Scalable Clustering of Signed Networks Using Balance Normalized Cut

Kai-Yang Chiang
Dept. of Computer Science
University of Texas at Austin
kychiang@cs.utexas.edu

Joyce Jiyoung Whang
Dept. of Computer Science
University of Texas at Austin
joyce@cs.utexas.edu

Inderjit S. Dhillon
Dept. of Computer Science
University of Texas at Austin
inderjit@cs.utexas.edu

ABSTRACT

We consider the general k-way clustering problem in signed social networks where relationships between entities can be either positive or negative. Motivated by social balance theory, the clustering problem in signed networks aims to find mutually antagonistic groups such that entities within the same group are friends with each other. A recent method proposed in [13] extended the spectral clustering algorithm to the signed network setting by considering the signed graph Laplacian. This has been shown to be equivalent to finding clusters that minimize the 2-way signed ratio cut. In this paper, we show that there is a fundamental weakness when we directly extend the signed Laplacian to the k-way clustering problem. To overcome this weakness, we formulate new k-way objectives for signed networks. In particular, we propose a criterion that is analogous to the normalized cut, called balance normalized cut, which is not only theoretically sound but also experimentally effective in k-way clustering. In addition, we prove that these objectives are equivalent to weighted kernel k-means objectives by choosing an appropriate kernel matrix. Employing this equivalence, we develop a multilevel clustering framework for signed networks. In this framework, we coarsen the graph level by level and refine the clustering results at each level via a k-means based algorithm so that the signed clustering objectives are optimized. This approach gives good quality clustering results, and is also highly efficient and scalable. In experiments, we see that our multilevel approach is competitive to other state-of-the-art methods, while it is much faster and more scalable. In particular, the largest graph we have considered in our experiments contains 1 million nodes and 100 million edges — this graph can be clustered in less than four hundred seconds using our algorithm.

Category: I.5.3 - Computing Methodologies - Pattern Recognition - Clustering
General Terms: Algorithms, Experimentation
Keywords: Clustering, Signed Networks, Social Balance Theory, Signed Graph Kernels

1. INTRODUCTION

Social network analysis has gained considerable attention in recent years. These networks can be modeled as graphs with nodes and edges, where nodes indicate individual actors and edges indicate the relationships between actors. In many real social networks, there are not only positive relationships but also negative relationships. For example, online news and review websites such as *Epinions* and *Slashdot* allow users to approve or denounce others. These kinds of networks can be modeled as *signed networks* where a positive relationship is denoted by a positive edge weight while a negative relationship is denoted by a negative edge weight.

It has been shown that signed networks tend to have a particular structure that is derived by the fact that relationships between entities tend to follow so-called "balanced" patterns in signed networks [3, 9]. Motivated by social balance theory, the clustering problem in signed networks is to find antagonistic clusters such that entities within the same cluster have a positive relationship with each other and entities between different clusters have a negative relationship with each other. While much research on graph clustering has been conducted on unsigned graphs which contain only positive relationship information, it is yet to be explored how many existing clustering algorithms can be extended to signed networks. In particular, since the goal of clustering changes when we consider signed networks, existing unsigned clustering methods cannot be directly applied to signed graph clustering.

Due to the massive size of real networks, scalability is an important issue. For unsigned networks, several scalable clustering algorithms have been proposed based on the multilevel framework [7, 11]. However, to the best of our knowledge, there is no scalable multilevel clustering algorithm for signed networks.

In this paper, we formulate new k-way objectives for signed networks based on social balance theory. In particular, we propose a criterion that is analogous to the normalized cut, called balance normalized cut, which is not only theoretically sound but also experimentally effective in k-way clustering. Furthermore, we show that these objectives are equivalent to a general weighted kernel k-means objective. Using this equivalence, we develop a fast multilevel clustering algorithm for signed networks. In this multilevel algorithm, clustering can be optimized level by level very efficiently.

The following are the contributions of our paper:
- We formulate new k-way objectives and kernels for signed networks, and compare them with the signed graph Laplacian considered in [13]. We also show a

fundamental weakness of the signed graph Laplacian in k-way clustering problems.

- We show that our new k-way objectives are mathematically equivalent to a general weighted kernel k-means objective. Using this equivalence, we show that the graph clustering objectives of signed networks can be optimized by using a kernel k-means like algorithm.
- Employing the equivalence between k-way graph clustering and weighted kernel k-means objectives, we develop a fast and scalable multilevel clustering algorithm for signed networks. Our multilevel clustering algorithm is comparable in accuracy to other state-of-the-art methods while it is much faster and more scalable.

This paper is organized as follows. In Section 2, we briefly review some related work, and in Section 3, we state some preliminaries. In Section 4, we show a weakness of the signed graph Laplacian and introduce our new k-way objectives and kernels for signed networks. Also, we prove the equivalence between these objectives and a general weighted kernel k-means objective. In Section 5, we embed our algorithm into a multilevel framework, and explain our multilevel clustering algorithm for signed networks. We show our experimental results in Section 6, and state our conclusions in Section 7.

2. RELATED WORK

The clustering problem in signed networks can be dated back to the 1950s. Cartwright and Harary [3, 9] introduced the notion of social balance. They defined balanced triads, and showed that a network which follows the balance notion can be clustered into two perfect antagonistic groups. This balance notion was generalized by Davis [4], who defined the concept of a weakly balanced network, and showed that a network can be partitioned into k antagonistic groups if it is weakly balanced. In Section 3.3, we formally state balance theory and weak balance theory.

Motivated by (weak) balance, many researchers have tried to develop algorithms for clustering signed networks. For example, Doreian et al. [8] proposed a local search strategy which is similar to the Kernighan-Lin algorithm [12]. Yang at el. [17] proposed an agent-based method which basically conducts a random walk on the graph. Anchuri at el. [1] proposed hierarchical iterative methods that solve 2-way frustration and signed modularity objectives using spectral relaxation at each hierarchy. On the other hand, Nikihil at el. also consider the signed graph clustering problem, though their formulation is motivated by correlation clustering [2]. They proposed two approximation algorithms and proved approximation bounds. Different from these works, our work starts with considering k-way signed cut and association objectives which directly measure the quality of clusters. Thereafter, we obtain k clusters by globally optimizing these signed objectives.

The most relevant work to ours is a spectral method proposed by Kunegis et al. [13], who showed that a signed version of the 2-way ratio-cut problem can be solved by considering the so-called signed graph Laplacian. However, there is a fundamental weakness when we directly extend the signed Laplacian to the k-way clustering problem, which we will discuss in detail in Section 4.1.

Another state-of-the-art framework for signed graph clustering is based on a low rank model proposed by Hsieh et al. [10]. They observe that matrix completion on signed networks can be thought as recovering the missing structure of the network. Thus, in this framework they first complete the network and derive low rank factors (for example, top k eigenvectors) as representations of the signed network. Afterward, they run k-means on the low rank factors to derive the clustering. We will compare this state-of-the-art approach with our proposed multilevel clustering algorithm in Section 6.

Scalability is always an important issue for graph clustering algorithms. For unsigned networks, many scalable graph clustering algorithms make use of a multilevel framework, such as Metis [11] and Graclus [7]. In the multilevel framework, the input graph is repeatedly coarsened level by level until the graph size becomes small enough. An initial clustering is then performed on the coarsest graph. Finally, the clustering result is refined as the graph is uncoarsened level by level. Furthermore, Dhillon et al. [7] showed that a weighted graph clustering objective is mathematically equivalent to a general weighted kernel k-means objective, and utilizing this equivalence into the multilevel framework can make the clustering procedure very efficient. However, the algorithms used in [7] cannot be directly applied to signed networks, and the modifications are not trivial either.

3. PRELIMINARIES

In this paper, we use capital letters to denote matrices, lowercase bold letters to denote vectors, and lowercase italics to denote scalars. We use $x_c(i)$ to denote the ith element of the vector $\mathbf{x}_c$ when the vector has a subscript; otherwise, we use x_i to denote the ith element of the vector $\mathbf{x}$.

In this section, we first review spectral clustering of unsigned networks in Section 3.1. In Section 3.2, we introduce weighted kernel k-means which has been shown to be effective in optimizing graph cut objectives of unsigned networks. In Section 3.3, we state some basic concepts for signed networks. Finally, in Section 3.4, we introduce a state-of-the-art spectral method that solves a 2-way signed network clustering problem.

3.1 Graph Cuts on Unsigned Networks

Let us consider k-way clustering on unsigned networks. One of the most popular approaches to solving this problem is to consider some objective, such as graph association or graph cuts, that quantifies the quality of clusters, and derive a clustering result by maximizing or minimizing the objective function. Here are some well-known objectives:

- **Ratio Association objective** aims to maximize the number of edges within clusters relative to the size of the cluster, where the size of a cluster is defined as the number of nodes it contains. Precisely, the ratio association objective may be written as:

$$\max_{\{\mathbf{x}_1,...,\mathbf{x}_k\} \in I} \left(\sum_{c=1}^{k} \frac{\mathbf{x}_c^T A \mathbf{x}_c}{\mathbf{x}_c^T \mathbf{x}_c} \right),$$

where A is the network's adjacency matrix, and the constraint $\{\mathbf{x}_1,...,\mathbf{x}_k\} \in I$ means $\{\mathbf{x}_1,...,\mathbf{x}_k\}$ is a k-cluster indicator set. Formally, an indicator set $\{\mathbf{x}_1,...,\mathbf{x}_k\}$ is defined as a set of k vectors $\mathbf{x}_1 \ldots \mathbf{x}_k$, 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 cth cluster.

- **Ratio Cut objective** aims to minimize the number of edges between different clusters relative to the size of

the cluster. It is known that the number of edges across clusters can be captured by the graph Laplacian of A. More precisely, let D be the diagonal degree matrix of A, i.e. $D_{ii} = \sum_{j=1}^n A_{ij}$, then the graph Laplacian is defined by $L = D - A$. The ratio cut objective is:

$$\min_{\{\mathbf{x}_1,\dots,\mathbf{x}_k\} \in I} \left(\sum_{c=1}^k \frac{\mathbf{x}_c^T L \mathbf{x}_c}{\mathbf{x}_c^T \mathbf{x}_c} \right). \qquad (1)$$

- **Normalized Association** and **Normalized Cut objectives** also consider within-cluster edges and between-cluster edges, respectively. However, these objectives are normalized by the volume of each cluster rather than the number of nodes in each cluster. The volume of a cluster is defined as the sum of degrees of nodes in the cluster. It has been shown that these two normalized objectives are equivalent, i.e.,

$$\max_{\{\mathbf{x}_1,\dots,\mathbf{x}_k\} \in I} \left(\sum_{c=1}^k \frac{\mathbf{x}_c^T A \mathbf{x}_c}{\mathbf{x}_c^T D \mathbf{x}_c} \right) \equiv \min_{\{\mathbf{x}_1,\dots,\mathbf{x}_k\} \in I} \left(\sum_{c=1}^k \frac{\mathbf{x}_c^T L \mathbf{x}_c}{\mathbf{x}_c^T D \mathbf{x}_c} \right). \qquad (2)$$

In addition, under the special case $k = 2$, the ratio cut objective (1) is equivalent to the following problem:

$$\min_{\mathbf{x}} \left(\mathbf{x}^T L \mathbf{x} \right), \qquad (3)$$

where the 2-class indicator $\mathbf{x}$ has the following form:

$$x_i = \begin{cases} \sqrt{|\pi_2|/|\pi_1|}, & \text{if node } i \in \pi_1, \\ -\sqrt{|\pi_1|/|\pi_2|}, & \text{if node } i \in \pi_2, \end{cases}$$

with $|\pi_1|, |\pi_2| > 0$. See [5] for detailed proof.

Note that all the above formulated problems are NP-hard problems [15]. In practice, these problems are usually solved with spectral relaxation, which is known as spectral clustering [16]. Specifically, if we consider the relaxed problems by dropping the combinatorial constraint $\{\mathbf{x}_1,\dots,\mathbf{x}_k\} \in I$, then the top k eigenvectors will yield the optimal solution. One way to round $\{\mathbf{x}_1,\dots,\mathbf{x}_k\}$ to a valid indicator set, we treat each $[x_1(i), x_2(i)\dots x_k(i)]$ as a k-dimensional vector for $1 \le i \le n$, and do k-means clustering on these n vectors.

Though spectral clustering works quite well in practice, it suffers a scalability issue since computing eigenvectors can be costly. In particular, the normalized objective usually squeezes the spectrum of the matrix to a small range, which makes the computation much more expensive due to the smaller eigengaps.

3.2 Weighted Kernel K-means

Weighted kernel k-means is a generalized version of k-means [7]. Given a set of vectors $\mathbf{v}_1 \dots \mathbf{v}_n$, the weighted kernel k-means objective is defined as follows:

$$\min_{\pi_1 \dots \pi_k} \sum_{c=1}^k \sum_{\mathbf{v}_i \in \pi_c} w_i \|\phi(\mathbf{v}_i) - \mathbf{m}_c\|^2, \qquad (4)$$

where w_i is a nonnegative weight of each vector $\mathbf{v}_i$, ϕ is a non-linear mapping, and $\mathbf{m}_c$ is the weighted centroid of π_c which is defined by:

$$\mathbf{m}_c = \frac{\sum_{\mathbf{v}_i \in \pi_c} w_i \phi(\mathbf{v}_i)}{\sum_{\mathbf{v}_i \in \pi_c} w_i}.$$

Like the traditional k-means algorithm, weighted kernel k-means computes the closest centroid for every node, and

---

Algorithm 1: Weighted Kernel k-means

Input: $\mathbf{v}_1 \dots \mathbf{v}_n$: data vectors, K: kernel matrix, k: number of clusters, $\{\pi_c\}_{c=1}^k$: initial clustering, t: maximum number of iterations.

Output: $\{\pi_c\}_{c=1}^k$: final clustering
- Initialize $\tau = 0$.
- **Do**
    1. For each $\mathbf{v} = \mathbf{v}_1 : \mathbf{v}_n$ and $c = 1 : k$, compute $D(\mathbf{v}_i, \mathbf{m}_c)$ using Equation (5).
    2. Compute C^* for each $\mathbf{v}$ defined by: $C^*(\mathbf{v}_i) := \arg\min_c D(\mathbf{v}_i, \mathbf{m}_c)$.
    3. Reform the clustering defined by: $\pi_c = \{\mathbf{v}_j \mid C^*(\mathbf{v}_j) = c\}$.
    4. $\tau = \tau + 1$.

    While not converged and $\tau < t$

---

assigns the node to the closest cluster. After all the nodes are considered, the centroids are updated. Given the Kernel matrix K, where $K_{ji} = <\phi(\mathbf{v}_j), \phi(\mathbf{v}_i)>$, the distance between $\phi(\mathbf{v}_i)$ and $\mathbf{m}_c$, denoted as $D(\mathbf{v}_i, \mathbf{m}_c)$, can be simplified as follows:

$$D(\mathbf{v}_i, \mathbf{m}_c) = K_{ii} - \frac{2 \sum_{j \in c} w_j K_{ji}}{\sum_{j \in c} w_j} + \frac{\sum_{j \in c} \sum_{l \in c} w_j w_l K_{jl}}{(\sum_{j \in c} w_j)^2}. \qquad (5)$$

This procedure is summarized in Algorithm 1.

It has been shown that all the objectives stated in Section 3.1 are equivalent to the weighted kernel k-means objective by choosing appropriate kernels [7]. Therefore, we can use the weighted kernel k-means algorithm to attempt to optimize these graph clustering objectives without computing eigenvectors.

3.3 Signed Networks and Social Balance

A signed network can be represented as an adjacency matrix A that describes relationships between entities. Formally, the adjacency matrix A is defined as follows:

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

We can break A into its positive part A^+ and negative part A^-. Formally, $A_{ij}^+ = \max(A_{ij}, 0)$ and $A_{ij}^- = -\min(A_{ij}, 0)$. By this definition, we have $A = A^+ - A^-$. For convenience, in the remaining part of this paper, we will use the term "network" as an abbreviation of *signed* network unless we specify it is unsigned. In addition, similar to the convention in spectral clustering, we will focus our discussion on undirected signed networks, i.e., A is symmetric.

The most prevailing theory related to signed networks is *social balance*. The intuition of social balance can be interpreted as "a friend of my friend is my friend", "an enemy of my friend is my enemy", and "an enemy of my enemy is my friend". We say that a graph is *balanced* if any part of the graph does not violate this intuition. It is known that if the network is balanced, then it has a clusterable global structure:

THEOREM 1 (BALANCE THEORY [3, 9]). *A network is balanced iff (i) all of its edges are positive, or (ii) nodes can*

be clustered into two groups such that edges within groups are positive and edges between groups are negative.

In addition, Davis [4] proposed a weaker notion of balance, called weak balance, that generalizes social balance. They relax the balanced relationships by allowing an enemy of one's enemy to still be an enemy. Under such relaxation, they show that a network which satisfies weak balance rules can be clustered as follows:

THEOREM 2 (WEAK BALANCE THEORY [4]). *A network is weakly balanced iff (i) all of its edges are positive, or (ii) nodes can be clustered into k groups such that edges within groups are positive and edges between groups are negative.*

We then say that a network is k-weakly balanced iff it can be divided into k antagonistic groups.

Motivated by Theorem 2, we now formally state the k-way clustering problem in signed networks as follows. Given a signed network, we are asked to partition the network into k clusters such that most edges within clusters are positive and most edges between clusters are negative. In addition, to avoid partitioning where most clusters contain only a few nodes, we prefer that clusters are of substantial size/volume. A similar desire occurs in unsigned network clustering when we consider normalized cut as the criterion [15].

3.4 Clustering via Signed Laplacian

Recently, Kunegis et al. [13] proposed a spectral method for clustering of signed graphs by defining a signed graph Laplacian. Let $\bar{D}$ be the diagonal absolute degree matrix, i.e., $\bar{D}_{ii} = \sum_{j=1}^{n} |A_{ij}|$, then the signed Laplacian $\bar{L}$ is defined in [13] to be $\bar{D} - A$. The signed Laplacian $\bar{L}$ is always positive semidefinite by the fact that $\forall \mathbf{x} \in \mathbb{R}^n$,

$$\mathbf{x}^T \bar{L} \mathbf{x} = \sum_{(i,j)} |A_{ij}|(x_i - \mathrm{sgn}(A_{ij})x_j)^2 \geq 0. \qquad (6)$$

Now let us define k-way ratio cut for signed networks. Given a signed network G, we define k-way ratio cut to be equal to the sum of positive edge weights of edges that lie between different clusters and the sum of negative edge weights of all edges lie within the same cluster, normalized by each cluster's size as in (1) (also see (13) in Section 4.1).

In [13], it has been shown that the 2-way signed ratio cut objective can be formulated as an optimization problem with a quadratic form:

$$\min_{\mathbf{x}} \left(\mathbf{x}^T \bar{L} \mathbf{x} \right), \qquad (7)$$

where the 2-class indicator $\mathbf{x}$ has the following form:

$$x_i = \begin{cases} \frac{1}{2}(\sqrt{|\pi_2|/|\pi_1|} + \sqrt{|\pi_1|/|\pi_2|}), & \text{if node } i \in \pi_1, \\ -\frac{1}{2}(\sqrt{|\pi_2|/|\pi_1|} + \sqrt{|\pi_1|/|\pi_2|}), & \text{if node } i \in \pi_2, \end{cases}$$

with $|\pi_1|, |\pi_2| > 0$. This objective has a similar form to the original unsigned ratio cut objective (3). However, by examining (7), we can verify that only negative edges within the same cluster ($A_{ij} < 0$ and $x_i = x_j$) and positive edges between clusters ($A_{ij} > 0$ and $x_i \neq x_j$) will contribute to the objective function. See [13] for more details.

4. SIGNED GRAPH CUTS AND KERNELS

In this section, we propose new criteria and objectives, including graph association and cut, for general k-way signed graph clustering. We can derive the signed graph kernel corresponding to each objective. In addition, we show that we can use the weighted kernel k-means algorithm to optimize these objectives, by selecting the proper kernel matrix.

4.1 k-way Signed Objectives for Clustering

Before we introduce our proposed objectives, we show why the signed Laplacian is hard to extend to the k-way clustering problem. While the signed Laplacian can be properly used in 2-way clustering, it is not clear how we can extend this definition to general k-way clustering. One intuitive solution is to directly extend (7) to the following k-way objective:

$$\min_{\{\mathbf{x}_1,...,\mathbf{x}_k\} \in I} \left(\sum_{c=1}^{k} \frac{\mathbf{x}_c^T \bar{L} \mathbf{x}_c}{\mathbf{x}_c^T \mathbf{x}_c} \right). \qquad (8)$$

However, unlike in unsigned networks, this direct extension suffers a weakness. To explain this weakness more clearly, let us first consider the unsigned ratio cut. We can observe that we use different representation for indicators ($\mathbf{x}$ in (3) and $\{\mathbf{x}_1, ..., \mathbf{x}_k\}$ in (1)) for 2-way objective and k-way objective. This generalization is valid because based on a 2-way objective (3) with well-defined indicator $\mathbf{x}$, we can properly generalize it to k-way objective (1) by selecting another appropriate representation for indicator set $\{\mathbf{x}_1, ..., \mathbf{x}_k\}$, such that the cut criterion remains the same. Similarly, now given a 2-way signed ratio cut objective (7), we aim to find an appropriate setting of the indicator set $\{\mathbf{x}_1, ..., \mathbf{x}_k\}$, such that the k-way objective (8) with such setting also minimizes the signed ratio cut. Nevertheless, the following theorem proves that this generalization is hopeless no matter how we choose as our indicators $\{\mathbf{x}_1, ..., \mathbf{x}_k\}$:

THEOREM 3. *There does not exist any representation of $\{\mathbf{x}_1, ..., \mathbf{x}_k\}$ such that the objective (8) minimizes the general k-way signed ratio cut (defined in Section 3.4).*

PROOF. Recall that to define a proper representation of $\{\mathbf{x}_1, ..., \mathbf{x}_k\}$, we need to pick two representatives $a, b \in \mathbb{R}$, $a \neq b$, $x_c(i) = a$ if $i \in \pi_c$ and $x_c(i) = b$ if $i \notin \pi_c$, such that the objective (8) minimizes the signed ratio cut.

To begin with, we rewrite the k-way objective (8) by plugging (6) into it as follows:

$$\min_{\{\mathbf{x}_1,...,\mathbf{x}_k\} \in I} \left(\sum_{c=1}^{k} \frac{1}{|\pi_c|} \sum_{(i,j)} |A_{ij}|(x_c(i) - \mathrm{sgn}(A_{ij})x_c(j))^2 \right). \qquad (9)$$

We can prove the theorem by showing that no matter how we choose a, b, we will incorrectly punish some favorable clustering patterns when minimizing (9).

First we argue that b must be 0. This is because when two nodes, i and j, are both not in π_c, we will never know whether i and j are in the same cluster or not by only seeing information about $\mathbf{x}_c$. Therefore, in this case, we have to assign $x_c(i) = x_c(j) = 0$ to ensure a zero penalty, or otherwise we may possibly penalize some patterns that conform to weak balance.

However, given $b = 0$, any $a \neq 0$ cannot make the objective (9) correctly minimize signed ratio cut. To see this, consider that $i \in \pi_c$ and $j \notin \pi_c$, in which case we should put penalty only when $\mathrm{sgn}(A_{ij}) = 1$. However, any fixed choice of a will lead to a penalty regardless of $\mathrm{sgn}(A_{ij})$. Thus, no matter how we select a, b, we will always punish some desirable clustering patterns. This proves the impossibility of a direct generalization of signed ratio cut objective by using (8). $\square$

Notice that the above proof, however, does not apply if we restrict $k = 2$. If there are only two clusters, $i, j \notin \pi_c$ implies that i, j are in the other cluster simultaneously. In this case, we can directly punish this clustering assignment if $\mathrm{sgn}(A_{ij}) = -1$. Therefore, under the special case $k = 2$, the first argument that b needs to be 0 in the proof is no longer valid. However, for general $k > 2$, b needs to be 0 as shown in the proof. This condition precludes the direct extension of the signed Laplacian to the k-way clustering problem.

To fix this basic theoretical flaw, we now propose alternate objectives that (i) follow the weak balance clusterability stated in Theorem 2, and (ii) are valid for general k-way clustering.

- **Positive/Negative Ratio Association**:
  In positive ratio association, we aim to maximize the number of positive edges within each cluster relative to cluster's size. The objective can be written as:

$$\max_{\{\mathbf{x}_1,\ldots,\mathbf{x}_k\}\in I} \left(\sum_{c=1}^{k} \frac{\mathbf{x}_c^T A^+ \mathbf{x}_c}{\mathbf{x}_c^T \mathbf{x}_c} \right) \qquad (10)$$

  Alternatively, we can also minimize the number of negative edges within each cluster relative to the cluster's size. This negative ratio association criterion corresponds to:

$$\min_{\{\mathbf{x}_1,\ldots,\mathbf{x}_k\}\in I} \left(\sum_{c=1}^{k} \frac{\mathbf{x}_c^T A^- \mathbf{x}_c}{\mathbf{x}_c^T \mathbf{x}_c} \right) \qquad (11)$$

- **Positive/Negative Ratio Cut**:
  In addition to within-cluster criteria, between-cluster criteria can also be considered. We define a positive ratio cut objective by minimizing the number of positive edges between clusters:

$$\min_{\{\mathbf{x}_1,\ldots,\mathbf{x}_k\}\in I} \left\{ \sum_{c=1}^{k} \frac{\mathbf{x}_c^T (D^+ - A^+)\mathbf{x}_c}{\mathbf{x}_c^T \mathbf{x}_c} = \sum_{c=1}^{k} \frac{\mathbf{x}_c^T L^+ \mathbf{x}_c}{\mathbf{x}_c^T \mathbf{x}_c} \right\} \qquad (12)$$

  where D^+ is the diagonal degree matrix of A^+. The negative ratio cut can also be defined similarly.

- **Balance Ratio Cut/Association**:
  We now combine both cut and association criteria together to obtain a balance criterion. According to Theorem 2, we prefer to partition nodes such that the number of positive edges between clusters and the number of negative edges within each cluster are both minimized. Thus, we define the balance ratio cut as a combination of positive ratio cut (12) and negative ratio association (11) as follows:

$$\min_{\{\mathbf{x}_1,\ldots,\mathbf{x}_k\}\in I} \left(\sum_{c=1}^{k} \frac{\mathbf{x}_c^T A^- \mathbf{x}_c}{\mathbf{x}_c^T \mathbf{x}_c} + \lambda \frac{\mathbf{x}_c^T L^+ \mathbf{x}_c}{\mathbf{x}_c^T \mathbf{x}_c} \right),$$

  where λ is a weighting parameter. In the following we set $\lambda = 1$ for the case where within-cluster criterion is as important as between-cluster criterion. The balance ratio cut can thus be further simplified as follows:

$$\min_{\{\mathbf{x}_1,\ldots,\mathbf{x}_k\}\in I} \left(\sum_{c=1}^{k} \frac{\mathbf{x}_c^T (D^+ - A)\mathbf{x}_c}{\mathbf{x}_c^T \mathbf{x}_c} \right). \qquad (13)$$

While this objective is almost the same as (8) except that the matrix in the quadratic form $D^+ - A$ changes to $\bar{D} - A$, this definition resolves the weakness of the signed Laplacian defined in [13]. In Section 6, we will show that clusters derived from solving (13) will be better than those derived from solving (8), which reconfirms our analysis.

Of course, we can derive another balance criterion by considering the maximization problem. Following the same flow, we can define balance ratio association as combination of negative ratio cut and positive ratio association with equal weight. This objective can be derived as follows:

$$\max_{\{\mathbf{x}_1,\ldots,\mathbf{x}_k\}\in I} \left(\sum_{c=1}^{k} \frac{\mathbf{x}_c^T (D^- + A)\mathbf{x}_c}{\mathbf{x}_c^T \mathbf{x}_c} \right). \qquad (14)$$

- **Balance Normalized Cut**:
  Similar to clustering tasks in unsigned networks, we can also consider the objectives normalized by cluster volume instead of by the number of nodes in the clusters. For example, we can extend (13) to balance normalized cut as follows:

$$\min_{\{\mathbf{x}_1,\ldots,\mathbf{x}_k\}\in I} \left(\sum_{c=1}^{k} \frac{\mathbf{x}_c^T (D^+ - A)\mathbf{x}_c}{\mathbf{x}_c^T \bar{D}\mathbf{x}_c} \right), \qquad (15)$$

Balance normalized association can also be derived from (14) in the same way.

As an analogy to normalized cut and association in unsigned networks, these two objectives hold a similar equivalence in (2). This nice property is summarized by the following theorem:

THEOREM 4. *Minimizing balance normalized cut is equivalent to maximizing balance normalized association.*

PROOF. Consider the objective (15), we have:

$$(15) \equiv \max_{\{\mathbf{x}_1,\ldots,\mathbf{x}_k\}\in I} \left(k - \sum_{c=1}^{k} \frac{\mathbf{x}_c^T (D^+ - A)\mathbf{x}_c}{\mathbf{x}_c^T \bar{D}\mathbf{x}_c} \right)$$

$$\equiv \max_{\{\mathbf{x}_1,\ldots,\mathbf{x}_k\}\in I} \left(\sum_{c=1}^{k} \frac{\mathbf{x}_c^T \left(\bar{D} - (D^+ - A) \right)\mathbf{x}_c}{\mathbf{x}_c^T \bar{D}\mathbf{x}_c} \right)$$

$$\equiv \max_{\{\mathbf{x}_1,\ldots,\mathbf{x}_k\}\in I} \left(\sum_{c=1}^{k} \frac{\mathbf{x}_c^T (D^- + A)\mathbf{x}_c}{\mathbf{x}_c^T \bar{D}\mathbf{x}_c} \right),$$

by the fact that $\bar{D} = D^+ + D^-$. Thus, (15) is equivalent to maximizing balance normalized association. $\square$

4.2 Solving Signed Objectives via Weighted Kernel K-means

One way to solve the signed cut/association objectives is to use spectral relaxation as traditional spectral clustering does. However, as described in Section 3.1, this approach fails to scale to very large networks due to computational issues. Thus, we now argue that similar to unsigned network clustering, we can use k-means like algorithm to optimize the objectives, by showing the following theorem:

THEOREM 5 (EQUIVALENCE OF OBJECTIVES). *For any signed cut or association objective, there exists some corresponding weighted kernel k-means objective (with properly chosen kernel matrix), such that these two objectives are mathematically equivalent.*

PROOF. We prove this by showing that both weighted kernel k-means objective and signed cut/association objectives can be represented as trace maximization problems. We will also show that given a signed objective, we can construct the kernel matrix such that the corresponding weighted kernel k-means objective is equivalent to the signed objective.

First we consider the weighted kernel k-means objective (4). As proved in [7], this objective can be rewritten as the following trace maximization problem:

$$\max_{Y} \operatorname{trace}\left(Y^T W^{1/2} K W^{1/2} Y \right), \quad (16)$$

where Y is an (weighted) indicator matrix, W is the diagonal weight matrix for the nodes, and K is the kernel matrix.

Now we show how to derive the corresponding trace maximization given the graph cut/association objectives in Section 4.1. Generally, every objective can be written in the following form:

$$\sum_{c=1}^{k} \frac{\mathbf{x}_c^T \acute{K} \mathbf{x}_c}{\mathbf{x}_c^T W \mathbf{x}_c}, \quad (17)$$

where W is some diagonal weight matrix, and the goal is to either maximize or minimize (17).

First, we represent the problem as a maximization problem. We use the following equivalence when the original problem is a minimization problem:

$$\min_{\{\mathbf{x}_1,...,\mathbf{x}_k\} \in I} \sum_{c=1}^{k} \frac{\mathbf{x}_c^T \acute{K} \mathbf{x}_c}{\mathbf{x}_c^T W \mathbf{x}_c} \equiv \max_{\{\mathbf{x}_1,...,\mathbf{x}_k\} \in I} \left(k - \sum_{c=1}^{k} \frac{\mathbf{x}_c^T \acute{K} \mathbf{x}_c}{\mathbf{x}_c^T W \mathbf{x}_c} \right)$$

$$\equiv \max_{\{\mathbf{x}_1,...,\mathbf{x}_k\} \in I} \sum_{c=1}^{k} \frac{\mathbf{x}_c^T (W - \acute{K}) \mathbf{x}_c}{\mathbf{x}_c^T W \mathbf{x}_c}$$

$$\equiv \max_{\{\mathbf{x}_1,...,\mathbf{x}_k\} \in I} \sum_{c=1}^{k} \frac{\mathbf{x}_c^T \tilde{K} \mathbf{x}_c}{\mathbf{x}_c^T W \mathbf{x}_c}, \quad (18)$$

Next, we enforce the positive semi-definiteness on $\tilde{K}$ in (18) by the fact that:

$$\max_{\{\mathbf{x}_1,...,\mathbf{x}_k\} \in I} \sum_{c=1}^{k} \frac{\mathbf{x}_c^T \tilde{K} \mathbf{x}_c}{\mathbf{x}_c^T W \mathbf{x}_c} \equiv \max_{\{\mathbf{x}_1,...,\mathbf{x}_k\} \in I} \left(\sigma k + \sum_{c=1}^{k} \frac{\mathbf{x}_c^T \tilde{K} \mathbf{x}_c}{\mathbf{x}_c^T W \mathbf{x}_c} \right)$$

$$\equiv \max_{\{\mathbf{x}_1,...,\mathbf{x}_k\} \in I} \sum_{c=1}^{k} \frac{\mathbf{x}_c^T (\sigma W + \tilde{K}) \mathbf{x}_c}{\mathbf{x}_c^T W \mathbf{x}_c}. \quad (19)$$

Therefore, every problem in form (18) can be rewritten as:

$$\max_{\{\mathbf{x}_1,...,\mathbf{x}_k\} \in I} \left(\sum_{c=1}^{k} \frac{\mathbf{x}_c^T \hat{K} \mathbf{x}_c}{\mathbf{x}_c^T W \mathbf{x}_c} \right), \quad (20)$$

where $\hat{K}$ is a kernel matrix. This is because we can always choose some sufficiently large σ in (19) such that $\hat{K} = \sigma W + \tilde{K}$ is positive semi-definite.

Finally, (20) is in fact equivalent to a trace maximization problem since:

$$(20) \equiv \max_{\mathbf{y}_1 \cdots \mathbf{y}_k} \sum_{c=1}^{k} \frac{\mathbf{y}_c^T (W^{-1/2} \hat{K} W^{-1/2}) \mathbf{y}_c}{\mathbf{y}_c^T \mathbf{y}_c}$$

$$\equiv \max_{Y} \operatorname{trace}\left(Y^T W^{1/2} (W^{-1} \hat{K} W^{-1}) W^{1/2} Y \right), \quad (21)$$

where $Y \in \mathbb{R}^{n \times k}$, whose c^{th} column is also the normalized indicator $\tilde{\mathbf{y}}_c$ for cluster π_c, i.e. $\tilde{\mathbf{y}}_c = \mathbf{y}_c / \sqrt{(\mathbf{y}_c^T \mathbf{y}_c)}$.

Hence, by choosing the kernel matrix to be $K = W^{-1} \hat{K} W^{-1}$, (21) is equivalent to (16). This proves the theorem. $\square$

This equivalence implies that we can define some implicit kernel matrix $\hat{K}$ given a signed cut/association objective by choosing a proper σ. In addition, running weighted kernel k-means with $W^{-1} \hat{K} W^{-1}$ as the input kernel matrix is equivalent to optimizing the corresponding signed cut/association objective, since both of them optimize the same trace maximization problem.

To explain how to derive the corresponding signed graph kernel more clearly, we take balance normalized cut (15) as an example. With the notations in the proof, the objective is to minimize (17) with $\acute{K} = D^+ - A$ and $W = \bar{D}$. By (18), we can rewrite it as a maximization problem by setting $\tilde{K} = W - \acute{K} = \bar{D} - (D^+ - A)$. The problem is further equivalent to (20) by setting $\hat{K} = \sigma W + \tilde{K} = \sigma' \bar{D} - (D^+ - A)$. This becomes our balance normalized cut kernel, since we can always choose a large enough σ' such that $\hat{K}$ is positive semi-definite. Also, by (21), if we set another kernel matrix K as $W^{-1} \hat{K} W^{-1} = \sigma' \bar{D}^{-1} - \bar{D}^{-1} (D^+ - A) \bar{D}^{-1}$, and use K as the input of weighted kernel k-means, we can optimize balance normalized cut efficiently.

5. MULTILEVEL APPROACH FOR LARGE-SCALE SIGNED GRAPH CLUSTERING

In Section 4.2, we have seen that there are two general approaches to optimize signed graph clustering objectives - spectral clustering and weighted kernel k-means algorithm. Typically, k-means based algorithm is more efficient and scalable than spectral clustering. However, k-means algorithm is also easier to be trapped into qualitatively poor local optima. In this section, we will use a popular multilevel framework which allows kernel k-means based algorithms to converge to better local optima.

Our approach can be viewed as a generalization of Graclus [7] for the signed network setting. The multilevel approach is a divide-and-conquer method that includes three phases: coarsening, base clustering and refinement.

5.1 Coarsening Phase

In the coarsening phase, given the input graph G_0, we generate a series of graphs $G_1 \ldots G_\ell$, such that $|V_{i+1}| < |V_i|$ for all $0 \leq i < \ell$. Given a graph G_i, the coarsened graph G_{i+1} is generated as follows: At the beginning every node is unmarked. We then visit each node in a random order, and try to merge two nodes in G_i into one supernode in G_{i+1} with the following strategy: If the node, say x, is already marked then we skip it. Otherwise, we consider all of x's unmarked neighbors y, and select the one such that the edge weight A_{xy}^i is the largest (where A^i is the adjacency matrix of G_i). We then merge x and y as a supernode z, and all the neighbors of x and y are added to neighbors of z. We mark both x and y. If all neighbors of x are marked, then we simply mark x and do not merge it into any node. After all nodes are visited, the supernodes and non-merged nodes become the vertex set of G_{i+1}.

The matching strategy is quite intuitive since the larger the edge weight between x and y, the more likely that x and y are in the same cluster. There can be some variants of the matching strategy such as visiting nodes ordered by their positive degree and so on.

5.2 Base Clustering Phase

When the original graph is coarsened to a small enough graph, we can directly perform clustering on the coarsest graph G_ℓ. We take two kinds of approaches to do base clus-

tering on G_ℓ: Minimize balance normalized cut of A^ℓ with spectral relaxation, or perform unsigned graph clustering on $A^{\ell+}$ using region-growing algorithm as in Metis [11].

Here are some pros and cons between these two approaches. Considering balance normalized cut using spectral relaxation usually gives a better initialization, but it could be slow if G_ℓ is still very large. The latter might occur if the original network is power-law so that the size reduction at each level might not be significant. On the other hand, the region-growing algorithm used in Metis is very efficient since it requires no eigenvector computation. However, since this method does not consider signed edges in G_ℓ, it is possible to derive a base clustering result such that both positive and negative edges are dense within clusters. This problem is not critical, however, since we will refine the clustering results with consideration of signed edges in the refinement phase. Therefore, for the base clustering method, we use Metis on $A^{\ell+}$ in our experiments.

5.3 Refinement Phase

After we derive the base clustering result in G_ℓ, we run the refinement algorithm to derive clustering results in $G_{\ell-1}$, $G_{\ell-2}$, ..., G_0. Thus, given a clustering result in G_i, the goal is to get a clustering result in G_{i-1}. To do this, we first project the clustering result in G_i to G_{i-1} as the initial clusters. In other words, for $x \in \pi_c$ in G_i, all the nodes in G_{i-1} which were merged to x at the coarsening phase are assigned to π_c. After having the initial clusters, we refine the clustering result by running weighted kernel k-means. As shown in Theorem 5, we can choose a suitable kernel of A^i such that optimizing the weighted kernel k-means objective is equivalent to optimizing the appropriate signed graph criterion. In our implementation, we choose balance normalized cut kernel as our kernel matrix. Notice that since at each level i we have a good initialization by projecting the clustering from level $i-1$, weighted kernel k-means usually converges quickly.

The quality of clusters is dominated by the local optimal of the weighted kernel k-means objective. To make the clustering result better, we can use a local search strategy, which allows k-means to converge to a better local optimal. We incorporate this local search strategy with the standard batch k-means algorithm. Basically, we alternatively run the batch k-means and incremental k-means at each refinement level. More details of this ping-pong procedure can be found in [6].

5.3.1 Choosing the Kernel Matrix

In Theorem 5 we have seen that we can choose a sufficiently large σ to construct a kernel matrix K. However, in practice, if σ is too large, it makes the k-means algorithm harder to escape a poor local optimal. As a result, the clustering result could be bad. On the other hand, a small σ which does not make K positive semi-definite can still be used for k-means algorithm since K might still possibly provide a pretty accurate distance measurement. Observing this fact, we develop a procedure for finding an appropriate σ for K.

We begin with some small σ and derive a corresponding K as the input of our ping-pong procedure. This K could be effective for distance measurement in k-means even if it is not positive semi-definite. However, it is also possible that k-means algorithm cannot converge with such K due to the lack of positive definiteness. If we observe lack of

convergence, we change K by increasing σ and repeat the ping-pong procedure.

On the other hand, if no move occurs in batch k-means, it might be a signal that the σ is too large to make k-means jump out from the local optimal. So, in this case, we change K by decreasing σ and repeat the ping-pong procedure. We do not change σ in other cases.

5.3.2 Computational Issues

Now let us discuss the computational complexity of our k-means based refinement procedure. In Algorithm 1, the bottleneck of computation is to compute the weighted distance $D(\mathbf{v}_i, \mathbf{m}_c)$ for every pair (i, c). If we observe the weighted distance formula in Equation (5), the first term K_{ii} is a constant for fixed i, so we only need to compute last two terms when reassigning nodes to clusters. Now consider the balance normalized cut kernel $K = \sigma \bar{D}^{-1} - \bar{D}^{-1} D^+ \bar{D}^{-1} + \bar{D}^{-1} A \bar{D}^{-1}$. We notice that the first two terms in K are diagonal matrices, which only contribute values for all K_{ii}, so that $K_{ij} = (\bar{D}^{-1} A \bar{D}^{-1})_{ij}$ for all $i \neq j$. By substituting these to (5) and simplifying all expressions, we have:

$$
D(\mathbf{v}_i, \mathbf{m}_c) \propto
\begin{cases}
-\dfrac{2\sum_{j \in c} A_{ij}}{w_i \sum_{j \in c} w_j} + \alpha_c, & \text{if } i \notin \pi_c, \\[2ex]
-\dfrac{2(\sum_{j \in c} A_{ij} + \sigma w_i - D^+_{ii})}{w_i \sum_{j \in c} w_j} + \alpha_c, & \text{if } i \in \pi_c,
\end{cases}
\tag{22}
$$

where α_c is a constant for fixed c:

$$
\alpha_c = \frac{\sum_{j,l \in c} A_{jl} + \sum_{j \in c}(\sigma w_j - D^+_{jj})}{(\sum_{j \in c} w_j)^2}.
$$

Therefore, when running weighted kernel k-means in Algorithm 1, we can first compute and store each α_c in $O(m)$ time, where m is the number of edges in the network. After that, for each node we compute $D(\mathbf{v}_i, \mathbf{m}_c)$ for each c using (22), which takes $O(m + nk)$ time. Finally it takes additional $O(n)$ time to compute $C^*(\mathbf{v}_i)$. If the number of iterations is t, the time complexity of Algorithm 1 is only $O(t(m + nk))$. The same time complexity can be derived similarly for incremental k-means.

We can furthermore make refinement stage more efficient by only considering refinement on boundary nodes. Specifically, at each level, we only refine nodes that have neighbors in different clusters in initial clustering. Considering only boundary nodes usually leads to a similar clustering result to one derived by considering all nodes, but this makes the whole procedure much faster.

6. EXPERIMENTAL RESULTS

In the first part of this section, we demonstrate the effectiveness of the graph objectives and kernels proposed in Section 4.1. In the second part, we show that the multilevel approach is faster than other state-of-the-art methods, and it yields satisfactory clustering results. In addition, we show that our approach is scalable to networks with millions of nodes and edges.

6.1 Graph Kernels

We now compare the graph objectives and kernels proposed in Section 4.1. In particular, we show that the proposed balance ratio cut and balance normalized cut kernels are effective in solving the k-way clustering problem.

Criteria and Kernels. In Section 4, we showed that each proposed objective corresponds to a kernel by repre-

Criterion	Kernel
Signed Laplacian	$\sigma I - \bar{L}$
Normalized Signed Laplacian	$\sigma \bar{D}^{-1} + \bar{D}^{-1} A \bar{D}^{-1}$
Positive Ratio Association	$\sigma I + A^+$
Positive Ratio Cut	$\sigma I - L^+$
Ratio Association	$\sigma I + A$
Balance Ratio Cut	$\sigma I - (D^+ - A)$
Balance Normalized Cut	$\sigma \bar{D}^{-1} - \bar{D}^{-1}(D^+ - A)\bar{D}^{-1}$

Table 1: Criteria and kernels considered in experiments.

senting the objective as in (19). Now we select the following kernels for comparison: Positive Ratio Association (**PosRatioAssoc**), Ratio Association (**RatioAssoc**), Positive Ratio Cut (**PosRatioCut**), Balance Ratio Cut (**BalRatioCut**) and Balance Normalized Cut (**BalNorCut**) as representatives of our proposed methods, and Signed Laplacian (**SignLap**), Normalized Signed Laplacian (**NorSignLap**) as the current state-of-the-art spectral methods [13]. We summarize these kernels in Table 1. For each kernel, we properly choose the smallest σ such that K is positive semi-definite, by computing the spectrum of the matrix.

Experimental Setup and Metrics. To compare the effectiveness of each kernel, we create some synthetic networks with the following procedure. We begin with a complete 5-weakly balanced network A_{com}, in which group sizes are $100, 200, \ldots 500$ respectively. We then uniformly sample some entries from A_{com} to form a weakly balanced network A, with two parameters: sparsity s and noise level ϵ. The sparsity s represents the percentage of edges sampled from A_{com}, and the noise level ϵ specifies the probability of flipping the sign of an edge when sampling. Afterwards, we consider each kernel and its corresponding objective in (19), and derive the clustering using spectral clustering. When we perform spectral clustering, we run k-means ten times with different initializations and select the one that gives us the smallest objective value since a bad initialization of k-means might influence the clustering result.

To evaluate a clustering result (or an indicator set $\{\mathbf{x}_1, ..., \mathbf{x}_k\}$ in other words), we first consider the following objective value:

$$\sum_{c=1}^{k} \frac{\mathbf{x}_c^T A^- \mathbf{x}_c + \mathbf{x}_c^T L^+ \mathbf{x}_c}{\mathbf{x}_c^T W \mathbf{x}_c}. \quad (23)$$

We call (23) the ratio objective if $W = I$ and normalized objective if $W = \bar{D}$. Both objectives measure the degree of imbalance of clusters, with size or volume normalization. Typically, if the underlying kernel is ratio normalized, we use ratio objective to measure the clustering result; otherwise, we use normalized objective to measure the clustering result. Note that the normalized objective is upper bounded by k. In addition, since we also have the "real" clustering as the ground truth in the synthetic datasets, we can calculate the error rate of clustering based on the ground truth. Specifically, we calculate the percentage of misclassified edges if we apply the clustering to A_{com}. We consider an edge to be misclassified if the edge is a between-cluster edge and it is positive, or if the edge is a within-cluster edge and it is negative. The precise definition of error rate is:

$$\sum_{c=1}^{k} \frac{\mathbf{x}_c^T A_{com}^- \mathbf{x}_c + \mathbf{x}_c^T L_{com}^+ \mathbf{x}_c}{n^2},$$

where n is the number of nodes in A_{com}. To make the comparison fair, all results are averaged over ten times trials.

Results. In the first experiment, we sample from A_{com} with different sparsity levels and a fixed $\epsilon = 0$, i.e., no noise. Since in most cases we expect a network to be sparse, we zoom in the range of sparsity $s \in [4 \times 10^{-3}, 10^{-1}]$. The results are shown in Figure 1. A lower error rate and objective implies a more effective method. We can see that PosRatioAssoc, NegRatioAssoc and PosRatioCut, which only consider one of positive or negative criterion, perform worse than others. This confirms that both positive and negative relationships are essential when we cluster signed networks. Furthermore, we see that BalRatioCut and BalNorCut outperform SignLap and NorSignLap under every sparsity level, and the difference becomes significant when a graph becomes more sparse. This is also not surprising since SignLap and NorSignLap are not optimizing the desired clustering patterns when $k = 5$, as explained in Theorem 3.

In the second experiment, we fix the sparsity at 10^{-2} and increase noise level ϵ from 0 to 0.5. The results are shown in Figure 2. Here we observe that for $\epsilon < 0.17$, BalRatioCut and BalNorCut still give the lowest error rate, but their error rates go higher for larger ϵ. However, if we see in Figures 2b and 2c, we observe that their objectives are still quite small compared with others. The possible reason is that when noise level goes too high, antagonistic clusters are no longer significant. In such case, ground-truth clusters cannot be captured by minimizing these objectives.

In summary, we see that BalRatioCut and BalNorCut are very effective in k-way clustering and improve the state-of-the-art signed Laplacian based methods.

6.2 Multilevel Clustering

We now show that the quality of results using the multilevel clustering method discussed in Section 5 is comparable to recent state-of-the-art methods. Furthermore, multilevel clustering is much faster than other methods, so it can scale up to large-scale signed networks. We have implemented our multilevel clustering algorithm in C++. In refinement phase, we use balance normalized cut kernel for weighted kernel k-means, since it gives us the most favorable result as shown in Section 6.1. For efficiency, we do not apply local search strategy and we only consider boundary nodes in refinement phase (See Section 5.3 for details). Similarly, to make a fair comparison, all results are averaged over ten trials.

Methods. To compare our multilevel clustering with other existing state-of-the-art methods, we consider the normalized signed Laplacian (**NorSignLap**) [13]. In addition, we also consider two more approaches, **MC-SVP** and **MC-MF**, which are based on the low rank model proposed in [10]. MC-SVP uses SVP to complete the network and runs k-means on k eigenvectors of the completed matrix to get the clustering result. while MC-MF completes the network using matrix factorization and derives two low rank factors $U, H \in \mathbb{R}^{n \times k}$, and runs k-means on both U and H. We then select the clustering that gives us smaller normalized balance cut objective. See [10] for more details.

Results. First we see the quality of clustering results. Using the same procedure conducted in Section 6.1, we create some networks sampled from a complete 10-weakly balanced network, in which each group contains $1,000$ nodes. We fix $\epsilon = 0$, and compare the objective and error rate of each method under different sparsity. To make the comparison
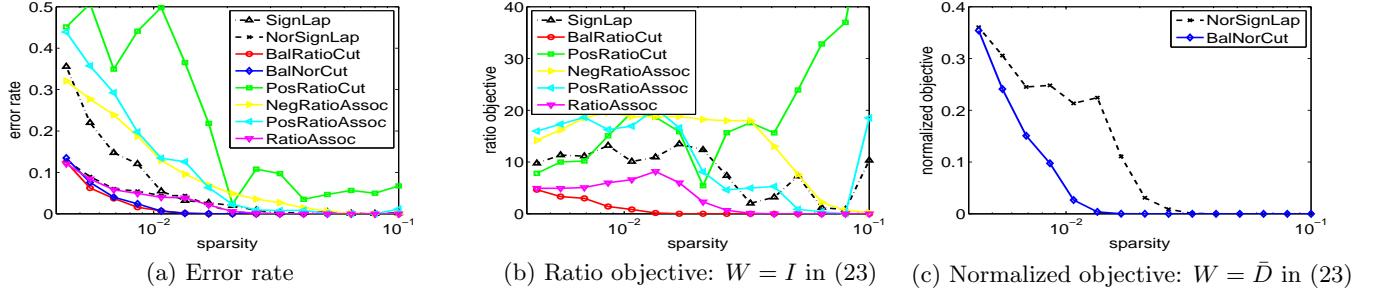
Figure 1: Spectral clustering results using different kernels on weakly balanced networks, with different sparsity.
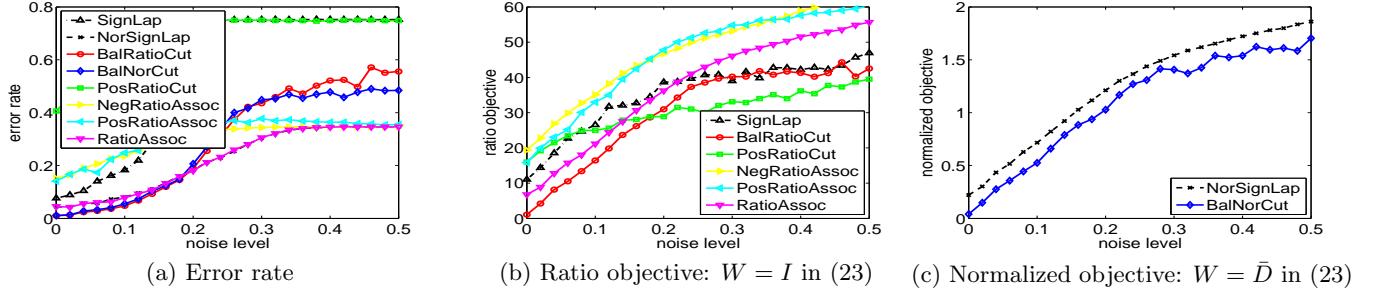
Figure 2: Spectral clustering results using different kernels on weakly balanced networks, with different noise levels.

clearer, here we consider the balance normalized objective for all methods. The result is shown in Figure 3. As a graph becomes more sparse, MC-MF and MC-SVP perform worse than NorSignLap and multilevel clustering. However, MC-MF requires much less running time than NorSignLap. We will discuss timing details issue in the next subsection. In addition, the multilevel clustering outperforms other state-of-the-art methods in most cases. It usually achieves the lowest error rate and normalized objective.
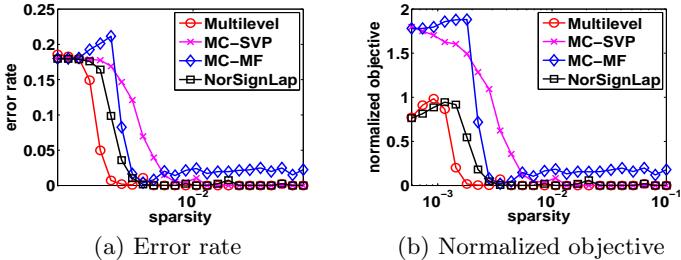
Figure 3: Clustering results for multilevel clustering and other state-of-the-art methods on weakly balanced networks, with different sparsity.

Now let us compare the efficiency of different methods. We consider A_{com} to be a large balanced network, which contains 20 groups, with $50,000$ nodes in each group. By choosing a proper sparsity, we randomly sample some edges from A_{com} to form A with desired number of edges. We then measure the running time required for each clustering algorithm under different A. Both NorSignLap and MC-SVP are too costly to apply on this large-scale network, since both of them require eigenvectors computation of top k eigenvectors. Therefore, here we only compare the multilevel algorithm with MC-MF method. While we report the running time of whole procedure for multilevel clustering, we only report the

time for computing two factors U and H for MC-MF. Thus, the MC-MF time we report is just an under-estimate of the real clustering time, since we ignore the time for doing k-means clustering on U and H. To solve matrix factorization efficiently, we implement alternating least squares (ALS) algorithm in C++ with MKL library, and run ALS iterations 3 times to derive U and H. In practice, ALS usually needs more than 3 iterations to obtain converged U and H, which will take more time than we report.

The running time of the multilevel clustering and MC-MF is shown in Figure 4. We can see that the multilevel clustering is faster than MC-MF even if we only consider the time for matrix factorization procedure of MC-MF. In particular, we can obtain the clustering result of a 20-weakly balanced network, with 1M nodes and 100M edges (so, sparsity = 10^{-4}), in only 398.77 seconds. We further plot the clustering result of this network in Figure 5. By reordering nodes based on clustering result, we can see A^{+} becomes nearly block-diagonal while A^{-} is pretty dense in the between-cluster part. This reconfirms that our multilevel algorithm is both efficient and effective.

Finally, we apply our multilevel clustering algorithm to large-scale real networks considered by Leskovec et al. [14], in which the authors argue that there are no significant (two) nearly antagonistic groups, though local structure in these networks tends to be balanced. We now extend the consideration from balance to weak balance, and examine whether there exist k nearly antagonistic groups in these networks. We first make the network symmetric by considering $\text{sgn}(A+A')$, and take the largest connected component as the input of the multilevel clustering algorithm. We then obtain several clustering results by considering k from 3 to 30. Since we do not have the ground truth of clustering in real networks, we can only compute the "empirical" error rate on A, i.e., the percentage of misclassified edges in A under such clustering.
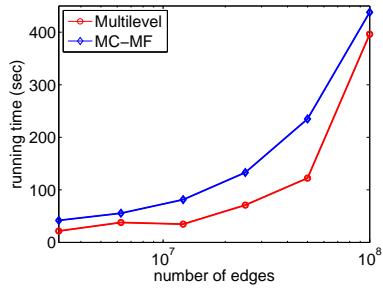
Figure 4: Running time of multilevel clustering and MC-MF on weakly balanced networks, with 1 million nodes and 100 million edges. For MC-MF we report the time to solve matrix factorization only, while in multilevel clustering we report the time for the whole procedure.
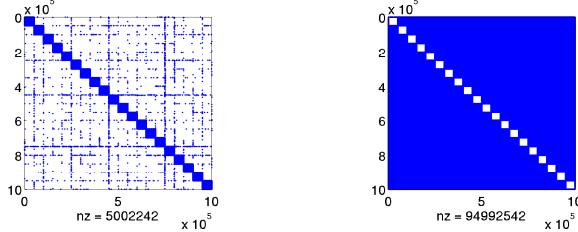
Figure 5: Clustering result of a 20-weakly balanced network, with 1M nodes and 100M edges. We see that most positive edges are within clusters and most negative edges are between clusters. The normalized objective of this clustering is 0.032, which is much smaller than 20.

We find the empirical error rate for each k is very close to each other, so we report the average error rate for each dataset, which is summarized in Table 2. Similar to the observation in [14], the empirical error rates of k-way clustering result are trivially achievable (by putting all nodes into one cluster). This implies that we can hardly find significant k nearly antagonistic groups in these networks even if we extensively consider weak balance.

7. CONCLUSIONS

In this paper, we study the k-way clustering problem in signed networks. We first find that the state-of-the-art signed graph kernels, (normalized) signed Laplacian, suffer from a weakness when we apply them to the k-way clustering problem. Thus, we propose some new signed cut and association objectives and kernels. In particular, we show one of our proposed objectives, balance normalized cut, is both theoretically sound and experimentally effective. Furthermore, we prove that our proposed objectives are mathematically

	Epinions	Slashdot	Wikipedia
# of nodes	119,070	82,052	7,064
# of edges	1,403,138	994,052	200,590
% of + edges	0.8321	0.7641	0.7855
error rate	0.1679	0.2369	0.2186
avg time (sec)	9.860	6.830	0.450

Table 2: Multilevel clustering results on large-scale real networks. The (empirical) error rate is the percentage of misclassified edges for $k = 3$ to 30; the above results support observation in [14] that these networks do not have significant clustering structure.

equivalent to the weighted kernel k-means objective. Based on this equivalence, we propose a multilevel clustering algorithm, which is competitive to recent state-of-the-art methods in terms of the quality of the clustering result, while it is much more efficient and scalable.

8. REFERENCES

[1] P. Anchuri and M. Magdon-Ismail. Communities and balance in signed networks: A spectral approach. *IEEE/ACM ASONAM*, 2012.

[2] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56:89–113, 2004.

[3] D. Cartwright and F. Harary. Structure balance: A generalization of Heider's theory. *Psychological Review*, 63(5):277–293, 1956.

[4] J. A. Davis. Clustering and structural balance in graphs. *Human Relations*, 20(2):181–187, 1967.

[5] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, 2001.

[6] I. S. Dhillon, Y. Guan, and J. Kogan. Iterative clustering of high dimensional text data augmented by local search. In *ICDM*, pages 131–138, 2002.

[7] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors: A multilevel approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(11):1944–1957, 2007.

[8] P. Doreian and A. Mrvar. A partitioning approach to structural balance. *Social Networks*, 18(21):149–168, 1996.

[9] F. Harary. On the notion of balance of a signed graph. *Michigan Mathematical Journal*, 2(2):143–146, 1953.

[10] C.-J. Hsieh, K.-Y. Chiang, and I. S. Dhillon. Low rank modeling of signed networks. In *KDD*, 2012.

[11] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1999.

[12] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(1):291–307, 1970.

[13] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. D. Luca, and S. Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. In *SDM*, pages 559–570, 2010.

[14] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, pages 641–650, 2010.

[15] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2000.

[16] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4), 2007.

[17] B. Yang, W. Cheung, and J. Liu. Community mining from signed social networks. *IEEE Trans. on Knowledge and Data Engineering*, 19(10):1333–1348, 2007.