Hermitian matrices for clustering directed graphs: insights and applications

Mihai Cucuringu
University of Oxford
The Alan Turing Institute

Huan Li
University of Pennsylvania

He Sun
University of Edinburgh

Luca Zanetti
University of Cambridge

Abstract

Graph clustering is a basic technique in machine learning, and has widespread applications in different domains. While spectral techniques have been successfully applied for clustering undirected graphs, the performance of spectral clustering algorithms for directed graphs (digraphs) is not in general satisfactory: these algorithms usually require symmetrising the matrix representing a digraph, and typical objective functions for undirected graph clustering do not capture cluster-structures in which the information given by the direction of the edges is crucial. To overcome these downsides, we propose a spectral clustering algorithm based on a complex-valued matrix representation of digraphs. We analyse its theoretical performance on a Stochastic Block Model for digraphs in which the cluster-structure is given not only by variations in edge densities, but also by the direction of the edges. The significance of our work is highlighted on a data set pertaining to internal migration in the United States: while previous spectral clustering algorithms for digraphs can only reveal that people are more likely to move between counties that are geographically close, our approach is able to cluster together counties with a similar socio-economical profile even when they are geographically distant, and illustrates how people tend to move from rural to more urbanised areas.

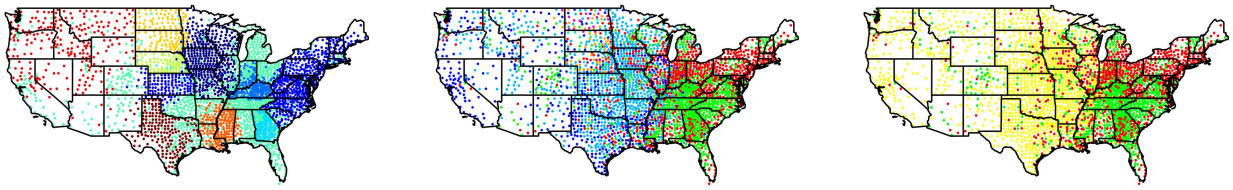
Proceedings of the 23[rd]International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

1 Introduction

Clustering is one of the most important techniques in analysing massive data sets, and has numerous applications ranging from machine learning to computer vision, from network analysis to social sciences. When the underlying graph to cluster is undirected, the objective is to partition the vertices of the graph into clusters such that vertices within the same cluster are on average better connected to one another than vertices belonging to different clusters. This notion can be formalised by introducing an objective function to minimise, such as the conductance or the normalised cut (Lee et al., 2014; Shi and Malik, 2000). For example, the widely used spectral clustering algorithm (Ng et al., 2001; Peng et al., 2017; von Luxburg, 2007), which uses eigenvectors of the adjacency matrix of a graph as input features for k-means, exploits a convex relaxation of the normalised cut to obtain a good partitioning of the graph.

However, when the underlying graph is directed, the normalised cut value and other clustering metrics based on edge-density often fail to uncover many of the significant patterns in a graph. For instance, let us consider a graph representing the number of people moving between different counties in the (mainland) United States during 1995-2000 (Census Bureau, 2002; Perry, 2003). If one tries to symmetrise its (asymmetric) adjacency matrix M in a naive way by considering $M + M^\intercal$, migration flows between counties in different states will be lost in the process. Indeed, when considering the outcome of spectral clustering on $M + M^\intercal$ of this migration data set as input, the visualisation in Figure 1a shows that clusters align particularly well with the political and administrative boundaries of the US states, as observed by Cucuringu et al. (2013). This is, somehow counterintuitively, an unsatisfactory outcome since it doesn't provide us with much information about migration patterns between far-away states.

Motivated by this example, we study spectral clustering for digraphs based on a *complex-valued* Hermitian



(a) Naive (b) Our method (c) Our method: top pair

Figure 1: Visualisation of the clustering obtained on a US migration data set: (a) spectral clustering on the symmetrised matrix $M + M^\intercal$, and (b) our procedure. The red and green clusters highlighted in (c) are such that 68% of the total weight of the edges between the two clusters is oriented from the green to the red one.

matrix representations considered by Guo and Mohar (2017); Singer (2011) and defined as follows: for any N-vertex digraph G, the Hermitian adjacency matrix $A \in \mathbb{C}^{N\times N}$ of G is the matrix where $A_{u,v} = \overline{A_{v,u}} = i$ if there is a directed edge $u \rightsquigarrow v$, and $A_{u,v} = 0$ otherwise, where i is the imaginary unity. Because of the use of i and $\bar{i}$ in expressing a directed edge, all the eigenvalues of A are real-valued. We show that, when the edge directions impart a cluster-structure on G, this structure is approximately encoded in the eigenvectors associated with the top eigenvalues of A. To demonstrate the significance of our Hermitian adjacency matrix, Figure 1b visualises the outcome of spectral clustering when A is used to encode the migration data set. It is clear such clustering is much less correlated with state boundaries than the one from Figure 1a. Furthermore, in Figure 1b we can observe several interesting migration patterns emerging, especially when considering pairs of clusters with large "imbalance" between their edge directions. The pair with the largest such imbalance (which we formalise in a later section) is shown in Figure 1c, showcasing that people tend to move from counties in green towards counties in red. In particular, Figure 1c highlights a migration pattern around the East Coast, where people tend to move from, for example, North and South Carolina to geographically distant areas such as the New York metropolitan area, Chicago, and the East side of Florida. From this perspective, while previous algorithms identify different clusters based on the relations between vertices in a cluster and vertices outside a cluster, our algorithm uncovers "higher-order" structures between clusters. In contrast to all the previous spectral algorithms for digraphs we experimented with, only our approach is able to uncover such patterns in this data set.

Our contributions and the organisation of this paper are as follows. In Section 2 we generalise the classical stochastic block model (SBM) to the setting of digraphs, and propose a directed stochastic block model (DSBM) with a latent structure defined with respect to imbalanced cuts between the clusters. In contrast to the classical SBM, the additional parameters of our model are used to assign different probabilities to the directions of the edges across different clusters. As graphs from the DSBM possess a ground truth clustering, this model will be used to analyse the performances of our algorithm. In Section 3 we present a spectral clustering algorithm for digraphs, and compare our algorithm with previous approaches. To convince the reader of the effectiveness of our algorithm, in Section 4 we provide theoretical guarantees for our algorithm when applied to a broad class of DSBMs. Complementing the theoretical analysis of our algorithm, in Section 5 we empirically demonstrate its practicality, and compare its performance against several competing approaches on synthetic and real-world data sets. We propose directions for future work in Section 6.

Related work. Because of its comprehensive applications and intriguing theoretical properties, graph clustering has received immense attention over the years. Now we review some related works most related to ours, and we refer the reader to Fortunato (2010) for a more comprehensive introduction.

First of all, we remark that while clustering undirected graphs has received most of the attention, the problem of clustering directed graphs is much less studied. Chung (2005) proposes a Cheeger inequality for digraphs, which relates the spectrum of a Laplacian operator to a notion of connectivity measuring how well a "flow" can spread through a digraph, where this flow is defined according to the stationary distribution of a random walk on the digraph. Finding clusters that minimise this measure would amount to find regions of the digraph with a limited amount of flow circulating between them. This is almost opposite to our objective: we want to uncover regions characterised by a strong and imbalanced flow circulating among them.

While graph clustering has been classically used to uncover structural information between nodes of a network, our work lies in a recent line of research that tries to uncover a higher-order structure between different groups of nodes in a network. For example, Benson et al. (2015) and Benson et al. (2016) propose tensor and spectral-based algorithms to find clusters in a (directed) graph so that small groups of nodes in the same cluster are more likely to form motifs selected by the user (such as triangles or small oriented cycles) than

groups of nodes belonging to different clusters. This is different from our aim: instead of preserving substructures *inside* clusters, our main focus is the relationships *among* clusters.

Finally, the works on co-clustering (Rohe et al., 2016) and bibliographic symmetrisation (Satuluri and Parthasarathy, 2011) are probably the most closely related to ours. We defer a more detailed comparison of these works to Section 3.

Notation. For any unweighted and directed graph G with N vertices, the Hermitian adjacency matrix of G is the matrix $A \in \mathbb{C}^{N\times N}$, where $A_{u,v} = \overline{A_{v,u}} = i$ if there is a directed edge from u to v, expressed by $u \rightsquigarrow v$, and $A_{u,v} = 0$ otherwise. When G is a weighted digraph with weight $w_{u,v}$ on any edge $u \rightsquigarrow v$, we define $A_{u,v} = (w_{u,v} - w_{v,u})i$. Notice that A is a Hermitian matrix, and therefore has N real-valued eigenvalues $\{\lambda_j\}_{j=1}^N$. We order these eigenvalues $|\lambda_1| \geq \ldots \geq |\lambda_N|$, and the eigenvector associated with λ_j is denoted by $g_j \in \mathbb{C}^N$ with $\|g_j\| = 1$, for $1 \leq j \leq N$. For any $y \in \mathbb{C}^N$, the complex conjugate of y is expressed by y^*. For any Hermitian matrix A, the image of A is denoted by $\mathrm{Im}(A)$ and the spectral norm of A is denoted by $\|A\|$. We use $\mathbf{1}_{k\times k}$ to express the $k \times k$ matrix where all the entries are 1. For ease of discussion, we always label the clusters, as well as the rows and columns of the matrix $F \in \mathbb{R}^{k\times k}$ introduced later, from 0 to $k-1$.

2 Directed stochastic block model

We study graphs generated from the directed stochastic block model (DSBM) defined by k, n, p, q, and matrix $F \in [0,1]^{k\times k}$, where $k \geq 2$ represents the number of clusters, n the number of vertices in each cluster, $p \in [0,1]$ the probability there is an edge between two vertices within the same cluster, $q \in [0,1]$ the probability there is an edge between two vertices belonging to two different clusters, while $F \in [0,1]^{k\times k}$ controls the edge orientations among clusters and satisfies $F_{\ell,j} + F_{j,\ell} = 1$ for any $0 \leq \ell, j \leq k-1$. This implies that $F_{\ell,\ell} = 1/2$ for any $0 \leq \ell \leq k-1$. The set $\mathcal{G}(k, n, p, q, F)$ consists of graphs G generated as follows: every $G \in \mathcal{G}$ is a directed graph defined on vertex set $V = \{1, \ldots, N\}$, where $N = k \cdot n$. These vertices belong to k clusters $C_0, \ldots, C_{k-1}$, where $|C_j| = n$ for $0 \leq j \leq k-1$. For any pair of vertices $\{u, v\}$, if they belong to the same cluster, they are connected by an edge with probability p; otherwise, they are connected with probability q. Moreover, if $u \in C_\ell$ and $v \in C_j$ are connected, the direction of this edge is determined by F: the direction is set to be $u \rightsquigarrow v$ with probability $F_{\ell,j}$, and $v \rightsquigarrow u$ with probability $F_{j,\ell} = 1 - F_{\ell,j}$. By definition, the direction of an edge inside a cluster is chosen uniformly at random. The matrix F can be viewed as the adjacency matrix of a weighted directed graph which represents the *meta-graph* describing the relations between the clusters. The example below explains the roles of these parameters.

Example. Let $k = 3$, $p = q$, and

$$F = \begin{pmatrix} 1/2 & 2/3 & 1/3 \\ 1/3 & 1/2 & 2/3 \\ 2/3 & 1/3 & 1/2 \end{pmatrix}$$

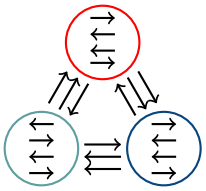


Figure 2

In this case, G consists of 3 clusters C_0, C_1 and C_2 of equal size, and any pair of vertices is connected by an edge with the same probability p. The directions of the edges inside a cluster are chosen uniformly at random, but directions of the edges crossing different clusters are chosen non-uniformly according to F. In particular, in expectation two thirds of the edges between $u \in C_j$ and $v \in C_{j+1 \bmod 3}$ are set to be $u \rightsquigarrow v$, and the remaining one third is set to be $v \rightsquigarrow u$, as shown in Figure 2. We notice that this "cyclic flow structure" of the edges across different clusters is particularly interesting, since in expectation all the vertices in G have the same in- and out-degrees, and the cluster-structure of G cannot be easily identified by the vertices' degree distribution.

Our model can be viewed as a generalisation of the classical SBM (Holland et al., 1983) into the setting of directed graphs. As a special case of our model, when $F_{\ell,j} = 1/2$ for $0 \leq \ell, j \leq k-1$, the edge directions play no role in defining a cluster-structure, and the clusters are completely determined by p and q, which is exactly the case for the SBM. On the other hand, the DSBM captures the setting where $p = q$ and the cluster structure is determined exclusively by the directions of the edges. We remark that our proposed DSBM is a special case of the co-SBM (Rohe et al., 2016), which also includes bipartite structures. We think, however, that what is lost by our model in generality is gained in clarity and simplicity.

3 Algorithm

Now we describe a spectral clustering algorithm for graphs generated from the DSBM. Given a graph $G = (V, E)$ generated from the DSBM $\mathcal{G}(k, n, p, q, F)$, our algorithm first computes the eigenvectors $g_1, \ldots, g_\ell$ corresponding to the eigenvalues λ_j satisfying $|\lambda_j| \geq \epsilon$ for some parameter ϵ. Secondly, the algorithm constructs a matrix P which is the projection matrix on the subspace spanned by $g_1, \ldots, g_\ell$, and applies k-means with the rows of P as input features. Finally, the algorithm partitions the vertex set of G based on the output of k-means. See Algorithm 1.

We remark that the number ℓ of eigenvectors used by

Algorithm 1 Spectral clustering for digraphs

Require: directed graph $G = (V, E)$ with Hermitian adjacency matrix A; $k \geq 2$; $\epsilon > 0$

1: Compute the eigenpairs $\{(\lambda_i, g_i)\}_{i=1}^{\ell}$ of A with $|\lambda_i| > \epsilon$.
2: $P \leftarrow \sum_{j=1}^{\ell} g_j g_j^*$
3: Apply k-means with input the rows of P.
4: Return a partition of V based on the output of k-means.

the algorithm depends on the parameters of the model, and in particular on the rank of F which defines the direction of the edges among different clusters. In general, $\ell \leq k$, but for practical purposes one can simply set $\ell = k$.[1] However, to obtain the optimal theoretical guarantees, at least for the case of $p = q$, we set $\epsilon = 10\sqrt{pn \log(pn)}$, whose value can be easily estimated with high probability since the average degree in the graph concentrates around pkn when $p \gg 1/n$. As it will become clear from our following analysis, in this way ℓ is set as the rank of F, without the need to actually know F. We also notice that including all the eigenvectors corresponding to the same eigenvalue in absolute value ensures that P is a *real* matrix. This follows from A being skew-symmetric. We also add that using the nk-dimensional embedding given by the rows of P is analogous to using the ℓ-dimensional embedding given by the rows of U, where U is the eigendecomposition of $P = UU^{\intercal}$.

Comparison with other spectral methods. We compare our algorithm with other spectral methods for digraph clustering that are based on the real-valued adjacency matrix M of an unweighted digraph $G = (V, E)$, defined as follows: for any pair of vertices u, v, $M_{u,v} = 1$ if $u \rightsquigarrow v$ and $M_{u,v} = 0$ otherwise. While Algorithm 1 exploits the top eigenvectors of the Hermitian adjacency matrix $A = (M - M^{\intercal}) \cdot i$, previous spectral clustering algorithms for directed graphs (Malliaros and Vazirgiannis, 2013; Rohe et al., 2016; Satuluri and Parthasarathy, 2011) typically use eigenvectors of $M^{\intercal}M$, $MM^{\intercal}$, or $M^{\intercal}M + MM^{\intercal}$ (or a regularised version of these matrices). To compare our algorithm with previous ones, notice that for any $u, v \in V$ these matrices' corresponding entries can be written as

$$(M^{\intercal}M)_{uv} = |\{w \colon w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}|, \quad (1)$$

$$(MM^{\intercal})_{uv} = |\{w \colon u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|, \quad (2)$$

$$(M^{\intercal}M + MM^{\intercal})_{uv} = |\{w \colon w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}| + |\{w \colon u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|. \quad (3)$$

By definition, $M^{\intercal}M$ keeps track of the common "parents" between two vertices, $MM^{\intercal}$ of the common "offspring", while their sum of both. To draw a direct comparison, we study the matrix A^2, since A and A^2 share the same eigenvectors and A^2 is easier to analyse. By definition, we have that

$$\begin{aligned} A^2_{uv} = &|\{w \colon (w \rightsquigarrow u \text{ and } w \rightsquigarrow v) \\ &\qquad \text{or } (u \rightsquigarrow w \text{ and } v \rightsquigarrow w)\}| \\ &- |\{w \colon (u \rightsquigarrow w \text{ and } w \rightsquigarrow v) \\ &\qquad \text{or } (w \rightsquigarrow u \text{ and } v \rightsquigarrow w)\}|, \end{aligned}$$

which implies that A keeps track of both common parents and offspring of two vertices u, v, while assigning a penalty for every node w that is simultaneously a parent of u and an offspring of v, or vice versa. Hence, A implicitly assigns a positive weight between a pair of vertices who have more common parents and offspring than "mismatched" relations with a third vertex, and a negative weight otherwise. This peculiar behaviour is at the heart of the better performances of our algorithm on some real-world data sets compared to the state-of-the-art. Moreover, it is worth mentioning that A can implicitly keep track of both common parents and offspring without the need to perform expensive matrix multiplications as in the case of $M^{\intercal}M + MM^{\intercal}$.

Normalisation of A. When dealing with real-world data sets, a proper normalisation of the graph adjacency matrix is usually required. For a diagonal matrix D, with $D_{jj} = \sum_{\ell=1}^{N} |A_{j\ell}|$, we define $A_{\text{rw}} = D^{-1}A$, which is similar to the Hermitian matrix $A_{\text{sym}} = D^{-1/2}AD^{-1/2}$ and has N real eigenvalues. The operator A_{rw} was studied in the context of angular synchronisation and the graph realisation problem (Cucuringu et al., 2012), and by Singer and Wu (2012), which introduced Vector Diffusion Maps for nonlinear dimensionality reduction. We also notice that these Hermitian operators have been successfully used in the ranking literature. In particular, Cucuringu (2016) formulated the ranking problem as an instance of the group synchronisation problem, considered an angular embedding of $M - M^{\intercal}$ and relied on the top eigenvector of A_{rw} to recover anordering of the players.

4 Analysis

We now analyse the performance of Algorithm 1 on the DSBM. Let $G \sim \mathcal{G}(k, n, p, q, F)$ with Hermitian adjacency matrix A. For simplicity, we assume that $p = q$. We remark this condition does not simplify the problem, since in this case edge densities do not give us any information on the cluster-structure of the graph, which is entirely determined by the edge orientations. We first study the expected adjacency matrix $\mathbb{E}A$. For any $u \in C_j$ and $v \in C_\ell$, we have $(\mathbb{E}A)_{u,v} = p\,(F_{j,\ell} - F_{\ell,j}) \cdot i = p\,(2F_{j,\ell} - 1) \cdot i$. Hence,

[1] More precisely, we recommend setting $\ell = k - 1$ when k is odd, since in this case F is always rank-deficient.

$\mathbb{E}A$ is Hermitian and can be decomposed into $k \times k$ blocks. Moreover, the rank of $\mathbb{E}A$ is at most k. To analyse the spectral property of $\mathbb{E}A$, we define the matrix $\widetilde{F} = (2F - \mathbf{1}_{k\times k})\cdot i$. Observe that, if $\widetilde{\lambda} \in \mathbb{R}$ is an eigenvalue of $\widetilde{F}$ with the corresponding eigenvector $\widetilde{f} \in \mathbb{C}^k$, then $\widetilde{\lambda}pn$ is an eigenvalue of $\mathbb{E}A$ with eigenvector $f \in \mathbb{C}^{kn}$ where $f(u) = \widetilde{f}(j)$ for any $u \in C_j$.

Now we explain why Algorithm 1 works for graphs generated from the DSBM. Note that, if A is close to $\mathbb{E}A$, which is the case for most instances, then the projection on the top eigenspaces of A will be close to $P_{\mathrm{Im}(\widetilde{F})} \otimes \mathbf{1}_{n\times n}$, where $P_{\mathrm{Im}(\widetilde{F})}$ is the projection on $\mathrm{Im}(\widetilde{F})$. Therefore, it suffices to ensure that $P_{\mathrm{Im}(\widetilde{F})}$ is actually able to distinguish different clusters. Because of this, we introduce the notion of θ-distinguishing image to ensure that the rows of $P_{\mathrm{Im}(\widetilde{F})}$ are not similar to each other. Formally, for any $\theta \in [0,1]$, we say that $\widetilde{F}$ has a *θ-distinguishing image*, if it holds for any $0 \le j \ne \ell \le k-1$ that $\left\|P_{\mathrm{Im}(\widetilde{F})}(j,\cdot) - P_{\mathrm{Im}(\widetilde{F})}(\ell,\cdot)\right\| \ge \theta$. Moreover, we say that $\widetilde{F}$ has a *nondistinguishing* image if the previous equation holds only for $\theta = 0$. Proposition 1 below shows that $\widetilde{F}$ has a nondistinguishing image if and only if F has two identical rows. When $p = q$, this condition implies every graph generated from the DSBM has two statistically indistinguishable clusters.

Proposition 1. *Let $G \sim \mathcal{G}(k,n,p,q,F)$. Then, the matrix $\widetilde{F}$ defined by $\widetilde{F} = (2F - \mathbf{1}_{k\times k})\cdot i$ has a nondistinguishing image if and only if there exist $0 \le j \ne \ell \le k-1$ such that $F(j,\cdot) = F(\ell,\cdot)$.*

Our analysis is based on matrix perturbation theory, and requires that the nonzero eigenvalues of $\widetilde{F}$ are far from 0 in order to ensure that projection on the the top eigenspaces of A is close to $P_{\mathrm{Im}(\widetilde{F})} \otimes \mathbf{1}_{n\times n}$. Hence, we define the *spectral gap* of $\widetilde{F}$ by $\widetilde{\rho} \triangleq \min_{1\le j\le k}\{|\rho_j| \colon \rho_j \ne 0\}$, where $\rho_1,\ldots,\rho_k$ are the eigenvalues of $\widetilde{F}$. Note that in the standard SBM a similar definition of spectral gap governs the performance of spectral clustering algorithms (Lei and Rinaldo, 2015, Corollary 3.2). Theorem 2 bounds the number of misclassified vertices by Algorithm 1 for graphs generated from the DSBM.

Theorem 2 (Main Theorem)**.** *Let $G \sim \mathcal{G}(k,n,p,q,F)$, where $p = q$. Assume that*

$$\widetilde{\rho} \ge C\ (k/\theta)\sqrt{(1/pn)\ \log n} \tag{4}$$

holds for a large absolute constant C and $\widetilde{F}$ has a θ-distinguishing image with $\theta > 0$. Then, with high probability, the number of misclassified vertices by Algorithm 1 is $O\left(k^2/(\tilde{\rho}^2\ \theta^2\ p)\ \log n\right)$.

For a family of graphs with k fixed and n growing, as long as p is not too small, assumption (4) is always met. It also implies that, for most cluster-structure matrices F, p needs to be greater than $k^2 \log n/n$, which is comparable to the connectivity threshold $p \ge \log(kn)/(kn)$.

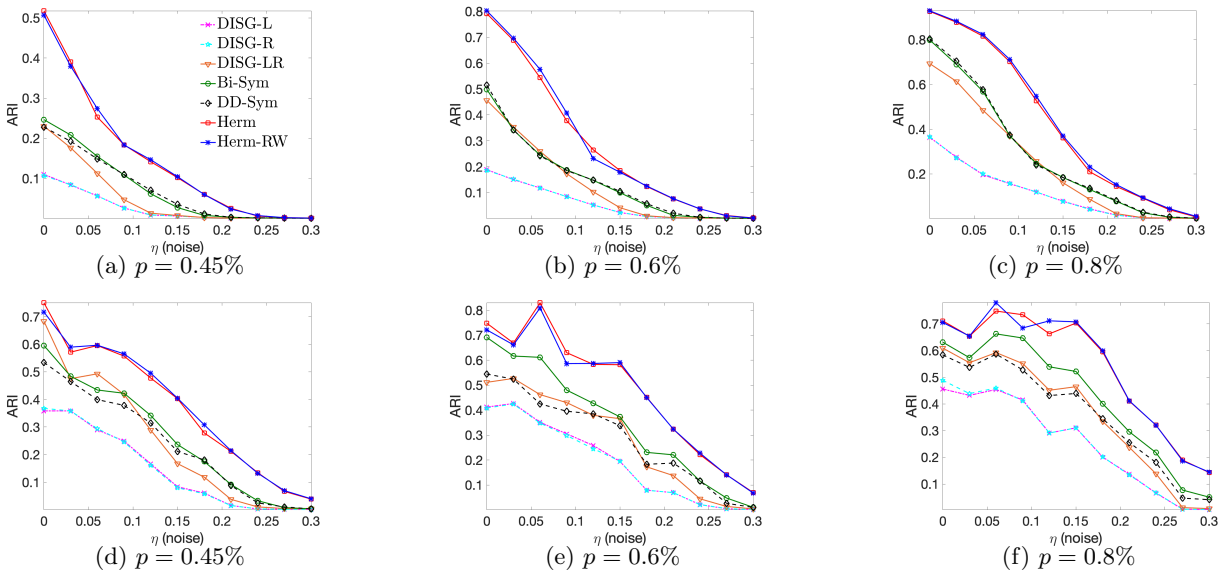
Next we evaluate the theoretical guarantee by Theorem 2 when $G \sim \mathcal{G}(k,n,p,q,F)$, $p = q$, and there exists a noise parameter $\eta \in [0,1/2)$ such that $F_{j,\ell} = 1-\eta$ if $j \equiv \ell - 1 \mod k$, $F_{j,\ell} = \eta$ if $j \equiv \ell + 1 \mod k$, and $F_{j,\ell} = 1/2$ otherwise. By definition, the connections among the k clusters can be represented by a directed cycle where each edge has weight $1-2\eta$, and hence we call this particular DSBM *the cyclic block model*. We believe this cyclic block model is particularly suitable to evaluate the performance of a clustering algorithm for digraphs due to the following reasons: (1) since every vertex of the graph has the same in-degree and out-degree in expectation, the vertices' degrees provide no information for clustering; (2) even for the case of $\eta = 1$, i.e., all the edges between two clusters C_j and $C_{j+1 \mod k}$ are oriented in the same direction, the clustering task could be still very challenging because the directions of most edges are randomly oriented. We summarise the performance of Algorithm 1 on the cyclic block model as follows.

Corollary 3. *Let G be a graph sampled from a cyclic block model with parameters $k, n, p = q = \omega\left(k^3/((1-2\eta)^2\ n)\ \log n\right)$, and $\eta \in [0,1/2)$. Then, with high probability, the number of misclassified vertices by Algorithm 1 is $O\left(k^4/((1-2\eta)^2 p)\ \log n\right)$.*

5 Experiments

We compare the performance of our algorithm with other spectral clustering algorithms for digraphs on synthetic and real-world data sets. Since ground truth clustering is available for graphs generated from the DSBM, we measure the recovery accuracy by the Adjusted Rand Index (ARI) (Gates and Ahn, 2017), which is closely related to and alleviates some of the issues of the popular Rand Index (Rand, 1971). Both measures indicate how well a recovered clustering matches the ground truth, with a value close to 1 (resp. 0) indicating an almost perfect recovery (resp. an almost random assignment of the vertices into clusters). For real-world data sets, due to the lack of a ground truth clustering, we will introduce appropriately defined new objective functions to measure the quality of a clustering, while taking the edge directions into account and aiming to uncover imbalanced cuts in the partition.

Experimental setup. We compare against the three variants of the DI-SIM algorithm (Rohe et al., 2016), and spectral clustering for digraphs when bibliometric and degree-discounted symmetrisations are applied (Satuluri and Parthasarathy, 2011). Note that all these algorithms follow the standard framework of spectral clustering, but employ different eigenvectors



(a) $p = 0.45\%$ (b) $p = 0.6\%$ (c) $p = 0.8\%$

(d) $p = 0.45\%$ (e) $p = 0.6\%$ (f) $p = 0.8\%$

Figure 3: Recovery rates for the circular pattern (top) and complete meta-graph (bottom) ($N = 5,000, k = 5$).

to construct the feature vectors for k-means++. More specifically, DI-SIM (LEFT) (denoted by DISG-L) and DI-SIM (RIGHT) (DISG-R) use, respectively, the top k eigenvectors of a regularised and normalised version of the matrix defined in (1) and (2) as input features for k-means; DI-SIM (LEFT+RIGHT) (DISG-LR) uses the top k eigenvectors of a regularised and normalised version of both matrices (1) and (2); BI-SYM and DD-SYM use the top k eigenvectors of the matrix in (3), with an additional normalisation for DD-SYM.

We also consider an additional variant of our Algorithm 1 based on a different normalisation of our Hermitian adjacency matrix. Specifically, we use HERM and HERM-RW to represent Algorithm 1 when the top eigenvectors of A and A_{rw} are applied as the input matrix, respectively. We remark that Algorithm 1 is described with respect to the non-normalised Hermitian adjacency matrix, since all the vertices of a graph generated from the DSBM have the same expected degree and normalising A is not needed. On the other hand, in real-world data sets, the degree distribution is typically very skewed with large outlier degrees and, as our experiments suggest, HERM-RW usually performs the best among the tested algorithms.

Results for the DSBM. We perform experiments on graphs randomly generated from the DSBM with different values of $n, p = q$, and matrix F. Ass spectral techniques perform better in the SBM for large p, our focus is to compare the performance of different algorithms when p is close to the connectivity threshold $\log(N)/N$ of a random $\mathcal{G}(N, p)$ graph. Our reported results are averaged over 10 independently generated graphs for every fixed parameter set. For ease of visualisation, we assume the entries of F have only three different values: $1/2$ (which corresponds to uniformly random edge-directions), η, and $1 - \eta$.

Figure 3 reports the performance of all the tested algorithms for input graphs from the DSBM with $N = 5,000$, $k = 5$, and the meta-graph is a directed cycle, or a complete graph with random orientations of the edges. The two variants of our algorithm give similar results due to the fact that all the vertices have the same expected degree, and they perform significantly better than all other algorithms. While all methods are unable to find a meaningful cluster structure when η is close to 0.3, our algorithm performs significantly better, especially for smaller values of η.

We further investigate the performance of all algorithms for a large value of k. Figure 4 reports the ARI values of a randomly generated graph with respect to different values of η, with $N = 5,000$, $k = 50$, $p = 1\%$, and the underlying meta-graph is a complete graph. This regime of parameters, i.e., large k and relatively small p, is of particular interest due to its prevalence in most real-world data sets, and clearly illustrates that our algorithm has overwhelmingly superior performance compared to the other algorithms.

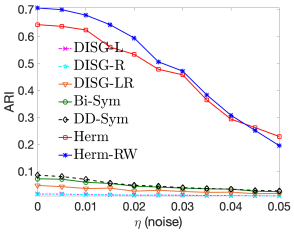


Figure 4: Complete meta-graph (DSBM, $k = 50$).

Results for real-world data. We also detail results on real-world data sets, showcasing the efficiency and robustness of our algorithm for identifying structures in

digraphs. Since no ground truth clustering is available, we compare performance as measured by three related objective functions (also referred to as scores), showing that our approach favours balanced cluster sizes. We consider a US-Migration network, and a BLOG network during the 2004 US presidential election.

For any two disjoint vertex sets X and Y, we define the Cut Imbalance ratio between X and Y by

$$\begin{aligned}\mathrm{CI}(X,Y) &= \frac{1}{2}\cdot\left|\frac{w(X,Y)-w(Y,X)}{w(X,Y)+w(Y,X)}\right| \\ &= \left|\frac{w(X,Y)}{w(X,Y)+w(Y,X)}-\frac{1}{2}\right|, \end{aligned} \tag{5}$$

where $w(X,Y)=\sum_{u\in X,v\in Y} w(u,v)$, and define the size and volume normalised versions by

$$\mathrm{CI}^{\mathrm{size}}(X,Y) = \mathrm{CI}(X,Y)\cdot\min\{|X|,|Y|\}, \tag{6}$$

$$\mathrm{CI}^{\mathrm{vol}}(X,Y) = \mathrm{CI}(X,Y)\cdot\min\{\mathrm{vol}(X),\mathrm{vol}(Y)\}, \tag{7}$$

where $\mathrm{vol}(X)$ is the sum of in-degrees and out-degrees of the vertices in X. To explain (6) and (7), notice that $\mathrm{CI}(X,Y)\in[0,1/2]$ quantifies the imbalance of the edge directions between X and Y, with $\mathrm{CI}(X,Y)=0$ (resp. $\mathrm{CI}(X,Y)=1/2$) indicating that the directions of the edges between X and Y are completely balanced (resp. imbalanced). Furthermore, since our objective is to identify pairs of clusters with a large CI-value, we scale $\mathrm{CI}(X,Y)$ by the minimum of their sizes or volumes to penalise small clusters, in the same spirit as the normalised cut value (Shi and Malik, 2000).

US-Migration *Network.* We consider the 2000 US Census data, which reports the number of people that migrated between pairs of counties in the US during 1995-2000 (Census Bureau, 2002; Perry, 2003). This data can be expressed as a matrix $M\in\mathbb{Z}_{\geq 0}^{N\times N}$, where $N=3107$ denotes the number of counties in mainland US, and $M_{j\ell}$ denotes the total number of people that migrated from county j to county ℓ. We consider the transformation $\widetilde{M}_{j\ell}=M_{j\ell}/(M_{j\ell}+M_{\ell j})$, which leads to a matrix often encountered in various applications. For example, in ranking, this could capture the fraction of games won by player j in the match against ℓ (Negahban et al., 2012). The input matrix to our pipeline is given by the skew symmetric matrix $G=\widetilde{M}-\widetilde{M}^{\intercal}$. Figure 5 shows the $\mathrm{CI}^{\mathrm{vol}}$ values for the top pairs for varying number of clusters. With respect to both scores, Herm and Herm-RW are consistently better across all top pairs, and outperform all other methods by a large margin especially for $k=10,20$.

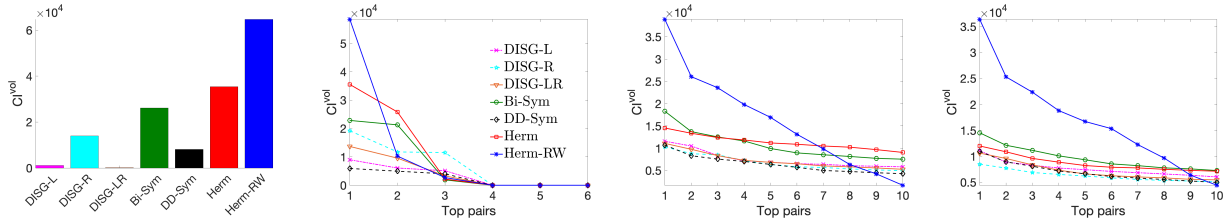
Figure 6 shows the clusterings recovered by several methods for $k=10$, and heatmaps of the adjacency matrices sorted by induced cluster membership, highlighting the fact that DISGLR and DD-Sym tend to uncover traditional clusters of high internal edge-density, as hinted by the prominent block-diagonal structure. On the other hand, Herm and Herm-RW do not exhibit such a structure, and contain block submatrices of high intensity (denoting a large cut imbalance) on the off-diagonal blocks. Figure 7 shows the three pairs of clusters for which $\mathrm{CI}^{\mathrm{size}}(C_j,C_\ell)$ is the largest. We highlighted the two clusters in each pair in red (source) and blue (destination), and provided the values for their respective CI, $\mathrm{CI}^{\mathrm{size}}$ and $\mathrm{CI}^{\mathrm{vol}}$. With respect to the two normalised cut imbalances, Herm-RW vastly outperforms all other methods.

BLOG *Network.* We consider the BLOG network from the 2004 US presidential election, as in Adamic and Glance (2005), who recorded the hyperlinks between $N=1,212$ political blogs and revealed that such connections were highly dependent on the blog's political orientation. Figure 8 shows the $\mathrm{CI}^{\mathrm{vol}}$ scores of the top pairs. We also consider the case $k=2$, as the network has an underlying structure with two clusters corresponding to the Republican and Democratic parties. The two variants of our algorithm vastly outperform other methods, with Herm-RW as the best performer.

6 Conclusions and future work

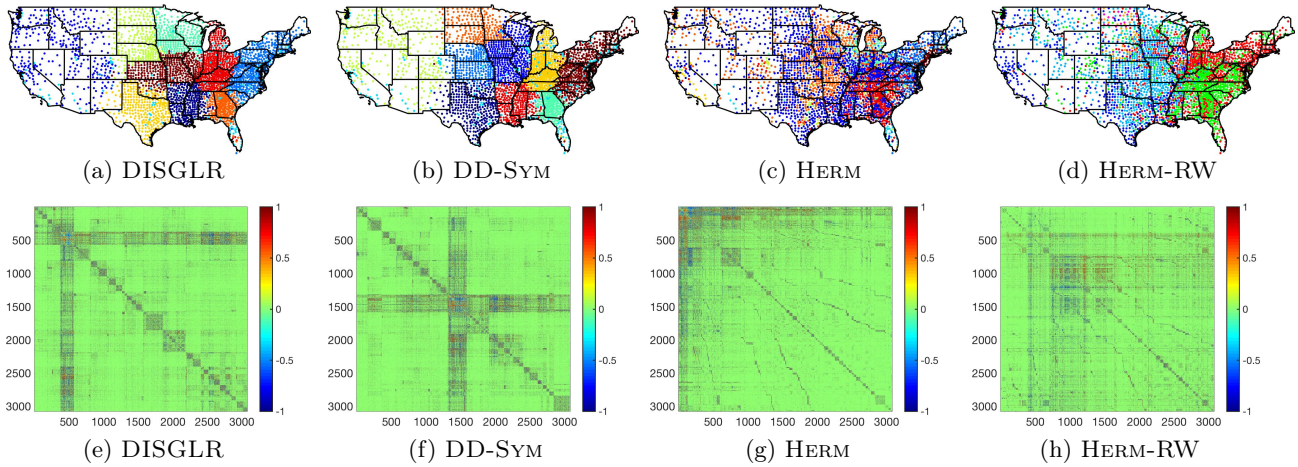
We proposed a spectral clustering algorithm for directed graphs that is able to uncover clusters characterised by strong imbalances in the direction of the crossing edges. The main theoretical gap we would like to address in future work is to further develop a connection between the Cut Imbalance Ratio measure defined in Section 5, and our spectral algorithm, in the same vein as the relation between spectral clustering and the normalised cut (Shi and Malik, 2000). However, it is unclear if such strong connection exists: while the normalised cut is the sum of the conductance of each cluster, each one a function of a single cluster, we are interested in the pairwise interactions between all pairs of clusters, a higher-order relation between vertices. For any k-way partition, this gives rise to $O(k^2)$ terms, of which, depending on the application (e.g., if the meta-graph is sparse) only a few should be considered, making it difficult to define a general relaxation for the problem.

Another issue with our approach is that it discards information given by undirected edges. This is not necessarily a drawback, since in applications where we only care about the net-flow between clusters, undirected edges do not add any information. However, it might still be interesting to develop approaches that can interpolate between clusters defined with respect to undirected edge densities and clusters defined with respect to imbalances in the orientation of the edges.



(a) $k = 2$ (b) $k = 3$ (c) $k = 10$ (d) $k = 20$

Figure 5: Top CI^{vol} scores attained by pairs of clusters, for the US-MIGRATION data set with varying k.



(a) DISGLR (b) DD-SYM (c) HERM (d) HERM-RW

(e) DISGLR (f) DD-SYM (g) HERM (h) HERM-RW

Figure 6: Top: Recovered clusterings for the US-MIGRATION data set with $k = 10$ clusters. Bottom: Heatmap of the graph adjacency matrices, sorted by induced cluster membership.

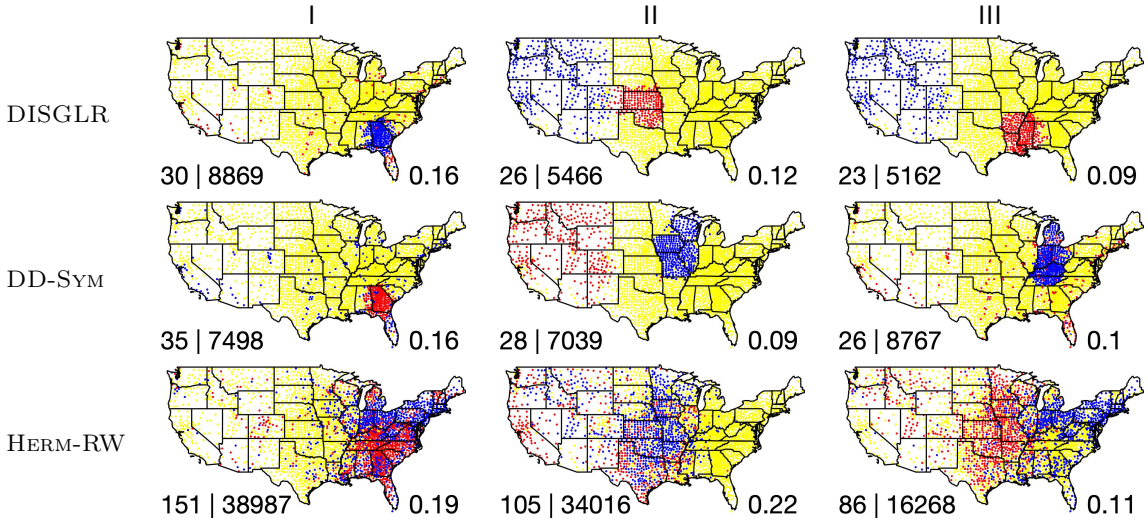
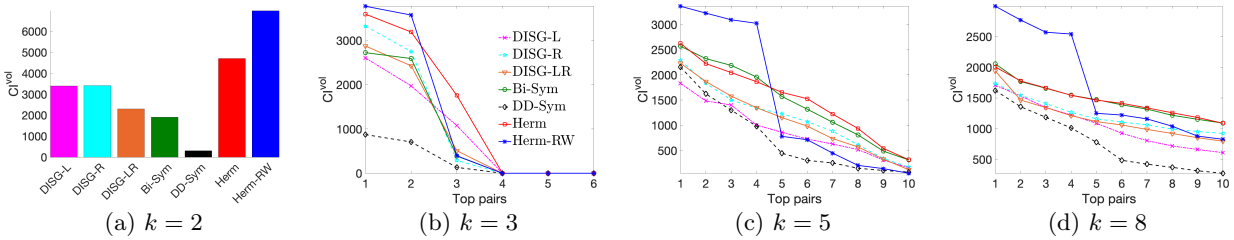


Figure 7: The top three largest size-normalised cut imbalance pairs for US-MIGRATION with $k = 10$ clusters. Red denotes the source cluster, and blue denotes the destination cluster. The bottom left of each plot shows the the normalised CI^{size} and CI^{vol} pairwise cut imbalance values, and the bottom right text contains the CI cut imbalance value in $[0, 1/2]$.



(a) $k = 2$ (b) $k = 3$ (c) $k = 5$ (d) $k = 8$

Figure 8: The top CI^{vol} scores attained by pairs of clusters, for the BLOG data set with varying k.

Acknowledgements

Mihai Cucuringu acknowledges support from the EPSRC grant EP/N510129/1 at The Alan Turing Institute. He Sun is supported by an EPSRC Early Career Fellowship (EP/T00729X/1). Luca Zanetti is supported by the ERC Starting Grant (DYNAMIC MARCH).

Bibliography

Adamic, L. A. and Glance, N. (2005). The political blogosphere and the 2004 US election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43.

Arthur, D. and Vassilvitskii, S. (2007). k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007*, pages 1027–1035.

Benson, A. R., Gleich, D. F., and Leskovec, J. (2015). Tensor spectral clustering for partitioning higher-order network structures. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 118–126. SIAM.

Benson, A. R., Gleich, D. F., and Leskovec, J. (2016). Higher-order organization of complex networks. *Science*, 353(6295):163–166.

Census Bureau, U. S. (2002). `www.census.gov/population/www.cen2000/ctytoctyflow/index.html`.

Chung, F. (2005). Laplacians and the cheeger inequality for directed graphs. *Annals of Combinatorics*, 9(1):1–19.

Chung, F. and Radcliffe, M. (2011). On the spectra of general random graphs. *Electronic Journal of Combinatorics*, 18(1).

Cucuringu, M. (2016). Sync-Rank: Robust Ranking, Constrained Ranking and Rank Aggregation via Eigenvector and Semidefinite Programming Synchronization. *IEEE Transactions on Network Science and Engineering*, 3(1):58–79.

Cucuringu, M., Blondel, V., and Van Dooren, P. (2013). Extracting spatial information from networks with low order eigenvectors. *Physical Review E*, 87.

Cucuringu, M., Lipman, Y., and Singer, A. (2012). Sensor network localization by eigenvector synchronization over the Euclidean group. *ACM Transactions on Sensor Networks*, 8(3):19:1–19:42.

Davis, C. and Kahan, W. M. (1970). The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7:1–46.

for National Statistics, O. (2018). Internal migration: detailed estimates by origin and destination local authorities, age and sex.

Fortunato, S. (2010). Community detection in graphs. *Physics reports*, 486(3-5):75–174.

Gates, A. J. and Ahn, Y.-Y. (2017). The impact of random models on clustering similarity. *Journal of Machine Learning Research*, 18(87):1–28.

Guo, K. and Mohar, B. (2017). Hermitian adjacency matrix of digraphs and mixed graphs. *Journal of Graph Theory*, 85(1):217–248.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: first steps. *Social Networks*, 5(2):109–137.

Kumar, A., Sabharwal, Y., and Sen, S. (2004). A simple linear time $(1+\varepsilon)$-approximation algorithm for k-means clustering in any dimensions. In *Proceedings of the 45th Symposium on Foundations of Computer Science*, pages 454–462.

Lee, J. R., Gharan, S. O., and Trevisan, L. (2014). Multiway spectral partitioning and higher-order Cheeger inequalities. *Journal of the ACM*, 61(6).

Lei, J. and Rinaldo, A. (2015). Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237.

Leskovec, J. and Krevl, A. (2014). SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`.

Malliaros, F. D. and Vazirgiannis, M. (2013). Clustering and community detection in directed networks: A survey. *Physics Reports*, 533(4):95–142.

Negahban, S., Oh, S., and Shah, D. (2012). Iterative ranking from pair-wise comparisons. In *Advances in Neural Information Processing Systems 25*, pages 2474–2482.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856.

Peng, R., Sun, H., and Zanetti, L. (2017). Partitioning well-clustered graphs: spectral clustering works! *SIAM Journal on Computing*, 46(2):710–743.

Perry, M. J. (2003). State-to-State Migration Flows: 1995 to 2000. *Census 2000 Special Reports*.

Rand, W. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Rohe, K., Qin, T., and Yu, B. (2016). Co-clustering directed graphs to discover asymmetries and directional communities. *Proceedings of the National Academy of Sciences*, 113(45):12679–12684.

Satuluri, V. and Parthasarathy, S. (2011). Symmetrizations for clustering directed graphs. In *Proceedings*

of the 14th International Conference on Extending Database Technology, pages 343–354.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Singer, A. (2011). Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis*, 30(1):20–36.

Singer, A. and Wu, H. T. (2012). Vector diffusion maps and the connection Laplacian. *Communications on Pure and Applied Mathematics*.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

White, J., Southgate, E., Thomson, J. N., and Brenner, S. (1986). The structure of the nervous system of the nematode c. elegans. *Philosophical transactions Royal Society London*, 314:1–340.

Supplementary material

In this supplementary material, we present a more detailed analysis of our algorithm and its performance on various additional data sets. This supplementary material is structured as follows: Section A presents all the omitted proofs of the theorems and lemmas from Section 4; in Section B, through additional experimental results, we give a detailed comparison between our algorithm and existing methods from the literature.

A Omitted proof details

In this section we present the omitted technical details about the analysis from Section 4. We first introduce some notation that will be used in the analysis. For any Hermitian matrix A and parameters $\alpha \leq \beta$, let $P_{(\alpha,\beta)}(A)$ be the projection on the subspace spanned by the eigenvectors of A with the corresponding eigenvalues in (α, β), and we define the matrix $P_{[\alpha,\beta]}(A)$ in a similar way. Notice that the matrix P defined in Algorithm 1 can be written as $P_{(-\infty,-\epsilon)\cup(\epsilon,+\infty)}(A)$.

We now state two results that will be used in the proofs below. The first is the well-known Davis-Kahan theorem, which bounds the perturbation of the eigenspaces of a matrix H subject to random noise expressed by a matrix R. It will be used in the proof of Lemma 7.

Theorem 4 (Davis-Kahan, Davis and Kahan (1970))**.** *Let $H, R \in \mathbb{R}^{d\times d}$ be Hermitian matrices. Then, for any $a \leq \beta$ and $\delta > 0$ it holds that*

$$\left\|P_{[\alpha,\beta]}(H) - P_{(\alpha-\delta,\beta+\delta)}(H+R)\right\| \leq \frac{\|R\|}{\delta}.$$

The other lemma that will be used in the analysis is the following matrix concentration inequality.

Theorem 5 (Chung and Radcliffe (2011))**.** *Let $X_1, X_2, \ldots, X_m$ be independent random $d\times d$ Hermitian matrices. Moreover, assume that $\|X_j - \mathbb{E}X_j\| \leq M$ for all j, and let $\sigma^2 = \|\sum_{j=1}^m \mathbb{E}(X_j - \mathbb{E}X_j)^2\|$. Let $X = \sum_{j=1}^m X_j$. Then, for any $a > 0$, it holds that*

$$\mathbb{P}\left[\|X - \mathbb{E}X\| > a\right] \leq 2d \exp\left(-\frac{a^2}{2\sigma^2 + 2Ma/3}\right).$$

We can now present the omitted proofs from Section 4.

Proof of Proposition 1. First of all, we assume that F is the matrix with two identical rows indexed by j and ℓ, and we prove that $\widetilde{F}$ has a nondistinguishing image. To this end, notice that

$$\widetilde{F}_{j,j} = \widetilde{F}_{\ell,\ell} = \widetilde{F}_{j,\ell} = \widetilde{F}_{\ell,j} = 0,$$

and there is an automorphism that swaps j and ℓ such that the remaining rows look like the same. This implies that $P_{\mathrm{Im}(\widetilde{F})}(j,\cdot) = P_{\mathrm{Im}(\widetilde{F})}(\ell,\cdot)$, which proves the claim.

Secondly, we prove the other direction. Assume that $P_{\mathrm{Im}(\widetilde{F})}(j,\cdot) = P_{\mathrm{Im}(\widetilde{F})}(\ell,\cdot)$ with $0 \leq j \neq \ell \leq k-1$, and consider the vector $\chi \in \{-1,0,1\}^k$ which is 1 in the jth entry, -1 in the ℓth entry, and zero otherwise. It is easy to check that $P_{\mathrm{Im}(\widetilde{F})}\chi = \mathbf{0}$. This means that $\chi \in \ker(\widetilde{F})$ and $\widetilde{F}\chi = 0$, which implies that the columns of $\widetilde{F}$ indexed by j and ℓ, as well as the corresponding rows are equal (since $\widetilde{F}$ is Hermitian). Hence, the corresponding rows of F must be equal. □

We now devote our attention to prove Theorem 2. The following lemma shows that the Hermitian adjacency matrix of a random graph generated from the DSBM is concentrated around its expectation.

Lemma 6. *Let $G \sim \mathcal{G}(k,n,p,q,F)$ with $p = q$. Then, with high probability, we have that $\|A - \mathbb{E}A\| \leq 10\sqrt{pkn\log n}$.*

Proof. Let $M^{uv} \in \mathbb{C}^{N\times N}$ be the matrix with exactly two non-zero entries defined by $(M^{uv})_{u,v} = 1$, $(M^{uv})_{v,u} = -1$. By definition, $(M^{uv})^2$ has exactly two nonzero entries, i.e.,

$$(M^{uv})^2_{u,u} = (M^{uv})^2_{v,v} = -1. \tag{8}$$

Let X^{uv} be a random matrix defined by

$$X^{uv} = \begin{cases} i\cdot M^{uv} & \text{if } u \rightsquigarrow v \\ -i\cdot M^{uv} & \text{if } v \rightsquigarrow u \\ 0 & \text{otherwise.} \end{cases}$$

Observe that $\sum_{\{u,v\}} X^{uv} = A$, the adjacency matrix of G.

Let $u, v \in V$ be a pair of vertices such that $u \in C_j$ and $v \in C_\ell$. Then, we have

$$\begin{aligned}\mathbb{E}X^{uv} &= p\ (F_{j,\ell}M^{u,v}\cdot i + F_{\ell,j}M^{v,u}\cdot i)\\ &= p\ (F_{j,\ell}M^{u,v}\cdot i - (1-F_{j,\ell})M^{u,v}\cdot i)\\ &= p\ (2F_{j,\ell}-1)M^{u,v}\cdot i\\ &= p\ \widetilde{F}_{j,\ell}M^{uv},\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}(X^{uv} - \mathbb{E}X^{uv})^2 &= \mathbb{E}\,(X^{uv})^2 - (\mathbb{E}X^{uv})^2\\ &= -p\,(M^{uv})^2 - p^2\left(\widetilde{F}_{j,\ell}\right)^2(M^{uv})^2\\ &= \left(-p\left(\widetilde{F}_{j,\ell}\right)^2 - 1\right)p\,(M^{uv})^2.\end{aligned}$$

Moreover, $\left|-p\left(\widetilde{F}_{j,\ell}\right)^2 - 1\right| \leq 2$. Therefore, since the spectral norm of a matrix is upper bounded by the sum of the absolute values of the entries in each row, by equation (8) it holds that

$$\left\| \sum_{u,v \in V} \mathbb{E}(X^{uv} - \mathbb{E}X^{uv})^2 \right\| \leq 2pkn.$$

Setting $a = 10\sqrt{pkn \log n}$, $M = 1$, $\sigma^2 \leq pkn$ and $d = kn$, we apply Theorem 5 to obtain the statement. □

We now combine Lemma 6 and Theorem 4 to bound how far the matrix P computed by Algorithm 1 is to the projection on the image of $\mathbb{E}A$.

Lemma 7. *Let $G \sim \mathcal{G}(k, n, p, q, F)$ with $p = q$. Let Q be the projection on the image of $\mathbb{E}A$, i.e., $Q = P_{\mathrm{Im}(\mathbb{E}A)}$ and P as in Algorithm 1. Moreover, set the parameter ϵ of Algorithm 1 to $\epsilon = 20\sqrt{pkn \log n}$ and assume (4) holds. Then, it holds with high probability that*

$$\|P - Q\| = O\left(\frac{\sqrt{k \log n}}{\widetilde{\rho}\sqrt{pn}}\right).$$

Proof. Let $(\lambda_1, g_1), \ldots, (\lambda_\ell, g_\ell)$ be the pairs of the eigenvalues and eigenvectors computed by Algorithm 1. Then, by Lemma 6 it holds for any $1 \leq j \leq \ell$ that $|\lambda_j| \geq \widetilde{\rho}pn - \epsilon$. Notice that the other eigenvalues of A have absolute value less than ϵ. Therefore, based on assumption (4), and the relationship between the eigenvalues of $\widetilde{F}$ and $\mathbb{E}A$, we apply Theorem 4 and obtain

$$\|P - Q\| \leq \frac{\|A - \mathbb{E}A\|}{\widetilde{\rho}pn - 2\epsilon} = O\left(\frac{\sqrt{k \log n}}{\widetilde{\rho}\sqrt{pn}}\right). \qquad \square$$

We are now ready to prove the main theorem, which gives an upper bound on the number of vertices misclassified by Algorithm 1. More precisely, given a graph $G = (V, E)$ with clusters $C_0, \ldots, C_{k-1} \subset V$ and a partition $A_0, \ldots, A_{k-1}$ of V, the number of misclassified vertices is defined as

$$\mathcal{M} = \min_{\sigma \in S_k} \sum_{j=0}^{k-1} \left(|A_{\sigma(j)} \setminus C_j| + |C_j \setminus A_{\sigma(j)}|\right),$$

where S_k is the symmetric group on $[k]$. We also assume that the k-means algorithm used in Algorithm 1 achieves a constant approximation ratio (e.g., Kumar et al. (2004)). Now we are ready to prove the main result of the submission.

Proof of Theorem 2. Let $Q = P_{\mathrm{Im}(\mathbb{E}A)}$ and P as in Algorithm 1. Observe that Q is a block matrix with the following properties: rows corresponding to vertices belonging to the same cluster are equal, while the distance between rows corresponding to different clusters is at least θ. For any cluster C_j, let c_j be the row of Q corresponding to any vertex in C_j (they are all equal). Let $\overline{c_j}$ be the average of the rows of P corresponding to C_j. By Lemma 7 we know that $\|c_j - \overline{c_j}\| = O\left(\frac{\sqrt{k \log n}}{\widetilde{\rho}\sqrt{pn}}\right)$, which implies, for any $\ell \neq j$,

$$\|\overline{c_j} - \overline{c_\ell}\| \geq \theta - \frac{20\sqrt{k \log n}}{\widetilde{\rho}\sqrt{pn}} = \theta/2, \tag{9}$$

where the second inequality follows from assumption (4). Moreover, the optimal k-means cost is at most

$$\sum_{j=0}^{k-1} \sum_{u \in C_j} \|P(u, \cdot) - c_j\|^2 \leq \mathrm{tr}(P - Q)^2 \leq \|P - Q\|^2 \cdot kn = O\left(\frac{k^2 \log n}{\widetilde{\rho}^2 p}\right) \tag{10}$$

where the last equality follows from Lemma 7.

Let $c_0^\star, \ldots, c_{k-1}^\star$ be the optimal centroids of a k-means clustering on the rows of P. For any $\ell \neq j$, we claim that $\|c_j^\star - c_\ell^\star\| \geq \theta/4$. Assume this isn't true. By equation (9), then, there must exist a $\overline{c_\ell}$ which is at least $\theta/4$ far from any point $c_j^\star$. We now show this implies that the optimal k-means cost is large, contradicting equation (10). Let $c^\star(u)$ be the centroid $c_j^\star$ which is closest to $P(u, \cdot)$. Then, by the triangle inequality and the trivial inequality $(x - y)^2 \geq x^2/2 - y^2$, the optimal cost is lower bounded by

$$\begin{aligned}
&\sum_{u \in C_\ell} \|P(u, \cdot) - c^\star(u)\|^2 \\
&\geq \sum_{u \in C_\ell} \left(\|\overline{c_\ell} - c^\star(u)\| - \|P(u, \cdot) - \overline{c_\ell}\|\right)^2 \\
&\geq \sum_{u \in C_\ell} \left(\frac{1}{2}\|\overline{c_\ell} - c^\star(u)\|^2 - \|P(u, \cdot) - \overline{c_\ell}\|^2\right) \\
&\geq \frac{n\theta^2}{32} - O\left(\frac{k^2 \log n}{\widetilde{\rho}^2 p}\right),
\end{aligned}$$

which, by assumption (4), contradicts the fact that the optimal k-means cost is upper bounded by equation (10). Therefore, it holds that $\|c_j^\star - c_\ell^\star\| \geq \theta/4$ for any $\ell \neq j$. Hence, every time we misclassify a vertex we pay a cost of $\Omega(\theta^2)$. Because of this, any constant factor approximation algorithm for k-means will misclassify at most $O\left(\frac{k^2 \log n}{\widetilde{\rho}^2 \theta^2 p}\right)$ vertices. □

Proof of Corollary 3. We start investigating the matrix $\widetilde{F} = (2F - \mathbf{1}_{k \times k}) \cdot i$, which, in cyclic block models, can be rewritten as follows: $\widetilde{F}_{j,\ell} = (1 - 2\eta) \cdot i$ if $j \equiv \ell - 1 \mod k$, $\widetilde{F}_{j,\ell} = -(1 - 2\eta) \cdot i$ if $j \equiv \ell + 1 \mod k$, and

$\widetilde{F}_{j,\ell} = 0$ otherwise. Therefore, $\widetilde{F}$ is a circulant matrix. From the theory of circulant matrices, we can deduce that $\widetilde{F}$ has a set of k orthonormal eigenvectors $\widetilde{f}_0, \ldots, \widetilde{f}_{k-1}$, such that, for any $0 \leq j, \ell \leq k-1$, $\widetilde{f}_j(\ell) = \omega_k^{j\ell} k^{-1/2}$, where ω_k is the k-th root of unity. Let $\rho_0, \ldots, \rho_{k-1}$ be the eigenvalues of $\widetilde{F}$ ordered so that ρ_j is the eigenvalue corresponding to $\widetilde{f}_j$. It holds that

$$\rho_j = (1-2\eta)\left(\omega_k^j - \overline{\omega_k}^j\right) \cdot i = -2\sin(2\pi j/k)(1-2\eta), \tag{11}$$

where the second equality holds because the difference between a complex number c and its conjugate is equal to twice the imaginary part of c.

From this we can easily obtain a bound on the spectral gap $\widetilde{\rho}$:

$$\widetilde{\rho} = \min_{j \in [k] \setminus \{0, k/2\}} 2(1-2\eta) \cdot |\sin(2\pi j/k)| = \Theta\left(\frac{1-2\eta}{k}\right).$$

From equation (11) we know the kernel of $\widetilde{F}$ is spanned by $\widetilde{f}_0$ and, if k is even, by $\widetilde{f}_{k/2}$. In both cases, however, $\widetilde{F}$ has a $\Omega(1)$-distinguishing image. Therefore, the assumption of equation (4) holds whenever $p = \omega\left(\frac{k^3 \log n}{(1-2\eta)^2 n}\right)$. We can apply Theorem 2 to deduce that the number of misclassified vertices is $O\left(\frac{k^4 \log n}{(1-2\eta)^2 p}\right)$ with high probability. □

B Additional experimental results

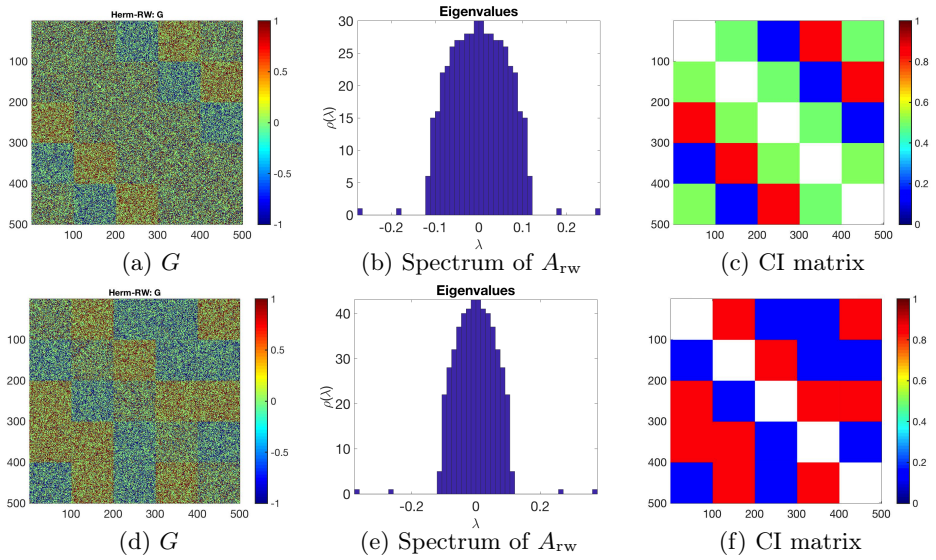
This section presents more detailed comparison on the performance of our algorithm with other spectral clustering algorithms for digraphs on both synthetic and real-world data sets. All of our experiments are performed in Matlab R2017b, on a MacBook Pro, with 2.8 GHz Intel Core i7 and 16 GB of memory. The spectral clustering algorithms are implemented using the Matlab function `eigs` to compute eigenvectors, and the k-means++ algorithm Arthur and Vassilvitskii (2007) to perform k-means clustering.

More detailed experimental results for the DSBM. In Figure 9 we consider two instances of graphs generated from the DSBM with $k = 5$ clusters, where each cluster is of size $n = 100$, $p = 50\%$, and noise parameter $\eta = 0.15$. The figures at the top concern a cyclic block model, while the figures at the bottom concern a randomly oriented complete meta-graph. We report the heatmap of the Hermitian adjacency matrices, the spectrum of A_{rw} leveraged by HERM-RW, as well as the final recovered cluster-structure with colours representing the CI score. From Figures 9 (a) and (c), respectively (d) and (f), we can clearly see the cyclic, respectively complete, pattern between the clusters. Moreover, Figures 9 (b) and (e) show that the bulk of the eigenvalues of A_{rw} is concentrated around 0, with exactly 4 outliers with larger absolute value: 4 corresponds to the rank of the matrix $\tilde{F}$ of the corresponding DSBM. Also notice that the eigenvalues outside these outliers are more concentrated in the case of the block model with complete meta-graph. This is not a surprise since in the latter we have a noise level of 0.15 between any pair of clusters, while in the cyclic block model we have noise level of 0.15 between k pairs, and noise level 0.5 (corresponding to completely random orientations of the edges) between the remaining pairs.

Figure 10 shows the recovery rate of spectral clustering algorithms for DSBM with a randomly oriented complete meta-graph with $k = 50$ clusters, each of size $n = 100$. In this regime with a very large number of clusters, our proposed methods perform drastically better than competing approaches. For edge density $p = 1\%$ only our approaches are able to achieve a meaningful ARI value, at least for low level of noise η (recall that when $\eta = 0$ noise due to intra-cluster edges is still present). When $p = 2\%$, other methods perform reasonably well up to a noise level of $\eta = 0.1$. Our method, instead, is able to achieve very good accuracy up to $\eta = 0.15$, and non-trivial accuracy up to $\eta = 0.2$.

Figure 11 (a) and (b) is a comparison on the DSBM model with $N = 10,000$ and $k = 20$ clusters, for both a complete meta-graph and a cyclic block model. This is the largest graph we have experimented with, and it shows that our Hermitian-based algorithms vastly outperform the competing methods, especially in the case of the cyclic block model where there are less pairwise interactions between the clusters. Note that in Figure 11 (b) we left out BI-SYM and DD-SYM from the comparison, due to their computational cost. It is also easy to see that our algorithms not only significantly outperform all the other tested methods, but also run significantly faster than BI-SYM and DD-SYM which involve matrix multiplication operations. For instance, Figure 11 (c) compares the runtime of all algorithms on graphs randomly generated from the DSBM for $N = 10,000$, $k = 20, p = 0.4\%$, and different η values, and this quantitative comparison holds for different choices of parameters in general.

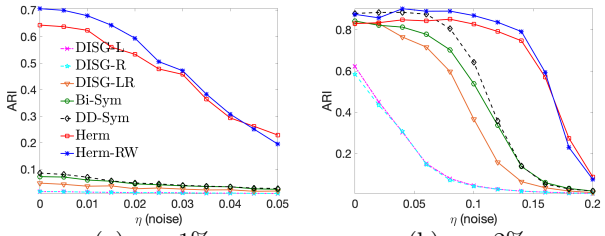
US-MIGRATION: We present further numerical results for the main data set of our submission, omitted from the main text due to page limit. We compare the performance of all the variants of the algorithms listed in the submission, and Figure 16 is a visualisation of the top three largest pairs in terms of the size-normalised cut imbalance ratio for $k = 10$. The NAIVE method performs standard spectral clustering



(a) G (b) Spectrum of A_{rw} (c) CI matrix

(d) G (e) Spectrum of A_{rw} (f) CI matrix

Figure 9: Recovery of an instance of the DSBM model, $N = 500$, $p = 50\%$, $\eta = 0.15$ and $k = 5$ clusters, for a cyclic block model (top) and a randomly oriented complete meta-graph (bottom).



(a) $p = 1\%$ (b) $p = 2\%$

Figure 10: Recovery rates for the complete meta-graph in the DSBM with $k = 50$ and $n = 100$.

on the symmetrised matrix $G = M + M^\intercal$. In addition to our two proposed Hermitian-based approaches, HERM and HERM-RW, considered thus far throughout the paper, we also evaluate the performance of a third method which we denote HERM-SYM. In a similar spirit, HERM-SYM considers the top k largest eigenvalues of the following matrix

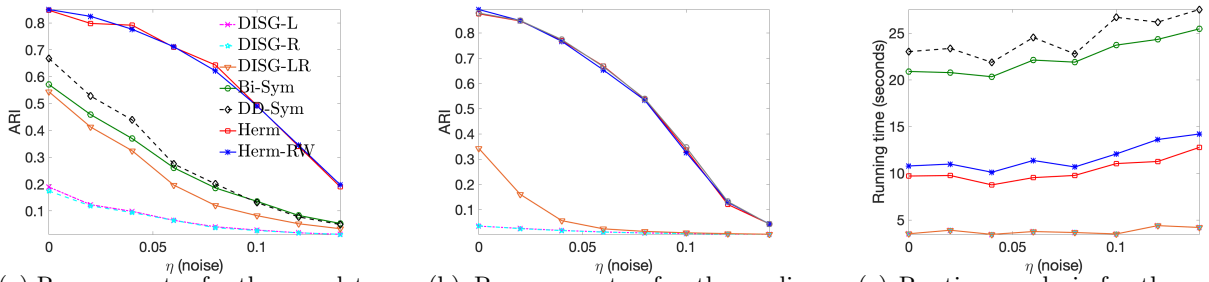
$$A_{\text{sym}} = D^{-1/2} A D^{-1/2}, \tag{12}$$

and recovers the clusters via k-means in this spectral embedding. The normalisation in A_{sym} is particularly suitable for the skewed degree distributions often encountered in real data. For each highlighted pair (shown in red and blue in the US map colorings, while yellow denotes the remaining nodes), we also show the numerical scores achieved by the respective pair in terms of the three performance metrics (CI, CI^{size}, and CI^{vol}). Here are our conclusions:

- In terms of the CI score, the top three methods are HERM-SYM (0.26), HERM-RW (0.19), followed by DD-SYM (0.16) and DISG-LR (0.16). We remind the reader that an imbalance score of CI = 0.26 as achieved by HERM-SYM essentially denotes that $26\% + 50\% = 76\%$ of the total weight of the edges between a pair of clusters is oriented in one direction, and the remaining 24% in the other direction.
- In terms of the CI^{size} score, the top three methods are HERM-RW (105), HERM-SYM (63), HERM (30), and BI-SYM (30).
- Finally, in terms of the CI^{vol} score, the top three methods are BI-SYM (20,062), HERM-SYM(19,570) and HERM-RW (16,268).

US-MIGRATION-II: Due to a small number of very large entries in the initial migration matrix M, many of the methods we compare against are not able to produce meaningful results.To this end, we pre-process the migration matrix M and cap all entries at 10,000, which corresponds to the 99.9% percentile. As shown in Figure 1a, a simple symmetrisation of the input matrix $M \mapsto M + M^\intercal$, followed by standard spectral clustering of undirected graphs von Luxburg (2007), will reveal clusters that align very well with the state boundaries Cucuringu et al. (2013). The top, respectively bottom, plots in Figure 12 show the CI, respectively CI^{vol}, score for the top pairs. For the CI score, HERM and HERM-RW are among the top performing methods along with BI-SYM, while for CI^{vol}, HERM-RW is the best performing method accross all values of $k = \{2, 10, 20, 40\}$.

UK-MIGRATION: Another data set we considered is the UK-MIGRATION network with $N = 354$, which captures in a directed graph the number of people



(a) Recovery rates for the complete meta-graph $p = 0.4\%$.

(b) Recovery rates for the cyclic block model $p = 3\%$.

(c) Runtime analysis for the complete meta-graph $p = 0.4\%$.

Figure 11: Recovery rates and running time for a complete meta-graph and a cyclic block model in a DSBM with $k = 20$ clusters, $N = 10,000$ at various levels of noise. Averaged over 10 runs.

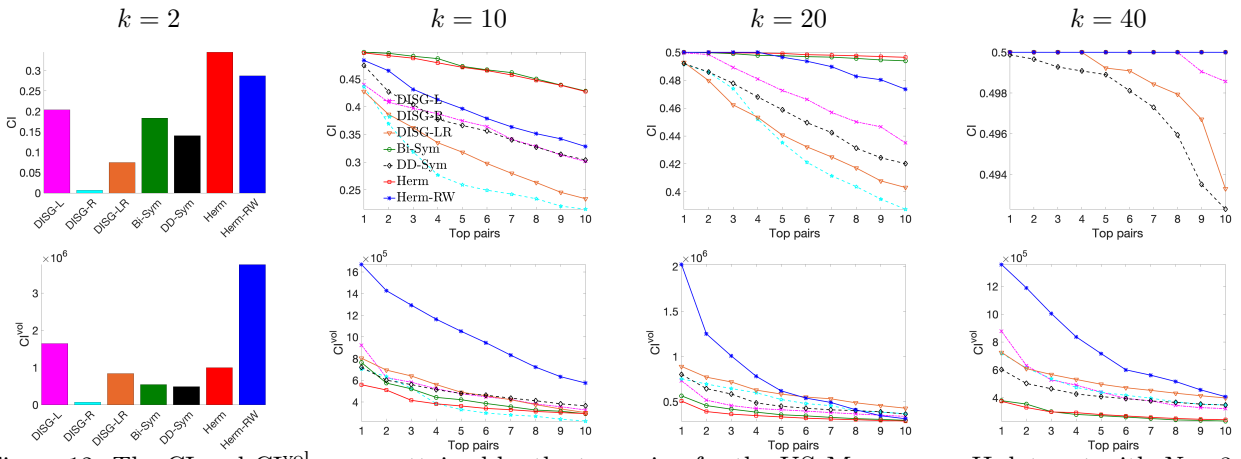


Figure 12: The CI and CI^{vol} scores attained by the top pairs, for the US-Migration-II data set with $N = 3,107$ and $k = \{2, 10, 20, 40\}$ clusters (averaged over 20 runs).

who migrated between local authority Districts in the UK, aggregated over the interval 2012-2017 for National Statistics (2018). Figure 14 shows the CI and CI^{vol} scores for the top pairs, for varying values of k. For $k = 2$, DISG-L and DISG-LR are the best performing methods. For $k = \{3, 5, 8\}$, a number of methods perform comparably well, with Herm-RW being the best performer in terms of the CI^{vol} scores. Finally, Figure 13 shows the clustering recovered by Herm-RW with $k = 8$ clusters, highlighting the Greater London metropolitan area, as well as counties such as Essex, Surrey, West Sussex and Oxfordshire.

c-Elegans: The last data set we studied is the c-Elegans neural connectome network, which encodes connection between the neurons in a directed network White et al. (1986). This popular data set Leskovec and Krevl (2014), also considered in Rohe et al. (2016), highlights significant dissimilarities between the sending and receiving patterns in the neural network. Figure 15 compares the CI and CI^{vol} scores corresponding to the top pairs, across all algorithms and for various values of k. For $k = 2$, respectively $k = 3$, Herm-RW is the best performer, followed closely by Bi-Sym, resp. Herm, while the rest of the algorithms perform significantly worse. For higher $k = \{5, 10\}$, results are mixed, with a number of methods performing similarly well.

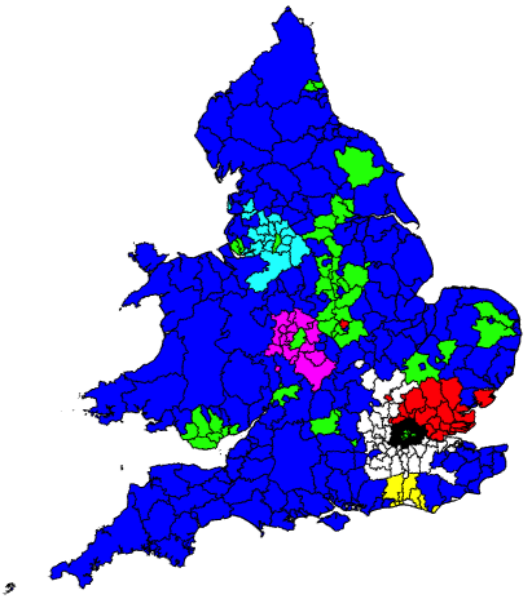


Figure 13: The clustering structure recovered by HERM-RW with $k = 8$ clusters, for the UK-MIGRATION data set.

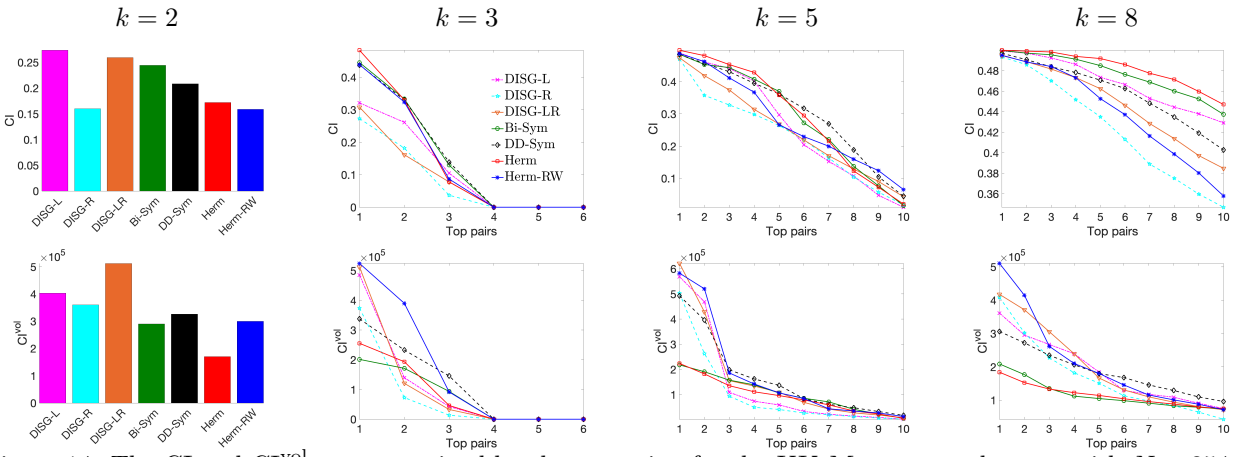


Figure 14: The CI and CI^{vol} scores attained by the top pairs, for the UK-MIGRATION data set with $N = 354$ and $k = \{2, 3, 5, 8\}$ clusters (averaged over 20 runs).

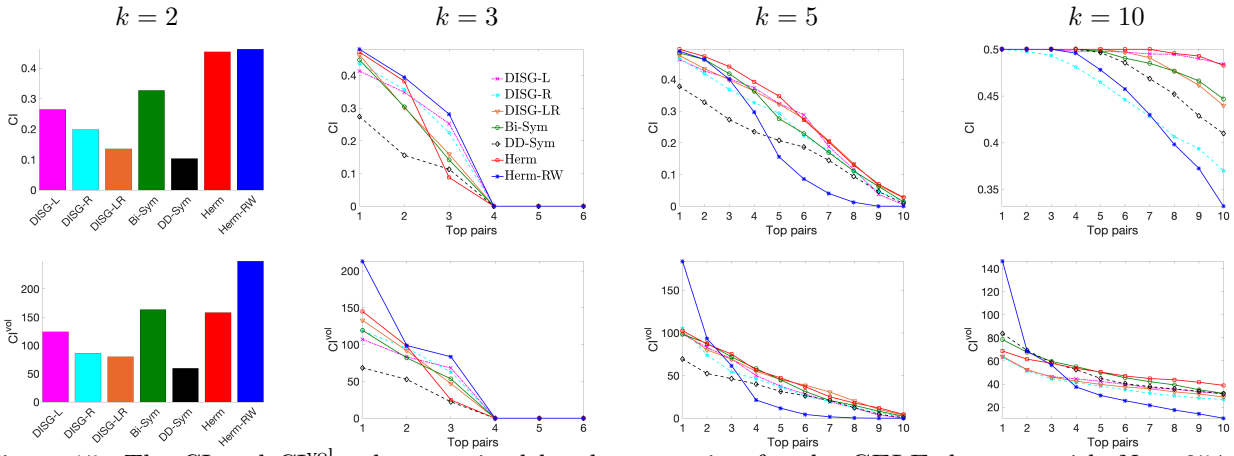


Figure 15: The CI and CI^{vol} values attained by the top pairs, for the CELE data set with $N = 354$ and $k = \{2, 3, 5, 10\}$ clusters (averaged over 20 runs).

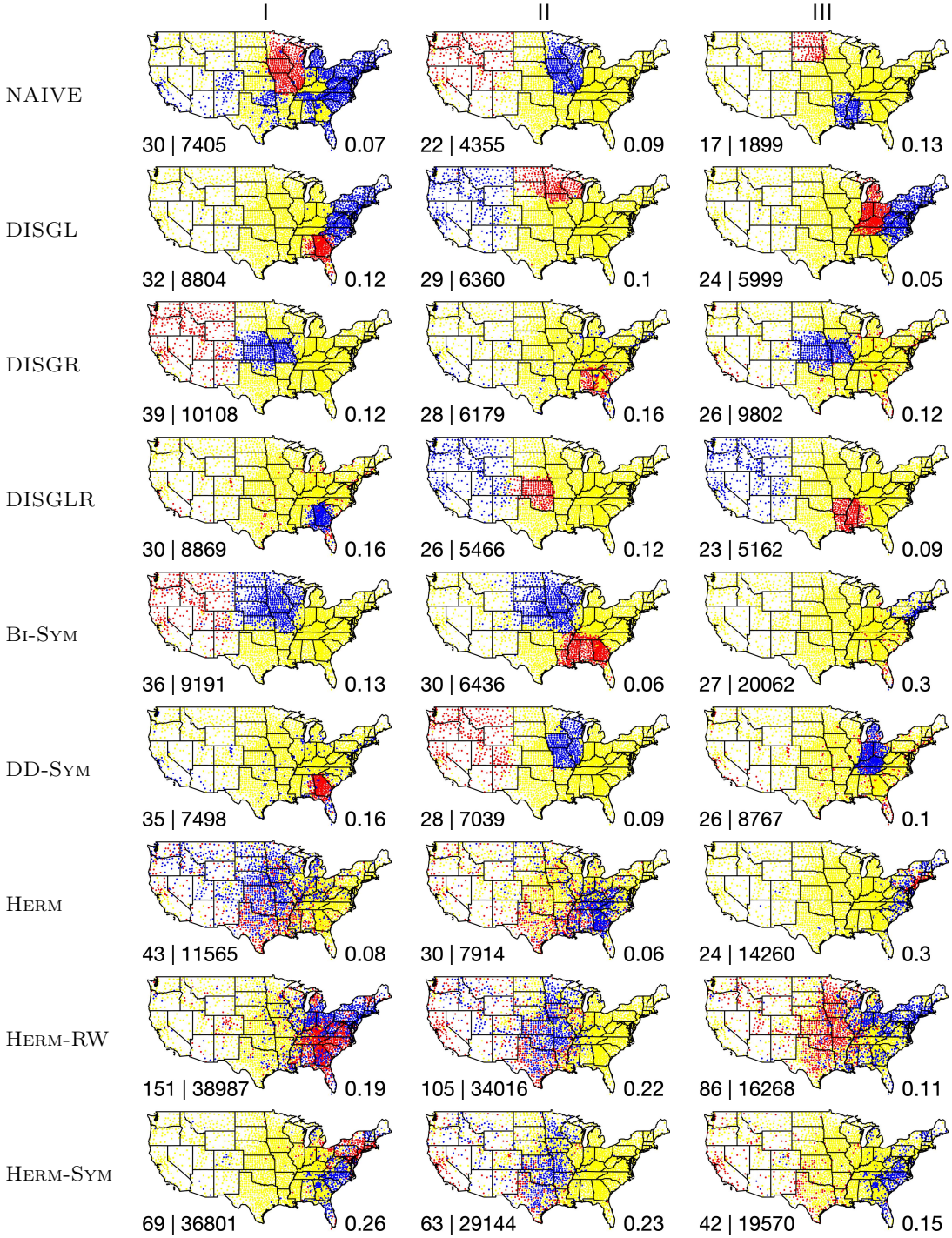


Figure 16: The top three largest size-normalised cut imbalance pairs for the US-Migration data with $k = 10$ clusters, for all the methods considered. Red denotes the source cluster, and blue denotes the destination cluster. For each plot, the bottom left text contains the numerical values (rounded to nearest integer) of the two normalised $\mathrm{CI}^{\mathrm{size}}$ and $\mathrm{CI}^{\mathrm{vol}}$ pairwise cut imbalance values, and the bottom right text contains the CI cut imbalance ratio in $[0, 0.5]$.