Hermitian matrices for clustering directed graphs: insights and applications

Mihai Cucuringu[*†] Huan Li[‡] He Sun[§] Luca Zanetti[¶]

Abstract

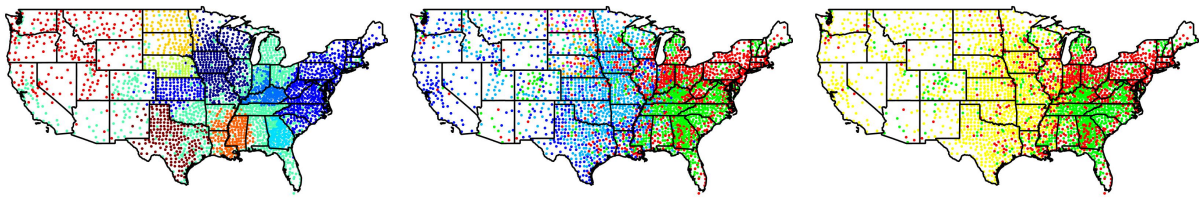
Graph clustering is a basic technique in data mining, and has widespread applications in different domains. While spectral techniques have been successfully applied for clustering undirected graphs, the performance of spectral clustering algorithms for directed graphs (digraphs) is not in general satisfactory, as these algorithms usually require symmetrising the matrix representing the digraph, and typical objective functions for undirected graph clustering (e.g., graph conductance and the normalised cut value) do not capture cluster-structures in which the information given by the direction of the edges is crucial.

To overcome these downsides faced by most existing spectral algorithms, we study complex-valued Hermitian matrix representations of digraphs and present a clustering algorithm based on such Hermitian matrix representations. Through extensive experimental results and a theoretical analysis on a directed version of the stochastic block model, we show that our algorithm is able to uncover cluster-structures which are not simply based on edge-density, but on imbalances in the direction of the edges between the clusters. We highlight the significance of our work 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 (a property independent of the direction of the edges in the underlying graph), our approach is able to cluster together counties with a similar socio-economical profile even when they are geographically distant, and illustrate how people tend to move from rural to more urbanised areas.

1 Introduction

Graph 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

[*]Department of Statistics and Mathematical Institute, University of Oxford, UK
[†]The Alan Turing Institute, London, UK
[‡]School of Computer Science, Fudan University, China
[§]School of Informatics, University of Edinburgh, UK
[¶]Department of Computer Science and Technology, University of Cambridge, UK



(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 proposed procedure. The two red and green clusters shown in (c) are such that 68% of the total weight of the edges between the two flows from the green cluster to the red one.

analysis to social sciences. When the underlying graph to cluster is undirected, the objective is to partition the vertices of the graph into subsets (i.e., 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 value [14, 23]. For example, the widely popular spectral clustering algorithm[25, 18], which uses the top eigenvectors of the adjacency or Laplacian matrix of a graph as input features for k-means, essentially 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 look at the graph that represents the number of people moving between different counties in the (mainland) United States during 1995-2000 [19, 2]. If one tries to symmetrise its (asymmetric) adjacency matrix M in a naive way by computing the symmetric matrix $M + M^\intercal$, migration flows between counties in different states will be lost in the process. Indeed, when considering the clustering computed by spectral clustering with input the symmetrised adjacency matrix $M + M^\intercal$ of this migration data set, the visualisation in Figure 1a shows that clusters align particularly well with the political and administrative boundaries of the US states, as observed previously in [5]. This is, maybe somehow counterintuitively, an unsatisfactory outcome: it is quite obvious that people are more likely to move to neighbouring counties than to far away ones, and it does not provide us with much information about higher-order migration patterns across the country.

Motivated by this example, we study spectral clustering techniques for directed graphs (digraphs). In contrast with most previous approaches that symmetrise the adjacency matrix of a digraph, we work with a complex-valued Hermitian adjacency matrix considered in [11, 24] and defined as follows: for any N-vertex digraph G, the Hermitian adjacency matrix $A_G \in \mathbb{C}^{N\times N}$ of G is the matrix where $(A_G)_{u,v} = \overline{(A_G)_{v,u}} = i$ if there is a directed edge $u \rightsquigarrow v$, and $(A_G)_{u,v} = 0$ otherwise, where i is the imaginary unity. Because of the use of i and its conjugate $\bar{i}$ in expressing a directed edge, all of the A_G's eigenvalues are real-valued. We show that, when the direction of the edges impart a cluster structure on G, this structure is approximately encoded in the eigenvectors associated with the top eigenvalues of A_G. To briefly demonstrate the usefulness of our Hermitian adjacency matrix, Figure 1b visualises the outcome of spectral clustering when a normalised version of the complex-valued Hermitian adjacency matrix 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, from this clustering we can see several interesting migration patterns emerging, especially when considering all pairs of resulting clusters, with an emphasis on those that exhibit the largest "imbalance" in the direction of the edges between the two clusters. 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, Virginia and 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 of a cluster, our algorithm uncovers the "higher-order" structure between the clusters. We highlight that of 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 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 in terms of imbalanced cuts/flows between the clusters. In contrast with the classical SBM, in our model additional parameters 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 theoretical and experimental performances of our algorithm.

- In Section 3 we present a spectral clustering algorithm for digraphs, based on the eigenvectors corresponding to the largest eigenvalues of A_G or various normalisation of it. We also provide an intuition about the different behaviour of our algorithm compared to other previous spectral clustering approaches for digraphs.
- 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 proposed algorithm, in Section 5 we empirically demonstrate its practicality, and compare its performance against several competing approaches on synthetic and real-world data.

Notation For any (unweighted) directed graph G with N vertices, the Hermitian adjacency matrix of G is the matrix $A_G \in \mathbb{C}^{N\times N}$, where $(A_G)_{u,v} = \overline{(A_G)_{u,v}} = i$ if there is a directed edge from u to v, expressed by $u \rightsquigarrow v$, and $(A_G)_{u,v} = 0$ otherwise. If G is, instead, a weighted digraph with weight $w_{u,v}$ on the edge $u \rightsquigarrow v$, we define $(A_G)_{u,v} = (w_{u,v} - w_{v,u})i$. In any case, A_G is a Hermitian matrix by definition and 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 randomly chosen from the directed stochastic block model (DSBM) defined by parameters $k, \{n_j\}_{j=0}^{k-1}, p, q$, and matrix $F \in [0,1]^{k\times k}$, where $k \geq 2$ represents the number of clusters, $\{n_j\}_{j=0}^{k-1}$ the number of vertices in each cluster, $p \in (0,1]$ the probability there exists an edge between two vertices within the same cluster, $q \in [0,1]$ the probability there exists an edge between two vertices belonging to two different clusters, while $F \in [0,1]^{k\times k}$ controls the orientations of the edges among clusters and satisfies the following rule: $F_{\ell,j} + F_{j,\ell} = 1$ for any $0 \leq \ell, j \leq k-1$, which implies that $F_{\ell,\ell} = 1/2$ for any $0 \leq \ell \leq k-1$. The set $\mathcal{G}\left(k, \{n_j\}_{j=0}^{k-1}, p, q, F\right)$ 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 $N = \sum_{j=0}^{k-1} n_j$. These vertices belong to k clusters $C_0, \ldots, C_{k-1}$, where $|C_j| = n_j$ 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, if they belong to different clusters, 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 matrix F, i.e., the direction is set to be $u \rightsquigarrow v$ with probability $F_{\ell,j}$, and the direction is set to be $v \rightsquigarrow u$ with probability $F_{j,\ell} = 1 - F_{\ell,j}$. Notice that, by the definition above, the directions of edges inside a cluster are randomly chosen with the same probability. The matrix F can be understood as the adjacency matrix of a weighted directed graph which represents the *meta-graph* describing the relations between the clusters. We look at the following example to explain the roles of these parameters and matrix F.

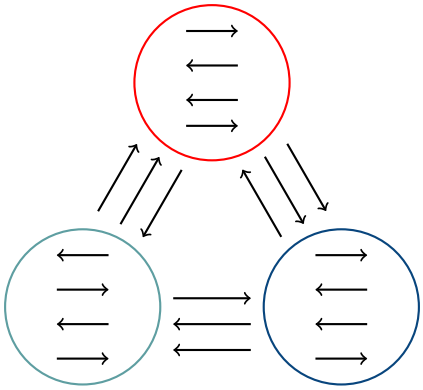


Figure 2

Example. Let $k = 3$, $n_0 = n_1 = n_2$, $p = q$, and

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

In this case, G consists of 3 clusters C_0, C_1 and C_2 of the same size, and any pair of vertices is connected by an edge with the same probability p. The directions of edges inside a cluster are chosen uniformly at random, but directions of the edges crossing different clusters are chosen non-uniformly and are defined by F. In particular, in expectation $2/3$ of the edges between $u \in S_j$ and $v \in S_{j+1 \bmod 3}$ are set to be $u \rightsquigarrow v$, and the remaining $1/3$ of the ones are 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 in this model, since in expectation all the vertices in G has the same in-degrees 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 stochastic block model [12] into the setting of directed graphs. As a special case of the DSBM, when $F_{\ell,j} = 1/2$ and $n_j = n_\ell$ for any $0 \leq j, \ell \leq k-1$, the edge directions play no roles in characterising a cluster structure, and the clusters are entirely determined by p and q, which is exactly the case for the SBM. On the other hand, the DSBM model captures the setting where $p = q$ and the cluster structure is determined by the distribution of edge directions. While the cluster structure is indistinguishable by ignoring the edge directions, we will show that Hermitian adjacency matrices can be used to recover the cluster structure of graphs generated from the DSBM.

Finally, we would like to mention that the stochastic block model for digraphs (DSBM) studied in our paper is a special case of the recently introduced co-SBM [21], which also includes bipartite structures for example. We think, however, that what is lost by our model in generality is gained in clarity and simplicity.

3 Algorithm

In this section we describe a spectral clustering algorithm for graphs generated from the DSBM. Given a graph $G = (V, E)$ generated from the DSBM $\mathcal{G}\left(k, \{n_j\}_{j=0}^{k-1}, p, q, F\right)$, 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 a k-means algorithm with the rows of P as input features. Finally, the algorithm partitions the vertex set of G based on the output of the k-means algorithm, see Algorithm 1 for formal description.

We remark that the number ℓ of eigenvectors used by the algorithm for k-means clustering depends on the parameters of the model, and in particular on the rank of the matrix F which

Algorithm 1: Spectral clustering for a digraph

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

1. Compute all the eigenvalues/eigenvectors pairs $\{(\lambda_1, g_1), (\lambda_2, g_2), \ldots, (\lambda_\ell, g_\ell)\}$ of A satisfying $|\lambda_i| > \epsilon$.
2. $P \leftarrow \sum_{j=1}^{\ell} g_j g_j^*$
3. Apply a k-means algorithm with input the rows of P.
4. Return a partition of V corresponding to the output of k-means.

defines the flow structure 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 $p = q$ and $n_0 = \ldots = n_{k-1} = n$, 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 $p \gg 1/n$. In this way, as it will become clear from our following analysis, ℓ will be automatically set as the rank of F, and this value is independent of the specific values of the entries in F. We also notice that including all the eigenvectors corresponding to the same eigenvalue in absolute value ensures that P is actually a *real* matrix. This follows from the fact that A is not only Hermitian, but also skew-symmetric.

3.1 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 $M \in \mathbb{R}^{N \times N}$ of a digraph G with N vertices 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 [21, 17, 22] mainly use the eigenvectors of $M^\intercal M$, $MM^\intercal$, or $M^\intercal M + MM^\intercal$ (or a normalised/regularised version of these matrices). To understand how our algorithm differs from previous approaches, let us briefly look at these matrices. In particular, for any $u, v \in V$ these matrices' corresponding entry indexed by u and v can be written as

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

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

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

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 square A^2 of our Hermitian adjacency matrix A, since these two matrices share the same eigenvectors and the A^2 is easier to analyse. By definition

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

In other words, 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, the matrix A implicitly assigns a positive weight between a pair of

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

vertices who have more common parents and offspring than "mismatched" relations with third vertices, and a negative weight otherwise. This peculiar behaviour is, we believe, at the heart of the better performances of our algorithm on some real world datasets compared to the state of the art. Moreover, it is interesting to notice that A can implicitly keep track of both common parents and offspring without the need to perform an expensive matrix multiplication as in the case of the matrix $M^\intercal M + MM^\intercal$.

3.2 Normalisation of A_G

When dealing with real-world data sets, proper normalisations of the graph adjacency matrices are usually required for most spectral clustering algorithms. Following this, we introduce two normalised Hermitian adjacency matrices. Given the above introduced Hermitian matrix A, denote by D the diagonal matrix with $D_{jj} = \sum_{\ell=1}^{N} |A_{j\ell}|$. Next, consider the normalised matrix

$$A_{\text{rw}} = D^{-1}A, \tag{4}$$

which is similar to the Hermitian matrix

$$A_{\text{sym}} = D^{-1/2}AD^{-1/2}, \tag{5}$$

via $A_{\text{rw}} = D^{-1/2}A_{\text{sym}}D^{1/2}$, and thus A_{rw} also has N real eigenvalues. The operator (4) on Hermitian matrices was considered in [6] in the context of angular synchronisation and the graph realisation problem, and in [24] who introduced Vector Diffusion Maps for nonlinear dimensionality reduction and explored the interplay with the connection-Laplacian operator for vector fields over manifolds. The normalisation in A_{sym} is suitable for the skewed degree distributions often encountered in real data. Throughout our experiments, we use the top k largest eigenvalues of A_{rw} and A_{sym}, and recover the clusters via k-means in this spectral embedding. We denote the resulting two algorithms (the analogues of our previously introduced Algorithm 1) by Herm-RW, respectively Herm-Sym.

We remark that another very related line of work, where the above Hermitian operators have been successfully used, comes from the ranking literature, where the net outcome of pairwise matches between players can be encoded in a directed graph with a skew-symmetric adjacency matrix. In particular, [4] 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} in (4) to recover a 1-D ordering of the players. Recently, [8] proposed a certain deformation of the combinatorial Laplacian, the *Dilation Laplacian*, which is shown to perform well for ranking in directed networks of pairwise comparisons.

4 Analysis

We now analyse the performance of Algorithm 1 on the DSBM. Let $G \sim \mathcal{G}\left(k, \{n_j\}_{j=0}^{k-1}, p, q, F\right)$ with Hermitian adjacency matrix A. For simplicity, we assume that all the clusters have equal size, i.e., $n_0 = n_1 = \cdots = n_{k-1} = n$, and the density inside the cluster equals the density between clusters, i.e., $p = q$. We remark the latter 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 instead is encoded only in the edge orientations.

In this section, we first analyse the performance of Algorithm 1 for graphs randomly generated from the DSBM, and the main theoretical result is summarised in Theorem 5. While our DSBM is defined in a quite general manner to characterise different practical flow scenarios, which is defined with respect to $F \in \mathbb{R}^{k\times k}$, for several important and interesting cases the statement of our main theorem can be greatly simplified. To demonstrate this, we prove in the second part of the section that, when the flow matrix F defines a cycle as illustrated in the Example from Section 2, the number of misclassified vertices with respect to the ground truth clustering can be bounded in a much cleaner way.

4.1 Analysis of the general case

We first study the expected adjacency matrix $\mathbb{E}A$. For any $u \in C_j$ and $v \in C_\ell$, we have by definition that

$$(\mathbb{E}A)_{u,v} = p\,(F_{j,\ell} - F_{\ell,j}) \cdot i = p\,(2F_{j,\ell} - 1) \cdot i.$$

These implies that $\mathbb{E}A$ is a Hermitian matrix 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

$$\widetilde{F} = (2F - \mathbf{1}_{k\times k}) \cdot i.$$

It is easy to verify 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 the corresponding eigenvector $f \in \mathbb{C}^{kn}$ where $f(u) = \widetilde{f}(j)$ for any $u \in C_j$.

Before the formal analysis, we first explain the intuition behind the design of our algorithm. Note that, if A is close to its expectation $\mathbb{E}A$, which is the case for most instances, then the top eigenspace of A will be close to the image of $\widetilde{F}$. Because of this, as well as the fact that vectors in the image of $\widetilde{F}$ always assign the same value to the vertices belonging to the same cluster, it suffices to ensure that the projection on the image of $\widetilde{F}$ is actually able to distinguish different clusters. Note that this is the cases when $\widetilde{F}$ is nonsingular. However, being a skew-symmetric matrix, $\widetilde{F}$ is singular whenever k is odd. Because of this, we introduce a weak constraint to ensure that the rows of the project on the image of $\widetilde{F}$ are not similar to each other, which is formalised by the following definition.

Definition 1. *Let $P_{\mathrm{Im}(\widetilde{F})}$ be the projection on the image of $\widetilde{F}$. For any $\theta \in [0,1]$, we say $\widetilde{F}$ has a θ-distinguishing image, if it holds for any $0 \le j \ne \ell \le k-1$ that*

$$\left\| P_{\mathrm{Im}(\widetilde{F})}(j,\cdot) - P_{\mathrm{Im}(\widetilde{F})}(\ell,\cdot) \right\| \ge \theta.$$

To help understand Definition 1, let us look at the following proposition, which shows that the image of $\widetilde{F}$ is a 0-distinguishing image if F has two identical rows. When $p = q$, this condition implies the model has two statistically undistinguishable clusters.

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

Our main analysis technique is based on matrix perturbation theory, which requires that the nonzero eigenvalues of $\widetilde{F}$ is far from 0 to ensure that the top eigenspace of $\mathbb{E}A$ is close to the image of $\widetilde{F}$. Because of this, we introduce the *spectral gap* $\widetilde{\rho}$ of $\widetilde{F}$ defined 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 matrix $\widetilde{F}$. We remark that in undirected stochastic block models a similar definition of spectral gap governs the performance of spectral clustering algorithms (see, e.g., [15, Corollary 3.2]).

Using the Davis-Kahan theorem (Theorem 6 in the Appendix) we will bound the distance between the image of $\mathbb{E}A$ and the top eigenspaces of A, which in turn depends on the spectral norm of $A - \mathbb{E}A$. We now apply this theorem to bound $\|A - \mathbb{E}A\|$.

Lemma 3. *Let $G \sim \mathcal{G}\left(k, \{n_j\}_{j=0}^{k-1}, p, q, F\right)$ with $n_j = n$ for all $0 \le j \le k-1$ and $p = q$. Then, with high probability, we have that $\|A - \mathbb{E}A\| \le 10\sqrt{pkn\log(kn)}$.*

We now combine Lemma 3 and the Davis-Kahan theorem to bound how far the matrix P computed by Algorithm 1 is to the projection on the image of $\mathbb{E}A$. For technical reasons, we assume that the spectral gap $\widetilde{\rho}$ of $\widetilde{F}$ satisfies

$$\widetilde{\rho} > 40\sqrt{\frac{k\log(kn)}{\theta^2 pn}}. \tag{6}$$

Note that for a family of graphs with k fixed and n growing, as long as p is not too small, this assumption is always satisfied. It also implies that, for most cluster-structure matrix F, p needs to be greater than $k\log(kn)/n$, which is comparable to the condition $p \geq \log(kn)/(pn)$ that is required for ensuring G being connected.

Lemma 4. *Let $G \sim \mathcal{G}\left(k, \{n_j\}_{j=0}^{k-1}, p, q, \Delta\right)$ with $n_j = n$ for all $0 \leq j \leq k-1$ and $p = q$. 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 $\epsilon = 10\sqrt{pkn\log(kn)}$. Then, it holds with high probability that*

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

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., [13]). As the main result, the performance of Algorithm 1 is summarised as follows:

Theorem 5 (Main Theorem)**.** *Let $G \sim \mathcal{G}\left(k, \{n_j\}_{j=0}^{k-1}, p, q, F\right)$ with $n_i = n$ for all $0 \leq i \leq k-1$ and $p = q$. Assume equation (6) holds and $\widetilde{F}$ has a θ-distinguishing image with $\theta > 0$. Then, the number of misclassified vertices by Algorithm 1 is $O\left(\frac{k^2\log(kn)}{\widetilde{\rho}^2\theta^2 p}\right)$ with high probability.*

Proof sketch. 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 c. 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 4 we know that $\|c_j - \overline{c_j}\| = O\left(\frac{\sqrt{k\log(kn)}}{\widetilde{\rho}\sqrt{pn}}\right)$, which implies, for any $\ell \neq j$, $\|\overline{c_j} - \overline{c_\ell}\| \geq \theta - \frac{20\sqrt{k\log(kn)}}{\widetilde{\rho}\sqrt{pn}} = \Omega(\theta)$, where the second inequality follows from assumption (6).

Therefore, every time we misclassify a vertex we pay a cost of $\Omega(\theta^2)$. Notice that the optimal k-means cost is at most

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

where the last equality follows from Lemma 4. Therefore, any constant factor approximation algorithm for k-means will misclassify at most $O\left(\frac{k^2\log(kn)}{\widetilde{\rho}^2\theta^2 p}\right)$ vertices. □

4.2 Analysis for cyclic block models

We now evaluate the theoretical guarantees given by Theorem 5 on a specific family of DSBM, which we call cyclic block models. We consider $G \sim \mathcal{G}\left(k, \{n_j\}_{j=0}^{k-1}, p, q, F\right)$ where, for any $0 \leq j \leq k-1$, $n_j = n$, $p = q$, and there exists $\eta \in [0, 1/2)$ such that F satisfies the following conditions: $F_{j,\ell} = 1 - \eta$ 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. Essentially, the relations between the clusters can be represented by a directed cycle where each edge has weight $1 - \eta$. We believe this family of the DSBM is particularly suited to test the performance of a clustering algorithm on directed graphs: first of all, since each

node has equal out- and in-degree in expectation, we cannot use information about the degree of a node to recover the clusters. Secondly, even when $\eta = 1$, which corresponds to the case where all the edges between two clusters C_j and $C_{j+1 \mod k}$ are oriented in the same direction, because the edges between most pairs of clusters are oriented randomly, recovery can be quite challenging.

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 $F_{j,\ell} = 0$ 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 $0 \le j, \ell \le k-1$, $\widetilde{f}_j(\ell) = \omega_k^{j\ell} k^{-1/2}$, 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

$$\rho_j = (1-2\eta)\left(\omega_k^j - \overline{\omega_k}^j\right)\cdot i = -2\sin(2\pi j/k)(1-2\eta). \tag{7}$$

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 (7) 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 equation (6) holds whenever $p = \omega\left(\frac{k^3\log(kn)}{(1-2\eta)^2 n}\right)$, and, applying Theorem 5 the number of misclassified vertices is $O\left(\frac{k^4\log(kn)}{(1-2\eta)^2 p}\right)$.

5 Experiments

We compare the performance of our algorithms with other existing clustering algorithms for digraphs on both synthetic and real-world data sets. Since for graphs generated from the DSBM we have a ground truth clustering available, we measure the recovery accuracy by the Adjusted Rand Index (ARI) [10], which is closely related and alleviates some of the issues of the popular Rand Index [20]. Both measures indicate how well the recovered partition matches the ground truth, with a value close to 1, resp. 0, indicating an almost perfect recovery, resp. an almost random assignment of the nodes into clusters. For the real-world data sets, due to the lack of a ground truth clustering, we will introduce an appropriately defined new objective function in order to measure the quality of the clustering, while taking the flow directions into account and aiming to uncover imbalanced structure flows in the partition.

5.1 Experimental setup

We compare our algorithm against three versions of the DI-SIM algorithm by Rohe et al. [21], and two algorithms (bibliometric symmetrisation and degree-discounted symmetrisation) by Satuluri et al. [22]. All these algorithms follow the standard framework of spectral clustering algorithms (such as Algorithm 1), but use different eigenvectors to construct the features vectors for k-means++. A brief description of each algorithm is given below, and we refer the reader to their respective references for additional details.

- DI-SIM (left) (which we denote by DISG-L): the top k eigenvectors of a regularised and normalised version of the matrix defined in equation (1) are used as input features for k-means.
- DI-SIM (right) (DISG-R): the top k eigenvectors of a regularised and normalised version of the matrix defined in equation (2) are used.

- DI-SIM (left+right) (DISG-LR): the top k eigenvectors of a regularised and normalised version of both matrix (1) and (2) are used.
- BI-SYM : the top k eigenvectors of the matrix defined in equation (3) are used.
- DD-SYM: the top k eigenvectors of a normalised version of the matrix defined in equation (3) are used.

We also consider three different versions of our Hermitian adjacency matrix, based on the various normalisations discussed in Section 3:

- HERM: the top eigenvectors of the standard Hermitian adjacency matrix are used.
- HERM-RW: based on the top k eigenvectors of A_{rw} (4)
- HERM-SYM: based on the top k eigenvectors of A_{sym} (5)

We remark that Algorithm 1 is defined according to the first of these matrices. This is because, as our experimental results show, no degree-normalisation is needed in DSBMs since all the vertices have the same expected degree. However, in real-world data sets, the degree distribution is typically very skewed with large outlier degrees, and indeed our experimental results illustrate that HERM-RW is usually the best performer of the eight algorithms compared.

5.2 Experimental results for the DSBM

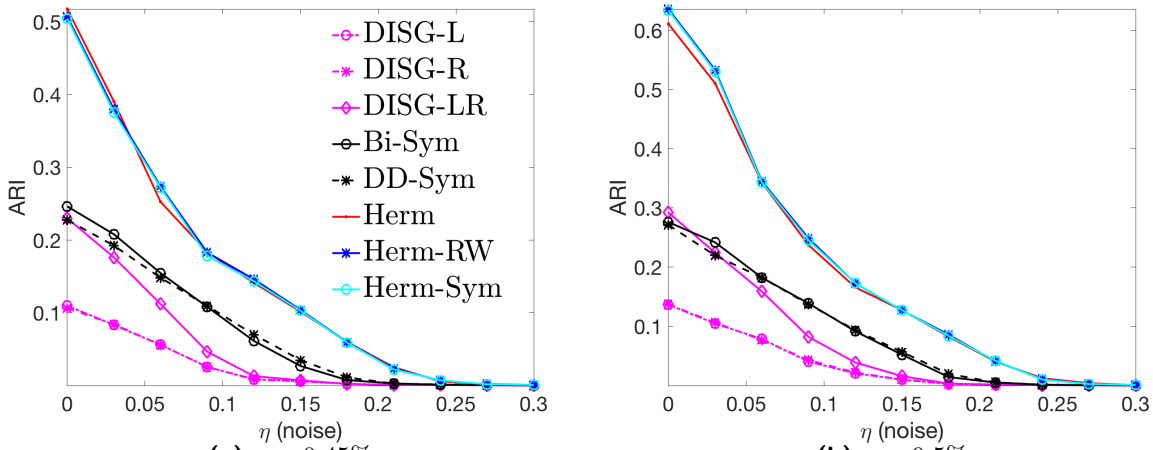
We run experiments on randomly generated block models with different values of n, p, q, and flow matrix F. Note that spectral techniques perform better for the stochastic block model for a large value of p, hence our focus here is to compare the performance of different algorithms when the value of p is close to the connectivity threshold $\log(N)/N$ of a random $\mathcal{G}(N, p)$ graph. Our reported results are based on the average over the 10 independently generated networks for every fixed parameters (i.e., n, k, p, q, and F). For the purpose of better visualisation, we assume that the non-zero entries of pattern matrix F have only two different values η and $1-\eta$, and the experimental results are reported with respect to η.

Figure 3 and Figure 4 report the performance of all the tested algorithms for the input graphs from the DSBM where $N = 5,000$, $k = 5$, and the flows among different clusters form a directed cycle (as in Section 4.2), and a complete graph where edges have random orientations, respectively. These two figures show that the three variants of our algorithm perform much better than all the other tested algorithms. Furthermore, these three variants give similar results due to the fact that all the vertices have the same expected degree. Also notice that, while all the algorithms are unable to find a meaningful cluster structure when η is close to 0.5, our algorithm performs much better than the others for a large η.

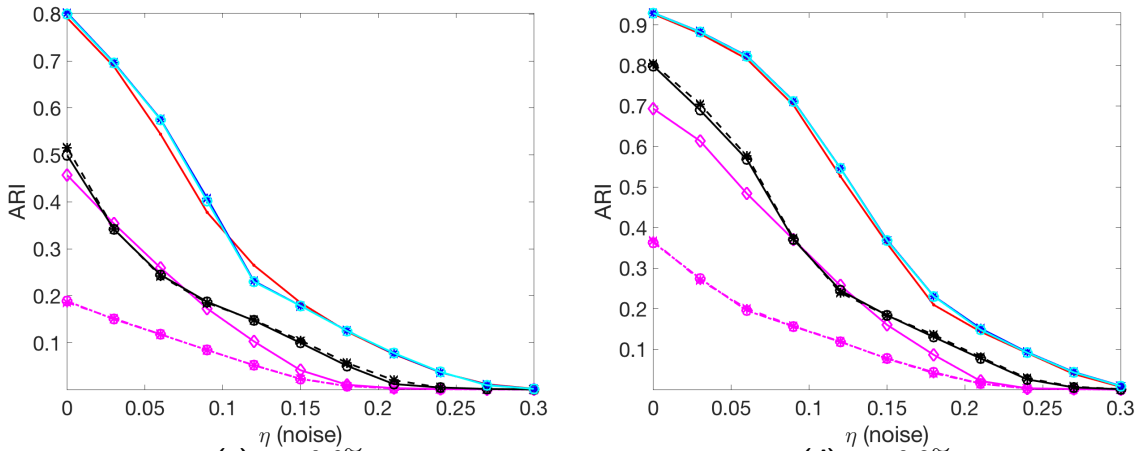
We further investigate the performance of all algorithms for a large value of k. The ARI values for a randomly generated graph with $N = 5,000$ vertices, $k = 50$ clusters, and the underlying meta-graph is a complete graph, are depicted in Figure 5, with respect to different values of p and η. This regime of parameters, i.e., large value of k and relatively small p, is particularly challenging and interesting because of its prevalence in most real-world data sets, and clearly illustrates that our Hermitian adjacency matrix based method has overwhelmingly superior performances than all other existing algorithms from the literature.

5.3 Experimental results for real-world data

This subsection details the outcomes of experiments on a variety of real-world data sets, showcasing the efficiency and robustness of our algorithm for identifying structures in directed graphs. In contrast to the graphs from the DSBM, there is no ground truth clustering in such data, and we compare performance on a number of objective functions, showing at the same time that our

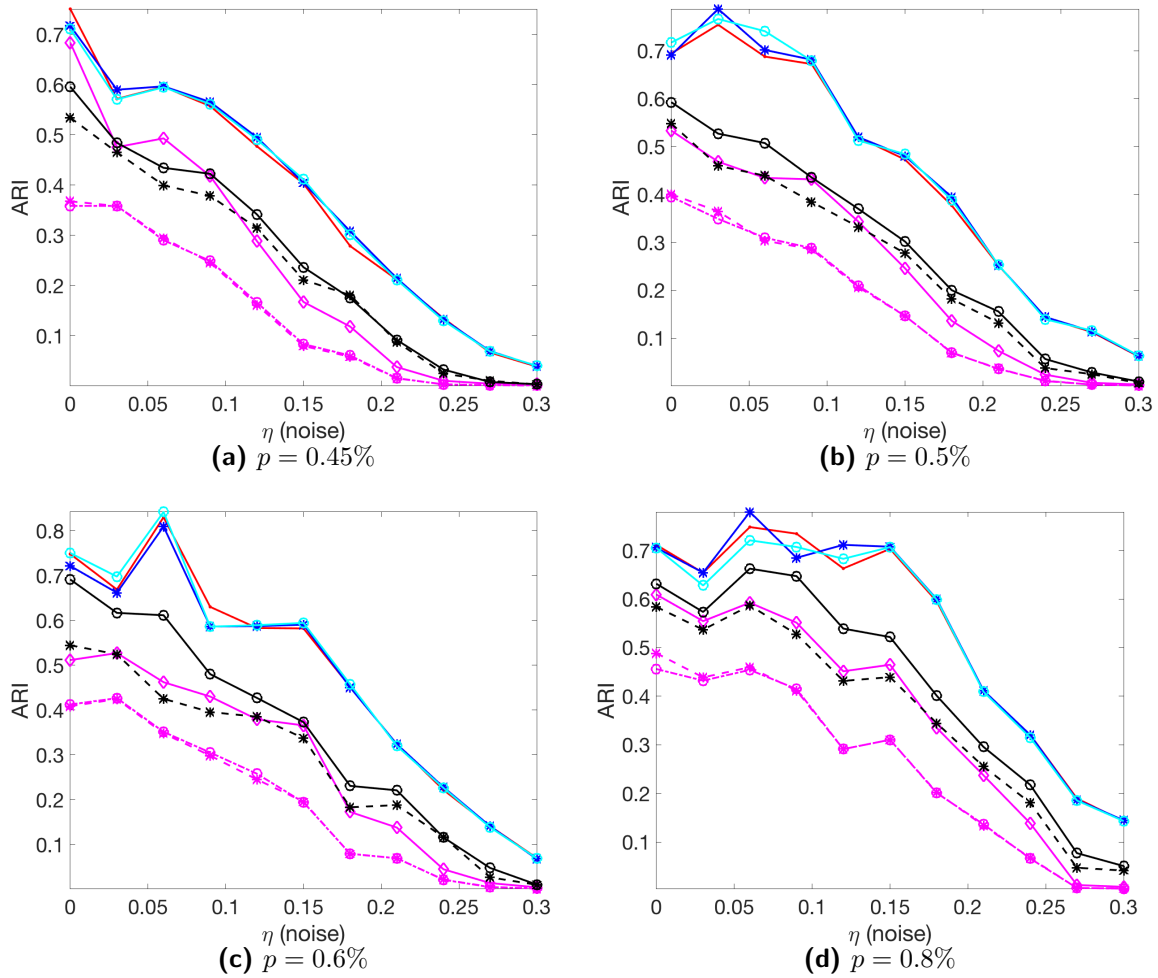


(a) $p = 0.45\%$ **(b)** $p = 0.5\%$



(c) $p = 0.6\%$ **(d)** $p = 0.8\%$

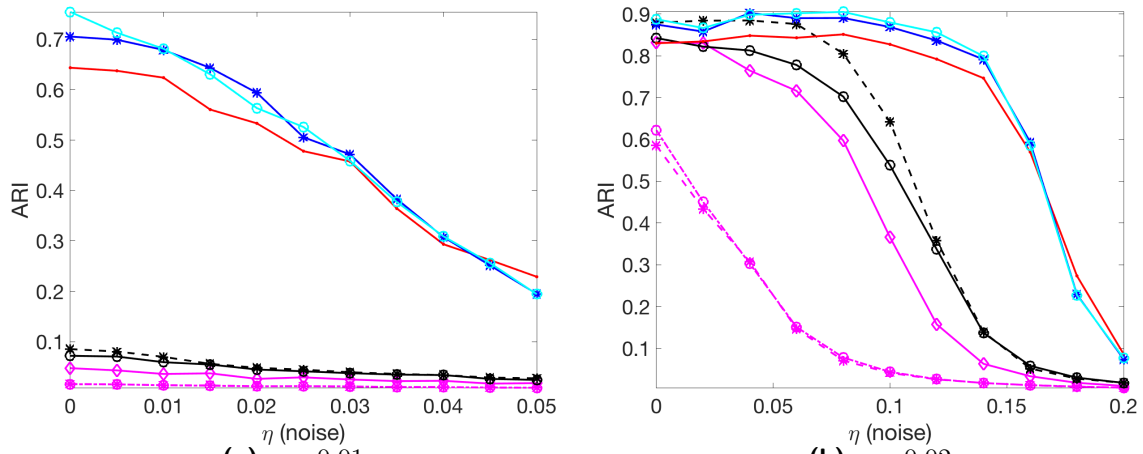
Figure 3: Recovery rates for the circular pattern in the DSBM with $k = 5$, $N = 5000$, at various levels of sparsity. Averaged over 10 runs.



(a) $p = 0.45\%$ **(b)** $p = 0.5\%$

(c) $p = 0.6\%$ **(d)** $p = 0.8\%$

Figure 4: Recovery rates for the complete meta-graph in the DSBM with $k = 5$, $N = 5000$, at various levels of sparsity. Averaged over 10 runs.



(a) $p = 0.01$ **(b)** $p = 0.02$

Figure 5: Recovery rates for the complete meta-graph in the DSBM with $k = 50$, $N = 5000$, two sparsity values p. Averaged over 10 runs.

clustering methodology favours balanced cluster sizes. We consider the following four data sets: the US-MIGRATION network, the BLOG network during the 2004 US presidential election, and in the Appendix, the UK-MIGRATION data set, and the c-ELEGANS neural network.

Given a pair of clusters (X, Y), we quantify the measure of imbalance of a cut via the Cut Imbalance ratio CI defined by

$$\mathrm{CI}(X,Y) = \frac{w(X,Y)}{w(X,Y)+w(Y,X)}, \tag{8}$$

which indicates an imbalance for values closer to 0 or 1. Furthermore, we introduce the normalisation by size, respectively volume, which we aim to maximise

$$\mathrm{CI}^{\mathrm{size}}(X,Y) = |\mathrm{CI}(X,Y) - 0.5| \cdot \min\{|X|, |Y|\}, \tag{9}$$

$$\mathrm{CI}^{\mathrm{vol}}(X,Y) = |\mathrm{CI}(X,Y) - 0.5| \cdot \min\{\mathrm{vol}(X), \mathrm{vol}(Y)\}, \tag{10}$$

where $\mathrm{vol}(X) = \sum_{j\in X} d_j^{\mathrm{in}} + d_j^{\mathrm{out}}$, denotes the sum of the total degrees of vertices in X, and $w(X,Y) = \sum_{j\in X, \ell\in Y} w(j,\ell)$ denotes the total weight of all edges flowing from X to Y. The term $|\mathrm{CI}(X,Y) - 0.5|$ is motivated by our goal of quantifying the distance to the uninteresting scenario of a balanced cut which has $CI(X,Y) = 0.5$, while the normalisation by minimum size or volume is to penalise for small-sized clusters, in the spirit of the Normalised Cut of [23].

Since our goal is to uncover structures in the graph in terms of flow imbalance between pairs of subsets of vertices, we quantify this measure for the entire graph by considering all $\binom{k}{2}$ pairs of clusters, and summing together the largest $2k$ $\mathrm{CI}^{\mathrm{vol}}$ values:

$$\mathrm{TopCI}^{\mathrm{vol}} = \sum_{t=1}^{2k} \mathrm{CI}^{\mathrm{vol}}(C_{j_t}, C_{\ell_t}) \tag{11}$$

where (C_{j_t}, C_{ℓ_t}) denotes the t-th largest $\mathrm{CI}^{\mathrm{vol}}$ cut imbalance pair. We only consider the top $2k$ largest values since we do not expect for cut imbalances to manifest between all pairs of clusters (in other words, the meta-graph F is typically also sparse). We also define an analogous measure for the cluster-size normalisation and we denote it by $\mathrm{TopCI}^{\mathrm{size}}$.

US Migration Network Our first and main data set is the 2000 US Census data, which reports the number of people that migrated between pairs of counties in the US during the 1995-2000 time frame [2, 19]. This data set can be captures by an $N \times N$ matrix M, where $N = 3107$ denotes the number of counties in mainland US and $M = (M_{j\ell})_{0\le j,\ell\le N-1}$ denotes

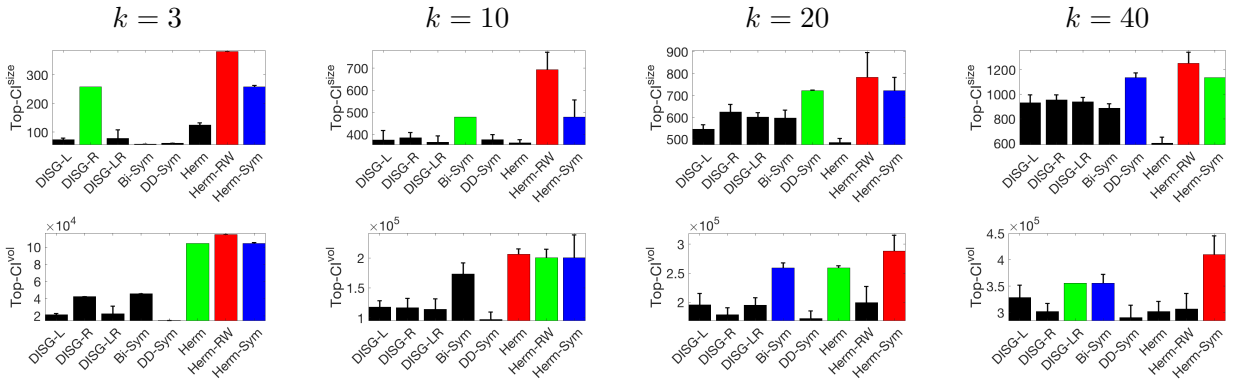


Figure 6: Objective function values TopCI$^{\text{size}}$ (top) and TopCI$^{\text{vol}}$ (bottom) for the US-MIGRATION-I data set Averaged over 10 runs.

the total number of people that migrated from county j to county ℓ during the five-year period. Data is also available on the population of each county, but we have not taken that into account in our study. We consider the two following variations of this data set; in both for which, the input matrix to our pipeline is the skew symmetric matrix $G = M - M^\intercal$.

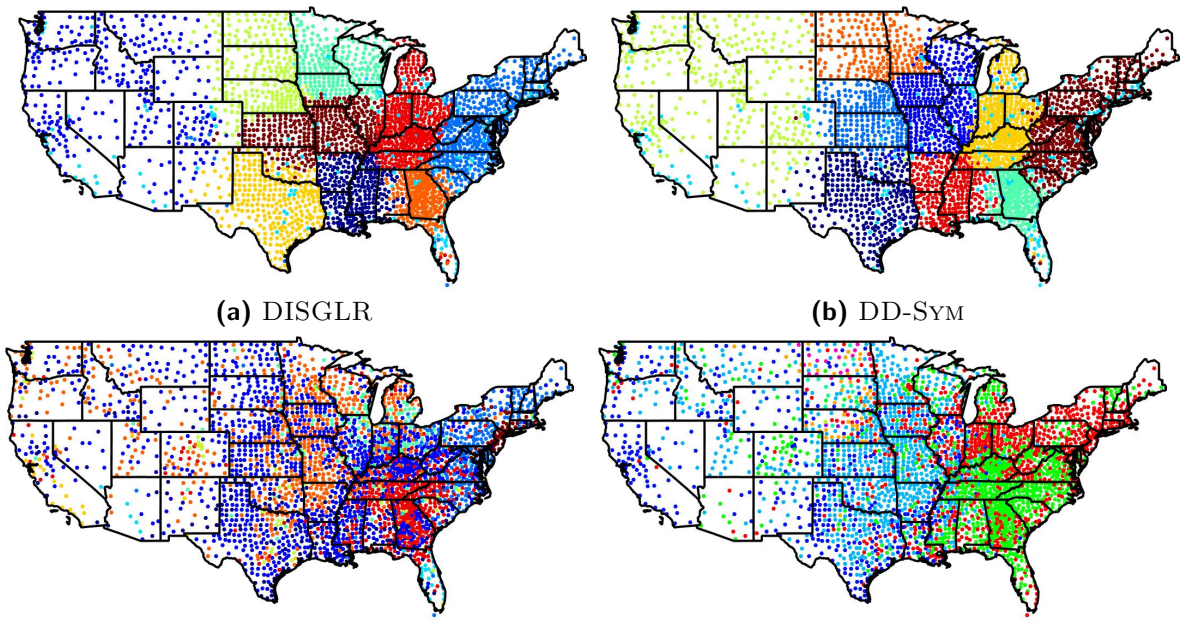
US-Migration-I: The first variation concerns the following transformation of the data $\widetilde{M}_{j\ell} = \frac{M_{j\ell}}{M_{j\ell}+M_{\ell j}}$, which leads to a matrix often encountered in certain applications. For example, in ranking, this could capture the fraction of games won by player j in the match against ℓ. Figure 6 shows the aggregated TopCI$^{\text{size}}$ and TopCI$^{\text{vol}}$ objective function values, for $k = \{3, 10, 20, 40\}$. When considering TopCI$^{\text{size}}$, HERM-RW is consistently ranked first across all values of k, and outperforms all other methods by a large margin especially for $k = \{3, 10\}$. Similar statements hold true also for TopCI$^{\text{vol}}$, especially for $k = \{3, 10\}$, when the three Hermitian-based approaches are ranked in the first three positions, and outperform the other five methods by a large margin.

Figure 7 shows the clusterings recovered by several methods for $k = 10$. Figure 8 shows heatmaps of the graph 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, our two methods 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 9 illustrate the four largest size-normalised cut imbalance pairs, i.e., the pairs of clusters for which CI$^{\text{size}}(C_j, C_\ell)$ is largest, for the case $k = 10$. We highlighted the two clusters in each pair in red (source) and blue (destination), and provided the numerical values for their respective cut imbalances CI, CI$^{\text{size}}$ and CI$^{\text{vol}}$. Looking at the values of the two normalised cut imbalances, HERM-RW vastly outperforms all of the competing methods.

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[2]. To this end, we pre-process the migration matrix M and cap all entries at $10,000$, which corresponds to the 99.9% percentile. A simple symmetrisation of the input matrix $M \mapsto M + M^\intercal$, followed by standard spectral clustering of undirected graphs [25], will reveal clusters that align very well with the state boundaries [5], as shown in Figure 1a. Figure 10 shows the TopCI$^{\text{vol}}$ objective function values from which we conclude that HERM-RW is the best performing method for $k = \{3, 5, 8\}$, and places third for $k = 10$ after DISG-LR and DISG-L.

BLOG: The last data set we consider is the the BLOG network during the 2004 US presidential election. This data set was considered by Adamic and Glance [1], who recorded

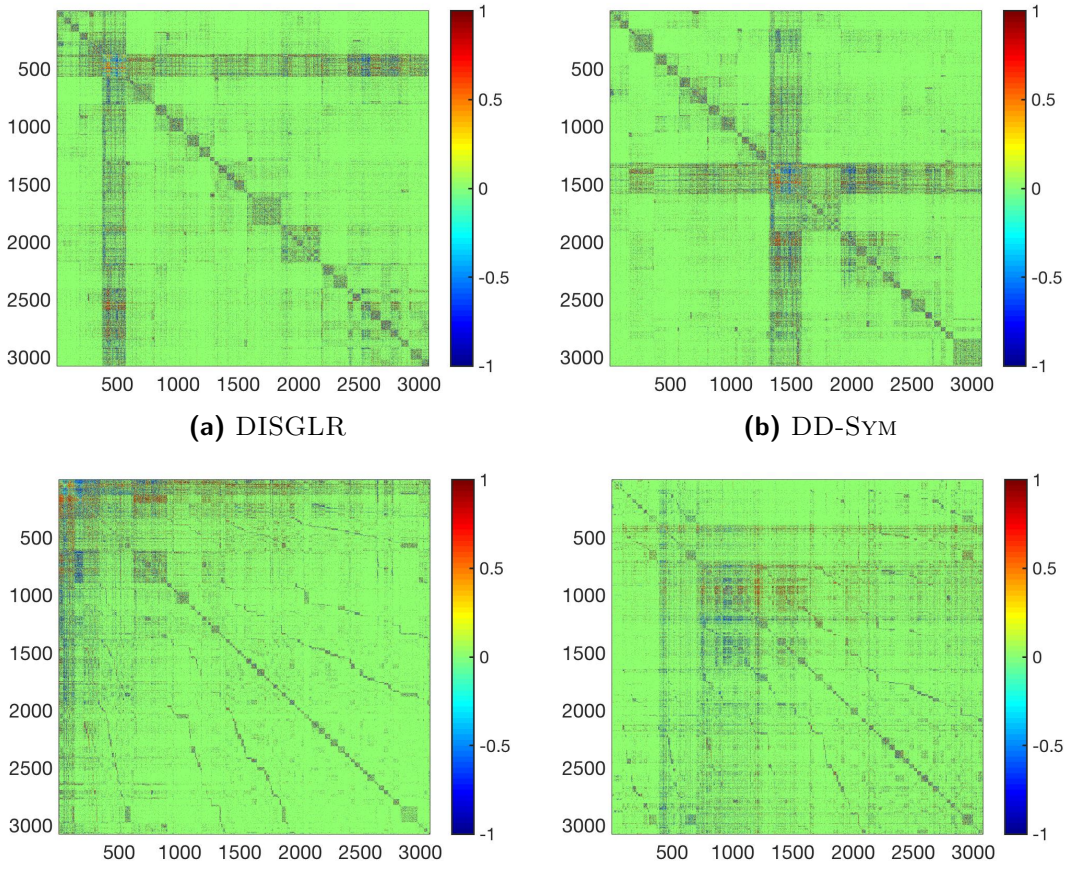
[2]We remark that without dropping the outlier entries, Herm-RW ranks first in terms of TopCI$^{\text{vol}}$ performance for every single value of $k = \{3, 5, 8, 10, 20, 30, 40\}$.



(a) DISGLR **(b)** DD-SYM

(c) HERM **(d)** HERM-RW

Figure 7: Recovered clusterings for the US-MIGRATION-I data set with $k = 10$ clusters, across all methods.



(a) DISGLR **(b)** DD-SYM

(c) HERM **(d)** HERM-RW

Figure 8: Heatmap of the graph adjacency matrices, sorted by induced cluster membership, for the US-MIGRATION-I data set with $k = 10$ clusters, across all methods.

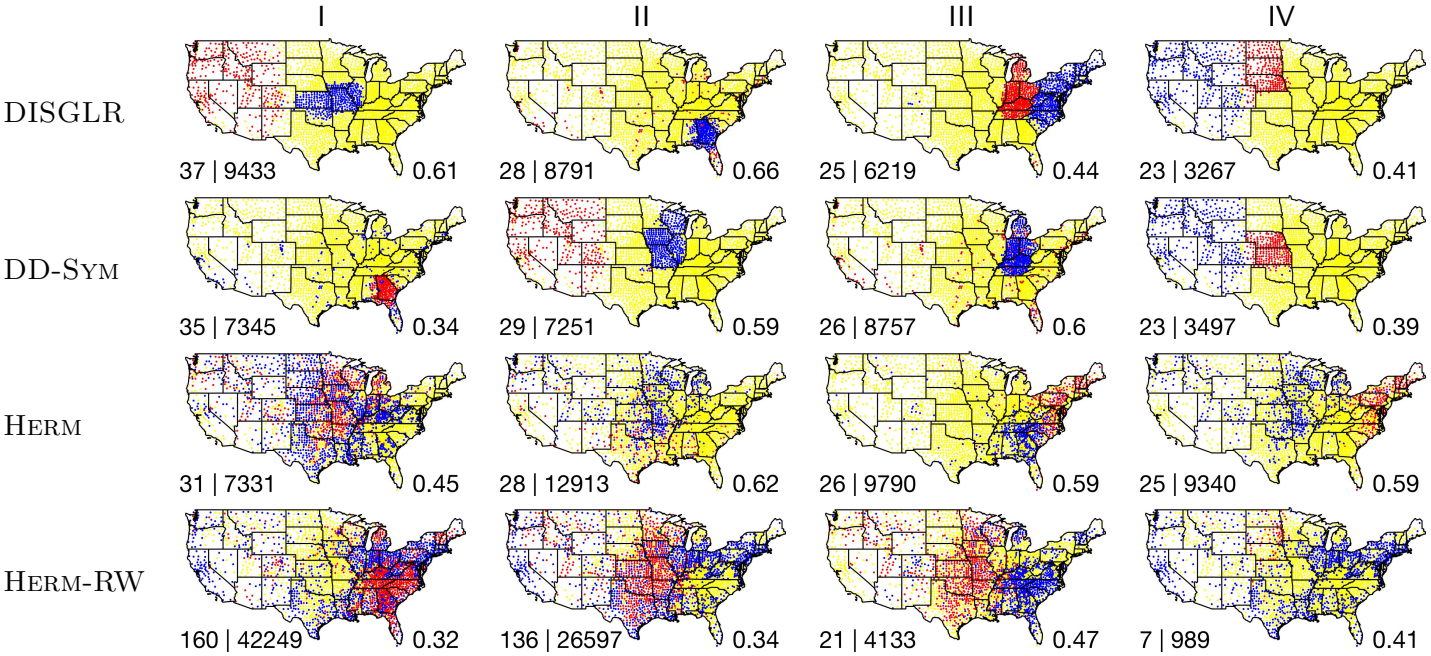


Figure 9: The top four largest size-normalised cut imbalance pairs for the US-Migration-I data with $k = 10$ clusters. Red denotes the source cluster, and blue denotes the destination cluster. For each plot, the bottom left text contains the numerical values (rounded to nearest integer) of the normalised CI^{size} and CI^{vol} pairwise cut imbalance values, and the bottom right text contains the CI cut imbalance value in $[0, 1]$.

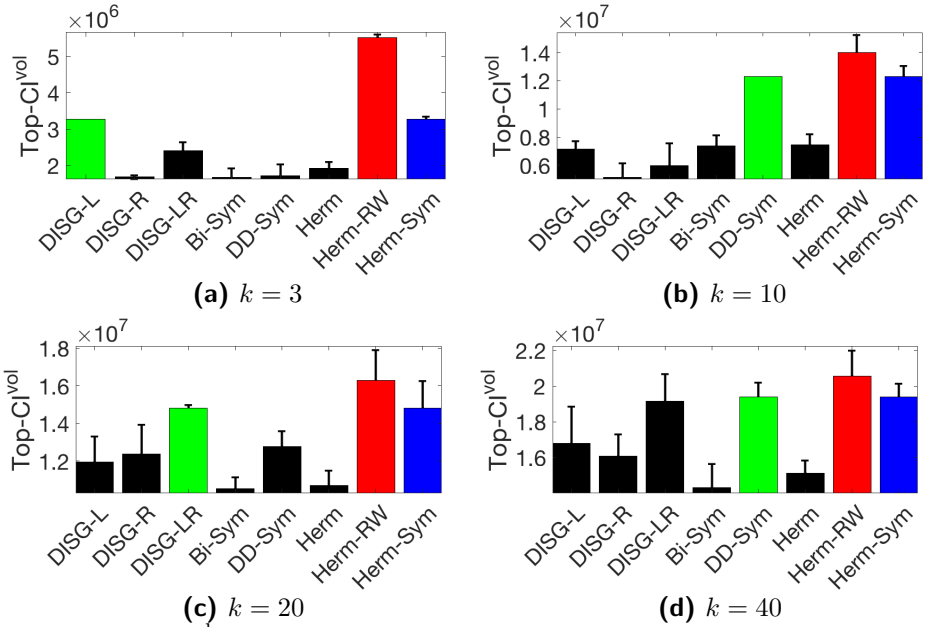


Figure 10: The $\text{TopCI}^{\text{vol}}$ objective function values for the US-Migration-II data set (averaged over 10 runs).

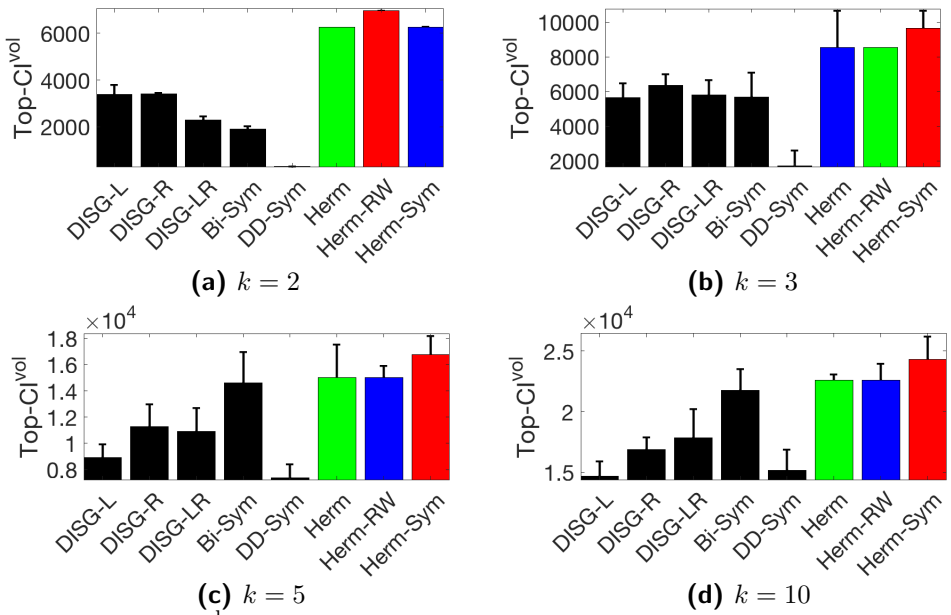


Figure 11: The TopCI$^{\text{vol}}$ objective function values, for the BLOG data set with $N = 1,212$ (averaged over 20 runs).

the hyperlink connections among political blogs and revealed that such connections were highly dependent on the blog's political orientation. For this data set, due to some of the nodes having zero in-degree or out-degree, we regularise the adjacency matrix G by shifting it by a multiple of the identity. Figure 11 compares the TopCI$^{\text{vol}}$ objective function across all algorithms, for the largest connected component of the BLOG network, which contains $N = 1,212$ blogs, making this graph the second largest in size of the four data sets we considered. Even so, we included in the analysis a clustering with $k = 2$ since the BLOG network has an underlying cluster structure with two clusters corresponding to the Republican and Democratic parties. On this data set, our three algorithms vastly outperform all other methods and always rank in the top three, with HERM-SYM and HERM-RW being the two best performers.

References

[1] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43, 2005.

[2] U. S. Census Bureau, 2002. www.census.gov/population/www.cen2000/ctytoctyflow/index.html.

[3] Fan Chung and Mary Radcliffe. On the spectra of general random graphs. *Electronic Journal of Combinatorics*, 18(1), 2011.

[4] M. Cucuringu. 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, 2016.

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

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

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

[8] M. Fanuel and J.A.K. Suykens. Deformed laplacians and spectral ranking in directed networks. *Applied and Computational Harmonic Analysis*, 2017.

[9] Office for National Statistics. Internal migration: detailed estimates by origin and destination local authorities, age and sex, 2018.

[10] Alexander J. Gates and Yong-Yeol Ahn. The impact of random models on clustering similarity. *Journal of Machine Learning Research*, 18(87):1–28, 2017.

[11] Krystal Guo and Bojan Mohar. Hermitian adjacency matrix of digraphs and mixed graphs. *Journal of Graph Theory*, 85(1):217–248, 2017.

[12] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: first steps. *Social Networks*, 5(2):109–137, 1983.

[13] Amit Kumar, Yogish Sabharwal, and Sandeep Sen. 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, 2004.

[14] James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multiway spectral partitioning and higher-order Cheeger inequalities. *Journal of the ACM*, 61(6), 2014.

[15] Jing Lei and Alessandro Rinaldo. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237, 2015.

[16] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[17] Fragkiskos D. Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks: A survey. 2013.

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

[19] M. J. Perry. State-to-State Migration Flows: 1995 to 2000. *Census 2000 Special Reports*, 2003.

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

[21] Karl Rohe, Tai Qin, and Bin Yu. Co-clustering directed graphs to discover asymmetries and directional communities. *Proceedings of the National Academy of Sciences*, 113(45):12679–12684, 2016.

[22] Venu Satuluri and Srinivasan Parthasarathy. Symmetrizations for clustering directed graphs. In *Proceedings of the 14th International Conference on Extending Database Technology*, pages 343–354, 2011.

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

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

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

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

A Omitted proof details

In this section we present the omitted technical details about the analysis of our algorithm presented in Section 3. We first introduce some notations 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 4.

Theorem 6 (Davis-Kahan, [7]). *Let $H, R \in \mathbb{R}^{d\times d}$ be Hermitian. Then, for any $a \leq \beta$ and $\delta > 0$,*

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

The other lemma that will be used in the analysis, more specifically in the proof of Lemma 3, is the following matrix concentration inequality.

Theorem 7 ([3]). *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 3.

Proof of Proposition 2. First of all, we assume that F is the matrix with two identical rows indexed by j and ℓ, and we prove that $\widetilde{F}$ is a 0-distinguishable 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 $P_{\mathrm{Im}(\widetilde{F})}(j,\cdot) = P_{\mathrm{Im}(\widetilde{F})}(\ell,\cdot)$ for $0 \leq j \neq \ell \leq k-1$ and consider the vector $\chi \in \{0,1\}^k$ which is 1 in the jth and ℓth entry, and zero elsewhere. Clearly, $P_{\mathrm{Im}(\widetilde{F})}\chi = \mathbf{0}$. 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. □

Proof of Lemma 3. Let $M^{uv} \in \mathbb{C}^{N\times N}$ be the matrix with exactly two non-zero entries defined by $(M^{uv})_{u,v} = i$, $(M^{uv})_{v,u} = -i$. By definition, $(M^{uv})^2$ has exactly two nonzero entries, i.e.,

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

Let X^{uv} be a random matrix defined as follows

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

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 $\mathbb{E}X^{uv} = p\widetilde{F}_{j,\ell}M^{uv}$ and

$$\mathbb{E}(X^{uv} - \mathbb{E}X^{uv})^2 = (1-p)\left(-\mathbb{E}X^{uv}\right)^2 + pF_{j,\ell}\left(M^{uv} - \mathbb{E}X^{uv}\right)^2 + pF_{\ell,j}\left(-M^{uv} - \mathbb{E}X^{uv}\right)^2.$$

In addition, it holds that

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

since the spectral norm of a matrix is upper bounded by the sum of the absolute values of the entries in each row. By setting $a = 10\sqrt{pkn\log(kn)}$, $M = 1$, $\sigma^2 \leq pkn$ and $d = kn$, we apply Theorem 7 to obtain the statement. □

Proof of Lemma 4. 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 3 it holds for any $1 \leq j \leq \ell$ that $\widetilde{\rho}pn - \epsilon \leq |\lambda_j| \leq \widetilde{\rho}pn + \epsilon$. Notice that the other eigenvalues of A have absolute value less than ϵ. Therefore, based on assumption (6), and the relationship between the eigenvalues of $\widetilde{F}$ and $\mathbb{E}A$, we apply Theorem 6 and obtain

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

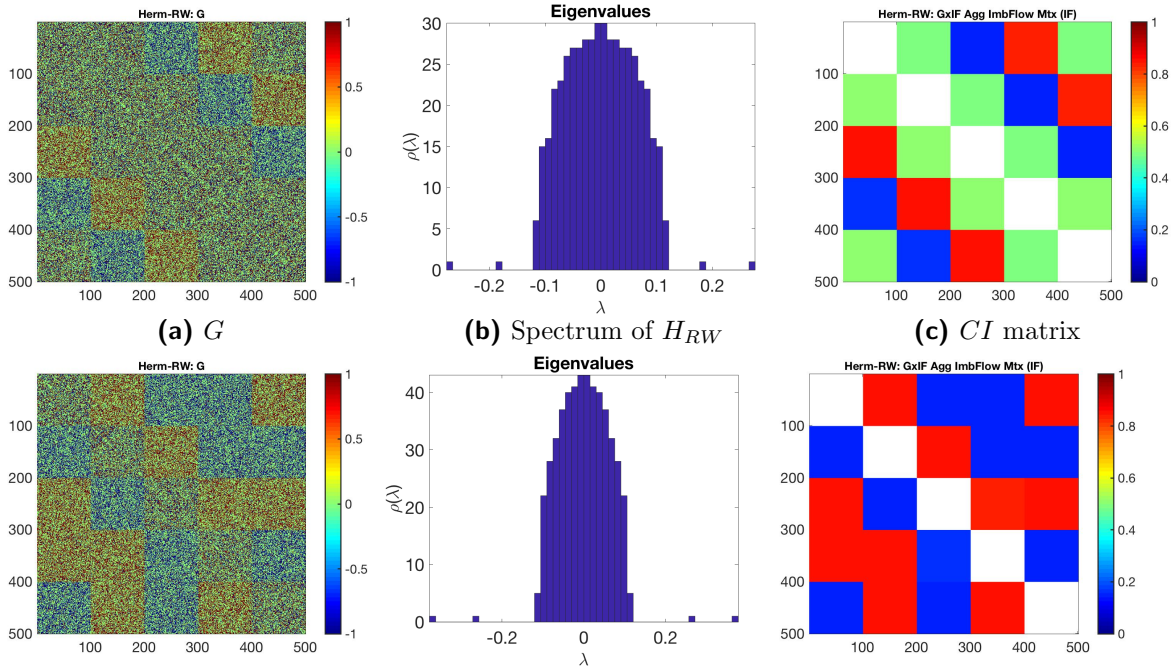
B Additional numerical experiments

All our experiments are performed in Matlab R2017b, on a *MacBook Pro, with 2.8 GHz Intel Core i7 and 16 GB of memory.*

Graphs from DSBM: Figure 12 depicts an instance of a graph from the DSBM model with $k = 5$ clusters, where each cluster is of size $n = 100$ with $p = 0.5$ and imbalance noise parameter $\eta = 0.15$. We show examples for the circular pattern structure (top) and the complete randomly oriented meta-graph structure F (bottom), highlighting the initial noisy adjacency matrices, the spectrum of A_{RW} leveraged by HERM=RW as well as the final recovered clustering-structure.

Figure 13 is a comparison on the DSBM model with $N = 10,000$ and $k = 20$ clusters, for both the complete graph and the circular structure. 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 circular pattern where the signal is weaker as there exist less pairwise interactions between the clusters. Note that in Figure 13 (b) we left out BI-SYM and DD-SYM from the comparison, due to their computational cost. Our algorithms not only significantly outperform all the other methods, but they also run significantly faster than Bi-Sym and DD-Sym which involve a sequence of matrix multiplication operations. For instance, for the fixed $N = 10,00$, $k = 20, p = 0.004$, and the pattern matrix defining a complete graph, Figure 14 illustrates the running time of the various algorithms, as a function of the noise level η, and this quantitative comparison holds in general for different choices of parameters.

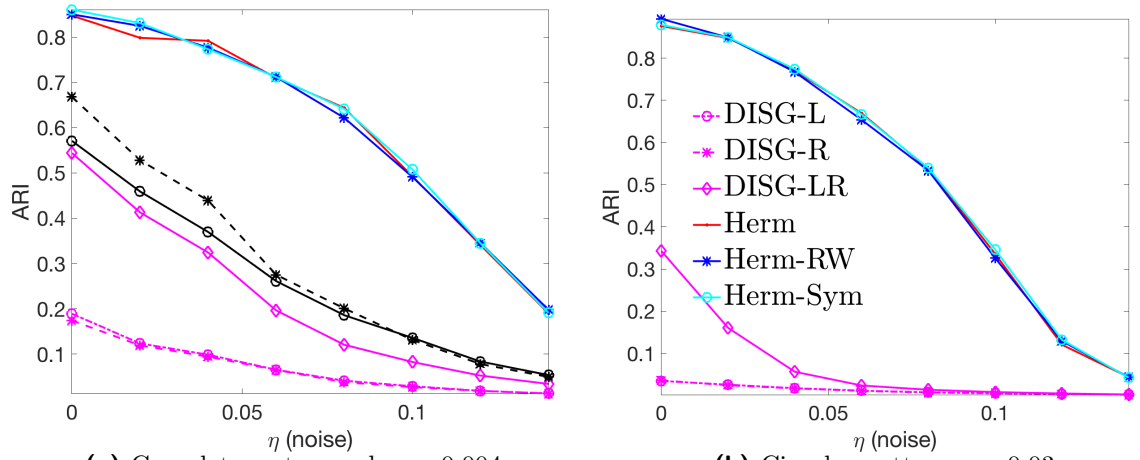
c-Elegans: Another data set we consider is the C-ELEGANS neural connectome network, which encodes connection between the neurons in a directed network [26]. This popular data set [16], also considered in [21], highlights significant dissimilarities between the sending and receiving patterns in the neural network. Figure 15 compares the TopCI$^{\text{vol}}$ objective function across all algorithms. For $k = 2$, respectively $k = 3$, our methods HERM-RW and HERM-SYM are the best performers, follows closely by BI-SYM, resp. HERM, while the rest of the algorithms perform significantly worse. For higher $k = \{5, 10\}$, results are mixed, with HERM and HERM-SYM among the top three performers, and BI-SYM as the best performer.



(a) G **(b)** Spectrum of H_{RW} **(c)** CI matrix

(d) G **(e)** Spectrum of H_{RW} **(f)** CI matrix

Figure 12: Recovery of an instance of the DSBM model, $N = 500$, $p = 0.5$, $\eta = 0.15$ and $k = 5$ clusters, for the circular pattern structure (top) and the complete randomly oriented meta-graph structure F (bottom).



(a) Complete meta-graph $p = 0.004$ **(b)** Circular pattern $p = 0.03$

Figure 13: Recovery rates for the complete meta-graph (left) and circular patterns (right) in the DSBM with $k = 20$ clusters, $N = 10,000$ at various levels of noise. Averaged over 10 runs.

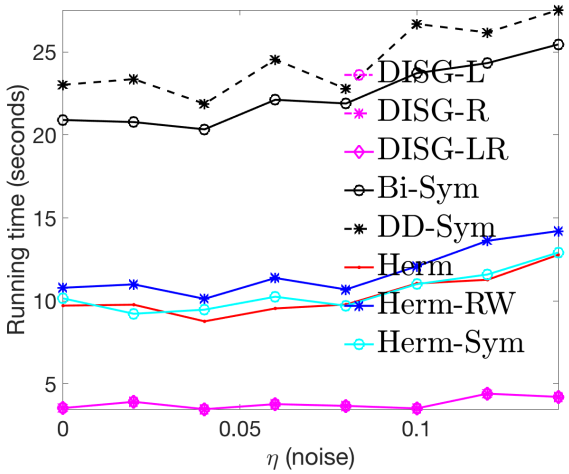
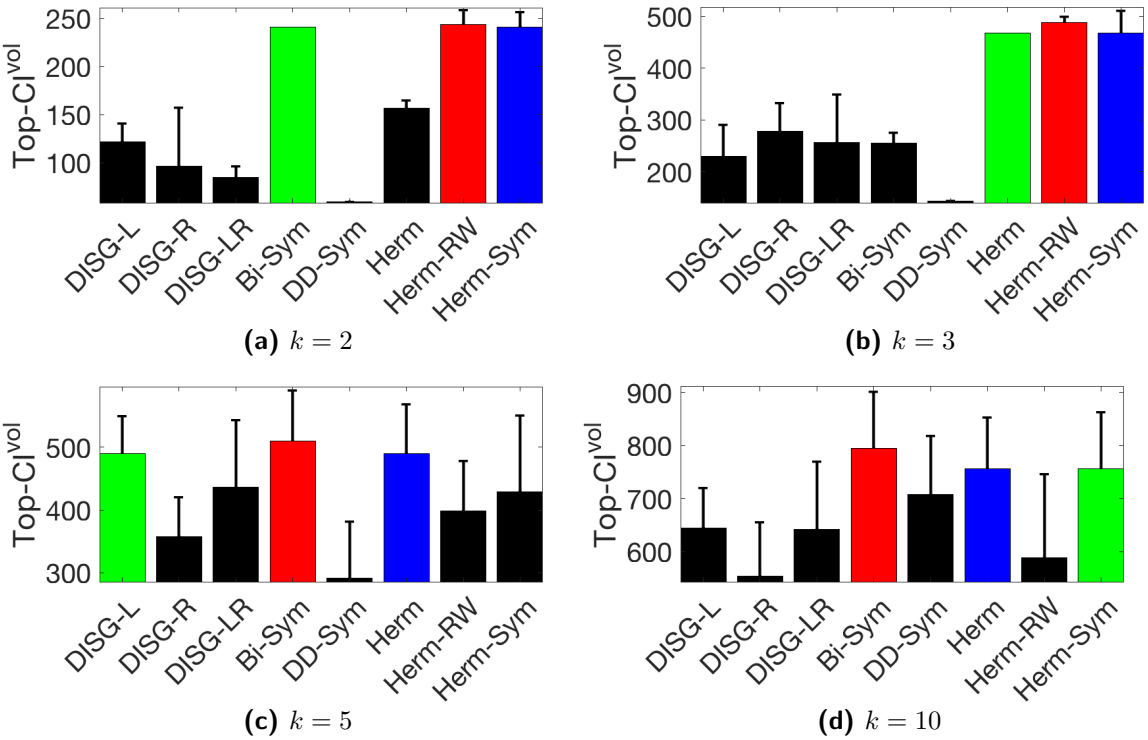


Figure 14: Runtime analysis



(a) $k = 2$ **(b)** $k = 3$

(c) $k = 5$ **(d)** $k = 10$

Figure 15: The TopCI$^{\text{vol}}$ objective function values for the C-ELEGANS data set with $N = 130$, for different number of clusters (averaged over 20 runs).

UK-Migration: The second real data set we considered is the UK-MIGRATION data set with $N = 354$, which captures in a directed graph the number of people who migrated between local authority Districts in the UK, aggregated over the interval 2012-2017 [9]. Figure 16 shows the TopCI$^{\text{vol}}$ objective function values for varying values of k. For $k = \{3, 5, 8\}$, HERM-RW is the best performer, and HERM-SYM ranks second for $k = \{3, 8\}$. For $k = 10$, the top three methods are DISG-L, DISG-LR and HERM-RW. Finally, Figure 17 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.

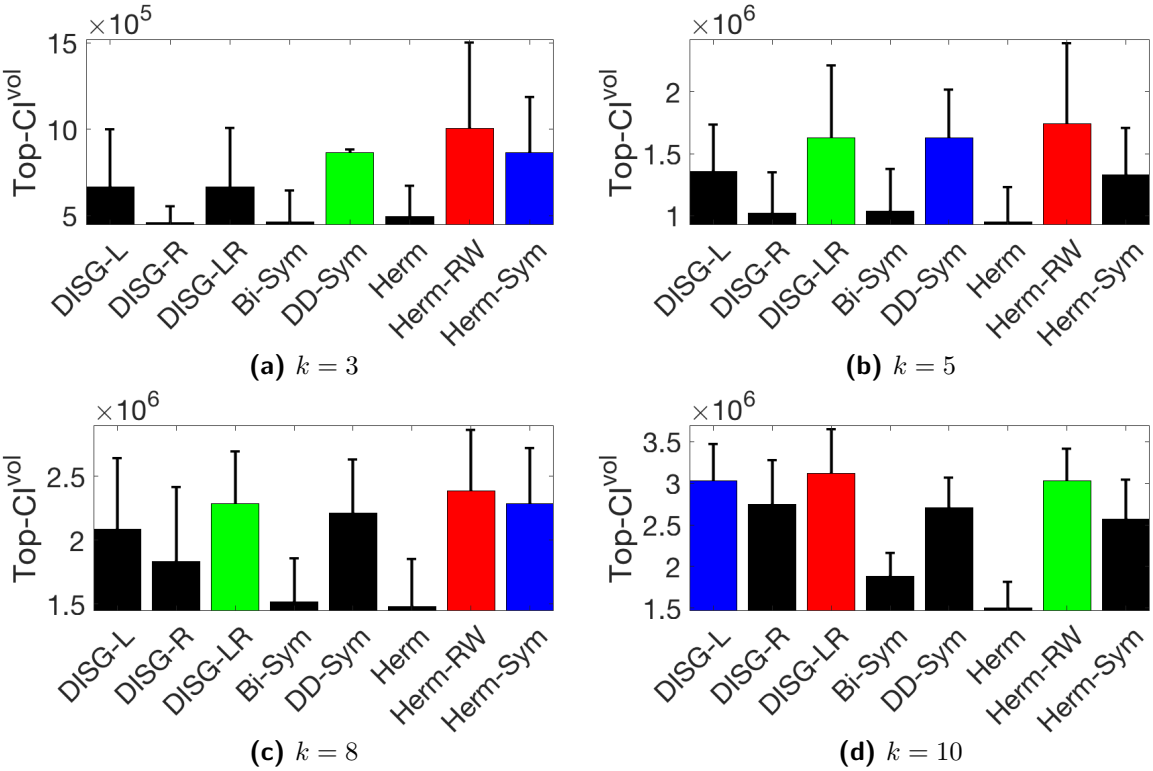


Figure 16: The TopCI$^{\text{vol}}$ objective function values for the UK-MIGRATION data set with $N = 354$ (averaged over 20 runs).

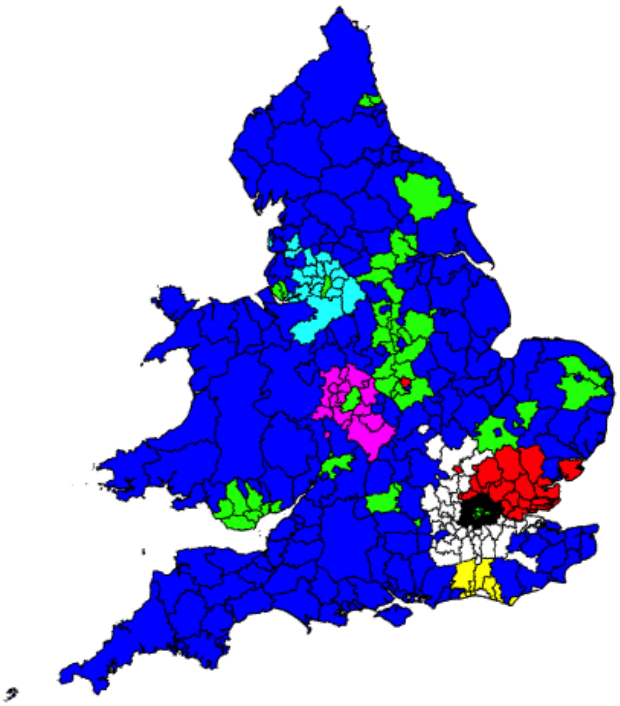


Figure 17: The clustering structure recovered by HERM-RW with $k = 8$ clusters, for the UK-MIGRATION data set.