Tendency-Aware Directed Graph Clustering based on Bi-directional Connections

I. Related Works

Focused on minimizing the cut size normalized by the number of vertices (Ncut, short for the normalized cut) of K clusters, the author of [1] proposed the BGSP algorithm, which is to apply the k-means algorithm on the eigenvectors corresponding to the K smallest eigenvalues of Laplacian matrix $\mathbf{L}$. With a different normalization method, in [2], the authors proposed the Ncut algorithm. The aforementioned algorithms are seen as spectral clustering because of the use of EVD. For undirected graphs, spectral clustering is generally seen as the state-of-the-art method for graph clustering quality-wise [3].

Although it works well on undirected graphs, spectral clustering is not effective for digraphs, because for digraphs, $\mathbf{L} \neq \mathbf{L}^T$, which means that the orthonormal eigenvectors of $\mathbf{L}$ no longer exist. To do the clustering for directed graphs, one line of research is to transform digraphs to undirected graphs maintaining directionality, such as making the undirected edge weights convey the information of the directions. The simplest scheme there may be to remove the directions but keep the weights, i.e. Dir-rem [4, 5]. This method is simple and effective when most relationships are reciprocal. Another method is to build the pair-wise similarities between vertices based on their common in-links or out-links [3]. The similarity values are normalized by the degrees of the vertices, which makes the method known as the Degree Discounted (Deg-dis). Overall, these methods hold strong interability and easy to implement. But it is also likely for them to fail in the cases that do not meet their specific assumptions. The second line of the digraph clustering is to extend the models and methods designed for undirected graphs to digraphs. The authors of [6] proposed the algorithm NGA-LP to minimize the proposed directed version of Modularity function, which is a well-known measure used to do community detection heuristics [7]. The algorithm is shown to be efficient over many digraph datasets, but similar to many other methods based on Modulary functions, the algorithm suffers from the resolution limit [8], i.e., detection of small clusters may fail. An alternative model is the random walk process, such as Ran-wal introduced in [9], which focused on minimizing the stationary probability of a random walk transitioning from clusters $\mathcal{V}_i$ to the rest of the graph, or the reverse, normalized by the probabilities of the random walk staying in each cluster $\mathcal{V}_i$ respectively. Also based on the random walk process, the authors of [10] proposed a novel nonlinear Laplacian operator for unweighted digraph analysis (Non-lin), and a Markov Stability based dynamic framework is proposed in [11] to detect communities (Mar-sta). Apart from the above two lines, some research works focused on designing methods specifically for the directed structures, of which the related models in undirected graphs may not exist. The authors of [12] tried to make the nodes holding direct mutual links be clustered into the same clusters, which can be called Mutual-link (Mut-lin).

Our method falls in the second line that extends the cut function from undirected graphs to digraph clustering, but with significant differences to all above methods. We try to answer **to what extent the bi-directional connections can be utilized for digraph clustering**, and demonstrate that an effective algorithm minimizing the bi-directional connectivity between clusters is powerful on digraph clustering. Among all the existing methods, the idea of Mut-lin can be the one closest to our work. But it only considers the mutual direct links but not those connections rooted in the paths. Futhermore, it can only handle the unweighted edges but not weighted ones, which is not as general as ours.

II. Model Description

In this section, we introduce our proposed optimization objectives, and the algorithms for digraph clustering. The problems for two clusters are solved at first, and then we generalize it to other cases further.

A. Optimization Objectives

Firstly, we give a supporting lemma on minimizing NMcut.

Lemma 1. *Minimizing* $\mathrm{NMcut}\left(\mathcal{V}_1, \mathcal{V}_2\right)$ *is NP-hard.*

Proof: Minimizing $\mathrm{NMcut}\left(\mathcal{V}_1, \mathcal{V}_2\right)$ for undirected graphs is NP-hard, as proven in [1], which is a special case of the NMcut problem. Thus, it is also NP-hard. ■

The optimal solutions of NP-hard problems cannot be found within polynomial time. Our strategy is to find good enough results for it while taking acceptable computation costs. To solve the problem efficiently, we do a skillful transformation on it at first. Before this, some functions are defined: $[x]_+ = \max\{x, 0\}$ and $[x]_- = \min\{x, 0\}$, with respect to a scalar $x \in \mathbb{R}$, and the mutual loss

$$\mathrm{MLoss}\left(\mathbf{x}\right) := \sum_{i,j=1}^{N} W_{ij} \left([x_i - x_j]_+^2 + \beta \, [x_i - x_j]_-^2 \right) \qquad (1)$$

on $\mathbf{x} \in \mathbb{R}^N$ with $\beta + \alpha = 1$, as shown in Figure 1. Then, to transform the NMcut problem, we have the supporting lemma.

Lemma 2. *Minimizing* NMcut *in* (**??**) *is equivalent to*

$$\begin{aligned} &\min_{\{\mathcal{V}_1, \mathcal{V}_2\}} \mathrm{NMcut}\left(\{\mathcal{V}_1, \mathcal{V}_2\}\right) \Longleftrightarrow \\ &\min_{\mathbf{y} \in \mathbb{R}^N} \frac{\mathrm{MLoss}(\mathbf{y})}{\mathbf{y}^T \mathbf{y}}, \textit{subject to} \end{aligned} \qquad (2)$$

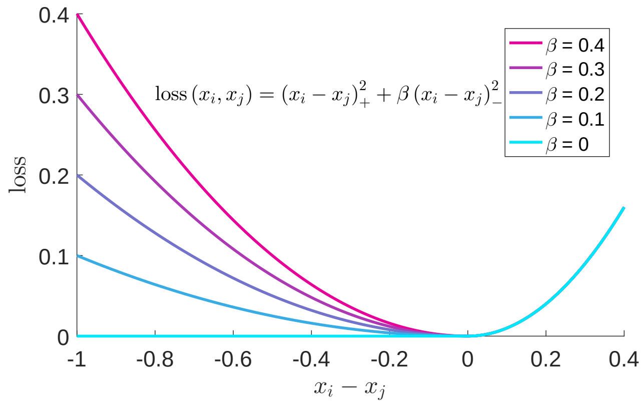

Figure 1: A novel loss function, termed the $Mutual\ Loss$, is proposed by us. The basic element of it with $W_{ij} = 1$ is shown.

the constraints $y_i \in \left\{1, -\frac{\# y_i > 0}{\# y_i < 0}\right\}$*. Note that* $y_i^\star = 1$ *indicates* $v_i \in \mathcal{V}_1^\star$*, and* $y_i^\star \neq 1$*, otherwise.*

Proof: We define an indicator variable $\mathbf{f} \in \mathbb{R}^N$: $f_i = 1$ indicates the vertex v_i belongs to $\mathcal{V}_1$ and -1, otherwise. Then with the function $b(\mathbf{x}) = \frac{\# x_i > 0}{\# x_i < 0}$ and $c(\mathbf{x}) = \mathrm{MLoss}(\mathbf{x}/2)$, we have

$$\begin{aligned}
\mathrm{NMcut}(\{\mathcal{V}_1, \mathcal{V}_2\}) = {} & \mathrm{Mcut}(\mathcal{V}_1, \mathcal{V}_2) \left(\tfrac{1}{\#\mathcal{V}_1} + \tfrac{1}{\#\mathcal{V}_2}\right) \\
= {} & [\alpha \min\{\mathrm{cut}(\mathcal{V}_1, \mathcal{V}_2), \mathrm{cut}(\mathcal{V}_2, \mathcal{V}_1)\} \\
& + (1-\alpha)(\mathrm{cut}(\mathcal{V}_1, \mathcal{V}_2) + \mathrm{cut}(\mathcal{V}_2, \mathcal{V}_1))] \\
& \times \left(\tfrac{1}{\#\mathcal{V}_1} + \tfrac{1}{\#\mathcal{V}_2}\right) \\
= {} & \min_{\mathbf{x} \in \{\mathbf{f}, -\mathbf{f}\}} \tfrac{\#(x_i > 0) + \#(x_i > 0)}{(\# x_i > 0)(\# x_i < 0)} c(\mathbf{x}) \\
= {} & \min_{\mathbf{x} \in \{\mathbf{f}, -\mathbf{f}\}} \tfrac{(\#(x_i > 0) + \#(x_i < 0))^2 c(\mathbf{x})}{N (\# x_i > 0)(\# x_i < 0)} \\
= {} & \min_{\mathbf{x} \in \{\mathbf{f}, -\mathbf{f}\}} \tfrac{(\#(x_i > 0) + \#(x_i < 0))^2 c(\mathbf{x})}{\#(x_i > 0)(\#(x_i < 0))^2 + (\#(x_i > 0))^2 \#(x_i < 0)} \\
= {} & \min_{\mathbf{x} \in \{\mathbf{f}, -\mathbf{f}\}} \tfrac{1 + 2b(\mathbf{x}) + b(\mathbf{x})^2}{\#(x_i > 0) + b(\mathbf{x})^2 \#(x_i < 0)} c(\mathbf{x}) \\
= {} & \min_{\mathbf{x} \in \{\mathbf{f}, -\mathbf{f}\}} \tfrac{(2 + 2b(\mathbf{x}))^2 c(\mathbf{x})}{4(\# x_i > 0) + 4b(\mathbf{x})^2 (\# x_i < 0)} \\
= {} & \min_{\mathbf{x} \in \{\mathbf{f}, -\mathbf{f}\}} \tfrac{\mathrm{MLoss}((1 + \mathbf{x}) - b(\mathbf{x})(1 - \mathbf{x}))}{4(\# x_i > 0) + 4b(\mathbf{x})^2 (\# x_i < 0)}.
\end{aligned}$$

The sixth equation holds by dividing $\#(x_i < 0)^2$ up and down. Denoting $\mathbf{y} = (1 + \mathbf{x}) - b(\mathbf{x})(1 - \mathbf{x})$, we have

$$\begin{gathered}
\min_{\{\mathcal{V}_1, \mathcal{V}_2\}} \mathrm{NMcut}(\{\mathcal{V}_1, \mathcal{V}_2\}) \Longleftrightarrow \\
\min_{\mathbf{f}} \min_{\mathbf{y} \in \{[(1 \pm \mathbf{f}) - b(\pm \mathbf{f})(1 \mp \mathbf{f})]\}} \frac{\mathrm{MLoss}(\mathbf{y})}{\mathbf{y}^T \mathbf{y}} \Longleftrightarrow \\
\min_{y_i \in \left\{1, -\frac{\#(y_i > 0)}{\#(y_i < 0)}\right\}} \frac{\mathrm{MLoss}(\mathbf{y})}{\mathbf{y}^T \mathbf{y}}.
\end{gathered}$$

■

B. Iterative and Divisive Algorithms

Denoting the objective function of the right-hand side problem of (2) as $g(\mathbf{y})$, we relax the original problem into $\mathbf{y} \in \mathbb{R}^N$ and keep the constraint $\mathbf{y}^T \mathbf{1} = 0$ (as implied by Lemma 2). Since $g(\mathbf{y})$ is positively scale invariant, we can add a constraint $\mathbf{y}^T \mathbf{y} = 1$ to the problem without loss of optimality, leading to the problem

$$\begin{aligned}
\underset{\mathbf{y} \in \mathbb{R}^N}{\text{minimize}} \quad & \mathrm{MLoss}(\mathbf{y}) \\
\text{subject to} \quad & \mathbf{y}^T \mathbf{y} = 1, \mathbf{y}^T \mathbf{1} = 0.
\end{aligned} \tag{3}$$

We solve (3) based on MM. At each iteration, a majorized function $\check{f}\left(\mathbf{y} | \mathbf{y}^{(t)}\right)$ around $\mathbf{y}^{(t)}$ is built for $f\left(\mathbf{y}^{(t)}\right) = \mathrm{MLoss}(\mathbf{y})$ and the corresponding majorized problem, with the original constraints, needs to be solved. $\check{f}\left(\mathbf{y} | \mathbf{y}^{(t)}\right)$ has to satisfy: $\check{f}\left(\mathbf{y}^{(t)} | \mathbf{y}^{(t)}\right) = f\left(\mathbf{y}^{(t)}\right)$, $\check{f}\left(\mathbf{y} | \mathbf{y}^{(t)}\right) \geq f(\mathbf{y})$, and $\nabla_{\mathbf{y}} \check{f}\left(\mathbf{y}^{(t)} | \mathbf{y}^{(t)}\right) = \nabla_{\mathbf{y}} f\left(\mathbf{y}^{(t)}\right)$ [13]. After this, the solution $\mathbf{y}^{(t+1)}$ to the majorized problem is used to update the problem. This process is iterated until convergence. To construct $\check{f}\left(\mathbf{y} | \mathbf{y}^{(t)}\right)$, we have the following lemma.

Lemma 3. *The following problem is a majorization of* (3)*:*

$$\begin{aligned}
\underset{\mathbf{y} \in \mathbb{R}^N}{\text{maximize}} \quad & \left(\lambda \mathbf{y}^{(t)} - \mathbf{P}\mathbf{P}^T \mathbf{y}^{(t)} + \mathbf{P}\mathbf{s}^{(t)}\right)^T \mathbf{y} \\
\text{subject to} \quad & \mathbf{y}^T \mathbf{y} = 1, \mathbf{y}^T \mathbf{1} = 0,
\end{aligned} \tag{4}$$

where $\lambda \mathbf{I} \succeq \mathbf{P}\mathbf{P}^T$*,* $\mathbf{P} \in \mathbb{R}^{N \times M}$ *is the weighted incidence matrix with binary elements being* $-\sqrt{W_{ij}}$*, and* $\mathbf{s}^{(t)} \in \mathbb{R}^M$ *is* $\mathbf{s}^{(t)} = \alpha \left[\mathbf{P}^T \mathbf{y}^{(t)}\right]_-$.

Proof: The original problem is the problem (3). The original objective function is $\mathrm{MLoss}(\mathbf{y}) = \sum_{i,j=1}^N W_{ij} \left([y_i - y_j]_+^2 + \beta [y_i - y_j]_-^2\right) = \sum_{i,j=1}^N W_{ij} [y_i - y_j]_+^2 + (1 - \alpha) \sum_{i,j=1}^N W_{ij} [y_i - y_j]_-^2$. Considering the basic function $[x]_+^2$, for $x^{(t)} > 0$, the majorized function is just x^2, for $x^{(t)} < 0$, the majorized function can be $\left(x - x^{(t)}\right)^2$. And for $(1 - \alpha)[x]_-^2$, when $x^{(t)} < 0$, we have the majorized function as $\left(x - \alpha x^{(t)}\right)^2 + const$. Thus, for $W_{ij} [y_i - y_j]_+^2$, the majorized function around $\mathbf{y}^{(t)}$ is

$$\left\| \mathbf{y}^T \mathbf{P} - \mathbf{s}^{(t)^T} \right\|_2^2, \tag{5}$$

where $\mathbf{P}$ and $\mathbf{s}^{(t)}$ are defined as before. Furthermore, for the majorized function (5), we do majorization for the second time. Based on Lemma 4 given in [14], we have the majorized function,

$$\begin{aligned}
& \left(\mathbf{y}^T \mathbf{P} - \mathbf{s}^{(t)^T}\right)^T \left(\mathbf{y}^T \mathbf{P} - \mathbf{s}^{(t)^T}\right) \\
= {} & \mathrm{Tr}\left(\mathbf{P}^T \mathbf{y}\mathbf{y}^T \mathbf{P}\right) - 2\mathbf{s}^{(t)^T} \mathbf{P}^T \mathbf{y} + const \\
= {} & \mathbf{y}^T \left(\mathbf{P}\mathbf{P}^T\right) \mathbf{y} - 2\mathbf{s}^{(t)^T} \mathbf{P}^T \mathbf{y} + const \\
\leq {} & \left[-2\left(\lambda \mathbf{y}^{(t)} - \mathbf{P}\mathbf{P}^T \mathbf{y}^{(t)}\right)^T \mathbf{y} + \lambda \|\mathbf{y}\|_2^2 + const\right] - \\
& 2\mathrm{Tr}\left(\mathbf{s}^{(t)^T} \mathbf{P}^T \mathbf{y}\right) + const \\
= {} & -2\left(\lambda \mathbf{y}^{(t)} - \mathbf{P}\mathbf{P}^T \mathbf{y}^{(t)} + \mathbf{P}\mathbf{s}^{(t)}\right)^T \mathbf{y} + const,
\end{aligned}$$

where $\lambda \mathbf{I} \geqslant \mathbf{P}\mathbf{P}^T$. Thus finally, we got the final majorized function $-2\left(\lambda \mathbf{y}^{(t)} - \mathbf{P}\mathbf{P}^T \mathbf{y}^{(t)} + \mathbf{P}\mathbf{s}^{(t)}\right)^T \mathbf{y}$, leading to the problem,

Algorithm 1 IGC Algorithm

```
Require: $N \in \mathbb{Z}_+$, $0 \le \alpha \le 1$, $\mathbf{W} \in \mathbb{R}^{N\times N}$, $\mathbf{T} \in \mathbb{R}^{N\times(N-1)}$.
1: Initialize $\mathbf{y}^{(0)} \in \mathbb{R}^N$.
2: Compute $\mathbf{P}$, $\lambda$ satisfying $\lambda\mathbf{I} \ge \mathbf{P}\mathbf{P}^T$.
3: repeat
4:     Update $\mathbf{s}^{(t)} \leftarrow \alpha\left[\mathbf{P}^T\mathbf{y}^{(t)}\right]_-$.
5:     Update $\mathbf{y}^{(t+1)} \leftarrow \frac{\mathbf{T}\mathbf{T}^T\left(\lambda\mathbf{y}^{(t)}-\mathbf{P}\mathbf{P}^T\mathbf{y}^{(t)}+\mathbf{P}\mathbf{s}^{(t)}\right)}{\left\|\mathbf{T}^T\left(\lambda\mathbf{y}^{(t)}-\mathbf{P}\mathbf{P}^T\mathbf{y}^{(t)}+\mathbf{P}\mathbf{s}^{(t)}\right)\right\|_2}$.
6: until Convergence
7: Find $p \in \mathbb{R}$ as the splitting point for $\mathbf{y}^{(t)}$.
8: return Vertex clusters $\{\mathcal{V}_1, \mathcal{V}_2\}$ seperated by $p$.
```

Algorithm 2 RGC Algorithm

```
Require: $N, K \in \mathbb{Z}_+$, $K < N$, $\mathbf{W} \in \mathbb{R}^{N\times N}$, $0 \le \alpha \le 1$.
1: Initialize the set of clusters $\mathcal{C}^{(0)} = \varnothing$, $\mathbf{W}^{(0)} = \mathbf{W}$.
2: for $t = 1 \to K$ do
3:     Apply IGC algorithm on $\mathbf{W}^{(t-1)}$ to get $\left\{\mathcal{V}_1^{(t)}, \mathcal{V}_2^{(t)}\right\}$.
4:     Update $\mathcal{C}^{(t)} \leftarrow \mathcal{C}^{(t-1)} \cup \left\{\mathcal{V}_1^{(t)}, \mathcal{V}_2^{(t)}\right\}$.
5:     Update $\mathcal{V}^{(t)} \leftarrow \arg\max_{\mathcal{V}\in\mathcal{S}^{(t)}} \mathrm{vol}(\mathcal{V})$.
6:     Update $\mathcal{C}^{(t)} \leftarrow \mathcal{C}^{(t)} - \mathcal{V}^{(t)}$.
7:     Build $\mathbf{W}^{(t)}$ from $\mathcal{V}^{(t)}$.
8: end for
9: return $\mathcal{S}^{(t)}$.
```

$$\begin{aligned} &\underset{\mathbf{y}\in\mathbb{R}^N}{\text{minimize}} && -2\left(\lambda\mathbf{y}^{(t)} - \mathbf{P}\mathbf{P}^T\mathbf{y}^{(t)} + \mathbf{P}\mathbf{s}^{(t)}\right)^T\mathbf{y} \\ &\text{subject to} && \mathbf{y}^T\mathbf{y} = 1, \mathbf{y}^T\mathbf{1} = 0 \\ &\Longleftrightarrow && \\ &\underset{\mathbf{y}\in\mathbb{R}^N}{\text{maximize}} && \left(\lambda\mathbf{y}^{(t)} - \mathbf{P}\mathbf{P}^T\mathbf{y}^{(t)} + \mathbf{P}\mathbf{s}^{(t)}\right)^T\mathbf{y} \\ &\text{subject to} && \mathbf{y}^T\mathbf{y} = 1, \mathbf{y}^T\mathbf{1} = 0, \end{aligned} \tag{6}$$

■

Next, we only need to obtain the solution to (4), which is much easier than the original problem 3. For it, we give a supporting lemma.

Lemma 4. *The closed-form solution to* (4) *is* $\mathbf{y}^\star = \frac{\mathbf{T}\mathbf{T}^T\left(\lambda\mathbf{y}^{(t)}-\mathbf{P}\mathbf{P}^T\mathbf{y}^{(t)}+\mathbf{P}\mathbf{s}^{(t)}\right)}{\left\|\mathbf{T}^T\left(\lambda\mathbf{y}^{(t)}-\mathbf{P}\mathbf{P}^T\mathbf{y}^{(t)}+\mathbf{P}\mathbf{s}^{(t)}\right)\right\|_2}$*, where* $\mathbf{T} \in \mathbb{R}^{N\times(N-1)}$ *is any constant matrix satisfying* $\mathbf{T}^T\mathbf{T} = \mathbf{I}$ *and* $\mathbf{1}^T\mathbf{T} = 0$.

Proof: Firstly, we define a new variable $\mathbf{u} \in \mathbb{R}^{N-1}$ meeting $\mathbf{T}\mathbf{u} = \mathbf{y}, \mathbf{u}^T\mathbf{u} = 1$,where $\mathbf{T} \in \mathbb{R}^{N\times(N-1)}$ satisfies: $\mathbf{T}^T\mathbf{T} = \mathbf{I}$ and $\mathbf{1}^T\mathbf{T} = 0$. Then, (4) becomes

$$\begin{aligned} &\underset{\mathbf{u}\in\mathbb{R}^{N-1}}{\text{maximize}} && \left[\mathbf{T}^T\left(\lambda\mathbf{y}^{(t)} - \mathbf{P}\mathbf{P}^T\mathbf{y}^{(t)} + \mathbf{P}\mathbf{s}^{(t)}\right)\right]^T\mathbf{u} \\ &\text{subject to} && \mathbf{u}^T\mathbf{u} = 1. \end{aligned} \tag{7}$$

From Cauchy-Schwarz's inequality, the solution to (4) is $\mathbf{y}^\star = \mathbf{T}\mathbf{u}^\star = \frac{\mathbf{T}\mathbf{T}^T\left(\lambda\mathbf{y}^{(t)}-\mathbf{P}\mathbf{P}^T\mathbf{y}^{(t)}+\mathbf{P}\mathbf{s}^{(t)}\right)}{\left\|\mathbf{T}^T\left(\lambda\mathbf{y}^{(t)}-\mathbf{P}\mathbf{P}^T\mathbf{y}^{(t)}+\mathbf{P}\mathbf{s}^{(t)}\right)\right\|_2}$. ■

Ideally, the optimal solution $\mathbf{y}^\star$ holds discrete values as given in Lemma 2. After embedding, this may not be the case, but the continuous values of $\mathbf{y}^\star$ can indicate the clustering result, on which we need to choose a splitting point to partition it into two parts. One can take 0 or the median value as the splitting point, or search for the appropriate splitting point so that NMcut is small. We take the former strategy in our work. The empirical experiments show that the latter two methods can improve the performance a little but come with more computation costs. This algorithm is named IGC (short for iterative algorithm for graph clustering).

Now, we consider the clustering with $K \ge 2$ clusters. Our basic idea is to obtain the result by recursively applying IGC on the graph from up to down until we have K partitions. Then the remaining problem is how to choose the cluster to be clustered further at each iteration. The strategy we use is to choose the cluster of the largest cardinality, since large clusters generally tend to contain smaller ones. Finally, we name this divisive algorithm for multiple clusters RGC (short for the recursive algorithm for graph clustering).

In IGC, the larger values of $\mathbf{y}^\star$ indicate that the corresponding cluster is the target cluster (the corresponding cluster tend to hold more out-paths), as implied by the proof of Lemma 2. In RGC, the tendencies between clusters are inferred by the results. At iterations, the addresses of temporary clusters can be stored in a linked list. In detail, at iteration t, the selected cluster $\mathcal{V}^{(t-1)}$ is clustered into two ones by IGC. Then, based on the tendency inferred by $\mathbf{y}^\star$, two new clusters are taken in order and replace $\mathcal{V}^{(t-1)}$ in the linked list. Thus, finally, on K clusters, the pair-wise tendencies are indicated by the positions of clusters in the linked list.

III. Experiments

In this section, we present the results of the synthetic and empirical experiments. The algorithms included are: IGC, RGC and the benchmark algorithms: Dir-rem, Deg-dis, NGA-LP, Ran-wal, Non-lin, Mar-sta, and Mut-lin. For the algorithms designed for unweighted edges, we perform linear grid searches between 0 and the largest edge weight with 20 intervals to find the optimal threshold, below which the edges are ignored. By default, IGC terminates when $\left\|\mathbf{y}^{(t+1)} - \mathbf{y}^{(t)}\right\|_{fro} < 1\times10^{-6}$. All the experiments are done on a computer with an i7-6700 3.4 GHz CPU and 8 GB RAM. The results are evaluated by NMI, of which the value is between 0 and 1, and a larger NMI means better performance [15]. NMI is defined as following:

$$\mathrm{NMI}(\Omega, \mathbb{C}) = \frac{I(\Omega; \mathbb{C})}{[H(\Omega) + H(\mathbb{C})]/2},$$

where $\Omega = \{\omega_1, \ldots, \omega_K\}$ and $\mathbb{C} = \{c_1, \ldots, c_J\}$ are two sets consisting of the clustering results and the ground-truth clusters respectively. Mutual information $I(\Omega, \mathbb{C})$ is defined as

$$I(\Omega; \mathbb{C}) = \sum_k \sum_j P(\omega_k \cap c_j) \log\frac{P(\omega_k \cap c_j)}{P(\omega_k) P(c_j)},$$

with $P(\omega_k)$, $P(c_j)$, $P(\omega_k \cap c_j)$ being the probability of a vertice belonging to ω_k, c_j, and the intersection of ω_k and c_j. And the entropy $H(\Omega)$ is defined as

Methods	Tra-mes1	Tra-mes2	Tra-tra1	Tra-tra2
Dir-rem	0.2599	0.2851	0.1930	0.2013
Deg-dis	0.3542	0.3513	0.2722	0.2654
NGA-LP	0.2987	0.3209	0.2634	0.2518
Ran-wal	0.2131	0.2368	0.1869	0.1902
Non-lin	0.2475	0.2691	0.2312	0.2135
Mar-sta	0.2683	0.2745	0.2691	0.2561
Mut-lin	0.2805	0.3147	0.2448	0.2479
RGC (Proposed)	**0.4368**	**0.4523**	**0.3542**	0.3473

Table I: NMI on Travian datasets achieved by different methods.

$$H\left(\Omega\right) = -\sum_{k} P\left(\omega_k\right) \log P\left(\omega_k\right).$$

It is obvious that a higher $I\left(\Omega;\mathbb{C}\right)$ infers a better clustering result. But considering that $K = J$ may not hold, and a larger K tends to make a higher $I\left(\Omega;\mathbb{C}\right)$, the normalized component $\left[H\left(\Omega\right) + H\left(\mathbb{C}\right)\right]/2$ is applied, which also tends to increase with K. It is proven that $\left[H\left(\Omega\right) + H\left(\mathbb{C}\right)\right]/2$ is a tight upper bound for $I\left(\Omega;\mathbb{C}\right)$, so $\mathrm{NMI}\left(\Omega,\mathbb{C}\right)$ is a real value metric between 0 and 1 for evaluting the goodness of the clustering result $\Omega = \{\omega_1, \ldots, \omega_K\}$ given the ground-truth one $\mathbb{C} = \{c_1, \ldots, c_J\}$.

A. Results on Real-world Datasets

In this subsection, we present the performance of the algorithms on three real datasets: an online social network, an email network, and a web document network, of which the properties are introduced as follows.

- **Travian** dataset contains the directed networks collected from a popular browser-based multi-player online game named Travian. There are around 1000 nodes and 5000 edges contained in the graphs. The edge weights W_{ij} represent the numbers of messages sent from the player i to j [16].
- **Email-Eu-core** dataset is a directed network generated using email data from a large European research institution [17]. The edge e_{ij} means that user i sent at least one email to user j. It contains 1005 nodes and 25571 edges. The department membership labels are treated as ground-truth clusters.
- **Wikipedia category** dataset contains a directed web graph of Wikipedia hyperlinks collected in September 2011. There are 1791489 nodes and 28511807 edges. The categories of articles are treated as the ground-truth communities [18].

Methods	Ema-Eu	Wik-cat	Avg
Dir-rem	0.4231	0.4571	0.303
Deg-dis	0.5712	0.3589	0.362
NGA-LP	**0.6948**	0.5111	0.390
Ran-wal	0.4651	0.4692	0.293
Non-lin	0.4973	0.4812	0.323
Mar-sta	0.5488	0.5128	0.355
Mut-lin	0.5032	0.4226	0.336
RGC (Proposed)	0.6882	**0.5413**	**0.468**

Table II: Clustering performance of different methods on Email-eu and Wik-cat datasets evaluated by NMI. The average performance over all the datasets is reported in the final column as well.

In all the experiments, we set the number of clusters K of different methods to be equal to that of the ground truth labels. Table I summarizes the performance of the different algorithms on the Travian dataset, which are evaluated by NMI. We use $\alpha = 0.95$ as a simple implementation of RGC but do not fine-tune it with much effort.

Firstly, two subsets in Travian are considered, the trade and the message networks. The digraphs of trade activities on the 1th and 2th of Dec, 2009 are denoted as Tra-tra1 and Tra-tra2 respectively, and those of messages on the 29th and 30th are denoted as Tra-mes1 and Tra-mes2. The experimental results are shown in Table I. It can be seen that RGC performs the best among all the methods, and are at least 20% better than Deg-dis. Secondly, in Table II, the results on the Email-Eu-core and Wikipedia category are shown, accompanied with the average result evaluated over all the datasets. On the Wikipedia datasets, our methods perform significantly better than the alternatives with at least 5% improvement on NMI. And on Email-Eu-core, RGC achieves similar performance to NGA-LP. This may be because the latter two datasets contain only unweighted edges, where RGC cannot utilize the information of the different importance of practical relationships.

B. Reviewing the Importance of Bidirectional Connections

In the above experiments, our algorithms proposed for minimizing the bi-direcitonal connectivity have shown to be effective and efficient to retrieve the clusters given the ground truth labels. But it is still valuable to question whether these advantages come from the utilization of the bidirectional connectivity. We propose to use the function NMcut over all the different clusters to give the answer. Firstly, a smaller NMcut implies that the pairs of vertices across different clusters tend to hold weaker bi-directional connecitivity. Secondly, if our proposed methods can achieve low NMcut over different synthetic and empirical datasets, the impact of introducing the relaxation are ensured to be limited and the effectiveness of the proposed algorithms can be validated.

The results are reported in Table III. $\mathcal{G}_1$ and $\mathcal{G}_2$ are following the default synthetic experimental settings with $N = 2600, p_k = 0.1, r_{42,31,41} = 0.8, r_{21,32,43} = 2, r_n = 0.2$ and $N = 2600, p_k = 0.1, r_{32,31,41,42} = 2, r_n = 0.2$ respectively. All the experiments are following the default settings in the above subsections. The results are rescaled from 0 to the maximum values among different approaches to the range between 0 and 100 for the ease of comparison. Oracle is obtained with the ground truth clusters. Basically, RGC achieves the smallest NMcut on all the datasets, which demonstrates that our algorithms are effective on decreasing the mutual connecitions between different clusters and thus retrieve plausible clusters with balanced volumes. Also, the results of Oracle keep at a low level, validating the importances of the bi-directional connections on digraph clustering. And it is worth noting that the Oracle results may not always be the smallest. This result reminds us that our proposal focused on

Methods	$\mathcal{G}_1$	$\mathcal{G}_2$	Tra-mes1	Tra-mes2	Tra-tra1	Tra-tra2	Ema-Eu	Wik-cat
Dir-rem	52.19	61.23	82.74	56.32	78.65	75.39	35.81	72.49
Deg-dis	12.69	50.84	39.38	100	42.89	36.52	61.22	100
NGA-LP	31.82	47.19	67.83	31.51	33.10	42.93	26.94	37.41
Ran-wal	100	28.51	100	79.14	100	100	100	48.56
Non-lin	81.92	24.56	82.96	84.62	85.94	76.59	62.37	38.94
Mar-sta	74.83	19.72	69.54	73.51	71.88	73.82	68.93	27.92
Mut-lin	72.36	100	70.54	47.81	69.62	51.82	72.18	88.91
RGC (Proposed)	**2.13**	**4.38**	**8.27**	**6.51**	**7.92**	**7.61**	**9.91**	**9.14**
Oracle	1.42	1.97	13.17	12.61	5.70	6.72	21.51	7.82

Table III: NMcut defined on different datasets achieved by different methods.

minimizing the bi-directional connections may not be the optimal choice for all the digraphs. As an unsupervised machine learning scheme, the framework has its own limitations and the real-world communities can exist because of more complicated causes.

References

[1] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 269–274.

[2] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[3] V. Satuluri and S. Parthasarathy, "Symmetrizations for clustering directed graphs," in *Proceedings of the 14th International Conference on Extending Database Technology*. ACM, 2011, pp. 343–354.

[4] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.

[5] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors a multilevel approach," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 11, 2007.

[6] R. Francisquini, M. C. Nascimento, and M. P. Basgalupp, "Nga-lp: A robust and improved genetic algorithm to detect communities in directed networks," in *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2018, pp. 1–8.

[7] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[8] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[9] M. Meilă and W. Pentney, "Clustering by weighted cuts in directed graphs," in *Proceedings of the 2007 SIAM international conference on data mining*. SIAM, 2007, pp. 135–144.

[10] Y. Yoshida, "Nonlinear laplacian for digraphs and its applications to network analysis," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 2016, pp. 483–492.

[11] R. Lambiotte, J.-C. Delvenne, and M. Barahona, "Random walks, markov processes and the multiscale modular organization of complex networks," *IEEE Transactions on Network Science and Engineering*, vol. 1, no. 2, pp. 76–90, 2014.

[12] Y. Li, Z.-L. Zhang, and J. Bao, "Mutual or unrequited love: Identifying stable clusters in social networks with uni-and bi-directional links," in *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 2012, pp. 113–125.

[13] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 794–816, 2017.

[14] Z. Wang, P. Babu, and D. P. Palomar, "Design of PAR-Constrained Sequences for MIMO Channel Estimation via Majorization-Minimization." *IEEE Trans. Signal Processing*, vol. 64, no. 23, pp. 6132–6144, 2016.

[15] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*. Cambridge University Press, 2008, vol. 39.

[16] A. Hajibagheri, G. Sukthankar, and K. Lakkaraju, "A holistic approach for link prediction in multiplex networks," in *International Conference on Social Informatics*. Springer, 2016, pp. 55–70.

[17] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 2, 2007.

[18] C. Klymko, D. Gleich, and T. G. Kolda, "Using triangles to improve community detection in directed networks," *arXiv preprint arXiv:1404.5874*, 2014.