Signed Graph Balancing with Graph Cut

Haruki Yokota[1] *Junya Hara*[1] *Yuichi Tanaka*[1] *Gene Cheung*[2]

[1] Osaka University, Osaka, Japan
[2] York University, Toronto, Canada

***Abstract*—Signed graphs—graphs with both positive and negative edge weights—are useful to specify pairwise dissimilarities as well as similarities in data. However, unlike graph variation operators (e.g., adjacency and graph Laplacian matrices) for unsigned graphs, the spectra of signed graph variation operators are not well understood in general. The lone exception is balanced signed graphs: there exists a one-to-one mapping of eigen-pairs between a balanced signed graph and its corresponding unsigned positive graph, which means that spectral filters for well-studied positive graphs can be reused if signed graphs are balanced. In this paper, we propose a simple yet effective method to balance a signed graph. Specifically, we balance a signed graph by removing carefully chosen edges, while the cuts of positive / negative edges are minimized / maximized, respectively. Experimental results on graph signal denoising and interpolation show that our signed graph balancing algorithms achieved promising results.**

***Index Terms*—Signed graph, graph signal processing**

I. INTRODUCTION

Graph signal processing (GSP) is a field of study where collections of discrete data on a network are treated as signals on a graph, and are analyzed based on their pairwise relations [1]–[3]. Given a graph embedded with pairwise signal relations as edges, various techniques of signal processing such as sampling [4], [5] and filtering [6], [7] are extended to data on a network.

Traditional GSP mainly focuses on signals on unsigned positive graphs, where *similarity* relations of incident nodes are encoded as non-negative edge weights. However, there are practical scenarios where there exist pairwise *dissimilarities* as well as similarities. A prime example is social networks, where nodes represent individuals, and their likes / dislikes of others are encoded as positive / negative edges, respectively. The resulting structure is a *signed graph*—a graph with both positive and negative edge weights [8]–[10]. Unlike unsigned graphs, signed graphs provide flexibility in modeling of data with inherent anti-correlations. Thus, extending GSP theory and applications to signed graphs is a crucial research topic.

It is known that the combinatorial graph Laplacian matrix $\mathbf{L}$ of an unsigned graph is *positive semi-definite* (PSD), i.e., its smallest eigenvalue $\lambda_{\min}(\mathbf{L}) \geq 0$ [11]. Interpreting non-negative eigenvalues $\{\lambda_n\}$ of $\mathbf{L}$ as graph frequencies, efficient graph filters in the frequency domain have been developed for different applications [12]–[14]. On the other hand, spectra of signed graph variation operators, such as adjacency and Laplacian matrices, are not well understood in general. One notable exception is *balanced* signed graphs[1]: there exists a one-to-one mapping of Laplacian eigen-pairs between a balanced signed graph and its corresponding unsigned positive graph [15]. This means that well-studied spectral filters for positive graphs can be reused for corresponding balanced signed graphs.

Given the desirability of balanced signed graphs, in this paper we propose a balancing algorithm to convert an unbalanced signed graph to a balanced one based on *graph cut* [16]. According to the *Cartwright-Harary Theorem* (CHT) [17], a signed graph is balanced when the nodes can be split into two clusters, so that edges within the clusters are positive only, and those between the clusters are negative only. Leveraging CHT, we focus on either the positive or negative edges in a given unbalanced graph, and cluster its nodes such that the smallest amount of positive edges are removed, or the majority of negative edges are preserved, when balancing the graph. We then simply remove all inconsistent edges, i.e., positive edges connecting nodes between different clusters and negative edges connecting within the cluster. Experimental results demonstrate that our graph balancing method was computationally efficient compared to an existing greedy method [18], and the balanced graph exhibited promising performance in signal denoising and interpolation.

Notation: Vectors and matrices are written in bold lowercase and uppercase letters, respectively. Operator $\|\cdot\|_p$ denotes the ℓ-p norm and $\|\cdot\|_F$ represents the Frobenius norm. The (i, j) element of the matrix $\mathbf{A}$ and the ith element in the vector $\mathbf{a}$ are denoted as $a_{i,j}$ and a_i. The jth column of matrix $\mathbf{A}$ is denoted as $\mathbf{a}_j$. The identity matrix and the vector of all ones are denoted as $\mathbf{I} = \mathrm{diag}(1, \ldots, 1)$ and $\mathbf{1} = [1, \ldots, 1]^\top$.

II. PRELIMINARIES

A. Signed Graphs

A signed graph is a weighted graph with arbitrary real valued edge weights. Mathematically, an undirected graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with set of nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ and set of weighted edges $\mathcal{E} = \{w_{i,j}\}_{v_i,v_j \in \mathcal{V}}$. The weight of an edge between v_i and v_j is denoted as $w_{i,j}$. The edge weights are described with weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$. Throughout the paper, we assume $w_{i,i} = 0$, i.e., the graphs

This work is supported in part by Japan Science and Technology Agency (JST) Advanced International Collaborative Research Program (AdCORP) under Grant JPMJKB2307, JST SPRING Grant Number JPMJSP2138, and JSPS KAKENHI (22J22176, 23H01415).

[1] A graph is balanced if there exist no cycle of odd number of negative edges.

have no self loops. We denote the degree matrix of a graph with $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_N)$, where $d_i = \sum_j |w_{i,j}|$ is the absolute sum of edge weights incident on node i. The signed graph Laplacian is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$ [19].

By this definition, the signed graph Laplacian will always be a diagonally dominant matrix with non-negative diagonal elements, which ensures the Laplacian to be PSD. More specifically, let $\mathbf{x} \in \mathbb{R}^N$ be a signal on $\mathcal{G}$ (graph signal), the following inequality is always satisfied for any $\mathbf{x}$.

$$\mathbf{x}^\top \mathbf{L}\mathbf{x} = \frac{1}{2}\sum_i \sum_j |w_{i,j}|(x_i - \mathrm{sign}(w_{i,j})x_j)^2 \geq 0 \qquad (1)$$

where x_i and x_j are signal values on the node i and node j, respectively. Let $\tilde{\mathbf{L}} = \mathbf{D} - |\mathbf{W}|$ be an unsigned graph Laplacian, a constant vector $\mathbf{x} = \mathbf{1}$ corresponds to the lowest eigenvalue of $\tilde{\mathbf{L}}$, i.e., $\mathbf{1}^\top \tilde{\mathbf{L}}\mathbf{1} = 0$. In contrast, the equality in (1) will hold iff the signed graph is *balanced*.

A signed graph is said to be balanced if the product of edge weights in any cycle is positive [17]. Intuitively, a signed graph is balanced when the nodes can be split into two disjoint sets $\mathcal{V}_1$ and $\mathcal{V}_2$ whereas all edges within each set are positive, and edges between the sets are all negative. Let us define an indicator vector $\mathbf{s} = \{-1, 1\}^N$, where $s_i = 1$ if $v_i \in \mathcal{V}_1$, $s_j = -1$ if $v_j \in \mathcal{V}_2$. If the graph is balanced, for pairs of nodes (i, j) connected with a positive edge, we have $s_i = s_j$, and for pairs connected with a negative edge, we have $s_i = -s_j$. Hence, $\mathbf{s}^\top \mathbf{L}\mathbf{s} = 0$ and the equality in (1) holds.

Balanced signed graph Laplacian is known to have same eigenvalues as those of the unsigned counterpart $\tilde{\mathbf{L}}$. That is, using indicator matrix $\mathbf{S} = \mathrm{diag}(\mathbf{s})$ the relation between eigendecompositions of $\mathbf{L}$ and $\tilde{\mathbf{L}}$ can be shown as,

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top = \mathbf{S}\tilde{\mathbf{U}}\mathbf{\Lambda}\tilde{\mathbf{U}}^\top \mathbf{S}^\top = \mathbf{S}\tilde{\mathbf{L}}\mathbf{S}^\top. \qquad (2)$$

Hereafter, we calculate the normalized Laplacian of an unsigned graph as $\hat{\mathbf{L}} = \mathbf{D}^{-\frac{1}{2}}\tilde{\mathbf{L}}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{D} = \mathrm{diag}(\mathbf{W}\mathbf{1})$.

B. Signed Graph Fourier Transform

An unsigned graph Laplacian $\tilde{\mathbf{L}} \in \mathbb{R}^{N \times N}$ can be decomposed into $\tilde{\mathbf{L}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{U}}^\top$, where $\tilde{\mathbf{U}} = [\tilde{\mathbf{u}}_{\tilde{\lambda}_1}, \ldots, \tilde{\mathbf{u}}_{\tilde{\lambda}_N}]$ is the eigenvector matrix and $\tilde{\mathbf{\Lambda}} = \mathrm{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_N)$ is the eigenvalue matrix. Without loss of generality, we assume that the eigenvalues are in ascending order as $\tilde{\lambda}_1 \leq \tilde{\lambda}_2 \leq \ldots, \tilde{\lambda}_N$. Let $\mathbf{x} \in \mathbb{R}^N$ be a vector of N signals observed on N nodes. Graph Fourier transform (GFT) is defined as $\bar{\mathbf{x}} = \tilde{\mathbf{U}}^\top \mathbf{x}$ as the inner product of eigenvectors and the graph signal.

In GFT, the eigenvectors and the eigenvalues are regarded as a Fourier basis of signals and its graph frequencies, respectively [1]. For the signed graph Laplacian $\mathbf{L}$, GFT can also be defined with $\mathbf{U}$ but its oscillation patterns of $\mathbf{U}$ are different from those of $\tilde{\mathbf{U}}$. With (2), we can define GFT for a balanced signed graph from the GFT of its corresponding unsigned graph as $\bar{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}, \quad$ where $\mathbf{U} = \mathbf{S}\tilde{\mathbf{U}}$.

C. Signed Graph Learning

Given a data matrix $\mathbf{X} = \{\mathbf{x}_k\}_{k=1}^K$ where $\mathbf{x}_k \in \mathbb{R}^N$ is the kth observation of a set of graph signals, the generalized graph Laplacian $\mathbf{L}$ can be estimated from data using the following formulation of graphical Lasso (GLASSO) [20], [21]:

$$\min_{\mathbf{L} \in \mathcal{L}} -\log\det(\mathbf{L}) + \mathrm{tr}(\mathbf{L}\bar{\mathbf{C}}) + \rho\|\mathbf{L}\|_1, \qquad (3)$$

where $\bar{\mathbf{C}} = \frac{1}{N}\mathbf{X}\mathbf{X}^\top$ is the empirical covariance matrix calculated from data, and $\mathcal{L}$ is a set of positive-definite matrices. The optimization problem in (3) is convex and can be solved using a variant of block coordinate descent (BCD) algorithm [22]. Specifically, we update the row/column of a dual variable $\mathbf{C} = \mathbf{L}^{-1}$ until convergence. We find the optimal solution for the row/column-wise subproblem with alternating direction method of multipliers (ADMM).

III. Signed Graph Balancing

In this section, we introduce two proposed graph balancing algorithms based on a graph cut perspective. We first describe the problem formulation, followed by descriptions of the proposed methods inspired by graph cut.

A. Problem formulation

Let us consider a signed graph $\mathcal{G}$ with its signed Laplacian $\mathbf{L}$. The idea behind our proposed method is that graph balancing can be regarded as a *bipartition of the signed graph*. As previously mentioned, if the signed graph is balanced, all positive edges are located within two distinct clusters and no positive edges exist between the clusters. Moreover, its nodes are bipartite with respect to the negative edges, i.e., negative edges only connect two clusters and there are no negative edges within clusters.

Leveraging the fact, if we could split the graph while 1) minimizing the number of positive edges between the clusters, or 2) maximizing the number of negative edges between the clusters, a balanced graph can be achieved just by removing a small fraction of inconsistent edges. Therefore, we cast graph balancing problem as graph clustering. While graph clustering is combinatorial and NP-hard as well as a known graph balancing algorithm, there exist a number of *spectral clustering* algorithms [16].

The first problem, a minimization of positive edges between clusters, can be seen as MinCut with respect to positive edges. Moreover, the second problem, the maximization of negative edges between clusters, can be seen as MaxCut with respect to negative edges. The proposed methods are overviewed in Fig. 1.

B. MinCut-based Balancing

For MinCut-based balancing, we consider to balance a given signed graph by removing a small fraction of positive edges. This can be realized by splitting the nodes into two clusters while minimizing the number of positive edges between them, and then removing all inconsistent edges from the graph.

When splitting the nodes, the topological properties of the original graph should be mostly maintained. Therefore, we employ Normalized Cut (NCut) [16]. In NCut, the nodes are partitioned to minimize the number of edges between clusters with a constraint that the absolute sum of edge weights within

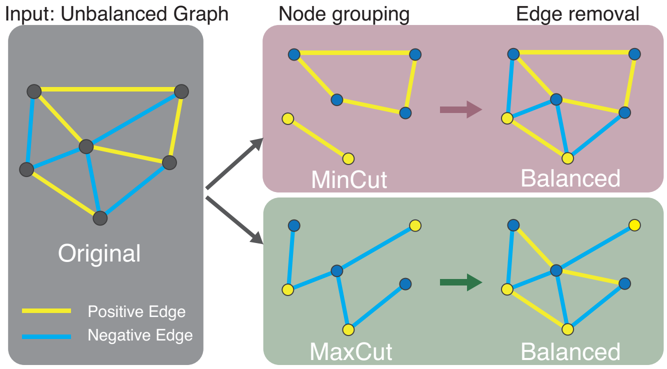


Fig. 1. Overview of the proposed graph balancing methods. The colors of nodes correspond to clusters.

each cluster is similar. We utilize this property for splitting the nodes in the signed graph.

For the first step of MinCut-based balancing, we create an unsigned graph that only contains the positive edges of the given signed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We define an unsigned graph $\mathcal{G}^{\text{pos}} = (\mathcal{V}, \mathcal{E}^{\text{pos}})$, corresponding to $\mathcal{G}$ where $\mathcal{E}^{\text{pos}}$ denotes the set of all positive edges in $\mathcal{E}$. The weighted adjacency matrix of $\mathcal{G}^{\text{pos}}$ is defined as follows:

$$\mathbf{W}^{\text{pos}} = \begin{cases} w_{i,j} & \text{if} \quad w_{i,j} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The graph Laplacian of $\mathcal{G}^{\text{pos}}$ is calculated as $\tilde{\mathbf{L}}^{\text{pos}} = \text{diag}(\mathbf{W}^{\text{pos}}\mathbf{1}) - \mathbf{W}^{\text{pos}}$ and we denote the corresponding normalized graph Laplacian as $\hat{\mathbf{L}}^{\text{pos}}$. Then we consider the NCut problem for separating the nodes of $\mathcal{G}^{\text{pos}}$ into two distinct node sets $\mathcal{V}_1$ and $\mathcal{V}_2$.

An approximated solution of NCut on $\mathcal{G}^{\text{pos}}$ is given by the eigenvector corresponding to the second smallest eigenvalue of $\hat{\mathbf{L}}^{\text{pos}}$, i.e., Fiedler vector [23]. As a result, the proposed MinCut-based balancing is computed as follows:

1) Compute the second smallest eigenvalue $\lambda_2(\hat{\mathbf{L}}^{\text{pos}})$ and the corresponding Fiedler eigenvector $\hat{\mathbf{u}}_2$.
2) Classify all nodes in two clusters $\mathcal{V}_1$ and $\mathcal{V}_2$ by performing k-means clustering on $\hat{\mathbf{u}}_2$.
3) Color nodes in $\mathcal{V}_1$ as C_1 and nodes in $\mathcal{V}_2$ as C_2.
4) Remove any inconsistent (negative) edges within clusters and inconsistent (positive) edges in between clusters.

C. MaxCut-based Balancing

For MaxCut-based balancing, we consider to balance a graph by removing a small fraction of negative edges. This can be realized by splitting the nodes into two clusters while maximizing the number of negative edges between them, and then removing any remaining inconsistent edges from the graph. Such a partitioning can be achieved by solving the MaxCut problem with respect to negative edges. In contrast to MinCut problem which can be relaxed to spectral clustering, MaxCut problem is NP-complete [24] and there have been few relaxation approaches. Instead, we consider a dual of MaxCut where we separate the nodes while having the minimum number of negative edges within each cluster.

To group the nodes into two clusters so that each cluster has the fewest edges within them, we focus on the polarity of the eigenvector $\hat{\mathbf{u}}_{\text{max}}$ that corresponds to the largest eigenvalue of the normalized unsigned graph Laplacian $\hat{\mathbf{L}}$. Since $\hat{\mathbf{u}}_{\text{max}}$ represents the frequency mode of the highest graph frequency, the polarities of elements in $\hat{\mathbf{u}}_{\text{max}}$ is expected to oscillate between all the pairs of connected nodes. Hence, if we split the nodes based on the polarities in $\hat{\mathbf{u}}_{\text{max}}$, edges within the two clusters are expected to be mostly disconnected [25].

Similar to MinCut-based balancing, we first create an unsigned graph which consists of only negative edges of the signed graph. Given a signed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let us define an unsigned graph $\mathcal{G}^{\text{neg}} = (\mathcal{V}, \mathcal{E}^{\text{neg}})$, where $\mathcal{E}^{\text{neg}}$ is a set of all negative edges in $\mathcal{E}$. The weighted adjacency matrix and graph Laplacian of $\mathcal{G}^{\text{neg}}$ is defined as

$$\mathbf{W}^{\text{neg}} = \begin{cases} -w_{i,j} & \text{if} \quad w_{i,j} < 0 \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The graph Laplacian of $\mathcal{G}^{\text{neg}}$ is calculated as $\tilde{\mathbf{L}}^{\text{neg}} = \text{diag}(\mathbf{W}^{\text{neg}}\mathbf{1}) - \mathbf{W}^{\text{neg}}$. Its corresponding normalized Laplacian matrix is denoted as $\hat{\mathbf{L}}^{\text{neg}}$. We perform a bipartition of $\mathcal{G}_{\text{neg}}$ with respect to the polarities of the elements in $\hat{\mathbf{u}}_{\text{max}}$ corresponding to the largest eigenvalue of $\hat{\mathbf{L}}^{\text{neg}}$. This is expected to split the nodes having the minimum possible number of negative edges within each cluster [25].

As a result, the proposed MaxCut-based balancing is computed as follows:

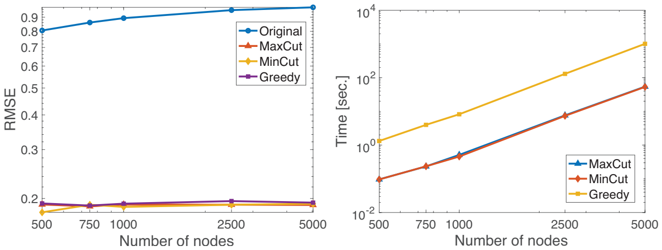
1) Compute the maximum eigenvalue $\lambda_{\text{max}}(\hat{\mathbf{L}}^{\text{neg}})$ and the corresponding eigenvector $\hat{\mathbf{u}}_{\text{max}}$.
2) Classify all nodes in two clusters $\mathcal{V}_1$ and $\mathcal{V}_2$ by performing k-means clustering on $\hat{\mathbf{u}}_{\text{max}}$.
3) Color nodes in $\mathcal{V}_1$ as C_1 and nodes in $\mathcal{V}_2$ as C_2.
4) Remove any inconsistent (negative) edges within clusters and inconsistent (positive) edges between clusters.

IV. Experimental Results

In this section, we evaluate the performance of graph balancing with 1) signal denoising based on graph lowpass filtering (LPF) and 2) signal interpolation. We compare the proposed graph balancing methods against a greedy method inspired by [18]. The greedy method (abbreviated as *Greedy* hereafter) is given below:

1) Given an unbalanced graph $\mathcal{G}$, initialize a set $\mathcal{S}$ with a randomly selected node v_i. Color v_i as C_1.
2) For all nodes incident on $\mathcal{S}$, calculate the number of consistent edges if target node is colored C_1 or C_2.
3) Find node v_j and corresponding color C_1 or C_2, in which the number of consistent edges incident on set $\mathcal{S}$ is maximized.
4) Remove inconsistent edges from v_j to $\mathcal{S}$. Add v_j to $\mathcal{S}$.
5) Repeat Steps 2 and 3 until all nodes are added to $\mathcal{S}$.

For Step 3, a random node is selected and colored C_1 if the number of consistent edges to $\mathcal{S}$ is equal or zero among all candidate nodes. *Greedy* splits nodes in two clusters while maximizing the number of edges to be preserved regardless of their signs.



(a) RMSE (b) Computation time [sec]

Fig. 2. (a) Resulting RMSE of denoising with balanced graphs, and (b) computational time of each method. Both plots are in log-scale.

Signal Denoising

Settings: We use synthetic data to validate the benefit of balancing graphs by graph LPF-based denoising [26]. We compare relative mean squared errors (RMSEs) between the original and filtered signals. We also compare the computation time for graph balancing.

We generate a synthetic unbalanced signed graph $\mathcal{G}_{\text{synth}}$ based on the stochastic block model (SBM) [27] with two communities $\mathcal{V}_1$ and $\mathcal{V}_2$ as follows:

1) Generate a SBM graph $\mathcal{G}_{\text{base}}$ having N nodes with two communities. The edge connection probability within a cluster is set to $p = 0.15$ and that between clusters is set to $q = 0.15$.
2) Set edge weights uniformly between $[0, 1]$.
3) From $\mathcal{G}_{\text{base}}$, flip the signs for 90% of edges between the clusters and 10% of edges within clusters.

The number of nodes is set to $N = \{250, 500, 750, 1000, 2500, 5000\}$. A synthetic signal $\mathbf{x}$ is generated from $\mathcal{G}_{\text{synth}}$ as $x_i = 1$ when $i \in \mathcal{V}_1$, and $x_i = -1$ when $i \in \mathcal{V}_2$. A noisy signal is given by $\mathbf{y} = \mathbf{x} + \mathbf{n}$ where $\mathbf{n} \sim \mathcal{N}(0, \sigma^2)$ is an additive noise vector. We set the noise level to $\sigma^2 = 0.50$. Ten signed graphs are generated and 300 graph signals are simulated for each graph. The graph spectral response of the used graph filter is represented as $g(\lambda_i) = 1/(1 + \lambda_i)$.

Results: Fig. 2(a) shows the RMSE comparison. As observed, filtered signals with the balanced graphs achieve lower RMSEs than that with the unbalanced graph. This is due to the minimum eigenvalue of the unbalanced signed graph being nonzero, while the minimum eigenvalue of the balanced graph is always zero.

Fig. 2(b) summarizes the computational time among the balancing methods[2]. The proposed graph balancing methods outperform *Greedy* in computational time by the factor of 20.

Signal Interpolation

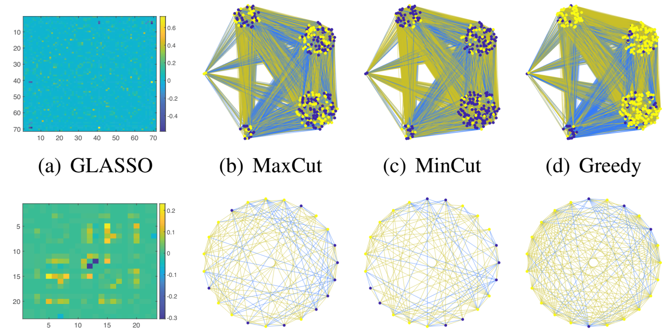
Settings: In this experiment, we validate the effectiveness of our graph balancing methods with signal interpolation [28]. We compare the RMSEs of the ground truth and interpolated signals. We describe the details of the real datasets below.

1) *Canadian Parliament Voting Records Dataset (CPV):* It consists of Canadian Parliament voting records from the 38th parliament in 2005 and contains voting records of 308 individual members voted in 190 elections. The votes were recorded as -1 for *no* and 1 for *yes* and 0 for *abstain/absent*. A signal for a vote is thus defined as $x \in \{-1, 0, 1\}^{308}$.
2) *Almanac of Minutely Power Dataset Version 2 (AMP):* It [29] contains two years of ON/OFF status data sampled at 1-minute intervals for 23 residential appliances in a Canadian household. We use 104000 recordings as graph signals. We defined -1 for status *OFF* and 1 for *ON*. Hence, a signal for a given time instance is defined as $x \in \{-1, 1\}^{23}$.

[2]All experiments, including interpolation, were performed in `MATLAB R2021b` on a PC with Apple M1 chip (8-core CPU) and 16 GB RAM.

TABLE I
CHARACTERISTICS OF ESTIMATED AND BALANCED GRAPHS.

	CPV		AMP	
Methods	$\|\mathbf{W}^+\|_F$	$\|\mathbf{W}^-\|_F$	$\|\mathbf{W}^+\|_F$	$\|\mathbf{W}^-\|_F$
Original	16.35	1.885	0.6810	0.4489
MaxCut	11.62	1.276	0.5465	0.1044
MinCut	13.29	0.7898	0.3589	0.0536
Greedy	15.60	1.136	0.6194	0.4225



(a) GLASSO (b) MaxCut (c) MinCut (d) Greedy

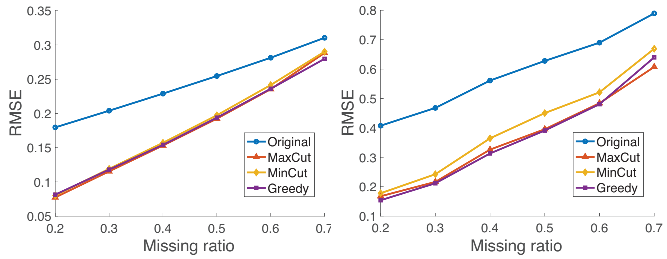
(e) GLASSO (f) MaxCut (g) MinCut (h) Greedy

Fig. 3. (a)(e) A part of weighted adjacency matrix estimated with GLASSO. The color indicate edge weights. (b)(f) Graph balanced with *MaxCut*. (c)(g) Graph balanced with *MinCut*. (d)(h) Graph balanced with *Greedy*. The clusters are plotted in yellow and blue. Pale yellow (light blue) colored edges indicate positive (negative) edge weight. The nodes are clustered according to their political parties. The rows correspond to different dataset, CPV (top) and AMP (bottom).

First, the unbalanced graph is estimated from the empirical covariance matrix with the first 90% recordings using GLASSO mentioned in Section II-C. Then, the estimated signed graph is balanced using *MaxCut*, *MinCut*, and *Greedy*. The remaining 10% of data are used for interpolation where a fraction of signal values is set to zero as missing values. The missing ratio is set to $s = \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$. We follow the formulation from [28] for signal interpolation.

Results: First, we show the characteristics of graphs balanced with different methods. The Frobenius norm of positive and negative edge weights for the estimated and balanced graphs is summarized in Table I. The balanced graphs as well as a part of the adjacency matrix of the estimated graphs are visualized in Fig. 3.

The adjacency matrices in Figs. 3(a) and 3(e) show that both real world data have strong anti-correlation among some pairs of nodes. From Table I and Fig. 3, we observe that 1) *MinCut* preserves more positive edges than *MaxCut*, and 2) *MaxCut* preserves more negative edges than *MinCut*. In terms of Frobenius norm of the remaining edges, *Greedy* is comparable with *MinCut* and *MaxCut*.



(a) CPV (b) AMP

Fig. 4. Signal interpolation RMSEs for estimated graph (Original) and balanced graphs, under different missing ratios. The RMSEs are averaged over 30 independent runs.

In Fig. 4, we show the RMSEs of signal interpolation results on the different graphs. For both datasets, interpolation with balanced graphs performs better than that with the unbalanced graph. Although the graphs balanced with *Greedy* have more edges than those with *MinCut* and *MaxCut* as shown in Table I, all balanced graphs have shown similar results for signal interpolation with CPV dataset. For AMP dataset, *MaxCut* and Greedy has comparable RMSEs and are slightly better than *MinCut*. On the other hand, the performance of *MinCut* has dropped for AMP dataset. This suggests that remaining negative edges affect the performance: All the balancing methods have comparable $\|\mathbf{W}^-\|_F$ for Canadian House dataset, while *MinCut* removes many negative edges for AMP dataset. Further investigation of the balancing effect is left for our future work.

V. Conclusion

In this paper, we present two algorithms for balancing signed graphs. The proposed methods are based on graph cuts, and is computationally efficient compared to the existing greedy method. In the applications with synthetic and real data, we demonstrated that the balanced graphs with the proposed algorithm exhibit promising results for graph signal denoising and interpolation.

VI. Acknowledgement

We thank Dr. Chinthaka Dinesh and Dr. Saghar Bagheri at York University for providing graph learning datasets.

References

[1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[2] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph Signal Processing: Overview, Challenges, and Applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[3] Y. Tanaka, Y. C. Eldar, A. Ortega, and G. Cheung, "Sampling Signals on Graphs: From Theory to Applications," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 14–30, 2020.

[4] A. Anis, A. Gadde, and A. Ortega, "Efficient Sampling Set Selection for Bandlimited Graph Signals Using Graph Spectral Proxies," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3775–3789, 2016.

[5] A. Sakiyama, Y. Tanaka, T. Tanaka, and A. Ortega, "Eigendecomposition-Free Sampling Set Selection for Graph Signals," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2679–2692, 2019.

[6] H. E. Egilmez and A. Ortega, "Spectral anomaly detection using graph-based filtering for wireless sensor networks," in *2014 IEEE Int. Conf. Acoust. Speech Signal Process. ICASSP*, 2014, pp. 1085–1089.

[7] M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka, "Graph Signal Denoising via Trilateral Filter on Graph Spectral Domain," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 137–148, 2016.

[8] Y. Hou, J. Li, and Y. Pan, "On the Laplacian Eigenvalues of Signed Graphs," *Linear Multilinear Algebra*, vol. 51, no. 1, pp. 21–30, 2003.

[9] L. Wu, X. Ying, X. Wu, A. Lu, and Z.-H. Zhou, "Spectral Analysis of k-Balanced Signed Graphs," in *Adv. Knowl. Discov. Data Min.*, J. Z. Huang, L. Cao, and J. Srivastava, Eds., Berlin, Heidelberg, 2011, Lecture Notes in Computer Science, pp. 1–12, Springer.

[10] T. Dittrich and G. Matz, "Signal Processing on Signed Graphs: Fundamentals and Potentials," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 86–98, 2020.

[11] G. Cheung, E. Magli, Y. Tanaka, and M. Ng, "Graph spectral image processing," in *Proceedings of the IEEE*, May 2018, vol. 106, no.5, pp. 907–930.

[12] J. Pang and G. Cheung, "Graph Laplacian regularization for inverse imaging: Analysis in the continuous domain," in *IEEE Transactions on Image Processing*, April 2017, vol. 26, no.4, pp. 1770–1785.

[13] X. Liu, G. Cheung, X. Wu, and D. Zhao, "Random walk graph Laplacian based smoothness prior for soft decoding of JPEG images," in *IEEE Transactions on Image Processing*, February 2017, vol. 26, no.2, pp. 509–524.

[14] Y. Bai, G. Cheung, X. Liu, and W. Gao, "Graph-based blind image deblurring from a single photograph," in *IEEE Transactions on Image Processing*, March 2019, vol. 28, no.3, pp. 1404–1418.

[15] C. Yang, G. Cheung, and W. Hu, "Signed Graph Metric Learning via Gershgorin Disc Perfect Alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 7219–7234, 2022.

[16] U. von Luxburg, "A tutorial on spectral clustering," *Stat Comput*, vol. 17, no. 4, pp. 395–416, 2007.

[17] F. Harary, "On the notion of balance of a signed graph.," *Mich. Math. J.*, vol. 2, no. 2, pp. 143–146, 1953.

[18] C. Dinesh, S. Bagheri, G. Cheung, and I. V. Bajić, "Linear-Time Sampling on Signed Graphs Via Gershgorin Disc Perfect Alignment," in *ICASSP 2022 - 2022 IEEE Int. Conf. Acoust. Speech Signal Process. ICASSP*, 2022, pp. 5942–5946.

[19] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. De Luca, and S. Albayrak, "Spectral Analysis of Signed Graphs for Clustering, Prediction and Visualization," in *Proceedings of the 2010 SIAM International Conference on Data Mining (SDM)*, Proceedings, pp. 559–570. Society for Industrial and Applied Mathematics, 2010.

[20] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[21] R. Mazumder and T. Hastie, "The graphical lasso: New insights and alternatives," *Electron J Stat*, vol. 6, pp. 2125–2149, 2012.

[22] S. J. Wright, "Coordinate descent algorithms," *Math. Program.*, vol. 151, no. 1, pp. 3–34, 2015.

[23] F. R. K. Chung, *Spectral Graph Theory*, American Mathematical Soc., 1997.

[24] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.

[25] D. I. Shuman, M. J. Faraji, and P. Vandergheynst, "A Multiscale Pyramid Transform for Graph Signals," *IEEE Trans. Signal Process.*, vol. 64, no. 8, pp. 2119–2134, 2016.

[26] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovacevic, "Signal denoising on graphs via graph filtering," in *2014 IEEE Glob. Conf. Signal Inf. Process. Glob.*, 2014, pp. 872–876.

[27] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Networks*, vol. 5, no. 2, pp. 109–137, 1983.

[28] I. Pesenson, "Variational Splines and Paley–Wiener Spaces on Combinatorial Graphs," *Constr Approx*, vol. 29, no. 1, pp. 1–21, 2009.

[29] S. Makonin, B. Ellert, I. V. Bajić, and F. Popowich, "Electricity, water, and natural gas consumption of a residential house in Canada from 2012 to 2014," *Sci Data*, vol. 3, no. 1, pp. 160037, 2016.