$

$c_{st} \leftarrow 0$

**for** each $v_i \in N_s$, the neighbor set of node $v_s$ **do**

**if** ($v_i$ `visited`) or ($|t_{si}| < \sigma_t$) or ($c_s < \sigma_c$) **then**

continue

**end if**

$(t_{it}, c_{it}) \leftarrow f(v_i, v_t, \lambda - 1, \frac{\sigma_t}{|t_{si}|}, \frac{\sigma_c}{c_s})$

**if** ($t_{si} < 0$ and $t_{it} < 0$) or ($t_{it} = 0$) **then**

continue

**end if**

$(t_{st}, c_{st}) \leftarrow (t_{st}, c_{st}) \oplus ((t_{si}, c_s) \otimes (t_{it}, c_{it}))$

**end for**

return

## 5.7 Iterative Trust Evaluation

Alg. 4 is built on the trust paths between two users. In real-world data, the trust information is often sparse, thus the coverage of trust inference method is limited by the amount of trust data present. In order to address the sparsity issue, we propose to introduce the collective method used in network analysis. Iterative method is a type of collective approaches in classification problem [109]. Such method is used when the data contains interconnection and correlation between objects, such as webpages and SNS, and has been shown to be very effective over network data [110]. The method treats the independent inference as a joint inference problem, and uses an iterative approach to predict labels; the new labels predicted in the previous iterations are used in the following iterations.

In the trust inference problem, as trust relationships among users are intercorrelated, we introduce iterative trust evaluation in our trust inference framework (Alg. 5) to improve the performance. The basic idea of the design is to iteratively evaluate the trust opinions associated with the edges in the test dataset, conditioned on the both ground truth and current predictions. Initially, the knowledge base is the training dataset, and the result set is empty. While running, the error in Alg. 5 is used to measure the difference of results between two iterations. The iteration ends when convergence on local predictions is reached or maximum number of iterations has finished.

**Algorithm 5** Iterative trust evaluation

Let $E$: set of edges to evaluate, $K$: max number of iterations

`knowledge_base` $\leftarrow$ trust relationships in training dataset

`result_set` $\leftarrow \emptyset$

**for** $i : 1$ to $K$ **do**

  error $\leftarrow 0$

  **for** edge $e_j \in$ **do**

    `knowledge_base` $\leftarrow$ `knowledge_base` $\setminus (t_j, c_j)$

    calculate trust metric $f(e_j)$

    **if** $((t_j, c_j) \notin$ `result_set`$)$ **then**

      error $\leftarrow$ error $+ 1$

    **else**

      error $\leftarrow$ error $+ \|f(e_j) - (t_j, c_j)\|$

      `result_set` $\leftarrow$ `result_set` $\setminus (t_j, c_j)$

    **end if**

    $(t_j, c_j) \leftarrow f(e_j)$

    `result_set` $\leftarrow$ `result_set` $\cup (t_j, c_j)$

    `knowledge_base` $\leftarrow$ `knowledge_base` $\cup (t_j, c_j)$

  **end for**

  **if** (error $< \epsilon$) **then**

    return `result_set`

  **end if**

**end for**

return `result_set`

## 5.8 Exploiting Reciprocity in Trust

As discussed in Sec. 4.2.4, (partial) reciprocity can be added into the trust inference algorithm to improve both coverage and accuracy. We extend the trust inference algorithm (Alg. 4) by introducing partial reciprocity in a careful way. Apart from considering trust evidences from neighbors, the truster also treat the direct trust opinion about herself from the trustee as a source of information, as a reflection of partial reciprocity. In order to reduce the error due to asymmetry in trust relationships, we only consider positive reciprocity (i.e. the reciprocity in positive trust relationships), and the certainty value in a trust opinion reached using reciprocity has a smaller magnitude compared the one reached through transitivity-based propagation.

## 5.9 Optimistic vs Pessimistic Semirings

In Sec. 5.6.3, the trust metric takes an optimistic definition for trust aggregation; the binary operation $\oplus$ takes the larger magnitude of the two and its sign as the combined trust value when the certainty levels are the same. This definition is based on the hypothesis that extreme opinions weigh more than neutral ones [60,94]. In a pessimistic model, when $c_a = c_b$, the addition operation for trust aggregation (Eq. 5.3) can be modified as

$$t = sign(t_a + t_b) \cdot min\{|t_a|, |t_b|\} \tag{5.18}$$

Such definition is "pessimistic", as the aggregated trust value conservatively

Table 5.1: The Epinions trust network statistics

| Statistics | Value |
|---|---|
| User size | 91053 |
| Edge amount | 841372 |
| Trust relationships | 717667 |
| Distrust relationships | 123705 |

picks the smaller magnitude between the two incoming trust values. With such model, trust aggregation is not non-decreasing any more.

When the addition operation is defined as the normal "add" $(+)$ on real set $\mathbb{R}$, the trust metric evolves to a PageRank-like definition with 1-step trust propagation.

## 5.10 Performance Evaluation

In order to verify the model and evaluate the performance of our semiring-based approach, we conduct experiments on the Epinions trust network dataset [111]. The dataset contains both direct trust connections (with edge weight of $+1$) and distrust ones (with edge weight of $-1$), and form the trust network $\mathcal{G}(V, E, t_e)$ where $t_e : E \mapsto \mathcal{T} = \{-1, 1\}$. To the best of our knowledge, it is the largest dataset available online that contains both explicit trust and distrust information marked in a binary ($+1$ and $-1$) format. The dataset has 91.053 users with 841,372 edges, 717,667 trust relationships and 123,705 distrust relationships.

Some statistics about the dataset is shown in Table 5.1.

In order to evaluate our trust metric, we first verify the transitivity and reciprocity of trust relationships using the Epinions dataset.

### 5.10.1 Transitivity and Reciprocity in The Data

Inspired by previous works, we investigate the transitivity phenomenon. For transitivity, we count the number of triangles represented by triplets $(i, j, k)$ such that

$$v_i, v_j, v_k \in V,\ e_{ij}, e_{jk}, e_{ik} \in E \tag{5.19}$$

$$\text{and} \qquad t_{ij}t_{jk}t_{ik} > 0 \qquad \text{(structural balance)} \tag{5.20}$$

Based on this definition, among all triangles that satisfy Eq. 5.19, 92.7% (10229847 out of 11033232) are transitive. We also zoom in and look at the case when $v_i$ distrusts $v_j$, $v_j$ distrusts $v_k$ (i.e. $t_{ij} < 0$ and $t_{jk} < 0$), what the relationship between $v_i$ and $v_k$ is. It turns out that $\approx 50\%$ situations have $t_{ik} < 0$ with another half having $t_{ik} > 0$, which endorses the setting "the enemy of your enemy is actually unknown" in our trust metric.

We also evaluate the reciprocity of trust relationships in Epinions dataset. We consider reciprocity of trust relationships as symmetry on signs, i.e. we consider pairs $(v_i, v_j)$ that have reciprocal trust relationships when both $e_{ij}$ and $e_{ji}$ exists and $t_{ij}t_{ji} > 0$.

Based on the experimental results, we notice that (partial) reciprocity commonly exists between nodes in the network. Among all 259,751 pairs of nodes having bi-directional relationships, 254,345 ($\approx 98\%$) are reciprocal relationships (both directions are of the same sign, either positive or negative), and 98% of reciprocal ones are of positive connections, which corresponds to the concept of partial reciprocity that we discussed in Sec. 4.2.4.

However, as both trust and distrust relationships are single-valued, the symmetry on magnitude of trust relationships is unable to be evaluated.

### 5.10.2 Experimental Design

In the Epinions dataset, all trust values in the training dataset are in the set $\mathcal{T} = \{-1, 1\}$. Based on the definition of our trust metric model, though certainty value of each predicted edge varies in $[0, 1]$, the set for predicted trust values will be $\mathcal{T}_p = \mathcal{T} \cup \{0\} = \{-1, 0, 1\}$, where the value of $t_{st} = 0$ represents the case when no enough information available for predicting the trust relationship associated with the edge $e_{st}$. Unlike linear approach such as [81], no rounding is needed for predicting discrete trust values.

To evaluate the performance of our trust evaluation approach, we measure and compare *accuracy* and *coverage*. Accuracy is the fraction of correctly predicted trust relationships among all nonzero ones in the test data:

$$\text{accuracy} = \frac{\|\{e_{ij} \in S_{\text{test}} \mid t_{ij} = t'_{ij}\}\|}{\|\{e_{ij} \in S_{\text{test}} \mid t_{ij} \neq 0\}\|} \tag{5.21}$$

where $S_{\text{test}}$ is the test edge set, $t_{ij}$ is the trust value reached using our trust inference algorithm, and $t'_{ij}$ is the ground truth trust value. The denominator in Eq. 5.21 only take edges that have nonzero predictions.

Coverage is defined as the number of edges predicted over the number of edges predictable in test data:

$$\text{coverage} = \frac{\|\{e_{ij} \in S_{\text{test}} \mid t_{ij} \neq 0\}\|}{\|\{e_{ij} \in S_{\text{test}} \mid \exists k, l \in V \text{ s.t. } e_{ik}, e_{lj} \in S_{\text{train}}\}\|} \tag{5.22}$$

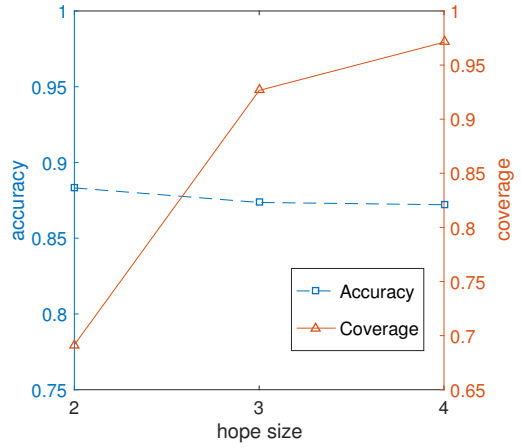
where an edge is predictable if both of its vertices are present in the training set.

We partition the Epinions dataset to generate training and test data. As the dataset is a network with vector edges, partition edges in different ways may have non-trivial impact on the graph connectivity. Our data partitioning uses two schemes, one randomly drop a certain percentage $\rho$ of edges. Instead of dropping edges randomly, another scheme uses the timestamp in the dataset. The dataset generated this way is referred as the time-ordered dataset. We conduct experiments on datasets of two sizes, and refer the test dataset generated with $\rho = 0.5\%$ as the small test dataset, while the one using $\rho = 5\%$ as the large test dataset. We use the trust metric developed above for trust evaluation on the two datasets respectively.

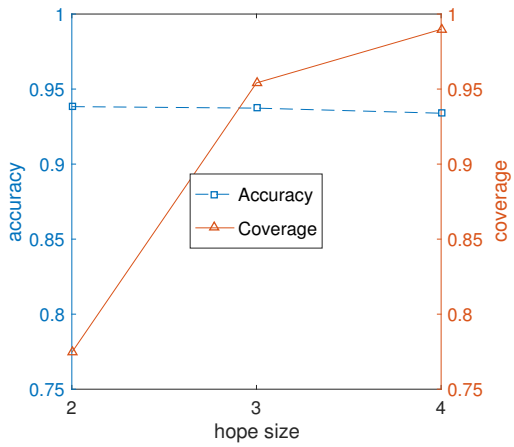
### 5.10.3 Experimental Results

*Training/test data ratio*: The experimental results for the large test dataset and the small one are shown in Fig. 5.2(a) and Fig. 5.2(b) respectively. From the results, we see that the performance of our approach is better on the smaller dataset, in terms of both accuracy and coverage. The major reason is that the training data size is larger for the small test dataset case, which means the trust network used for prediction is more connected and has more trust evidence for prediction. Thus, the performance of our approach is positively correlated with the training/test data ratio.

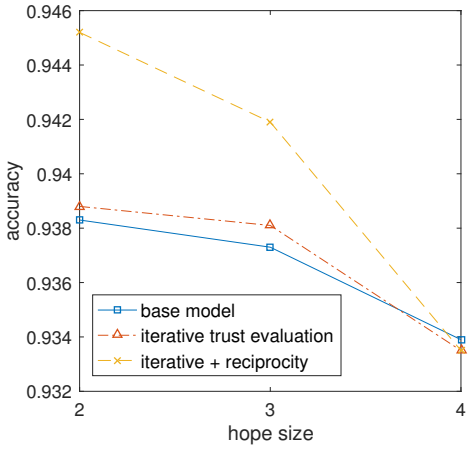
*Varying trust path length $\lambda$*: As discussed in previous literature [59, 83], the longer the trust path in graphical trust evaluation models, the more noise may be introduced. While the introduction of distant friends improves the coverage of the



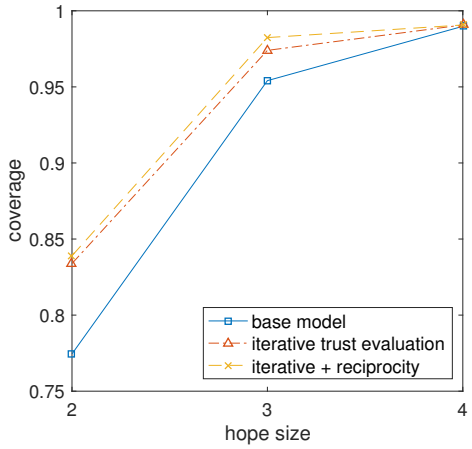

(a) Large test dataset




(b) Small test dataset




(c) Accuracy




(d) Coverage

Figure 5.2: Evaluation results of the STAR trust inference framework

social recommendation algorithm, it affects its accuracy. As the average shortest path length in Epinions dataset is about 4, in the experiment, we set the maximum hop length $\lambda$ as 2, 3 and 4 respectively, and evaluate the performance of our trust metric under different settings.

In Fig. 5.2(a) and Fig. 5.2(b) we can see that, the coverage of our approach is better with paths of more hops considered, as more nodes are reachable and used

in trust evaluation. However, though the variation is subtler compared to coverage change. The accuracy result is still decreasing with an increasing maximum hop length parameter.

*Iterative trust evaluation and the additional partial reciprocity information*: Apart from data size and maximum hop length, we also evaluate the impact of these two addition components. We compare our base model for trust inference (Alg. 4) versus the model applying iterative trust evaluation and exploiting the additional partial reciprocity information. As shown in Fig. 5.2(c) and (d), introducing an iterative trust evaluation method improves the performance in both coverage and accuracy. By getting trust information based on partial reciprocity for trust evaluation, the coverage and accuracy of the trust inference algorithm are even higher. By applying collective methods and partial reciprocity, more trust evidence can be used in the trust metric.

When applying iterative approaches in classification problem [109], the ordering of value updates in the iterative trust evaluation may affect the predictive accuracy and convergence rate. Here, in order to investigate the influence of ordering in trust iteration over test data, we randomize the order of the node pairs for prediction, and compare the experimental results. We list the results for the 4-hop case in Table 5.2. According to the results, the application of iterative approach for the trust metric in social recommender system setting is fairly robust to randomized orderings of the test dataset.

*Parameter tuning in certainty*: As discussed in Sec. 5.5, we proposed two tunable models for certainty based on degree. Here we vary the values of the parameters

Table 5.2: Performance of iterative trust evaluation using random-ordered test data

| random set | Accuracy | Coverage |
|---|---|---|
| 1 | 0.9314 | 0.9895 |
| 2 | 0.9314 | 0.9899 |
| 3 | 0.9328 | 0.9895 |
| 4 | 0.9314 | 0.9899 |
| 5 | 0.9319 | 0.9895 |

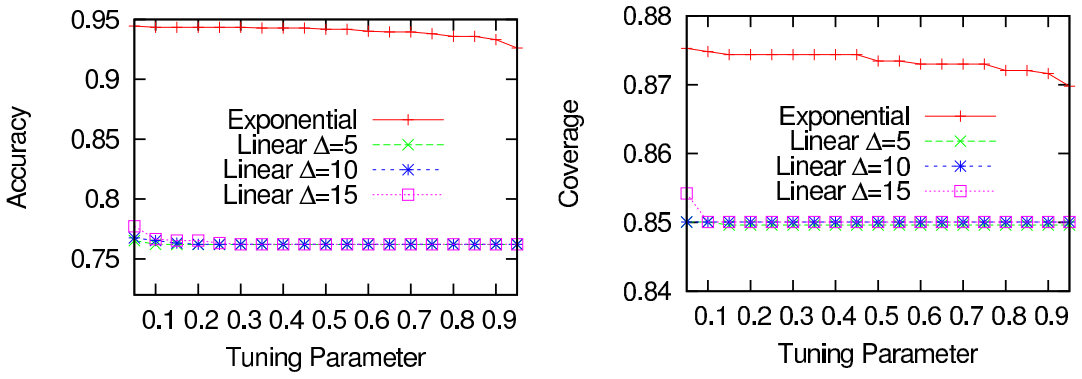
and investigate the influence of parameter tuning/ optimization towards model performance.

In the linear model (Eq. 5.12), there are two parameters, $\beta$ and $\gamma$. $\beta$ is in the range of $[0, 1]$, and $\gamma$ is dependent on $\beta$ and the cut-off degree (i.e. maximum degree). The exponential model described in Eq. 5.13 has one parameter $\eta \in [0, 1]$. In the experiment, we fix maximum degree value used in linear model to be 5, 10 and 15, and let $\beta$ and $\eta$ both vary from 0 to 1 with step change to be 0.05.

Results are shown in Fig. 3, where $\Delta$ represents maximum degree value. From the plots we can see that the linear model leads to lower accuracy and coverage, while the exponential model doesn't change very much with varying parameter values. Using exponential model for certainty with a relatively small $\eta$ around 0.2 would be a good setting for trust inference.

### 5.10.4 Comparison with Other Approaches

The performance of our semiring-based trust inference method can reach as small as 5.8% error rate and as good as 98.3% coverage rate. As a fair comparison, the graph-theoretic linear approach based on matrix operations [81] has an optimal




(a) Accuracy variation

(b) Coverage variation

Figure 5.3: Performance with parameter change

prediction error rate of 6.4%. The machine learning approach introduced in [106] can reach an accuracy about 0.934 (i.e. $\approx$ 6.7% error rate). For [86] which used probabilistic confidence models for trust inference, it achieved an accuracy of 89% using Epinions dataset. From comparison, we can see an improvement on accuracy can be obtained using our semiring-based approach.

### 5.10.5 Discussion

Because the trust (and distrust) relationships in the dataset are binary ($+1$ and $-1$), a lot of nuance and variance in the trust relationships are lost. This prevents us from conducting some evaluations. For example, since the trust values are not continuous in $[-1, 1]$, the deviation of evaluated results from ground-truth values (e.g. RMSE) is difficult to measure and analyzed.

On the other hand, the high accuracy of our approach on this dataset shows its flexibility and power in trust evaluation for social recommendation.

## 5.11 Summary and Future Direction

In this chapter, we discuss local trust inference in social network setting. We model the trust relationships in SNS as 2-dimensional vectors consisting of both trust and certainty information. Trust and distrust (i.e. negative trust) are considered in our model. Accordingly, we propose a novel semiring structure, distrust semiring, for trust propagation and aggregation over the trust network, where transitivity of trust and distrust are handled differently. Based on the distrust semiring structure and certainty models, we develop a trust inference framework. In order to validate the model and evaluate the performance of our approach, we conduct a series of experiments using the Epinions dataset. The experimental results show that the semiring-based trust inference approach has advantages in both accuracy and coverage.

As pointed out in [81], there are more than one possible types of transitivity within the trust network. In our future work, we will consider other transitivity models in our framework of trust inference and explore its influence over the performance. As discussed in Sec. 5.9, an optimistic definition is used for trust aggregation. We will evaluate other possible definitions of the operation in a semiring structure and compare with our current settings. As trust is domain specific and people may have different trust ratings for the same user over different contexts, we will also investigate how trust network forms and trust inference operates in the multi-domain case.

# CHAPTER 6: GLOBAL TRUST EVALUATION WITH CONFLICTING OPINIONS

In this chapter, we consider reaching global trust (reputation) of nodes in a social network under the scenario of opinion divergence. In such a general setting, the evaluation of global trust suffers from the curse of opinion divergence in the network, as simple aggregation of controversial opinions can hardly offer meaningful insights about users' reputation. We propose a method to reach different global trust values between groups of users with controversial opinions in the network. We model the global trust opinion formation in discrete-time dynamics and introduce bipartite consensus as the approach to establish global trust in such circumstances. Such approach works upon the property of structural balance, and can be extended to more general situations where eventual positivity applies. Via our approach, non-trivial global trust can be reached within the network, which can guide users' social behaviors, and support many SNS-based applications.

## 6.1 Introduction

In social networks, both traditional ones and SNS, trust is seen as a critical social feature and plays an important role in social interactions between nodes in

the network. It provides a guideline for nodes to interact and make decisions [62], and is fundamental in many SNS-based applications, e.g. social recommender and personalized services [60, 83, 112].

As discussed in Sec. 4.3, within social networks, the concept of trust has two levels, namely local and global. Local trust refers to the subject trust opinions that each user holds about her neighbors in the network. Global trust, also called as reputation, is a combination of local trust opinions gathered from users within the network, and represents the public perspective about the target entity's identity. The value of global trust is considered as a measure of the credibility and homophily within the population. As a more objective measure compared to local trust, global trust is very important in network security, decision making processes as well as improving the quality of SNS-based applications.

There has been significant work on trust metrics and inference for local trust evaluation, for example [72, 81, 83, 88, 89, 107]. However, the research on evaluation of global trust is limited and is still at its early stage [113]. A classic approach is via aggregation of local trust opinions of users in the network, e.g. the ratings of users on Ebay. This scheme works when users in the network are homophily and malicious users do not exist [114]. When considering a more general situation where users may have distrust relationships due to controversial opinions or existence of malicious users, the aggregation of nodes' local trust opinions for global trust would suffer from the curse of opinion divergence in the network. Simple average over population with controversial opinions can hardly offer meaningful insights about true reputation. The case of two-party political system would be a good example.

It's a community with two sub-communities that have similar opinions within the group but controversial ones between groups. The true global trust (reputation) of each node cannot be reached by simple combination of opinions from both groups.

Recent research shows the limitation of global trust evaluation when considering distrust (i.e. negative trust). Kamvar et al. proposed to use EigenTrust [67], a PageRank-like algorithm to evaluate global trust of peers in P2P network. The issue of malicious nodes is addressed by computing trust value on other peers and majority voting. Li and Wang [113] proposed a subjective probability based approach for global trust evaluation in service-oriented computing (SOC). However, distrust is not well-modeled in both approaches, and the clustering of controversial opinions within the network cannot be captured. This renders both approaches ineffective for global trust evaluation in SNS scenarios. In [114], DeFigueiredo et al. discussed trust between users in online interactions, and they concluded that applying single universal trust ratings (i.e. global trust) is vulnerable to manipulation by malicious users.

The problem of consensus in signed networks has attracted increasing interest in research community [115–118]. Shi et al. [115] studied opinion dynamics over signed social graphs, where phase transition phenomena is discussed. In [116–118], Altafini et al. discussed bipartite consensus in networks with antagonistic interactions (signed networks), where consensus can be reached separately in each antagonistic group.

In this chapter, we propose to solve the problem of global trust evaluation in social network setting. We consider a general case where both trust and distrust

relationships exist within the social network. The trust network established on the trust and distrust relationships is modeled as a directed, signed and weighted graph. The direction of an edge in the graph represents a directed trust relationship, with the signed weight implying the level of trust (distrust) associated with the edge.

We formulate the problem of global trust evaluation in social network setting as an bipartite consensus problem in the directed and signed trust network. We first consider reaching bipartite consensus for global trust in networks that are structurally balanced; we further extend the results to a more general setting with the notion of eventual positivity. Based on such global trust evaluation approach, user reputation and grouping information are both available for decision making in social network environment and can be used to boost social-aware applications.

The rest of this chapter is arranged as follows. In Sec. 6.3, we discuss some preliminaries. We then propose our bipartite consensus formulation for global trust evaluation in social network setting in a discrete-time dynamical system, considering the existence of distrust relationships (Sec. 6.3). In Sec. 6.4, we consider reaching bipartite consensus for global trust under the condition of structural balance, and further extend the results to a more general case. We discuss the integration of global trust in SNS-based applications in Sec. 6.5. We conclude our work in Sec. 6.6 and highlight the future research directions.

## 6.2 Preliminaries

In the following of this chapter, all vectors are column vectors and denoted by lower case letters, and matrices are denoted by upper case letters. For a square matrix $M$, $M^T$ denotes its transpose and $M^k$ denotes its $k$-th power, and the $ij$-entry of $M$ is denoted as $M_{ij}$.

### 6.2.1 Graph Theory

In this work, we utilize the concept of weighted signed graphs.

**Definition.** A *weighted signed graph* $\mathcal{G}$ can be denoted by a triple $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$, where $\mathcal{V}$ is a finite set of vertices (nodes), $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ is the set of edges, and $\mathcal{A} \in \mathbb{R}^{n \times n}$ is the adjacency matrix of the signed weights of the edges in graph $\mathcal{G}$. For $\mathcal{A}$: $\mathcal{A}_{ij} \neq 0 \Leftrightarrow (v_i, v_j) \in \mathcal{E}$.

For simplicity, we use $\mathcal{G}(\mathcal{A})$ to represent a weighted signed graph with adjacency matrix $\mathcal{A}$. We call a graph undirected if the order of the nodes is irrelevant in representing edges, and the matrix is symmetric $\mathcal{A} = \mathcal{A}^T$. For a directed graph (i.e. digraph), the edge $(v_i, v_j) \in \mathcal{E}$ is directed, where $v_i$ is the tail and $v_j$ is the head of the edge. In a digraph, the pair of edges between the same nodes is called a digon. A digraph is digon sign-symmetric if $A_{ij}A_{ji} \geq 0$ for all $i, j \in \{1, \cdots, n\}, i \neq j$.

Fig. 6.1 is a weighted signed digraph with $\mathcal{V} = \{v_1, v_2, v_3\}$, $\mathcal{E} = \{(v_1, v_2), (v_1, v_3), (v_2, v_1), (v_2, v_3), (v_3, v_2)\}$, and

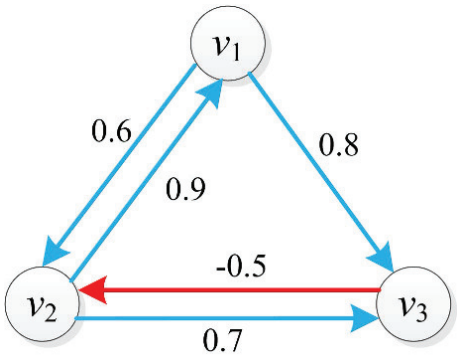



Figure 6.1: An example of weighted signed digraph

$$\mathcal{A} = \begin{bmatrix} 0 & 0.6 & 0.8 \\ 0.9 & 0 & 0.7 \\ 0 & -0.5 & 0 \end{bmatrix}$$

A (directed) path of length $k$ is a sequence of distinct nodes, $\{v_1, v_2 \cdots v_{k+1}\}$, such that $\{v_m, v_{m+1}\} \in \mathcal{E}$, $m \in \{1, \cdots, k\}$. A (directed) cycle is a path beginning and ending with the same node. We say that the graph is strongly connected if for any $v_i, v_j \in \mathcal{V}$, there exists a path $\{v_i, \cdots, v_j\}$ in $\mathcal{G}$.

We say matrix $\mathcal{A} \in \mathbb{R}^{n\times n}$ is *nonnegative* ($\mathcal{A} \geq 0$) by meaning $\mathcal{A}_{ij} \geq 0$ for $\forall i, j \in \{1, \cdots, n\}$, and $\mathcal{A} \neq 0$; we say $\mathcal{A}$ is positive ($\mathcal{A} > 0$) when $\mathcal{A}_{ij} > 0$ for $\forall\ i, j \in \{1, \cdots, n\}$.

$sp(\mathcal{A}) = \{\lambda_1(\mathcal{A}), \cdots, \lambda_n(\mathcal{A})\}$ denotes the spectrum of $\mathcal{A}$, where $\lambda_i(\mathcal{A}), i \in \{1, \cdots, n\}$ are the eigenvalues of matrix $\mathcal{A}$. The spectral radius $\rho(\mathcal{A})$ of $\mathcal{A}$ is the smallest real positive number such that $\rho(\mathcal{A}) \geq |\lambda_i(\mathcal{A})|, \forall\ i \in \{1, \cdots, n\}$.

### 6.2.2 Structural Balance

Based on [119], the structural balance property of a signed network can be defined as follows.

**Definition.** A signed network is *structurally balanced* if it admits a bipartition of $\mathcal{V}$ into $\mathcal{V}_1, \mathcal{V}_2$, where $\mathcal{V}_1 \cup \mathcal{V}_2 = \mathcal{V}$, and $\mathcal{V}_1 \cap \mathcal{V}_2 = 0$, such that $\mathcal{A}_{ij} \geq 0$, for $v_i, v_j \in \mathcal{V}_m$, and $\mathcal{A}_{ij} \leq 0$, for $v_i \in \mathcal{V}_m$ and $v_j \in \mathcal{V}_n$, $m \neq n$, $m, n \in \{1, 2\}$.

The sign of a cycle is the product of all edges' signs; a (directed) cycle is positive if it contains even number of negative edges, and is negative otherwise. It can be shown that a signed graph is structurally balanced if and only if all the cycles in the graph are positive [116,119]. As an extension, a signed digraph is structurally balanced when all directed cycles are positive [116].

### 6.2.3 Perron-Frobenius Property

Following [120], we give a definition of the Perron-Frobenius property:

**Definition.** A real square matrix $\mathcal{A} \in \mathbb{R}^{n \times n}$ is said to have the *Perron-Frobenius*

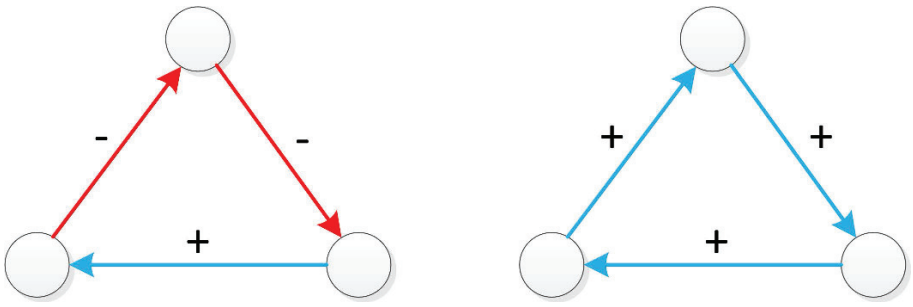



Figure 6.2: Structural balance

*property* if $\rho(\mathcal{A})$ is an eigenvalue of $\mathcal{A}$ (i.e. Perron-Frobenius eigenvalue), with corresponding nonnegative left and right eigenvectors. When $\mathcal{A}$ nonnegative and regular,

- $\rho(\mathcal{A})$ is a simple positive eigenvalue,
- The eigenvector corresponding to $\rho(\mathcal{A})$ can be chosen to be positive (called a Perron vector),
- No other eigenvalue has the same modulus, i.e. for any other $\lambda$ an eigenvector of $\mathcal{A}$, $\lambda < \rho(\mathcal{A})$.

We use $PF_n$ to denote the set of matrices in $\mathbb{R}^{n\times n}$ satisfying the Perron-Frobenius property.

**Definition.** A real square matrix $\mathcal{A} \in \mathbb{R}^{n\times n}$ is *eventually positive* if $\exists\ k_0 \in \mathbb{N}_+$ such that $\mathcal{A}^k > 0\ ,\ \forall\ k \geq k_0,\ k \in \mathbb{N}_+$.

The smallest $k_0$ in Definition 6.2.3 is called the power index of $\mathcal{A}$. We follow the notation in [117, 121] and denote eventually positive matrices as $\mathcal{A} \overset{\vee}{>} 0$.

Although Perron-Frobenius property is defined on nonnegative square matrices, it is show in [117, 122] that eventually positive matrices, as well as their trans-

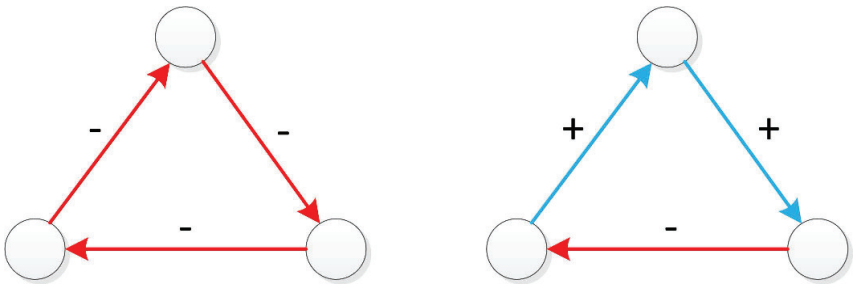



Figure 6.3: Structural unbalance

pose, also have the Perron-Frobenius property. As discussed in [117], the Perron-Frobenius property can be used for adjacency matrices with eventual positivity to evaluate the formation of unanimous opinions, by introducing the concept of holdability [123]:

**Definition.** *Holdability*: for a discrete-time dynamical system with $x_0 \in \mathbb{R}^n$ as the initial state, a set $S \subset \mathbb{R}^n$ is *holdable* if for $\forall\ x_0 \in \mathbb{R}^n,\ \lim_{k\to\infty} dist(x(k), S) = 0$, where $dist(x(k), S) = \inf_{y\in S}\|x(t) - y\|$ with a certain norm in $\mathbb{R}^n$, and $\exists k_o \in \mathbb{N}_+$ such that $x(k) \in S$, for $\forall k \geq k_0$.

## 6.3 Problem Formulation

Based on the trust relationships between users in the social network, there is an associated trust network. Referring to the discussion in Sec. 4.2.2, we define the trust network used in our formulation as follows:

**Definition.** *Trust Network*: A trust network in SNS can be represented as a directed weighted signed graph $\mathcal{G}_T(\mathcal{V}, \mathcal{E}, \mathcal{A})$ established via social interactions, where $\mathcal{V}$ is the set of nodes (i.e. users), $\mathcal{E}$ is the set of directed edges (i.e. trust links), and $\mathcal{A}$ the matrix of signed edge weight. $e_{ij} = (v_i, v_j) \in \mathcal{E}, v_i, v_j \in \mathcal{V}$, is a directed trust link from node $v_i$ to $v_j$, and its value (weight) is an entry $\mathcal{A}_{ij}$ of $\mathcal{A}$. $N_i = \{v_j | e_{ij} \in \mathcal{E}\}$ is the neighbor set of node $v_i$.

*Remark*: Here $\mathcal{A}_{ij} \in [-1, 1]$ indicates the extent of trust that $v_i$ has on $v_j$. $\mathcal{A}_{ij} = 1$ means $v_i$ "totally agrees with"or "likes"$v_j$, while $\mathcal{A}_{ij} = -1$ means $v_i$ "totally disagrees with"or "dislikes"$v_j$. Weights are not necessarily symmetric.

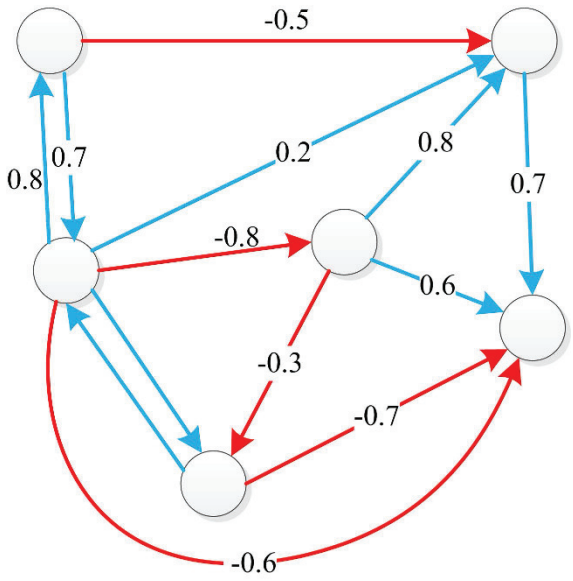



Figure 6.4: An example of signed trust network

### 6.3.1 Distributed Scheme for Global Trust Evaluation

Global trust evaluation in SNS is typically very challenging. Due to lack of a central authority, different agents may provide different or even contradictory information in the evaluation process, and a unanimous global trust opinion is difficult to compute. In order to tackle the challenge and obtain meaningful results for global trust in SNS settings, we consider reaching opposite global trust opinions between groups of controversial opinions. We introduce the idea of bipartite consensus in signed networks and formulate our problem of global trust evaluation in SNS as a bipartite consensus problem over the signed trust network. In this formulation, opinion divergence between two antagonistic groups is well-handled.

Consider a strongly connected trust network $\mathcal{G}_T(\mathcal{V}, \mathcal{E}, \mathcal{A})$. We assume that the bipartite consensus for global trust is evaluated via distributed opinion update

scheme on $\mathcal{G}_T$ through a discrete-time linear dynamic system:

$$x(k+1) = W(k)x(k) \tag{6.1}$$

where vector $x(k) = (x_1(k), \cdots, x_n(k))$ is the temporary trust opinion held by nodes in $\mathcal{V}$ at time $k$. The update matrix $W(k)$ can be decomposed into 2 parts:

$$W(k) = \Sigma(k) + F(k) \tag{6.2}$$

where matrix $F(k)$ at time $k$ is the off-diagonal matrix used in integrating opinions from neighbors, and $\Sigma(k)$ is a diagonal one describing the influence that users put on themselves. A starting point for $F(k)$ would be a static matrix in accordance with $\mathcal{A}$ of the trust graph $\mathcal{G}_T(\mathcal{A})$, and $\Sigma(k)$ can be arranged correspondingly such that users put a trust value of 1 on themselves. Note that in order to reach consensus on global trust, $W(k) = W$ is normalized in the system (6.1) as follows:

$$w_{ij} = \frac{\mathcal{A}_{ij}}{1 + \sum_{j \in adj(i)} |\mathcal{A}_{ij}|} \tag{6.3}$$

such that $\sum_{j \in \mathcal{V}} |w_{ij}| = 1$ for any $i \in \{1, \cdots, n\}$.

The monotonicity of (6.1) can be defined on a partial orthant order [124]. A partial orthant order in $\mathbb{R}^n$ is a vector:

$$\sigma = [\sigma_1, \cdots \sigma_n]^T, \qquad \sigma_i \in \{1, -1\}, \ \forall \ i \in \{1, \cdots, n\} \tag{6.4}$$

where $\sigma_i = 1$ denotes the natural order, and $\sigma_i = -1$ denotes the opposite. Corresponding to $\sigma$, we define the matrix $D_\sigma = D_\sigma^T = diag(\sigma) \in \mathbb{R}^{n \times n}$ as the gauge transformation matrix [116, 118]. $D_\sigma$ can be used to define an orthant $K_\sigma = \{x \in \mathbb{R}^n | D_\sigma x \geq 0\}$. The partial order $\sigma$ can be indicated by "$\leq_\sigma$":

$$x_1 \leq_\sigma x_2 \Leftrightarrow x_1 - x_2 \in K_\sigma \tag{6.5}$$

which can be used to change the orthant order in $\mathbb{R}^n$.

$$D_\sigma = \mathbf{1} = [1, \cdots, 1]^T \in \mathbb{R}^n \tag{6.6}$$

The set of all gauge transformation matrices is $\mathcal{D} = \{D = diag(\sigma), \sigma = [\sigma_1, \cdots \sigma_n]^T, \sigma_i \in 1, -1\} \subset \mathbb{R}^{n \times n}$.

The discrete-time system (6.1) is called monotone w.r.t $\sigma$ if for any initial conditions $x_1(0), x_2(0)$ s.t. $x_1(0) \leq_\sigma x_2(0)$,

$$x_1(t) \leq_\sigma x_2(t) \qquad \forall t > 0 \tag{6.7}$$

The monotonicity of the system (6.1) can be verified using the off-diagonal matrix $F$; the system is monotone w.r.t $\sigma$ if and only if:

$$\sigma_i \sigma_j F_{ij} \geq 0 \qquad \forall i, j \in 1, \cdots, n, i \neq j \tag{6.8}$$

A gauge transformation matrix can be applied to the adjacency matrix of a trust graph to change the sign of the edges as well as opinions held by nodes in the graph. As an example, Fig. 6.5 shows applying gauge transformation to a signed trust digraph. For the trust network on the left, its adjacency matrix has negative weighted entries $\mathcal{A}_{21}$ and $\mathcal{A}_{13}$. After a gauge transformation with $D = diag(-1, 1, 1)$, all the entries in the gauge transformed matrix $\mathcal{A}' = D\mathcal{A}D$ are positive. The system is monotone w.r.t $\sigma = [-1, 1, 1]$. Accordingly, the opinion held by $v_1$ changes from $O_1$ to $O_1' = -O_1$.

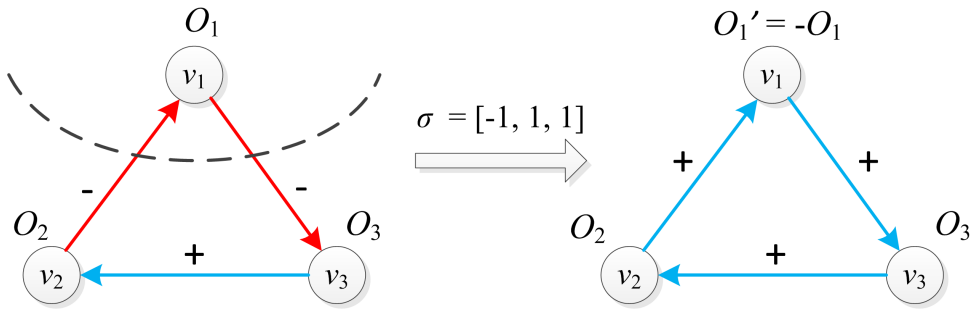



Figure 6.5: Gauge transformation

## 6.4 Main Results

In order to evaluate a node's global trust value within a trust network of controversial opinions, we formulate it as a bipartite consensus problem with the goal of calculating the node's global trust information. We start from the case where the trust network is of structural balance, and extend the result to a trust network of approximate structural balance in the next section.

*Remark*: By bipartite consensus for global trust, we mean that the global trust of users have the following convergence result:

$$\lim_{k\to\infty} |x_i(k)| = c > 0 \qquad \forall i \in \{1, \cdots, n\} \tag{6.9}$$

Additionally, if the final state is

$$c = \frac{1}{n}|\omega^T x(0)| \tag{6.10}$$

for some constant weight vector $\omega$, then we say that all the agents in the network reach bipartite consensus.

### 6.4.1 Global Trust Evaluation in Structurally Balanced Networks

A structurally balanced signed social network, as discussed in Sec. 6.2, can be partitioned into two disjoint antagonistic groups, where each group contains only friends, while any two individuals from different groups are adversaries. The dynamics of opinion forming in structurally balanced communities obeys monotonicity.

Given the trust network $\mathcal{G}_T(\mathcal{A})$, which is a digraph, We have the following lemma for structural balance.

**Lemma 6.4.1.** For the trust network $\mathcal{G}_T(\mathcal{A})$ that is strongly connected and digon sign-symmetric, it is structurally balanced if and only if either of the following holds:

1. All directed cycles of $\mathcal{G}_T(\mathcal{A})$ are positive;

2. $\exists$ a gauge transformation matrix $D \in \mathcal{D}$, such that the adjacency matrix $\mathcal{A}' = D\mathcal{A}D$ is nonnegative;

*Proof*: 1) This comes from the definition of structurally balanced network as discussed in Sec. 6.2.

2) From Definition 6.2.2, for the node set $\mathcal{V}$ of graph $\mathcal{G}_T$ which is structurally balanced, it can be partitioned into $\mathcal{V}_1$ and $\mathcal{V}_2$ such that all and only the negative edges connect nodes in $\mathcal{V}_1$ and $\mathcal{V}_2$. If we choose a partial orthant order $\sigma = [\sigma_1, \cdots, \sigma_n]$, where $\sigma_i = 1$ when $v_i \in \mathcal{V}_1$ and $\sigma_i = -1$ if $v_i \in \mathcal{V}_2$, then through the gauge transform matrix $D_\sigma = diag(\sigma)$, the adjacency matrix $\mathcal{A}$ would satisfy that $\mathcal{A}' = D_\sigma \mathcal{A} D_\sigma$ is nonnegative. For the sufficient condition, it can be proved via contradiction. Suppose there doesn't exist such orthant order $\sigma$ and corresponding

matrix $D_\sigma = diag(\sigma)$ such that $D\mathcal{A}D$ is nonnegative, then for any bipartition of $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$, from the proof of the necessary condition, $\mathcal{V}_1$ and $\mathcal{V}_2$ can not satisfy the condition that all and only negative edges connect the two set, which is equivalent to structural unbalance. ■

Based on Lemma 6.4.1, we have the following theorem:

**Theorem 6.4.2.** For the strongly connected trust network $\mathcal{G}_T(\mathcal{A})$ that is directed and digon sign-symmetric, the discrete-time system in (6.1) can reach a bipartite consensus on global trust, if $\mathcal{G}_T(\mathcal{A})$ is structurally balanced.

*Proof*: From Lemma 6.4.1, we know that for $\mathcal{G}_T(\mathcal{A})$ that is strongly connected and digon sign-symmetric, there exists a gauge transformation matrix $D = diag(\sigma)$, such that $\mathcal{A}' = D\mathcal{A}D \geq 0$. If in system (6.1), we choose the off-diagonal matrix $F(k) = F = C\mathcal{A}$, and the diagonal matrix $\Sigma(k) = \Sigma = C\mathbb{I} = C$, where $C = diag(c_1, \cdots, c_n)$ is the normalization matrix with $c_i = 1/(1 + \sum_{j \in adj(i)} |\mathcal{A}_{ij}|) \geq 0, \forall i \in \{1, \cdots, n\}$, then for the same $D$, we have:

$$W' = DWD = D(C + C\mathcal{A})D = C + C\mathcal{A}' \geq 0 \tag{6.11}$$

The solution $y^* = D_t x^*$ would be the result of a usual consensus problem over a strongly connected unsigned graph, where the unsigned graph is the trust network $\mathcal{G}_\mathcal{T}(\mathcal{A})$ after the gauge transformation on $\mathcal{A}$. After reaching the consensus $y^* = \lim_{k\to\infty} y(k) = \mathbf{c} = [c, \cdots, c] \in \mathbb{R}^n$, the bipartite consensus on the original graph $\mathcal{G}_\mathcal{T}(\mathcal{A})$ can be evaluated using

$$x^* = D^{-1}y^* = Dy^* \tag{6.12}$$

which is the bipartite global trust reached by the users in the trust network with controversial opinions.

Based on the bipartite consensus result,the node set $\mathcal{V}$ can be partitioned into $\mathcal{V}_1 = \{v_i \in \mathcal{V} | x_i^* = c, i \in \{1, \cdots, n\}\}$ and $\mathcal{V}_2 = \{v_j \in \mathcal{V} | x_j^* = -c, j \in \{1, \cdots, n\}\}$, which is the corresponding clustering due to structural balance. ■

*Remark*: As mentioned above, the bipartite consensus solution is in the form of Eq. (6.9).

**Corollary 1.** As the consensus result, $\lim_{k\to\infty} x(k) = \nu^T D x(0) D\mathbf{1}$, where $D$ is the gauge transformation s.t. $D\mathcal{A}D$ nonnegative, and $\nu$ is the normalized nonzero left eigenvector of $W = DWD$ s.t. $\nu^T\mathbf{1} = 1$.

*Proof*: From Theorem 6.4.2, $y^*$, a standard consensus can be reached on the gauge transformed graph $\mathcal{G}(D\mathcal{A}D)$. According to [125], $y^* = \lim_{k\to\infty} y(k) = \nu^T y(0)\mathbf{1}$, $\nu$ is the normalized nonzero left eigenvector of $DLD$ s.t. $\nu^T\mathbf{1} = 1$. Thus $x^* = D^{-1}y^* = Dy^* = (\nu^T Dx(0))D\mathbf{1}$. ■

Similarly, when $\mathcal{G}_T(\mathcal{A})$ is weight balanced, the consensus result would be $\lim_{k\to\infty} x(k) = (1/n)\mathbf{1}^T Dx(0)D\mathbf{1}$.

### 6.4.2 Extension Based on Eventual Positivity

In real cases, structural balance can rarely be satisfied. By combining eventual positivity with the gauge transformation used in Sec. 6.4.1, the approach can be extended to cases where structural balance property is not satisfied.

**Definition.** Matrix $\mathcal{A} \in \mathbb{R}^{n\times n}$ has the *signed Perron-Frobenius property* [117] if the following are satisfied:

1. Spectral radius $\rho(\mathcal{A})$ is a real positive eigenvalue of $\mathcal{A}$
2. $\rho(\mathcal{A}) > \lambda,\ \forall\lambda \in sp(\mathcal{A}), \lambda \neq \rho(\mathcal{A})$
3. All elements in $v_r$, the right eigenvector corresponding to $\rho(\mathcal{A})$, are nonzero, i.e. $v_{r,i} \neq 0, \forall i \in \{1, \cdots, n\}$

We denote the set of matrices of signed Perron-Frobenius property as $SPF_n$, and have the following proposition:

**Proposition 1.** For the matrix $\mathcal{A} \in \mathbb{R}^{n\times n}$ of a directed trust graph $\mathcal{G}_T(\mathcal{A})$, $\exists$ an orthant order $\sigma$ and $D = diag(\sigma)$ s.t. $D\mathcal{A}D \overset{\vee}{>} 0$, iff $\mathcal{A} \in SPF_n,\ \mathcal{A}^T \in SPF_n$, and the left eigenvector $v_l$ and the right eigenvector $v_r$ of $\mathcal{A}$ satisfies:

1. $v_{l,i}v_{r,i} > 0,\ \forall i \in \{1, \cdots, n\}$, or
2. $v_{l,i}v_{r,i} < 0,\ \forall i \in \{1, \cdots, n\}$

*Proof*: If there exists a gauge transformation such that $\mathcal{A}' = D\mathcal{A}D \overset{\vee}{>} 0$, then from Sec. 6.2.3 and [117, 122], we know that $\mathcal{A} \in PF_n, \mathcal{A}^T \in PF_n$, and therefore $\rho(\mathcal{A}') > 0$ and strict larger than all other eigenvalues. As $sp(\mathcal{A}) = sp(\mathcal{A}'), sp(\mathcal{A}^T) = sp(\mathcal{A}')$ according to [116], thus $\rho(\mathcal{A}) = \rho(\mathcal{A}')$ is positive and such that $\rho(\mathcal{A}) > \lambda,\ \forall\lambda \in sp(\mathcal{A}), \lambda \neq \rho(\mathcal{A})$. From Definition 6.2.3, we can find positive $v_r'$ and $v_l'$ the right and left eigenvectors (Perron vectors), such that:

$$\begin{aligned} D\mathcal{A}Dv_r' = \mathcal{A}'v_r' = \rho(\mathcal{A}')(v_r') \\ (v_l')^T D\mathcal{A}D = (v_l')^T\mathcal{A}' = \rho(\mathcal{A}')(v_l')^T \end{aligned} \tag{6.13}$$

as $D^2 = \mathbb{I}$ and $\rho(A) = \rho(\mathcal{A}')$,

$$\begin{aligned} \mathcal{A}v_r &= \rho(\mathcal{A})v_r, \\ v_l^T\mathcal{A} &= \rho(\mathcal{A})v_l^T \end{aligned} \tag{6.14}$$

where $v_l = Dv_l'$, and $v_r = Dv_r'$.

Obviously both $v_l$ and $v_r$ have no elements of 0. Thus from Definition 6.4.2 about signed Perron-Frobenius property, we know that $\mathcal{A} \in SPF_n$, $\mathcal{A}^T \in SPF_n$. Note that $v_{l,i}v_{r,i} = (D_{ii})^2 v_{l,i}'v_{r,i}' = v_{l,i}'v_{r,i}'$. If we choose both $v_l'$ and $v_r'$ to be positive, the condition (1) is satisfied. Similarly, if choose one of the two vectors to be negative (multiply it by $-1$), and the other to be positive, then the second condition can be satisfied.

The sufficient condition can be proved in a similar way. ■

**Lemma 6.4.3.** Consider the system (6.1), where the normalized weight matrix $W = \Sigma + F$ with diagonal matrix $\Sigma = diag(c_1, \cdots, c_n)$ and off-diagonal matrix $F$. If $\exists\ d \geq 0$ s.t. $F + D \overset{\vee}{>} 0$ with $D = \Sigma - d\mathbb{I}$, then system (6.1) holds to $\mathbb{R}_{\{-,+\}}$.

*Proof*: Let $B = F + D$, then $W$ can be written as:

$$W = \Sigma + F = d\mathbb{I} + D + F = d\mathbb{I} + B \tag{6.15}$$

Since $B = F + D \overset{\vee}{>} 0$, then from Proposition 1 and [122], $B, B^T \in SPF_n$, and $\rho(B)$ is a positive real eigenvalue of $B$ that strictly larger than other eigenvalues. Let $v_l$ and $v_r$ be left and right eigenvectors of $B$, then $d > 0$ implies that $W$ must have $d + \rho(B) \in \mathbb{R}_+$ as the largest eigenvalue, and $v_l$ and $v_r$ as left and right eigenvectors,

as

$$v_l^T W = v_l^T d + v_l^T B = (d + \rho(B))v_l^T$$
$$W v_r = d v_r + B v_r = (d + \rho(B))v_r \tag{6.16}$$

Therefore, $W, W^T \in PF_n$ and $W \overset{\vee}{>} 0$, and we have:

$$x^* = \lim_{k\to\infty} x(k) = \frac{v_l^T x_0 v_r}{v_l^T v_r} \tag{6.17}$$

If $v_l^T x_0 > 0$ then $x^* \in int(\mathbb{R}^n_+) \cup \emptyset$, similarly $v_l^T x_0 < 0$ leads to $x^* \in int(\mathbb{R}^n_-) \cup \emptyset$. From Definition 6.2.3 about holdability, it can be shown that the system (6.1) holds to $\mathbb{R}^n_{\{-,+\}}$. ■

Based on Proposition 1 and Lemma 6.4.3, we have the following theorem for a relaxed bipartite consensus on global trust within a trust network of eventual positivity.

**Theorem 6.4.4.** For the strongly connected trust network $\mathcal{G}_T(\mathcal{A})$ that is described by system (6.1), if there exists $d \geq 0$ such that proposition 1 holds for $F + D$, where $D = \Sigma - d\mathbb{I}$, then the system (6.1) holds to the orthant pair $\mathbb{R}^n_{\{-\sigma,+\sigma\}}$.

*Proof*: For the system (6.1) that describes the trust dynamics of $\mathcal{G}_T(\mathcal{A})$, we have the normalized weight matrix $W = \Sigma + F$ with diagonal matrix $\Sigma = C\mathbb{I} = C$ and off-diagonal $F = C\mathcal{A}$, where $C = diag(c_1, \cdots, c_n)$ is the normalization matrix with $c_i = 1/(1 + \sum_{j\in adj(i)} |A_{ij}|) \geq 0, \forall i \in \{1, \cdots, n\}$.

Let $B = F + D$, then $W = d\mathbb{I} + B$, $B, B^T \in SPF_n$, with $v_l$ and $v_r$ the left and right eigenvectors of identical signs. From Proposition 1, we know that there exists an orthant order $\sigma$ and a corresponding gauge transformation $D = diag(\sigma) \in \mathcal{D}$ s.t.

$B' = D\mathcal{B}D \overset{\vee}{>} 0$. Let $y = Dx$, then the system w.r.t. $y$ is:

$$y(k+1) = D(d\mathbb{I} + B)Dy(k) = (d\mathbb{I} + B')y(k) \tag{6.18}$$

Thus from Theorem 6.4.3, the system (6.18) holds to orthant pair $\mathbb{R}^n_{\{-,+\}}$. Since $x = D^{-1}y = Dy$, we know that the system (6.1) holds to $\mathbb{R}^n_{\{-\sigma,+\sigma\}}$. ■

As shown above, via relaxing bipartite consensus for global trust in SNS as bipartite opinions holdable in two opposite orthants, structural balance is no longer required, instead it only requires eventual positivity after gauge transformation. Such extension makes our global trust evaluation approach available to more general case.

## 6.5 Application of Global Trust in Social Network Environment

### 6.5.1 Clustering Effect for System Security

The bipartite consensus of global trust comes naturally with clusters of controversial opinions within the trust network. When the opinion differences come from different tastes, the clusters represents two communities of opposite preferences. However, it is also possible that one of the clusters is formed due to the identity of adversary, in which case the global trust evaluation process plays the role of clustering adversaries within social network based on the distrust relationship between users. The level of system security can be improved by implementing global trust evaluation for adversary detection in the network.

### 6.5.2 Distrust Filtering in Recommender System

Along with the popularity of recommender systems, there are various types of attacks towards the system, e.g. random attack and bandwagon attack [126].

In order to enforce the integrity of the recommendation, the global trust information can be integrated in the system to filter out users of low reputation [112] . If a user's global trust value is lower than the threshold (e.g. negative), her rating will not be considered in rating prediction.

When applying global trust into recommender system, the classic user-based collaborative filtering (CF) [127] can be modified as:

$$\tilde{r_{ik}} = b_{ik} + \frac{\sum_{u_j \in S(k;i), w_j \geq \eta} s_{ij} \cdot (r_{jk} - b_{jk})}{\sum_{u_j \in S(k;i)} s_{ij}} \tag{6.19}$$

where $S(k;i)$ is the neighbor set of $u_i$ about item $o_k$, with $s_{ij}$ the similarity between $u_i$ and $u_j$. $w_j$ is the reputation of $u_j$ and $\eta$ is the threshold. $b_{ik}$ and $b_{jk}$ are the baseline estimates for $r_{ik}$ and $r_{jk}$ respectively, and $r_{jk}$ is the rating of $u_j$ about $o_k$. This means that only people of global trust values above threshold (e.g. positive) are considered as a source of reference for item recommendation. Note that here because of bipartite global trust, users of the same cluster in global trust evaluation will mark each other with positive global trust values, and the opposite if the users are from two groups of controversial opinions.

Apart from these mentioned above, there are more scenarios in social network setting that can apply users' global trust information to improve service quality. For example, when seeding advertisements, users' global trust (reputation) can be

interpreted as as a measure of quality and influence in information diffusion, and can be considered as an additional constraint in SNS advertisement allocation [128].

## 6.6 Summary and Future Direction

In this chapter, we investigate the problem of global trust evaluation in SNS with controversial opinions. We consider both trust and distrust relationships in the associated trust network, and propose to reach different global trust between antagonistic groups. We introduce the approach of bipartite consensus in signed graphs and formulate the problem of global trust evaluation in SNS as bipartite consensus for global trust with controversial opinions. We use a discrete-time dynamical system to describe the distributed evaluation process. Under the condition of structural balance, we prove that the dynamic system considered in our formulation can reach a bipartite consensus for global trust. In order to further extend the result of bipartite consensus for global trust to a more general case, the concept of eventual positivity is introduced and the definition of bipartite consensus is accordingly adjusted to be holdable cones. Finally we discuss the application of global trust reached via our approach for system security in SNS and recommendation integrity in social recommender systems.

In the future we will consider time-varying adjacency matrix in the dynamical system and explore its influence on reaching bipartite consensus for global trust. We will also discuss the robustness of the system. Meanwhile, we are interested in structural balance approximation, which would connect the ideal case of structural

balance and general cases in solving the problem of global trust evaluation in SNS. We will apply global trust in more SNS-based scenarios.

# Bibliography


[1] Facebook. Facebook Key Facts. `https://newsroom.fb.com/key-Facts`. [Online; accessed 16-Feb-2014].

[2] Sharad Goel, Jake Hofman, and M. Irmak Sirer. Who does what on the web: Studying web browsing behavior at scale. In *Proceedings of the 6th International Conference on Weblogs and Social Media (ICWSM 2012)*, 2012.

[3] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42. ACM, 2007.

[4] Emilio Ferrara and Giacomo Fiumara. Topological features of online social networks. *arXiv preprint arXiv:1202.0331*, 2012.

[5] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[6] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969.

[7] Lun Li, David Alderson, John C Doyle, and Walter Willinger. Towards a theory of scale-free graphs: Definition, properties, and implications. *Internet Mathematics*, 2(4):431–523, 2005.

[8] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

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

[10] 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.

[11] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66. ACM, 2001.

[12] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[13] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.

[14] Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[15] Ove Frank and David Strauss. Markov graphs. *Journal of the american Statistical association*, 81(395):832–842, 1986.

[16] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440–442, 1998.

[17] Matthew O Jackson. An overview of social networks and economic applications. *The handbook of social economics*, pages 511–585, 2010.

[18] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[19] James W Anderson. *Hyperbolic geometry*. springer, 2005.

[20] Luther Pfahler Eisenhart. *Riemannian geometry*. Princeton university press, 1997.

[21] James W Anderson. *Hyperbolic geometry*. springer, 2006.

[22] Robert Kleinberg. Geographic routing using hyperbolic space. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1902–1909. IEEE, 2007.

[23] F. Papadopoulos, C. Psomas, and D. Krioukov. Network mapping by replaying hyperbolic growth. *Networking, IEEE/ACM Transactions on*, PP(99):1–1, 2014.

[24] A. Cvetkovski and M. Crovella. Hyperbolic embedding and routing for dynamic graphs. In *INFOCOM 2009, IEEE*, pages 1647–1655, 2009.

[25] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010.

[26] Wei Chen, Wenjie Fang, Guangda Hu, and Michael W Mahoney. On the hyperbolicity of small-world and treelike random graphs. *Internet Mathematics*, 9(4):434–491, 2013.

[27] F. de Montgolfier, M. Soto, and L. Viennot. Treewidth and hyperbolicity of the internet. In *Network Computing and Applications (NCA), 2011 10th IEEE International Symposium on*, pages 25–32, Aug 2011.

[28] Onuttom Narayan and Iraj Saniee. Large-scale curvature of networks. *Physical Review E*, 84(6):066108, 2011.

[29] Jussi Väisälä. Gromov hyperbolic spaces. *Expositiones Mathematicae*, 23(3):187–231, 2005.

[30] Marián Boguñá, Fragkiskos Papadopoulos, and Dmitri Krioukov. Sustaining the internet with hyperbolic mapping. *Nature Communications*, 1:62, 2010.

[31] Fragkiskos Papadopoulos, Constantinos Psomas, and Dmitri Krioukov. Replaying the geometric growth of complex networks and application to the as internet. *ACM SIGMETRICS Performance Evaluation Review*, 40(3):104–106, 2012.

[32] Marian Boguna, Dmitri Krioukov, and Kimberly C Claffy. Navigability of complex networks. *Nature Physics*, 5(1):74–80, 2008.

[33] Eleni Stai, John S. Baras, and Symeon Papavassiliou. A class of backpressure algorithms for networks embedded in hyperbolic space with controllable delay-throughput trade-off. In *Proceedings of the 15th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '12, pages 15–22, New York, NY, USA, 2012. ACM.

[34] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22, 2007.

[35] Eytan Bakshy, Dean Eckles, Rong Yan, and Itamar Rosenn. Social influence in social advertising: evidence from field experiments. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 146–161. ACM, 2012.

[36] Facebook. One million thank yous. `http://newsroom.fb.com/News/639/One-Million-Thank-Yous`, June 2013. [Online; accessed 22-Feb-2013].

[37] Robert Johnson. Scaling Facebook to 500 Million Users and Beyond. `https://www.facebook.com/note.php?note_id=409881258919/`, July 2010. [Online; accessed 16-Dec-2013].

[38] Facebook. Campaign cost & budgeting. `https://www.facebook.com/help/www/318171828273417`, 2014. [Online; accessed 16-Feb-2014].

[39] Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.

[40] Nikhil R Devenur and Thomas P Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 71–78. ACM, 2009.

[41] Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web*, pages 519–528. ACM, 2012.

[42] Facebook. Advertise on facebook. `https://www.facebook.com/ads/create/`, 2014. [Online; accessed 02-Mar-2014].

[43] Michael S Bernstein, Eytan Bakshy, Moira Burke, and Brian Karrer. Quantifying the invisible audience in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 21–30. ACM, 2013.

[44] Facebook. News feed fyi: A window into news feed. `https://www.facebook.com/business/news/News-Feed-FYI-A-Window-Into-News-Feed`, 2013. [Online; accessed 16-Oct-2014].

[45] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5, 2007.

[46] Sharad Goel, Duncan J. Watts, and Daniel G. Goldstein. The structure of online diffusion networks. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 623–638. ACM, 2012.

[47] Salesforce. The facebook ads benchmark report. `http://www.salesforcemarketingcloud.com/wp-content/uploads/2013/06/The-Facebook-Ads-Benchmark-Report.pdf`, 2013. [Online; accessed 02-Mar-2014].

[48] Robert Hof. You Know What's Cool? 1 Million Advertisers On Facebook. `http://www.forbes.com/sites/roberthof/2013/06/18/you-know-whats-cool-1-million-advertisers-on-facebook/`, June 2013. [Online; accessed 16-Dec-2013].

[49] Chinmay Karande, Aranyak Mehta, and Ramakrishnan Srikant. Optimizing budget constrained spend in search advertising. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 697–706. ACM, 2013.

[50] Eric K Clemons. The complex problem of monetizing virtual electronic social networks. *Decision Support Systems*, 48(1):46–56, 2009.

[51] P Alex Dow, Lada A Adamic, and Adrien Friggeri. The anatomy of large facebook cascades. In *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013*, 2013.

[52] Yahoo! Webscope A1: Yahoo! Search Marketing Advertiser Bidding Data. `http://webscope.sandbox.yahoo.com/`, 2005.

[53] Shuai Yuan and Jun Wang. Sequential selection of correlated ads by POMDPs. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 515–524. ACM, 2012.

[54] J. Leskovec. Stanford network analysis package (snap). `http://snap.stanford.edu/`. 2013.

[55] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '99, pages 251–262. ACM, 1999.

[56] Michael Braun and Wendy W Moe. Online advertising response models: Incorporating multiple creatives and impression histories. *Available at SSRN 1896486*, 2012.

[57] Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*, 2011.

[58] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[59] Paolo Massa and Paolo Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 17–24. ACM, 2007.

[60] Jennifer Golbeck. *Generating predictive movie recommendations from trust in social networks*. Springer, 2006.

[61] Ido Guy, Naama Zwerdling, David Carmel, Inbal Ronen, Erel Uziel, Sivan Yogev, and Shila Ofek-Koifman. Personalized recommendation of social software items based on social relations. In *Proceedings of the third ACM conference on Recommender systems*, pages 53–60. ACM, 2009.

[62] Günter Gans, Matthias Jarke, Stefanie Kethers, and Gerhard Lakemeyer. Modeling the impact of trust and distrust in agent networks. In *Proc. of AOIS01*, pages 45–58, 2001.

[63] Merriam Webster. Merriam-webster online dictionary. 2006.

[64] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2):618–644, 2007.

[65] Ueli Maurer. Modelling a public-key infrastructure. In *Computer SecurityE-SORICS 96*, pages 325–350. Springer, 1996.

[66] Philip R Zimmermann and Philip R Zimmermann. *The official PGP user's guide*, volume 265. MIT press Cambridge, 1995.

[67] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651. ACM, 2003.

[68] George Theodorakopoulos and John S Baras. Trust evaluation in ad-hoc networks. In *Proceedings of the 3rd ACM workshop on Wireless security*, pages 1–10. ACM, 2004.

[69] Jin-Hee Cho, Ananthram Swami, and Ray Chen. A survey on trust management for mobile ad hoc networks. *Communications Surveys & Tutorials, IEEE*, 13(4):562–583, 2011.

[70] Michael Kinateder, Ernesto Baschny, and Kurt Rothermel. Towards a generic trust model–comparison of various trust update algorithms. In *Trust Management*, pages 177–192. Springer, 2005.

[71] Cai-Nicolas Ziegler and Georg Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7(4-5):337–358, 2005.

[72] Jennifer Golbeck. *Computing with social trust*. Springer, 2008.

[73] Patricia Victor, Chris Cornelis, and Martine De Cock. *Trust networks for recommender systems*, volume 4. Springer, 2011.

[74] Catholijn M Jonker and Jan Treur. Formal analysis of models for the dynamics of trust based on experiences. In *Multi-Agent System Engineering*, pages 221–231. Springer, 1999.

[75] Jordi Sabater i Mir and Carles Sierra García. *Trust and reputation for agent societies*. PhD thesis, 2004.

[76] A. Jøsang and R. Ismail. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*, Bled, Slovenia, 2002.

[77] B. Yu and M.P. Singh. An evidental model of distributed reputation management. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 294–301, Bologna, Italy, 2002. ACM Press.

[78] Tao Jiang and John S Baras. Graph algebraic interpretation of trust establishment in autonomic networks. *Preprint Wiley Journal of Networks*, 2009.

[79] Alfarez Abdul-Rahman and Stephen Hailes. Supporting trust in virtual communities. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 9–pp. IEEE, 2000.

[80] Rana Forsati, Iman Barjasteh, Farzan Masrour, Abdol-Hossein Esfahanian, and Hayder Radha. Pushtrust: An efficient recommendation algorithm by leveraging trust and distrust relations. In *ACM RecSys*, 2015.

[81] Ramanthan Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*, pages 403–412. ACM, 2004.

[82] Jiliang Tang, Xia Hu, and Huan Liu. Social recommendation: a review. *Social Network Analysis and Mining*, 3(4):1113–1133, 2013.

[83] Jennifer Ann Golbeck. Computing and applying trust in web-based social networks. 2005.

[84] Frank Edward Walter, Stefano Battiston, and Frank Schweitzer. A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems*, 16(1):57–74, 2008.

[85] Charif Haydar, Azim Roussanaly, and Anne Boyer. Local trust versus global trust networks in subjective logic. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 29–36. IEEE, 2013.

[86] Thomas DuBois, Jennifer Golbeck, and Aravind Srinivasan. Predicting trust and distrust in social networks. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 418–424. IEEE, 2011.

[87] Christiano Castelfranchi and Rino Falcone. *Trust theory: A socio-cognitive and computational model*, volume 18. John Wiley & Sons, 2010.

[88] Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. Trust management for the semantic web. In *The Semantic Web-ISWC 2003*, pages 351–368. Springer, 2003.

[89] Audun Jøsang, Stephen Marsh, and Simon Pope. Exploring different types of trust propagation. In *Trust management*, pages 179–192. Springer, 2006.

[90] Peixin Gao, John S Baras, and Jennifer Golbeck. Semiring-based trust evaluation for information fusion in social network services. In *Information Fusion (Fusion), 2015 18th International Conference on*, pages 590–596. IEEE, 2015.

[91] Jinshan Liu and Valérie Issarny. Enhanced reputation mechanism for mobile ad hoc networks. In *Trust management*, pages 48–62. Springer, 2004.

[92] Paolo Massa and Paolo Avesani. Controversial users demand local trust metrics: an experimental study on epinions.com communicty. In *Proceedings of 25th AAAI Conference*, 2005.

[93] Peixin Gao, Zhixin Liu, and John S Baras. Bipartite consensus for global trust in social network services. In *Global Telecommunications Conference (GLOBECOM 2015), 2015 IEEE*. IEEE, 2015.

[94] Jennifer Golbeck. Personalizing applications through integration of inferred trust values in semantic web-based social networks. In *Semantic Network Analysis Workshop at the 4th International Semantic Web Conference*, volume 16, page 30, 2005.

[95] Jennifer Golbeck and Ugur Kuter. The ripple effect: change in trust and its impact over a social network. In *Computing with Social Trust*, pages 169–181. Springer, 2009.

[96] Paolo Avesani, Paolo Massa, and Roberto Tiella. A trust-enhanced recommender system application: Moleskiing. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 1589–1593. ACM, 2005.

[97] Paolo Massa and Paolo Avesani. Trust metrics on controversial users: Balancing between tyranny of the majority. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3(1):39–64, 2007.

[98] Thomas DuBois, Jennifer Golbeck, and Aravind Srinivasan. Rigorous probabilistic trust-inference with applications to clustering. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 655–658. IET, 2009.

[99] Ugur Kuter and Jennifer Golbeck. Using probabilistic confidence models for trust inference in web-based social networks. *ACM Transactions on Internet Technology (TOIT)*, 10(2):8, 2010.

[100] Bert Huang, Angelika Kimmig, Lise Getoor, and Jennifer Golbeck. A flexible framework for probabilistic models of social trust. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 265–273. Springer, 2013.

[101] D. Kreps and R. Wilson. Reputation and imperfect information. *J. Economic Theory*, 27, 1982.

[102] B. Holmstrom. Managerial incentive problems: A dynamic perspective. *Rev. Economic Studies*, 66(1), 1999.

[103] Lik Mui, Mojdeh Mohtsahemi, Cheewee Ang, Peter Szolovits, and Ari Halberstadt. Ratings in distributed systems: A bayesian approach. In *Proceedings of the 11th Workshop on Information Thchnologies and Systems*, New Orleans, Louisiana, December 2001.

[104] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[105] Reid Andersen, Christian Borgs, Jennifer Chayes, Uriel Feige, Abraham Flaxman, Adam Kalai, Vahab Mirrokni, and Moshe Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In *Proceedings of the 17th international conference on World Wide Web*, pages 199–208. ACM, 2008.

[106] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650. ACM, 2010.

[107] George Theodorakopoulos and John S Baras. On trust models and trust evaluation metrics for ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 24(2):318–328, 2006.

[108] Stefano Bistarelli, Simon N Foley, Barry O'Sullivan, and Francesco Santini. Semiring-based frameworks for trust propagation in small-world networks and coalition formation criteria. *Security and Communication Networks*, 3(6):595–610, 2010.

[109] Ben London and Lise Getoor. Collective classification of network data. *Data Classification: Algorithms and Applications*, 399, 2014.

[110] Jennifer Neville and David Jensen. Iterative classification in relational data. In *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, pages 13–20, 2000.

[111] Epinions trust network dataset – KONECT, October 2014.

[112] Peixin Gao, John S Baras, and Jennifer Golbeck. Trust-aware social recommender system design. In *Doctor Consortium of 2015 International Conference on Information Systems Security and Privacy*, pages 19–28. INSTICC, 2015.

[113] Lei Li and Yan Wang. A subjective probability based deductive approach to global trust evaluation in composite services. In *Web Services (ICWS), 2011 IEEE International Conference on*, pages 604–611. IEEE, 2011.

[114] Dimitri do B DeFigueiredo, Earl T Barr, and Shyhtsun Felix Wu. Trust is in the eye of the beholder. In *CSE (3)*, pages 100–108, 2009.

[115] Guodong Shi, Alexandre Proutiere, Mikael Johansson, John S Baras, and Karl H Johansson. The evolution of beliefs over signed social networks. *arXiv preprint arXiv:1307.0539*, 2013.

[116] Claudio Altafini. Consensus problems on networks with antagonistic interactions. *Automatic Control, IEEE Transactions on*, 58(4), 2013.

[117] Claudio Altafini and Gabriele Lini. Predictable dynamics of opinion forming for networks with antagonistic interactions. *Automatic Control, IEEE Transactions on*, 60(2):342–357, 2015.

[118] Claudio Altafini. Dynamics of opinion forming in structurally balanced social networks. *PloS one*, 7(6):e38135, 2012.

[119] Dorwin Cartwright and Frank Harary. Structural balance: a generalization of heider's theory. *Psychological review*, 63(5):277, 1956.

[120] Abed Elhashash and Daniel B Szyld. On general matrices having the perron-frobenius property. *Electronic Journal of Linear Algebra*, 2008.

[121] NJ Pullman. A geometric approach to the theory of nonnegative matrices. *Linear Algebra and its Applications*, 4(4), 1971.

[122] Dimitrios Noutsos. On perron–frobenius property of matrices having some negative entries. *Linear Algebra and its Applications*, 2006.

[123] Dimitrios Noutsos and Michael J Tsatsomeros. Reachability and holdability of nonnegative states. *SIAM Journal on Matrix Analysis and Applications*, 30(2):700–712, 2008.

[124] Hal L Smith. *Monotone dynamical systems: an introduction to the theory of competitive and cooperative systems*, volume 41. American Mathematical Soc., 2008.

[125] Wei Ren and Randal W Beard. Overview of consensus algorithms in cooperative control. *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*, pages 3–22, 2008.

[126] Bamshad Mobasher, Robin Burke, Runa Bhaumik, and Chad Williams. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Transactions on Internet Technology (TOIT)*, 7(4):23, 2007.

[127] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[128] Peixin Gao, Hui Miao, and J.S. Baras. Social network ad allocation via hyperbolic embedding. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 4875–4880. IEEE, Dec 2014.