Toward the Development of the Clustering Method for Large-scale Signed Graphs

Sun, Shurong
Graduate School of Economics, Kyushu University

https://doi.org/10.15017/7325746

出版情報：経済論究. 175, pp.11-35, 2023-07-26. 九州大学大学院経済学会
バージョン：
権利関係：

Toward the Development of the Clustering Method for Large-scale Signed Graphs

Shurong Sun †

Abstract

Signed graphs are graphs whose edges are assigned either a positive or negative sign. Many real-world relationships can be represented by signed graphs with positive and negative links. With the rise of social networks, signed graph clustering become an important task to be solved. Several methods have been proposed so far to solve the task of signed graph clustering. Especially, some methods that have applied spectral method have developed rapidly in recent years. In this paper, we first present a comprehensive overview of the existing spectral clustering methods for signed graph. Next, we collect eleven real-world networks of varying scales. Finally, we conduct numerical experiments to test the performance of each spectral clustering methods on real-world networks. From the experiment, we find that there is no very significant difference in performance between these methods when encountering relatively small-scale signed graphs. However, when the target of clustering is relatively large-scale graphs, some methods perform better according to the evaluation indicators, and some methods show a relatively good value of evaluation indicators but a very long execution time, which is not efficient enough, and some methods are quite uncompetitive in both evaluation indicators and execution time.

Keywords: Clustering, Signed Graph, Real-world Networks

1 Introduction

A wide range of topics on multidisciplinary can be represented as the analysis of graphs, e.g., neural science, social science, and pharmaceutical engineering. In general terms, nodes typically represent individuals or products, the edges connecting nodes typically represent some relationship between individuals (nodes). The graph connects individuals to individuals and builds a close link. In this paper, graph and network can be assumed to be synonymous. Community detection, which is also

† Graduate School of Economics, Kyushu University

called graph partition, is a worldwide task in network analysis, where the goal is to find groups of nodes that are, in some sense, more similar to each other than to the other nodes. By analyzing the results, it is possible to help us to reveal the hidden relations among the nodes in the graph.

Community structure is an important feature of complex networks, which indicates the fact that nodes are gathered into several groups (Chen et al., 2012), and each group is a community of nodes where the density of edges within communities is higher than among communities. Community detection technology is widely used in the analysis of social networks to identify people with common interests and keep them closely connected (Bedi and Sharma, 2016). Community detection can be used in machine learning to detect groups with similar properties and extract groups for some commonalities. Many different clustering algorithms have been proposed and implemented for community detection. Each of these has various pros and cons depending on the nature of the network as well as the domain of the applied problem.

In social network analysis, signed graph clustering is an important task to be solved by standard community detection algorithms. A signed graph is a graph in which each edge has a positive or negative sign, many real-world relationships can be represented by signed graphs with positive and negative links (Heider, 1946), such as like/dislike, trust/distrust, or friends/enemies. Therefore, community network analysis in signed graphs is receiving increasing multidisciplinary attention. Spectral clustering has been one of the widely used methods for community detection in networks. This gives us great motivation to explore the applicability of spectral methods to community detection on signed graphs.

In this paper, we present an overview of the existing community detection methods in signed graphs. We conduct extensive experiments on existing spectral clustering algorithms for signed graphs to compare and evaluate the effectiveness and time complexity of each method on real-world datasets. We find that the performance of the detection algorithm varies considerably depending on the number of groups detected and the volume of the datasets. When the number of detection groups is relatively small, the differences between methods are not very significant, but as the number of detection groups increases, some algorithms are unable to obtain results due to limitations, and in contrast, some algorithms show very competitive performance. Also, on relatively small-scale datasets, the performance between algorithms does not differ significantly regardless of the number of groups detected, but when large-scale datasets are encountered, some algorithms have good clustering results, but the execution time is quite long, while some algorithms show good clustering results in a relatively short time. Meanwhile, the eigenvalue decomposition of large-scale graphs by spectral methods poses great computational challenges, which prompted us to explore deeper in this study.

his paper is organized as follows. Section 2 summarizes some related work from unsigned network clustering and signed network clustering literature. Section 3 presents a comprehensive overview of four spectral clustering methods for the signed graph. Section 4 contains numerical experiments on various real-world datasets. Finally, Section 5 summarizes the findings from our experiments along with future research directions.

2 Related Work

2.1 Unsigned Network Analysis

The emergence of participatory networks and social media over the past decade has brought people together in many creative ways. (Tang and Liu, 2010) introduce the characteristics of social media from a data mining perspective and reviews representative tasks for computing using social media. They investigate graph-based community detection techniques and many important extensions to deal with dynamic, heterogeneous networks in social media, and show how the discovered community patterns can be used for social media mining.

When dealing with large-scale networks, the labels of subsets of graph vertices can indicate vertex characteristics. The core problem that arises from this is to use this information to expand the labels so that all vertices are assigned a label (or multiple labels). (Bhagat et al., 2011) explores classification techniques proposed for this problem, methods based on the iterative application of traditional classifiers using graph information as features, and methods for propagating existing labels through random walks. They take a common view of these methods to highlight the similarities between the different methods within and across the two categories.

(Liben-Nowell and Kleinberg, 2003) formalizes the interaction problem as a link prediction problem and develops link prediction methods based on a "proximity" measure of nodes in the network. Experiments on large collocated networks show that information about future interactions can be extracted from the network topology alone, and that fairly subtle measures for detecting node proximity can outperform more direct measures.

As the network evolves, the results of data mining algorithms such as community detection need to be updated accordingly. In addition, specific changes in the network structure need to be analyzed. Dynamic networks are networks that vary over time, the analysis of dynamic networks is particularly challenging, as it needs to be performed using online methods within the one-time constraints of data flow. The combination of contents can further increase the complexity of the evolutionary analysis process. (Aggarwal and Subbian, 2014) provides an overview of graph evolutionary analysis and the numerous applications that have emerged in different contexts.

(Sen et al., 2008) conduct four of the most widely used inference algorithms for classifying unsigned networked data and empirically comparing them to both synthetic and real-world data.

2.2 Signed Network Analysis

Since the rise of the research topic of signed network analysis, many aspects have been extended to study, such as signed network embedding (Huang et al., 2021), signed network clustering, prediction of signed links (Chiang et al., 2014), (Wang et al., 2017), and graph neural network for signed network (He et al., 2022). In this paper, we focus on the detection of multiple conflict groups in signed graphs, which belong to the study of signed network clustering.

The study by (Heider, 1946) we mentioned above represented a signed graph with positive and negative links. (Cartwright and Harary, 1956) developed a generalization of Heider's theory of balance by the use of concepts from the mathematical theory of linear graphs. (Davis, 1967) first defined clustering of signed graphs as partitioning the set of nodes into subsets such that each positive line connects two points in the same subset and each negative line connects two points from different subsets, it is clusterable of a graph if the node set can be divided into $k(\geq 2)$ subsets, where any two subsets are mutually hostile. There are various ways for clustering signed graphs. Correlation clustering (Bansal et al., 2004), also called CC method, considers clustering problem for the complete graph as an "agnostic learning" problem, then presents a constant-factor approximation for minimizing disagreements (the number of negative edges inside cluster plus the number of positive edges between clusters), and a PTAS (polynomial-time approximation scheme) for maximizing agreements (the number of positive edges inside cluster plus the number of negative edges between clusters), for the problem of clustering vertices in the signed graph. Minimizing the correlation clustering error is hard, (Ailon et al., 2008) propose a algorithm for approximating CC is KwikCluster. On any instance of CC, KwikCluster achieves an expected error bounded by 3OPT. (Yang et al., 2007) propose a new algorithm, called FEC, to mine signed social networks so that both positive within-group relations and negative between-group relations are dense.

(Hou, 2005) extends some fundamental concepts of the Laplacian matrices from graphs to signed graphs and investigates the relationships between the least Laplacian eigenvalue and the imbalance of signed graphs. (Kunegis et al., 2010) demonstrate that signed Laplacian is positive semidefinite and, in some cases, positive definite, thus guaranteeing signed Laplacian is a suitable basis for spectral clustering. Moreover, spectral clustering using the signed Laplacian is shown to be equivalent to the k-way signed ratio cut problem, which counts positive edges between clusters and negative edges within clusters. (Anchuri and Magdon-Ismail, 2012) develop a spectral method enhanced by iterative optimization that extends the existing community detection algorithm, which is only applicable to unsigned networks, to apply to signed networks. (Chiang et al., 2012) find that the signed Laplacian suffers from weakness when they apply them to the k-way clustering problem. Thus, they propose a new signed cut objective, which is called balanced normalized cut (BNC). (Jiang, 2015) proposes an extension of the stochastic block model (SSBM) to study the mesoscopic structural patterns in signed networks. (Cucuringu et al., 2019) propose a principled spectral algorithm (SPONGE) for clustering signed graphs, that amounts to solving a generalized eigenvalue problem. (Mercado et al., 2019) extend spectral clustering to signed graphs via the one-parameter family of Signed Power Mean Laplacians, which is defined as the matrix power mean of normalized standard and signless Laplacians of positive and negative edges. (Hua et al., 2020) employ a *random walk gap* (RWG) as a subroutine and use the RWG matrix to reweight the edges, extract more cluster structure information, and propose a fast clustering for signed graphs (FCSG) algorithm.

2.3 Application Fields

Signed graphs analysis has many applications from modeling users' interactions in social media (Slashdot technology news site) (Kunegis et al., 2009), to mining the task of link analysis in real-world relations (Beigi et al., 2016), to investigating the influence of diffusion dynamics and influence maximization in online social networks (Li et al., 2013), to learning to recommend products with trust and distrust relationships in e-commerce sites (Ma et al., 2009), to providing the experimental study of using distrust in the recommendation process (Victor et al., 2011), to estimating the dynamics for social networks (Antal et al., 2006), and to classifying the modular structure of complex system composed of close-knit communities (Marvel et al., 2009).

3 Analysis of Methods for Signed Graphs

3.1 Basic Knowledge of Signed Graph

A **graph** is a pair $G=(V, E)$, where V is a set whose elements are called vertices, and E is a set of paired vertices, whose elements are called edges. In this paper, graph and network can be assumed to be synonymous. A **weighted graph** or a network is a graph in which a number (the weight) is assigned to each edge, graphs can be weighted or unweighted. Compare to an **unsigned graph** shown in Figure 1, a **signed graph**, edges are assigned $+1$ or -1 weights, as illustrated in Figure 2. In graph theory, a **dense graph** is a graph in which the number of edges $|E|$ is close to the maximal number of edges. The opposite, a graph with only a few edges, is a **sparse graph**. A **simple graph** is a graph that does not have more than one edge between any two vertices and no edge starts and ends at the same vertex. In other words, a graph without loops and multiple edges is a simple graph. The graph ***density*** of simple graphs is defined to be the ratio of the number of edges $|E|$ concerning the maximum possible edges. For undirected simple graphs, the graph density is represented by Eq.1.

$$D=\frac{2|E|}{|V|(|V|-1)}, \tag{1}$$

where $|E|$ is the number of edges and $|V|$ is the number of vertices in the graph. The maximum

Figure 1: Unsigned Graph

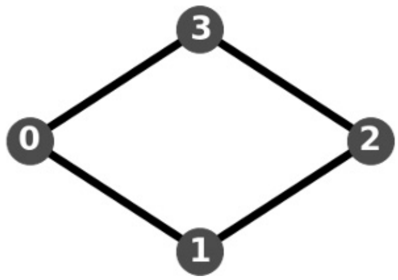
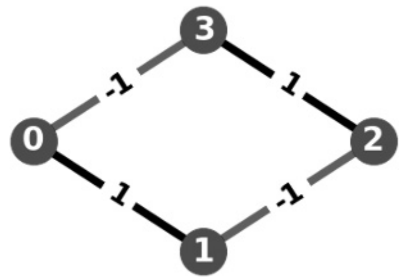


Figure 2: Signed Graph



number of edges for an undirected simple graph is $\frac{|V|(|V|-1)}{2}$, so the maximal density is 1 (for complete graphs) and the minimal density is 0 (Coleman and Moré, 1983). In this paper, we use several real-world networks to evaluate the effectiveness and scalability of clustering algorithms for the signed graph.

Several algorithms have been proposed for clustering signed graphs. (Tang et al., 2016) have given a comprehensive survey on the research topic. In this section, we will study the spectral methods of signed graphs in detail. A simple example of clustering signed graph is illustrated in Figure 3, 4. The signed graph of Figure 3 can be clustered intuitively, and the result with two groups is obtained in Figure 4, within groups are all positive edges while across groups are all negative edges.

3.2 Signed Laplacian Matrix

Spectral clustering begins by finding the Laplacian of the matrix representation of graphs, then the eigenvalues are computed for the Laplacian. Note the eigenvalues of the Laplacian are assumed non-negative and real, in other words, the Laplacian is positive semidefinite, which guarantees the

Figure 3: Signed Graph

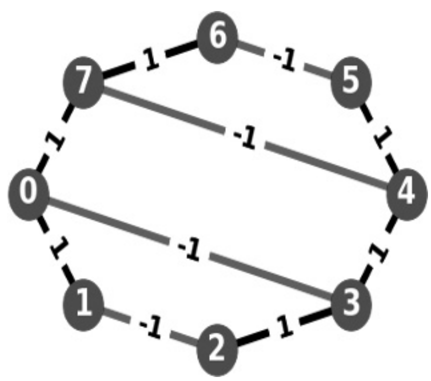
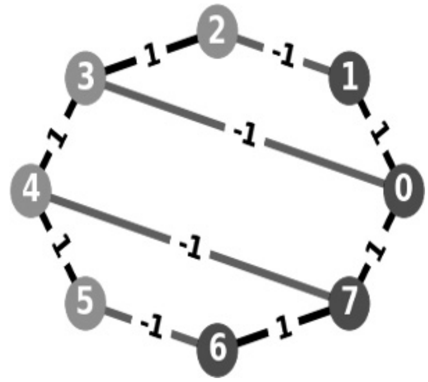


Figure 4: Clustering



effectiveness of computation. The Laplacian matrix of unsigned graph G is given by $L=D-A$, where the adjacency matrix A is defined as follows:

$$A_{ij}=\begin{cases} >0, & \text{if relationship of } i,\ j \text{ is positive,} \\ <0, & \text{if relationship of } i,\ j \text{ is negative,} \\ =0, & \text{if relationship of } i,\ j \text{ is unknown.} \end{cases}$$

A^{+}, A^{-} denote positive edge-only adjacency matrix and the negative edge-only adjacency matrix. Here, $A_{ij}^{+}=\max\{A_{ij}, 0\}$ and $A_{ij}^{-}=\max\{-A_{ij}, 0\}$. And D is the diagonal degree matrix defined by $D_{ii}=\sum_{j=1}^{n}Aij$. D^{+}, D^{-} denote the diagonal matrix with degrees $D_{ii}^{+}=\sum_{j=1}^{n}A_{ij}^{+}$, $D_{ii}^{-}=\sum_{j=1}^{n}A_{ij}^{-}$. The standard Laplacian matrix of a signed graph is indefinite and does not yield real, non-negative eigenvalues. Thus, (Kunegis et al., 2010) propose a modified signed Laplacian matrix.

Definition 1 *The signed Laplacian matrix $\overline{L}$ of a graph G with adjacency matrix A is given by*

$$\overline{L}=\overline{D}-A \tag{2}$$

where the signed degree matrix $\overline{D}$ is the diagonal matrix given by $\overline{D}_{ii}=\sum_{j=1}^{n}|A_{ij}|$. (Kunegis et al., 2010) prove the signed Laplacian matrix is positive-semidefinite and extends two normalized Laplacian to signed graphs, which are random walk Laplacian and symmetric normalized Laplacian matrices.

Definition 2 *The random walk normalized Laplacian for signed graphs is given by*

$$\overline{L}_{rw}=I-\overline{D}^{-1}A \tag{3}$$

Definition 3 *The symmetric normalized Laplacian for signed graphs is given by*

$$\overline{L}_{sym}=\overline{D}^{-1/2}\overline{L}\overline{D}^{-1/2}=I-\overline{D}^{-1/2}A\overline{D}^{-1/2} \tag{4}$$

Spectral clustering algorithms are usually derived by formulating a minimum cut problem. Let $G=(V, \mathrm{E})$ be an unsigned graph, a cut of G is a partition of the vertices $V=X\cup Y$. The ratio cut of unsigned graph is given by

$$RatioCut(X, Y)=cut(X, Y)\left(\frac{1}{|X|}+\frac{1}{|Y|}\right) \tag{5}$$

where the $cut(X, Y)=\sum_{i\in X, j\in Y}A_{ij}$.

Definition 4 *For a signed graph G, A^{+} and A^{-} for the adjacency matrices containing only positive and negative edges. the signed ratio cut is defined by*

$$SignedRatioCut(X, Y)=scut(X, Y)\left(\frac{1}{|X|}+\frac{1}{|Y|}\right) \tag{6}$$

where

$$scut(X, Y)=2cut^{+}(X, Y)+cut^{-}(X, Y)+cut^{-}(Y, Y),$$
$$cut^{+}(X, Y)=\sum_{i\in X, j\in Y}A_{ij}^{+},\quad cut^{-}(X, Y)=\sum_{i\in X, j\in Y}A_{ij}^{-}.$$

Definition 5 *The signed normalized cut is given by*

$$SignedNormalizedCut(X, Y)=scut(X, Y)\left(\frac{1}{vol(X)}+\frac{1}{vol(Y)}\right) \tag{7}$$

normalized with the number of edges $vol(X)$ instead of normalizing with the number of vertices $|X|$.

3.3 Balanced Normalized Cut

By considering the signed graph Laplacian, Kunegis et al. in Section 3.2 extend the spectral clustering algorithm to the signed graph setting, and this is also equivalent to finding the clusters that minimize 2-way signed ratio cuts. (Chiang et al., 2012) encounter a fundamental weakness when they directly extended to k-way clustering problem. To overcome this weakness, they propose new k-way objectives for signed graphs.

First, consider the graph cuts on unsigned graphs, the ratio cut objective is defined as:

$$\min_{(x_1,\ldots,x_k)\in I}\left(\sum_{c=1}^{k}\frac{x_c^T L x_c}{x_c^T x_c}\right) \tag{8}$$

where the constraint $(x_1, \ldots, x_k)\in I$ means $(x_1, \ldots, x_k)$ is a k-cluster indicator set. Formally, an indicator set $(x_1, \ldots, x_k)$ is defined as a set of k vectors $x_1, \ldots, x_k$, such that:

$$x_c(i)=\begin{cases}1, & \text{if node } i\in\pi_c,\\ 0, & \text{otherwise.}\end{cases}$$

with $|\pi_c|>0$, where π_c denotes the c^{th} cluster. The graph Laplacian is defined by $L=D-A$.

By directly extending Eq.8 to the following objective:

$$\min_{(x_1,\ldots,x_k)\in I}\left(\sum_{c=1}^{k}\frac{x_c^T \overline{L} x_c}{x_c^T x_c}\right) \tag{9}$$

where the $\overline{L}$ is signed Laplacian. Chiang et al. have proved that minimizing the k-way signed ratio cut will not lead to an optimal solution. Therefore, they propose a series of new criteria and objectives, including graph association and cut, for general k-way signed graph clustering.

・Positive/Negative Ratio Association

The *positive ratio association* aims to maximize the number of positive edges within each cluster relative to the cluster's size:

$$\max_{(x_1,\ldots,x_k)\in I}\left(\sum_{c=1}^{k}\frac{x_c^T A^+ x_c}{x_c^T x_c}\right) \tag{10}$$

The *negative ratio association* minimizes the number of negative edges within each cluster relative to the cluster's size:

$$\min_{(x_1,\ldots,x_k)\in I}\left(\sum_{c=1}^{k}\frac{x_c^T A^- x_c}{x_c^T x_c}\right) \tag{11}$$

・Positive/Negative Ratio Cut

The *positive ratio cut* minimizes the number of positive edges between clusters:

$$\min_{(x_1,\ldots,x_k)\in I}\left(\sum_{c=1}^{k}\frac{x_c^T L^+ x_c}{x_c^T x_c}\right) \tag{12}$$

The *negative ratio cut* maximizes the number of negative edges between clusters:

$$\max_{(x_1,\ldots,x_k)\in I}\left(\sum_{c=1}^{k}\frac{x_c^T L^- x_c}{x_c^T x_c}\right) \tag{13}$$

・Balance Ratio Cut/Association

The *balance ratio cut* is defined as a combination of positive ratio cut and negative ratio association as follows:

$$\min_{(x_1,\ldots,x_k)\in I}\left(\sum_{c=1}^{k}\frac{x_c^T (D^+ - A) x_c}{x_c^T x_c}\right) \tag{14}$$

The *balance ratio association* is defined as a combination of negative ratio cut and positive ratio association with equal weight as follows:

$$\max_{(x_1,\ldots,x_k)\in I}\left(\sum_{c=1}^{k}\frac{x_c^T (D^- + A) x_c}{x_c^T x_c}\right) \tag{15}$$

・Balance Normalized Cut

Considering the objectives normalized by cluster volume instead of by the number of nodes in the clusters. Chiang et al. extend Eq.14 to *balance normalized cut* as follows:

$$\min_{(x_1,\ldots,x_k)\in I}\left(\sum_{c=1}^{k}\frac{x_c^T (D^+ - A) x_c}{x_c^T \bar{D} x_c}\right) \tag{16}$$

Similarly, the *balance normalized association* can be defined as follow:

$$\max_{(x_1,\ldots,x_k)\in I}\left(\sum_{c=1}^{k}\frac{x_c^T (D^- + A) x_c}{x_c^T \bar{D} x_c}\right) \tag{17}$$

In the above definition, A, A^+, A^- represent the full adjacency matrix, the positive edge-only adjacency matrix, and the negative edge-only adjacency matrix respectively. D, D^+, D^- represent the diagonal matrix, the positive edge-only diagonal matrix, and the negative edge-only diagonal matrix respectively. and $L=D-A$, $L^+=D^+-A^+$, $L^-=D^--A^-$ respectively. The property minimizing the balance normalized cut is equivalent to maximizing the balance normalized association. Thus, the choice between balance normalized cut and association is inconsequential (Chiang et al., 2014).

3.4 Signed Positive Over Negative Generalized Eigenproblem

(Cucuringu et al., 2019) propose a principled spectral algorithm (SPONGE) for clustering signed graphs, that amounts to solving a generalized eigenvalue problem to find the k smallest eigenvectors before k-means clustering. In particular, SPONGE can recover clustering for very sparse graphs and a large number of clusters.

For a given signed graph G with adjacency matrix A, the objective function for SPONGE is derived from the positive-edges subgraph normalized cut $ncut(G^+)$ and the negative-edges subgraph inverse normalized cut $(ncut(G^-))^{-1}$. $G^+=(V, E^+)$ (resp. $G^-=(V, E^-)$) denote the unsigned subgraphs of positive (resp. negative) edges with adjacency matrices A^+ (resp. A^-), such that $A=A^+-A^-$. Moreover $E^+\cap E^-=\emptyset$, and $E^+\cup E^-=E$. τ^+, τ^- denote the trade-off or regularization parameters. A natural extension to $k>2$ disjoint clusters $C_1, \ldots, C_k$ leads to the following discrete optimization problem:

$$\min_{C_1,\ldots,C_k}\sum_{i=1}^{k}\frac{cut_{G^+}(C_i, \overline{C}_i)+\tau^- vol_{G^-}(C_i)}{cut_{G^-}(C_i, \overline{C}_i)+\tau^+ vol_{G^+}(C_i)} \tag{18}$$

where $C_1, \ldots, C_k$ represents a partitioning of G. $cut_G(C, \overline{C})=\sum_{i\in C, j\in\overline{C}}A_{ij}$ is the total weight of edges crossing from C to $\overline{C}$. The volume of C: $vol_G(C)=\sum_{i\in C}\sum_{j=1}^{n}A_{ij}$ is the sum of degrees of nodes in C. Cucuringu et al. demonstrate the objective function is equivalent to the discrete optimization problem:

$$\min_{C_1,\ldots,C_k}\sum_{i=1}^{k}\frac{x_{C_i}^T(L^+ +\tau^- D^-)x_{C_i}}{x_{C_i}^T(L^- +\tau^+ D^+)x_{C_i}} \tag{19}$$

where L^+ denotes the Laplacian of G^+ (resp. G^-), and D^+ (resp. D^-) denotes a diagonal matrix with the degrees of G^+ (resp. G^-), which is an NP-hard problem. Therefore, they introduce a new set of vectors $z_1, \ldots, z_k\in\mathbb{R}^n$, such that they are orthonormal with respect to $L^-+\tau^+D^+$, i. e., $z_i^T(L^-+\tau^+D^+)z_i=1$ and $z_i^T(L^-+\tau^+D^+)z_j=0$ for $i\neq j$. This leads to the modified version of the objective function as follows:

$$\min_{C_1,\ldots,C_k}\sum_{i=1}^{k}\frac{z_i^T(L^+ +\tau^- D^-)z_i}{z_i^T(L^- +\tau^+ D^+)z_i} \tag{20}$$

Objective function in Eq.20 can be formulated as the generalized eigenproblem whose solution is given by the eigenvectors of $(L^-+\tau^+D^+)^{-1/2}(L^++\tau^-D^-)(L^-+\tau^+D^+)^{-1/2}$ (Sameh and Tong, 2000). They also consider the variant of SPONGE, namely SPONGE$_{sym}$, where the embedding is generated using the smallest k generalized eigenvectors of $(L_{sym}^+ +\tau^- I, L_{sym}^- +\tau^+ I)$, wherein $L_{sym}^+=(D^+)^{-1/2}L^+(D^+)^{-1/2}$ is the so-called symmetric Laplacian of G^+, Similarly for L_{sym}^-.

3.5 *k*-Oppositive Cohesive Groups Detection

(Chu et al., 2016) propose a new problem in finding *k*-Oppositive Cohesive Groups in the signed graph. A *k*-OCG is a set of k subgraphs, each subgraph is cohesive, i.e., has high intra-subgraph cohesion, and the subgraphs are opposite to each other, i.e., there is high inter-subgraph opposition among the subgraphs. They show that the classical Proximal Gradient method (PG) (Boyd et al., 2014) is very costly in finding *k*-OCGs on large signed networks since it has to use the entire adjacency matrix. They develop a two-phase iterative method FOCG that is experimentally two orders of magnitudes faster than PG on average. Given a signed graph $G=(V, E)$, *k*-subgraph set denoted by $S=\{S_1, S_2, \ldots, S_k\}$, which is a set of k subgraphs. A subgraph G_S is represented by $G_S=(V_S, E_S)$, where $V_S=\{v_i|i\in S\}$, $E_S=\{(v_i, v_j)|i, j\in S\}$. A common way to represent a subgraph G_S is to associate it with a non-negatively-valued n-dimensional column vector $x=[x_1, x_2, \ldots, x_n]$ in the standard simplex $\triangle^n$, where the i-th dimension x_i indicates the participation of vertex v_i in G_S, that is, the weight of vertex v_i in G_S, and simplex $\triangle^n=\{x|\sum_{i=1}^{n}x_i=1, x_i>0\}$. Each subgraph $S_j\in S(1\leq j\leq k)$ by an n-dimensional column vector $X_j\in\triangle^n$. A k-oppositive cohesive groups is a k-subgraph set that has a large intra-subgraph cohesion $g^+(S)=\sum_{j=1}^{k}g^+(X_j)=tr(x^TA^+X)$ and a large inter-subgraph opposition $g^-(S)=\sum_{h\neq j}g^-(X_h, X_j)=\overline{tr}(x^TA^-X)$. Accordingly, they define the k-OCG detection problem as:

$$\begin{aligned}&\max_{X\in\triangle n\times k}\quad F(X)\\&F(X)=g^+(S)+\alpha g^-(S)-\beta\overline{tr}(X^TX)\\&\qquad=tr(X^TA^+X)+\overline{tr}(X^THX)\end{aligned}\tag{21}$$

where the $H=\alpha A^- -\beta I$, α controls the tradeoff between intra-subgraph cohesion and inter-subgraph opposition, and the term $\beta tr(X^TX)$ penalizes the overlap between different cohesive subgraphs. The classic numerical optimization methods are not efficient in solving the problem in Eq.21 on large graphs. Therefore, Chu et al. propose the FOCG algorithm that solves the problem in Eq.21 by iteratively seeking the KKT points (Bertsekas, 1997) of $F(X)$, and achieves high efficiency by confining all iterates on small subgraphs. Re-organize Eq.21 For any $j\in[1, k]$,

$$F(X)=f(X_j)+\sum_{h\neq j}X_h^TA^+X_h+\sum_{l\neq j}\sum_{h\neq j,\, h\neq l}X_l^THX_h\tag{22}$$

where $f(X_j)=X_j^TA^+X_j+2\sum_{h\neq j}X_j^THX_h$ consists of all the terms in $F(X)$ that are related with X_j.

Algorithm 1: The IOCG Algorithm (Chu et al., 2016)

Input: Adjacency matrices of A^+ and A^-.
Output: An initialization $X(0)$ for FOCG algorithm.
Initialize the seed index set $\eta=\emptyset$
Calculate the vertex degree vector on the positive network as $d^+=[d_1^+, \ldots, d_n^+]$, where $d_i^+=\sum_{j=1}^{n}A_{i,j}^+$.
Select the first seed $h_1=Roul(d^+)$ by *roulette wheel selection* and update $\eta\leftarrow\eta\cup h_1$
for $l\in[2, k]$ **do**
  Calculate opposition vector $o^-=[o_1^-, \ldots, o_i^-, \ldots, o_n^-]$, where $o_i^-=\frac{1}{|\eta|}\sum_{j\in\eta}A_{i,j}^-$.
  Select seed by $h_l=Roul(o^-)$ and update $\eta\leftarrow\eta\cup h_l$
end for
Initialize X as a n-by-k dimensional all zero matrix.
for $j\in[1, k]$ **do**
  Set $X_{\eta_j,j}=1$, where η_j is the j-th seed index in η.
end for
return $X(0)\leftarrow X$.

Thus, the KKT points of $F(X)$ by optimizing $f(X_j)$ on each column of X, the corresponding optimization problem is

$$\max_{X_j \in \triangle^n} f(X_j) \tag{23}$$

where $f(X_j) = X_j^T A^+ X_j + 2X_j^T M_j$ and $M_j = \sum_{h \neq j} HX_h$ is an n-dimensional column vector. **Algorithm 1** introduces the complete algorithm of kOCG that randomly selects k seed vertices as the initializations for the k subgraphs $S_i\,(1 \leq i \leq k)$ in S. The *roulette wheel selection* method $h = Roul(d^+)$ randomly selects a vertex v_h with probability $\frac{d_h}{\sum_{i=1}^n d_i}$. This approach finds opposite groups only in local regions and is sensitive to initialization.

3.6 Spectral Conflicting Group Detection

From the experience of spectral methods in signed graph, (Bonchi et al., 2019) illustrate the problem of discovering two polarized (opposite) communities in the signed graph as a Rayleigh quotient problem (**Problem 1**), and refer to the objective function of problem as polarity. The algorithm they proposed named EIGENSIGN, works by simply discretizing the entries of the eigenvector of the adjacency matrix corresponding to the largest eigenvalue. EIGENSIGN generally outputs a solution comprised of all the vertices in the graph, unless some components of the eigenvector are exactly zero. To overcome this issue they propose a randomized algorithm, RANDOM-EIGENSIGN, which also computes the first eigenvector, i.e., v, of the adjacency matrix. Instead of simply discretizing the entries of v, it randomly sets each entry of x to 1 or -1 with probabilities determined by the entries of v.

> **Problem 1**: *2-Polarized-Communities:*
>
> Given a signed graph $G = (V, E_+, E_-)$ with n vertices and signed adjacency matrix A, find a vector $x \in \{-1, 0, 1\}^n$ that maximizes
>
> $$\frac{x^T A x}{x^T x}$$

(Tzeng et al., 2020) show that the proposed objective can be interpreted in terms of the Laplacian of a complete graph, and rely on the spectral properties of this matrix to derive a novel optimization framework, spectral conflicting group detection (SCG). By carefully examining the invariant subspaces of the aforementioned Laplacian, they reformulate the problem as a maximum discrete Rayleigh quotient (MAX-DRQ) objective, which is an APX-hard problem. And then propose two algorithms, one deterministic, and one randomized with approximation guarantees.

Given a signed graph $G = (V, E)$ and an integer k. $E(V_1, V_2)$ denote the set of edges between two subsets $V_1, V_2 \subseteq V$, where V_1, V_2 are not required to be disjoint, $E(V_1)$ to be $E(V_1, V_1)$ for any $V_1 \subseteq V$. To find k mutually-disjoint node sets $S_1, \ldots, S_k \subseteq V$, the objective function derived from (Bonchi et al., 2019) is

$$f(S_1, \ldots, S_k) = \sum_{h \in [k]} \sum_{(i,j) \in E(S_h)} A_{i,j} + \frac{1}{k-1} \sum_{h, l \in [k], h \neq l} \sum_{(i,j) \in E(S_h, S_l)} (-A_{i,j}) \tag{24}$$

Eq.24 shows for all $i, j \in [k]$, with $i \neq j$, the edges in $E(S_i)$ are mostly positive, whereas the edges in $E(S_i, S_j)$ are mostly negative. Rewritten the Eq.24 by introducing an indicator matrix $X \in \{0, 1\}^{n \times k}$ with $X_{i,j}=1$ if node $i \in S_j$ and 0 otherwise, the objective function become

$$f(S_1, \ldots, S_k) = \langle A, XX^T \rangle_F - \frac{\langle A, XJ_kX^T \rangle_F - \langle A, XX^T \rangle_F}{k-1} = \frac{\langle A, XL_kX^T \rangle_F}{k-1} \tag{25}$$

where $L_k = kI_k - J_k$. Let $L_k = UDU^T$ be the eigendecomposition of L_k, $D = diag([0, k, \ldots, k] \in \mathbb{R}^{k \times k}$, and $U \in \mathbb{R}^{k \times k}$ is a real-valued orthogonal matrix. As the geometric multiplicity of eigenvalue k is $k-1$, the matrix U is not unique. They strictly the choice of U to be the following

$$\begin{aligned} &(U_{:,1})^T = 1\sqrt{k}[1, \ldots, 1] && (U_{:,2})^T = c_1[k-1, -1, \ldots, -1] \\ &(U_{:,3})^T = c_2[0, k-2, -1, \ldots, -1], \ldots, && (U_{:,k})^T = c_{k-1}[0, \ldots, 0, -1, 1] \end{aligned} \tag{26}$$

where $c_i = 1/\sqrt{(k-i+1)(k-i)}$, for $i=1, \ldots, k-1$. Change the variables $Y = XU$ and rewritten the Eq.25 as

$$\langle A, XL_kX^T \rangle_F = \langle A, YDY^T \rangle_F = \langle A, Ydiag([0, k, \ldots, k])Y^T \rangle_F = ktr((Y_{:,2:}^T A(Y_{:,2:})) \tag{27}$$

The total number of nodes in the groups $S_1, \ldots, S_k$ is

$$\sum_{i \in [k]} |S_i| = tr(Y^TY) = k(Y_{:,1})^T(Y_{:,1}) = \frac{k}{k-1} tr((Y_{:,2:})^T(Y_{:,2:})) \tag{28}$$

Finally replacing the constraints on the indicator matrix X with the constraint that the rows of Y should take values in the set $\{0, U_{1,:}, \ldots, U_{k,:}\}$, the objective function of this problem can be represented as follow:

$$\max_{Y \in \mathbb{R}^{n \times k} \setminus \{0\}} \frac{tr((Y_{:,2:})^T A(Y_{:,2:}))}{tr((Y_{:,2:})^T(Y_{:,2:}))} \tag{29}$$

subject to $Y_{i,:} \in \{0, U_{1,:}, \ldots, U_{k,:}\}$, for all $i=1, \ldots, n$

The efficient spectral **Algorithm 2** by leveraging the problem formulation Eq.(29). Round(v, q) denote to round v to a discrete vector $r \in \{0, 1, -q\}^n$. The algorithm for Finding the maximum discrete Rayleigh quotient (Solve-Max-DRQ) is shown in **Algorithm 3**.

Algorithms 2: Spectral Conflicting Group detection SCG(A, k)

Input : A is the adjacency matrix of the signed network; k is the number of groups.
Output: Groups $S_1, \ldots S_k$.
$A^{(0)} \leftarrow A$;
for $t=1, \ldots, k-1$ **do**
  $r^{(t)} \leftarrow$ Solve-Max-DRQ $(A^{(t-1),k-t})$;
  if $t<k-1$ **then**
    $S_t \leftarrow \{i \notin \cup_{j=1}^{t-1} S_j : |r_i^{(t)}|=(k-t)\}$;
    $A^{(t)} \leftarrow A^{(t-1)}$;
    $A_{i,:}^{(t)} \leftarrow 0_{1\times n}$ and $A_{:,i}^{(t)} \leftarrow 0_{n\times 1}$ for all $i \in S_t$;
  else $S_{k-1} \leftarrow \{i \notin \cup_{j=1}^{t-1} S_j : r_i^{(t)}=1\}$ and $S_k \leftarrow \{i \notin \cup_{j=1}^{t-1} S_j : r_i^{(t)}=-1\}$;
end
return $S_1, \ldots, S_k$;

Algorithms 3: Solve-Max-DRQ (A, q)

Input : Square and symmetric matrix A, and positive integer q.
Output: The rounded vector $r \in \{0, 1, -q\}$.
$v \leftarrow$ the leading eigenvector of A;
$(d_1, r_1) \leftarrow$ Round(v, q); $//d_1=\sin\theta(v, r_1)$
$(d_2, r_2) \leftarrow$ Round$(-v, q)$; $//d_2=\sin\theta(-v, r_2)$
if $d_1 \leq d_2$ **then** $r \leftarrow r_1$;
else $r \leftarrow r_2$;
return r;

To make the method scalable, Tzeng et al. also propose a deterministic rounding by minimum-angle heuristic in **Algorithm 4**, which finds a local optimum solution. Let v be the leading eigenvector of $A^{(t-1)}$, in other words, the real-valued maximizer of the quotient $x^T A^{(t-1)} x/(x^T x)$. This idea is to round v to a discrete vector $u^* \in \{0, -1, q\}^n$ that minimizes $\sin\theta(u, v)$, among all vectors $u \in \{0, -1, q\}^n$.

They also give an algorithm for maximizing $x^T A^{(t-1)} x/(x^T x)$ in $\{0, -1, q\}^n$ is a randomized-rounding scheme starting with the eigenvector v of $A^{(t-1)}$. Round v onto $\{0, -1, q\}^n$ by drawing Bernoulli trials. Pseudocode is shown in **Algorithm 5**.

Algorithms 4: MinAngleRound (v, q)

Input : Vector $v \in \mathbb{R}^n$ and positive integer q.
Output: Vector $u^* \in \{0, -1, q\}^n$ with min angle to v.
$\{i_k\}_{k=1}^n \leftarrow$ Sort v and return the indexes such that $v_{i_1} \geq \ldots \geq v_{i_n}$;
$(d, u^*) \leftarrow (\infty, 0)$;
$(k_1, k_2) \leftarrow (0, n+1)$;
while $k_1 < k_2$ **do**
    $u_1 \leftarrow$ set the i_{k_1+1}-th element of u^* to q;
    $u_2 \leftarrow$ set the i_{k_2+1}-th element of u^* to -1;
    if $\min\{\sin\theta(v, u_1), \sin\theta(v, u_2)\} \geq d$ **then** break;
    if $\sin\theta(v, u_1) < \sin\theta(v, u_2)$ **then** $(k_1, d, u^*) \leftarrow (k_1+1, \sin\theta(v, u_1), u_1)$;
    else $(k_2, d, u^*) \leftarrow (k_2-1, \sin\theta(v, u_2), u_2)$;
end
return (d, u^*);

Algorithms 5: RandomRound(v, q)

Input : Vector $v \in \mathbb{R}^n$ and positive integer q.
Output: Vector $u^* \in \{0, -1, q\}^n$, a randomized rounded vector v.
$u \leftarrow 0$;
for $i=1, \ldots, n$ **do**
    If $v_i > 0$ **then** $u_i \leftarrow q$ Bernoulli$(|v_i|/q)$;
    else if $v_i < 0$ **then** $u_i \leftarrow -1$ Bernoulli$(|v_i|)$;
end
$d \leftarrow \sin\theta(v, u)$;
return (d, u);

4 Experimental Evaluation in Real-world Datasets

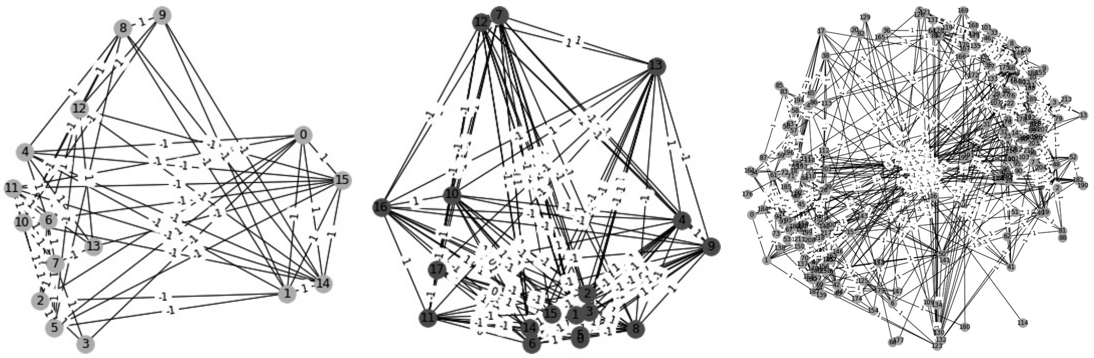
In this section, we conduct a comprehensive study of the state-of-the-art signed graph clustering method on real-world datasets of varying complexity and evaluate the effectiveness and scalability of each method. We will follow a similar evaluation setup by (Tzeng et al., 2020). To give a clearer comparison, we add three datasets, Highland, Sampson, and ConVote with $|V|=16,18,219$ respectively. Then test the performance of each algorithm for the signed graph on more groups of $k=8$ and $k=20$, we give the intuitively execution time in section 4.2.2 after testing the polarity (Eq.29) of each method. All experiments are executed by Python3 on an 8-core Apple M1-Chip personal computer at 3.2GHz with 16GB RAM.

4.1 Real-World Datasets

Highland This is a clusterable signed graph illustrated by HageHarary using the Gahuku-Gama system described by Read for the Eastern Central Highlands of New Guinea (Hage and Harary,

Figure 5: Three real-world signed graphs

(a) Highland Tribes　　(b) Sampson　　(c) Congress Vote



1983), (Read, 1954). Nodes represent tribes, and edges represent friendships. According to the relationship between vertices and edges, the Highland tribes' dataset is shown in Figure 5a.

Sampson The Sampson dataset records social interactions between a group of monks residing as visual experimenters containing monks associated with a crisis in a New England (USA) cloister (or monastery) and collects several sociometric rankings (Sampson, 1968), with the crisis leading to the departure of several monks. Vertices represent monks, and edges represent ratings. Sampson is shown in the Figure 5b.

ConVote This dataset is built from congressional speech data (Thomas et al., 2006), vertices represent politicians speaking in the United States Congress, and edges denote that a speaker mentions another speaker. Congress vote is shown in Figure 5c.

WoW-EP8 This is a dataset of legislative amendments collected from the European Parliament website (Kristof et al., 2020). The dataset spans the 8th legislature (EP8) from 2014 to 2019. Vertices represent parliament representatives (MEPs, for Member of the European Parliament), and edges represent attitude (accepted or rejected).

Bitcoin It is a who-trusts-whom network of people who use Bitcoin to transact on a platform called Bitcoin OTC. Since Bitcoin traders are anonymous, there is a need to maintain a record of the user's reputation to prevent transactions with fraudulent and risky users (Leskovec and Krevl, 2014). Vertices represent Bitcoin traders, Positive edges denote trust, and negative edges denote distrust.

WikiVote For a user to become an administrator, a Request for adminship (RfA) is issued and the Wikipedia community via a public discussion or a vote decides who to promote to adminship (Leskovec and Krevl, 2014). The vertices in this network represent Wikipedia users, and the directed edges from vertex i to vertex j represent user i voting on user j.

Referendum In the past few years, when people focus on the field of online political debate, interest in automatically categorizing users' positions on a given entity, such as a controversial topic or a politician, has grown significantly in polarized debates. The dataset has been collected from a corpus of Italian tweets where users were discussing a highly polarized debate in Italy, i.e. the 2016 referendum on the reform of the Italian Constitution (Lai et al., 2018). Edge signs indicate if two users(vertices) are classified to have the same stance or not.

Slashdot This is a signed social network for users of the technology news site Slashdot (slashdot.org), connected by directed "friend" and "foe" relationships. The "friend" and "foe" tags are used on Slashdot to mark users and influence the scores as seen by each user (Leskovec and Krevl, 2014).

WikiCon Wikipedia conflict collects positive and negative conflicts between users editing the English Wikipedia (Kunegis, 2013). Vertices represent users and edges represent a conflict between two users, with the edge sign representing positive and negative interactions.

Epinions This is an online social network of who trusts whom for the general consumer review site Epinions.com. The network consists of individual users who are connected by directed trust and distrust edges. Members of the site can decide whether to "trust" each other or not. All trust relationships interact and form a trusted network, which is then combined with review ratings to determine which reviews are shown to users (Leskovec and Krevl, 2014).

WikiPol The network contains interpretive interactions between English-language Wikipedia users who have edited pages about politics. Each interaction, such as text editing, reverting, restoring, and voting, is assigned a positive or negative value (Maniu et al., 2011). The results can be interpreted as a web of trust. Edges represent interactions and vertices represent users.

Table 1 shows the volume and negative edges proportion of each signed graphs.

Table 1: Datasets Description

Name	$\|V\|$	$\|E\|$	$\|E_-\|/\|E\|$
Highland	16	58	0.5
Sampson	18	189	0.5
ConVote	219	764	0.2
WoW-EP8	790	116009	0.2
Bitcoin	5881	21492	0.2
WikiVote	7115	100693	0.2
Referendum	10884	251406	0.1
Slashdot	82140	500481	0.2
WikiCon	116717	2026646	0.6
Epinions	131580	711210	0.2
WikiPol	138587	715883	0.1

4.2 Evaluation and Comparisons

4.2.1 Polarity

Table 2: Methods

Name	Description
SCG-MA	*Minimum angle*: the deterministic rounding algorithm in Section 3.6
SCG-R	*Randomized rounding*: the randomized rounding algorithm in Section 3.6
SCG-MO	*Maximum objective*: a generalization of EigenSign (Bonchi et al., 2019), that rounds $v_1(A)$ by finding an optimal threshold to maximize the objective.
SCG-B	*Bansal*: motivated by the pivot for correlation clustering.(Bansal et al., 2004)
kOCG-1/ kOCG-r	*Baseline* in Section 3.5: returns of k-OCG containing k conflicting groups, picking the k groups contained in their $top-1$ and $top-r$ subgraphs
BNC-k BNC-$k+1$	Apply balanced normalized cut (Section 3.3) to detect k clusters as conflicting groups. Apply balanced normalized cut (Section 3.3) to detect $k+1$ clusters, return the k smallest clusters as the detected conflicting groups.
SPONGE-k SPONGE-$k+1$	Apply SPONGE (Section 3.4) to detect k clusters as conflicting groups. Apply SPONGE (Section 3.4) to detect $k+1$ clusters, return the k smallest clusters as the detected conflicting groups.

The detailed information of each testing method label are shown in Table 2.　Note the baseline we use is the same work of (Tzeng et al., 2020)'s work, which is the k-OCG method.　The authors implemented k-OCG[1] with the default hyperparameters $\alpha=1/(k-1)$, $\beta=50$, and $l=5000$.　BNC-k and SPONGE-k mean to use these two methods to detect k clusters and the output will be the detected conflicting groups.　BNC-$k+1$ and SPONGE-$k+1$ will detect $k+1$ clusters in the process, and not consider $k+1$ to be the conflicting group, but return the k smallest clusters as the detected conflicting groups.　The implementation of BNC and SPONGE is open source in SigNet[2].　Polarity objective, i. e., Eq.29 of these methods are measured.　When setting the detecting groups as 2 ($k=2$), the results are shown in Table 3.

We can analyze the results of Table 3 in several aspects.　First of all, when the graph volume is small (i.e. the number of vertices $|V|$ is less than 1000), in our datasets which are Highland, Sampson, and ConVote, we observe that the polarity value of other methods except BNC shows similar values compared to the k-OCG method, the polarity value of BNC is even negative in a very small signed graph with 16 vertices and 58 edges.　Since BNC-($k+1$) and SPONGE-($k+1$) can use the spare cluster to put the non-conflicting nodes, we get better performance than BNC-k and SPONGE-k on small-size graphs.　Unfortunately, BNC-($k+1$) and SPONGE-($k+1$) does not hold up well on a large

1) https://github.com/lingyangchu/KOCG.SIGKDD2016
2) https://github.com/alan-turing-institute/signet

Table 3: $k=2$

Polarity	SCG-MA	SCG-MO	SCG-B	SCG-R	KOCG-1	KOCG-r	BNC-k	BNC-k+1	SPONGE-k	SPONGE-k+1
Highland	6.2	6.2	3.5	5.5	3.6	4.2	-0.5	-1.1	4.0	5.1
Sampson	7.5	7.5	4.2	6.5	5.3	5.8	0.9	1.0	6.0	5.8
ConVote	6.5	6.6	3.1	5.3	3.7	2.9	0.1	1.7	0.1	3.6
WoW-EP8	**236.6**	**236.6**	**200.6**	**214.0**	13.0	13.0	**184.6**	-0.7	**191.4**	88.0
Bitcoin	28.8	29.5	21.7	14.4	1.0	3.8	5.3	-10.8	5.1	1.0
WikiVote	71.5	71.7	37.6	53.5	7.6	2.3	15.8	-1.1	15.8	1.0
Referendum	**172.2**	**174.1**	**116.3**	**119.5**	11.6	15.4	41.5	-1.0	38	1.0
Slashdot	77.5	79.7	61	28.6	2	2.6	-	-	-	-
WikiCon	**155.2**	**175.7**	**129.3**	**100.2**	5.9	3.4	-	-	-	-
Epinions	**128.3**	**128.7**	**156.4**	**73.0**	8.2	14.0	-	-	-	-
WikiPol	82.8	88.4	46.5	35.2	3.0	1.3	-	-	-	-

scale of graphs, or even worse. On very large-scale datasets, where the volume is over 500000, the BNC and SPONGE methods exceed the memory limit because of the k-means. When we turn our attention to the large-scale graph, we can find that the maximum objective and minimum angle of the SCG method shows the best performance. It is worth mentioning that although the polarity of SCG-B is very high, the correlation clustering method by Bansal makes SCG-B very time-consuming, as we will explain in detail in the next section.

There are several algorithms for detecting antagonistic communities in signed graphs. We expect to widen our horizon to multiple groups detection, Table 4 shows the polarity when $k=6$, not surprisingly methods of SCG still performed extremely well. But we can find Referendum always shows bigger polarity than Bitcoin and WikiVot, we intuitively believe that the graph with more negative edges is better for clustering. To explain this phenomenon, we should note that although the negative edge ratio of Referendum is only 0.1, while the negative edge ratio of Bitcoin and WikiVot are both 0.2, there are only 1.8 and 1.5 times more nodes than Bitcoin and WikiVot, but 4.7 and 2.5 times more edges than them, which means that referendum is a more dense graph compared to Bitcoin and WikiVot. As the graph volume increases, the performance of the randomized rounding SCG-R method gets worse. BNC and SPONGE just show the effective polarity in k clustering detection in the case of $k=6$.

Table 4: $k=6$

Polarity	SCG-MA	SCG-MO	SCG-B	SCG-R	KOCG-1	KOCG-r	BNC-k	BNC-k+1	SPONGE-k	SPONGE-k+1
Highland	3.4	4.0	3.8	4.0	1.7	1.7	1.2	-0.4	2.2	1.1
Sampson	3.3	4.4	4.3	2.8	2.9	2.9	1.8	1.3	3	2.4
ConVote	2.3	4.6	4.4	3.8	2.5	2.3	0.0	1.8	0.1	0.2
WoW-EP8	**207.3**	**226.9**	**211.6**	**194.8**	7.3	6.2	**185.2**	-0.2	58.5	48.1
Bitcoin	14.6	15.2	9.3	6.6	5.7	3.2	5.2	-4.2	5.1	1.2
WikiVote	45.5	47.0	23.3	5.3	4	3.2	15.8	-1.1	15.8	1.0
Referendum	84.9	55.6	116.2	56.2	8.9	4.1	41.5	-0.8	41.5	1.0
Slashdot	**737.8**	**34.6**	47.7	7.9	3.5	2.6	-	-	-	-
WikiCon	**102.6**	**111.6**	46.1	13.7	4.7	3.6	-	-	-	-
Epinions	**88.8**	**129.2**	94.5	30.6	7.1	5.5	-	-	-	-
WikiPol	**57.5**	**41.8**	46.0	3.6	3.2	3.2	-	-	-	-

In Table 5, we list the polarity values in the case of detecting 8 groups in these real-world networks. All algorithms still perform not well in very small-scale community detection, and as k increases, the graphs with small $|V|$ naturally no longer have a high objective value. As in the case of $k=6$, the SCG series of algorithms still performs better on referendum than on Bitcoin and WikiVot, but the BNC-k and SPONGE-k show better values than SCG-MO and SCG-R. Due to the limitation of the k-OCG method, we cannot give the polarity at $k=8$ and $k=20$. In a very sparse network, which is the WOW-EP8, BNC shows a similar performance to the series of SCG.

Table 5: $k=8$

Polarity	SCG-MA	SCG-MO	SCG-B	SCG-R	BNC-k	BNC-k+1	SPONGE-k	SPONGE-k+1
Highland	3.1	3.7	3.3	3.8	1.3	0.5	1.9	0.6
Sampson	3.1	3.7	3.3	3.8	1.5	1.2	2.3	1.6
ConVote	4.5	4.2	2.4	4.1	0	1.3	0.1	0.1
WoW-EP8	**199.0**	**227.1**	**200.6**	**195.4**	**185.4**	-0.2	**51.9**	**51.3**
Bitcoin	14.0	16.2	21.7	4.3	5.0	-4.1	4.8	0.6
WikiVote	43.4	43.6	37.6	5.0	15.8	-1.0	15.8	1.0
Referendum	69.1	48.4	**116.3**	18.5	41.5	-0.7	41.5	1.0
Slashdot	38.6	33.9	61	15.6	-	-	-	-
WikiCon	**98.1**	**88.8**	**129.3**	13.7	-	-	-	-
Epinions	**88.2**	**129.3**	**156.4**	20.7	-	-	-	-
WikiPol	**57.1**	**41.9**	**46.5**	4.1	-	-	-	-

We try to extend these methods on detecting more groups, and the results are illustrated in Table 6. Because the dataset of highland and Sampson have input samples 16 and 18, but cluster should be greater than 21. Therefore, we investigate the remaining other datasets at $k=20$. The correlation clustering method by Bansal in SCG runs over time, so we do not list the results of this method either. As the number of groups to be examined increases, SCG-MA, SCG-MO, and SCG-R method does not drop much in polarity, and SCG-R performs even better in clustering on Bitcoin and WikiVot than at $k=8$.

From the analysis above, the results of the polarity objective verify that the SCG algorithm is more adept at clustering very sparse and large-scale graphs than other methods. In the next section, we give an execution time comparison to illustrate the effectiveness of each algorithm.

Table 6: $k=20$

Polarity	SCG-MA	SCG-MO	SCG-R	BNC-k	BNC-k+1	SPONGE-k	SPONGE-k+1
ConVote	3.8	3.6	2.5	1.3	1.6	2.2	1.9
WoW-EP8	**177.3**	**181.2**	**108.5**	**186.3**	0.0	52.7	41.6
Bitcoin	8.9	8.1	7.1	5.0	-3.1	3.9	0.5
WikiVote	36.2	29.9	27.6	15.8	-1.1	13.7	0.1
Referendum	52.2	31.1	29.0	41.5	-0.4	39.3	0.4
Slashdot	**34.5**	9.7	-	-	-	-	-
WikiCon	**85.4**	61.9	-	-	-	-	-
Epinions	**62.6**	49.7	-	-	-	-	-
WikiPol	**81.4**	21.3	-	-	-	-	-

4.2.2 Execution Time

Figure 6, 7 ,8, and 9 show the execution time at $k=2$, $k=6$, $k=8$, and $k=20$, respectively. To see the comparison more visually from the figure and due to the long execution time of datasets such as Slashdot, WikiCon, Epinions, and WikiPol, we set up a double y-axis in the figure. Although the SCG-B method has better polarity values compared with other methods, the execution time tends to be the longest. WikiCon has the most complex network structure, in the case of $k=2$, SCG-B almost needs 15 hours to obtain results. When k, the number of clusters increase, the execution time becomes almost 22 hours and 36 hours in the case of $k=6$ and $k=8$, respectively. SCG-MO seems to exhibit a shorter execution time than SCG-MA in the case of more groups to be detected. BNC often shows a faster execution time than SPONGE for detecting multiple groups in a relatively small scale graph as well.

Figure 6: Execution Time ($k=2$)

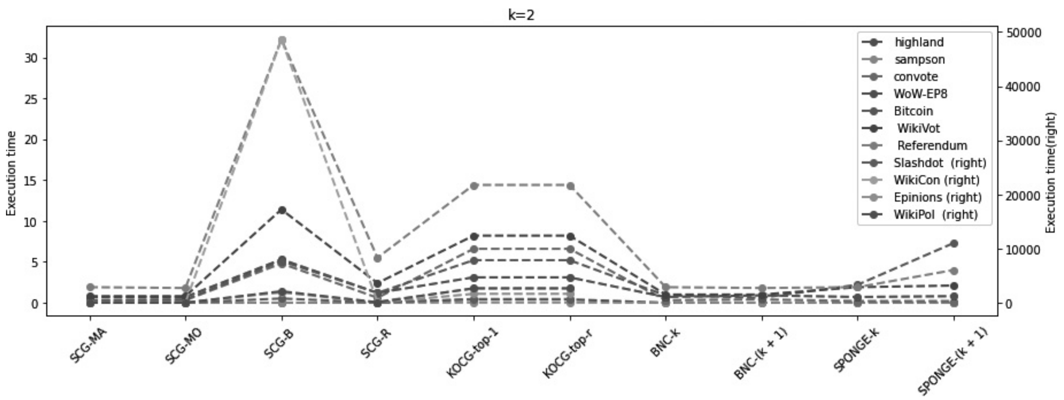


Figure 7: Execution Time ($k=6$)

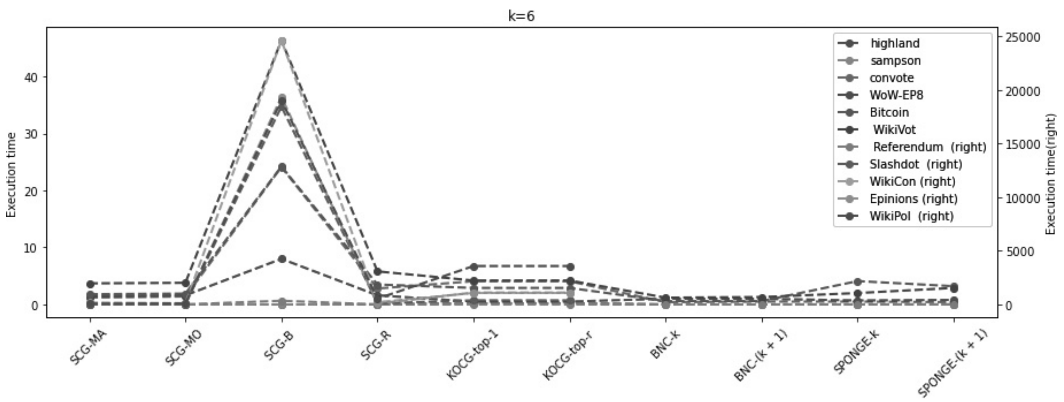


Figure 8: Execution Time（k=8）

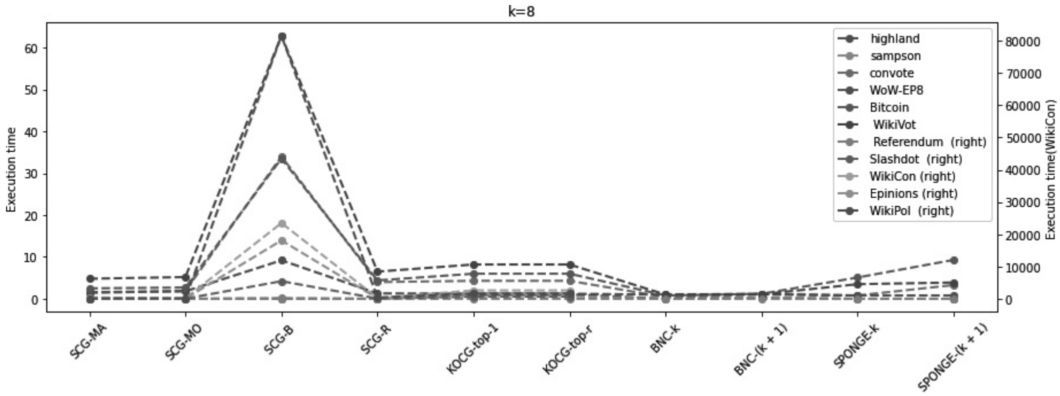
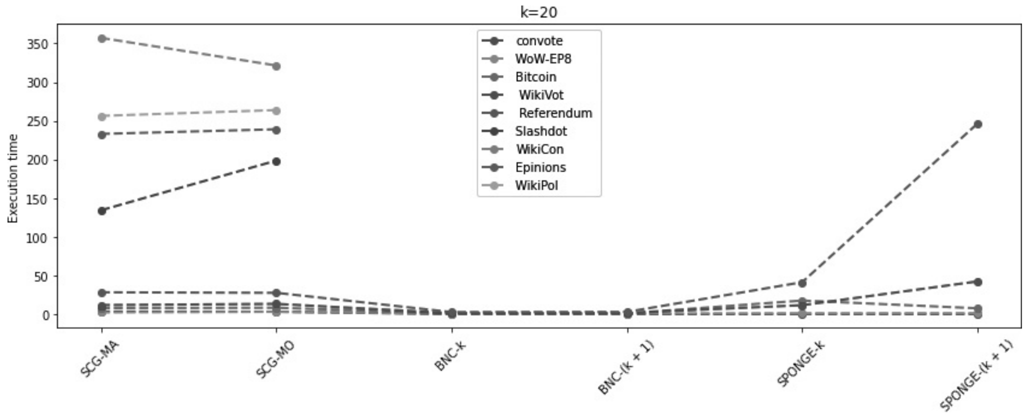


Figure 9: Execution Time（k=20）



5　Conclusion and Future Work

In this paper, we present a comprehensive overview of the existing methods for detecting multiple groups in signed graphs.　we conduct several numerical experiments on real-world networks to compare the performance of each algorithm, and the real-world network study provided us with many new avenues to explore.　From the experiment, we can find that as the number of groups to be detected increases, there is no very significant difference in performance between the methods listed in this paper.　SCG-MA and SCG-MO obtain the best polarity scores in most cases when the target is a large-scale network.　SCG-R also has relatively good polarity values, but it is not as competitive as SCG-MA or SCG-MO with slower execution times.　SCG-B has the best polarity values in some cases, but has much larger execution times than the other methods.　BNC seems to perform better than SPONGE when it comes to relatively complex large networks.　kOCG remains less effective compared to other methods under the same total group size.

It is important to generalize and conduct experiments on such a state-of-the-art research topic. We find that this work is extremely challenging in terms of both algorithmic difficulty and time complexity as the number of detection groups increases, which provides us with great motivation for future work. In this paper we focus extensively on conflict multiply groups in signed networks based on spectral methods, there are still some advanced detection methods that are not based on spectral methods, in the next work we will try to investigate whether we can develop synergy between the two types of methods. We also expect to solve the open question proposed by (Tzeng et al., 2020), which is whether we can use an approach that does not rely on rounding the leading eigenvector to develop the $O(\sqrt{n})$-approximation for the maximum discrete Rayleigh quotient problem. We will continue to contribute to efficient algorithms for handling network clustering of large-scale networks. We open our minds on apply clustering detection methods to real-world interactions, such as social media, and recommendation systems as well.

Acknowledgement

This work was supported by JST SPRING, Grant Number JPMJSP2136.

References

Duanbing Chen, Linyuan Lü, Ming-Sheng Shang, Yi-Cheng Zhang, and Tao Zhou. Identifying influential nodes in complex networks. *Physica a: Statistical mechanics and its applications*, 391(4): 1777-1787, 2012.

Punam Bedi and Chhavi Sharma. Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(3): 115-135, 2016.

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

Lei Tang and Huan Liu. Community detection and mining in social media. *Synthesis lectures on data mining and knowledge discovery*, 2(1): 1-137, 2010.

Smriti Bhagat, Graham Cormode, and S Muthukrishnan. Node classification in social networks. In *Social network data analytics*, pages 115-148. Springer, 2011.

David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 556-559, 2003.

Charu Aggarwal and Karthik Subbian. Evolutionary network analysis: A survey. *ACM Computing Surveys (CSUR)*, 47 (1): 1-36, 2014.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3): 93-93, 2008.

Zexi Huang, Arlei Silva, and Ambuj Singh. Pole: Polarized embedding for signed networks. *arXiv preprint arXiv:2110.09899*, 2021.

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

Jing Wang, Jie Shen, Ping Li, and Huan Xu. Online matrix completion for signed link prediction. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 475-484, 2017.

Yixuan He, Xitong Zhang, Junjie Huang, Mihai Cucuringu, and Gesine Reinert. Pytorch geometric signed directed: A survey and software on graph neural networks for signed and directed graphs. *arXiv preprint arXiv:2202.10793*, 2022.

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

James A Davis. Clustering and structural balance in graphs. *Human relations*, 20(2): 181-187, 1967.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine learning*, 56(1): 89-113, 2004.

Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5): 1-27, 2008.

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.

Yao Ping Hou. Bounds for the least laplacian eigenvalue of a signed graph. *Acta Mathematica Sinica*, 21(4): 955-960, 2005.

Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto W De Luca, and Sahin Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. In *Proceedings of the 2010 SIAM international conference on data mining*, pages 559-570. SIAM, 2010.

Pranay Anchuri and Malik Magdon-Ismail. Communities and balance in signed networks: A spectral approach. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 235-242. IEEE, 2012.

Kai-Yang Chiang, Joyce Jiyoung Whang, and Inderjit S Dhillon. Scalable clustering of signed networks using balance normalized cut. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 615-624, 2012.

Jonathan Q Jiang. Stochastic block model and exploratory analysis in signed networks. *Physical Review E*, 91(6): 062805, 2015.

Mihai Cucuringu, Peter Davies, Aldo Glielmo, and Hemant Tyagi. Sponge: A generalized eigenproblem for clustering signed networks. In *The 22nd International Conference on Artificial Intelligence and Statistics,* pages 1088-1098. PMLR, 2019.

Pedro Mercado, Francesco Tudisco, and Matthias Hein. Spectral clustering of signed graphs via matrix power means. In *International Conference on Machine Learning*, pages 4526-4536. PMLR, 2019.

Jialin Hua, Jian Yu, and Miin-Shen Yang. Fast clustering for signed graphs based on random walk gap. *Social Networks*, 60: 113-128, 2020.

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

Ghazaleh Beigi, Jiliang Tang, and Huan Liu. Signed link analysis in social media networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 10, pages 539-542, 2016.

Yanhua Li, Wei Chen, Yajun Wang, and Zhi-Li Zhang. Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 657-666, 2013.

Hao Ma, Michael R Lyu, and Irwin King. Learning to recommend with trust and distrust relationships. In *Proceedings of the third ACM conference on Recommender systems*, pages 189-196, 2009.

Patricia Victor, Chris Cornelis, Martine De Cock, and Ankur M Teredesai. Trust-and distrust-based recommendations for controversial reviews. *IEEE Intelligent Systems*, 26(1): 48-55, 2011.

Tibor Antal, Paul L Krapivsky, and Sidney Redner. Social balance on networks: The dynamics of friendship and enmity. *Physica D: Nonlinear Phenomena*, 224(1-2): 130-136, 2006.

Seth A Marvel, Steven H Strogatz, and Jon M Kleinberg. Energy landscape of social balance. *Physical review letters*, 103 (19): 198701, 2009.

Thomas F Coleman and Jorge J Moré. Estimation of sparse jacobian matrices and graph coloring blems. *SIAM journal on Numerical Analysis*, 20(1): 187-209, 1983.

Jiliang Tang, Yi Chang, Charu Aggarwal, and Huan Liu. A survey of signed network mining in social media. *ACM Computing Surveys (CSUR)*, 49(3): 1-37, 2016.

Ahmed Sameh and Zhanye Tong. The trace minimization method for the symmetric generalized eigenvalue problem. *Journal of computational and applied mathematics*, 123(1-2): 155-175, 2000.

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

Neal Parikh Stephen Boyd et al. Proximal algorithms. foundations and trends® in optimization 1 3 (2014) 127-239. neal parikh stephen boyd et al. 2014. proximal algorithms. foundations and trends® in optimization 1 3 (2014) 127-239, 2014.

Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3): 334-334, 1997.

Francesco Bonchi, Edoardo Galimberti, Aristides Gionis, Bruno Ordozgoiti, and Giancarlo Ruffo. Discovering polarized communities in signed networks. In *Proceedings of the 28th acm international conference on information and knowledge management*, pages 961-970, 2019.

Ruo-Chun Tzeng, Bruno Ordozgoiti, and Aristides Gionis. Discovering conflicting groups in signed networks. *Advances in Neural Information Processing Systems*, 33: 10974-10985, 2020.

Per Hage and Frank Harary. Structural models in anthropology. 1983. http://konect.cc/networks/ucidata-gama/.

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

Samuel F Sampson. A novitiate in a period of change: An experimental and case study of relationships. *Unpublished Ph. D. dissertation, Department of Sociology, Cornell University*, 1968. http://konect.cc/networks/moreno_sampson/.

Matt Thomas, Bo Pang, and Lillian Lee. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. *arXiv preprint cs/0607062*, 2006. http://konect.cc/networks/convote/.

Victor Kristof, Matthias Grossglauser, and Patrick Thiran. War of words: The competitive dynamics of legislative processes. In *Proceedings of The Web Conference 2020*, pages 2803-2809, 2020.

Jure Leskovec and Andrej Krevl. Snap datasets: Stanford large network dataset collection, 2014. https://snap.stanford.edu/data/index.html.

Mirko Lai, Viviana Patti, Giancarlo Ruffo, and Paolo Rosso. Stance evolution and twitter interactions in an italian political debate. In *International Conference on Applications of Natural Language to Information Systems*, pages 15-27. Springer, 2018.

Jérôme Kunegis. Konect: the koblenz network collection. In *Proceedings of the 22nd international conference on world wide web*, pages 1343-1350, 2013.

Silviu Maniu, Talel Abdessalem, and Bogdan Cautis. Casting a web of trust over wikipedia: an interaction-based approach. In *Proceedings of the 20th international conference companion on World wide web*, pages 87-88, 2011.