Random Models and Heuristic Algorithms for Correlation Clustering Problems on Signed Social Networks

by

Dewan Ferdous Wahid

B.Sc. Hons., University of Chittagong, 2008
M.Sc., University of Chittagong, 2009

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE COLLEGE OF GRADUATE STUDIES

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

April 2017

The undersigned certify that they have read, and recommend to the College of Graduate Studies for acceptance, a thesis entitled: RANDOM MODELS AND HEURISTIC ALGORITHMS FOR CORRELATION CLUSTERING PROBLEMS ON SIGNED SOCIAL NETWORKS submitted by DEWAN FERDOUS WAHID in partial fulfilment of the requirements of the degree of Master of Science

---

Supervisor, Dr. Yong Gao, Professor, Computer Science, UBC, Okanagan

---

Co-supervisor, Dr. Paramjit Gill, Associate Professor, Statistics, UBC, Okanagan

---

Supervisory Committee Member, Dr. Heinz Bauschke, Professor, Mathematics, UBC, Okanagan

---

University Examiner, Dr. Zheng Liu, Associate Professor, School of Engineering, UBC, Okanagan

---

External Examiner, Professor (please print name and faculty/school above the line)

(Date Submitted to Grad Studies)

Additional Committee Members include:

(please print name and faculty/school above the line)

(please print name and faculty/school above the line)

Abstract

In social sciences, the signed directed networks are used to represent the mutual friendship and foe attitudes among the members of a social group. Recent studies show that different real-world properties (e.g. preferential attachment, copying etc.) can be observed in the web-based social networks. In this thesis, we study the positive/negative - in/out - degree distributions in three online signed directed social networks. We observe that all signed-directed degree distributions in the web-based social networks with multiple edges possibilities (in both directions) follow a power law with exponents in the range $2.0 \leq \gamma \leq 3.5$. We present three random models, which capture the preferential attachment and copying properties, for web-based signed directed social networks. The signed-directed degree distributions in the networks simulated by the proposed random models also indicate a power-law trait with an exponent in the range $2.0 \leq \gamma \leq 3.5$.

We also present a heuristic algorithm for the Correlation Clustering (CC) which is a class of community detection problem in the signed network. The CC problem can be defined as follow: for a given signed network, finding an optimal partition in the vertices such that the edges inside a group are positives and the edges between two groups are negative. We present the algorithm based on the relaxing integer linear programming formulation of the minimum disagreement CC problem and rounding the approximate ultrametric distance matrix by using a given *threshold*. The experimental results show that, in the random signed $G(n, e, p)$ network, the runtime of this algorithm is nearly independent for the cases $e \geq 0.4$ and $p \leq 0.6$, where e and p are the probabilities of connecting two vertices by an edge and an edge to be positive respectively. But this algorithm does not give any convincing argument in the variation of the minimum disagreements due to the changing of the given *threshold*. We also apply this algorithm to the International National Bilateral Tread Growth Network derived from the bilateral trading data in 2011-2015 from the International Trade Center (ITC) to identify the groups of countries with average positive trade growth.

Table of Contents

List of Tables

List of Figures

Acknowledgments

First, I would like to thank to my supervisor Professor Dr.Yong Gao for his mentorship and support throughout this program and research. His patience, motivation, enthusiasm and immense knowledge have been the driving force behind all the work in this thesis.

A very special thank goes to my co-supervisor Dr. Paramjit Gill for his encouragements and insightful comments.

A special gratitude goes out to the university examiner Dr. Zheng Liu for his profound insights to my dissertation.

I would also like to thank Dr.James Nastos for always being there whenever I needed help in object oriented programming.

Finally, I am incredibly grateful to my family, friends and roommates for always being the solid anchor in my life.

Chapter 1

Introduction

Scientists are using networks to explain different real-world complex phenomena for a long times. For example, in social sciences, people are using the concept of social interaction by the words 'web', 'social fabric' and 'network.' In 1934, Moreno [Mor34] first used a structure, called *'sociogram'* to represent the formal properties of social configuration. In sociogram, he represented individuals by 'points' and their social attitudes to one another by 'lines.' Moreno proposed to use the sociometric 'star' to identify leaders and isolated individuals based on the popularity of the social group. In 1946, Heider [Hei46] first used the signed version on the network, in which the edges are labeled by positive and negative signs, to represent the mutual attitudes of friendship and foe behaviors in a social group respectively. Heider [Hei46] used *signed directed networks* to introduce the notion of balance theory. In 1965 Cartwright and Harary [CH56] formalized the definition of balance state in graph-theoretic language. According to their definition, a signed network is in balance state if there exists two or more subgroups/partitions in the network such that the mutual interactions among the members in the same subgroup are supportive (i.e. connected with positive edge) and the attitudes between two subgroups are hostile (i.e. connected with negative edges). Later, the both directed and undirected versions of signed networks have been growing significantly in the different scientific disciplines such as computer science, biology, and physics, etc.. The analysis of these networks is evolving in both data-centric and problem-centric perspectives.

Before the beginning of the world-wide-web era, the signed networks involved a small number of vertices and edges in general and usually derived from studying the physical world [TCAL16]. With the development of the online social networks the number of the web-based signed networks, such as Epinions Trust Network [LHK10], Slashdot [LHK10], have been increasing significantly in the recent times. In a web-based signed social network, the mutual positive and negative interactions are often determined by the like/dislike, or trust/distrust between two users of a common platform. The vertex sets in the web-based signed social networks are most often enormous in size, but the networks are sometimes very sparse and noisy.

To examine new ideas or to find solutions for the real-world complex network problems, it is always desirable to test those ideas/solutions on an artificial network with tractable structural properties that can precisely simulate the real-world networks phenomena. Over the years, several attempts have been proposed to design random models for the web and social networks that can capture different real-world properties. The first attempt to design such model can be seen in Watts-Strogatz's [WS98] *small-world* model proposed in 1998. This model successfully captures the 'small-world' property: having high density and small diameter, which can be found in many real-world networks such as neural, power grid and film actors collaboration networks. This property was studied by some social scientists including Milgram [Mil67] in the early 1960s. In 1999, Faloutsos, Faloutsos, and Faloutsos [FFF99] observed that the degree distributions in many real-world networks such as the Internet network follow a certain power-law also known as 'scale-free' property. The *preferential-attachment* model proposed by Barabási and Albert [BA99] in 1999 successfully captured the scale-free property in the random networks. The *copying model* captures the process of creating a new web page by copying and then modifying links from existing web-pages [KRR$^+$00].

Due to the evolution of web-based signed networks, there is now a considerable necessity of designing random models to capture different aspects of these networks. Recently, Ciotti et al. [CBC$^+$15] studied the signed-degree distributions in the signed social networks such as Epinions-trust (`www.epinions.com`) and Slashdot (`www.slashdot.org`) networks. Their study suggests that the signed-degree distributions of those networks follow a power law with exponent $2.2 \leq \gamma \leq 4.5$. Ciotti et al. [CBC$^+$15] also proposed the *power-law degree distribution model* for signed undirected network to capture this property. To the best of our knowledge, there has been no study on the random modeling and the degree distributions in the signed directed networks. In this thesis, we have studied singed-directed degree distributions in three real-world signed directed social networks: Wikipedia Request for Adminship (`www.wikipedia.org`), Epinions-trust (`www.epinions.com`) and Slashdot (`www.slashdot.org`). Our study suggests that the signed-directed degree distributions in those networks obey the power-law with an exponent in the range $2.0 \leq \gamma \leq 3.5$. Then, we propose three random models for signed directed social network to capture the properties observing in the real-world networks.

According to Cartwright and Harary [CH56]'s study in 1956, a balanced signed network can be partitioned into one or more mutually hostile (i.e. negatively connected) *balanced communities*. A balanced community is a group of vertices in a social network that is "positively connected", i.e., the mutual interaction among the member inside the community are supportive/friendly. In 2004, Bansal et al. [BBC04] formulated the CORRELATION CLUSTERING problem to find a optimal partition in the signed networks. Later, this problem is becoming a very natural way of identifying communities in network analysis [MMP12] as well as other scientific areas such as machine learning and data mining [CDK14, GMT07], portfolio analysis in risk management [FF14, HLW02], biological system networks [HBN07, DESZ07] etc..

The CORRELATION CLUSTERING problem is NP-hard [BBC04]. In recent years, several approximate algorithms have been proposed by Bansal et.al. [BBC04], Ailon, Charikar, and Newman [AAELvZ12], Charikar et al. [CGW03], Demaine et al. [DEFI06] etc.. In this thesis we propose a heuristic algorithm for the CORRELATION CLUSTERING problem and then apply it to International Bilateral Trade Growth network.

The rest of the chapters of this thesis are organized as following.

Chapter 2 introduces definitions, notation and background material for the rest of the thesis.

Chapter 3 proposes three models to generate signed directed random network in which the signed-directed-degrees follow power-law distributions. These models are *preferential attachment model*, *edge copying model* and *clique copying model*.

Chapter 4 presents a heuristic algorithm for the CORRELATION CLUSTERING (CC) problems by solving the relaxed integer linear program of the CC problem and then finding the closest ultrametric distance matrix problem from the solution matrix of the relaxed problem.

Summary and concluding remarks are presented in Chapter 5.

Chapter 2

Preliminaries

In this chapter, we present graph-theoretic background material and notions used in the following chapters. All graph-theoretic definitions, terminologies, models, and algorithms are used in this thesis follow the books *Network Analysis* [BE05], *Algorithm Design* [KT06], and *Handbook of Graphs and Networks* [BS06].

2.1 Graphs and Networks

A *network* is an abstract structure, which represents the mutual interactions among different objects/members called vertices. For example, a social group is a network composed of vertices (members/persons) and the mutual friendship/foe attitudes between the members as the connection between these vertices. Mathematically, a network can be represented by a graph.

A *graph* $G = (V, E)$ is a structure formed by a set V of vertices and a set E of edges that connect pair of vertices. In this thesis, we use *network* and *graph* interchangeably.

An *edge* $e \in E$ that connects two vertices can be written as $e = (u, v)$, or $\{u, v\}$ or simply as uv, where $u, v \in V$. Different attributes can be assigned to an edge e (e.g. sign, direction etc.), based on the nature of the relation between the vertices u and v.

Based on the edge direction attribute, we have two types of networks: undirected and directed.

An *undirected network* is defined by $G = (V, E)$, where each edge in E is undirected. In an undirected network, (u, v) and (v, u) represent the same edge. The vertices u and v are called *endpoints* of the edge $e = (u, v)$. The number of distinct edges having a vertex v as an endpoint is called the *degree* of v. The degree of $v \in V$ in G is denoted by $d_G(v)$.

A *directed network* is also defined by $G = (V, E)$, where each edge $(u, v) \in E$ is directed. In the edge (u, v), the vertex u is called *source vertex* and v is called *target vertex*. That is, in directed network, (u, v) and (v, u) represent two distinct edges between u and v in two opposite directions. The number of distinct edges having a vertex v as the source vertex is called the *out-degree* of v and is denoted by $d_G^{out}(v)$. Similarly, the number of distinct edges having a vertex v as the target vertex is called the *in-degree* of v and is denoted by $d_G^{in}(v)$.

If we consider both edge attributes, sign and direction together, then we can categorize two types of networks: signed undirected network, and signed directed network.

A *signed undirected network* or simply *signed network* is defined by $G = (V, E, s)$, where V is the vertex set and E is the edge set. Also, the function $s : E \rightarrow \{+, -\}$ assigns a sign to each edge in E. Based on the sign, the edge set E can be written as $E = E^+ \cup E^-$, where E^+ is the set of all positive edges, E^- is the set of all negative edges, and $E^+ \cap E^- = \varnothing$. That is, the edges $(u, v) = (v, u)$, if $s(u, v) = s(v, u)$, where $u \in V$ and $v \in V$ are the endpoints. The number of distinct positive edges having a vertex v as an endpoint is called *positive-degree* of v and is denoted by $d_G^+(v)$. Similarly, the number of distinct positive edges having a vertex v as an endpoint is called the *negative-degree* of v and is denoted by $d_G^-(v)$.

A *signed directed network* is also defined by $G = (V, E, s)$, where V is the set of all vertices and E is the set of all directed edges in G and $s : E \rightarrow \{+, -\}$. Also, $E = E^+ \cup E^-$, $E^+ \cap E^- = \varnothing$, where E^+ is the set of all positive directed edges and E^- is the set of all negative directed edges. The number of distinct positive edges having a vertex v as the source vertex is called the *positive-out-degree* of v and is denoted by $d_G^{+out}(v)$. Also, the number of distinct positive edges having a vertex v as the target vertex is called the *positive-in-degree* of v and is denoted by $d_G^{+in}(v)$. Similarly, the distinct number negative edges having a vertex v as the source and target vertex are defined by the *negative-out-degree* $d_G^{-out}(v)$ and *negative-in-degree* $d_G^{-in}(v)$ respectively.

A *dynamic network* that evolves with the time t is denoted by $G_t = (V_t, E_t)$, where V_t and E_t are the vertex and edge sets in G_t at time t. Here, the sets V_t and E_t depend on time t, i.e. $V_t = f(t)$ and $E_t = g(t)$, where f and g are some functions of t.

A *subgraph* or *subnetwork* $H = (V', E')$ of a network $G = (V, E)$ is also a network such that $V' \subseteq V$ and $E' \subseteq E$. A subgraph $H = (V', E')$ of a graph $G = (V, E)$ is said to be an *induce subgraph* if $V' \subseteq V$, $E' \subseteq E$, where for every pair $u, v \in V'$, $(u, v) \in E'$ only if $(u, v) \in E$.

A *path* P in $G = (V, E)$ is a sequence of distinct vertices $v_1, ..., v_k$ such that $(v_i, v_{i+1}) \in E$ where $1 \leq i \leq k$. We can denote a path between two vertices $u, v \in V$ by $P(u, v)$. If there exist a path between vertices u and v, they are called *connected* vertices. A path $v_1, ..., v_k$ is said to be a *cycle* if

$(v_1, v_k) \in E$. A cycle of three vertices is called a *triangle*.

A *clique* C is a set of vertices in $G = (V, E)$ such that for all $u, v \in C$, $u \neq v$ implies $(u, v) \in E$. In other words, a clique is a set of vertices which are pairwise adjacent. A clique is said to be a *maximal clique* if it is not a subgraph in any other clique. A *maximum clique* in G is the clique with the maximum number of vertices. A maximum clique is a maximal clique, but the converse is not always true.

A *tree* is an undirected, acyclic connected graph. A graph in which every disjoint connected component is called *forest*. A *spanning tree* $T = (V, E_T)$ of an undirected graph $G = (V, E)$ is a tree that includes all vertices of G and $E_T \subseteq E$.

2.2 Random Network Models

The proposed network models can be divided into four groups: classical random network model, small-world property model, scale-free models, and signed random network models. A brief discussion and definitions are given in the following.

2.2.1 Erdős-Rényi Model

Erdős-Rényi [ER59] model is first classical model for generating random network.

Definition 2.1 (Erdős-Rényi Model)**.** The model generates a network $G(n, p)$, where n is the fixed number of vertices and p is the probability of joining any two vertices by an edge. Each edge is created independently of other edges, therefore, the probability distribution $P(d)$ of the degree d of a vertex v in $G(n, p)$ is

$$P(d) = \binom{n-1}{d} p^d (1-p)^{n-1-d}. \tag{2.1}$$

When, np the average degree of a vertex is a fixed constant c, then this probability approaches to the Poisson probability $\frac{c^d e^{-c}}{d!}$ as $n \to \infty$ [BE05].

2.2.2 Watts-Strogatz's Small-World Model

Many real-world networks (e.g. neural networks, the power-grid network of the western US and the collaboration network of film actors) exhibit *'small-world'* properties, which are having a small diameter and highly clustered networks. Watts-Strogatz[WS98] models simulate 'small-world' properties of real-world networks. Where the model parameters are, the number of vertices N, the average degree of a vertex d such that $\ln(N) < d < N$ and probability of edge rewiring $\beta \in [0, 1]$. The Watts-Strogatz's Small-World model is defined as follows:

Definition 2.2 (Small-World Model). Initially, the network G starts with a set of vertices V of size n placed in a cyclic order and with no edge. Then, it follows:

(1) Connect each vertex $v \in V$ with next $\frac{d}{2}$ vertices on both sides of v from the cycle.

(2) For each vertex $u \in V$,

   (a) select a vertex $v \in V$ such that $u \neq v$ and $(u, v) \notin E$,

   (b) rewire edge (u, v) with the probability β.

When the rewiring parameter $\beta \to 1$ the generated model network work approaches to the Erdős-Rényi network [WS98].

Beside the 'small-world' properties, many real-world networks (e.g. Internet, telephone call networks etc.) show *'scale-free'* property, which is the existence of hubs. That is, these networks show a *power-law (heavy-tailed) distribution* in vertex degrees. The following network models are proposed to capture the 'scale-free' property of real-world networks.

2.2.3 Barabási-Albert Preferential Attachment Model

Barabási-Albert [BA99] proposed this simple but elegant random model to grasp two important phenomena of real-work networks such as world-wide-web (**www**) etc. The first phenomenon is *growth*, i.e. the network grows with the time and there is no restriction on the number of vertices that can be added to the network. The second phenomenon is *preferential attachment*, which often is referred as the '*rich-getting-richer*' property. The Barabási-Albert model is defined as follows:

Definition 2.3 (Barabási-Albert Model). At $t = 0$, the random process starts with an initial connected network G_0^k of size $|V_0| = m_0$ and $m_0 \geq k$. Then, a sequence of vertices, $v_1, v_2, ..., v_N$, are entered to the existing network inductively, one vertex at a time, to produce a sequence of networks

$\{G_t^k\}$ by connecting with k number of existing vertices. At time t, the new vertex v_{t+1} enters to the network $G_t^k(V_t, E_t)$ and connects with an existing vertex $v \in V_t$ with the probability

$$\mathbb{P}[v_{t+1} \text{ connects with } v] = \frac{d_{G_t^k}(v)}{\sum_{v \in V_t}^{N} d_{G_t^k}(v)}, \tag{2.2}$$

where, $d_{G_t^k}(v)$ is the degree of the vertex v in G_t^k and $\sum_{v \in V_t}^{N} d_{G_t^k}(v)$ is the total degree of all vertices in G_t^k.

Barabási-Albert [BA99] proved the following theorem to find the power law bound of the degree distribution.

Theorem 2.4. *The probability distribution $P(d)$ of the degree d of a vertex is reduced to d^{-3} for large d when $t \to 0$.*

2.2.4 Cooper-Frieze Model

The Cooper-Frieze model, proposed in [CF03], is a a mixture of preferential attachment (by degree) and uniformly at random (u.a.r) selection. This model needs to fix a set of parameters in advance. The fixed parameters are defined as follows ([CF03]):

Procedure selection at each step t:

α : Probability to follow OLD procedure,

$1 - \alpha$: Probability to follow NEW procedure,

Procedure NEW:

$\boldsymbol{p} = (p_i : i \geq 1)$: Probability that new vertex generates i new edges,

β : Probability that the target vertices are selected uniformly,

$1 - \beta$: Probability that the target vertices are selected accordant to degree,

Procedure OLD:

$\boldsymbol{q} = (q_i : i \geq 1)$: Probability that existing vertex generates i new edges,

δ : Probability that the source vertex is selected uniformly,

$1 - \delta$: Probability that the source vertex is selected accordant to degree,

γ : Probability that the target vertices are selected uniformly,

$1 - \gamma$: Probability that the target vertices are selected accordant to degree,

This model also needs two fixed integer parameters j_0 and j_1 such that $p_j = 0, j > j_0$ and $q_j = 0, j > j_1$. The model definition is given in following.

Definition 2.5 (Cooper-Frieze Model)**.** The random process starts with an initial network G_0 with single vertex v_0 and no edge. Then a sequence of random networks $\{G_t\}$ evolves according to the following procedure.

At time t, the edges are added by choosing either NEW or OLD method with probability $1 - \alpha$ or α respectively. In NEW method, a new vertex v_{t+1} is added to the network $G_t^k(V_t, E_t)$, and connects v_{t+1} with one or more existing vertices. In OLD method, a number of new edges are added to a selected existing vertex v.

This model follows a mixture of preferential attachment (by degree) and uniformly at random (u.a.r) rules to select the endpoints, target vertex in the NEW method and source & target vertices in the OLD method, for the newly added edges.

Let, at time t, let $\mathbb{E}[X_d(t)]$ be the expected numbers of vertices with degree d in G_t and $\{\beta_d\}$ is a sequence of positive integers. Cooper et al. [CF03] proved that the following theorem for $t \to \infty$ and small k.

Theorem 2.6 ([CF03])**.** *There exist a constant $M > 0$ such that almost surely for all t, $k \geq 1$*

$$|\mathbb{E}[X_d(t)] - t\beta_d| \leq Mt^{1/2} \log t$$

Therefore, for $t \to \infty$ and small k, $\mathbb{E}[X_d(t)]$ can be approximated by $t\beta_d$, i.e. $\mathbb{E}[X_d(t)] \approx t\beta_d$, for $t \to \infty$, where β_d is a sequence of positive integers, which obeys power-law bounds [CF03].

2.2.5 Copying Model

Kumar et al. [KRR$^+$00] proposed this model to capture the copying property which is observed in the web-base networks. The core idea behind this copying property is that, when a new web-page (vertex) is created, most often it copies all out-links (directed edges) from an existing web-page and then modifies some links. Based on this observation the copying model can be defined as follows:

Definition 2.7 (Copying Model)**.** At time t, a new vertex v_{t+1} enters to the network $G_t^k(V_t, E_t)$ and creates k directed edges (out-links) as follows:

(1) Selects a 'prototype' vertex $v \in V_t$ uniformly at random.

(2) For all $(v, w) \in E_t$, such that $w \in V_t$, adds (v_{t+1}, w) to the network G_{t+1}^k.

(3) For each edge $(v_{t+1}, w) \in E_{t+1}$, such that $w \in V_{t+1}$,

   (i) with the probability $1 - \alpha$ re-wires the edge with a randomly selected vertex $u \in V_t$.

   (ii) with probability α, keeps the edge unchanged.

Let $\mathbb{E}[X_d(t)]$ be the expected number of vertices of degree d in the network generated by the copying model at time d. Then the following results was proved by Kumar et al. [KRR$^+$00].

Theorem 2.8. *For $d > 0$, the limit $P(d) = \lim_{t\to\infty} \frac{\mathbb{E}[X_d(t)]}{t}$ exist, and satisfies*

$$P(d) = P(0) \prod_{i=1}^{r} \frac{1 + \frac{\alpha}{i(1-\alpha)}}{1 + \frac{2}{i(1-\alpha)}}$$

and

$$P(d) = \Theta(d^{\frac{2-\alpha}{a-\alpha}}).$$

The motivation of this model is that it creates a lot of induced bipartite subgraphs that are common phenomena in real world web networks. But the networks generated by copying model do not show high clustering, which is another common phenomenon of web networks.

2.2.6 k-Tree Random Model

Gao [Gao09] proposed the k-Tree random model which can generate random networks with a well-defined graph structure. The degree distribution in the simulated networks obeys a power-law. The model definition is given in following ([Gao09]):

Definition 2.9 (k-Tree Random Model)**.** The random process starts with an initial clique G_0^k of size $|V_t| = k+1$. A sequence of vertices, $\{v_1, v_2, ..., v_N\}$, is added to the existing network inductively to generate a sequence of random networks $\{G_t^k\}$. At time t, a new vertex v_{t+1} enters the existing network $G_t^k(V_t, E_t)$ and generate G_{t+1}^k as follows.

(1) Selects k-clique, $\mathcal{C}_t$, uniformly at random from $\{G_t^k\}$.

(2) Connects v_{t+1} with all k vertices in $\mathcal{C}_t$

Let, X_d be the random variable for the total number of vertices of degree d in G_t^k and $\{\beta_d\}$ be a sequence of positive integers. Gao [Gao09] proved the following theorem to approximate the expected number of vertices with degree d.

Theorem 2.10 ([Gao09])**.** *Let,* $\mathbb{E}[X_d(t)]$ *be the expected number of be vertices with degree* d *in the random* k*-tree* G_t^k*. There exists a constant* $N = N(k)$ *(independent of* d*) such that for any* $n > N$*,*

$$|\mathbb{E}[X_d(t)] - -t\beta_d| \leq C$$

where $C = C(k)$ *is a constant that is independent of* d *and* n *and* β_d *obeys a power law bound*

$$d^{-\left(1+\frac{k}{k-1}\right)}$$

.

In 2011, Sridharan et al. [SGWN11] showed that the edge embeddedness $d_{G_t^k}(e) = D$ of k-tree random network also follows a power-law $D^{-\left(1+\frac{k}{k-2}\right)}$.

2.2.7 Signed Random Network Models

Recently, Ciotti et al. [CBC+15] has proposed two models for signed social networks: *Binomial degree distribution model* and *Power-law degree distribution model*.

Definition 2.11 (Binomial Degree Distribution Model)**.** This model constructs a signed random network by applying the following procedure:

(1) Generating Unsigned Network: Generate an unsigned network $G(V, E)$ of size $|V| = N$, by connecting any pair of vertices through an edge with probability p.

(2) Attributing Signs: Attribute sign to each edge in G as follows:

  (i) Divide all vertices in V into two groups A and B with probabilities m and $1 - m$ respectively.

  (ii) An edges is attributed by the positive sign if the end vertices are in the same group, otherwise attributed by the negative sign.

Definition 2.12 (Power-law Degree Distribution Model)**.** According to this model, we can construct a signed random network with power-law positive and negative degree distributions by applying following procedures:

(1) Generating Unsigned Network: Generate an unsigned network $G(V, E)$ of size $|V| = N$ by a power-law degree distribution network model, e.g. Barabási-Albert model, copying model, etc..

(2) Attributing signs: Attribute sign to each edge in G by following the similar procedure from the above *Binomial Distribution Model.*

2.3 Algorithmic Problems on Graphs

Many algorithmic problems have been studied on graphs over the years. Here, we only give a short discussion on the algorithmic problems relevant to this thesis.

2.3.1 Shortest Path

The shortest path problem on a weighted, directed, and connected graph $G = (V, E, w)$ in which V is the vertices set, E is the edge set, and $w : E \to \mathbb{R}_0^+$ is the distance (weight) function for each edge in E, can be defined as follows:

Problem 2.1. SHORTEST PATH.
INSTANCE: Given a weighted, directed, and connected graph $G = (V, E, w)$ and a fixed source vertex s.
TASK: Find the shortest path from s to every other vertex in $v \in G$.

Different algorithms have been proposed for solving the shortest path problems such as Dijkstra's algorithm [Dij59], Bellman-Ford algorithm [Bel58], Floyed-Warshall algorithm [Flo62], etc.. Here, we discuss the Dijkstra's algorithm for solving the single-source shortest path problem.

Dijkstra's Algorithm: Let $S \subseteq V$ such that the final shortest-path of the vertices in S from a fixed source s have already been determined. Let $\bar{S}$ be the complement of S in V, i.e. $\bar{S} = V \setminus S$ and $d(s, \bar{S})$ be the shortest-path distance from s to any vertex in $\bar{S}$.

Consider a path $P = (s, ..., u, v)$ from the source vertex s to a vertex $v \in \bar{S}$ and $u \in S$. Therefore, the path distance $d(s, u)$ must be the shortest-path distance from s to u. Thus the shortest-path from s to v can be written as

$$d(s, v) = d(s, u) + w_{uv},$$

where w_{uv} is the distance (weight) of the edge (u, v). Therefore, we can find the shortest-path distance from s to any vertex in $\bar{S}$ by

$$d(s, \bar{S}) = \min_{u \in S;\, v \in \bar{S}} \{d(s, u) + w_{uv}\}.$$

Initially, Dijkstra's algorithm starts with vertex set $S_0 = \{s\}$. Then, a sequence of vertices sets $S_1, S_2, ... \subseteq V$ is constructed which satisfies the following conditions.

1. In S_0, the shortest-path distance $d(s, s) = 0$.
2. If $S = \{s, u_1, ..., u_i\}$, where $s, u_1, ..., u_i \in V$, then $d(s, u_1) \leq ... \leq d(s, u_i)$.
3. In the step of constructing the set S_i, the shortest-path distances from the source s to all of the vertices $u_1, ..., u_i$ are already known.

The generic Dijkstra's algorithm can be represented by the pseudo-code given in *Algorithm 1*.

Algorithm 1: DIJKSTRA'S ALGORITHM

Data: A directed, weighted graph $G(V, E, w)$ and source vertex s.
Result: $S \leftarrow$ the set of explored vertices.
$d(s, u) \leftarrow$ the shortest-path distance from s, $\forall u \in S$.
Initialization: $S \leftarrow s$; $d(s, s) \leftarrow 0$;
while $S \neq V$ **do**
  Select a vertex $v \in \bar{S}$ such that
  $d_{min}(u, v) = \min_{u \in S;(uv) \in E}\{d(s, u) + w_{uv}\}$;
  $d(u, v) \leftarrow d_{min}(u, v)$;
  $S \leftarrow S \cup \{v\}$;
end

The runtime of a straightforward implementation of Dijkstra's algorithm needs $\mathcal{O}(mn)$ whereas a min-priority queue implemented by a Fibonacci-heap based implementation takes $\mathcal{O}(m \log n)$ [FT87].

2.3.2 Minimum Spanning Tree

Let $G = (V, E, w)$ be an undirected, weighted connected graph in which $w : E \rightarrow \mathbb{R}_0^+$ is the weight function that defines the weight of an edge $(u, v) \in E$ by w_{uv}. A *minimum spanning tree* of G is a spanning tree with minimum total edges weight. Consider the weight for an edge $(u, v) \in E$ is w_{uv}, which is defined by the cost to connect the vertices $u, v \in V$. Then, we can define the minimum spanning tree problem as follows:

Problem 2.2. MINIMUM SPANNING TREE;
INSTANCE: An undirected, weighted connected graph $G = (V, E, w)$.
QUESTION: Find a spanning tree $T = (V, E_T)$ such that, $w(T) = \sum_{(u,v) \in E_T} w_{uv}$ is minimized. Here $w(T)$ is the total weight of the edges in the spanning tree T.

There are several algorithms for finding the minimum spanning tree: Kruskal's algorithm [Kru56], Prim's algorithm [Pri57], etc. In the following section, we briefly discuss the Kruskal's algorithm.

Algorithm 2: KRUSKAL'S ALGORITHM[Cor09]

Input: An undirected, weighted graph $G(V, E, w)$.
Output: Minimum spanning tree $T(V, E_T)$.
Initialization: $E_T = \phi$
for *each vertex* $v \in V$ **do**
    MAKE-SET(v)
end
$E_{sort} \leftarrow$ sorted edges in E into nondecreasing order by weight w
for *each edge* $(u, v) \in E_{sort}$ **do**
    if FIND-SET(u) $\neq$FIND-SET(v) **then**
        $E_T \leftarrow E_T \cup (u, v)$
    end
end
return E_T

Kruskal's Algorithm: In the Kruskal's algorithm, proposed in [Kru56], the edges set E_T is a forest on the vertex set V of G. At each step, Kruskal's algorithm finds a safe edge $(u, v) \in E$, with least-edge-weight that connects two disjoint connected components, to add to E_T. The implementation of this algorithm needs to use UNION-FIND data structure to maintain individual disjoint sets of elements. The operation FIND-SET(u) is used to see whether or not a vertex u belongs to a set by returning the representing element from the set. The operation UNION(u, v) is used to merge two disjoint sets that contains the vertices u and v. The pseudo-code of the Kruskal's algorithm is given in *Algorithm 2*.

2.3.3 Maximal Cliques Problem

The decision problem to identify all maximal cliques in a network can be formulated as follows:

Problem 2.3. MAXIMAL CLIQUES.
INSTANCE: A graph $G = (V, E)$.
QUESTION: Find all maximal cliques in G.

Moon and Moser [MM65] showed that if $|V| = n$ then G has at most $3^{n/3}$ maximal cliques. Bron and Kerbosch [BK73] proposed a recursive backtracking algorithm for identifying maximal cliques in an undirected graph. This algorithm is known as *Bron-Kerbosch* algorithm.

Consider an undirected graph $G = (V, E)$ and three vertices sets R, P, and X. The Bron-Kerbosch algorithm finds the set R that belongs the maximal clique with all vertices of V, the set P that belongs the maximal cliques with some vertices of V and the null set X. The pseudo-code of the classical implementation of the Bron-Kerbosch algorithm is given bellow.

Algorithm 3: Bron-Krebosch Algorithm

```
Bron-Krebosch(R, P, X)
if P and X are both empty then
 | report R as a maximal clique;
end
for each vertex v in P do
 | BronKrebosch(R ∪ v, P ∩ N(v), X ∩ N(v));
 | P ← P \ v;
 | X ← X ∪ v;
end
```

2.3.4 Maximum Clique Problem

The problem to identify a maximum clique in a network can be formulated as follows:

Problem 2.4. Maximum Clique.
Instance: A graph $G = (V, E)$.
Task: Find a set of vertices $S \subseteq V$, if there exist, of size at least k such that for every $u, v \in S$, $(u, v) \in E$, i.e. S is a maximum clique in G.

The maximum clique problem is NP hard [GJ79] and it is computationally equivalent to some other algorithm problems, e.g., *minimum vertex cover problem*, *maximum independent set problem*.

2.4 Network Communities and Clustering

Let $G = (V, E)$ be an unweighted and undirected graph in which V is the vertex set, and E is the edge set. A community in G is a set $\mathcal{C} \subseteq V$ in which any vertex $v \in \mathcal{C}$ has comparatively more connections compared to the connection with any other vertex in $V \setminus \mathcal{C}$. The attribute of the high edge density inside a community can be categorized by different graph classes. Gao [Gao14] proposed the following general graph-theoretic definition of community.

Definition 2.13 (Π-Community)**.** A Π-community (or a Π-graph) in a network is a maximal, connected, and induced subgraph that belongs to the Π-graph class.

Therefore, identifying communities in a network can be interpreted as identifying subgraphs induced by a particular Π-graph class. A Π*-graph class* is a subgraph defined with a particular structural property. A Π-graph class is called *hereditary* if it satisfies the following property: for $G \in \Pi$, every induced subgraph H of G is also in Π.

In the following subsections, we briefly discuss on some of the graph classes.

2.4.1 Cluster Graphs

An undirected graph $G = (V, E)$ is said to be a cluster graph if every connected component in G is a maximal clique. That is, a cluster graph is a collection of disjoint maximal cliques.

For any three vertices $i, j, k \in V$, the graph G is a cluster graph if and only if $(i, j), (j, k) \in E \implies (i, k) \in E$, where $(i, j) = (j, i); \forall i, j \in V$.

This above transitive property of the cluster graph can also be interpreted as P_3*-free*. Note that, an induced subgraph of three ordered vertices is called P_3 if any those three vertices are connected as a simple path. Therefore, we can say that a graph G is a cluster graph if and only if G is P_3-free.

The cluster graphs have numerous applications in different fields such as in data mining to find a group of an object with maximum inter-class similarity and minimum intra-class similarity [HPK11], in computational biology to cluster and visualize the gene expression data [SMKS03], in image segmentation [WL93], etc..

2.4.2 Quasi-Threshold Graphs

An induced subgraph of four ordered vertices is said to be a P_4 if the all four of those vertices are connected as a simple path. A C_4 is an induced subgraph of four ordered vertices in which the first and the last vertices are connected to form a close circuit.

A graph G is said to be (P_4, C_4)-*free* if and only if P_4 and C_4 are the forbidden induced subgraph classes in G. That is, in G there exist no path and close circuit of size four.

In literature, the (P_4, C_4)-*free* graphs are also known as *quasi-threshold graphs* [MWW89, YCC+96], *comparability graphs of tree* [Wol65] and *trivially perfect graphs* [Gol78]. A quasi-threshold graph can be recognized in linear time [YCC+96].

2.4.3 Cographs

A graph G is said to be a *cograph*, if there exist no P_4 induced subgraph in G. That is, cograph is a P_4-free graph. In a cograph every connected induced subgraph has a disconnected complement.

In literature, different authors studied cographs independently with different names: *decay graphs* [Sum74], D^**-graphs* [Jun78], and *2-party graphs.*

2.4.4 Threshold Graphs

A connected graph is $2K_2$-free, first studied by El-Zahar and Erdős [EZE85], if it does not contain a pair of independent edges as an induced subgraph.

A graph G is said to be *threshold* (or $(P_4, C_4, 2K_2)$-free) if and only if there exist no P_4, C_4, $2K_2$ induced subgraphs in G. It was first studied by Chvátal and Hammer [CH73].

2.4.5 Community Editing Problems

Based on Gao's [Gao14] Π-community definition, we can generalize the graph editing problems as the Π-Community Editing, which is defined as follows.

Problem 2.5. Π-Community Editing(G)
Instance: An unweighted graph $G = (V, E)$.
Task: Determine a modified graph $G' = (V, E')$ with minimum number of edge editions (insertions or deletions) in which each connected component

is a Π-community.

Based on the desired Π-community we can redefine *Problem 2.5* to different community editing problem. A brief on the well known community editing problems is given next.

Clustering Editing Problem: The CLUSTERING EDITING problem is a class of community editing problems. It can be defined from *Problem 2.5* if the desired Π-community network is a cluster graph. The CLUSTERING EDITING problem is a NP-hard [KM86]. In this thesis, we devote Chapter 4 to design a heuristic algorithm for the CORRELATION CLUSTERING EDITING problem which is a class of community editing problem on signed networks.

Quasi-Threshold Editing Problem: The QUASI-THRESHOLD EDITING problem can be formulated from *Problem 2.5* if the desired Π-community network is a quasi-threshold graph. Nastos and Gao [NG13] first studied the QUASI-THRESHOLD EDITING problem to define the communities in social networks. They also showed that this problem is NP-hard.

Cograph Editing Problem: The COGRAPH EDITING problem can be also be formulated from *Problem 2.5* if the desired Π-community network is a cograph. Liu et al. [LWGC12] showed that COGRAPH EDITING problem is NP-complete. An *Edge P_4 centrality*-based divisive algorithm for identify cograph communities in graph is proposed by Jia et al. [JGG+15].

2.5 Balance Theory and Clustering

In 1946, Heider [Hei46] first introduced the balance theory to explain the cognitive balance by resolving the sentimental inconsistency in a social system. Heider also first used the signed network to represent the mutual sentimental interaction among the members of the social system. In the positive (negative) network each vertex represents a member/person, and a sign edge represents the mutual friend (hostile) interaction between two members. According to the balance theory, the balanced state in a social system is based on the following principles:

"friend of my friend is my friend"
"enemy of my friend is my enemy"

In signed network, Heider's balance states can be represented by a triad (sub-network with three vertices) and then the number of positive edges in the triad. A balanced state in a triad can be achieved by only having an odd number of positive edges. On the other hand if a triad has an even number of positive edges then it is imbalanced. In Fig.2.1, the triads T_3 and T_1 are in balanced state, and the triad T_2 is in imbalanced.

Davis [Dav77] extended this idea by considering *"enemy of my enemy is my enemy"*, which can be represented by the triad T_0, is also a balanced state. Cartwright and Harary [CH56, Har59] formalized the definition of the balance theory in graph-theoretic language. They also showed that a signed network is said to be in balance state or structural balanced if the vertex set of the network can be partitioned into mutually hostile subgroups in which the internal attitudes (edges) among the members of a subgroup are friendly to each other.

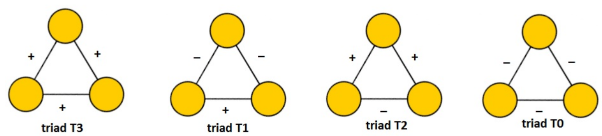


Figure 2.1: Triads with odd number or no positive edges or are balanced (T_3, T_1) and with even number edges are imbalanced (T_2).

A signed network is called *k-clusterable* or *k-correlation-clusterable* if its vertex set can be partitioned into k subgroups in such way that the signed edges inside a subgroup are positive and the signed edges between two subgroups are negative.

Chapter 3

Random Models for Signed Directed Social Networks

3.1 Signed Directed Social Network

In a social group, the mutual attitude among the members (e.g. persons) can be represented by a signed directed network $G = (V, E, s)$, where V the vertex set, E the directed-edge set and the function $s : E \to \{+, -\}$ assigns a sign for each directed-edge in the network [Hei46, Dav77].

In G, each vertex $v \in V$ represents an individual member, and each signed-directed-edge $e \in E$ represents the attitude from a source vertex (member) to a target vertex. Based on the sign, E can be partitioned as $E = E^+ \cup E^-$, where $E^+ \cap E^- = \varnothing$, and E^+ and E^- are the set of all positive and negative edges respectively. Therefore, each edge $e \in E$ has two attributes: sign and direction. A positive edge $e \in E^+$ directed from a member A to another member B indicates the friendly attitude from A to B. Similarly, a negative edge $e \in E^-$ directed from a member A to another member B indicates the hostile attitude from A to B.

In the rest of this chapter, we will simply denote signed directed social network as $G = (V, E)$.

3.2 Motivation for Modeling Signed Directed Social Networks

Our motivation to design random models for signed directed networks follows the observations of the signed-directed degree distributions in three real-world signed directed social networks: Wikipedia Request for Adminship (WikiRfA) [WPLP14], Epinions Social Netwrok [LHK10], and Slashdot Social Network [LHK10]. A brief description of the studied networks is given in the following.

3.2.1 Empirical Networks

Wiki-RfA: To be an admin from an editor, in Wikipedia (`www.wikipedia.org`), an application has to be submitted either by a candidate or by any member on behalf of a candidate. Then any member of the Wikipedia community can vote to either support (+1) or oppose (−1) or neutral (0) to the adminship request. In this network, each vertex represents a community member (either an editor or an applicant or both) and each signed-directed-edges represents a vote from a community member to an applicant. The data were collected from all the votes for RfA process between 2003 to May 2013. Since many candidates applied for adminship for several times during that period, there exist multiple signed-directed-edges between the same pair of members.

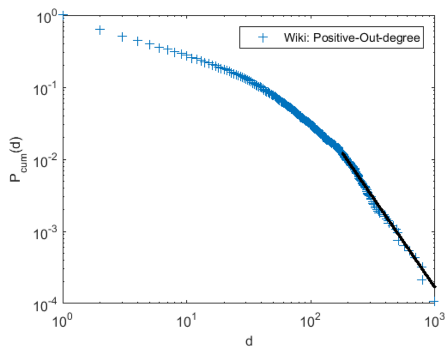
Slashdot: The Slashdot (`www.slashdot.org`) is a technology-related news website where users can tag each other as friends or foes. In this network, each tag is represented by a signed-directed-edge from a tag-given-user to a tag-receiving-user. If a user A tags another user B as a friend (foe) then there is a positive (negative) edge from A to B. This network contains 81,867 vertices (users) and 545,671 edges (links).

Epinions: This network data obtained from a general consumer review site called Epinions (`www.epinions.com`), where users can post their opinions on various products. Users can also rate each other as trustworthy (positive) or not (negative) base on their reviews. This network contains 131,828 vertices (users) and 841,372 edges (mutual attitudes).

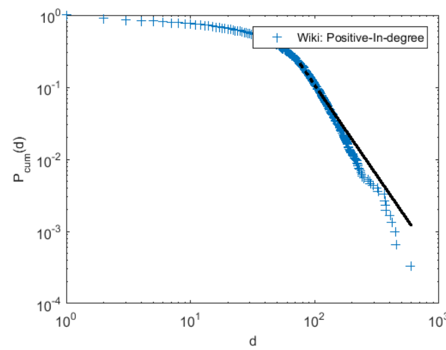
3.2.2 Analyzing Real-World Networks

To analyze the signed-directed degree distributions in our studied real-world networks, we fit the power-law distribution $p(d) \approx d^{-\gamma}$ to all of the four types signed-directed-degrees (i.e. positive-in-degree, negative-in-degree, positive-out-degree, and negative-out-degree) and calculate the values of exponent γ with the corresponding p-value individually. We use the procedure and implementations given by Clauset et al. [CSN09] to estimate the exponents γ and the corresponding p-values. In this procedure use about 2500 synthetic data sets to test the null-hypothesis against the given data set and the corresponding p-value.

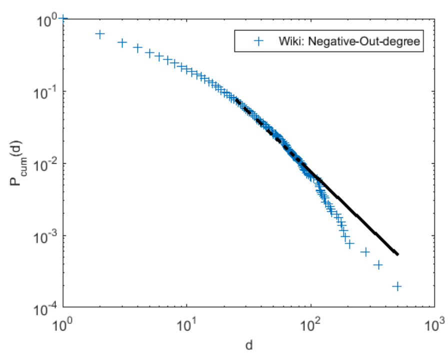
The results of the fitting power law models are given in the *Table* 3.1.



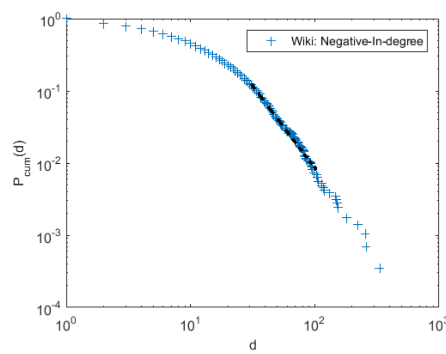
(a)



(b)

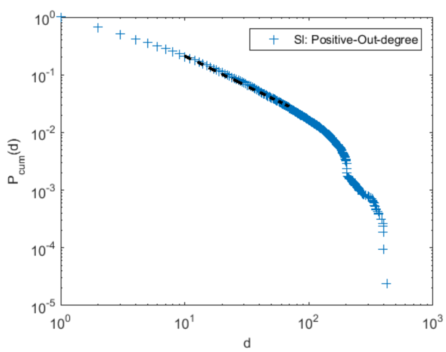


(c)

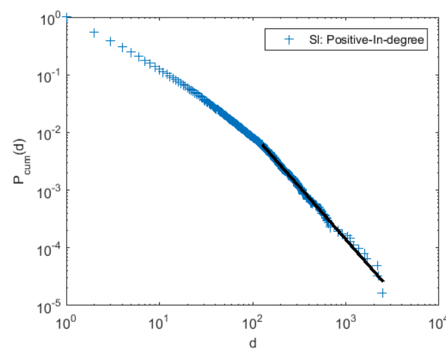


(d)

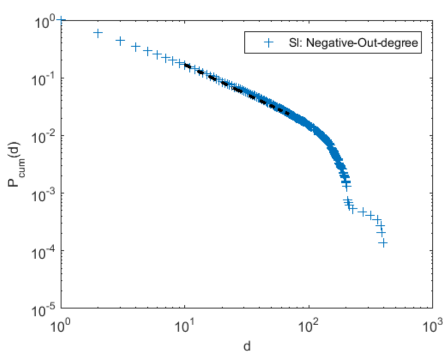
Figure 3.1: Cumulative signed-directed-degree distribution $P_{cum}(d)$ of Wiki-RfA social network.



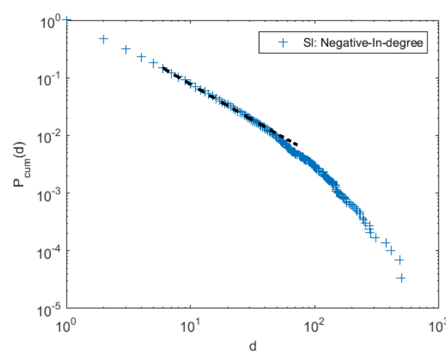
(a)



(b)

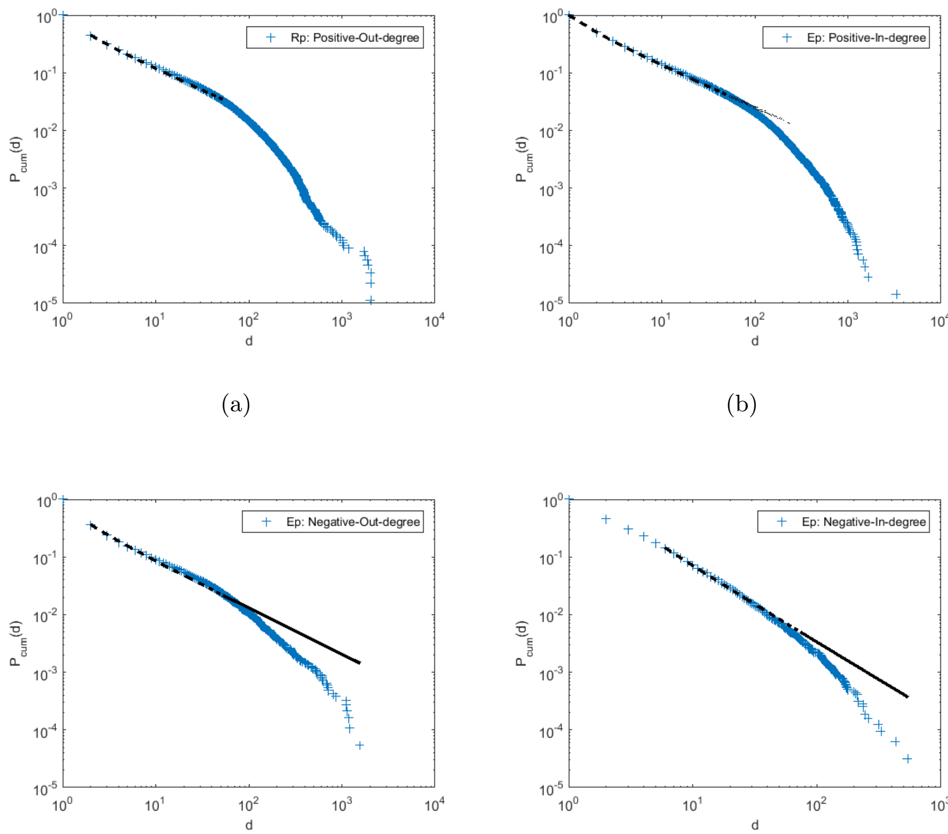


(c)



(d)

Figure 3.2: Cumulative singed-directed-degree distribution $P_{cum}(d)$ of Slashdot social network.



(a) (b) (c) (d)

Figure 3.3: Cumulative signed-directed-degree distribution $P_{cum}(d)$ of Epinions social network.

Empirical data sets results					
Datasets	$\lvert V\rvert$, $\lvert E\rvert$	Dist. type	n	γ	p-value
Wiki-RfA	11381	pos-out-deg	9331	3.500	0.529
	185612	pos-in-deg	3036	3.500	0.013
		neg-out-deg	5170	2.640	0.010
		neg-in-deg	2860	3.250	0.218
Slashdot	81867,	pos-out-deg	42105	2.020	0.000
	545671	pos-in-deg	61894	2.830	0.906
		neg-out-deg	14611	2.000	0.000
		neg-in-deg	29297	2.200	0.000
Epinions	131828,	pos-out-deg	88180	1.730	0.000
	841372	pos-in-deg	69900	1.720	0.000
		neg-out-deg	18499	1.800	0.000
		neg-in-deg	31791	2.30	0.000

Table 3.1: Power-law exponents γ and the corresponding p-values for different signed-directed-degree distributions for above empirical data sets.

Also, the Fig.3.1 - Fig.3.3 show the cumulative signed-directed-degree distributions in the studied real-world social networks.

In *Table* 3.1, we can observe that the power-law model fitting on all of the four signed-directed-degree distributions in the Wiki-RfA network as being statistically significant (i.e. $p\text{-}value \geq 0.01$). On the other hand, in Slashdot network, the power-law model fitting is statistically significant only for the positive-in-degree distribution. But for the Epinions network, none of the power-law models for the signed-directed-distributions are statistically significant.

Again in the *Table* 3.1, we see that the values of the power law components γ are in the range $2.5 \leq \gamma \leq 3.5$ for the Wiki-RfA and are in the range $2.0 \leq \gamma \leq 3.0$ for the Slashdot. But the values of γ in Epinions are less than 2.5. Therefore, from the above observations, we can say that in a network in which the signed-directed-degree distributions has a tendency to follow the power law the component γ should be in the range $2.0 \leq \gamma \leq 3.5$.

Let's look at the evolving process of the Wiki-RfA network. This data was collected over a period, and many candidates had requested for the adminship for several times during this time because of failing in the election. Therefore, there exist multiple edges between the same pair of members with the same or different signs and directions. On the other hand, in the Slashdot and Epinions networks, there exist only one edge between a pair of members. From this above observations, we can conclude that the signed-directed-degree distributions in an attitude based signed directed social network with multiple edge possibilities between a same pair of vertices follow the power-law.

In *Table* 3.1, we can observe another property in the column n which represents the number of vertices having positive/negative-in-degrees and positive/negative-out-degrees in the Wiki-RfA and Slashdot networks. In Wiki-RfA network, the numbers of vertices having positive-out-degrees and negative-out-degrees are greater than the numbers of vertices having positive-in-degrees and negative-in-degrees respectively. That is the members in this network have more prone to give votes (either positive or negative) than receiving votes. On the other hand, in Slashdot network, the numbers of vertices having positive-out-degrees and negative-out-degrees are less than the numbers of vertices having positive-in-degrees and negative-in-degrees respectively. That is the members this network have more prone than votes (either positive or negative) compare to giving votes. But this inverse relation between the numbers of in and out degree vertices (for both positive and negative) is not visible in the case of the Epinions network.

Again, from the first observation, we know that all of the four signed-directed-degree distributions in Wiki-RfA and only positive-in-degree distribution in Slashdot follow the power law property with the component γ in the ranges $2.0 \leq \gamma \leq 3.5$. But the signed-directed-degree distributions in Epinions network do not follow the power-law property. Now from our first and second observation, we can say that a signed directed social network in which the signed-directed-degree distributions have a tendency to follow power law has an inverse relation between the numbers of in and out degree vertices (for the case both positive and negative).

	Attributes/Properties	**Wiki-RfA**	**Slashdot**	**Epinions**
A1	Power-law component in the range $2.0 \leq \gamma \leq 3.5$	Yes	Yes	No
A2	# of ***in*** and ***out*** degree vertices are inversely related	Yes	Yes	No
A3	Exists multiple edges in both directions	Yes	No	No

Table 3.2: The list of observed attributes in the real-world networks. We denote these attributes by A1, A2, and A3 respectively.

The summary of observed attributes in the real-world signed directed networks is given in the *Table* 3.2.2. These observations inspired us to design random models for signed directed networks (given in sections 3.4, 3.5, and 3.6) with the above observed properties and with some other specified controlling features in the network structure.

3.3 Literature Review

The study of classical models for random graphs or network dates back to Erdős and Rényi with the series of papers [ER59, ER60, ER61]. But the first attempt to model a random network to explain real-world phenomena is observed in 1998 by Watts and Strogatz [WS98]. Their random model represents the Milgram's [Mil67] 'small-world' properties which are highly clustered and having a small diameter in social networks. In recent empirical studies suggests that the real-world complex networks mostly demonstrate the 'scale-free' attributes. For example, the degree distributions of the *Internet router networks* studied by Faloutsos, Faloutsos and Faloutsos [FFF99] and the *telephone call networks* studied by Aiello et al. [ACL00] both follows power-laws. Since then, different network models with power-law degree distributions and with other structural features have been proposed to duplicate the scale-free phenomena in the real-world complex networks.

In 1999, Barabási and Albert [BA99] proposed a random model with preferential attachment trait. Later, in 2003, Bollobás et al. [BR03] presented a rigorous proof for the power-law degree distribution for this model. Preferential attachment models with adjustable parameter investigate by Dorogovtsev et al. [DMS00], Aielllo et al. [ACL01] and Jordan [Jor06]. In 2003, Cooper et al. [CF03] proposed a more generalized form of preferential attachment model which removes the restrictions on the creation of edges between two existing vertices and the number of new edges adding to the network.

Beside the preferential attachment models, Kumar et al. [KRR$^+$00] proposed a random network model, called *copying model*, to capture the link copying property in creating a new web-page in the world-wide-web network. They showed that the degree distribution in this model follows the scale-free property.

A random model for the complex network with a well defined graph structural property was proposed in [Gao09], which is called *k-Tree random model*. Later, Sridharan et al. [SGWN11] showed that the edge embeddedness of k-tree random network follows a power-law distribution.

In 2015, Ciotti et al.[CBC$^+$15] proposed two models for signed social networks: *binomial degree distribution model* and *power-law degree distribu-*

tion model. Both of these models follow two main steps to produce a signed random network. In the first step, models generate an unsigned network in which the (unsigned) degree follow either binomial or power-law degree distribution. In the second step, determine sign to each edge of the simulated network by dividing the vertices into two groups. If the endpoints of an edge are in the same group then label this edge as a positive edge, otherwise; if the endpoints of an edge are in different groups then label this edge as a negative edge. Ciotti et al.[CBC$^+$15] showed that the positive and negative degree distributions in the simulated networks obey power-law. But they did not present any analytical proof for the degree distribution in these models. The main features of these models are that the generated signed networks are structurally balanced, and that the each vertex is a member of one of the two mutually exclusive groups.

3.4 Model A: Preferential Attachment Model

3.4.1 Model Definition

The random process start with a signed directed initial network $G_0^k = (V_0, E_0)$ of size $|V_0| = 2k+1$. Suppose there exist exactly k positive and k negative directed edges in G_0^k. At time step $t+1$, we add a new vertex v_{t+1} to construct G_{t+1}^k. The new vertex v_{t+1} connects with k existing vertices from G_t^k as their *positive-out-neighbor* with the probability

$$\mathbb{P}[v_{t+1} \text{ is positive-out-neighbor of } v] = \frac{2d_{G_t^k}^{+out}(v)}{\sum_v d_{G_t^k}^{+}(v)}; \tag{3.1}$$

where $v \in V_t$. Also, v_{t+1} connects with k existing vertices as their *negative-out-neighbor* with the probability

$$\mathbb{P}[v_{t+1} \text{ is negative-out-neighbor of } v] = \frac{2d_{G_t^k}^{-out}(v)}{\sum_v d_{G_t^k}^{-}(v)}; \tag{3.2}$$

where $v \in V_t$. On the other hand, the new vertex v_{t+1} also connects with k existing vertices as their *positive-in-neighbor* and with k existing vertices as their *negative-in-neighbor* with the following probabilities.

$$\mathbb{P}[v_{t+1} \text{ is positive-in-neighbor of } v] = \frac{2d_{G_t^k}^{+in}(v)}{\sum_v d_{G_t^k}^{+}(v)}; \tag{3.3}$$

$$\mathbb{P}[v_{t+1} \text{ is negative-in-neighbor of } v] = \frac{2d^{-in}_{G^k_t}(v)}{\sum_v d^-_{G^k_t}(v)}; \tag{3.4}$$

where $v \in V_t$.

3.4.2 Degree Dynamics

In this section, we investigate the degree dynamics in $G^k_{t\geq 1}$, which we simply denote as G^k_t. That is, we ignore the initial network G^k_0. Let the random model A for signed directed network generates the σ-algebra which can be denoted by $\mathcal{F}_t = \sigma(G^k_t, t \geq 1)$.

First, we analyze the dynamics of the positive-in-degree in G^k_t. Let, $X^{+in}_d(t)$ be the random variable for the number of vertices with positive-in-degree d in G^k_t. By the following lemma, we find the minimum positive-in-degree for a vertex and the total positive-degree (in and out) in G^k_t.

Lemma 3.1. *For any vertex $v \in V_t$, positive-in-degree of v, $d^{+in}_{G^k_t}(v) \geq k$ and the total positive-degree in G^k_t is $\sum_v d_{G^k_t}(v) = 4kt$.*

Proof. By ignoring the initial vertices, when a new vertex enters to the network, it selects k existing vertices as its positive-in-neighbors. Therefore, each vertex enters in the network with exactly k positive-in-degrees, i.e. $d^{+in}_{G^k_t}(v) \geq k$, for all $v \in V_{t\geq 1}$.

At time t, the new entering vertex adds k additional positive edges in the both directions with respect to itself. That is, in total $2k$ new positive edges are added at the time when the new vertex is entering to the existing network G^k_t. So, at the end of the time t, the total number of the positive degrees in G^k_t is increased by $4k$. Therefore, if we ignore the initial network, the total number of positive degrees in G^k_t is $\sum_v d_{G^k_t}(v) = 4kt$. □

According to the model construction, at time $t+1$, an existing vertex $v \in V_t$ can only increase its positive-in-degree if the new vertex v_{t+1} connects as a positive-in-neighbor of the vertex v. Therefore, the probability of a vertex $v \in V$ receives a positive-in-degree is (using Eq.(3.3))

$$\mathbb{P}[v \text{ receives a positive-in-degree}] = \frac{2\, d^{+in}_{G^k_t}(v)}{\sum_{v\in V_t} d^+_{G^k_t}(v)}. \tag{3.5}$$

Again, for given $d^{+in}_{G^k_t}(v) = d$, the conditional probability that $v \in V_t$ receives a positive-in-degree is (using Eq.(3.5))

$$\mathbb{P}[v \text{ receives a positive-in-degree}|d^{+in}_{G^k_t}(v) = d] = \frac{2d}{\sum_{v \in V_t} d^{+}_{G^k_t}(v)}. \tag{3.6}$$

Since, v_{t+1} connects as positive-in-neighbor with k existing vertices, then for given G^k_t, by using Eq.(3.6) and *Lemma 3.1*, the expected number of vertices with positive-in-degree d that receive a positive-in-degree in G^k_{t+1} is

$$\frac{2kd}{4kt} X^{+in}_d(t) = \frac{d}{2t} X^{+in}_d(t) \tag{3.7}$$

which is independent of k.

Let $\{\beta^{+in}_d\}$ be a sequence of positive integers. We now show that, $\left|\mathbb{E}\left[X^{+in}_d(t)\right] - t\beta^{+in}_d\right|$ is asymptotically bounded by a constant, where $\{\beta^{+in}_d\}$ satisfies the following equations

$$\beta^{+in}_d = \frac{d-1}{d+2}\beta^{+in}_{d-1}, \qquad \beta^{+in}_k \approx 1, \tag{3.8}$$

as $t \to \infty$.

Theorem 3.2. *Let $\mathbb{E}\left[X^{+in}_d(t)\right]$ be the expected number of vertices with positive-in-degree d in the random network G^k_t generated by the Model A. Then*

$$\left|\mathbb{E}\left[X^{+in}_d(t)\right] - \beta^{+in}_d t\right| \leq C,$$

where C is a constant and β^{+in}_d has a power-law bound d^{-3}.

Proof. First, consider the base case $d = k$. If we ignore the initial vertices, then according to the *Lemma 3.1*, any vertex $v \in V_t$ has at least k positive-in-degree in G^k_t, i.e. $d^{+in}_{G^k_t}(v) = k;\ \forall\ v \in V_t$. That is, if $v \in V_t$ connects with v_{t+1} as a positive-out-neighbor, then $d^{+in}_{G^k_{t+1}}(v) = k+1$. Therefore, from Eq.(3.7), for given G^k_t, the expected number of vertices with positive-in-degree $(k+1)$ in G^k_{t+1} and k in G^k_t is

$$\frac{d}{2t} X^{+in}_k. \tag{3.9}$$

Again, at the time step $t+1$, the new vertex v_{t+1} has exactly k-positive-in-degree and k-positive-out-degree in G^k_{t+1}. Therefore, for given the value of X_k^{+in}, the net difference in the number of vertices with positive-in-degree k from G_t^k to G_{t+1}^k is

$$\mathbb{E}\big[X_k^{+in}(t+1)\big|\mathcal{F}_t\big] - X_k^{+in}(t) = 1 - \frac{k}{2t}X_k^{+in}. \tag{3.10}$$

By taking expectation on the both side of the Eq.(3.10), we get

$$\begin{aligned}\mathbb{E}\big[X_k^{+in}(t+1)\big] &= 1 - \frac{k}{2t}\mathbb{E}\big[X_k^{+in}(t)\big] + \mathbb{E}\big[X_k^{+in}(t)\big],\\ &\approx \mathbb{E}\big[X_k^{+in}(t)\big] + 1,\end{aligned} \tag{3.11}$$

as $t \to \infty$.

According to the model structure, at time t, exactly one vertex enters to the network with positive-in-degree k. Now, if we ignore the initial network, then $\mathbb{E}\big[X_k^{+in}(t)\big] = 1$ at $t = 1$. Then, by solving the above recurrence Eq.(3.11), we get

$$\mathbb{E}\big[X_k^{+in}(t)\big] = t + \mathcal{O}(1). \tag{3.12}$$

Now, consider the general case $d > k$. Due to the model construction, we have to consider two cases to estimate the difference in the number of vertices with positive-in-degree d during the transition from G_t^k to G_{t+1}^k. First, if the new vertex v_{t+1} connects with $v \in V_t$ as a positive-in-neighbor, and if $d_{G_t^k}^{+in}(v) = d - 1$, then positive-in-degree of v increases to d in G_{t+1}^k. Therefore, from Eq.(3.7) for given G_t^k, the expected number of vertices with positive-in-degree d in G_{t+1}^k and $(d-1)$ in G_t^k is

$$\frac{d-1}{2t}X_{d-1}^{+in}. \tag{3.13}$$

Second, if v_{t+1} connects with $v \in V_t$ as a positive-in-neighbor, and if $d_{G_t^k}^{+in}(v) = d$, then the positive-in-degree of v increases to $(d+1)$ in G_{t+1}^k. Therefore, again from Eq.(3.7), for given G_t^k, the expected number of vertices with positive-in-degree $(d+1)$ in G_{t+1}^k and d in G_t^k is

$$\frac{d}{2t}X_d^{+in}. \tag{3.14}$$

Therefore, by using Eq.s (3.13) and (3.14), for given the value of X_d^{+in}, the net difference in the number of vertices with positive-in-degree d from G_t^k to G_{t+1}^k is

$$\mathbb{E}\big[X_d^{+in}(t+1)\big|\mathcal{F}_t\big] - X_d^{+in}(t) = \frac{d-1}{2t}X_{d-1}^{+in} - \frac{d}{2t}X_d^{+in}. \tag{3.15}$$

By taking expectation on the both side of the Eq.(3.15), we get

$$\mathbb{E}\big[X_d^{+in}(t+1)\big] = \mathbb{E}\big[X_d^{+in}(t)\big] + \frac{d-1}{2t}\mathbb{E}\big[X_{d-1}^{+in}(t)\big] - \frac{d}{2t}\mathbb{E}\big[X_d^{+in}(t)\big]. \quad (3.16)$$

Assume $\beta_d^{+in} = 0$ for $d < k$ in $\{\beta_d^{+in}\}$. Let $\mathbb{E}\big[X_d^{+in}(t)\big]$ can be approximated by $t\beta_d^{+in}$. Then, Eq.(3.16) satisfies

$$\begin{aligned}(t+1)\beta_d^{+in} &= t\beta_d^{+in} + \frac{1}{2t}((d-1)t\beta_{d-1}^{+in} - dt\beta_d^{+in}),\\ \beta_d^{+in} &= \frac{d-1}{d+2}\beta_{d-1}^{+in}.\end{aligned} \quad (3.17)$$

Again, from the Eq.(3.11), as $t \to \infty$, we have

$$\begin{aligned}(t+1)\beta_k^{+in} &= 1 + t\beta_k^{+in},\\ \beta_k^{+in} &= 1.\end{aligned} \quad (3.18)$$

Here, Eq.s(3.17) and (3.18) give the recurrence equations, which are satisfied by the sequence $\{\beta_d^{+in}\}$.

Let $\Delta_d^{+in}(t) = \mathbb{E}\big[X_d^{+in}(t)\big] - t\beta_d^{+in}$. To show that, $\mathbb{E}\big[X_d^{+in}(t)\big]$ can be approximated by $t\beta_d^{+in}$, we need to proof by induction that $\big|\Delta_d^{+in}(t)\big|$ is bounded by a constant.

For the base case $d = k$, from the Eq.s(3.12) and (3.18), we get

$$\begin{aligned}\big|\Delta_k^{+in}(t)\big| + t\beta_k^{+in} &= t + \mathcal{O}(1),\\ \big|\Delta_k^{+in}(t)\big| &= \mathcal{O}(1),\end{aligned} \quad (3.19)$$

i.e. $\big|\Delta_k^{+in}(t)\big|$ is bounded by a constant which is independent of d and t.

Consider, $\big|\Delta_d^{+in}(t)\big|$ is also bounded by a constant which is independent of d and t. Thus, we get

$$\big|\Delta_d^{+in}(t)\big| = \big|\mathbb{E}\big[X_d^{+in}(t)\big] - t\beta_d^{+in}\big| \leq \mathcal{O}(1). \quad (3.20)$$

Now, from the Eq.(3.16), we get

$$\begin{aligned}\Delta_d^{+in}(t+1) + (t+1)\beta_d^{+in} = \Delta_d^{+in}(t) + t\beta_d^{+in} + \frac{1}{2t}\big((d-1)(\Delta_{d-1}^{+in}(t)\\ + t\beta_{d-1}^{+in}) - d(\Delta_d^{+in}(t) + t\beta_d^{+in})\big).\end{aligned}$$

By rearranging the above equation, we have

$$\Delta_d^{+in}(t+1) = \frac{d-1}{2t}\Delta_{d-1}^{+in}(t) + \big(1 - \frac{d}{2t}\big)\Delta_d^{+in}(t) + \frac{d-1}{2t}\beta_{d-1}^{+in} - \big(1 + \frac{d}{2}\big)\beta_d^{+in}. \tag{3.21}$$

Now, using the definition of β_d^{+in} from the Eq.(3.17), we get

$$\begin{aligned}\frac{d-1}{2}\beta_{d-1}^{+in} - \big(1+\frac{d}{2}\big)\beta_d^{+in} &= \frac{d-1}{2}\beta_{d-1}^{+in} - \frac{d+2}{2}\beta_d^{+in}\\ &= \frac{d-1}{2}\,\frac{d+2}{d-1}\beta_d^{+in} - \frac{d+2}{2}\beta_d^{+in}\\ &= 0.\end{aligned} \tag{3.22}$$

Then, from the Eq.s (3.21) and (3.22), we get

$$\begin{aligned}\big|\Delta_d^{+in}(t+1)\big| &\le \frac{d-1}{2t}\big|\Delta_{d-1}^{+in}(t)\big| + (1-\frac{d}{2t})\big|\Delta_d^{+in}(t)\big|,\\ &\le \left(\frac{d-1}{2t} + \frac{2t-d}{2t}\right) \max\big(\big|\Delta_{d-1}^{+in}(t)\big|, \big|\Delta_d^{+in}(t)\big|\big),\\ &= \left(1 - \frac{1}{2t}\right) \max\big(\big|\Delta_{d-1}^{+in}(t)\big|, \big|\Delta_d^{+in}(t)\big|\big),\end{aligned} \tag{3.23}$$

Therefor, as $t \to \infty$ then, we get from the Eq.s (3.19), (3.20) and (3.23)

$$\big|\Delta_d^{+in}(t+1)\big| = \big|\mathbb{E}\big[X_d^{+in}(t+1)\big] - (t+1)\beta_d^{+in}\big| \le O(1), \tag{3.24}$$

i.e., $\big|\Delta_d^{+in}(t+1)\big|$ is bounded by a constant which is independent of d and t.

Hence, by using the induction hypothesis we can say that, $\big|\mathbb{E}\big[X_d^{+in}(t)\big] - t\beta_d^{+in}\big|$ is bounded by a constant which is independent of d and t, i.e. $\mathbb{E}\big[X_d^{+in}(t)\big]$ can be approximated by $t\beta_d^{+in}$.

To find the power-law bound for the positive-in-degree distribution in the signed directed networks generated by the Model A, we get from the definition of β_d^{+in} at Eq.(3.17), that

$$\beta_d^{+in} = \prod_{i=1}^{d} \frac{i-1}{i+2} \approx d^{-3}, \tag{3.25}$$

which is independent of k. □

Similarly, to find the power-law bound for the positive-out-degree distribution, let $\{\beta_d^{+out}\}$ be a sequence of the positive integers. Then by following the above procedure for positive-out-degree, we can prove the following theorem.

Theorem 3.3. *Let $\mathbb{E}\big[X_d^{+out}(t)\big]$ be the expected number of vertices with positive-out-degree d in the random network G_t^k generated by the Model A. Then*

$$\big|\mathbb{E}\big[X_d^{+out}(t)\big] - \beta_d^{+out} t\big| \leq C,$$

where C is a constant and β_d^{+out} has a power-law bound d^{-3}.

In similar manner, let $\{\beta_d^{-in}\}$ and $\{\beta_d^{-out}\}$ be two sequence of positive integers. Then we can also prove the following theorems for negative-in-degree and negative-out-degree distributions respectively.

Theorem 3.4. *Let $\mathbb{E}\big[X_d^{-in}(t)\big]$ be the expected number of vertices with negative-in-degree d in the random network G_t^k generated by the Model A. Then*

$$\big|\mathbb{E}\big[X_d^{-in}(t)\big] - \beta_d^{-in} t\big| \leq C,$$

where C is a constant and β_d^{-in} has a power-law bound d^{-3}.

Theorem 3.5. *Let $\mathbb{E}\big[X_d^{-out}(t)\big]$ be the expected number of vertices with negative-out-degree d in the random network G_t^k generated by the Model A. Then*

$$\big|\mathbb{E}\big[X_d^{-out}(t)\big] - \beta_d^{-out} t\big| \leq C,$$

where C is a constant and β_d^{-out} has a power-law bound d^{-3}.

3.5 Model B: Edge Copying Model

3.5.1 Model Definition

Initially at $t = 0$, we start with a initial signed directed network G_0^k with the vertex set of size $|V_0| = 2k + 1$. The initial network G_0^k is connected by exactly k positive and k negative directed edges in such way that, there exist exactly k number of vertices having each type of degrees: positive-in-degree, positive-out-degree, negative -in-degree, and negative-out-degree.

At the time $t+1$, the new vertex v_{t+1} is added to the network to construct G_{t+1}^k. The vertex v_{t+1} connects with k existing vertices by copying k distinct edges from G_t^k. The copy procedure obeys the following steps:

(1) Selects a distinct (i.e. not copied already) signed-directed-edge e from G_t^k uniformly at random (u.a.r). Let $e[src]$, $e[trg]$ and $e[sign]$ are the e's source, target vertices and e's sign respectively.

(2) Add a new signed-directed-edge

   (a) from v_{t+1} to $e[trg]$ with probability $(1-\alpha)$, by copying e's sign and source $e[src]$ vertex.

   (b) or, from $e[src]$ to v_{t+1} with probability α, by copying e's sign and target $e[trg]$ vertex.

(3) Repeat the above procedure k times.

3.5.2 Comparison with Preferential Attachment Model

Before studying the degree dynamics in the signed directed network generated by the edge copying model, first we analyze following facts.

According to the edge copying model, at time $t+1$, the new vertex v_{t+1} selects uniformly at random an edge, which to be copied, from G_t^k. Regardless the signs and directions, at each time step, exactly k new edges are added to the network. Therefore, the total number of edges in G_t^k is

$$\left|E_t\right| = kt + 2k \approx kt, \qquad \text{(for large } t\text{)} \tag{3.26}$$

where E_t is the set of all edges in G_t^k and $\left|E_0\right| = 2k$ is the number of edges in initial network G_0^k.

Again, according to the model construction, the new vertex v_{t+1} adds k new edges by randomly copying k vertices (either the source or target) from the selected k distinct edges in the existing network. Therefore, any vertex v can only receives a positive-in-degree if v is the target vertex of the selected edge $(v', v) \in E_t^+$ for copying in which the source vertex v' is copied by the v_{t+1}. That is, if v_{t+1} connects with v by the edge (v_{t+1}, v), such that $(v_{t+1}, v) \in E_t^+$, $(v', v) \in E_t^+$ and v' is copied by v_{t+1}, then v receives a positive-in-degree.

Also, we know that the number of positive-directed-edges, in which $v \in V_t$ is the target vertex, is equal to the positive-in-degree of v in G_t^k, i.e. $d^{+in}_{G_t^k}(v)$. Now, v receives a positive-in-degree only if v_{t+1} copied the source vertex from the selected positive-directed-edge e, in which v is the target vertex. Therefore, in the process of copying k edges from the existing network, there is a chance of copying one more edge in which $v \in V_t$ is the target vertex. That is, v may receive more than one positive-in-degree in the transition from G_t^k to G_{t+1}^k.

Therefore, the probability that the vertex $v \in V_t$ receives exactly l positive-in-degree in time $t+1$ is

$$\mathbb{P}[v \text{ receives exactly } l \text{ positive-in-degree}] = \alpha \frac{\binom{d^{+in}_{G_t^k}(v)}{l}\binom{kt-d^{+in}_{G_t^k}(v)}{k-l}}{\binom{kt}{k}}, \tag{3.27}$$

where $e \in E_t^+$, $l \geq 1$. In the Eq.(3.27), the term α is the probability of copying the source vertex for the selected edge (i.e. connecting v_{t+1} with the target vertex) and the fractional term is the probability of selecting l edges in which v is the target vertex of e.

For given $d^{+in}_{G_t^k}(v) = d$, the conditional probability that a vertex $v \in \mathcal{V}_t$ receives exactly l positive-in-degree in time $t+1$ is

$$\mathbb{P}[v \text{ receives exactly } l \text{ positive-in-degree } |d^{+in}_{G_t^k}(v) = d] = \alpha \, \frac{\binom{d}{l}\binom{kt-d}{k-l}}{\binom{kt}{k}}, \tag{3.28}$$

which is dependent on t, d and k.

At this point, if we look at the above probability, then it is very unclear to have any preferential attachment property.

3.5.3 Notations

We introduce the following parameters:

$$a_k = 1 - \frac{k-1}{kt} \to 1 \text{ as } t \to \infty, \tag{3.29}$$

$$b_d = \prod_{i=0}^{k-2} \big(1 - \frac{d}{kt-i}\big) \to 1 \text{ as } t \to \infty, \tag{3.30}$$

$$b_{d-1} = \prod_{i=0}^{k-2} \big(1 - \frac{d-1}{kt-i}\big) \to 1 \text{ as } t \to \infty. \tag{3.31}$$

3.5.4 Degree Dynamics

Before investigating the degree dynamics, first, we analyze the evolution of the number of positive and negative edges in G_t^k. Consider the edge copying model for signed directed networks generates the σ-algebra which is denoted by $\mathcal{F}_t = \sigma(G_t^k, t \geq 1)$.

Let, $X_e^+(t)$ and $X_e^-(t)$ are the random variables for the number of positive and negative edges in G_t^k respectively. According to the model construction, there exist exactly k positive and k negative edges in the initial network G_0^k.

At time t, the new vertex adds k edges (both positive and negative) to the network by copying k distinct edges from the existing network. That is, a positive edge will add to network if the new vertex select a positive existing edge. Therefore, for given the value of $X_e^+(t)$, the expected number of newly added positive edges in G_{t+1}^k is

$$\frac{k}{|E_t|} X_e^+(t). \tag{3.32}$$

Therefore, the net difference in the number of positive edges from G_t^k to G_{t+1}^k is

$$\mathbb{E}\big[X_e^+(t+1)\big|\mathcal{F}_t\big] - X_e^+(t) = \frac{k}{|E_t|} X_e^+(t). \tag{3.33}$$

By taking mathematical expectation on the both side of the Eq.(3.33), we get (using Eq.(3.26))

$$\mathbb{E}\big[X_e^+(t+1)\big] = \big(1 + \frac{k}{kt+2k}\big)\mathbb{E}\big[X_e^+(t)\big] = \frac{t+3}{t+2}\mathbb{E}\big[X_e^+(t)\big]. \tag{3.34}$$

Therefore, we can write

$$\mathbb{E}\big[X_e^+(t)\big] = \frac{t+2}{t+1}\mathbb{E}\big[X_e^+(t-1)\big]. \tag{3.35}$$

Since, $\mathbb{E}\big[X_e^+(0)\big] = k$, by solving the Eq.(3.35), we get (using Eq.(3.26))

$$\mathbb{E}\big[X_e^+(t)\big] = \frac{t+2}{2} k = \frac{|E_t|}{2}. \tag{3.36}$$

Now, we focus on analyzing the dynamic of the positive-in-degree distribution in G_t^k.

Let, $X_d^{+in}(t)$ be random variable for the number of vertices with positive-in-degree d in G_d^k generated by edge copying model. Now, we prove the following lemmas for calculating the expected number of vertices with positive-in-degree d in G_{t+1}^k.

Lemma 3.6. *For each $d \leq k$, the expected number of vertices with positive-in-degree d in G^k_{t+1} satisfies*

$$\mathbb{E}\big[X^{+in}_d(t+1)\big] \approx \mathcal{O}(1) + \frac{(d-1)\alpha}{ta_k} b_{d-1}\, \mathbb{E}[X^{+in}_{d-1}(t)] + (1 - \frac{d\alpha}{ta_k} b_d)\, \mathbb{E}[X^{+in}_d(t)] + \sum_{l=2}^{d} \mathcal{O}(t^{-l});$$

where $\mathbb{E}\big[X^{+in}_{d-1}(t)\big]$ and $\mathbb{E}\big[X^{+in}_d(t)\big]$ are the expected number of vertices with positive-in-degree $d-1$ and d in G^k_t respectively. Also, b_d and b_{d-1} are defined in Eq's(3.30) and (3.31) respectively.

Proof. For the case $d \leq k$, a vertex $v \in V_t$ may receive at most k positive-in-degree in the transition G^k_t to G^k_{t+1}. To find the expected number of positive-in-degree d in G^k_{t+1}, in this case, we have to consider the following three situations.

The first situation is, if $v \in V_t$, with $d^{+in}_{G^k_t}(v) = d - l$; $1 \leq l \leq d$, receives exactly l positive-in-degree in G^k_{t+1}. Then, for given G^k_t, the expected number of vertices with positive-in-degree $d-l$ in G^k_t and d in G^k_{t+1} is (by using Eq.s(3.26) and (3.28))

$$\begin{aligned}
&\sum_{l=1}^{d} \alpha\, \frac{\binom{d-l}{l}\binom{kt-d+l}{k-l}}{\binom{kt}{k}}\, X^{+in}_{d-l}(t) \\
&= \alpha\, \frac{\binom{d-1}{1}\binom{kt-d+1}{k-1}}{\binom{kt}{k}}\, X^{+in}_{d-1}(t) + \sum_{l=2}^{d} \alpha\, \frac{\binom{d-l}{l}\binom{kt-d+l}{k-l}}{\binom{kt}{k}}\, X^{+in}_{d-l}(t) \\
&= \frac{\alpha(d-1)}{t - \frac{k-1}{k}} \prod_{i=0}^{k-2} (1 - \frac{d-1}{kt-i}) X^{+in}_{d-1}(t) \\
&\quad + \sum_{l=2}^{d} \alpha \prod_{i=0}^{l-1} \left(\frac{(d-l-i)(k-i)}{(l-i)(kt-i)} \right) \prod_{i=0}^{k-l-1} (1 - \frac{d-l}{kt-i}) X^{+in}_{d-l}(t) \\
&= \frac{\alpha(d-1)}{ta_k} b_{d-1} X^{+in}_{d-1}(t) + \sum_{l=2}^{d} \mathcal{O}(t^{-l}) X^{+in}_{d-l}(t);
\end{aligned} \tag{3.37}$$

where $a_k = 1 - \frac{k-1}{kt}$ and $b_{d-1} = \prod\limits_{i=0}^{k-2} (1 - \frac{d-1}{kt-i})$.

The second situation is, if $v \in V_t$, with $d^{+in}_{G^k_t}(v) = d$; $1 \leq l \leq d$, receives exactly l positive-in-degree in G^k_{t+1}. Then, for given G^k_t, the expected number of vertices with positive-in-degree d in G_t^k and $d+l$ in G_{t+1}^k is

$$\begin{aligned}
&\sum_{l=1}^{d} \alpha \, \frac{\binom{d}{l}\binom{kt-d}{k-l}}{\binom{kt}{k}} \, X_d^{+in}(t) \\
&= \alpha \, \frac{\binom{d}{1}\binom{kt-d}{k-1}}{\binom{kt}{k}} \, X_{d-1}^{+in}(t) + \sum_{l=2}^{d} \alpha \, \frac{\binom{d}{l}\binom{kt-d}{k-l}}{\binom{kt}{k}} \, X_d^{+in}(t) \\
&= \frac{\alpha d}{t-\frac{k-1}{k}} \prod_{i=0}^{k-2} (1 - \frac{d}{kt-i}) X_d^{+in}(t) \\
&\quad + \sum_{l=2}^{d} \alpha \prod_{i=0}^{l-1} \left(\frac{(d-i)(k-i)}{(l-i)(kt-i)} \right) \prod_{i=0}^{k-l-1} (1 - \frac{d}{kt-i}) X_d^{+in}(t) \\
&= \frac{\alpha d}{t a_k} b_d X_d^{+in}(t) + \sum_{l=2}^{d} \mathcal{O}(t^{-l}) X_d^{+in}(t); \qquad (3.38)
\end{aligned}$$

where $a_k = 1 - \frac{k-1}{kt}$ and $b_d = \prod\limits_{i=0}^{k-2} (1 - \frac{d}{kt-i})$.

The third situation is, whether the new vertex v_{t+1} has positive-in-degree d in G_{t+1}^k or not. The vertex v_{t+1} can achieve a positive-in-degree in G_{t+1}^k if the random process selects an edge $(v_i, v_j) \in E_t^+$ in which the target vertex v_j is selected for copying. That is, v_{t+1} receives a positive-in-degree in G_{t+1}^k, if v_{t+1} connects with the source vertex v_i from the randomly selected edge $(v_i, v_j) \in E_t^+$ by the new edge $(v_i, v_{t+1}) \in E_{t+1}^+$. Therefore, according to the model construction, the probability that v_{t+1} receives a positive-in-degree in G_{t+1}^k is

$$(1-\alpha) \frac{|E_t^+|}{|E_t|}, \qquad (3.39)$$

where E_t^+ and E_t are the sets of all positive-edges and all edges in G_t^k respectively.

Since, v_{t+1} has k neighbors in G_{t+1}^k, then we can write the expectation of the event that the vertex v_{t+1} has exactly d positive-in-degree in G_{t+1}^k, where $1 \le d \le k$, as

$$\begin{aligned}
\mathbb{E}\big[I_d^k(v_{t+1}) \big| \mathcal{F}_t\big] &= \binom{k}{d} \left(1 - \frac{(1-\alpha)\big|E_t^+\big|}{\big|E_t\big|}\right)^{k-d} \left(\frac{(1-\alpha)\big|E_t^+\big|}{\big|E_t\big|}\right)^{d} \\
&= \binom{k}{d} \left(\frac{\big(|E_t| - (1-\alpha)\, |E_t^+|\big)^{k-d}}{|E_t|^k} \right) (1-\alpha)^d \, |E_t^+|^d
\end{aligned}$$

$$\leq \binom{k}{d}\left(\frac{|E_t|^{k-d} + (1-\alpha)^{k-d}|E_t^+|^{k-d}}{|E_t|^k}\right)(1-\alpha)^d\,|E_t^+|^d$$
$$= \binom{k}{d}\left((1-\alpha)^d\,\frac{|E_t^+|^d}{|E_t|^d} + (1-\alpha)^k\,\frac{|E_t^+|^k}{|E_t|^k}\right); \qquad (3.40)$$

where $I_d^k(v_{t+1})$ is the indicator function of the event that vertex v_{t+1} has positive-in-degree d, such that $1 \leq d \leq k$, in G_{t+1}^k.
Since, $\mathbb{E}[X_e^+(t)]$ be the expected number of positive edges in G_t^k, then we get (using Eq.(3.36))

$$|E_t^+| \approx \mathbb{E}[X_e^+(t)] = \frac{|E_t|}{2}. \qquad (3.41)$$

Then, from Eq.s(3.40) and (3.41), we get

$$\mathbb{E}\big[I_d^k(v_{t+1})\big|\mathcal{F}_t\big] \leq \binom{k}{d}\left(\frac{1}{2^d}(1-\alpha)^d + \frac{1}{2^k}(1-\alpha)^k\right) = I_M. \quad \text{(let)}, \qquad (3.42)$$

where $1 \leq d \leq k$. Here, I_M, which is independent of t and equals to zero for $d > k$, is constant for a random process.

Therefore, by using Eq.s(3.37), (3.38) and (3.42), for given the value of $X_d^{+in}(t)$, the net difference in the number of vertices with positive-in-degree d from G_t^k to G_{t+1}^k can be approximated as (after rearranging)

$$\mathbb{E}\big[X_d^{+in}(t+1)\big|\mathcal{F}_t\big] - X_d^{+in}(t) \approx I_M + \frac{(d-1)\alpha}{ta_k}b_{d-1}X_{d-1}^{+in}(t)$$
$$- \frac{d\alpha}{ta_k}b_d X_d^{+in}(t) + \sum_{l=2}^{d}\mathcal{O}(t^{-l}). \qquad (3.43)$$

By taking mathematical expectation on the both side of Eq.(3.43), we get

$$\mathbb{E}\big[X_d^{+in}(t+1)\big] \approx I_M + \frac{(d-1)\alpha}{ta_k}b_{d-1}\mathbb{E}[X_{d-1}^{+in}(t)]$$
$$+ \big(1 - \frac{d\alpha}{ta_k}b_d\big)\mathbb{E}[X_d^{+in}(t)] + \sum_{l=2}^{d}\mathcal{O}(t^{-l}); \qquad (3.44)$$

where $d \leq k$ and I_M is a constant which is independent of t. $\square$

Lemma 3.7. *For each $d > k$, the expected number of vertices with positive-in-degree d in G^k_{t+1} satisfies*

$$\mathbb{E}\big[X_d^{+in}(t+1)\big] = \frac{(d-1)\alpha}{ta_k} b_{d-1}\mathbb{E}[X_{d-1}^{+in}(t)] + \big(1 - \frac{d\alpha}{ta_k} b_d\big)\mathbb{E}[X_d^{+in}(t)] + \sum_{l=2}^{k} \mathcal{O}(t^{-l}),$$

where $\mathbb{E}\big[X_{d-1}^{+in}(t)\big]$ and $\mathbb{E}\big[X_d^{+in}(t)\big]$ are the expected number of vertices with positive-in-degree $d-1$ and d in G^k_t respectively. Also, b_d and b_{d-1} are defined in Eq's(3.30) and (3.31) respectively.

Proof. For the case $d > k$, a vertex $v \in V_t$ may receive at most k positive-in-degree in the transition from G^k_t to G^k_{t+1}. To find the expected number of positive-in-degree d in G^k_{t+1}, in this case, we have to consider the following two situations.

The first situation is, if $v \in V_t$, with $d^{+in}_{G^k_t}(v) = d - l$; $1 \le l \le k$, receives exactly l positive-in-degree in G^k_{t+1}. Then, for give G^k_t, the expected number of vertices with positive-in-degree $d - l$ in G^k_t and d in G^k_{t+1} is

$$\begin{aligned} &\sum_{l=1}^{k} \alpha \, \frac{\binom{d-l}{l}\binom{kt-d+l}{k-l}}{\binom{kt}{k}} \, X_{d-l}^{+in}(t) \\ &= \frac{\alpha(d-1)}{ta_k} b_{d-1} X_{d-1}^{+in}(t) + \sum_{l=2}^{k} \mathcal{O}(t^{-l}) X_{d-l}^{+in}(t), \end{aligned} \tag{3.45}$$

where $a_k = 1 - \frac{k-1}{kt}$ and $b_{d-1} = \prod_{i=0}^{k-2}(1 - \frac{d-1}{kt-i})$.

The second situation is, if $v \in V_t$, with $d^{+in}_{G^k_t}(v) = d$; $1 \le l \le k$, receives exactly l positive-in-degree in G^k_{t+1}. Then, for given G^k_t, the expected number of vertices with positive-in-degree d in G^k_t and $d + l$ in G^k_{t+1} is

$$\begin{aligned} &\sum_{l=1}^{k} \alpha \, \frac{\binom{d}{l}\binom{kt-d}{k-l}}{\binom{kt}{k}} \, X_{d}^{+in}(t) \\ &= \frac{\alpha d}{ta_k} b_d X_d^{+in}(t) + \sum_{l=2}^{k} \mathcal{O}(t^{-l}) X_d^{+in}(t), \end{aligned} \tag{3.46}$$

where $a_k = 1 - \frac{k-1}{kt}$ and $b_d = \prod_{i=0}^{k-2}(1 - \frac{d}{kt-i})$.

Therefore, by using the Eq.s(3.45) and (3.46), for given the value of $X_d^{+in}(t)$, the net difference in the number of vertices with positive-in-degree d from G_t^k to G_{t+1}^k is (after rearranging)

$$\mathbb{E}\big[X_d^{+in}(t+1)\big|\mathcal{F}_t\big] - X_d^{+in}(t) = \frac{(d-1)\alpha}{ta_k} b_{d-1} X_{d-1}^{+in}(t) - \frac{d\alpha}{ta_k} b_d X_d^{+in}(t) + \sum_{l=2}^{k} \mathcal{O}(t^{-l}). \tag{3.47}$$

By taking mathematical expectation on the both side of the Eq.(3.43), we get

$$\mathbb{E}\big[X_d^{+in}(t+1)\big] = \frac{(d-1)\alpha}{ta_k} b_{d-1} \mathbb{E}[X_{d-1}^{+in}(t)] + \big(1 - \frac{d\alpha}{ta_k} b_d\big)\mathbb{E}[X_d^{+in}(t)] + \sum_{l=2}^{k} \mathcal{O}(t^{-l}) \tag{3.48}$$

where $d > k$. □

Let $\{\beta_d^{+in}\}$ be a sequence of positive integers. In next theorem, we show that, $\big|\mathbb{E}\big[X_d^{+in}(t)\big] - t\beta_d^{+in}\big|$ is asymptotically bounded by a constant where $\{\beta_d^{+in}\}$ satisfies the following recurrence equations

$$\beta_d^{+in} = \frac{d-1}{d+\frac{1}{\alpha}} \beta_{d-1}^{+in}; \quad d > k, \quad \text{and} \quad \beta_k^{+in} \simeq c, \tag{3.49}$$

as $t \to \infty$ and c is a constant.

Theorem 3.8. *Let $\mathbb{E}[X_d^{+in}(t)]$ be the expected number of vertices with positive-in-degree d in G_t^k generated by Model B. If α is the probability of coping source vertex from a randomly selected edge, then*

$$\big|\mathbb{E}[X_d^{+in}(t)] - t\beta_d^{+in}\big| \leq \mathcal{O}(1),$$

where β_d^{+in} has a power-law bound $d^{-\left(1+\frac{1}{\alpha}\right)}$.

Proof. First, we investigate the base case for $d = 1$. In the Eq.(3.44), the second term becomes zero for $d = 1$. Then, the expected number of vertices with positive-in-degree 1(one) in G_{t+1}^k can be approximated as

$$\mathbb{E}\big[X_1^{+in}(t+1)\big] \approx I_M + \big(1 - \frac{\alpha}{ta_k} b_d\big)\mathbb{E}[X_1^{+in}(t)] + \sum_{l=2}^{d} \mathcal{O}(t^{-l}). \tag{3.50}$$

By using $\mathbb{E}\big[X_1^{+in}(t)\big] = k$ at $t = 0$, we get from solving the Eq.(3.50)

$$\mathbb{E}\big[X_1^{+in}(t)\big] \approx I_M t + \mathcal{O}(1). \tag{3.51}$$

Now, we investigate the case $2 \leq d \leq k$. We get from the Eq.(3.44), the expected number of vertices with positive-in-degree d in G_{t+1}^k can be approximated as

$$\begin{aligned}\mathbb{E}\big[X_d^{+in}(t+1)\big] \approx I_M &+ \frac{(d-1)\alpha}{ta_k} b_{d-1}\mathbb{E}[X_{d-1}^{+in}(t)] \\ &+ \big(1 - \frac{d\alpha}{ta_k} b_d\big)\mathbb{E}[X_d^{+in}(t)] + \sum_{l=2}^{d} \mathcal{O}(t^{-l}),\end{aligned} \tag{3.52}$$

where $2 \leq d \leq k$.
Finally, for $d > k$ the expected number of vertices with positive-in-degree d in G_{t+1}^k is (using Eq.(3.48))

$$\begin{aligned}\mathbb{E}\big[X_d^{+in}(t+1)\big] = &\frac{(d-1)\alpha}{ta_k} b_{d-1}\mathbb{E}[X_{d-1}^{+in}(t)] \\ &+ \big(1 - \frac{d\alpha}{ta_k} b_d\big)\mathbb{E}[X_d^{+in}(t)] + \sum_{l=2}^{k} \mathcal{O}(t^{-l}).\end{aligned} \tag{3.53}$$

Assume, $\beta_d^{+in} = 0$ for $d \leq 0$. Let, $\mathbb{E}\big[X_d^{+in}\big]$ can be approximate by $t\beta_d^{+in}$. Then, the Eq.(3.50) satisfies

$$(t+1)\beta_1^{+in} \approx I_M + \big(1 - \frac{\alpha}{ta_k} b_d\big)t\beta_1^{+in} + \sum_{l=2}^{d} \mathcal{O}(t^{-l}),$$

$$\big(1 + \frac{d\alpha b_d}{a_k}\big)\beta_1^{+in} \approx I_M + \sum_{l=2}^{d} \mathcal{O}(t^{-l}).$$

Since, as $t \to \infty$, then $a_k \to 1$, $b_d \to 1$, and $\sum_{l=2}^{d} \mathcal{O}(t^{-l}) \to 0$, and also I_M is a constant. Therefore, from the above equation, we get

$$\beta_1^{+in} \approx I_M; \qquad \text{as } t \to \infty. \tag{3.54}$$

Also, from the Eq.(3.52), we get for $2 \leq d \leq k$

$$\begin{aligned}(t+1)\beta_d^{+in} \approx I_M &+ \frac{(d-1)\alpha}{ta_k} b_{d-1} t\beta_{d-1}^{+in} \\ &+ \big(1 - \frac{d\alpha}{ta_k} b_d\big)t\beta_d^{+in} + \sum_{l=2}^{d} \mathcal{O}(t^{-l}).\end{aligned}$$

By rearranging the above equation and using the facts that, as $t \to \infty$, then a_k, b_{d-1}, b_d all approach to 1, $\sum_{l=2}^{d} \mathcal{O}(t^{-l}) \to 0$, and I_M is a constant, we get

$$\beta_d^{+in} \approx \mathcal{O}(1) + \frac{d-1}{d+\frac{1}{\alpha}} \beta_{d-1}^{+in}; \qquad 2 \le d \le k \tag{3.55}$$

as $t \to \infty$.
Also, from the Eq.(3.53), we get for $d > k$

$$(t+1)\beta_d^{+in} = \frac{(d-1)\alpha}{ta_k} b_{d-1} t\beta_{d-1}^{+in} + \big(1 - \frac{d\alpha}{ta_k} b_d\big) t\beta_d^{+in} + \sum_{l=2}^{k} \mathcal{O}(t^{-l}) \tag{3.56}$$

Since, as $t \to \infty$, then a_k, b_{d-1}, b_d all approach to 1, $\sum_{l=2}^{d} \mathcal{O}(t^{-l}) \to 0$, then by rearranging above equation we get

$$\beta_d^{+in} = \frac{d-1}{d+\frac{1}{\alpha}} \beta_{d-1}^{+in}; \qquad d > k, \tag{3.57}$$

as $t \to \infty$.

Let, $\Delta_d^{+in}(t) = \mathbb{E}\big[X_d^{+in}(t)\big] - t\beta_d^{+in}$. To show that, $\mathbb{E}\big[X_d^{+in}(t)\big]$ can be approximated by $t\beta_d^{+in}$, we have to prove by induction that, $\big|\Delta_d^{+in}(t)\big|$ is bounded by a constant.

For $d = 1$, we get from the Eq.s(3.51) and (3.54)

$$\begin{aligned} \Delta_1^{+in}(t) + t\beta_1^{+in} &= I_M t + \mathcal{O}(1), \\ \Delta_1^{+in}(t) &= \frac{\alpha I_M t}{1+\alpha} + \mathcal{O}(1), \end{aligned} \tag{3.58}$$

Therefore, for $d = 1$ we get

$$|\Delta_1^{+in}(t)| \le \mathcal{O}(1). \tag{3.59}$$

Consider $\big|\Delta_d^{+in}(t)\big|$ is also bounded by a constant. Thus, we can write

$$\big|\Delta_d^{+in}(t)\big| = \big|\mathbb{E}\big[X_d^{+in}(t)\big] - t\beta_d^{+in}\big| \le \mathcal{O}(1). \tag{3.60}$$

Now, from the Eq.(3.52), we get

$$\begin{aligned} \Delta_d^{+in}(t+1) + (t+1)\beta_d^{+in} \approx{}& I_M + \frac{(d-1)\alpha}{ta_k} b_{d-1}\big(\Delta_{d-1}^{+in}(t) + t\beta_{d-1}^{+in}\big) \\ &+ \big(1 - \frac{d\alpha}{ta_k} b_d\big)\big(\Delta_d^{+in}(t) + t\beta_d^{+in}\big) + \sum_{l=2}^{d} \mathcal{O}(t^{-l}). \end{aligned}$$

By rearranging the above equation, we get

$$\begin{aligned}\Delta_d^{+in}(t+1) &= I_M + \frac{(d-1)\alpha b_{d-1}}{ta_k}\Delta_{d-1}^{+in}(t) + \frac{ta_k - d\alpha b_d}{ta_k}\Delta_d^{+in}(t)\\ &\quad + \frac{(d-1)\alpha b_{d-1}}{a_k}\beta_{d-1}^{+in} - \frac{d\alpha b_d + a_k}{a_k}\beta_d^{+in} + \sum_{l=2}^{d}\mathcal{O}(t^{-l}). \qquad (3.61)\end{aligned}$$

Now, by using the definition of β_d^{+in} from the Eq.(3.55), we can write

$$\begin{aligned}&\frac{(d-1)\alpha b_{d-1}}{a_k}\beta_{d-1}^{+in} - \frac{d\alpha b_d + a_k}{a_k}\beta_d^{+in}\\ &\quad = \frac{(d-1)\alpha b_{d-1}}{a_k}\frac{d\alpha+1}{(d-1)\alpha}\beta_d^{+in} - \frac{d\alpha b_d + a_k}{a_k}\beta_d^{+in} + \mathcal{O}(1)\\ &\quad = \frac{1}{a_k}((d\alpha+1)b_{d-1} - (d\alpha b_d + a_k))\beta_d^{+in} + \mathcal{O}(1)\\ &\quad = \left(\frac{b_{d-1}}{a_k} + \frac{d\alpha(b_{d-1}-b_d)}{a_k} - 1\right)\beta_d^{+in} + \mathcal{O}(1)\\ &\quad = A\beta_d^{+in} + \mathcal{O}(1), \qquad (3.62)\end{aligned}$$

where $A = \left(\frac{b_{d-1}}{a_k} + \frac{d\alpha(b_{d-1}-b_d)}{a_k} - 1\right)$.

Since, $b_{d-1} - b_d < 0$ and also $b_{d-1} \to 1$, $a_k \to 1$ for $t \to \infty$. Hence, $A \to 0$ as $t \to \infty$. Therefore, from the Eq.s(3.61) and (3.62), we get

$$\begin{aligned}\left|\Delta_d^{+in}(t+1)\right| &\leq I_M + \frac{(d-1)\alpha b_{d-1}}{ta_k}\left|\Delta_{d-1}^{+in}(t)\right| + \frac{ta_k - d\alpha b_d}{ta_k}\left|\Delta_d^{+in}(t)\right|\\ &\quad + A\beta_d^{+in} + \mathcal{O}(1)\\ &\leq I_M + \left(\frac{(d-1)\alpha b_{d-1}}{ta_k} + \frac{ta_k - d\alpha b_d}{ta_k}\right)\max\left(\left|\Delta_{d-1}^{+in}(t)\right|, \left|\Delta_d^{+in}(t)\right|\right)\\ &\quad + A\beta_d^{+in} + \mathcal{O}(1)\\ &= I_M + \left(1 - \frac{\alpha b_{d-1}}{ta_k} + \frac{d\alpha(b_{d-1}-b_d)}{ta_k}\right)\max\left(\left|\Delta_{d-1}^{+in}(t)\right|, \left|\Delta_d^{+in}(t)\right|\right)\\ &\quad + A\beta_d^{+in} + \mathcal{O}(1)\\ &= I_M + B\max\left(\left|\Delta_{d-1}^{+in}(t)\right|, \left|\Delta_d^{+in}(t)\right|\right) + A\beta_d^{+in} + \mathcal{O}(1), \qquad (3.63)\end{aligned}$$

where $B = \left(1 - \frac{\alpha b_{d-1}}{ta_k} + \frac{d\alpha(b_{d-1}-b_d)}{ta_k}\right)$.

Since, $b_{d-1} - b_d < 0$ and also $b_{d-1} \to 1$, $a_k \to 1$ for $t \to \infty$. Hence, $B \to 1$ as $t \to \infty$. Hence, from the Eq.s(3.59), (3.60), and (3.63), we get

$$\left|\Delta_d^{+in}(t+1)\right| = \left|\mathbb{E}\left[X_d^{+in}(+1)\right] - (t+1)\beta_d^{+in}\right| \leq \mathcal{O}(1). \tag{3.64}$$

Therefore, by using the induction hypothesis, we can say that, $\mathbb{E}\left[X_d^{+in}(t) - t\beta_d^{+in}\right]$ is bounded by a constant.

Now, to find the the power-low bound for the positive-in-degree distribution in G_t^K generated by edge copying model, we have to solve the following recurrence equation

$$\beta_d^{+in} = \frac{d-1}{(d+\frac{1}{\alpha})}\beta_{d-1}^{+in}; \quad \text{for } d > k, \tag{3.65}$$

with the initial conditions

$$\beta_1^{+in} \approx I_M; \quad \text{for } d = 1, \tag{3.66}$$

$$\beta_d^{+in} \approx \mathcal{O}(1) + \frac{d-1}{(d+\frac{1}{\alpha})}\beta_{d-1}^{+in}; \quad \text{for } 2 \leq d \leq k, \tag{3.67}$$

when $t \to \infty$.

From the Eq.(3.42), we know that, I_M, which is independent of t, is constant for a random process. Therefore, in Eq.(3.66), β_1^{+in} is also a constants.

Again, from the Eq.(3.67), the first term is constant for $2 \leq d \leq k$. Therefore, for $1 \leq d \leq k$, we can write (using Eq.(3.66))

$$\beta_k^{+in} = K \tag{3.68}$$

where K is a constant.

Therefore, from the Eq.(3.65), we get

$$\begin{aligned}\beta_d^{+in} &= K\prod_{i=k}^{d}\frac{i-1}{i+\frac{1}{\alpha}}\\ &= K\frac{\Gamma\left(k+\frac{1}{\alpha}\right)}{\Gamma(k-1)}\frac{\Gamma(d)}{\Gamma\left(d+\frac{1}{\alpha}+1\right)}\end{aligned} \tag{3.69}$$

By using Stirling's approximation in the above equation, we can write $\beta_d^{+in} \approx d^{-\left(1+\frac{1}{\alpha}\right)}$. □

Now, we analyze the dynamics of the positive-out-degree distribution in G_t^k. The model parameter $1-\alpha$ is for probability of the copying target vertex.

Let $\{\beta_d^{+out}\}$ be a sequence of positive integers. Then, by following the above procedure for positive-out-degrees, we can prove the following theorem for the positive-out-degree distribution in G_t^k.

Theorem 3.9. *Let $\mathbb{E}[X_d^{+out}(t)]$ be the expected number of vertices with positive-out-degree d in G_t^k generated by Model B. If $1-\alpha$ is the probability of coping target vertex from a randomly selected edge, then*

$$\left|\mathbb{E}[X_d^{+out}(t)] - t\beta_d^{+out}\right| \leq \mathcal{O}(1),$$

where β_d^{+out} has a power-law bound $d^{-\left(1+\frac{1}{1-\alpha}\right)}$.

In similar manner, let $\{\beta_d^{-in}\}$ and $\{\beta_d^{-out}\}$ be two sequences of positive integers. Then we can also prove the following theorems for the negative-in-degree and the negative-out-degree distributions respectively.

Theorem 3.10. *Let $\mathbb{E}[X_d^{-in}(t)]$ be the expected number of vertices with negative-in-degree d in G_t^k generated by Model B. If α is the probability of coping source vertex from a randomly selected edge, then*

$$\left|\mathbb{E}[X_d^{-in}(t)] - t\beta_d^{-in}\right| \leq \mathcal{O}(1),$$

where β_d^{-in} has a power-law bound $d^{-\left(1+\frac{1}{\alpha}\right)}$.

Theorem 3.11. *Let $\mathbb{E}[X_d^{-out}(t)]$ be the expected number of vertices with negative-out-degree d in G_t^k generated by Model B. If $1-\alpha$ is the probability of coping target vertex from a randomly selected edge, then*

$$\left|\mathbb{E}[X_d^{-out}(t)] - t\beta_d^{-out}\right| \leq \mathcal{O}(1),$$

where β_d^{-out} has a power-law bound $d^{-\left(1+\frac{1}{1-\alpha}\right)}$.

3.6 Model C: Clique Copying Model

3.6.1 Model Definition

In this model, we try to generalize our *edge copying model* for signed directed networks. According to the edge copying model, at each time, a new vertex enters to the network and copy an existing edge u.a.r to connect with a vertex (either source or target) from the selected edge. Alternatively, we can consider an edge as a k-clique where $k = 2$ and a vertex is a $(k-1)$-clique. Therefore, in other words, we can express the general form of the edge copying model as follows.

Initially, we start with an arbitrarily signed directed clique G_0^k of size $|V_t| = k+1$. At the time $t+1$, the new vertex v_{t+1} enters to the network to construct G_{t+1}^k. The vertex v_{t+1} connects with $k-1$ existing vertices and creates a new k-clique in the following ways:

(1) Select a k-clique uniformly at random form G_t^k.

(2) Select a vertex v from the selected k-clique uniformly at random. This process gives a $(k-1)$-clique in which the vertex v does not belong.

(3) Connect v_{t+1} with the vertices in $(k-1)$-clique by copying signed-directed-edges between the vertex v and the $(k-1)$-clique vertices.

3.6.2 Structural Balanced

The signed directed network generated by the clique copying model shows following structural property.

Theorem 3.12. *If the initial network G_0^k is structurally balanced, then the signed directed network $G_t^k = (V_t, E_t)$ generated by clique copying model is also structurally balanced.*

Proof. Let, at any time $t-1$, the network $G_{t-1}^k = (V_{t-1}, E_{t-1})$ is structurally balanced. Therefore, according to the balanced theory, we can find a partition in V_{t-1} such that the end vertices of a positive edge belong to the same group, and the end vertices of a negative edge belong to two different groups.

According to the clique model, at time t, a new vertex v_t enters to the network G_{t-1}^k and connects with all vertices in a $(k-1)$-clique by copying their one of the common vertices v. That is, the signed-directed-edges between v and the $(k-1)$-clique vertices are copied by the signed-directed-edges between v_t and the $(k-1)$-clique vertices. Let, $V(C_{k-1})$ is the set of vertices in the selected $(k-1)$-clique.

First, assume the existing network G_{t-1}^k is structurally balanced. Let, $k = 2$, i.e., $k-1 = 1$. Therefore, there exist only one vertex, let v_i, in the set $V(C_{k-1})$. Then, if the edge between v and $v_i \in V(C_{k-1})$ is positive then the new edge between v_t and v_i is also positive. In that case, v_t join the v_i's balanced partition in G_t^k. Again, if the edge between v and v_i is negative then the new edge between v_t and v_i is also negative. In that case, v_t creates a new vertex partition in G_t^k. In both cases, G_t^k preserves its structural balance.

Let, $k > 2$. Therefore, there exist more than one vertex in the set $V(C_{k-1})$. Since G_{t-1}^k is structurally balanced, then any two vertices $v_i, v_j \in V(C_{k-1})$ and their common neighbor vertex v are in the same partition if the edges among v_i, v_j and v are positive. If v_t connects with v_i and v_j by copying two positive edges (v, v_i) and (v, v_j), then v_t is also in the same partition with v_i and v_j in G_t^k. This addition of the new vertex preserves the balanced state of G_t^k.

Again, since G_{t-1}^k is structurally balanced, if v_i, v_j and v are in two partitions there exist exactly one positive edge among these vertices. Then v_t may copies one positive and one negative edges or both negative edges. If v_t copies one positive and one negative edges, then the edge between v_i and v_j must be a negative edge, i.e. v_i and v_j are in different partitions. Therefore, v_t enters either v_i or v_j's partition in G_t^k based on the new positive edge. In this case, G_t^k is also structurally balanced.

Again, if v_i, v_j and v are in three different partitions there exist no positive edge among these vertices. Then, v_t copies two negative edges to connect with v_i and v_j, which are already in different partitions. Therefore, v_t creates a new partition in V_t. This case also preserve the balanced state of G_t^k.

Next assume G_{t-1}^k is not balanced. Therefore, there exist at least three vertices v, v_i and v_j such that they are connected by exactly two positive edges and one negative edge. Now, let the vertex v_t connect with v_i and v_j by copying the edges (v, v_i) and (v, v_j). If v_t copies both positive edges then the edge between v_i and v_j must be negative, which leads G_t^k is not structurally balanced.

Again, if v_t copies one positive and one negative edge, then edges between v_i and v_j is positive, i.e. v_i and v_j are in same partition. Now, v_t has a positive edge and a negative edge with two vertices from the same partition, which leads G_t^k is not structurally balanced.

Therefore, if G_{t-1}^k is balanced, then G_t^k is also balanced. By using back induction, we conclude that, if the initial network G_0^k is balanced, then at any time the network generated by the clique copying model is also structurally balanced. $\square$

Model A: Preferential Attachment Model					
$\|V\|$, $\|E\|$	Param.'s	Dist. type	n	γ	p-value
10000, 79968	$k = 2$	pos-out-deg	9999	2.860	0.533
		pos-in-deg	9999	2.830	0.480
		neg-out-deg	9999	2.730	0.019
		neg-in-deg	9999	2.790	0.724
10000, 119928	$k = 3$	pos-out-deg	9999	2.930	0.510
		pos-in-deg	9999	2.840	0.688
		neg-out-deg	9999	2.840	0.449
		neg-in-deg	9999	2.800	0.337
10000, 159872	$k = 4$	pos-out-deg	9999	2.800	0.840
		pos-in-deg	9999	2.870	0.249
		neg-out-deg	9999	2.870	0.353
		neg-in-deg	9999	2.800	0.023

Table 3.3: Power-law exponents γ and the corresponding p-values for different signed-directed-degree distributions in the synthetic networks generated by preferential attachment model.

3.7 Simulation and Results Discussion

In the preferential attachment model (Model A), at each time, k number of positive and negative directed-edges are added in the both directions (in and out) with respect to the new vertex. So according to the *Theorem* 3.2-*Theorem* 3.5, all of the signed-directed-degrees follow the same power-law distribution with a exponent in the range $\gamma \approx 3$. In *Table 3.3*, the values of the exponent γ for the power-law model fitting for the signed-directed-degree distributions in the random networks generated by the preferential attachment model is $\approxeq 2.8$ which supports the theoretical argument.

Again, compare to our empirical study, the preferential attachment model only captures the observing property that the signed-directed-degree distributions follow the power-law with exponents in the range $2.0 \leq \gamma \leq 3.5$. But this model fails to capture the another observing property of having the inverse relationship between the number of vertices with in-degree and out-degree (for both positive and negative). This is because, in this model, the new vertex enters to the existing network with equal numbers of all the four types of signed-directed-degrees and there is no parameter to control the direction or sign of the newly added edges.

Model B: Edge copying model					
$\|V\|$, $\|E\|$	Param.'s	Dist. type	n	γ	p-value
100000,	$k = 2$,	pos-out-deg	42005	2.240	0.439
299996	$\alpha = 0.25$	pos-in-deg	87386	3.500	0.000
		neg-out-deg	23300	2.310	0.146
		neg-in-deg	57930	3.500	0.000
100000,	$k = 2$,	pos-out-deg	40771	2.850	0.165
299996	$\alpha = 0.50$	pos-in-deg	40797	2.770	0.119
		neg-out-deg	71259	2.760	0.003
		neg-in-deg	71379	2.780	0.012
100000,	$k = 2$,	pos-out-deg	89861	3.500	0.000
299996	$\alpha = 0.75$	pos-in-deg	44679	2.270	0.323
		neg-out-deg	51817	3.500	0.000
		neg-in-deg	20179	2.220	0.769
100000,	$k = 3$,	pos-out-deg	49327	2.260	0.246
399995	$\alpha = 0.25$	pos-in-deg	92175	3.500	0.000
		neg-out-deg	32638	2.270	0.541
		neg-in-deg	73278	3.500	0.000
100000,	$k = 3$,	pos-out-deg	54980	2.850	0.479
399995	$\alpha = 0.50$	pos-in-deg	54617	2.770	0.086
		neg-out-deg	78642	2.890	0.251
		neg-in-deg	78460	2.940	0.495
100000,	$k = 3$,	pos-out-deg	94961	3.500	0.000
399995	$\alpha = 0.75$	pos-in-deg	53777	2.250	0.654
		neg-out-deg	63809	3.500	0.000
		neg-in-deg	26806	2.270	0.870

Table 3.4: Power-law exponents γ and the corresponding p-values for different signed-directed-degree distributions in the network instances generated by edge copying model.

Model C: Clique copying model					
$\lvert V\rvert$, $\lvert E\rvert$	Param.'s	Dist. type	n	γ	p-value
100000, 100000	$k = 2$	pos-out-deg	16705	2.510	0.012
		pos-in-deg	16780	2.890	0.141
		neg-out-deg	8430	2.560	0.007
		neg-in-deg	8231	2.650	0.070
100000, 199998	$k = 3$	pos-out-deg	24138	2.300	0.003
		pos-in-deg	24611	2.420	0.146
		neg-out-deg	16908	2.360	0.954
		neg-in-deg	16902	2.370	0.200
100000, 299995	$k = 4$	pos-out-deg	28719	2.330	0.365
		pos-in-deg	31431	2.210	0.851
		neg-out-deg	19835	2.280	0.919
		neg-in-deg	19724	2.170	0.262

Table 3.5: Power-law exponents γ and the corresponding p-values for different signed-directed-degree distributions in the network instances generated by clique copying model.

From the results given in the *Table 3.4* for the edge copying model, we can observe that the power-law model fitting for the signed-directed-degree distributions in the random networks generated by this model are mostly statistically significant (p-*value* ≥ 0.01) with components in the range $2.0 \leq \gamma \leq 3.5$. Therefore, this model captures the real-world signed directed social networks property of having signed-directed-degree distributions with a component in the range $2.0 \leq \gamma \leq 3.5$.

Again, in the edge copying model (Model B), the signed-directed-degree distributions depend on the parameter α which is the probability of copying the target vertex from the randomly selected edge. That is, when $\alpha \to 1$, more vertices receive signed-out-degrees (both positive and negative) compare to the number of vertices that receive singed-in-degrees (both positive and negative). In *Table 3.4*, the values in the column n support this argument. Therefore, the edge copying model captures the inverse property between the number of in and out degree vertices (positive and negative) of the real-world signed directed social networks.

The results are given in *Table 3.5*, show that power law can also characterize the signed-directed degree distributions in random networks generated by clique copying model with an exponent in the range $2.0 \leq \gamma \leq 3.5$. Since there is no parameter for controlling the in and out direction of newly added edges, this model also does not capture the inverse property between the number of in and out degree vertices (positive and negative) of the real-world signed directed social networks.

The summary of the capturing our observed attributes (from *Table 3.2.2*) by the proposed random models for signed directed networks is given in the following *Table 3.7*.

Features	**Model A**	**Model B**	**Model C**
$(+/-)$-out-deg*	d^{-3}	$d^{-(1+\frac{1}{1-\alpha})}$	No
$(+/-)$-in-deg*	d^{-3}	$d^{-(1+\frac{1}{\alpha})}$	No
$2 \leq \gamma \leq 3.5$	Yes	Yes	Yes
Captured Attributes	A3	A1, A2, A3	A1, A2, A3

Table 3.6: Summary of capturing observed attributes by the proposed random models.

Chapter 4

Heuristic Algorithm for Correlation Clustering Problems

4.1 Correlation Clustering Problem

On a given signed weighted network (either undirected or directed), in which each edge is labeled by either a positive or a negative sign, the Correlation Clustering problem is to find a partition $\mathcal{P}$ in the vertex set that is consistent with the edge-sign labels as much as possible. This problem can, equivalently, be expressed in terms of two different objectives: *maximum agreements* and *minimum disagreements*. A positive edge can be regarded as a clustering agreement if the both end-vertices are in the same cluster, whereas, it can be regarded as a clustering disagreement if the end-vertices are in different clusters. On the other hand, a negative edge can be regarded as a clustering agreement if the both end-vertices are in different clusters, whereas, it can be regarded as clustering disagreement if the both end-vertices are in the same cluster. For the case of maximizing agreements, the correlation clustering problem looks at the total weight of positive (+) edges inside clusters, and negative (−) edges between the clusters. On the other hand, for the case of minimizing disagreements, the correlation clustering problem looks at the total weight of negative (−) edges inside the clusters and positive (+) edges between the clusters. In this chapter, we define the maximizing agreements and minimizing disagreements correlation clustering problems as Max-Agree-CC and Min-Agree-CC respectively.

Let $G = (V, E, s)$ be a signed network with n vertices, where every edge $e = (i, j)$ in E has a non-negative weight w_{ij}. We also define the weight of an edge $e = (i, j)$ equivalently as $w_e = w_{ij}$. Assume every edge $e \in E$ is labeled by a sign function $s : E \rightarrow \{+, -\}$. An edge (i, j) labeled with positive-sign (+) suggests that the vertices i and j are similar and should belong to the same cluster, whereas an edge (i, j) labeled with negative-sign

($-$) suggests that the vertices i and j are different and should be in different clusters. Let E^+ and E^- denote the set of all positive and negative edges in G respectively. Therefore, we can write $E = E^+ \cup E^-$ and $E^+ \cap E^- = \varnothing$.

Let $\mathcal{P} = \{P_1, P_2, ..., P_k\}$ be a partition of V. Let $C(i)$ be the set of vertices in the same cluster as i. Then we can define the total weight of the positive and negative edges inside the clusters due to the partition $\mathcal{P}$ respectively as

$$W^+_{IC}(\mathcal{P}) = \sum\{w_{ij} : (i,j) \in E^+,\ i \in C(j)\},$$
$$W^-_{IC}(\mathcal{P}) = \sum\{w_{ij} : (i,j) \in E^+,\ i \in C(j)\}.$$

Similarly, the total weight of the positive and negative edges between the clusters due to the partition $\mathcal{P}$ respectively as

$$W^+_{BC}(\mathcal{P}) = \sum\{w_{ij} : (i,j) \in E^+,\ i \notin C(j)\},$$
$$W^-_{BC}(\mathcal{P}) = \sum\{w_{ij} : (i,j) \in E^+,\ i \notin C(j)\}.$$

Therefore, the total weight of the positive edges insider the clusters and negative edges between the clusters due to the partition $\mathcal{P}$ is

$$f_w(\mathcal{P}) = W^+_{IC}(\mathcal{P}) + W^-_{BC}(\mathcal{P}) \tag{4.1}$$

Similarly, the total weight of the positive edges between clusters and negative edges inside clusters due to the partition $\mathcal{P}$ is

$$g_w(\mathcal{P}) = W^+_{BC}(\mathcal{P}) + W^-_{IC}(\mathcal{P}) \tag{4.2}$$

Based on the definition of Bansal et al. [BBC04], we can formulate the Max-Agree-CC and Min-Disagree-CC problems as follows:

Problem 4.1 Max-Agree-CC Problem.
Instance: A weighted signed graph $G = (V, E, s)$, where $|V| = n$ and $s : E \to \{+, -\}$.
Task: Find a partition $\mathcal{P}^*$ of vertices such that

$$f_w(\mathcal{P}^*) = \max_{\mathcal{P}} f_w(\mathcal{P}).$$

Problem 4.2 Min-Disagree-CC Problem.
Instance: A weighted signed graph $G = (V, E, s)$, where $|V| = n$ and $s : E \to \{+, -\}$.
Task: Find a partition $\mathcal{P}^*$ vertices such that

$$g_w(\mathcal{P}^*) = \min_{\mathcal{P}} g_w(\mathcal{P}).$$

In this chapter, we focus on *Problem 4.2*, i.e. minimizing disagreements due to the partition. In the following sections of this chapter we refer Min-Disagree-CC Problem equivalently as Correlation Clustering Problem or Clustering Problem.

4.2 Literature Review

The term Correlation Clustering was first used by Doreian and Mrvar [DM96] as a criteria for analyzing the structural balance in social networks. In 2003, Charikar et al. [CGW03] investigate the correlation clustering editing problem on both complete and general graphs. They also proved that this editing problem is APX-hard on complete graphs. Bansal et al. [BBC04], in 2004, formalized the Correlation Clustering problem as an optimization problem and showed that this is a special case of Clustering Editing problem defined on signed network. They also showed that this problem is a NP-hard and can be formulated in two different ways: *maximum agreements* (Max-Agree) and *minimum disagreements* (Min-Disagree). Since then, two distinct traits can be seen to solve this problem. Bansal et al. [BBC04] first presented a polynomial time approximation scheme (PTAS) for the Max-Agree problem when the edge weights of the signed networks are ± 1. In 2006, Giotis et al. [GG06] proposed another PTAS for the Max-Agree problem to signed network with ± 1 edge weights to find a partition $\mathcal{P}$ in which the maximum number of clusters in $\mathcal{P}$ is fixed, say k. Coleman et al. [CSW08] presented an efficient local-search approximation for this problem when $k = 2$. A 0.766-approximation algorithm for the Max-Agree problem to signed network with arbitrary edge weights was proposed by Charikar et al. [CGW05]. In 2015, Ahn et al. [ACG+15] introduced a Max-Agree of the Correlation Clustering problem in the dynamic data stream model and presented a polynomial time $\mathcal{O}(n.polylog\, n)$-space approximation algorithm.

On the other hand, Charikar et al. [CGW05] first proposed an approximate algorithm to solve the Min-Disagree correlation clustering problems in 2005. In 2006, Demaine et al. [DEFI06] studied this problem on general weighted graphs and presented an $\mathcal{O}(\log n)$-approximation algorithm based on linear programming rounding and *region growing* technique. An agent-based heuristic algorithm of the correlation clustering problems was proposed by Yang et al. [YCL07], in which no prior knowledge on hidden community structure is needed. A 3-approximation and implementable in the computational model such as MapReduce was introduced by Chierichetti et al. [CDK14].

The Correlation Clusterings problem is important in network science as well as other scientific areas [MMP12]. In social networks, this problem becomes a natural way to identify communities [CBGV$^+$12] and predicting missing edge sign in the link classification problem [CSX12]. For example, Figueiredo and Moura [FM13] used this problem to evaluate balanced partition in signed directed social networks by ignoring the edges directions. The Correlation Clustering problem has an significant use in the area of machine learning and data mining [CDK14, GMT07, ACG$^+$15], portfolio analysis in risk management [FF14, HLW02], biological system networks [HBN07, DESZ07] etc..

4.3 Heuristic Algorithm for Correlation Clustering Problems

4.3.1 Integer Linear Programming Formulation

In this section, we restate the integer linear programming formulation of correlation clustering problem on general weighted signed graph proposed by Demaine et al. [DEFI06]. We also used Grötschel and Wakabayashi [GW89] integer linear programming formulation of clustering editing problem for simplifying the constraints, which later studied by Charikar et al. [CGW03], and Böcker et al. [BBK11].

Consider a set of $\binom{n}{2}$ binary decision variables $X = (x_{ij}; 1 \leq i < j \leq n)$ to represent each pair of vertices in G. Then, for a given clustering partition $\mathcal{P}$, set $x_{ij} = 0$ if i and j are in a same cluster, and $x_{ij} = 1$ otherwise. Here, the solution matrix X for a given partition $\mathcal{P}$ can be represented as an underlying induced undirected and unsigned graph G_X with the same set of vertices as G. We can define this underlying graph G_X by the following definition.

Definition 4.1 (X-Induced Graph)**.** A graph $G_X = (V, E_X)$ is said to be *X-Induced* for a given matrix X if and only if $(i, j) \in E_X$, $i, j \in V$ then

$x_{ij} = 0$.

Therefore, we can draw the relation among the signed graph G, a given solution matrix X, and the underlying X-Induced graph G_X in such way that,

$$\begin{aligned} \text{if } x_{ij} = 0, &\implies (i,j) \in E_X, \\ &\implies i \text{ and } j \text{ are in the same cluster in } G \end{aligned}$$

Alternatively, we can express this relation as for a given partition $\mathcal{P}$ in a signed graph G if vertices i and j are in the same cluster then $(i, j) \in G_X$.

Now by definition, we know that if $1 - x_{ij} = 1$ then vertices i and j are in the same cluster, and $1 - x_{ij} = 0$ then they are in the different clusters. Thus, we can express $g_w(\mathcal{P})$ given Eq.(4.2) as follows:

$$g_w(\mathcal{P}) = \sum_{(i,j)\in E^+} w_{ij}x_{ij} + \sum_{(i,j)\in E^-} w_{ij}(1 - x_{ij}), \tag{4.3}$$

Described in Demaine et al. [DEFI06], the integer linear programming formulation for the Correlation Clustering Problems, given in the *Problem 4.2* which minimizes the objective function given in the Eq.(4.3), can be defined as follows:

$$\min \quad \sum_{(i,j)\in E^+} w_{ij}x_{ij} + \sum_{(i,j)\in E^-} w_{ij}(1 - x_{ij}); \quad \forall i, j \in V, \tag{4.4}$$

$$\text{subject to:} \quad x_{ij} + x_{jk} \geq x_{ik}; \quad \forall i, j, k \in V, \tag{4.5}$$

$$x_{ij} = x_{ji}; \quad \forall i, j \in V, \tag{4.6}$$

$$x_{ij} \in \{0, 1\}; \quad \forall i, j \in V. \tag{4.7}$$

The inequality constraint, in Eq.(4.5), enforces the condition that any distinct vertices $i, j, k \in V$ such that, if i and j are in a same cluster then k is also in this cluster. This is also called *triangle inequality constraint.* The equality constraint, in Eq.(4.6), is to represent the undirected edge constraint.

Therefore, our goal is to solve the integer linear programming problems given in Eq.s(4.4)-(4.7) to find the solution matrix X which leads us to a vertex partition $\mathcal{P}$. The underlying X-induced graph G_X induced by this solution matrix X will be a collection of disjoint maximal clique, in which the vertices set corresponding to each maximal clique represents a cluster in $\mathcal{P}$.

4.3.2 Relaxed-ILP

In this step, we relax the integer constraint in the Eq.(4.7). Then we get the following linear programming problem:

$$\min \quad \sum_{(i,j)\in E^+} w_{ij}x_{ij} + \sum_{(i,j)\in E^-} w_{ij}(1-x_{ij}); \quad \forall i,j \in V, \tag{4.8}$$

$$\text{subject to:} \quad x_{ij} + x_{jk} \geq x_{ik}; \quad \forall i,j,k \in V, \tag{4.9}$$

$$x_{ij} = x_{ji}; \quad \forall i,j \in V, \tag{4.10}$$

$$x_{ij} \in [0,1]; \quad \forall i,j \in V. \tag{4.11}$$

Based on Grötschel and Wakabayashi [GW89], the linear programming formulation for the CORRELATION CLUSTERING PROBLEMS, the above relaxed linear programming problem, given in Eq.s(4.8)-(4.11), equivalently can be written as follows :

$$\min \sum_{(i,j)||(j,i)\in E^+} w_{ij}x_{ij} + \sum_{(i,j)||(j,i)\in E^-} w_{ij}(1-x_{ij}); \quad \forall\, 1 \leq i < j \leq n, \tag{4.12}$$

$$\text{subject to:} \; x_{ij} + x_{jk} \geq x_{ik}; \quad \forall\, 1 \leq i < j < k \leq n, \tag{4.13}$$

$$x_{ij} + x_{ik} \geq x_{jk}; \quad \forall\, 1 \leq i < j < k \leq n, \tag{4.14}$$

$$x_{jk} + x_{ik} \geq x_{ij}; \quad \forall\, 1 \leq i < j < k \leq n, \tag{4.15}$$

$$0 \leq x_{ij} \leq 1; \quad \forall\, 1 \leq i < j \leq n. \tag{4.16}$$

This relaxed problem, given in Eq.s(4.12)-(4.16), can be solved by using any standard linear programming algorithm by the time polynomial of the input size.

Let $X_R = (x_{ij}; 1 \leq i < j \leq n)$ be the solution of the above relaxed problem. Here, we may consider X_R as a distance matrix in which $x : V \times V \to [0,1]$ is the distance function with the following properties:

$$0 \leq x_{ij} \leq 1; \quad \forall\, i,j \in V, \tag{4.17}$$

$$x_{ij} = x_{ji}; \quad \forall\, i,j \in V, \tag{4.18}$$

$$x_{ij} + x_{jk} \geq x_{ik}; \quad \forall\, i,j,k \in V. \tag{4.19}$$

Therefore, after solving the linear programming problem, given in Eq.s(4.12)-(4.16), we get a complete weighted graph G_{X_R} induced by the solution (distance) matrix X_R in which all entries (distances) satisfies the above conditions Eq.s(4.17)-(4.19) and lies between $[0,1]$.

At this point, our goal is to calculate a distance matrix $X^* = (x^*_{ij}; 1 \leq i < j \leq n)$, which is closest to the solution distance matrix X_R and is a feasible solution of the integer program problem given in Eq.s(4.8)-(4.11). Thus the distance matrix X^* satisfy the constraints given in Eq.s(4.9)-(4.11), the underlying X-Induced graph $G_{X^*}(V, E_{X^*})$ induced by X^* satisfy , if $x^*_{ij} = 0, (i,j) \in V$ then $(i,j) \in E_{X^*}$ and otherwise, and then all of the connected components in G_{X^*} can be interpreted as approximate cluster in the signed graph G.

4.3.3 Ultrametric Distance Matrix

In this step, we calculate the ultrametric distance matrix U_X for the given solution distance matrix X_R which is the solution of the relaxed linear program of the Correlation Clustering Problem. An *ultrametric* on the set V is defined as follows.

Definition 4.2 (Ultrametric)**.** A distance function $u : V \times V \to \mathbb{R}_0^+$ is said to be ultrametric if

$$\max\{u_{ij}, u_{jk}\} \geq u_{ik}; \qquad \forall\, i, j, k \in V, \tag{4.20}$$

where, u_{ij} is the distance between i and j for all $i, j \in V$.

Ultrametric Definition as Linear Inequality: Consider the above distance function as $u : V \times V \to \{0, 1\}$. Then the ultrametric condition given in Eq.(4.20) can be written as:

$$u_{ij} + u_{jk} \geq u_{ik} \tag{4.21}$$

which is equivalence to the triangle inequality constraint, given in Eq.(4.5), in the Correlation Clustering Problem given in Eq.s(4.4)-(4.7). Based on the *Definition 4.2* and Eq.(4.21), we define the following definition.

Definition 4.3 (0-1 Ultrametric Distance Matrix)**.** A distance matrix U is said to be *0-1 Ultrametric Distance Matrix* if each of the elements in U satisfies the linear inequality conditions given in Eq.(4.21), where $u : V \times V \to \{0, 1\}$.

From the above *Definition 4.3* and Eq.(4.21), we can say that any feasible solution matrix of the integer linear programming formulation for the Correlation Clustering Problem problem given in Eq.s(4.4)-(4.7) is also a 0-1 Ultrametric Distance Matrix.

Therefore, here, our goal to find the closest 0-1 Ultrametric Distance Matrix U_X for a given solution distance matrix X_R in which all entries satisfy the distance function $x : V \times V \to [0, 1]$. Here, X_R is the solution of the relaxed linear programming problem given in Eq.s(4.12)-(4.16). This problem can be formulated as follows:

Problem 4.3. 0-1 Ultrametric Distance Matrix.
Instance: A distance matrix X_R with $x : V \times V \to [0, 1]$.
Task: Find a 0-1 Ultrametric Distance Matrix U_X, where $u : V \times V \to \{0, 1\}$, in minimum distance (cost).

In the above problem, finding U_X with minimum cost is hard. We can solve this issue by using two following steps: approximation and rounding. In the first step, we solve the *Problem 4.3* to approximate the closest ultrametric distance matrix by relaxing the integer constraint. The relaxed version of the *Problem 4.3* can be described as follows:

Problem 4.4. Closest Ultrametric.
Instance: A distance matrix X_R with $x : V \times V \to [0, 1]$.
Task: Find the closest ultrametric distance matrix U_R with $u : V \times V \to [0, 1]$.

After finding the ultrametric distance matrix U_R (relaxed) by solving *Problem 4.4*, we can use a rounding method by using a given threshold k to determine the 0-1 ultrametric distance matrix U_X. The rounding problem can be formulated as follows:

Problem 4.5. Rounding.
Instance: A distance matrix $U_R = (u_{ij})$ with $u : V \times V \to [0, 1]$ and a given threshold k.
Task: Find a distance matrix $U_X = (u^*_{ij})$, such that $u^* : V \times V \to [0, 1]$ by using a rounding process.

4.3.4 Closest Ultrametric

In this section, we focus on solving the *Problem 4.4*, which is a *closest ultrametric* problem. The complexity and algorithm for finding the closest ultrametric from $V \times V$, where V is set of vertices of a complete weighted graph $G' = (V, E)$, depends on the type of distortion we are looking for. The *Problem 4.4*, which is finding and ultrametric u which is closest to x

on V can be formulated under l_p-distortion as follows:

Problem 4.6. Closest Ultrametric (l_p-Distortion).
Instance: A distance matrix X_R with $x : V \times V \to [0, 1]$.
Task: Find a ultrametric distance matrix U_X with $u : V \times V \to [0, 1]$, such that

$$\min_{u \in \mathcal{U}} \max_{i,j \in V} \Big(\sum_{i,j \in V} |u_{ij} - x_{ij}|^p \Big)^{1/p},$$

where $u : V \times V \to \mathbb{R}_0^+$ and $\mathcal{U}$ is the set of all ultrametrics on V.

Křivánek and Morávek[KM86] proved that this problem is NP-hard for $p = 1$, i.e. for the case of additive distortion. Later, Harb et al.[HKM05] proved it is APX-hard for any fixed $p \geq 1$.

Again, the *Problem 4.4* of finding the closest ultrametric u on V can be formulated under l_∞-distortion as follows:

Problem 4.7. Closest Ultrametric (l_∞-Distortion).
Instance: A distance matrix X_R with $x : V \times V \to [0, 1]$.
Task: Find a ultrametric distance matrix U_X with $u : V \times V \to [0, 1]$, such that

$$\min_{u \in \mathcal{U}} \max_{i,j \in V} |u_{ij} - x_{ij}|,$$

where $u : V \times V \to \mathbb{R}_0^+$ and $\mathcal{U}$ is the set of all ultrametrics on V.

Křivánek [Kři88] showed that the complexity of the algorithm to solve *Problem 4.7*, i.e. to find closest ultrametric u on V from x under l_∞-distortion is $\mathcal{O}(n^3)$. In Křivánek's algorithm, the ultrametric distance between vertices $i, j \in V$ are adjusted by the '*bottleneck*' in the minimum spanning tree T on G'. This bottleneck in T can be defined as

$$\max_{e \in T(i,j)} x_e, \tag{4.22}$$

where $T(i, j)$ is the path between the vertices i and j in T. It can be noted that, the graph G' may have more than one minimum spanning tree, but the value in Eq.(4.22) is independent of the selection of T. Křivánek[Kři88] proved the following theorem:

Theorem 4.4 ([Kři88])**.** *If T be a minimum spanning tree on a complete* *weighted graph G, then*

$$2 \min_{u \in \mathcal{U}} \max_{i,j \in V} |x_{ij} - u_{ij}| = \max_{i,j \in V} \{x_{ij} - \max_{e \in T(i,j)} x_e\} \tag{4.23}$$

where $T(i,j)$ is the edge set of the path between i to j in T, x_e is the edge weight (distance) of an edge $e \in T(i,j)$, and $\mathcal{U}$ is the set of all ultrametrics on V.

Křivánek[Kři88] also proved that, for a given weighted completed graph $G = (V, E)$ and a minimum spanning tree $T = (V, E_T)$ on G, an ultrametric $u^* : V \times V \to \mathbb{R}_0^+$ on V such that

$$u^*_{ij} = \frac{1}{2} \max_{e \in T(i,j)} \{x_e + x'_e\}; \quad \text{for all } i, j \in V, \tag{4.24}$$

satisfies Eq.(4.23), where x'_e is the adjustment valuation can be defined by

$$x'_{ij} = \max_{e \in T(i,j)} x_e; \quad \text{for each } (i,j) \notin E_T \text{ and,} \tag{4.25}$$

$$x'_e = \max_{(i,j) \notin E_T} \{x'_{i,j}, x_e\}; \quad \text{for each } e \in E_T. \tag{4.26}$$

In this point, by using Eq.s(4.24)-(4.26), our goal is to find the closest ultrametric distance matrix U_X on V in which $u^* : V \times V \to \mathbb{R}_0^+$ from the given distance matrix X_R obtained from the solution of the relaxed linear program given in Eq.s(4.12)-(4.16).

4.3.5 Rounding

In this step, our focus to solve the *Problem 4.5* to get an 0-1 ultrametric distance matrix $U_X = (u^*_{ij})$; $\forall i, j \in V$, from the calculated relaxed ultrametric distance matrix U_R with the distance function $u : V \times V \to [0, 1]$, and a given threshold k. We use a simple rounding process such that, take $u^*_{ij} = 0$ if $u_{ij} \leq k$ for each $i, j \in V$, otherwise $u^*_{ij} = 1$.

4.3.6 The Algorithm and Implementation: Summary

The algorithmic steps and implementations procedures of the above algorithm for solving the Correlation Clustering Problems are given in the follows:

Step 1: Solve the relaxed linear program problem, given in Eq.s(4.12)-(4.16). Let $G_{X_R} = (V, E_{X_R})$ be the underlying X-induced complete weighted-graph by the solution (distance) matrix X_R. Each real number $x_{ij} \in X_R$ represents the weight (distance) corresponds to the edge $(i,j) \in E_{X_R}$.

Step 2: Find a minimum spanning-tree $T_{X_R} = (V, E_T)$ of G_{X_R}. We use Kruskal's[Kru56] algorithm for minimum spanning-tree.

Step 3: Compute adjust valuation x'_{ij} for all $i, j \in V$ from Eq.s(4.24)-(4.26). This can be implemented as follows:

Algorithm 4: $u^* : V \times V \to [0, 1]$.

Data: Complete, weighted graph $G(V, E)$ and a spanning tree $T(V, E_T)$ on G.
Result: Ultrametric distance matrix U_X.

for *each* $(i, j) \notin E_T$ **do**
    $T(i, j) \leftarrow$ set of edges in the path between i and j;
    $x'_{ij} \leftarrow \max_{e \in T(i,j)} x_e$;
end
for *each* $e \in E_T$ **do**
    $x'_e \leftarrow \max\{x'_{ij}; (i, j) \notin E_T \,\&\, x'_{ij} = x_e\}$
end
for *each* $(i, j) \in E$ **do**
    $T(i, j) \leftarrow$ set of edges in the path between i and j;
    $u^*_{ij} \leftarrow \frac{1}{2} \max_{e \in T(i,j)}\{x'_e + x_e\}$;
end

Step 4: Find $U_X = (u^*_{ij})$ from $U_R = (u_{ij})$ by using the given threshold k and return the partition $\mathcal{P}$ such that the vertices i and j are in same cluster if $u^*_{ij} <= k$; $\forall i, j \in V$. Otherwise, i and j are in different clusters.

Corollary 4.1. (Complexity) The proposed heuristic algorithm runs in polynomial-time of the input network size.
Proof. The complexity of the step 1 for solving relaxed linear program is polynomial with the input graph size [Meg86]. In step 2 and 3, the complexity of finding the closest ultrametric distance matrix from the solution matrix X_R is $\mathcal{O}(n^3)$ [Kři88]. Finally, the complexity of a straight forward implementation of the rounding in step 4 is $\mathcal{O}(n^2)$. $\boxdot$

4.4 Experimental Results

Evaluation Platform: We implements the proposed algorithm in Java and use the IBM CPLEX V.12.1 solver for solving the relaxed ILP problem. We also use graph package jGrapht to deal with the graph properties. The running times we calculated on a system with Intel Core i5 @ 1.70 GHz, 64 bit and 8GB memory.

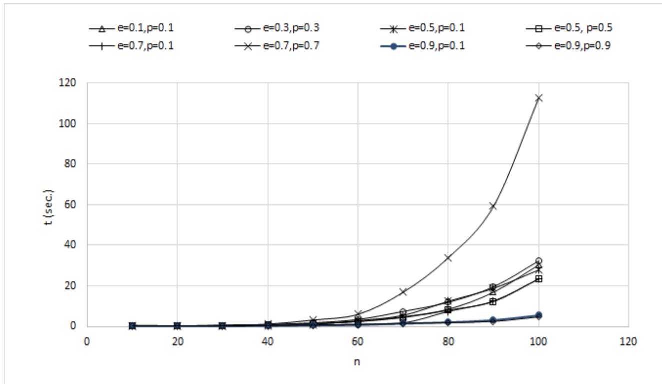
4.4.1 Random $G(n, e, p)$ Signed Networks:

We generate random $G(n, e, p)$ network instances, in which n the number of vertices and e is the probability of connecting two vertices, and if there is an edge, then p is the probability for that edge is positive. Therefore, the probability of connecting two vertices with a positive edge is ep and with a negative edge is $e(1-p)$. The experimental results of the proposed heuristic algorithm are given in Fig.4.1 for different random network instances.

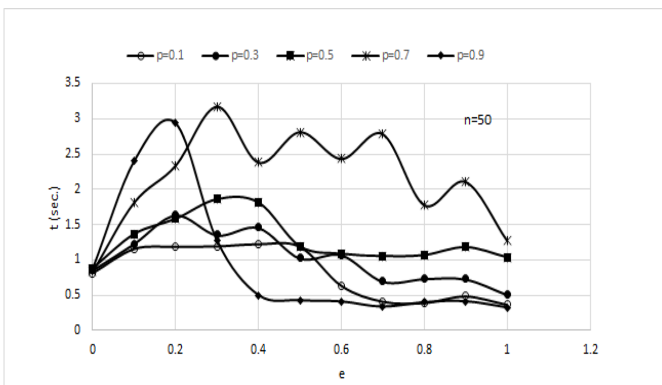
According to *Corollary 4.1* any striaght forward implementation of the proposed algorithm should run in polynomial time. In Fig.4.1(a), it looks like the running time graphs for changing networks size in different network instances are polynomial except for the case when $e = 0.7$, $p = 0.7$. For this case the run time graph seems like increasing exponentially. This exception may arise due to some issues in our implementation which we failed to identify.

In the proposed heuristic algorithm, after solving the ILP-relaxed problem, we deal with the complete induced weighted network to determine the $0-1$-ultrametric distance matrix. Also in the ILP-relaxed problem, the number of decision variable only depends on the size of the vertex set and independent from the size of the edge set. Therefore, according to our hypothesis, the run time should be independent from the edge density (for both positive and negative edges). The Fig.4.1(b) support this hypothesis for the cases when $e \geq 0.4$. With the same argument, the runtime should be independent from the ratio of positive or negative edge densities. The Fig.4.1(c) also supports the argument for the cases $p \leq 0.6$.

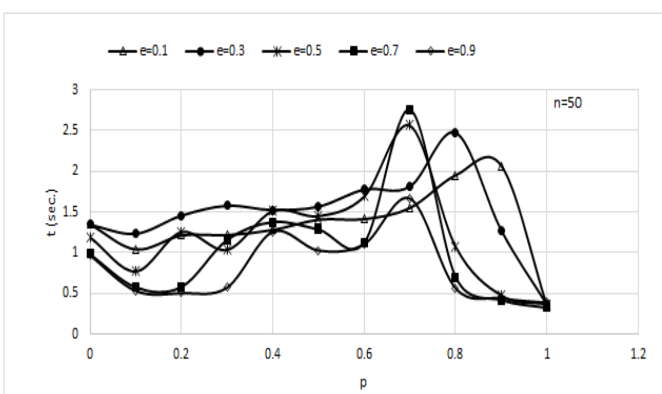
Next, we tested the variations of the minimum disagreements due to the partition with the changing of the given threshold. For do this we have tested the variations in ten random signed $G(n, e, p)$ networks with fixed $n = 100, e = 0.5, p = 0.5$. The results, in Fig.4.2, shows an inconclusive argument on the relation between the minimum disagreement due to the partition and user-given threshold.



(a)



(b)



(c)

Figure 4.1: (a) Runtime (in sec.) for changing n when e and p are fixed. (b) Runtime (in sec.) for changing e when $n = 50$ and p are fixed. (c) Runtime (in sec.) for changing p when $n = 50$ and e are fixed.

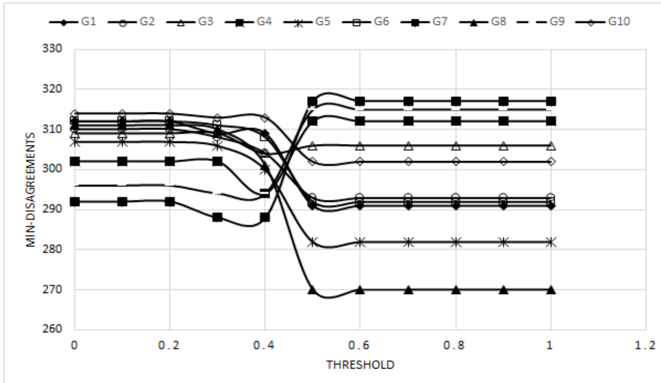


Figure 4.2: Minimum disagreement due the changing of threshold in ten random signed network instances $G_1, ..., G_{10}$ with fixed $n = 100, e = 0.5, p = 0.5$s.

4.4.2 International Bilateral Trade Growth Rate Network

The International Trade Centre (ITC)[1] is an auxiliary group of the World Trade Organization (WTO)[2] and the United Nations Conference on Trade and Development (UNCTAD)[3]. It provides trade-related technical assistance and bilateral trading data between its member countries and economical territories. We have collected bilateral average trade growth rate between the years 2011-2015 among 231 countries and territories. The network consist 231 vertices (country or territory) and 16,356 edges. Each signed and weighted edge indicates the average of the total trade rate (import and export) between two members. The sign of each edge depends on the positive-negative growth rate. The summary of the trade growth rate network is given in the following *Table 4.3*.

We have solved to find the partition in the country set by using the proposed heuristic algorithm and different thresholds. At $threshold = 0.45$ the algorithm return 189 clusters. Most of these clusters include a single county or economic territory except few. The clusters with more than three countries are given in Fig.4.3. Again by using $threshold = 0.5$ the algorithm returns only 5 (five) clusters, in which all of the countries are in one cluster excepts the countries: Iran, Kazakhstan, Greenland, Syrian Arab Republic. These four countries are in four separate clusters. From this result, the only

[1] `http://www.intracen.org/`
[2] `https://www.wto.org`
[3] `http://www.unctad.org`

Number of Countries	231
Edges	16356
Positive edges	7471
Negative edges	8885
Balance triangles	340495
Imbalance triangles	353107

Table 4.1: Summary of the International Bilateral Trade Growth Rate Network 2011-2015.

Figure 4.3: Clusters of countries when $threshold = 0.45$.

information we can predict that due to the UN economic sanction on Iran and recent Syrian civil war the bilateral trading with these two countries with rest of the world has been drastically decreased in the period 2011-2015. The algorithm returns a single cluster for the $threshold > 0.5$ and puts each countries in separate clusters for the cases $threshold \leq 0.4$.

Chapter 5

Conclusion

In this thesis, we attempted to study the two prominent areas of network science: the evolution of the signed directed social network, e.g. Wikipedia's request for adminship (Wiki-RfA), etc. and to design a heuristic algorithm for the CORRELATION CLUSTERING PROBLEMS in the signed networks. Those works are presented in *Chapter 3* and *Chapter 4* respectively.

5.1 Random Models for Signed Directed Social Networks

In *Chapter 3*, to the best of our knowledge, we have studied (for the first time) the signed-directed-degree distributions in the real-world web-based signed directed social networks and proposed three random models: preferential attachment model, edge copying model, and clique copying model. Our analysis and simulation results suggest that the signed-directed degree distributions in the networks simulated by the proposed models follow a power law with an exponent in the range $2.0 \leq \gamma \leq 3.5$. For the clique copying model, we have proved that if the initial network is structurally balanced, then the signed directed networks generated by this model is also structurally balanced.

Future Works: We have presented theoretical proof for the power-law signed-directed degree distributions in the networks generated by preferential attachment and edge copying models. Despite this theoretical justification, we still need to prove that the number of vertices of degree d concentrates on its expectation. For the clique copying model, one also requires a theoretical analysis for its power-law signed-directed distributions. Also, an empirical experiment is needed to justify for the balance network theorem in this model.

5.2 Heuristic Algorithm for Correlation Clustering Problems

In *Chapter 4*, we have proposed a heuristic algorithm for the Correlation Clustering Problem which is a NP-hard problem. Our experimental results for random signed $G(n, e, p)$ network instances have shown that the runtime of this algorithm is independent of the case when $e \geq 0.4$ or when $p \leq 0.6$. The limitation of this algorithm is that it can not give any conclusive argument for the changing of the minimum disagreements due to the variation of given *threshold*.

Future Works: To improve the runtime performance of this algorithm we can apply a data reduction technique to reduce the input graph size. The process given in [BBK11] and [GHK$^+$10] may lead us to this research. Again, after solving the closest ultrametric problem, we use simple rounding based on the given threshold. We would also like to improve an efficient rounding technique to get a better result.

Bibliography

[AAELvZ12] Nir Ailon, Noa Avigdor-Elgrabli, Edo Liberty, and Anke van Zuylen. Improved approximation algorithms for bipartite correlation clustering. *SIAM Journal on Computing*, 41(5):1110–1121, 2012. → pages 3

[ACG+15] KookJin Ahn, Graham Cormode, Sudipto Guha, Andrew McGregor, and Anthony Wirth. Correlation clustering in data streams. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, pages 6–11, 2015. → pages 56, 57

[ACL00] William Aiello, Fan Chung, and Linyuan Lu. A random graph model for massive graphs. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 171–180. ACM, 2000. → pages 27

[ACL01] William Aiello, Fan Chung, and Linyuan Lu. A random graph model for power law graphs. *Experimental Mathematics*, 10(1):53–66, 2001. → pages 27

[BA99] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. → pages 2, 7, 8, 27

[BBC04] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004. → pages 3, 55, 56

[BBK11] Sebastian Böcker, Sebastian Briesemeister, and Gunnar W. Klau. Exact algorithms for cluster editing: Evaluation and experiments. *Algorithmica*, 60(2):316–334, 2011. → pages 57, 70

[BE05] Ulrik Brandes and Thomas Erlebach. *Network Analysis: Methodological Foundations*, volume 3418. Springer Science, 2005. → pages 4, 6

[Bel58] Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958. → pages 12

[BK73] Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973. → pages 15

[BR03] Béla Bollobás and Oliver M. Riordan. Mathematical results on scale-free random graphs. *Handbook of graphs and networks: from the genome to the internet*, pages 1–34, 2003. → pages 27

[BS06] Stefan Bornholdt and Heinz Georg Schuster. *Handbook of graphs and networks: from the genome to the internet.* John Wiley & Sons, 2006. → pages 4

[CBC+15] V. Ciotti, G. Bianconi, A. Capocci, F. Colaiori, and P. Panzarasa. Degree correlations in signed social networks, 2015. → pages 2, 11, 27, 28

[CBGV+12] Nicolo Cesa-Bianchi, Claudio Gentile, Fabio Vitale, Giovanni Zappella, et al. A correlation clustering approach to link classification in signed networks. In *COLT*, pages 34–1, 2012. → pages 57

[CDK14] Flavio Chierichetti, Nilesh Dalvi, and Ravi Kumar. Correlation clustering in mapreduce. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 641–650. ACM, 2014. → pages 3, 57

[CF03] Colin Cooper and Alan Frieze. A general model of web graphs. *Random Structures & Algorithms*, 22(3):311–335, 2003. → pages 8, 9, 27

[CGW03] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 524–533. IEEE, 2003. → pages 3, 56, 57

[CGW05] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. *Journal of Computer and System Sciences*, 71(3):360–383, 2005. → pages 56

[CH56] Dorwin Cartwright and Frank Harary. Structural balance: a generalization of heider's theory. *Psychological review*, 63(5):277, 1956. → pages 1, 2, 19

[CH73] V. Chvátal and P. Hammer. Set packing and threshold graphs, univ. *Waterloo Res. Report*, pages 73–21, 1973. → pages 17

[Cor09] Thomas H. Cormen. *Introduction to Algorithms*. MIT press, 2009. → pages 14

[CSN09] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009. → pages 21

[CSW08] Tom Coleman, James Saunderson, and Anthony Wirth. A local-search 2-approximation for 2-correlation-clustering. In *European Symposium on Algorithms*, pages 308–319. Springer, 2008. → pages 56

[CSX12] Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering sparse graphs. In *Advances in neural information processing systems*, pages 2204–2212, 2012. → pages 57

[Dav77] James A. Davis. Clustering and structural balance in graphs. *Social networks. A developing paradigm*, pages 27–34, 1977. → pages 19, 20

[DEFI06] Erik D. Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2):172–187, 2006. → pages 3, 57, 58

[DESZ07] Bhaskar DasGupta, German Andres Enciso, Eduardo Sontag, and Yi Zhang. Algorithmic and complexity results for decompositions of biological networks into monotone subsystems. *Biosystems*, 90(1):161–178, 2007. → pages 3, 57

[Dij59] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959. → pages 12

[DM96] Patrick Doreian and Andrej Mrvar. A partitioning approach to structural balance. *Social Networks*, 18(2):149–168, 1996. → pages 56

[DMS00] Sergey N. Dorogovtsev, José Fernando F Mendes, and Alexander N Samukhin. Structure of growing networks with preferential linking. *Physical Review Letters*, 85(21):4633, 2000. → pages 27

[ER59] Paul Erdős and Alfréd Rényi. On random graphs. *Publ. Math. Debrecen*, 6:290–297, 1959. → pages 6, 27

[ER60] Paul Erdös and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(17-61):43, 1960. → pages 27

[ER61] Paul Erdős and Alfréd Rényi. On the strength of connectedness of a random graph. *Acta Mathematica Hungarica*, 12(1-2):261–267, 1961. → pages 27

[EZE85] M. El-Zahar and P. Erdős. On the existence of two non-neighboring subgraphs in a graph. *Combinatorica*, 5(4):295–300, 1985. → pages 17

[FF14] Rosa Figueiredo and Yuri Frota. The maximum balanced subgraph of a signed graph: Applications and solution approaches. *European Journal of Operational Research*, 236(2):473–487, 2014. → pages 3, 57

[FFF99] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM computer communication review*, volume 29, pages 251–262. ACM, 1999. → pages 2, 27

[Flo62] Robert W. Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962. → pages 12

[FM13] Rosa Figueiredo and Gisele Moura. Mixed integer programming formulations for clustering problems related to structural balance. *Social Networks*, 35(4):639–651, 2013. → pages 57

[FT87] Michael L. Fredman and Robert Endre Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987. → pages 13

[Gao09] Yong Gao. The degree distribution of random k-trees. *Theoretical Computer Science*, 410(8):688–695, 2009. → pages 10, 11, 27

[Gao14] Yong Gao. Community structure-lecture 07. Course Note, Computer Science, The University of British Columbia, Okanagan, 2014. → pages 16, 17

[GG06] Ioannis Giotis and Venkatesan Guruswami. Correlation clustering with a fixed number of clusters. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1167–1176. Society for Industrial and Applied Mathematics, 2006. → pages 56

[GHK+10] Jiong Guo, Sepp Hartung, Christian Komusiewicz, Rolf Niedermeier, and Johannes Uhlmann. Exact algorithms and experiments for hierarchical tree clustering. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI10)*, pages 457–462, 2010. → pages 70

[GJ79] Michael R. Garey and David S. Johnson. A guide to the theory of np-completeness. *WH Freemann, New York*, 1979. → pages 15

[GMT07] Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):4, 2007. → pages 3, 57

[Gol78] Martin Charles Golumbic. Trivially perfect graphs. *Discrete Mathematics*, 24(1):105–107, 1978. → pages 17

[GW89] Martin Grötschel and Yoshiko Wakabayashi. A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45(1-3):59–96, 1989. → pages 57, 59

[Har59] Frank Harary. On the measurement of structural balance. *Behavioral Science*, 4(4):316–323, 1959. → pages 19

[HBN07] Falk Hüffner, Nadja Betzler, and Rolf Niedermeier. Optimal edge deletions for signed graph balancing. In *International*

Workshop on Experimental and Efficient Algorithms, pages 297–310. Springer, 2007. → pages 3, 57

[Hei46] Fritz Heider. Attitudes and cognitive organization. *The Journal of Psychology*, 21(1):107–112, 1946. → pages 1, 18, 20

[HKM05] Boulos Harb, Sampath Kannan, and Andrew McGregor. Approximating the best-fit tree under l p norms. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 123–133. Springer, 2005. → pages 62

[HLW02] Frank Harary, Meng-Hiot Lim, and Donald C Wunsch. Signed graphs for portfolio analysis in risk management. *IMA Journal of management mathematics*, 13(3):201–210, 2002. → pages 3, 57

[HPK11] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques.* Elsevier, 2011. → pages 16

[JGG+15] Songwei Jia, Lin Gao, Yong Gao, James Nastos, Yijie Wang, Xindong Zhang, and Haiyang Wang. Defining and identifying cograph communities in complex networks. *New Journal of Physics*, 17(1):013044, 2015. → pages 18

[Jor06] Jonathan Jordan. The degree sequences and spectra of scale-free random graphs. *Random Structures & Algorithms*, 29(2):226–242, 2006. → pages 27

[Jun78] Heinz A. Jung. On a class of posets and the corresponding comparability graphs. *Journal of Combinatorial Theory, Series B*, 24(2):125–133, 1978. → pages 17

[KM86] Mirko Křivánek and Jaroslav Morávek. Np-hard problems in hierarchical-tree clustering. *Acta Informatica*, 23(3):311–323, 1986. → pages 18, 62

[Kři88] Mirko Křivánek. The complexity of ultrametric partitions on graphs. *Information Processing Letters*, 27(5):265–270, 1988. → pages 62, 63, 64

[KRR+00] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D. Sivakumar, Andrew Tomkins, and Eli Upfal. Stochastic models for the web graph. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 57–65. IEEE, 2000. → pages 2, 9, 10, 27

[Kru56] Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956. → pages 14, 64

[KT06] Jon Kleinberg and Eva Tardos. *Algorithm Design*. Pearson Education India, 2006. → pages 4

[LHK10] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1361–1370, New York, NY, USA, 2010. ACM. → pages 1, 20

[LWGC12] Yunlong Liu, Jianxin Wang, Jiong Guo, and Jianer Chen. Complexity and parameterized algorithms for cograph editing. *Theoretical Computer Science*, 461:45–54, 2012. → pages 18

[Meg86] Nimrod Megiddo. *On the complexity of linear programming*. IBM Thomas J. Watson Research Division, 1986. → pages 64

[Mil67] Stanley Milgram. The small world problem. *Psychology Today*, 2(1):60–67, 1967. → pages 2, 27

[MM65] John W. Moon and Leo Moser. On cliques in graphs. *Israel journal of Mathematics*, 3(1):23–28, 1965. → pages 15

[MMP12] Kevin T. Macon, Peter J. Mucha, and Mason A Porter. Community structure in the united nations general assembly. *Physica A: Statistical Mechanics and its Applications*, 391(1):343–361, 2012. → pages 3, 57

[Mor34] Jacob Levy Moreno. *Who shall survive?: A new approach to the problem of human interrelations*. Nervous and Mental Disease Publishing Co, 1934. → pages 1

[MWW89] S. Ma, WD. Wallis, and J. Wu. Optimization problems on quasi-threshold graphs. *J. Comb. Inf. Syst. Sci*, 14:105–110, 1989. → pages 17

[NG13] James Nastos and Yong Gao. Familial groups in social networks. *Social Networks*, 35(3):439–450, 2013. → pages 18

[Pri57] Robert Clay Prim. Shortest connection networks and some generalizations. *Bell Labs Technical Journal*, 36(6):1389–1401, 1957. → pages 14

[SGWN11] Ajay Sridharan, Yong Gao, Kui Wu, and James Nastos. Statistical behavior of embeddedness and communities of overlapping cliques in online social networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 546–550. IEEE, 2011. → pages 11, 27

[SMKS03] Roded Sharan, Adi Maron-Katz, and Ron Shamir. Click and expander: a system for clustering and visualizing gene expression data. *Bioinformatics*, 19(14):1787–1799, 2003. → pages 16

[Sum74] David P. Sumner. Dacey graphs. *Journal of the Australian Mathematical Society*, 18(04):492–502, 1974. → pages 17

[TCAL16] Jiliang Tang, Yi Chang, Charu Aggarwal, and Huan Liu. A survey of signed network mining in social media. *ACM Computing Surveys (CSUR)*, 49(3):42, 2016. → pages 1

[WL93] Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 15(11):1101–1113, 1993. → pages 16

[Wol65] ES. Wolk. A note on" the comparability graph of a tree". *Proceedings of the American Mathematical Society*, 16(1):17–20, 1965. → pages 17

[WPLP14] Robert West, Hristo S. Paskov, Jure Leskovec, and Christopher Potts. Exploiting social network structure for person-to-person sentiment analysis. *arXiv preprint arXiv:1409.2450*, 2014. → pages 20

[WS98] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of small-worldnetworks. *Nature*, 393(6684):440–442, 1998. → pages 2, 7, 27

[YCC+96] Jing-Ho Yan, Jer-Jeong Chen, Gerard J. Chang, et al. Quasi-threshold graphs. *Discrete Applied Mathematics*, 69(3):247–255, 1996. → pages 17

[YCL07] Bo Yang, William Cheung, and Jiming Liu. Community mining from signed social networks. *IEEE transactions on knowledge and data engineering*, 19(10):1333–1348, 2007. → pages 57