Directed Graph Clustering Using Hermitian Laplacians

Steinar Laenen

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2019

Abstract

Graph clustering is a much-studied and important topic in computer science with widespread applications. Spectral clustering techniques have been a popular and effective method to cluster undirected graphs. However, performance of spectral techniques for directed graphs (digraphs) has been lacking. Issues that arise are (i) the broken symmetry of the adjacency matrix, (ii) the lack of a good objective function to measure clustering quality. Furthermore, the runtime of spectral clustering in general is dominated by the number of edges. Recently, a new digraph spectral clustering method based on Hermitian adjacency matrices has been proposed. This thesis analyses this new method theoretically and experimentally. Theoretically, we show that the cluster structure of a digraph with 3 clusters is encoded by the first eigenvector of the Hermitian Laplacian, from which we prove that spectral clustering provably works well. We further propose a distributed sparsification algorithm that reduces the number of edges – and thus decreases runtime – while maintaining the cluster structure. Experimentally, we use the UN Comtrade Database for the first time in the spectral clustering literature to show that the Hermitian method outperforms previous methods for spectral clustering of digraphs on our proposed objective functions. We also observe that the Hermitian method captures higher-order patterns in the cluster output. Finally, we test our distributed sparsification algorithm, and confirm that it indeed preserves cluster structure.

Acknowledgements

I would like to sincerely thank my supervisor, Dr. He Sun, for entrusting me to work on this exciting and fascinating topic with him. His guidance and untiring support motivated me tremendously.

I also express my gratitude to Luca Zanetti and Jasper Roosmale Nepveu, for being so kind to download the data we needed for our experiments and share it with us. Without their help, the data collection process would have taken significantly more time.

I thank my family and friends for their continuous support and help.

Contents

Chapter 1

Introduction

Clustering is the task of grouping similar objects together, and it is a much-studied and important topic in computer science. When the items to be clustered can be represented in a graph, we refer to it as *graph clustering*. Graph clustering (community detection) has become an important tool in machine learning and theoretical computer science to analyse large datasets, and has widespread applications in computer vision, social science, and network analysis and protein discovery [22, 36].

A simple and effective method to perform graph clustering is *spectral clustering*. Here, the graph is encoded into a k dimensional eigenspace, after which a k-means algorithm is applied to divide the graph into k clusters [43, 57]. Most work on spectral clustering has focused on undirected graphs. However, many datasets are better represented as directed graphs (digraphs), such as trading or flow networks. This makes it desirable to study spectral clustering of digraphs.

There are two main issues when applying spectral clustering on digraphs. Firstly, spectral clustering requires a symmetric adjacency matrix as input to the algorithm, and this is not the case for the standard adjacency matrix of a digraph. Secondly, the function to measure the quality of spectral clustering output for undirected graphs is based on edge densities [34, 52]. These objective functions ignore patterns that emerge from digraphs due to their directional nature (e.g. cycles, attractor or repelling clusters). Furthermore, the runtime of spectral clustering in general is dominated by the number of edges in a graph. This makes the algorithm slow for graphs with a large number of vertices and high connectivity.

This thesis studies the digraph clustering based on the Hermitian matrix representation of an input graph, which is initiated by Cucuringu et al. [15]. The thesis is organised as follows: In Chapter 2, we introduce basic concepts of graphs, matrices

and spectral graph theory that we use throughout the thesis. In Chapter 3, we analyse the eigen-structure of the Hermitian Laplacian matrix of a digraph. More specifically, we show that the Hermitian graph matrix encodes the structure of digraphs with $k = 3$ clusters, and we prove that spectral clustering obtains a good clustering result. In addition, we present and analyse a distributed sparsification algorithm for digraphs that preserves the input graph's cluster structure. In Chapter 4, we experimentally evaluate the performance of our algorithms on the UN Comtrade Database [42]. To the best of our knowledge, this is the first time that this database is studied in the clustering literature. We propose objective functions to measure the quality of clustering, and show that the Hermitian matrix method is able to extract the most meaningful clusters from the data. We finish Chapter 4 with experiments to show that our distributed sparsification algorithm works in practice and preserves the cluster structure. Due to page limit, all the omitted proofs and additional details can be found in the appendix.

Chapter 2

Background

This chapter serves as a brief introduction to review the basic concepts and definitions about graphs and matrices needed for the rest of the thesis. The chapter is organised as follows: in Section 2.1 we introduce some of the basic concepts of graphs, matrices, and their relation. In Section 2.2 we introduce the spectral clustering algorithm. In Section 2.3 we review related work on spectral clustering of digraphs and we review theoretical guarantees on the spectral clustering of undirected graphs.

2.1 Graphs and Matrices

A graph is a set of vertices that are connected by edges. We denote a digraph as a tuple $G = (V, E)$, where V is the set of vertices, and $E \subset V \times V$ is the set of edges that belong to the graph. We let $n = |V|$ denote the number of vertices and $m = |E|$ denote the number of edges. A directed edge $e \in E$ is denoted by an ordered tuple $e = (u, v)$ meaning the edge has a starting point (tail) $u \in V$ and an end point (head) $v \in V$. Since we work with real-world data in this thesis, where graphs will represent relations between objects, we will use weighted graphs $G = (V, E, w)$, where $w : V \times V \to \mathbb{R}_{\geqslant 0}$ is a function expressing the weights of edges. In this thesis we will mostly work with digraphs, meaning that $(u, v) \in E$ implies $(v, u) \notin E$.

Given two disjoint sets of vertices $S \subset V$ and $T \subset V$, we define the cut value from S to T as

$$w(S, T) \triangleq \sum_{x \in S, y \in T} w(x, y).$$

Note that in digraphs it does not always hold that $w(S, T) \neq w(T, S)$. We define the Net Flow (NF) between two clusters S and T as $\mathrm{NF}(S, T) \triangleq w(S, T) - w(T, S)$. If

S and T are not disjoint, i.e. $S \cap T \neq \emptyset$, we define their *symmetric difference* as $S \triangle T \triangleq (S \setminus T) \cup (T \setminus S)$.

A *matrix* $A \in \mathbb{R}^{m \times n}$ is an array with m rows and n columns. Taking the transpose $A^\intercal$ of a matrix is equivalent to swapping the rows and columns of a matrix, i.e., for an element in the matrix $A_{ij} = A^\intercal_{ji}$. If $A^\intercal = A$, we call a matrix symmetric. If it holds that for a matrix $A \in \mathbb{R}^{n \times n}$, a vector $v \in \mathbb{R}^n \setminus \{0\}$ and a scalar $\lambda \in \mathbb{R}$ that $Av = \lambda v$, then we say that v is an *eigenvector* of matrix A with corresponding *eigenvalue* λ. If $A \in \mathbb{C}^{m \times n}$ is complex-valued, we let $A^\dagger$ denote the conjugate transpose. If $A^\dagger = A$, we say that a matrix is *Hermitian*. We also note that the eigenvalues and eigenvectors of $A \in \mathbb{C}^{m \times n}$ are in the complex plane, i.e., $v \in \mathbb{C} \setminus \{0\}$ and $\lambda \in \mathbb{C}$ such that $Av = \lambda v$. For a given complex Hermitian matrix $A \in \mathbb{C}^{n \times n}$, and non-zero vector $x \in \mathbb{C}^n$, the *Rayleigh quotient* $\mathcal{R}(A,x)$ is defined as

$$\mathcal{R}(A,x) \triangleq \frac{x^\dagger A x}{x^\dagger x}.$$

Sometimes we write $\mathcal{R}(A,x)$ as $\mathcal{R}(x)$ when it is clear from the context which matrix we use for the Rayleigh quotient. An n^{th} root of unity is a complex number $\omega_n \in \mathbb{C}$ satisfying the equation $(\omega_n)^n = 1$, where n is a positive integer. Here we also drop the subscript n of ω when it is clear which root of unity we are using.

To encode a digraph $G = (V,E,w)$ into a matrix, we define the standard adjacency matrix $M \in \mathbb{R}^{n \times n}$ as $M_{uv} = 1$ if there is an edge (u,v), and 0 otherwise. Note that in digraphs the adjacency matrix is not symmetric. We define the Hermitian adjacency matrix $A \in \mathbb{R}^{n \times n}$ of a digraph G as

$$A_{uv} \triangleq \begin{cases} i \cdot w(u,v) & \text{if } (u,v) \in E \\ -i \cdot w(v,u) & \text{if } (v,u) \in E \\ 0 & \text{otherwise,} \end{cases} \tag{2.1}$$

where i is the imaginary number. We define the outer degree d_u^{out} of a vertex $u \in V$ as $d_u^{\text{out}} = \sum_{(u,v)} w(u,v)$, and the inner degree d_u^{in} we define as $d_u^{\text{in}} = \sum_{(v,u)} w(v,u)$. We denote the total degree d_u as $d_u = d_u^{\text{out}} + d_u^{\text{in}}$. We define the volume $\text{vol}(S)$ of a set of vertices S to be $\text{vol}(S) \triangleq \sum_{u \in S} d_u$.

Since graphs can contain vertices with high degrees, normalised matrices are used instead of the adjacency matrix. For normalisation it is customary to work with the *Laplacian* matrix of a graph. Let the degree matrix D be the diagonal matrix such that $D_{uu} = d_u$, and $D_{uv} = 0$ for $u \neq v$, and we define the Laplacian [54] L_G of a

graph $G = (V, E)$, with adjacency matrix A (which can be real or complex-valued) and degree matrix D as

$$L_G \triangleq D - A.$$

Sometimes we drop the subscript G if it is clear from the context. We then define the normalised Laplacian as $\mathcal{L}_G = D^{-\frac{1}{2}} L_G D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$. The Laplacian matrix is often used in spectral graph theory instead of the adjacency matrix, because the Laplacian is seen as more intuitive from a linear algebraic perspective [54].

Another form of normalising the adjacency matrix is using the *random walk* matrix, which is defined by

$$A_{\text{rw}} \triangleq D^{-1} A.$$

This matrix is widely studied in the literature [54], and is used as a transition matrix to encode a random walk on a graph, that starts in a specific vertex and randomly selects an edge to "walk" across to an adjacent vertex. We will use this matrix in our experiments chapter. For brevity, we refer to [13] for what graph properties can be derived from the algebraic properties of the Laplacian and random walk matrices.

For the rest of the thesis, we let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of $\mathcal{L}$ in *increasing* order with corresponding eigenvectors $f_1, \ldots, f_n$, and if we are referring to the eigenvalues $\lambda_1, \ldots, \lambda_n$ of the random walk matrix they will be in *decreasing* order. This is because the spectral clustering algorithm either uses the bottom k or top k eigenvectors of the Laplacian or random walk matrix respectively.

2.2 Spectral Clustering

For this section we assume that we use the normalised Laplacian instead of the random walk matrix. Using the normalised Laplacian, we can define spectral clustering as follows: (i) After encoding a graph G into its normalised Laplacian matrix $\mathcal{L}_G$, we compute the bottom k eigenvectors $f_1, ..., f_k$ of $\mathcal{L}_G$. Then, we map every vertex $u \in V$ to a point $F(u)$ in $\mathbb{R}^k$ according to

$$F(u) = \frac{1}{\sqrt{d_u}} \cdot (f_1(u), ..., f_k(u))^{\mathsf{T}}, \tag{2.2}$$

where d_u is the total degree of vertex u. (ii) we run k-means clustering on all the embedded points $F(u)$, and group the vertices into k clusters according to the k-means output [46]. See Figure 2.1 for illustration.

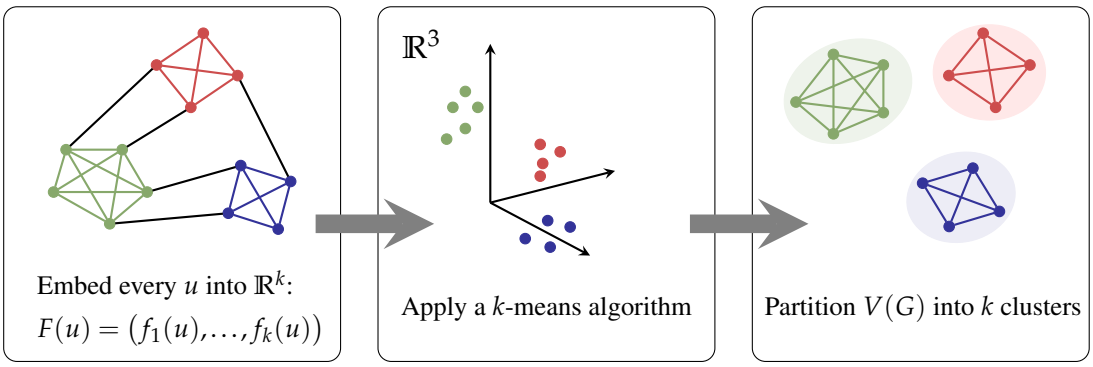

Figure 2.1: The three steps of spectral clustering for undirected graphs: (1) the algorithm embeds every vertex u of G to a point in $\mathbb{R}^k$ based on the bottom k eigenvectors of $\mathcal{L}_G$; (2) a k-means algorithm is applied to group the embedded points into k clusters; (3) the algorithm returns the partition from k-means as its output. Figure taken from [59].

Given a set of $X \subset \mathbb{R}^d$ points, in *k-means clustering*, the objective is to find a set of k centers $c_1, \cdots c_k$ that minimise the distance between each point $x \in X$ and the cluster c_i it is assigned to. For any *k-way partition of* X into $X_1, \ldots X_k$, where $\forall_{i \neq j} X_i \cap X_j = \emptyset$ and $\cup_{i=1}^k X_i = X$, we define the objective function as

$$\text{COST}(X_1, \cdots, X_k) \triangleq \min_{c_1,\ldots,c_k} \sum_{i=1}^{k} \sum_{x \in X_i} \|x - c_i\|^2, \tag{2.3}$$

which is being minimised in k-means clustering [46, 59].

2.3 Related Work

Spectral Clustering of Directed Graphs Not much work has been done on spectral clustering of digraphs. The most naive way to perform spectral clustering on digraphs with adjacency matrix M would be to turn the digraph into an undirected graph by $M^\intercal + M$. Cucuringu et al. [15] performed spectral clustering using this transformation on the US-Migration Dataset [11], which contains the migration data of the United States from county to county. The resulting clusters from the naive experiment are closely aligned with state borders. While the clustering result is easily interpretable, it is unsatisfactory; it is unsurprising that people tend to move within a state. This example highlights that there are two things that need to be adjusted compared to the undirected case. Firstly, the clustering metrics based on edge densities, such as the conductance and normalised cut value [34, 52], do not uncover the meaningful

patterns that emerge from digraphs [15]. Secondly, the real-valued adjacency matrix of a digraph is no longer symmetric. Therefore, the eigenvalues of the matrix are not necessarily real-valued, and it becomes unclear how to sort the eigenvectors. Since the first point is closely linked to the choice of objective functions with respect to specific datasets, we will defer to the experiments chapter for a more in-depth discussion. For the next chapter, the following notion of clustering in digraphs, with Net Flow NF as objective function, suffices: The edge direction imbalance within a cluster is lower than its edge direction imbalance with other clusters. More formally, the absolute value of the Net Flow NF within a cluster S is lower than the absolute of the NF between S and other clusters. We now address the second point, and look at different proposed methods to symmetrise the standard adjacency matrix to discuss it a bit.

Existing spectral methods for digraph clustering use the eigenvectors and eigenvalues of transformed forms of the standard adjacency matrix M, such as $M^\intercal M$, $MM^\intercal$ or a combination of both $M^\intercal M + MM^\intercal$. These have been applied in the context of bibliographic coupling and co-citation strength [37, 50, 51]. Experiments performed on the Wikipedia dataset and Cora dataset showed that $M^\intercal M + MM^\intercal$ performs better clustering than naive symmetrisation of the digraph [51]. Although these transformations symmetrise the adjacency matrix, they use expensive matrix multiplications, which time-computationally cost $O(n^\omega)$ where $\omega = 2.3737$ [14].

This thesis focuses on a new matrix representation proposed by Cucuringu et al., which uses a complex-valued Hermitian matrix representation of digraphs for spectral clustering [15]. The eigenvalues of a Hermitian matrix are real-valued, and thus can be used to sort the eigenvectors in spectral clustering. However, since each eigenvector is in $\mathbb{C}^n \cong \mathbb{R}^{2\times n}$, instead of using k real-valued eigenvectors to partition the graph into k clusters, one can use $k/2$ complex-valued eigenvectors, since that keeps the rank equal to k. If k is odd, [15] recommend taking the floor of $k/2$ (e.g for $k = 3$ use $f_1 \in \mathbb{C}^n$).

All methods ($M^\intercal M$, $MM^\intercal$, $M^\intercal M + MM^\intercal$ and Hermitian A method) encode different properties of the graph. However, we will leave the discussion of these differences to the end of Chapter 4. This allows us to place the discussion in the context of experimental results, and makes the comparison more clear.

Theoretical Guarantees of Spectral Clustering on Undirected Graphs Before theoretically analysing the Hermitian Laplacian method proposed by [15] for spectral clustering of digraphs, we first review some existing theoretical results for undirected graphs. This is because our analysis will aim to generalise some of these result to

digraphs.

Although undirected graph clustering procedures have seen widespread use over the past few decades, it is only recently that the relationship between the eigenvalues of the graph Laplacian, and the k-way expansion coefficient $\rho(k)$ has been established [34, 35, 46], where informally $\rho(k)$ is a measure for the optimal clustering result. In particular, Lee et al. [34] proved the higher-order Cheeger inequality, which upper and lower bounds $\rho(k)$ using $\lambda_k(\mathcal{L}_G)$. This means that the optimal clustering is low if and only if λ_k is small. Furthermore, a large gap between $\rho(k)$ and λ_{k+1} guarantees the existence of a k-way partition which has a better clustering result than a $(k+1)$-way partition. This result leads to the use of the stronger assumption that a gap between λ_k and λ_{k+1} ensures a good k-way clustering [61]. This assumption is well-grounded in experimental work. The number of clusters in a graph has been shown to be estimated well by large differences between adjacent eigenvalues [22, 57]. In spectral clustering, this gap assumption between λ_k and λ_{k+1} is typically used by setting k to the first index at which a large enough eigengap occurs.

To analyse the performance of spectral algorithms, random graphs are used to show performance guarantees. Randomly generated graphs exhibit the nice property that the ground-truth clustering is known, from which the number of correctly classified vertices can be deduced. An example of such a random graph model is the Stochastic Block Model [1]. SBMs generate graphs with k clusters, with parameters p and q. The parameter p determines the probability that there is an edge between to vertices within clusters. The parameter q determines the probability that there is an edge between vertices from different clusters. It is known that spectral clustering algorithms give correct clustering results for ranges of values p and q [40, 49, 58]. Although SBMs are useful for theoretical analyses, they do not always adequately capture all graph structures.

Peng et al. [46] study the performance of spectral clustering on graphs that are not captured by SBMS by making a slightly weaker gap assumption. They assume a large gap between $\rho(k)$ and λ_{k+1}. They show that given a large enough gap, the partition produced by spectral clustering is a good approximation of the optimal partitioning of the graph. Their gap assumption is a structural one, and does not depend on edges being chosen independently, like in the SBM. Thus, they give a stronger result that captures a larger general family of graphs with a multi-cluster structure.

The proposed Hermitian adjacency method for digraph spectral clustering by Cucuringu et al. [15] provides an analysis of the algorithm performance based on a gen-

eralisation of SBMs to Directed Stochastic Block Models (DSBM), where there are extra parameters that determine edge orientations between and within clusters. They provide an untight upper bound for the number of misclassified vertices in a DSBM with three clusters, and a cyclic orientation of the edge imbalance structure.

This thesis analyses the Hermitian adjacency method with a structural assumption, as was done in [46] for undirected graphs, rather than an input model for random graphs (DSBM). We show for $k = 3$ clusters with a cyclic orientation of the edge imbalance structure, that the output generated by a spectral clustering algorithm is close to the optimal clustering. Moreover, [46] assumes that each cluster has the same number of vertices, while our result generalises to clusters of different sizes. This thesis provides the first rigorous structural analysis of spectral clustering on Hermitian Laplacians of its kind.

Chapter 3

Algorithms

This chapter is organised organised as follows: in Section 3.1 we present our Structure Theorem for digraphs. This theorem is essential for analysing why spectral clustering performs well, and forms the basis in the design and analysis of our sparsification algorithm. In Section 3.2 we show that spectral clustering using Hermitian Laplacians performs well by proving several properties about the spectral embedding of "well-clustered" digraphs. In Section 3.3 we present a distributed algorithm for sparsifying digraphs that maintains the overall cluster structure of a digraph. In Section 3.4 we further discuss the gap between λ_1 and λ_2, and some limitations to our results.

Throughout the analysis of the algorithm we assume that the digraph being clustered has $k = 3$ clusters and we embed the vertices to points in $\mathbb{R}^2$ based on the bottom eigenvector of the Hermitian Laplacian matrix according to (2.2). We refer to the three clusters as S, T, R or S_1, S_2, S_3, respectively. We note that partitioning a digraph into three clusters such that the net flow $\mathrm{NF}(S,T) = w(S,T) - w(T,S)$ between the three clustered is maximised is NP-hard, and computing such quantity within a constant factor is NP-hard as well [23].

3.1 Structure Theorem

This section presents the main structural result of this thesis, which will be used in all subsequent analyses. We first give an informal discussion on our structure theorem. We introduce a parameter ζ which measures how well the graph is clustered. Given the first eigenvector $f_1 \in \mathbb{C}^n$ of $\mathcal{L}$, and a suitably defined indicator vector $h_G \in \mathbb{C}^n$ that encodes a cluster-structure of a graph, we show that f_1 and h_G are closely related if the gap ζ is large, i.e., the cluster structure of the graph is encoded well by the first

eigenvector. This suggests we can use the first eigenvector f_1 for k-means clustering, which we will show in the next section.

Before stating and proving the theorem of this section, we define quantities that allow us to study the relationship between the net flow NF between three clusters S, T and R, and the eigenvectors of the digraphs normalised Laplacian matrix, for the case that there are $k = 3$ clusters. We define an indicator vector $\chi_G(u)$ by

$$\chi_G(u) \triangleq \begin{cases} \omega & \text{if } u \in S \\ \omega^3 & \text{if } u \in T \\ \omega^2 & \text{if } u \in R, \end{cases}$$

where ω is the third root of unity. We further define the normalised indicator vector of χ_G as h_G, i.e.,

$$h_G \triangleq \frac{D^{1/2}\chi_G}{\left\|D^{1/2}\right\|}. \tag{3.1}$$

To prove the structure theorem, we make use of the following key lemma, which relates the first eigenvalue λ_1 of the normalised Laplacian $\mathcal{L}_G$ to the NF between S, T and R. This proof is a new contribution by this thesis, and can be found in Appendix A.1.

Lemma 3.1. *Given a weighted digraph $G = (V, E, w)$ with normalised Hermitian Laplacian $\mathcal{L}_G \in \mathbb{C}^{n\times n}$ and three non-empty disjoint clusters $S, T, R \subset V$, it holds that*

$$\lambda_1 \leqslant \mathcal{R}(h_G) = 1 - \frac{\sqrt{3}\cdot \mathrm{CF}(G)}{\mathrm{vol}(V)},$$

where the cyclic flow $\mathrm{CF}(G)$ is defined by

$$\mathrm{CF}(G) \triangleq \mathrm{NF}(S,T) + \mathrm{NF}(T,R) + \mathrm{NF}(R,S).$$

Following this lemma, we apply the techinque similar to Peng et al. [46] and define the quantity ξ as

$$\xi = \max_{\substack{S_i \subset V \\ \forall_{i\neq j} S_i \cap S_j = \emptyset \\ \forall_{i\neq j} S_i \cup S_j \neq V}} \frac{\lambda_2}{1 - \frac{\sqrt{3}\cdot \mathrm{CF}(G)}{\mathrm{vol}(V)}} = \max_{\substack{S_i \subset V \\ \forall_{i\neq j} S_i \cap S_j = \emptyset \\ \forall_{i\neq j} S_i \cup S_j \neq V}} \frac{\lambda_2}{\mathcal{R}(h_G)} \leqslant \frac{\lambda_2}{\lambda_1}. \tag{3.2}$$

It might not be immediately clear that the gap ξ for the directed case (based on the difference between λ_2 and cyclic flow CF) is equivalent to the gap Υ for the undirected case (based on the difference between λ_3 and $\rho(4)$ for $k = 3$ clusters). Indeed, it is true that in the undirected case the clustering objective is to *minimise* the conductance

$\rho(4)$, while in the directed case the objective is to *maximise* CF between clusters. However, because of our key Lemma 3.1, it suffices to minimise the quantity $1 - (\sqrt{3} \cdot \mathrm{CF}(G))/\mathrm{vol}(V)$, which thus makes the gaps ξ and Υ essentially equivalent. Now we formalise our Structure Theorem as follows:

Theorem 3.2 (The Structure Theorem)**.** *Let ξ be defined as in (3.2), and let h_G be defined as in (3.1). Then the following statements hold:*

1. *There is a coefficient $\alpha_1 \in \mathbb{C}$: $\widetilde{f_1} = \alpha_1 f_1$, such that $\left\| h_G - \widetilde{f_1} \right\|^2 \leqslant 1/\xi$.*
2. *There is a coefficient $\beta_1 \in \mathbb{C}$: $\widetilde{h_G} = \beta_1 h_G$, such that $\left\| f_1 - \widetilde{h_G} \right\|^2 \leqslant 1/(\xi - 1)$.*

Our theorem states that when ξ is large, we would expect that the span of the first eigenvector f_1 would be similar to that of the indicator vector h_G, in the sense that (i) h_G can be expressed using f_1, and (ii) f_1 can be expressed using h_G. We remark that our assumption that the ratio between λ_1 and λ_2 is large if they are clustered is experimentally verified and will be discussed in Section 3.4

For brevity and due to the page limit, we have omitted the proof of the Structure Theorem to Appendix A.2. We sketch the proof here as follows: We write h_G as a linear combination of the eigenvectors $f_1, \ldots, f_n$. We can also write the Rayleigh quotient as a sum of weighted eigenvalues, based on the projection of h_G. By rearranging terms, and choosing β_1 cleverly, we get the desired statements.

3.2 Analysis of Spectral Clustering

In this section we analyse the performance of spectral clustering on the spectral embedding of the Hermitian Laplacian of a digraph with $k = 3$ clusters. We start our analysis by using our Structure Theorem to show that points from the same cluster will be close to each other in the spectral embedding, but far from points in other clusters. This allows us to prove that given a suitable lower bound of the gap ξ, k-means clustering on this subspace provides a good approximation of the optimal result. The main result of this section is summarised as follows:

Theorem 3.3. *Let G be a digraph satisfying the condition that $\xi = \Omega(k)$, where ξ is as defined in (3.2), and $k = 3$. Let $F : V[G] \to \mathbb{C}$ be the embedding defined in (2.2). Let A_1, A_2, A_3 be a 3-way partition by any k-means algorithm running in $\mathbb{R}^2 \cong \mathbb{C}$ that achieves an approximation ratio APT, and let S_1, S_2, S_3 be the optimal partitioning. Then it holds for any $1 \leqslant i \leqslant 3$ that $\mathrm{vol}(A_i \triangle S_i) = O(APT/(\xi - 1)) \cdot \mathrm{vol}(S_i)$.*

We note that this theorem is similar to Theorem 1.2 of [46]. Although the overall structure of our proof is similar, several non-trivial adjustments are made to translate this proof from the undirected to directed case.

Analysis of spectral embedding First we investigate the spectral embedding of a digraph $G = (V, E, w)$ with three clusters. If we plot the projection of a digraph with three clusters onto its first eigenvector, we see that the points are separable (Figure 3.1), suggesting that k-means clustering will perform well. To prove these facts illustrated in Figure 3.1, we will show two nice properties of the spectral embedding in particular: (i) embedded points from the same cluster are concentrated around their center $c_i \in \mathbb{C}$, and (ii) embedded points from different clusters in G are far away from each other. These properties ensure that a k-means algorithm will produce a good clustering of the embedding.

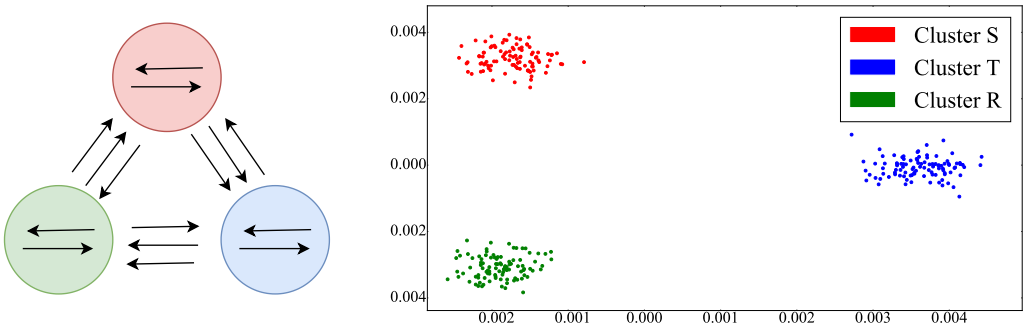

Figure 3.1: Embedding of a cyclic flow digraph G with three clusters S, T, R onto the f_1 of $\mathcal{L}_G$, where each cluster contains 100 vertices. The graph structure is illustrated on the left (taken from [33]), the embedding is illustrated on the right. Each point in the right illustration is a single vertex. For any edge, the probability that it exists is 0.8, and hence the digraph is not fully connected. For edges within the same cluster their orientation is chosen uniformly at random. The probability that there is a directed edge between vertices from S to T, T to R and R to S is 0.8.

Let us we define points $p^{(1)}, p^{(2)}$ and $p^{(3)} \in \mathbb{C}$ by

$$p^{(1)} \triangleq \frac{\omega}{\sqrt{\mathrm{vol}(S)}} \cdot \beta_1, \qquad p^{(2)} \triangleq \frac{\omega^3}{\sqrt{\mathrm{vol}(T)}} \cdot \beta_1, \qquad p^{(3)} \triangleq \frac{\omega^2}{\sqrt{\mathrm{vol}(R)}} \cdot \beta_1,$$

where ω is the third root of unity and β_1 is the parameter defined in statement 2 of Theorem 3.2. Even though these three points might not be the optimal centers, they suffice

for our analysis. The following lemma bounds the distance between the embedded vertices $F(u)$ to their corresponding points $p^{(i)}$:

Lemma 3.4. *It holds that*

$$\sum_{i=1}^{3}\sum_{u\in S_i} d_u \left\| F(u) - p^{(i)} \right\|^2 \leqslant \frac{1}{\xi - 1}.$$

Proof. We use $\omega^{(i)}$ to denote which exponent of the third root of unity is used per cluster, i.e., $\omega^{(1)} = \omega$, $\omega^{(2)} = \omega^3$ and $\omega^{(3)} = \omega^2$. Recall that we have by definition

$$\widetilde{h_G}(u) = \beta_1 h_G(u) = \omega^{(i)} \cdot \sqrt{\frac{d_u}{\mathrm{vol}(S_i)}}\beta_1.$$

Therefore, we have that

$$\begin{aligned}
\sum_{i=1}^{3}\sum_{u\in S_i} d_u \left\| F(u) - p^{(i)} \right\|^2 &= \sum_{i=1}^{3}\sum_{u\in S_i} d_u \left\| \frac{1}{\sqrt{d_u}} f_1(u) - \frac{\omega^{(i)}}{\sqrt{\mathrm{vol}(S_i)}}\beta_1 \right\|^2 \\
&= \sum_{i=1}^{3}\sum_{u\in S_i} \left\| f_1(u) - \omega^{(i)} \cdot \sqrt{\frac{d_u}{\mathrm{vol}(S_i)}}\beta_1 \right\|^2 \\
&= \sum_{i=1}^{3}\sum_{u\in S_i} \left\| f_1(u) - \widetilde{h_G}(u) \right\|^2 \\
&= \left\| f_1 - \widetilde{h_G} \right\|^2 \leqslant \frac{1}{\xi - 1},
\end{aligned}$$

where the last inequality follows by the second part of Theorem 3.2. □

The lemma above shows that, assuming that the value of ξ is large, the distance between the embedded vertices and their corresponding $p^{(i)}$s are small. Next, notice that the norm of each $p^{(i)}$ is

$$\left\| p^{(i)} \right\|^2 = \frac{1}{\mathrm{vol}(S_i)} \left\| \omega^{(i)}\beta_1 \right\|^2 = \frac{\|\beta_1\|^2}{\mathrm{vol}(S_i)}, \tag{3.3}$$

which implies that large clusters are closer to the origin and smaller clusters are further away from the origin. Finally, we show that the distance between each $p^{(i)}$ is lower bounded by the inverse square root of the smaller volume of each pair, i.e.,

$$\left\| p^{(i)} - p^{(j)} \right\|^2 \geqslant \frac{1}{\min\left\{\mathrm{vol}(S_i), \mathrm{vol}(S_j)\right\}} \tag{3.4}$$

We use a geometric argument to justify the inequality. Notice that all the $p^{(i)}$s are a $\frac{2\pi}{3}$ rotation away from each other in $\mathbb{C}$. This means that the points 0, $p^{(i)}$ and $p^{(j)}$ form

an obtuse triangle. Let's denote the distance between $p^{(i)}$ and $p^{(j)}$ as c, the distance between 0 and $p^{(i)}$ as a, and the distance between 0 and $p^{(j)}$ as b. For the obtuse triangle, it holds that $a^2 + b^2 < c^2$. Clearly it also holds that $a < c$, where we can assume without loss of generality (wlog) that $a > b$. Hence we can lower bound the distance between $p^{(i)}$ and $p^{(j)}$ by the longest distance from the origin of either $p^{(i)}$ or $p^{(j)}$. We complete the inequality by observing that $\|\beta_1\|^2 = 1/(\|\alpha_1\|^2) \geqslant 1$, since $\|\alpha_1\|^2 \leqslant 1$, where we know that $\beta_1 = 1/\alpha_1$ from the proof of Theorem 3.2.

Approximation guarantees of spectral clustering on digraphs Now we are ready to show why spectral clustering works on digraphs for $k = 3$. Following the analysis in [46], we map every vertex $u \in V$ to d_u points in $\mathbb{C}$ using the embedding defined in (2.2). Using this, we can bound the overlap between clusters that are returned by a k-means algorithm. First, we define the optimal clustering cost Δ for $k = 3$ as

$$\Delta \triangleq \min_{\text{partition} S_1, S_2, S_3} \text{COST}(S_1, S_2, S_3),$$

where we defined COST in (2.3). We can immediately derive an upper bound on Δ. For technical reasons each point is counted d_u times, and this gives us that

$$\Delta \leqslant \sum_{i=1}^{3} \sum_{u \in S_i} d_u \left\| F(u) - p^{(i)} \right\|^2 \leqslant \frac{1}{\xi - 1}, \tag{3.5}$$

where the last inequality follows by Lemma 3.4. Let APT denote the approximation ratio of a k-means algorithm that outputs partition A_1, A_2, A_3. From equation (3.5) we have that $\text{COST}(A_1, A_2, A_3) \leqslant (\text{APT})/(\xi - 1)$.

This upper bound on the cost of the optimal clustering is good enough to prove that the output A_1, A_2, A_3 is close to the optimal clustering S_1, S_2, S_3. More formally, we show that under a well-chosen permutation $\sigma : \{1,2,3\} \to \{1,2,3\}$ the symmetric difference between A_i and $S_{\sigma(i)}$ is small, where we recall that the symmetric difference between sets X and Y is defined as $X \triangle Y \triangleq (X \setminus Y) \cup (Y \setminus X)$. We will prove this by using the following contradiction: if it is always possible to pick a set A_i with a high symmetric difference to its corresponding $S_{\sigma(i)}$, we show that the value of COST will be high, which contradicts the upper bound of $\text{COST}(A_1, A_2, A_3) \leqslant \text{APT}/(\xi - 1)$. To prove Theorem 3.3, we first state a technical lemma which will we will prove later (this lemma is similar to Lemma 4.5 in [46]).

Lemma 3.5. *Let A_1, A_2, A_3 be a three way partition of V. Suppose that for every permutation of the indices $\sigma : \{1,2,3\} \to \{1,2,3\}$, there exists i such that* $\text{vol}(A_i \triangle$

$S_{\sigma(i)}) \geqslant 2 \cdot \varepsilon \cdot \mathrm{vol}(S_{\sigma(i)})$ *for* $\varepsilon \geqslant 16 \cdot c/(\xi - 1)$*, where* $c \geqslant 1$ *is a constant, then* $\mathit{COST}(A_1, A_2, A_3) \geqslant c/(\xi - 1)$.

Proof of Theorem 3.3. Let A_1, A_2, A_3 be a three way partition that achieves an approximation ratio APT, and we let

$$\varepsilon = \frac{32 \cdot \mathrm{APT}}{(\xi - 1)}.$$

We show that there exists a permutation σ of the indices such that

$$\mathrm{vol}(A_i \triangle S_{\sigma(i)}) \leqslant \varepsilon \cdot \mathrm{vol}(S_{\sigma(i)}) \qquad 1 \leqslant i \leqslant 3.$$

Assume by contradiction that for all permutations σ there is an $i \in \{1,2,3\}$ such that

$$\mathrm{vol}(A_i \triangle S_{\sigma(i)}) > \varepsilon \cdot \mathrm{vol}(S_{\sigma(i)}).$$

This implies by Lemma 3.5 , where we set $c = \mathrm{APT}$ that

$$\mathrm{COST}(A_1, A_2, A_3) \geqslant \mathrm{APT}/(\xi - 1),$$

which contradicts the upper bound of $\mathrm{COST}(A_1, A_2, A_3) \leqslant (\mathrm{APT})/(\xi - 1)$, and thus we prove the theorem. □

Now it remains to prove Lemma 3.5. Suppose by contradiction there is an optimal cluster S_j which is different from every found cluster A_ℓ. Then there must be a cluster cluster A_i with significant overlap between two different optimal clusters S_j and $S_{j'}$. However, we showed in the previous paragraph that any two clusters are far away from each other, i.e., (3.4). This means $\mathrm{COST}(A_1, A_2, A_3)$ will be high, leading to a contradiction. We will use Lemma 4.6 from [46], (omitted to Appendix A.3).

Proof of Lemma 3.5. We consider the case when part 1 of Lemma A.1 holds:

$$\mathrm{vol}(A_i \cap S_{i_1}) \geqslant \varepsilon_i \min\{\mathrm{vol}(S_{i_1}, \mathrm{vol}(S_{i_2}))\}$$
$$\mathrm{vol}(A_i \cap S_{i_2}) \geqslant \varepsilon_i \min\{\mathrm{vol}(S_{i_1}, \mathrm{vol}(S_{i_2}))\}$$

for some $\varepsilon \geqslant 0$ and $\sum_{i=1} \varepsilon_i \geqslant \varepsilon$. Let c_i be the center of A_i. We assume wlog that $\left\|c_i - p^{(i_1)}\right\| \geqslant \left\|c_i - p^{(i_2)}\right\|$, which implies $\left\|c_i - p^{(i_1)}\right\| \geqslant \left\|p^{(i_1)} - p^{(i_2)}\right\|/2$. However, points in $B_i = A_i \cap S_{i_1}$ are far away from c_i, see Figure 3.2 for illustration. By only looking at the contribution of points in the B_i's, we can lower bound the value of $\mathrm{COST}(A_1, A_2, A_3)$. Recall from Lemma 3.4 that the sum of squared-distances between points in B_i and $p^{(i_1)}$ is at most $1/(\xi - 1)$, while the distance between $p^{(i_1)}$ and $p^{(i_2)}$ is large. Hence, we have

$$\mathrm{COST}(A_1, A_2, A_3) = \sum_{i=1}^{3} \sum_{u \in A_i} d_u \|F(u) - c_i\|^2 \geqslant \sum_{i=1}^{3} \sum_{u \in B_i} d_u \|F(u) - c_i\|^2.$$

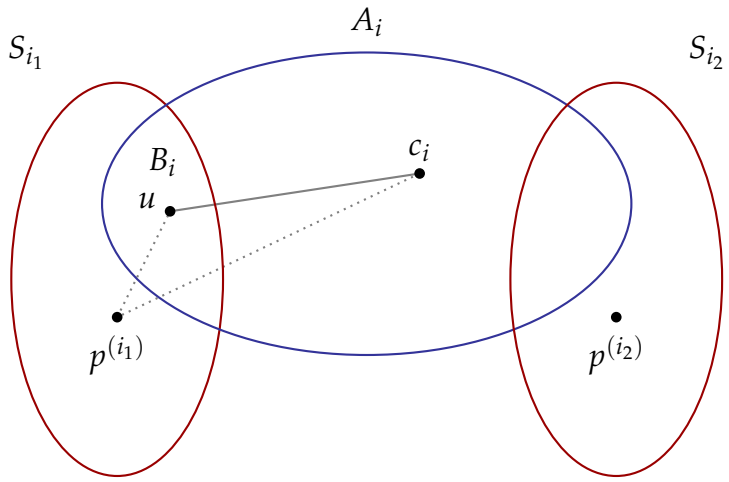

Figure 3.2: The fact that $\left\|p^{(i_1)} - c_i\right\| \geqslant \left\|p^{(i_2)} - c_i\right\|$ can be used to lower bound the value of COST function by only looking at the contribution of points $u \in B_i = S_{i_1} \cap A_i$ for all $i = 1,2,3$. Figure taken from [46].

Recall the inequality $a^2 + b^2 \geqslant (a-b)^2/2$. Applying this inequality, we have that

$$\begin{aligned}
\mathrm{COST}(A_1,A_2,A_3) &\geqslant \sum_{i=1}^{3}\sum_{u\in B_i} d_u \left(\frac{\left\|p^{(i_1)} - c_i\right\|^2}{2} - \left\|F(u) - p^{(i_1)}\right\|^2 \right) \\
&\geqslant \sum_{i=1}^{3}\sum_{u\in B_i} d_u \frac{\left\|p^{(i_1)} - c_i\right\|^2}{2} - \sum_{i=1}^{3}\sum_{u\in B_i} d_u \left\|F(u) - p^{(i_1)}\right\|^2 \\
&\geqslant \sum_{i=1}^{3}\sum_{u\in B_i} d_u \frac{\left\|p^{(i_1)} - c_i\right\|^2}{2} - \frac{1}{\xi - 1} \qquad (3.6) \\
&\geqslant \sum_{i=1}^{3}\sum_{u\in B_i} d_u \frac{\left\|p^{(i_1)} - p^{(i_2)}\right\|^2}{8} - \frac{1}{\xi - 1} \\
&\geqslant \sum_{i=1}^{3} \frac{\mathrm{vol}(B_i)}{8\min\left\{\mathrm{vol}(S_{i_1}),\mathrm{vol}(S_{i_2})\right\}} - \frac{1}{\xi - 1} \\
&\geqslant \sum_{i=1}^{3} \frac{\varepsilon_i \min\left\{\mathrm{vol}(S_{i_1}),\mathrm{vol}(S_{i_2})\right\}}{8\min\left\{\mathrm{vol}(S_{i_1}),\mathrm{vol}(S_{i_2})\right\}} - \frac{1}{\xi - 1} \\
&\geqslant \sum_{i=1}^{3} \frac{\varepsilon_i}{8} - \frac{1}{\xi - 1} \geqslant \frac{\varepsilon}{8} - \frac{1}{\xi - 1} \geqslant \frac{c}{\xi - 1},
\end{aligned}$$

where 3.6 holds because of Theorem 3.2, and the last inequality holds because of the assumption that $\varepsilon \geqslant 16 \cdot c/(\xi - 1)$ and $c \geqslant 1$.

Now suppose that the second part of Lemma A.1 holds. We can repeat the proof from above by setting for any $j \neq i$, $B_j = A_i \cap S_j$, $S_{j_1} = S_\ell$ and $S_{j_2} = S_j$. □

3.3 Distributed Algorithm

Since the runtime of spectral clustering is dominated by the number of edges in the graph, we propose a distributed sparsification algorithm, where edge sampling is only based on the inner and outer degree of vertices. In particular, we introduce an algorithm that satisfies the following theorem:

Theorem 3.6. *There exists an algorithm that, given a digraph $G = (V, E, w)$ with k clusters as input, with probability greater than 0.99, computes a sparsifier $H = (V, F \subset E, \widetilde{w})$ with $|F| = O((1/\lambda_2) \cdot n \log n)$ edges such that the following holds:*

1. *For any $S, T \subset V$ satisfying $\mathrm{NF}(S,T)^2 = \Omega(w(S,T))$, it holds that $\mathrm{NF}_H(S,T) = \Theta(\mathrm{NF}_G(S,T))$.*
2. *$\xi_H = \Omega(\xi_G)$,*

where ξ is defined in (3.2). Moreover, this algorithm can be implemented in $O(1)$ rounds in the distributed setting, and the total information exchanged among all vertices is $O((1/\lambda_2) \cdot n \log n)$ words.

We discuss the $\mathrm{NF}(S,T)^2 = \Omega(w(S,T)$ assumption here in statement 1 of Theorem 3.6. The assumption does not hold for clusters where $w(S,T) \simeq w(T,S)$, and $w(S,T) > 0$ and $w(T,S) > 0$ because then $\mathrm{NF}(S,T) \simeq 0$. We note, however, that it is not essential to guarantee the preservation of NF between these clusters, because we are interested in preserving the *maximum* net flow between clusters. Hence, we consider this a weak assumption.

We also note that our algorithm is similar to the distributed sparsification algorithm proposed by Sun and Zanetti [55] for undirected graphs. Although our algorithm is similar, our analysis for statement 1 of Theorem 3.6 in particular is novel.

Algorithm Description The algorithm to sparsify digraphs works as follows: Every vertex u in the graph checks each of its outgoing edges $e = (u,v)$ and incoming edges $e = (v,u)$, and samples each outgoing edge with probability

$$p_u(v) \triangleq \min\left\{ w(u,v) \cdot \frac{\alpha \cdot \log n}{\lambda_2 \cdot d_u^{\mathrm{out}}}, 1 \right\},$$

and each incoming edge with probability

$$p_v(u) \triangleq \min\left\{ w(v,u) \cdot \frac{\alpha \cdot \log n}{\lambda_2 \cdot d_u^{\mathrm{in}}}, 1 \right\}.$$

for a constant $\alpha \in \mathbb{R}_{\geqslant 0}$ which is determined experimentally and where d_u^{out} is the outer degree of vertex u, and d_v^{in} is the inner degree of vertex v. As the algorithm goes through each vertex, it maintains all the sampled edges in a set F. Once all the edges have been checked, the algorithm returns a new weighted graph $G_H = (V, F, w_H)$, where each sampled edge has a new weight

$$w_H(u,v) = \begin{cases} \frac{w(u,v)}{p_u(v)} & \text{if edge } e = (u,v) \text{ is sampled} \\ 0 & \text{otherwise.} \end{cases}$$

We note that we can write down the probability p_e for any edge $e = (u,v)$ to be sampled as

$$p_e = p_u(v) + p_v(u) - p_u(v)p_v(u).$$

Algorithm Analysis Now we analyse the algorithm and prove Theorem 3.6. Specifically, we will prove the following facts: (1) we show that the number of sampled edges is indeed upperbounded by $O((1/\lambda_2) \cdot n \log n)$; (2) we show that the Net Flow NF between any pair of clusters S and T is preserved as long as $\text{NF}(S,T) = \Omega(w(S,T))$. (3) We prove that the top $n-1$ eigenspace of $\mathcal{L}_G$ is preserved in $\mathcal{L}_H$, meaning that $\xi_H = \Omega(\xi_G)$. Proving these three properties suffices to show that the algorithm described above satisfies the requirements in Theorem 3.6.

Proof of Theorem 3.6. We first analyse the size of F. Since

$$\sum_{u \in V} \sum_{e=(u,v)} w(u,v) \cdot \frac{\alpha \log n}{d_u^{\text{out}} \cdot \lambda_2} = O\left(\frac{n \log n}{\lambda_2}\right),$$

and

$$\sum_{u \in V} \sum_{e=(v,u)} w(v,u) \cdot \frac{\alpha \log n}{d_v^{\text{in}} \cdot \lambda_2} = O\left(\frac{n \log n}{\lambda_2}\right),$$

it holds by Markov inequality that the number of edges $e = (u,v)$ and $e = (v,u)$ with $p_u(v) \geqslant 1$ and $p_v(u) \geqslant 1$ is $O((1/\lambda_2) \cdot n \log n)$. Without loss of generality, we assume that these edges are in F, and in the remaining part of the proof we assume it holds for any edge $e = (u,v)$ that

$$w(u,v) \cdot \frac{\alpha \cdot \log n}{d_u^{\text{out}} \cdot \lambda_2} < 1, \qquad w(v,u) \cdot \frac{\alpha \cdot \log n}{d_u^{\text{in}} \cdot \lambda_2} < 1.$$

Then, the expected number of edges in H equals to

$$\begin{aligned} \sum_{e=(u,v)} p_e \leqslant \sum_{e=(u,v)} p_u(v) + p_v(u) &= \frac{\alpha \cdot \log n}{\lambda_2} \sum_{e=(u,v)} \left(\frac{w(u,v)}{d_u^{\text{out}}} + \frac{w(v,u)}{d_v^{\text{in}}} \right) \\ &= O\left(\frac{n \log n}{\lambda_2}\right), \end{aligned}$$

and thus by Markov's inequality we have that with constant probability the number of sampled edges $|F| = O((1/\lambda_2) \cdot n \log n)$. This implies that the total information exchanged among all vertices is lowerbounded $|F| = O((1/\lambda_2) \cdot n \log n)$. The $O(1)$ rounds needed for the algorithm is simply by the algorithm description.

Preserving Net Flow Next we show that the algorithm preserves the net flow NF between two disjoint sets of vertices S and T in the sparsified graph H up to a constant factor with high probability. Let $S \subset V$ and $T \subset V$ be two arbitrary non-empty disjoint subsets of vertices. For any two vertices u and v, such that $u \in S$ and $v \in T$, we define random variable Y_e as

$$Y_e = \begin{cases} (w_u(v) - w_v(u))/p_e & \text{with probability } p_e \\ 0 & \text{otherwise,} \end{cases}$$

where $Y_e = \frac{w_u(v)}{p_e}$ with probability p_e if there is an edge $e = (u,v)$, and $Y_e = -\frac{w_v(u)}{p_e}$ if there is an edge (v,u). We also define $Z_{ST} \triangleq \sum_{u \in S, v \in T} Y_e = \mathrm{NF}_H(S,T) = w_H(S,T) - w_H(T,S)$ such that

$$\mathbb{E}[Z_{ST}] = \sum_{\substack{u \in S \\ v \in T}} \mathbb{E}[Y_e] = \sum_{\substack{u \in S \\ v \in T}} p_e \cdot \left(\frac{w(u,v)}{p_e} - \frac{w(v,u)}{p_e} \right) = \mathrm{NF}(S,T).$$

If we look at the second moment, we get

$$\begin{aligned} R_{ST} &= \sum_{\substack{u \in S \\ v \in T}} \mathbb{E}[Y_e^2] = \sum_{\substack{u \in S \\ v \in T}} p_e \cdot \left(\frac{w(u,v)}{p_e} - \frac{w(v,u)}{p_e} \right)^2 \\ &= \sum_{\substack{u \in S \\ v \in T}} p_e \cdot \left(\frac{(w(u,v))^2}{(p_e)^2} + \frac{(w(v,u))^2}{(p_e)^2} - 2 \cdot \frac{w(u,v)w(v,u)}{(p_e)^2} \right) \\ &= \sum_{\substack{u \in S \\ v \in T}} p_e \cdot \left(\frac{(w(u,v))^2}{(p_e)^2} + \frac{(w(v,u))^2}{(p_e)^2} \right) = \sum_{\substack{u \in S \\ v \in T}} \frac{(w(u,v))^2 + (w(v,u))^2}{p_e}, \end{aligned}$$

where the fourth line holds because either one of $w(u,v)$ and $w(v,u)$ is zero, so their product is always zero. Notice that $p_e \geqslant p_u(v)$ and $p_e \geqslant p_v(u)$. By considering that the sum consists of two parts, where one part sums over the directed edges from S to

T, and the other one the directed edges from T to S, we get that

$$\begin{aligned}
R_{ST} &= \sum_{\substack{u\in S\\ v\in T}} \frac{(w(u,v))^2 + (w(v,u))^2}{p_e} \leqslant \sum_{\substack{e=(u,v)\\ u\in S\\ v\in T}} \frac{(w(u,v))^2}{p_u(v)} + \sum_{\substack{e=(v,u)\\ u\in S\\ v\in T}} \frac{(w(v,u))^2}{p_v(u)} \\
&= \sum_{\substack{e=(u,v)\\ u\in S\\ v\in T}} \frac{(w(u,v))^2 d_u^{\text{out}}}{w(u,v)\cdot\alpha\cdot\log n} + \sum_{\substack{e=(v,u)\\ u\in S\\ v\in T}} \frac{(w(v,u))^2 d_u^{\text{in}}}{w(v,u)\cdot\alpha\cdot\log n} \\
&= \frac{1}{\alpha\cdot\log n}\left(\sum_{\substack{e=(u,v)\\ u\in S\\ v\in T}} w(u,v)d_u^{\text{out}} + \sum_{\substack{e=(v,u)\\ u\in S\\ v\in T}} w(v,u)d_u^{\text{in}}\right) \\
&\leqslant \frac{\Delta_S}{\alpha\cdot\log n}(w(S,T)+w(T,S)),
\end{aligned}$$

where Δ_S is the maximum in or out degree of all the vertices in S. Also notice that for any edge we get that

$$0 \leqslant \left|\frac{w(u,v)-w(v,u)}{p_e}\right| = \left|\frac{w(u,v)d_u^{\text{out}}}{w(u,v)\cdot\alpha\cdot\log n}\right| \leqslant \frac{\Delta_S}{\alpha\cdot\log n}$$

if there is an edge (u,v), and

$$0 \leqslant \left|\frac{w(u,v)-w(v,u)}{p_e}\right| = \left|\frac{-w(v,u)d_u^{\text{in}}}{w(v,u)\cdot\alpha\cdot\log n}\right| \leqslant \frac{\Delta_S}{\alpha\cdot\log n}$$

if there is an edge (v,u). Now we can apply Bernstein's Inequality (Lemma A.2) to show that with high probability we preserve the edge differences between sets of vertices in the sparsified graph. We set the parameters of Bernstein's inequality such that $M = \Delta_S/(\alpha\cdot\log n)$, $X_i = Y_e$, $X = Z_{ST}$ and $R = R_{ST}$, and $t = \text{NF}(S,T)$ and we get

$$\begin{aligned}
&\mathbb{P}\left[|\text{NF}_H(S,T) - \text{NF}(S,T)| \geqslant \frac{1}{2}\text{NF}(S,T)\right] = \mathbb{P}\left[|Z_{ST} - \mathbb{E}[Z_{ST}]| \geqslant \frac{1}{2}\mathbb{E}[Z_{ST}]\right] \\
\leqslant& 2\exp\left(-\frac{\frac{1}{8}\cdot(\text{NF}(S,T))^2}{\frac{\Delta_S}{\alpha\cdot\log n}\left((w(S,T)+w(T,S)) + \frac{1}{6}\text{NF}(S,T)\right)}\right) \\
\leqslant& 2\exp\left(-\frac{\alpha\cdot\log n}{8\Delta_S}\cdot\frac{(\text{NF}(S,T))^2}{w(S,T)+w(T,S)+w(S,T)-w(T,S)}\right) \\
=& 2\exp\left(-\frac{\alpha\cdot\log n}{16\Delta_S}\cdot\frac{(\text{NF}(S,T))^2}{w(S,T)}\right) = o\left(\frac{1}{n}\right),
\end{aligned}$$

where the last equality follows from the assumption that $\text{NF}(S,T)^2 = \Omega(w(S,T))$ Thus we have shown that $\text{NF}_H(S,T) = \Theta(\text{NF}_G(S,T))$. We note that the total flow

between clusters, $w(S,T) + w(T,S)$, is preserved with high probability in an almost equivalent proof, but we can guarantee preservation for all cases without any extra assumption. This means we preserve the normalised cut value as well.

Preserving Top $(n-1)$ Eigenspace Given a directed graph $G = (V,E)$, we define the incidence matrix $B \in \mathbb{C}^{m\times n}$. The rows of B are indexed by the edges of G, and the columns of G are indexed by the vertices of G. For any edge $e \in E$, and any vertex $v \in V$, we let

$$B_{e,v} \triangleq \begin{cases} \frac{1+i}{\sqrt{2}} & \text{if } v \text{ is } e\text{'s head} \\ \frac{1-i}{\sqrt{2}} & \text{if } v \text{ is } e\text{'s tail} \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

We also define a diagonal weight matrix $W \in \mathbb{R}^{m\times m}$ for the edges, where $W_{e,e} = w(e)$.

Lemma 3.7. *Given a directed graph $G = (V,E)$, where the Laplacian L is constructed using the Hermitian adjacency matrix, it holds that $L = B^{\dagger}WB$, where B is defined in (3.7) and W is the diagonal weight matrix of edges.*

We refer to Appendix A.6 for the proof of this lemma.

Observe that showing that $\xi_H = \Omega(\xi_G)$ is equivalent to showing that the top $n-1$ eigenspace of $\mathcal{L}_G$ is preserved in $\mathcal{L}_H$. Using our new Lemma 3.7, we can use a near identical proof to that of Sun and Zanetti, who prove the preservation of the top $(n-k)$ eigenspace for undirected graphs using a similar sparsification algorithm [55]. The difference in our proof compared to theirs is the use of complex eigenvectors instead of real ones. We omit our proof from the main text. However, for completeness we have included it in Appendix A.5 This completes the proof of Theorem 3.6. □

3.4 Further Discussion

In this section we show experimentally that the normalised Laplacian of a digraph with $k = 3$ clusters has a gap between λ_1 and λ_2. This means that if there is a gap between λ_1 and λ_2, there exists a good three-way partition of the digraph which is encoded by the first eigenvector. We set up the experiment as follows: we generate a fully connected digraph $G = (V,E)$, with three clusters A, B, C, where $|A| = |B| = |C| = n/3 = 100$ and $n = 300$. Within each cluster, the direction of an edge is chosen uniformly at random. For edges between A and B, we orient all the edges from A to B with probability q. For edges between B and C, we orient all the edges from B

to C with probability q. For edges between C and A, we orient all the edges from C to A with probability q. This gives us a cyclic graph cluster structure, similar to the illustration in Figure 3.1. We then compute the bottom 5 eigenvalues for values of $q = 0.5, 0.6, 0.7, 0.8, 0.9,$ and 1. These values can be found in Table 3.1. We can observe in the table that as q tends to 1, the gap between λ_1 and λ_2 increases significantly, whereas the gap between the other eigenvalues does not change. This indicates that the gap between λ_1 and λ_2 points to the existence of a 3-way cluster structure in the graph, and therefore provides some experimental evidence on why we can assume a lower bound on the gap ξ.

	λ_1	λ_2	λ_3	λ_4	λ_5
$q = 0.5$	0.89	0.89	0.89	0.89	0.89
$q = 0.6$	0.86	0.89	0.89	0.89	0.89
$q = 0.7$	0.75	0.89	0.9	0.9	0.9
$q = 0.8$	0.64	0.9	0.9	0.91	0.91
$q = 0.9$	0.53	0.91	0.92	0.92	0.92
$q = 1$	0.42	0.92	0.92	0.92	0.92

Table 3.1: Value of $\lambda_1, \cdots, \lambda_5$ for the Cyclic digraph 3.1 with $k = 3$, $n = 300$, and each cluster containing 100 vertices. We report the bottom 5 eigenvalues for $0.5, 0.6, 0.7, 0.8, 0.9,$ and 1.

Chapter 4

Experimental Analysis: UN Global Comtrade

In this chapter we provide the experimental results of spectral clustering on the UN Comtrade Database [42]. The UN Comtrade database contains the import and export amounts of 97 commodities between 247 countries and territories. Due to the directed nature of the data, we consider this dataset as an ideal testbed for analysis of spectral clustering on digraphs.

This chapter is organised as follows: In Section 4.1 we provide details on data-collection and what information is contained in the data. In Section 4.2 we introduce a new objective function to measure the quality of clustering for digraphs, and discuss why it is a good objective function based on literature of economics and econometrics. In Section 4.3 we analyse the simplest case of clustering where $k = 2$ on the international trade of wood. In Section 4.4 we look at $k > 2$, and show that spectral clustering using the Hermitian adjacency matrix finds meaningful patterns in the international oil trade network. Section 4.5 contains insights into the effect of a changing value of k for this dataset. In Section 4.6 we experimentally test our distributed sparsification algorithm we proposed in Section 3.3.

The hardware that was used to run the experiments was an ASUS ZenBook Pro UX501VW with an Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz with 12GB of RAM. We would also like to note that we have made YouTube videos [29, 30, 31, 32] to visualise our many plots. We do plot the most relevant figures in the main text, but to see all the plots see Appendix B.3.

4.1 UN Comtrade Database

We perform clustering experiments on the UN Comtrade Database [42]. This database contains the import-export tradeflow data of 97 specific commodities across $N = 246$ countries and territories over the period 1964 - 2018, where the tradeflow is reported in the current value of USD$. The database can be obtained from the UN Comtrade website.

The API provided by the UN gives a lot of flexibility in which type of data can be selected, as seen in figure 4.1. In header 1, it is possible to specify the *product type* to either trade in goods (e.g. oil, wood, appliances) or services (e.g financial services, construction services). In header 2 the *classification code* can be selected, which we set to the Harmonised System (HS). The HS categorises goods according to a 6 number classification code (e.g. 060240, where the first 2 digits "06" encode "plants", the second 2 digits "02" encode "alive", and the last 2 digits "40" code for "roses"). The *reporting* countries and *partner* countries can be specified here as well, where the reporting country reports about its own reported tradeflow with partner countries. The settings we used to download the data for our experiments were Goods on an annual frequency, the HS code as reported, over the period 1988-2017, with all reporting and all partner countries, all trade flows and all HS commodity codes. The total size of the data in zipped files was 99.8GB, where each file contained around 20,000,000 lines of text.

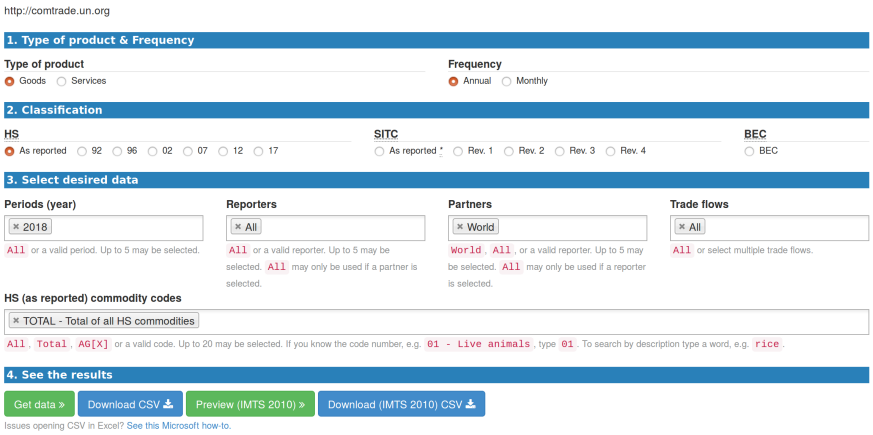

Figure 4.1: Screenshot of the UN comtrade free download API [42].

Preprocessing Data Now we specify how we construct the adjacency matrix of the digraph. The database contains the data of *reporting* countries, who report the import and export of each of the 97 specific commodities with their respective *partner* coun-

tries. So for pairs of countries i and j, where i is the reporting country and j is the partner country, the database contains the amount that country i imports from country j for a specific commodity, and also the amount i exports to j. There are several cases where countries i and j report different trading amounts with each other. Usually, the larger value is considered more accurate and is therefore used instead of the average for example [19]. To construct the digraph of the world trade network and its corresponding adjacency matrix, we fill in each entry of the adjacency matrix M^c for commodity c as follows: for each pair of countries i and j, we compute $d_{ij}^c = e_{ij}^c - e_{ji}^c$, where e_{ij}^c is the amount country i exports to country j for commodity c. If $d_{ij}^c > 0$, we set $M_{ij}^c = d_{ij}^c$ and $M_{ji}^c = 0$. If $d_{ij}^c < 0$ (and thus $d_{ji}^c > 0$), we set $M_{ji} = d_{ji}^c$ and $M_{ij}^c = 0$. In words, we create a directed edge from i to j if the export of a specific commodity c from country i to j is greater than the export from j to i, which indicates that there is a greater *net flow* of the product from country i to j. We create a separate digraph for each year from $1988 - 2017$, and perform spectral clustering experiments on the graph of each year. In particular, we will use 6 different matrices as input to spectral clustering that are normalised or unnormalised matrices of the forms we discussed in Section 2.2. *Herm* will be the Hermitian adjacency matrix A, *Herm-RW* will be the random walk normalised version of *Herm*. *DISG-L* and *DISG-R* respectively use the top eigenvalues and eigenvectors of the adjacency matrices $M^\intercal M$ and $MM^\intercal$. *BI-SYM* will use the adjacency matrix $M^\intercal M + MM^\intercal$, and *DD-SYM* a normalised version of *BI-SYM*. From now on throughout the rest of the thesis we will refer to these matrices with these names.

All the programming and implementation was done in Python 2.7 [47], where we use the NumPy and Pandas libraries [39, 44] to do the data loading, processing and linear algebraic computations. To make the plots of the clusters and to perform spectral clustering, we use Matplotlib, GeoPandas and sci-kit learn. To implement spectral clustering, we build code on top off an existing repository [41]. To see the full code implementation of the experiments the code attached to the thesis submission.

4.2 Objective Functions

When graph clustering is performed on undirected graphs, the objective is to find sets of vertices that are more closely connected to each other than to vertices outside the set, and therefore the conductance or the normalised cut value are used as an objective function [34, 52]. The conductance between two disjoint sets of vertices in a cluster is

the ratio of edges crossing the two sets to the minimum of the volumes of the cluster. It is the standard metric to measure the quality of the output of a spectral clustering algorithm. However, designing an objective function for a digraph is more involved. We illustrate it with the following example. Suppose we have a fully connected digraph with a large number ($n > 100$) of vertices as the one sketched in Figure 3.1, where each cluster contains approximately $n/3$ vertices. Edge direction within clusters is selected uniformly at random. Edge direction between the red, green and blue clusters are imbalanced. If we symmetrise its adjacency matrix by naively removing directionality ($M' = M^T + M$), we obtain the complete undirected graph, and there is no relevant cluster structure in that graph. However, if we look at the digraph, one could argue there is some form of cluster structure, due to the imbalances in edges. We can see in Figure 3.1 that there is a circular direction to the edges that cross the clusters. All this information is lost if one only considers edge density between clusters.

For this reason we introduce new objective functions to measure the performance of the different algorithms. Since we are working with a trading dataset, we specify our objective functions for trade. The first metric we define is the Trade Flow (TF) between two clusters, which is simply the absolute value of the Net Flow (NF) from our previous chapter. For any two disjoint sets X and Y, we define the Trade Flow between X and Y to be

$$\mathrm{TF}(X,Y) \triangleq |w(X,Y) - w(Y,X)|, \tag{4.1}$$

where $w(X,Y) = \sum_{x\in X, y\in Y} w(x,y)$. The trade flow of different commodities is an important metric in economics and econometrics. Since the 1940s, economists have considered international trade as an interconnected directed network [26]. In fact, there exists a whole field aimed at modelling the trade flow of different commodities using gravity models [9, 38]. These gravity models predict trade flows between countries based on metrics such as their proximity, shared colonial past, and the countries' economic/political characteristics. This emphasises that trade flow is an important relational metric for countries in a trading network, with many underlying causes that guide trading patterns. Trade flow is also used extensively in the network analysis literature as the edge weights in world trade networks to encode the trade relationship between countries. Sometimes the edge weights are normalised based on country populations, GDPs, or total trade [18, 20, 21]. Because of these facts, we decide to use TF as an objective function that measures important relationships between countries or clusters of countries. We do make adjustments to the TF metric. We experimen-

tally observed that the TF between two clusters will be high if two clusters are both very large. This is likely because some countries export and some countries import significantly more than other countries. Having two large clusters with all importing and all exporting countries maximizes the TF in a trivial way. We therefore define the normalised trade flow as

$$\mathrm{TF}^{\mathrm{norm}}(X,Y) \triangleq \frac{|w(X,Y) - w(Y,X)|}{|X| + |Y|}.$$

By dividing the TF by the total number of countries in both clusters, we decrease the value of TF if there are many countries. Hence we penalise results that consist of large clusters and do not provide meaningful information. The other way of normalising for the trade would be to take the ratio of TF with respect to the total trade between two countries. In this manner we introduce the Trade Imbalance ratio (TI) [15] between two disjoint sets of vertices X and Y as:

$$\mathrm{TI}(X,Y) \triangleq \frac{1}{2}\left|\frac{w(X,Y) - w(Y,X)}{w(X,Y) + w(Y,X)}\right|.$$

Notice that if two sets of vertices have edges crossing between them that are completely imbalanced, then $\mathrm{TI} = 0.5$, and if they are perfectly balanced then $\mathrm{TI} = 0$. In the case of TI, it can occur that an algorithm might find two clusters, one consisting of the country i and one consisting of the country j. If country i exports some commodity c to country j, and country j doesn't export anything to country i, then $\mathrm{TI} = 0.5$. Although the objective function is maximised, the information gained from these two clusters is not meaningful. To avoid TI scores that are high but consist of clusters X and Y that are small, we also introduce

$$\mathrm{TI}^{\mathrm{size}}(X,Y) \triangleq \mathrm{TI}(X,Y) \cdot \min\{|X|,|Y|\}.$$

By normalising by the size smallest cluster, we penalise cluster pairs where one cluster consists of very few countries.

Throughout the experiments in the next sections, we compute the $\mathrm{TF}^{\mathrm{norm}}$ and $\mathrm{TI}^{\mathrm{size}}$ for every pair of clusters that are found by our algorithms. We sort these pairs based on their $\mathrm{TF}^{\mathrm{norm}}$ and $\mathrm{TI}^{\mathrm{size}}$ scores. The highest pair in that ordering is called the top-1 pair, and the k^{th} element is called the top-k pair. For ease of discussion, we consider the top-1 pair to be the most important pair in the network, since these two clusters trade most with each other with respect to the cluster size.

4.3 Case study for $k = 2$: Wood Trade Network

To start the experimental analysis we first show results on the simplest case of clustering, where $k = 2$. We compare the performance of the different algorithms on clustering the world trade graph for wood (HS classification code 44) on $k = 2$ clusters. In the undirected case, computing the optimal conductance of a graph when clustering a graph into 2 clusters is NP-Hard [53], and the best approximation algorithm can give a $O\left(\sqrt{\log n}\right)$ approximation [7]. Assuming that the small set conjecture is true, it is also known that a constant-factor approximation algorithm does not exist for finding the sparsest cut in an undirected graph [48]. In the directed case, however, there exists a polynomial time algorithm that finds the optimal solution which maximises the Trade Flow TF between two disjoint sets. The algorithm is very simple. It first computes the net flow through each vertex by computing the difference between incoming and outgoing edge weights. The algorithm then returns a partition of two disjoint sets of vertices, where one partition contains all the vertices with negative flow, and the other all the vertices with positive flow. These two clusters are the two clusters that have the maximum TF between them. The time complexity of the algorithm is $O(m)$, since for each vertex u, we compute the difference between its in and outgoing edge weights, which computationally costs $O(d_u)$, where d_u is the total degree of vertex u. The total computational time cost is $O(\sum_{u \in V} \cdot d_u) = O(2m) = O(m)$. We therefore have a linear time algorithm in the case of $k = 2$ to compute the max flow between two sets.

The proof that this algorithm indeed outputs the clustering with maximum flow is straightforward. To sketch the proof, notice that for any two way partitioning, one can always increase the flow between the two partitions by either removing vertices with positive trade flow from the cluster that has negative trade flow, or adding vertices with negative trade flow. In this way, one can show that to construct the partitioning with maximum trade flow one cluster must contain all the vertices with negative net flow and the other with positive net flow. We refer to Appendix B.1 in the appendix for a formal proof.

Now we can very efficiently compute the optimal clustering using the algorithm described for the case $k = 2$, and thus we can compare the true performance of the different spectral techniques. We run the spectral clustering algorithm using the different adjacency matrices, and see how close they come to the optimum. We do this to show which spectral algorithm performs best when $k = 2$, and from there we generalise to

say that this might also be the case for $k > 2$, where we cannot efficiently compute the optimum solution.

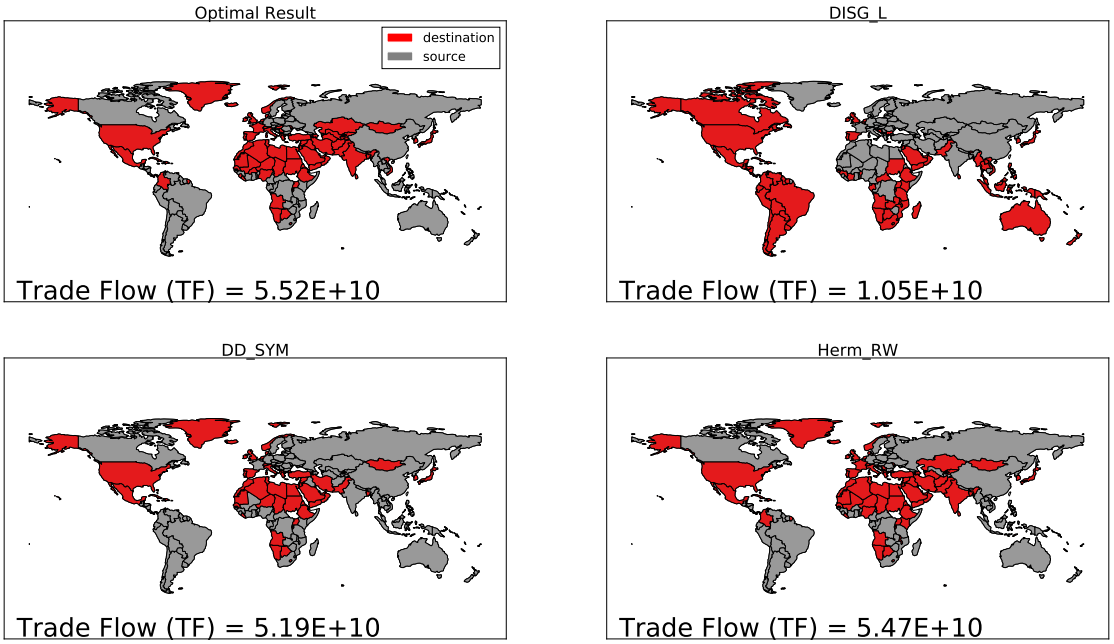

Figure 4.2: Comparison of 4 different clustering results on the world wood trade network of 2005. The simple max flow algorithm (top left). Matrices used for spectral clustering are the DISG-L matrix (top right), the DD-SYM matrix (bottom left) and the Herm-RW matrix (bottom right). Red cluster is destination cluster (more ingoing weight), and grey cluster is source cluster (more outgoing weight). Trade flow between two clusters reported in bottom left of subfigures.

In Figure 4.2 we plot the clustering for $k = 2$ results on the world wood trade network in 2005. Red countries belong to the cluster that import more wood than it exports, and grey countries belong to the cluster that export more wood than they import. We excluded three matrices, DISG-R, BI-SYM and Herm. This is because these matrices perform worst, and to save space we only plot DISG-L, Herm-RW and DD-SYM Herm. We included the full results in the appendices (Appendix B.2). In this figure we can see that the Herm-RW algorithm obtains a TF of $5,4705 \cdot 10^{10}$ between both clusters that is very close to the optimum ($5,5186 \cdot 10^{10}$). From the plots we can also visually see that the clusters between the optimal result (top left) is nearly identical to the Herm-RW method (bottom right). All other matrices obtain a lower score. We remark that all these tested spectral algorithms have similar relative performance in other years, and our algorithm beats the other algorithms in 23 out of 30 years. This

result indicates that the Herm-RW algorithm achieves an almost optimal solution for the case when $k = 2$. Next, we compare performance of the algorithms when $k > 2$. Note that for $k > 2$ the problem becomes NP-hard [23].

4.4 Oil World Trade Network ($k > 2$)

Instead of analysing wood trade, we shift our trade network to the one of mineral fuels. We construct a digraph of the International Oil Trade Network (IOTN) using data from the UN Comtrade Database, where we use the HS code 27. The IOTN is an essential element of the world economy, which is highlighted by the fact that the highest traded commodity trade in 2017 was crude oil [8]. The international trade dynamics of oil are complex, due to changing energy demand by countries, excessive availability in some countries, and lack of oil in other countries. This makes oil trade an important factor in geopolitics (e.g., 2003 Iraq War). Research into the international oil trade network has therefore become an important topic in both policy making and academic research [10, 16, 17]. Usually, the international trade of mineral fuels is studied by the fields international trade theory and econometrics, where the emphasis is usually on price fluctuations [3, 5], market behaviour & risk analysis, [28] and policy making [4]. While these studies provide complex in-depth analyses, they often focus on a small subset of countries that are key players in the international oil trade community, rather than on larger patterns or trends.

Recent work in network analysis has looked at the IOTN from a broader perspective by considering the evolution of cluster dynamics over time, and the trading hierarchy between countries. These complex network analyses have given more insight into the dynamic and topological structure of the IOTN [6, 60]. However, to the best of our knowledge there has been no work on the world trade network in the spectral clustering community. We therefore performed spectral clustering experiments on the IOTN, to see what type of patterns the different spectral clustering algorithms uncover.

First, we perform experiments on the IOTN of 2017, where we compute the TF^{norm} and TI^{size} for each pair of clusters that we obtain from the spectral clustering algorithm using the 6 different input matrices . We then plot the top 10 pairs for both TF^{norm} and TI^{size} for all the algorithms and for varying values of k, where we set $k = 5$, $k = 10$, and $k = 20$ (Figure 4.3).

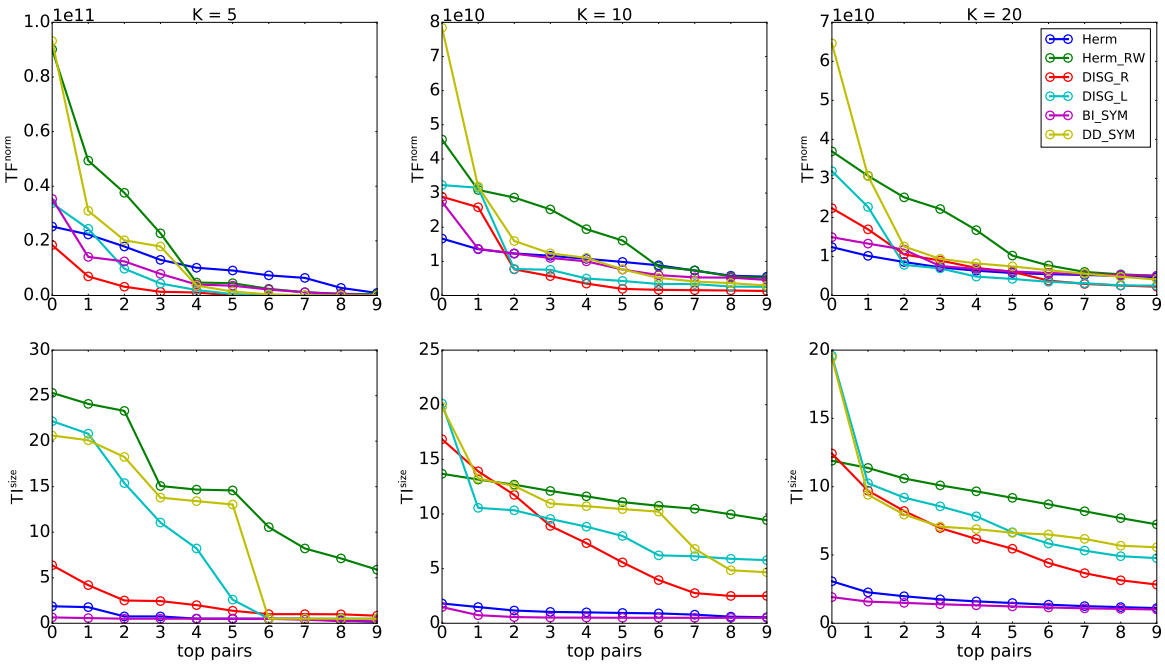

Figure 4.3: Top 10 pairs for TI^{size} and TF^{norm} for the six different input matrices for spectral clustering. Results plotted where $k = 5, 10$ and 20 . Results averaged over 100 runs. The Herm-RW method performs best in all cases, with occasional exception of the top-1 pair. This indicates that the Herm-RW method finds more balanced clusters.

From the results we can see that except for the top-1 pair, the Herm-RW method performs best in almost all the plots over the other top pairs. If we look at the top-1 pairs, we see that the DD-SYM or DISG-L occasionally perform better than Herm-RW. This is likely due to having a top pair that consists of two large clusters of countries, since the top-2 and top-3 pair drop drastically for those two matrices. This signifies that the Herm-RW method finds more balanced top pairs of clusters, which is also an observation found by [15] on other datasets.

Cluster Visualisations In the next section we show that our Herm-RW method not only performs best on the objective functions as seen in Figure 4.3, but that the cluster visualisations also show that the Herm-RW method can extract meaningful patterns that correspond to literature in economics and network analysis. We visualise the pairs over the period 2001-2003, which corresponds to the Venezuelan oil strike and the second Persian Gulf war, and over the period 2007-2009, which corresponds to the economic collapse of 2008.

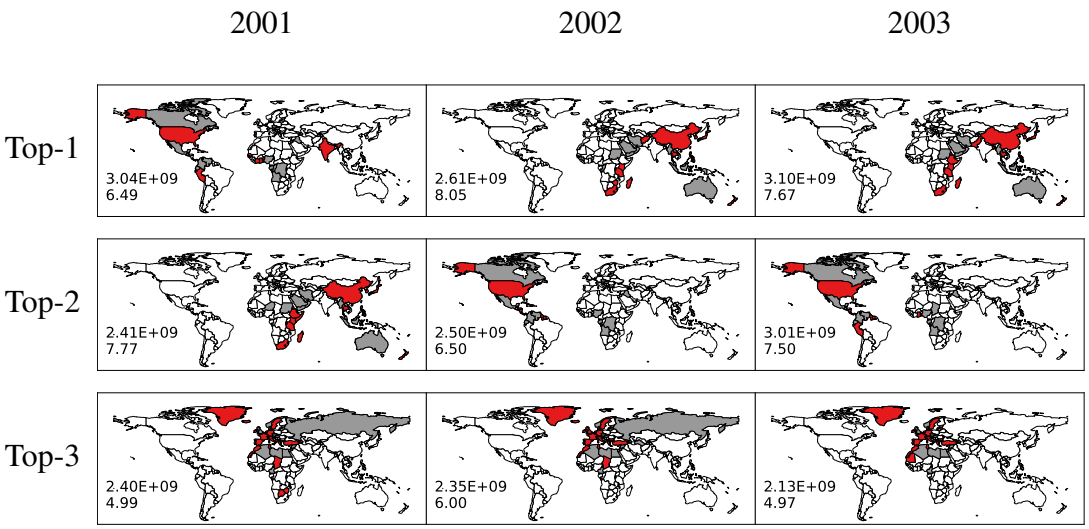

Figure 4.4: Top 3 clusters for the results of performing spectral clustering using Herm-RW method on the IOTN in 2001, 2002 and 2003. Top pairs are sorted on TF^{norm} score. The first, second and third row correspond to the top-1, top-2, and top-3 pair respectively. Column 1, 2, and 3 correspond to the years 2001, 2002, and 2003 respectively. Grey clusters are the source clusters with more outgoing trade flow, red clusters are destination clusters with more incoming trade flow. Bottom left of each plot reports the TF^{norm} (above) and TI^{size} (below) scores

Venezuelan oil strike and Iraq war The year 2003 was a turbulent one in the oil industry. The two main oil-related events that year were the Venezuelan oil strike, and the second Persian Gulf war. Both these events led to the decrease in oil barrel production by 5.4 million of barrels per day [25]. Although compared to other events in the 21st century this year did not have as much of an effect on global oil supply, it has been argued that this year does belong to the post-war oil shocks [27]. Since this was such a tumultuous year, we decide to observe the top pairs of clusters throughout this period, to see if there are any notable changes in oil trading communities as was found by [6, 60]. In Figure 4.4 we can see that our Herm-RW method uncovers a change in the cluster dynamics over the period 2001-2003. After 2001, the top-1 cluster containing the USA, Iraq and Venezuela, moves down to the top-2 cluster in 2002. On the other hand, the top-2 pair from 2001 moves up one place into 2002. The top-3 pair doesn't change from 2001 to 2003. From 2002 to 2003 we do not see much of a change in the top 2 pairs. This change in top 3 cluster ordering from 2001 to 2003 suggests that our Herm-RW method uncovers a pattern in an unsupervised manner that is in accordance with the economic literature on world oil trade.

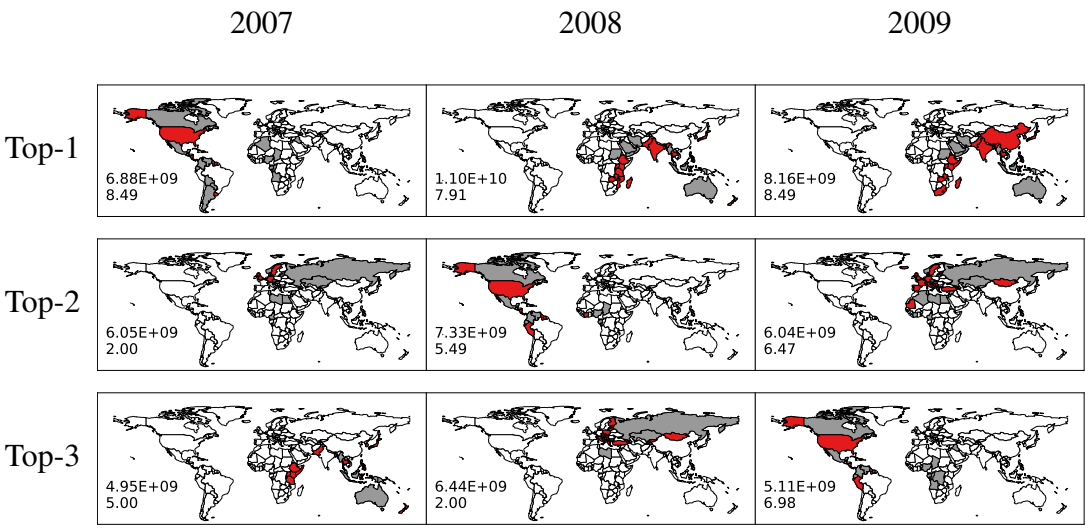

Figure 4.5: Top 3 clusters for the results of performing spectral clustering using Herm-RW method on the IOTN in 2007, 2008 and 2009. Top pairs are sorted on TF^{norm} score. The first, second and third row correspond to the top-1, top-2, and top-3 pair respectively. Column 1, 2, and 3 correspond to the years 2007, 2008, and 2009 respectively. Grey clusters are the source clusters with more outgoing trade flow, red clusters are destination clusters with more incoming trade flow. Bottom left of each plot reports the TF^{norm} (above) and TI^{size} (below) scores

Oil Spike of 2008 In 2007/2008 the world economy experienced the largest post World War II oil shock [24]. In Figure 4.6 we can see a dramatic shift in the price of oil per barrel after 2008, which was accompanied by the last big economic collapse. In the economics literature, it is agreed that the main reason for this sharp price spike in 2008 has to do with an increasing demand for crude oil due to rapidly growing developing countries (i.e. China and India), while supply stagnated over the period 2004-2008 [25]. As other literature in network analysis observed a change in cluster stability during this period [6, 60], we decide to observe the top 3 pairs of clusters around this period. In Figure 4.5 we see that the top-1 pair from 2007 moves down one place each consecutive year, indicating a drastic change in trading hierarchy. The top-2 pair moves down one place and then back up throughout the three years. In 2009 we see a new top-1

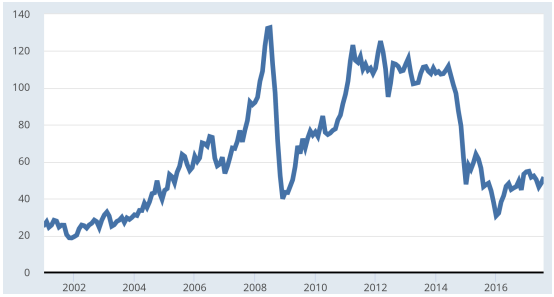

Figure 4.6: Price of crude oil per barrel (USD$) from 2001-2017 [2]

pair appear containing the emerging economies India and China as the main destination countries, indicating a significant increase in oil trade for this pair.

Our method again seems to find a change in trading hierarchy which is in accordance with literature from economics and network analysis as mentioned before. In 2009 we see that China and India belong to the destination cluster in the top-1 pair, whereas they did not appear before 2009. This could be due to the rapid increase in oil demand for these countries. The top-2 pair in 2009 contains Russia as the main source cluster, and Europe as the main destination cluster, which might indicate a higher reliance of Europe on Russian mineral fuels.

Other Adjacency Matrices We have plotted the results for the other matrices in the appendices (Appendix B.3). We decide to only plot the Herm-RW method, since the other methods did not exhibit any significant pattern changes over these period, or the top pair of countries contain too many or too few countries. The patterns found by the DD-SYM method do show some similarities with the Herm-RW method. In the DD-SYM method the pairs of clusters are usually larger, and contain countries from all over the world, while the Herm-RW method usually contains more localised clusters.

4.5 Insights into the Value of k

When performing spectral clustering on undirected graphs, a good method to choose k is to look at the eigenvalues of the Laplacian matrix and find the first large gap in the spectrum $\lambda_1, \ldots, \lambda_n$, and set k to the value where that gap occurs. We have discussed this in more depth in Section 2.3. However, since in directed graphs conductance is not necessarily used to measure the quality of clustering, it is not clear how to choose the right value of k. In this section we provide some insight into the value of k when performing clustering on the UN Comtrade database, and what changes when increasing the size of k. Throughout performing experiments, we noticed that when $k > 15$, the Herm-RW method finds more geographically localised top pairs, whereas other methods tend to find clusters which are more spread geographically. In Figure 4.7 we plot the top-1 pair for wood trade (HS 44) in 2015 with $k = 5$, $k = 10$ and $k = 20$ for the Herm-RW method and DD-SYM method. We see that as k increases, the top pair for the Herm-RW pair becomes more geographically concentrated around the Americas, whereas for the DD-SYM method the top pair keeps a global structure.

An explanation for this might lie in what the different adjacency matrices represent.

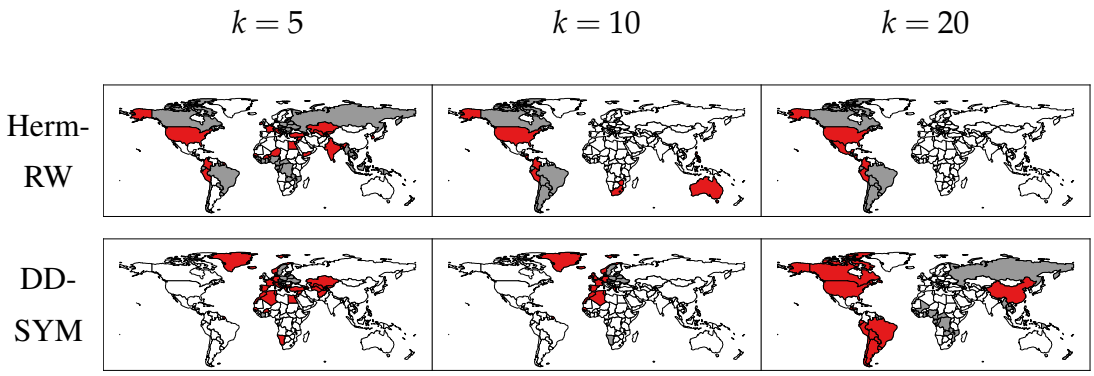

Figure 4.7: Plots of the top-1 pairs sorted by TF^{norm} for the Herm-RW and DD-SYM method, for $k = 5, 10$ and 20. As k increases, we see that the Herm-RW method finds a pair of clusters geographically focused on the Americas. The DD-SYM method maintains a top pair with global structure.

We let M be the standard adjacency matrix, as defined in Section 2.1. Now we can take a look at an arbitrary entry of these matrices indexed by u, v whose closed form can be written as follows.

$$\begin{aligned}(M^T M)_{uv} &= |\{w : (w,u) \text{ and } (w,v)\}| \\ (MM^T)_{uv} &= |\{w : (u,w) \text{ and } (v,w)\}| \\ (MM^T + M^T M)_{uv} &= |\{w : (u,w) \text{ and } (v,w)\}| + |\{w : (w,u) \text{ and } (w,v)\}|.\end{aligned}$$

We can see that $M^T M$ keeps track of the number of common parents of vertices, MM^T keeps track of the common children of vertices, and $MM^T + M^T M$ keeps track of both. Let A be the Hermitian adjacency matrix as defined in (2.1). If we consider the matrix A^2 (which has the same eigenvectors as A and is simpler to investigate) we can write each individual entry of the matrix A^2 as

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

So A keeps track of common children and common parents, but penalises vertices that are a parent to u and a child of v. It is important to note that the matrix A implicitly keeps track of a similar metric as the matrix $MM^T + M^T M$, without performing two expensive matrix multiplications. This makes this new representation not only attractive from a representational aspect, but also a computational one.

The individual entries of A and $M^T M + MM^T$ implicitly keep track of common offspring and children, with the difference being that the Hermitian adjacency matrix

penalises the value if there are vertices in the flow path between u and v. If we consider the UN Comtrade database, this means that the Hermitian adjacency matrix penalises the number of countries w, where u has a net export to w, and w has a net export to v (or vice versa). Two countries (u and v) that are geographically distant might have indirect trade through big economies like the USA or China, since both those countries export and import a lot. This type of trading relation is penalised by the Herm-RW method. On the other hand, countries that are close to each other are more likely to trade directly with each other, a fact that is also used in gravity trading models to predict total trade between pairs of countries [38, 9]. Hence, in the $M^TM + MM^T$ method countries u and v that are distant but trade indirectly with each other through other economies will lie closer to each other in the matrix space than in the Hermitian A matrix space. Informally, this fact might provide an explantion for why the Herm-RW method finds geographically closer top pairs. So depending on what type of clustering is desired, k can be increased to give more locally relevant structure, or decreased to give a more global structure.

4.6 Distributed Sparsification

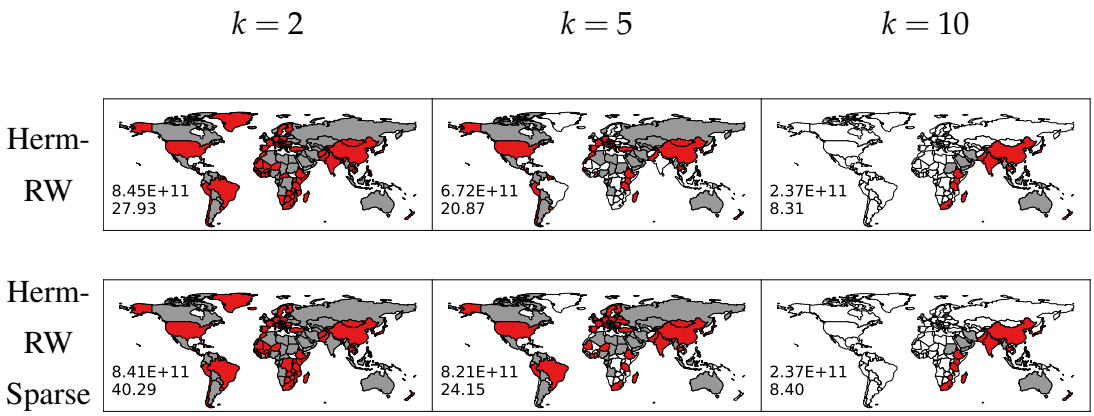

Figure 4.8: Comparison of the non-sparsified (top row) and sparsified (bottom row) clustering results for $k = 2,5$ and 10. For the $k = 2$ sparsifier we sample 8.9% of edges, for $k = 5$ we sample 7.1% and for $k = 10$ we sample 15%. Bottom left of each plot reports the TF^{norm} (above) and TI^{size} (below) scores.

In this section we experimentally test the distributed sparsification algorithm that we proposed in Section 3.3. We perform spectral clustering experiments on the IOTN of 2005. There are two matrices that we use as input for the spectral clustering algo-

rithm. One is the Herm-RW method on the unsparsified graph, and the other is the Herm-RW method on the sparsified the graph using the proposed algorithm. We compare the top-1 cluster for both methods for clustering results where we set $k = 2$, $k = 5$ and $k = 10$. Our results can be found in Figure 4.8.

Visually inspecting the clusters it indeed seems that the top-1 pair is preserved. Furthermore, we can see that the TF^{norm} and TI^{size} scores are preserved with high accuracy in the sparsified graphs. These results indicate that we still preserve the cluster structure and the NF cuts, even though we sample only $10 \sim 15\%$ of the edges in the digraps. This is particularly remarkable since we only have 220 and vertices and $24,089$ edges in the oil trade network of 2005. We do observe that more edges need to be sampled

We also compare the runtime of spectral clustering between the non-sparsified and the sparsified graph. We perform digraph spectral clustering where we set $k = 20$, sample 5% of the edges, and we average the runtime over 50 runs. On average for the non-sparsified method each iteration of spectral clustering takes 0.704 seconds for the unsparsified method, and 0.668 for the sparsified method. This is not a significant difference, but might be attributed to a number of reasons. To compute the eigenvalues and eigenvectors we use the NumPy library, and to perform k-means clustering we use scikit-learn library [44, 45] . It might be the case that these packages do not achieve massive increases in performance for matrices which are only 220×220 dimensional. This is highlighted that the experiments only take around $0.6 \sim 0.7$ seconds to run. In theory, the runtime of our method should perform better once m is increased, and the number of sparsified edges is much lower. It is known that the matrix multiplication Bx, where matrix $B \in \mathbb{R}^{m \times n}$, and vector $x \in \mathbb{R}^n$ takes $O(n + \text{nnz}(B))$, where nnz denotes the number of non zero entries. If the eigenvectors and eigenvalues are computed using the power method, it should be the case that the runtime of a sparser graph is indeed faster.

We remark that in the paper which proposes the same algorithm for undirected graphs by Sun and Zanetti [55], they manage to achieve high classification accuracies while only sampling $0.1 \sim 1\%$ of edges. They are able to do this because the graphs they perform clustering on graphs with thousands of vertices and millions of edges, which thus contain relatively more redundancy.

Chapter 5

Concluding Remarks

In this thesis, we have studied a new method of spectral clustering of digraphs using Hermitian Laplacians, proposed by Cucuringu et al. [15]. The work that has been done can be split into two parallel components: a theoretical component where we analyse the theoretical guarantees of the new method, and an experimental component where we investigate what kind of patterns the new method can extract from the UN Comtrade Database.

Theory We generalise theoretical results for spectral clustering on undirected graphs to digraphs. We introduce a Structure Theorem, which relates the first eigenvector of the normalised Hermitian Laplacian of a digraph to the cyclic flow between $k = 3$ clusters. Using this Structure Theorem, we are able to prove that spectral clustering using Hermitian Laplacians will return a partitioning that is close to the optimal result.

An immediate avenue for future work would be to try and generalise the theoretical results of this thesis to any value of k. However, this is not trivial, since the number of pairs of clusters increases exponentially, meaning that it is not evident how to define indicator vectors that capture the cluster structure of the entire graph. Furthermore, even though the gap assumption holds for $k = 3$, it is not as clear that the gap assumption can be generalised to higher values of k. Another topic for future theoretical work would be to analyse the differences of the guarantees provided by the Structure Theorem and Directed Stochastic Block Models (DSBM). In the case of undirected graphs, it is possible to cluster SBMs using spectral clustering, even though they do not satisfy the gap property of the Structure Theorem [59]. This is because SBMs contain strong regularities, which can be exploited by spectral clustering. Seeing if this translates to the directed case remains an open question, and a particularly interesting one since

there are more patterns in the DSBM to exploit.

We further develop an algorithm that sparsifies digraphs in a distributed manner by only sampling edges based on vertex degrees, and we prove theoretically and show experimentally that this sampling technique preserves the cluster structure of a digraph. In theory, the algorithm should run faster, however, due to the small size of the UN Comtrade Database graphs, experiments cannot confirm that. Future work should run the sparsification algorithm on larger digraphs to thoroughly test whether spectral clustering runs faster on sparsified digraphs using our algorithm.

Experiments To ground our theory in the real world, we perform experiments on the UN Comtrade database, which has never been analysed in the clustering literature before. We define new objective functions, TF^{norm} and TI^{size}, to measure the quality of clustering results, since objective functions based on edge densities between clusters ignore patterns in the digraph that relate to its directional nature. We show that the Hermitian method outperforms other previously proposed methods on both objective functions. Still, these objective functions were specifically designed for the UN Comtrade database. Finding a general objective function that can measure clustering performance of any digraph is still an open and very important question.

Remarkably, we observe that our Hermitian method can extract higher level pattern changes from the data, such as the effect of the Iraq war on world oil trade, and the economic collapse in 2008. Future collaborations with researchers in the social sciences might explore the use of spectral clustering with Hermitian matrices on other types of data.

In the undirected spectral clustering literature, choosing the value of k is aimed at minimising the objective function. What we see in experiments is that increasing the value of k decreases the maximal flow between clusters, but leads to changes in the properties of clusters. In our case, we found that higher values of k output top pairs of countries that are more geographically localised. This raises the pertinent question that perhaps, for clustering digraphs, the value of k is more relevant in influencing clustering characteristics, rather than in maximising an objective based on a spectral gap.

Bibliography

[1] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

[2] U.S. Energy Information Administration. Crude oil prices: Brent - europe. `https://fred.stlouisfed.org/series/DCOILBRENTEU`, Jun 2019.

[3] Bahram Adrangi, Arjun Chatrath, Kanwalroop Kathy Dhanda, and Kambiz Raffiee. Chaos in oil prices? evidence from futures markets. *Energy Economics*, 23(4):405–425, 2001.

[4] Hunt Allcott and Nathan Wozny. Gasoline prices, fuel economy, and the energy paradox. *Review of Economics and Statistics*, 96(5):779–795, 2014.

[5] Haizhong An, Xiangyun Gao, Wei Fang, Yinghui Ding, and Weiqiong Zhong. Research on patterns in the fluctuation of the co-movement between crude oil futures and spot prices: A complex network approach. *Applied energy*, 136:1067–1075, 2014.

[6] Haizhong An, Weiqiong Zhong, Yurong Chen, Huajiao Li, and Xiangyun Gao. Features and evolution of international crude oil trade relationships: A trading-based network analysis. *Energy*, 74:254 – 259, 2014.

[7] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):5, 2009.

[8] Futures Industry Association. Total 2017 volume 25.2 billion contracts, down 0.1% from 2016. `https://fia.org/articles/total-2017-volume-252-billion-contracts-down-01-2016`, Jan 2018. Accessed: 2019-08-14.

[9] Badi H. Baltagi, Peter Egger, and Michael Pfaffermayr. A generalized design for bilateral trade flow models. *Economics Letters*, 80(3):391 – 397, 2003.

[10] Niaz Bashiri Behmiri and José R Pires Manso. Crude oil conservation policy hypothesis in oecd (organisation for economic cooperation and development) countries: A multivariate panel granger causality test. *Energy*, 43(1):253–260, 2012.

[11] U.S. Census Bureau. `https://www.census.gov/population/www/cen2000/ctytoctyflow/`, 2002. Accessed: 2019-08-14.

[12] Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: a survey. *Internet Mathematics*, 3(1):79–127, 2006.

[13] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[14] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *Journal of symbolic computation*, 9(3):251–280, 1990.

[15] Mihai Cucuringu, Huan Li, He Sun, and Luca Zanetti. Hermitian matrices for clustering directed graphs: insights and applications. *arXiv preprint arXiv:1908.02096*, 2019.

[16] Lian-Biao Cui, Pan Peng, and Lei Zhu. Embodied energy, export policy adjustment and china's sustainable development: a multi-regional input-output analysis. *Energy*, 82:457–467, 2015.

[17] Na Cui, Yalin Lei, and Wei Fang. Design and impact estimation of a reform program of china's tax and fee policies for low-grade oil and gas resources. *Petroleum Science*, 8(4):515–526, 2011.

[18] Luca De Benedictis and Lucia Tajoli. The world trade network. *The World Economy*, 34(8):1417–1454, 2011.

[19] Monika Dittrich and Stefan Bringezu. The physical dimension of international trade: Part 1: Direct global flows between 1962 and 2005. *Ecological Economics*, 69(9):1838 – 1847, 2010.

[20] Giorgio Fagiolo, Javier Reyes, and Stefano Schiavo. On the topological properties of the world trade web: A weighted network analysis. *Physica A: Statistical Mechanics and its Applications*, 387(15):3868–3873, 2008.

[21] Giorgio Fagiolo, Javier Reyes, and Stefano Schiavo. The evolution of the world trade web: a weighted-network analysis. *Journal of Evolutionary Economics*, 20(4):479–514, 2010.

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

[23] Michel X. Goemans and David P. Williamson. Approximation algorithms for max-3-cut and other problems via complex semidefinite programming. *Journal of Computer and System Sciences*, 68(2):442 – 470, 2004. Special Issue on STOC 2001.

[24] James D Hamilton. Causes and consequences of the oil shock of 2007-08. Technical report, National Bureau of Economic Research, 2009.

[25] James D Hamilton. Historical oil shocks. Technical report, National Bureau of Economic Research, 2011.

[26] Folke Hilgerdt. The case for multilateral trade. *The American Economic Review*, pages 393–407, 1943.

[27] Lutz Kilian. Exogenous oil supply shocks: how big are they and how much do they matter for the us economy? *The Review of Economics and Statistics*, 90(2):216–240, 2008.

[28] Lutz Kilian and Daniel P Murphy. The role of inventories and speculative trading in the global market for crude oil. *Journal of Applied Econometrics*, 29(3):454–478, 2014.

[29] Steinar Laenen. Yearly progression of top pair 1 from 1988 to 2017. `https://www.youtube.com/watch?v=HIfPewuqfnU`.

[30] Steinar Laenen. Yearly progression of top pair 2 from 1988 to 2017. `https://youtu.be/7_nw3PHXsds`.

[31] Steinar Laenen. Yearly progression of top pair 3 from 1988 to 2017. `https://youtu.be/E9PRZrPIAnQ`.

[32] Steinar Laenen. Yearly progression of top pairs 1, 2 and 3 from 1988 to 2017. `https://www.youtube.com/watch?v=wLRmxjzX9Ws`.

[33] Steinar Laenen. Informatics project proposal: An analysis of using hermitian matrices for community detection in directed stochastic block models. *Informatics Project Proposal*, 2019.

[34] James R Lee, Shayan Oveis Gharan, and Luca Trevisan. Multiway spectral partitioning and higher-order cheeger inequalities. *Journal of the ACM (JACM)*, 61(6):37, 2014.

[35] Anand Louis, Prasad Raghavendra, Prasad Tetali, and Santosh Vempala. Many sparse cuts via higher eigenvalues. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1131–1140. ACM, 2012.

[36] Fragkiskos D Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics Reports*, 533(4):95–142, 2013.

[37] Fragkiskos D. Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks: A survey. *CoRR*, abs/1308.0971, 2013.

[38] László Mátyás. Proper econometric specification of the gravity model. *World Economy*, 20(3):363–368, 1997.

[39] Wes McKinney. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

[40] Frank McSherry. Spectral partitioning of random graphs. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 529–537. IEEE, 2001.

[41] Alexandre Mejean and Jonas Maison. Spectral Methods for community detection in large networks. `https://github.com/Jonas1312/CommunityDetection`, 2016. Accessed: 2019-08-14.

[42] United Nations. Un comtrade free api. `https://comtrade.un.org/data/`. Accessed: 2019-08-13.

[43] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[44] Travis Oliphant. NumPy: A guide to NumPy. USA: Trelgol Publishing, 2006–.

[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[46] Richard Peng, He Sun, and Luca Zanetti. Partitioning well-clustered graphs: Spectral clustering works! In *Conference on Learning Theory*, pages 1423–1455, 2015.

[47] Python Core Team. *Python: A dynamic, open source programming language*. Python Software Foundation, 2019.

[48] Prasad Raghavendra, David Steurer, and Madhur Tulsiani. Reductions between expansion problems. In *2012 IEEE 27th Conference on Computational Complexity*, pages 64–73. IEEE, 2012.

[49] Karl Rohe, Sourav Chatterjee, Bin Yu, et al. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

[50] 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.

[51] Venu Satuluri and Srinivasan Parthasarathy. Symmetrizations for clustering directed graphs. In *Proceedings of the 14th International Conference on Extending Database Technology*, EDBT/ICDT '11, pages 343–354, New York, NY, USA, 2011. ACM.

[52] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Departmental Papers (CIS)*, page 107, 2000.

[53] Jiří Šíma and Satu Elisa Schaeffer. On the np-completeness of some graph cluster measures. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 530–537. Springer, 2006.

[54] D. A. Spielman. Spectral graph theory and its applications. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 29–38, Oct 2007.

[55] He Sun and Luca Zanetti. Distributed graph clustering and sparsification. *arXiv preprint arXiv:1711.01262*, 2017.

[56] Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

[57] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[58] Van Vu. A simple svd algorithm for finding hidden partitions. *arXiv preprint arXiv:1404.3918*, 2014.

[59] Luca Zannetti. *Algorithms for Partitioning Well-Clustered Graphs*. PhD thesis, University of Bristol, 2017.

[60] Weiqiong Zhong, Haizhong An, Xiangyun Gao, and Xiaoqi Sun. The evolution of communities in the international oil trade network. *Physica A: Statistical Mechanics and its Applications*, 413:42 – 52, 2014.

[61] Zeyuan Allen Zhu, Silvio Lattanzi, and Vahab S Mirrokni. A local algorithm for finding well-connected clusters. In *ICML (3)*, pages 396–404, 2013.

Appendix A

Omitted Proofs and Additional Details of Chapter 3

A.1 Omitted Proof Details of Lemma 3.1

In the proof of the structure theorem we use the indicator vector

$$\chi_G(u) = \begin{cases} \omega & \text{if } u \in S \\ \omega^3 & \text{if } u \in T \\ \omega^2 & \text{if } u \in R. \end{cases}$$

where ω is the third root of unity. Its corresponding normalised indicator vector is

$$h_G = \frac{D^{1/2}\chi_G}{\|D^{1/2}\|}.$$

We show that the following inequality holds:

$$\lambda_1 \leqslant \mathcal{R}(h_G) = 1 - \frac{\sqrt{3} \cdot CF(G)}{vol(V)},$$

where the cyclic flow $CF(G) = NF(S,T) + NF(T,R) + NF(R,S)$.

Proof. Recall that for a given complex Hermitian matrix $M \in \mathbb{C}^{n\times n}$, and non-zero vector $x \in \mathbb{C}^n$, the Rayleigh quotient $R(M,x)$ is defined as

$$R(M,x) = \frac{x^\dagger Mx}{x^\dagger x}.$$

Let $\mathcal{L}_G$ denote the normalised Hermitian Laplacian representation of a weigthed digraph $G = (V,E,w)$, and let $x \in \mathbb{C}^n$ be a non-zero vector. We can write the numer-

ator of the Rayleigh quotient as

$$R^{\text{num}}(M,x) = x^\dagger Mx = \sum_{u\in V} x_u^\dagger x_u - i \cdot \sum_{(u,v)\in E} \frac{w(u,v)\cdot(x_u^\dagger x_v - x_v^\dagger x_u)}{\sqrt{d_u d_v}}.$$

We first compute the numerator of the Rayleigh quotient $R(\mathcal{L}_G, h_G)$. For ease of notation, we refer to $\chi_G(u)$ as x_u:

$$R^{\text{num}}(\mathcal{L}_G,h_G) = R^{\text{num}}(L,\chi_G) = \sum_{u\in V} d_u \cdot x_u^\dagger x_u - \sum_{e=(u,v)} i\cdot w(u,v)\cdot(x_u^\dagger x_v - x_v^\dagger x_u)$$

Let us compute the value inside the sum for $R^{\text{num}}(\mathcal{L}_G,h_G)$ for the 9 different cases of edge $e=(u,v)$, where the tail $u \in S,T,$or R and where the head $v \in S,T,$or R.

Notice that if u and v belong to the same cluster, we get that the term $x_u^\dagger x_u - x_u^\dagger x_u = 0$. We now compute the remaining six cases, where we recall that $\omega = (\omega^2)^* = (-1+i\sqrt{3})/2$, $\omega^2 = \omega^* = (-1-i\sqrt{3})/2$, and $\omega^3 = 1$:

case (1), $u \in S$ and $v \in T$:

$$i\cdot(x_u^\dagger x_v - x_v^\dagger x_u) = i\cdot((\omega^*\cdot 1) - ((1\cdot\omega) = i\cdot(\omega^* - \omega) = i\cdot -i\sqrt{3} = \sqrt{3}.$$

case (2), $u \in S$ and $v \in R$:

$$i\cdot(x_u^\dagger x_v - x_v^\dagger x_u) = i\cdot((\omega^*\cdot\omega^2) - ((\omega^2)^*\cdot\omega) = i\cdot(\omega - \omega^*) = i\cdot i\sqrt{3} = -\sqrt{3}.$$

case (3), $u \in T$ and $v \in S$:

$$i\cdot(x_u^\dagger x_v - x_v^\dagger x_u) = i\cdot((1\cdot\omega) - (\omega^*\cdot 1) = i\cdot(\omega - \omega^*) = i\cdot i\sqrt{3} = -\sqrt{3}.$$

case (4), $u \in T$ and $v \in R$:

$$i\cdot(x_u^\dagger x_v - x_v^\dagger x_u) = i\cdot((1\cdot\omega^2) - ((\omega^2)^*\cdot 1) = i\cdot(\omega^* - \omega) = i\cdot -i\sqrt{3} = \sqrt{3}.$$

case (5), $u \in R$ and $v \in S$:

$$i\cdot(x_u^\dagger x_v - x_v^\dagger x_u) = i\cdot((\omega^2)^*\cdot\omega) - (\omega^*\cdot\omega^2) = i\cdot(\omega^* - \omega) = i\cdot -i\sqrt{3} = \sqrt{3}.$$

case (6), $u \in R$ and $v \in T$:

$$i\cdot(x_u^\dagger x_v - x_v^\dagger x_u) = i\cdot((\omega^2)^*\cdot 1) - (1\cdot\omega^2) = i\cdot(\omega - \omega^*) = i\cdot i\sqrt{3} = -\sqrt{3}.$$

Given this, we can rewrite $R^{\text{num}}(\mathcal{L}_G, h_G)$ as

$$\begin{aligned}
R^{\text{num}}(\mathcal{L}_G, h_G) = &\sum_{u \in V} d_u \\
&- \sqrt{3} \cdot \sum_{\substack{e=(u,v) \\ u \in S, v \in T}} w(u,v) + \sqrt{3} \cdot \sum_{\substack{e=(u,v) \\ u \in T, v \in S}} w(u,v) \\
&- \sqrt{3} \cdot \sum_{\substack{e=(u,v) \\ u \in T, v \in R}} w(u,v) + \sqrt{3} \cdot \sum_{\substack{e=(u,v) \\ u \in T, v \in R}} w(u,v) \\
&- \sqrt{3} \cdot \sum_{\substack{e=(u,v) \\ u \in R, v \in S}} w(u,v) + \sqrt{3} \cdot \sum_{\substack{e=(u,v) \\ u \in R, v \in S}} w(u,v) \\
&= vol(V) - \sqrt{3} \cdot (NF(S,T) + NF(T,R) + NF(R,S))
\end{aligned}$$

The denominator of the Rayleigh quotient $R^{\text{denum}}(\mathcal{L}_G, h_G)$ is

$$R^{\text{denum}}(\mathcal{L}_G, h_G) = (h_G)^\dagger h_G = \sum_{u \in V} d_u = vol(V).$$

Hence, we can now write the full Rayleigh quotient as

$$\begin{aligned}
R(\mathcal{L}_G, h_G) =& \frac{vol(V) - \sqrt{3} \cdot (NF(S,T) + NF(T,R) + NF(R,S))}{vol(V)} \\
=& 1 - \frac{\sqrt{3} \cdot (NF(S,T) + NF(T,R) + NF(R,S)}{vol(V)} \\
=& 1 - \frac{\sqrt{3} \cdot CF(G)}{vol(V)}.
\end{aligned}$$

By the Rayleigh characterisation of eigenvalues we know that

$$\lambda_1(\mathcal{L}_G) = \min_{x \in \mathbb{C}^n \setminus \{0\}} \left[\frac{x^\dagger \mathcal{L}_G x}{x^\dagger x} \right] \leqslant R(\mathcal{L}_G, h_G),$$

which concludes the proof. □

A.2 Omitted Proof Details of Theorem 3.2

We restate the Structure Theorem Here for convenience:

Theorem (The Structure Theorem)**.** *Let ξ be defined as in (3.2), and let h_G be defined as in (3.1). Then the following statements hold:*

1. *There is a coefficient $\alpha_1 \in \mathbb{C}$: $\widetilde{f}_1 = \alpha_1 f_1$, such that $\left\| h_G - \widetilde{f}_1 \right\|^2 \leqslant 1/\xi$.*

2. There is a coefficient $\beta_1 \in \mathbb{C}$: $\widetilde{h_G} = \beta_1 h_G$, such that $\left\| f_1 - \widetilde{h_G} \right\|^2 \leqslant 1/(\xi - 1)$.

Proof of Theorem 3.2. The proof we write here is essentially a simpler version of the proof done by Peng et al. ([46], Section 3) for their Structure Theorem, since we only need to consider the case that $k = 3$. We first prove the first statement. We write h_G as a linear combination of the eigenvectors of $\mathcal{L}_G$:

$$h_G = \alpha_1 f_1 + \cdots + \alpha_n f_n,$$

where $\alpha_i \in \mathbb{C}$ and $f_i \in \mathbb{C}^n$. Let $\widetilde{f_1}$ be the projection of h_G onto the first eigenvector, i.e.,

$$\widetilde{f_1} = \alpha_1 f_1.$$

By the definition of the Rayleigh quotient for Hermitian matrices we have that

$$\begin{aligned}
\mathcal{R}(h_G) &= (\alpha_1 f_1 + \cdots + \alpha_n f_n)^\dagger \mathcal{L}_G (\alpha_1 f_1 + \cdots + \alpha_n f_n) \\
&= \|\alpha_1\|^2 \lambda_1 + \cdots + \|\alpha_n\|^2 \lambda_n \\
&\geqslant \|\alpha_1\|^2 \lambda_1 + (\|\alpha_2\|^2 + \cdots + \|\alpha_n\|^2) \lambda_2 \\
&\geqslant (1 - \|\alpha_1\|^2) \lambda_2,
\end{aligned}$$

where the first inequality holds because $\lambda_2 \leqslant \lambda_3 \leqslant \cdots \leqslant \lambda_n$, and in the second inequality we use the fact that $(\|\alpha_2\|^2 + \cdots + \|\alpha_n\|^2) = 1 - \|\alpha_1\|^2$, which holds since $\|h_G\|^2 = 1$. We can see that

$$\left\| h_G - \widetilde{f_1} \right\|^2 = \|\alpha_2\|^2 + \cdots + \|\alpha_n\|^2 = 1 - \|\alpha_1\|^2 \leqslant \frac{\mathcal{R}(h_G)}{\lambda_2} \leqslant \frac{1}{\xi},$$

which finishes the proof of the first statement.

Next, we prove the second statement. We assume that $\lambda_1 > 0$. This assumption is valid, since this also ensures that the gap $\xi \leqslant \frac{\lambda_2}{\lambda_1}$ is well defined. We remark that, while the smallest eigenvalue of the normalised Laplacian matrix for an undirected graph is always equal to 0, it is not the case for directed graphs (see Section 3.4 for further discussion). Assuming $\lambda_1 > 0$, we have $\operatorname{span}\left\{\widetilde{f_1}\right\} = \operatorname{span}\{f_1\}$, and hence can write f_1 as a linear combination of $\widetilde{f_1}$

$$f_1 = \beta_1 \widetilde{f_1},$$

where we can easily see that $\beta_1 = 1/\alpha_1$, since $f_1 = \beta_1 \widetilde{f_1} = \beta_1 \alpha_1 f_1$. We now define $\widetilde{h_G}$ as

$$\widetilde{h_G} = \beta_1 h_G = \frac{1}{\alpha_1} h_G = \frac{1}{\alpha_1} (\alpha_1 f_1 + \cdots + \alpha_n f_n) = f_1 + \frac{1}{\alpha_1} (\alpha_2 f_2 + \cdots + \alpha_n f_n).$$

We get that

$$\left\|f_1 - \widetilde{h_G}\right\|^2 = \left\|\frac{\alpha_2}{\alpha_1}\right\|^2 + \cdots \left\|\frac{\alpha_n}{\alpha_1}\right\|^2 = \frac{1}{\|\alpha_1\|^2}\left(\|\alpha_2\|^2 \cdots \|\alpha_n\|^2\right) = \frac{1}{\|\alpha_1\|^2}\left(1 - \|\alpha_1\|^2\right)$$
$$\leqslant \frac{1}{\|\alpha_1\|^2 \cdot \xi},$$

where the last inequality follows from the inequality $1 - \|\alpha_1\|^2 \leqslant 1/\xi$ that we found in the proof of Part 1 of this theorem. Since $\|f_1\|^2 = 1$, we can write $1/\|\alpha_1\|^2$ as

$$\begin{aligned} \|\alpha_1\|^2 &\leqslant 1 \\ 0 \leqslant 1 - \|\alpha_1\|^2 &\leqslant \frac{1}{\xi} \\ 1 - \frac{1}{\xi} &\leqslant \|\alpha_1\|^2 \\ \frac{1}{1 - \frac{1}{\xi}} &\geqslant \frac{1}{\|\alpha_1\|^2}. \end{aligned}$$

And finally we get

$$\left\|f_1 - \widetilde{h_G}\right\|^2 \leqslant \frac{1}{\|\alpha_1\|^2 \cdot \xi} \leqslant \frac{1}{\left(1 - \frac{1}{\xi}\right) \cdot \xi} = \frac{1}{\xi - 1},$$

which finishes the proof of the theorem. □

A.3 Omitted Proof Details of Lemma 3.5

Lemma A.1. *([46], Lemma 4.6) Suppose that for every permutation $\pi : \{1,2,3\} \to \{1,2,3\}$ there exists an index i such that $\mathrm{vol}(A_i \triangle S_{\pi(i)}) \geqslant 2\varepsilon \mathrm{vol}(S_{\pi(i)})$. Then at least one of the following two cases holds:*

1. *for any index i there are indices $i_1 \neq i_2$ and $\varepsilon_i \geqslant 0$ such that*

$$\mathrm{vol}(A_i \cap S_{i_1}) \geqslant \mathrm{vol}(A_i \cap S_{i_2}) \geqslant \varepsilon_i \min\left\{\mathrm{vol}(S_{i_1}), \mathrm{vol}(S_{i_2})\right\}$$

   and $\sum_{i=1}^3 \varepsilon_i \geqslant \varepsilon$;

2. *there exists indices i, ℓ and $\varepsilon_j \geqslant 0$ such that for $j \neq \ell$,*

$$\mathrm{vol}(A_i \cap S_\ell) \geqslant \varepsilon_i \mathrm{vol}(S_\ell)$$
$$\mathrm{vol}(A_i \cap S_j) \geqslant \varepsilon_i \mathrm{vol}(S_\ell)$$

   and $\sum_{i=1}^3 \varepsilon_i \geqslant \varepsilon$.

A.4 Omitted Proof Details of Statement 1 of Theorem 3.6

Lemma A.2 (Bernstein's Inequality, [12])**.** *Let $X_1,...X_n$ be independent random variables such that $|X_i| \leqslant M$ for any $i \in \{1,...,n\}$. Let $X = \sum_i^n X_i$ and let $R = \sum_i^n \mathbb{E}[X_i^2]$. Then, it holds that*

$$\mathbb{P}\left[|X - \mathbb{E}[X]| \geqslant t\right] \leqslant 2exp\left(-\frac{t^2}{2(R + Mt/3)}\right).$$

A.5 Omitted Proof Details of Statement 2 of Theorem 3.6

We use the following lemma in the analysis:

Lemma A.3 (Matrix Chernoff Bound, [56])**.** *Consider a finite sequence $\{X_i\}$ of independent, random, PSD matrices of dimension d that satisfy $\|X_i\| \leqslant R$. Let $\mu_{\min} \triangleq \lambda_{\min}\left(\mathbb{E}\left[\sum_i X_i\right]\right)$ and $\mu_{\max} \triangleq \lambda_{\max}\left(\mathbb{E}\left[\sum_i X_i\right]\right)$. Then it holds that*

$$\mathbb{P}\left[\lambda_{\min}\left(\sum_i X_i\right) \leqslant (1-\delta)\mu_{\min}\right] \leqslant d \cdot \left(\frac{\mathrm{e}^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu_{\min}/R} \quad \textit{for } \delta \in [0,1], \textit{ and}$$

$$\mathbb{P}\left[\lambda_{\max}\left(\sum_i X_i\right) \geqslant (1+\delta)\mu_{\max}\right] \leqslant d \cdot \left(\frac{\mathrm{e}^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu_{\max}/R} \quad \textit{for } \delta \geqslant 0.$$

Proof Statement 2 Theorem 3.6. We prove that the top $n-1$ eigenspace is approximately preserved in H. Let $\overline{\mathcal{L}}_G$ be the projection of $\mathcal{L}_G$ on its top $n-1$ eigenspaces. We can write $\overline{\mathcal{L}}_G$ as

$$\overline{\mathcal{L}}_G = \sum_{i=2}^{n} \lambda_i f_i f_i^\dagger.$$

With a slight abuse of notation we call $\overline{\mathcal{L}}_G^{-1/2}$ the square root of the pseudoinverse of $\overline{\mathcal{L}}_G$, i.e.,

$$\overline{\mathcal{L}}_G^{-1/2} = \sum_{i=2}^{n} (\lambda_i)^{-1/2} f_i f_i^\dagger.$$

We call $\overline{\mathcal{I}}$ the projection on $\mathrm{span}\{f_2,\ldots,f_n\}$, i.e.,

$$\overline{\mathcal{I}} = \sum_{i=2}^{n} f_i f_i^\dagger.$$

We will prove that the top $n-1$ eigenspaces of $\mathcal{L}_G$ are preserved, which implies that $\lambda_2(\mathcal{L}_G) = \Theta(\lambda_2(\mathcal{L}_H))$. To prove this, recall that the probability of any edge $e = (u,v)$ being sampled in H is

$$p_e = p_u(v) + p_v(u) - p_u(v) \cdot p_v(u),$$

and it holds that $\frac{1}{2}(p_u(v) + p_v(u)) \leqslant p_e \leqslant p_u(v) + p_v(u)$. Now for each edge $e = (u,v)$ of G we define a random matrix $X_e \in \mathbb{C}^{n\times n}$ by

$$X_e = \begin{cases} w_H(u,v) \cdot \overline{\mathcal{L}}_G^{-1/2} b_e b_e^\dagger \overline{\mathcal{L}}_G^{-1/2} & \text{if } e = (u,v) \text{ is sampled by the algorithm,} \\ 0 & \text{otherwise,} \end{cases}$$

where we recall that $b_e = (1\ \sqrt{2}) \cdot ((1-i)\chi_u + (1+i)\chi_v)$ as defined in (3.7), where χ_u and χ_v are the indicator vectors for vertices u and v. Notice that

$$\sum_{e\in E[G]} X_e = \sum_{\text{sampled edges } e=(u,v)} w_H(u,v) \cdot \overline{\mathcal{L}}_G^{-1/2} b_e b_e^\dagger \overline{\mathcal{L}}_G^{-1/2} = \overline{\mathcal{L}}_G^{-1/2} \mathcal{L}'_H \overline{\mathcal{L}}_G^{-1/2},$$

where it follows from Lemma 3.7 that

$$\mathcal{L}'_H = \sum_{\text{sampled edges } e=(u,v)} w_H(u,v) \cdot b_e b_e^\dagger$$

is essentially the Laplacian matrix of H but is normalised with respect to the degrees of the vertices in the original graph G, i.e., $\mathcal{L}'_H = D_G^{-1} D_H - D_G^{-1/2} A_H D_G^{-1/2}$. We will prove that, with high probability, the top $n-1$ eigenspaces of $\mathcal{L}'_H$ and $\mathcal{L}_G$ are approximately the same. Later we will show the same holds for $\mathcal{L}_H$ and $\mathcal{L}'_H$, which implies that $\lambda_2(\mathcal{L}'_H) = \Omega(\lambda_2(\mathcal{L}_G))$.

We will use the matrix Chernoff bound for our proof. We start looking at the first moment of the expression above:

$$\begin{aligned} \mathbb{E}\left[\sum_{e\in E} X_e\right] &= \sum_{e=(u,v)\in E[G]} p_e \cdot w_H(u,v) \cdot \overline{\mathcal{L}}_G^{-1/2} b_e b_e^\dagger \overline{\mathcal{L}}_G^{-1/2} \\ &= \sum_{e=(u,v)\in E[G]} p_e \cdot \frac{w(u,v)}{p_e} \cdot \overline{\mathcal{L}}_G^{-1/2} b_e b_e^\dagger \overline{\mathcal{L}}_G^{-1/2} \\ &= \overline{\mathcal{L}}_G^{-1/2} \mathcal{L}_G \overline{\mathcal{L}}_G^{-1/2} = \overline{\mathcal{I}}. \end{aligned}$$

Moreover, for any sampled $e = (u,v) \in E$ we have that

$$\begin{aligned} \|X_e\| &\leqslant w_H(u,v) \cdot b_e^\dagger \overline{\mathcal{L}}_G^{-1/2} \overline{\mathcal{L}}_G^{-1/2} b_e = \frac{w(u,v)}{p_e} \cdot b_e^\dagger \overline{\mathcal{L}}_G^{-1} b_e \leqslant \frac{w(u,v)}{p_e} \cdot \frac{1}{\lambda_2} \cdot \|b_e\|^2 \\ &\leqslant \frac{2\lambda_2}{\alpha \cdot \log n \cdot \left(\frac{1}{d_u^{\text{out}}} + \frac{1}{d_v^{\text{in}}}\right)} \cdot \frac{1}{\lambda_2} \left(\frac{1}{d_u^{\text{out}}} + \frac{1}{d_v^{\text{in}}}\right) \leqslant \frac{2}{\alpha \log n}, \end{aligned}$$

where the second inequality follows by the min-max theorem of eigenvalues. Now we apply the matrix Chernoff bound (Lemma A.3) to analyse the eigenvalues of $\sum_{e\in E} X_e$, and build a connection between $\lambda_2(\mathcal{L}'_H)$ and $\lambda_2(\mathcal{L}_G)$. By setting the parameters of

Lemma A.3 by $\mu_{\max} = \lambda_{\max}\left(\mathbb{E}\left[\sum_{e\in E[G]} X_e\right]\right) = \lambda_{\max}\left(\overline{\mathcal{I}}\right) = 1$, $R = 2/\left(\alpha \cdot \log n\right)$ and $\delta = 1/2$, we have that

$$\mathbb{P}\left[\lambda_{\max}\left(\sum_{e\in E[G]} X_e\right) \geqslant 3/2\right] \leqslant n \cdot \left(\frac{\mathrm{e}^{1/2}}{(1+1/2)^{3/2}}\right)^{\alpha \log n/2} = O\left(1/n^c\right)$$

for some constant c. This gives us that

$$\mathbb{P}\left[\lambda_{\max}\left(\sum_{e\in E[G]} X_e\right) \leqslant 3/2\right] = 1 - O(1/n^c). \tag{A.1}$$

On the other side, since our goal is to analyse $\lambda_2(\mathcal{L}'_H)$ with respect to $\lambda_2(\mathcal{L}_G)$, it suffices to work with the top $(n-1)$ eigenspace of $\mathcal{L}_G$. Since $\mathbb{E}\left[\sum_{e\in E} X_e\right] = \overline{\mathcal{I}}$, we can assume without loss of generality that $\mu_{\min} = 1$. Hence, by setting $R = 2/\left(\alpha \cdot \log n\right)$ and $\delta = 1/2$, we have that

$$\mathbb{P}\left[\lambda_{\min}\left(\sum_{e\in E[G]} X_e\right) \leqslant 1/2\right] = n \cdot \left(\frac{\mathrm{e}^{-1/2}}{(1/2)^{1/2}}\right)^{\alpha \log n/2} = O\left(1/n^c\right)$$

for some constant c. This gives us that

$$\mathbb{P}\left[\lambda_{\min}\left(\sum_{e\in E[G]} X_e\right) > 1/2\right] = 1 - O(1/n^c). \tag{A.2}$$

Combining (A.1), (A.2), and the fact of $\sum_{e\in E[G]} X_e = \overline{\mathcal{L}}_G^{-1/2} \mathcal{L}'_H \overline{\mathcal{L}}_G^{-1/2}$, with probability $1 - O\left(1/n^c\right)$ it holds for any non-zero $x \in \mathbb{C}^n$ in the space spanned by $f_2, \ldots, f_n$ that

$$\frac{x^\dagger \overline{\mathcal{L}}_G^{-1/2} \mathcal{L}'_H \overline{\mathcal{L}}_G^{-1/2} x}{x^\dagger x} \in (1/2, 3/2)\,. \tag{A.3}$$

By setting $y = \overline{\mathcal{L}}_G^{-1/2} x$, we can rewrite (A.3) as

$$\frac{y^\dagger \mathcal{L}'_H y}{y^\dagger \overline{\mathcal{L}}_G^{1/2} \overline{\mathcal{L}}_G^{1/2} y} = \frac{y^\dagger \mathcal{L}'_H y}{y^\dagger \overline{\mathcal{L}}_G y} = \frac{y^\dagger \mathcal{L}'_H y}{y^\dagger y} \frac{y^\dagger y}{y^\dagger \overline{\mathcal{L}}_G y} \in (1/2, 3/2).$$

Since $\dim(\mathrm{span}\{f_2, \ldots, f_n\}) = n-1$, we have just proved there exist $n-1$ orthogonal vectors whose Rayleigh quotient with respect to $\mathcal{L}'_H$ is $\Omega(\lambda_2(\mathcal{L}_G))$. By the Courant-Fischer Theorem, we have

$$\lambda_2(\mathcal{L}'_H) \geqslant \frac{1}{2}\lambda_2(\mathcal{L}_G). \tag{A.4}$$

It remains to show that $\lambda_2(\mathcal{L}_H) = \Omega\left(\lambda_2(\mathcal{L}'_H)\right)$, which implies that $\lambda_2(\mathcal{L}_H) = \Omega\left(\lambda_2(\mathcal{L}_G)\right)$ by (A.4). By the definition of $\mathcal{L}'_H$, we have that for the Laplacian $\mathcal{L}_H =$

$D_H^{-1/2} D_G^{1/2} \mathcal{L}'_H D_G^{1/2} D_H^{-1/2}$. Therefore, for any $x \in \mathbb{C}^n$ and $y = D_G^{1/2} D_H^{-1/2} x$, it holds that

$$\frac{x^\dagger \mathcal{L}_H x}{x^\dagger x} = \frac{y^\dagger \mathcal{L}'_H y}{x^\dagger x} \geqslant \frac{1}{2} \cdot \frac{y^\dagger \mathcal{L}'_H y}{y^\dagger y}, \tag{A.5}$$

where the last equality follows from the fact that the degrees in H and G differ just by a constant multiplicative factor, and therefore,

$$y^\dagger y = \left(D_G^{1/2} D_H^{-1/2} x\right)^\dagger \left(D_G^{1/2} D_H^{-1/2} x\right) = x^\dagger D_G D_H^{-1} x \geqslant \frac{1}{2} \cdot x^\dagger x.$$

Finally, we show that (A.5) implies that $\lambda_2(\mathcal{L}_H) \geqslant (1/2) \cdot \lambda_2(\mathcal{L}'_H)$. To see this, let $S_1 \subseteq \mathbb{C}^n$ be a (2)-dimensional subspace of $\mathbb{C}^n$ such that

$$\lambda_2(\mathcal{L}_H) = \max_{x \in S_1} \frac{x^\dagger \mathcal{L}_H x}{x^\dagger x}.$$

Let $S_2 = \left\{ D_G^{1/2} D_H^{-1/2} x \colon x \in S_1 \right\}$. Notice that since $D_G^{1/2} D^{-1/2}$ is full rank, S_2 has dimension 2. Therefore,

$$\lambda_2(\mathcal{L}'_H) = \min_{S \colon \dim(S)=2} \max_{y \in S} \frac{y^\dagger \mathcal{L}'_H y}{y^\dagger y} \leqslant \max_{y \in S_2} \frac{y^\dagger \mathcal{L}'_H y}{y^\dagger y} \leqslant 2 \max_{x \in S_1} \frac{x^\dagger \mathcal{L}_H x}{x^\dagger x} = 2\lambda_2(\mathcal{L}_H), \tag{A.6}$$

where the last inequality follows by (A.5). Combining (A.4) with (A.6) gives us that $\lambda_2(\mathcal{L}_H) = \Omega(\lambda_2(G))$, which implies $\xi_H = \Omega(\xi_G)$. This concludes the proof. □

A.6 Omitted Proof Details of Lemma 3.7

Lemma A.4. *Given a directed graph $G = (V, E)$, where the Laplacian L is constructed using the Hermitian adjacency matrix, it holds that $L = B^\dagger W B$*

Proof. We can write each individual entry of $B^\dagger W B$ as

$$(B^\dagger W B)_{u,v} = \sum_e B^\dagger_{u,e} (WB)_{e,v} = \sum_e B^\dagger_{u,e} W_{e,e} B_{e,v}. \tag{A.7}$$

Let us consider the diagonal first, i.e. $u = v$. We can write

$$\sum_e B^\dagger_{u,e} W_{e,e} B_{e,u} = \sum_e W_{e,e} |B_{e,u}|^2 = \sum_e \frac{1}{2} \cdot W_{e,e} (1+i)(1-i) = d_u, \tag{A.8}$$

where the last equality sign holds because $B_{e,u} \neq 0$ only if edge e is adjacent to vertex v, no matter if it is an in or outgoing edge. The factor $\frac{1}{2}$ disappears because $(1-i)(1+i) = 2$.

Now we assume that $u \neq v$, but that they are connected by an edge. Then there are two cases. The first case is that $(v,u) \in E$. Then it holds that

$$\sum_e B^\dagger_{u,e} W_{e,e} B_{e,v} = \frac{1}{2} \cdot (1-i)(1-i) = -i, \tag{A.9}$$

since we only get one non-zero term from the sum because there is only one edge between two vertices, and v is e's tail and u is e's head. For the second case, we have that $(u,v) \in E$ and by symmetry we get

$$\sum_e B^\dagger_{u,e} W_{e,e} B_{e,v} = \frac{1}{2} \cdot (1+i)(1+i) = i. \tag{A.10}$$

If there is no edge between vertices v and u, then the term will be 0. This concludes the proof that $L = B^\dagger W B$, since on the diagonal entries we get the degree of each vertex, and on the off-diagonal entries we get either i or $-i$ depending on the direction of the edge. □

Appendix B

Omitted Proofs and Additional Details of Chapter 4

B.1 Omitted Proof Details of Algorithm Proposed in Section 4.3

Lemma B.1. *Given a directed graph $G = (V, E)$, there exists an algorithm that finds a subset of vertices $S \subset V$, such that it maximises the net flow* $\mathrm{NF}(S) = w(S, V \setminus S) - w(V \setminus S, S)$, *where* $w = \sum_{x \in X, y \in Y} w(x, y)$.

We can write the net flow for a single vertex to be the difference between the number of incoming vertices and outgoing vertices, so for a vertex $v \in V$:

$$n(v) = \sum_{w \in V} w(v, w) - \sum_{w \in V} w(w, v). \tag{B.1}$$

Let us define the following algorithm that satisfies the requirement of Lemma B.1:

1. For each vertex $v \in V$, compute the $n(v)$
2. Then we let the set $S = \{v \in V \mid n(v) > 0\}$, and $T = \{v \in V \mid n(v) \leqslant 0\}$.

We now prove that this algorithm gives the maximum net flow cut.

Proof. Assume by contradiction that the set S resulting from the algorithm is not the maximum net flow cut. This implies there exists a set $S' \subset V$, such that:

$$w(S, V \setminus S) - w(V \setminus S, S) < w(S', V \setminus S') - w(V \setminus S', S'). \tag{B.2}$$

Since S is the set with all the vertices v, such that $n(v) > 0$, and $S' \neq S$, we know that there exists a vertex $v' \in V \setminus S'$, such that $n(v') > 0$. Let's construct a new set $T = S' \cup \{v'\}$. The net flow for set T is:

$$\begin{aligned} NF(T) &= w(S', V \setminus S') - w(V \setminus S', S') + \\ &\quad w(v', V \setminus S') + w(v', S') - w(V \setminus S', v') - w(S', v') \\ &= w(S', V \setminus S') - w(V \setminus S', S') + n(v') \\ &= \mathrm{NF}(S') + n(v') > \mathrm{NF}(S') \end{aligned}$$

However, we assumed that $F(S')$ had the maximum net flow, so we reached a contradiction. Therefore we showed that for any set that does not contain all vertices with positive net flow, we can add the vertices with positive net flow and improve the net flow over the set. Furthermore, it is easy to show that by symmetry that removing all vertices with negative net flow from S improves the net flow of the set. Therefore the set that the algorithm outputs must have the maximum net flow. □

B.2 Omitted Plots for Section 4.3

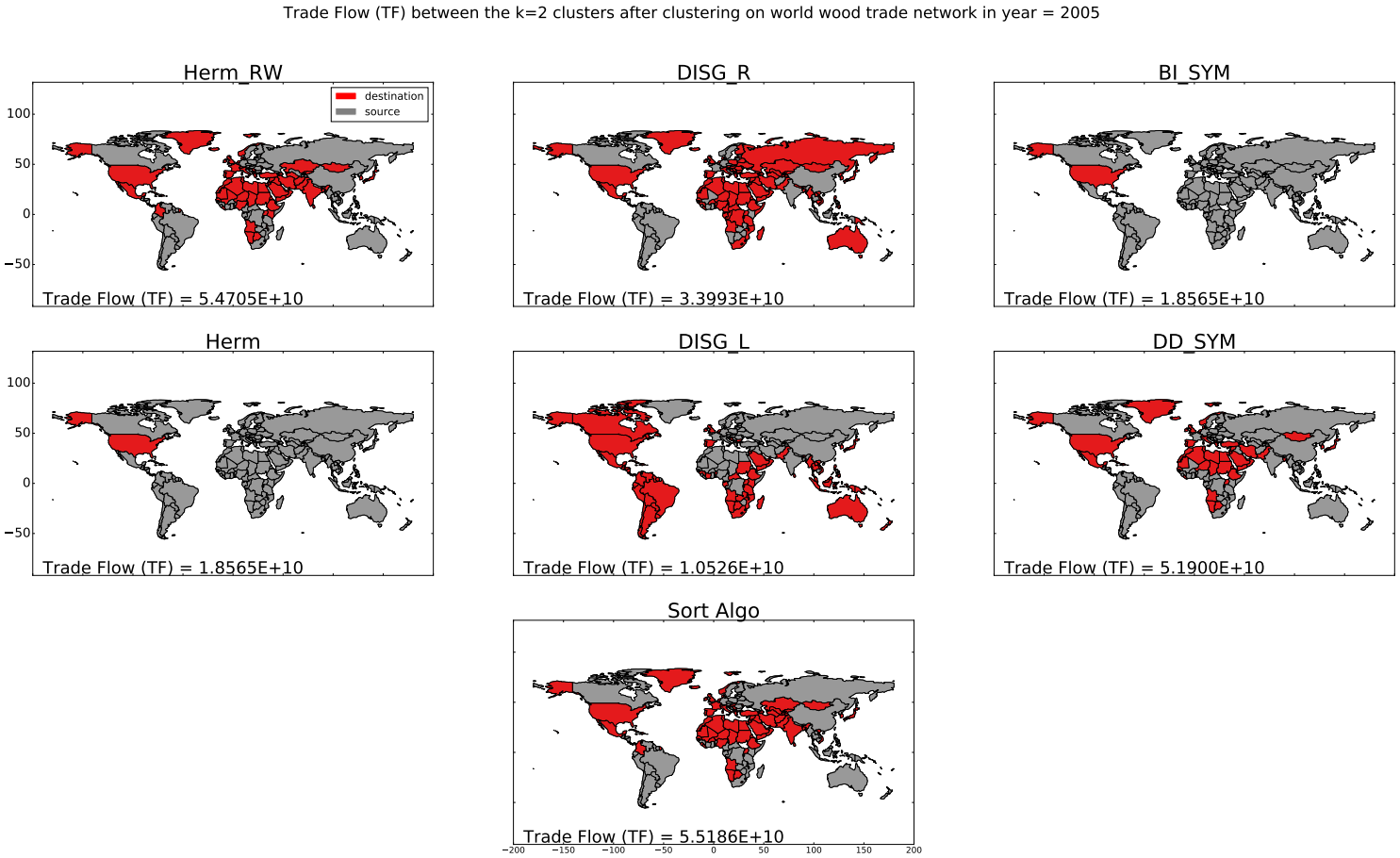

Figure B.1: Comparison of different spectral clustering results, where $k = 2$ from 6 different input matrices for the spectral clustering algorithm, and the simple max flow algorithm described in Section 4.3

B.3 Omitted Plots for Section 4.4

Since we plot the top 3 pairs, for 6 different methods over 30 years, we have 540 individual map plots. Including all of these in the pages of the appendices would become extremely long and unreadable. We therefore decided to create videos with all the plots. Our first video contains all the plots, and shows the top 3 pair in order for each year [32]. The second, third and fourth video show the progression of the top-1, top-2 and top-3 respectively for each of the six pairs [29, 30, 31].

While watching the video, we suggest using the pause and play button to stop

videos at different time-frames. This makes it easier to compare different pairs across different years. YouTube also provides the ability to move forward only one single frame in a video by pressing the full stop key: (.) and back one frame using the comma key: (,).