Constrained Directed Graph Clustering through Differential Equations

By

Ziyu Zhou

Manuela Manetta, Ph.D.
Advisor

Mathematics

2023

Constrained Directed Graph Clustering through Differential Equations

By

Ziyu Zhou

Manuela Manetta, Ph.D.
Advisor

Abstract

Constrained Directed Graph Clustering through Differential Equations
By Ziyu Zhou

This thesis introduces an algorithmic approach to the problem of clustering a weighted directed graph under constraints such as cardinality or membership requirements. The proposed approach extends a two-level iterative approach for solving weighted undirected graph minimum-cut problems to the minimum-cut and general clustering problems of directed graphs. This two-level method restates the constrained minimum-cut problems as matrix nearness problems, for which the key to numerical solutions is the gradient systems of matrix differential equations for minimizing a functional of an eigenvalue and eigenvectors of the graph Laplacians. In the proposed approach, a directed graph is transformed into an undirected one that preserves information about directionality. Numerical experiments are presented to validate the proposed approach and compare it with existing ones for unconstrained cases. The major contribution of this work is the ability to adapt a versatile technique to directed graphs and further requirements on clustering criteria.

Constrained Directed Graph Clustering through Differential Equations

By

Ziyu Zhou

Manuela Manetta, Ph.D.
Advisor

A thesis
submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors
Mathematics
2023

Acknowledgments

Words cannot express my gratitude to my advisor Dr. Manuela Manetta for her inspiring passion and priceless feedback. I also could not have undertaken this journey without the continuous support and invaluable advice from Dr. Nicola Guglielmi. Additionally, this endeavor would not have been possible without the generous support from my defense committee Dr. Dorian Arnold and Dr. Alessandro Veneziani.

Lastly, I would like to thank my friends and classmates at Emory University - Steven, Kristina, and Kai - for providing me with unfailing support and continuous encouragement.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Social networks are part of our everyday life. In math, a network (often referred to as graphs) is a set of objects (called nodes or vertices) connected together. The connections between the nodes are called edges or links. Take social networks as an example: in the Facebook social network, each vertex is a user, and the presence of an edge between two vertices reveals their friendship. Networks are at the very foundation of many everyday objects, concepts, and processes, such as transportation networks (airline flight routes), the internet, food webs, and language networks.

To represent real networks, we also need to distinguish between undirected and directed graphs. In the former, we have two-way connections between the nodes – edges have no direction (i.e., on Facebook, if A is a friend of B, B is also a friend of A); in the latter, edges have directions and indicate one-way connections (i.e., on Instagram, A follows B, but B does not follow A).

When we consider a huge dataset, organizing the graph into different modules called communities or clusters is often useful. In the same cluster, vertices present strong similarities, whereas vertices across the communities have low similarities. This procedure is called graph clustering and has been a field of interest.

In practice, it is ubiquitous that graph clustering problems come with prior constraints from background knowledge on the minimum size of clusters and membership of vertices. However, constrained clustering for directed graphs remains a developing area.

In [1], the authors present an algorithm for constrained graph minimum cut (reduction of clustering to only 2 clusters) solved as *matrix nearness problems*. In a few words, the goal is to minimize the distance between the weight matrix associated with the graph and a set of perturbed weight matrices, imposing the condition that a functional of an eigenvalue of the graph Laplacian takes its minimal value. This study aims to extend this approach to the minimum cut and general clustering problems of directed graphs.

This work is organized as follows. In Chapter 2, we present some basic definitions and results for undirected and directed graphs. In Chapter 3, we introduce the main methods, that is, *constrained minimum-cut* and *constrained clustering* for undirected graphs and we extend them to directed graphs. In Chapter 4, we provide some numerical experiments, and we summarize our work in Chapter 5.

Chapter 2

Graphs background

2.1 Undirected graphs

2.1.1 Definitions

In this section, we introduce the basic definitions of undirected graphs.

Definition 1 (Graph)**.** A graph is a pair $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$ is a set whose elements are called vertices with $|V| = n$, and $E \subset V \times V$ is a set of paired vertices whose elements are called edges.

The vertices v_i and v_j of an edge (v_i, v_j) are called the endpoints of the edge. The edge is said to connect v_i and v_j and to be incident on v_i and v_j. Moreover, v_i and v_j are said to be adjacent.

A graph G is weighted if a non-negative weight w_{ij} is assigned to each $(v_i, v_j) \in V \times V$, where $w_{ij} > 0$ if and only if $(v_i, v_j) \in E$, i.e., the vertices v_i and v_j are connected by an edge.

Definition 2 (Weighted adjacency matrix)**.** The weighted *adjacency matrix* of graph G is defined as

$$W = (w_{ij}) \in \mathbb{R}^n.$$

Definition 3 (Undirected Graph)**.** A weighted graph G is called *undirected* if $w_{ij} = w_{ji}$, with $i, j = 1, \dots, n$, i.e., the adjacency matrix W is symmetric. In an undirected graph, $(v_i, v_j) \in E$ implies $(v_j, v_i) \in E$, with $i, j = 1, \dots, n$.

In an undirected graph G, the degree of a vertex $v_i \in V$ is $d_i = \sum_{j=1}^{n} w_{ij}$.

Definition 4 (Degree matrix)**.** The *degree matrix* of G is defined as the diagonal matrix of degrees

$$D = \operatorname{diag}(d_i) = \operatorname{diag}(W\mathbb{1}),$$

where $\mathbb{1} := (1, \dots, 1)^T \in \mathbb{R}^n$.

A path in G from v_{i_0} to v_{i_ℓ} is a sequence $(v_{i_0}, v_{i_1}), (v_{i_1}, v_{i_2}), \dots, \left(v_{i_{\ell-1}}, v_{i_\ell}\right) \in E$ of arbitrary length ℓ, such that $\{v_{i_k}\}_{k=0}^{\ell}$ are all distinct and $w_coi_{k-1}, i_k > 0$ for all $k = 1, \dots, \ell$. That is, a finite or infinite sequence of edges that joins a sequence of vertices that are all distinct.

Two vertices v_i and v_j are called *connected* if G contains a path from v_i to v_j. Otherwise, they are called *disconnected*.

Definition 5 (Connected graph)**.** G is said to be *connected* if any pair of vertices of G are connected, i.e., for any pair of vertices $v_i, v_j \in V$, there exists a path from v_i to v_j.

A *subgraph* of G is another graph formed from a subset of the vertices V and edges E. The vertex subset must include all endpoints of the edge subset but may also include additional vertices. Then a *connected component* of G is a connected subgraph that is not part of any larger connected subgraph.

Definition 6 (Graph Laplacian)**.** The unnormalized graph *Laplacian matrix* of undirected graph G is defined as

$$L = \operatorname{Lap}(W) = D - W.$$

The Laplacian matrix L is symmetric and positive semi-definite [2].

2.1.2 Clustering and spectral graph theory

Graph clustering can be thought as

1. Minimum-cut problem;
2. Graph partitioning problem.

When working on clustering, we need to assume that the corresponding adjacency matrix is weighted, and we need to define a partition of a graph.

Definition 7 (Partition of G)**.** A partition of a graph G is the reduction of G to a smaller graph by partitioning its set of vertices V into mutually exclusive groups. For a group A, the indicator vector is given by

$$\mathbb{1}_A = (\mathbb{1}_{Ai}) \in \mathbb{R}^n \quad \text{with} \quad \mathbb{1}_{Ai} = \begin{cases} 1 & \text{if } v_i \in A, \\ 0 & \text{otherwise.} \end{cases}$$

The problem of clustering for undirected graphs can be formulated as finding a partition of the graph such that the edges between different partitions (clusters) have low weights and the edges within a partition (cluster) have high weights. In general, our aim is to find a number of clusters, say k, of the same graph. However, there is a special case, i.e., $k = 2$, that is referred to as the minimum-cut problem.

Spectral graph theory, as pioneered by Fiedler [3] and Chung [4], is the study of the properties of a graph in relationship to the characteristic polynomial, eigenvalues, and eigenvectors of the adjacency matrix or of the Laplacian matrix of a graph. Supported by this theory, spectral clustering is one of the modern techniques used for undirected graphs. It exploits the spectrum (eigenvalues) of the Laplacian matrix, as described in [5]. Spectral clustering has been extensively studied in data mining communities and is considered superior to traditional clustering algorithms, such as K-means [6], in terms of having deterministic and polynomial-time solutions [7].

Another effective approach for detecting clustering structure in undirected graphs is the optimization of a function known as *modularity* over the possible divisions of a graph. Informally, modularity measures the deviation in the concentration of edges in the clustered graph from that expected in the case of randomly distributed edges, which are expected to have no community structure [8, 9].

Constrained problems

In the process of finding the partitions above, there could be additional requirements. For instance, we could be asked to find a group of user A containing at least 20 Facebook users or a group of friends of user A, which necessarily contains user B. These two cases are referred to as *cardinality constraint* and *membership constraint*. These are active problems of research (see, e.g., [10, 11, 7]), as no standard methods exist, and the proposed ones are often heuristic or tailored to specific problems.

A versatile methodology based on spectral clustering, that can easily incorporate various kinds of constraints, or combinations of such constraints, in a unified way was presented in [1]. In the proposed method, the constrained partitioning problems were reformulated as *matrix nearness problems*. As the goal is to identify components of the graph that can be disconnected, in this method a system of matrix differential equations is used to lead the smallest nonzero eigenvalue of the Laplacian to zero. To this end, the distance between the graph adjacency matrix and a set of perturbed adjacency matrices must be minimized.

This algorithm features a two-level iterative procedure for matrix nearness problems, which is in common with algorithms for eigenvalue optimization via differential equations, such as given in [12, 13, 14, 15, 16].

2.2 Directed graph

2.2.1 Definitions

In this section, we extend the basic definitions of undirected graphs to directed graphs.

Definition 8 (Directed Graph)**.** A *directed graph* or *digraph* is a pair $G = (V, E)$, where $V = \{v_1, \dots, v_n\}$ is a set whose elements are called vertices with $|V| = n$, and $E \subset V \times V$ is a set of paired vertices whose elements are called edges. In a directed graph, $(v_i, v_j) \in E$ does not imply $(v_j, v_i) \in E$, with $i, j = 1, \dots, n$.

An edge $(v_i, v_j) \in E$ in a directed graph has directionality and is called as directed from v_i to v_j. The vertices v_i and v_j are called the endpoints of the edge, with v_i the tail of the edge and v_j the head of the edge. The edge is said to connect v_i and v_j and to be incident on v_i and v_j.

A directed graph G is weighted if a non-negative weight w_{ij} is assigned to each $(v_i, v_j) \in V \times V$, where $w_{ij} > 0$ if and only if $(v_i, v_j) \in V$, that is, there is an edge directed from v_i to v_j.

Definition 9 (Weighted adjacency matrix)**.** The weighted *adjacency matrix* of a directed graph G is defined as

$$W = (w_{ij}) \in \mathbb{R}^n.$$

It is crucial to notice that, in general, W is not symmetric.

Definition 10 (Degree matrix)**.** In a directed graph G, the *in-degree* of a vertex $v_i \in V$ is $d_i^{in} = \sum_{j=1}^n w_{ji}$, i.e., the sum of weights of edges directed to v_i; the *out-degree* of a vertex $v_i \in V$ is $d_i^{out} = \sum_{j=1}^n w_{ij}$, i.e., the sum of weights of edges directed from v_i. The *in-degree matrix* and *out-degree matrix* of G is defined as

$$D_{in} = \operatorname{diag}(d_i^{in}) = \operatorname{diag}(W^T \mathbb{1}) \quad \text{and} \quad D_{out} = \operatorname{diag}(d_i^{out}) = \operatorname{diag}(W \mathbb{1}).$$

The notion of connectivity can also be generalized to directed graphs, resulting in the so-called *strong* and *weak* connectivity.

Definition 11 (Strongly and weakly connected graph)**.** With the same definition of paths as for undirected graphs, a directed graph $G = (V, E)$ is *strongly connected* if for any pair of vertices $v_i, v_j \in V$, there exists a directed path from v_i to v_j (and therefore a directed path from v_j to v_i); a directed graph $G = (V, E)$ is *weakly connected* if it is connected in the undirected graph sense, i.e., ignoring the direction of the edges.

Directed graph Laplacian

Since the adjacency matrix is not symmetric, the definition of the Laplacian matrix becomes challenging for directed graphs. We can consider two Laplacians, one related to *in-edges)* and one referring to *out-edges.* Otherwise, we can find different ways to "symmetrize" the problem, depending on which method we are going to use to make an adjacency matrix symmetric. A couple of possibilities are illustrated and discussed below.

Asymmetric Laplacians. Define two distinct Laplacians, one per each degree matrix:

$$L_{in} = D_{in} - W \quad \text{and} \quad L_{out} = D_{out} - W. \tag{2.1}$$

Where L_{in} satisfies $\mathbb{1}^T L_{in} = 0$ but $L_{in}\mathbb{1}^T \neq 0$ in general and L_{out} satisfies $L_{out}\mathbb{1} = 0$, but $\mathbb{1}^T L_{out} \neq 0$ in general.

Since the Laplacians L_{in} and L_{out} are not symmetric, we expect they have complex eigenvalues, which cannot provide us with any direct physical meaning.

Symmetric Laplacian via a bipartite model. In this case, the goal is to turn a directed graph into an undirected one, by duplicating the vertices and connecting them with unoriented edges.

An undirected graph $G = (V, E)$ is said to be bipartite if $V = V_1 \cup V_2$ with V_1, V_2 such that $V_1 \cap V_2 = \emptyset$ and vertices in V_1 can be only connected with vertices in V_2 and vice versa.

Any directed graph on n vertices can be uniquely represented by a bipartite graph on $2n$ vertices as follows: let $G = (V, E)$ be the directed graph, a bipartite graph $G_b = (V_b, E_b)$ can be constructed by defining $V_b = V \cup V_1$, where $V = \{v_1, v_2, \ldots, v_n\}$, $V_1 = \{v_{n+1}, v_{n+2}, \ldots, v_{2n}\}$ and $E_b = \{(v_i, v_{j'}) \mid j' = n + j,\ (v_i, v_j) \in E\}$. Therefore, the corresponding adjacency matrix of G_b is

$$W_{bp} = \begin{bmatrix} 0 & W \\ W^T & 0 \end{bmatrix},$$

which is symmetric. Thus, it is possible to define the Laplacian as $\mathcal{L} = \mathcal{D} - W_{bp}$ where $\mathcal{D}$ is the degree matrix relative to the bipartite graph. The Laplacian, in terms of the original adjacency matrix and the out-degree and in-degree matrices, can be written as

$$\mathcal{L} = \begin{bmatrix} D_{out} & -W \\ -W^T & D_{in} \end{bmatrix}.$$

It is natural to wonder about the meaning of the eigenvalues and the eigenvectors of $\mathcal{L}$ in terms of the original directed graph.

Assuming the graph is strongly connected, then D_{out} and D_{in} are invertible. We can normalize the Laplacian as follows:

$$\hat{\mathcal{L}} = \begin{bmatrix} D_{out}^{-\frac{1}{2}} & 0 \\ 0 & D_{in}^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} D_{out} & -W \\ -W^T & D_{in} \end{bmatrix} \begin{bmatrix} D_{out}^{-\frac{1}{2}} & 0 \\ 0 & D_{in}^{-\frac{1}{2}} \end{bmatrix} = I_{2n} + \begin{bmatrix} 0 & -D_{out}^{-\frac{1}{2}} W D_{in}^{-\frac{1}{2}} \\ D_{in}^{-\frac{1}{2}} W^T D_{out}^{-\frac{1}{2}} & 0 \end{bmatrix}$$

Thus, the eigenvalues of $\hat{\mathcal{L}}$ are of the form

$$\lambda_i = 1 \mp \sigma_i \left(D_{out}^{-\frac{1}{2}} W D_{in}^{-\frac{1}{2}} \right).$$

Spectral properties of an undirected graph do not seem to be meaningful when working with a bipartite graph.

2.2.2 Directed graph clustering background

Unconstrained problem

The problem of clustering for directed graphs lacks a standard concrete definition as in the undirected cases. With the same definition of partition for undirected graphs, a high-level definition can be given as finding a partition of the graph such that each cluster (partition) can be considered as a set of vertices that share common or similar features (characteristics). It is considered more challenging compared to the undirected case for several reasons [17]:

I. **Complexity in the problem formulation.** Unlike in undirected cases where the problem is formulated in terms of weights of edges within clusters and between clusters, there are no precise and common problem formulations for directed graph clustering problems. While extending the formulation for undirected graphs to directed ones is not a trivial procedure, such density-based formulation for directed graphs clustering cannot represent the direction of edges and thus often fails to capture more sophisticated clustering structures.

II. **asymmetricity.** Many clustering results from spectral graph theory rely on the symmetricity of adjacency matrices of undirected graphs. Since adjacency matrices for directed graphs are asymmetrical, only a few methods in spectral clustering can be extended from the undirected case directly [18].

III. **Complexity in Laplacian.** Many undirected graph clustering approaches make use of graph Laplacian matrices. However, as introduced above, the notion of Laplacian becomes complicated for directed graphs due to the direction of edges.

Specifically, for the symmetric Laplacian $\mathcal{L}$ defined via the bipartite model, each vertex in the directed graph is represented by two vertices in the bipartite model, making it nontrivial to partition G exploiting properties of $\mathcal{L}$. Similarly, for the asymmetric Laplacians, L_{in} and L_{out}, the asymmetricity of L_{in} and L_{out} as well as the necessity to combine information retrieved from two Laplacians make designing a simple algorithm a challenging task.

Popular methods for directed graph clustering problems fall into two categories [17]:

I. **Transforming the digraphs to undirected weighted graphs.** In this class of methods, directed graphs are converted to undirected weighted graphs, and then appropriate undirected graph clustering approaches can be applied. In some approaches, directed graphs are transformed to undirected ones where information about directionality is introduced via weights on the edges of the graph [19], while in other approaches, the directed network is converted into a bipartite graph [20].

II. **Extending undirected graph clustering methods to directed graphs.** This class of approaches constitutes extensions of methodologies from the undirected case. Some approaches adapt the spectral clustering methods based on the Laplacian matrix to directed graphs, while another prominent class of methods extends modularity measure for undirected graphs and solves the corresponding optimization problems [21, 22].

    In particular, a generalization for directed graphs of the modularity optimization method in [9] was proposed in [21]. The author of [22] observed that this definition of modularity could not distinguish the directionality of the edges in some digraphs and proposed a new PageRank-based [23] modularity definition, which accommodates directionalities and is consistent with the undirected version of modularity. Specifically, based on their definition, given *Google Matrix* $R = (r_{ij}) \in \mathbb{R}^{n\times n}$ and its stationary vector $\pi = (\pi_i) \in \mathbb{R}^n$ (known as *PageRank vector*), the *LinkRank* matrix is

$$S = (s_{ij}) \in \mathbb{R}^{n\times n} \quad \text{where} \quad s_{ij} = \pi_i r_{ij},$$

where s_{ij} indicates the importance of the edge from i directed to j, i.e., the probability of a random surfer following this edge in the stationary state. Then the generalized modularity is constructed as

$$Q = \sum_{v_i,v_j\in V} (s_{ij} - \pi_i\pi_j)\,\delta_{ij}, \tag{2.2}$$

where $\delta_{ij} = 1$ if vertices v_i and v_j are in the same cluster and 0 otherwise. Here $\sum_{v_i,v_j\in V} s_{ij}\delta_{ij}$ denotes the fraction of time spent by random surfer while walking within communities and $\sum_{v_i,v_j\in V} \pi_i\pi_j\delta_{ij}$ denotes the expected value of this fraction in the case of randomly distributed edges.

We adopt this modularity definition and optimize it with techniques proposed in [21] in solving directed graph clustering problems to validate our proposed method in unconstrained cases. Since modularity-based methods divide graphs into more than two communities through subdividing the detected two clusters, we focus our validation on two-cluster (minimum-cut) experiments.

Constrained problem

As the directed graph clustering problem is still an active research topic, constrained directed graph clustering techniques are less investigated. [24] proposed methods tailored to object co-segmentation in computer vision, while [25] proposed a segmentation method for directed graphs that incorporates the attribute values, link structure, and prior constraints.

For this reason, the ability to extend a versatile algorithm for undirected graphs to directed graphs is crucial to research in this field.

Chapter 3

Minimum-cut and clustering methods for directed graphs

3.1 From undirected graphs to directed graphs

Consider a directed graph $G' = (\mathcal{V}, \mathcal{E}')$ with vertex set $\mathcal{V} = \{v_1, \ldots, v_n\}$ and edge set $\mathcal{E}' \subset \mathcal{V} \times \mathcal{V}$. Let the associated asymmetric adjacency weight matrix be

$$W' = (w'_{ij}) \in \mathbb{R}^{n \times n}.$$

In this section, we formulate the problem of minimum-cut and clustering applied to the directed graph G', with and without constraints. Our formulation is based on

- The reinterpretation of clusters in digraphs;
- The symmetrization of the adjacency matrix; and
- The techniques illustrated in [1].

3.1.1 Pattern-based clustering for directed graphs

As mentioned, we need to reinterpret the meaning of clustering for directed graphs. In literature, there are two main formulations: [17]:

I. **Density-based clusters**, formed based on edge density characteristics. They can be regarded as the more traditional definition of clusters in directed graphs, the most natural extension from clusters in undirected graphs.

II. **Pattern-based clusters**, formed according to similar connectivity patterns between vertices. These similarities could be lost with density-based criteria, making pattern-based clusters of more interest in directed graph clustering literature, including this study.

Different patterns can be exploited to discover clusters or community structures in directed graphs. For example, the LinkRank modularity method [22] extracts *flow-based patterns* using random walks with the intuition that when digraphs are considered as interlinked webpages, a community can be defined as a group of vertices where the random surfer is more likely to be trapped into.

However, in this work, we are interested in a specific class of patterns in directed graphs, the *co-citation* (and *co-reference*) patterns, i.e., the members of a set of vertices link to another set of vertices, and this structure implies a similarity among the members of each set. The graph in Figure 3.1 gives an example where vertices in each cluster have the same out-links or in-links to or from the same vertices.

A remark is that flow-based and co-citation patterns may co-exist in a directed graph and could align or differ, as discussed in our numerical examples in Section 4.

3.1.2 Directed graph symmetrization

To study the clustering of directed graph G' based on co-citation and co-reference patterns, we take the bibliography symmetrization approach proposed by the authors

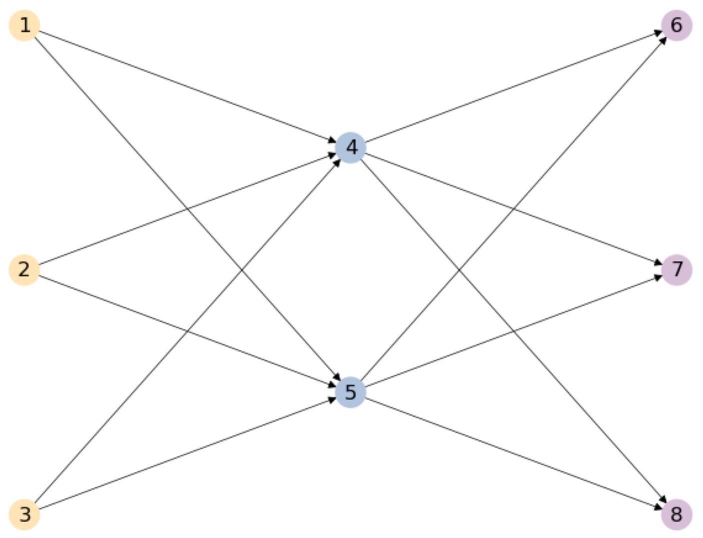


Figure 3.1: An example of a directed graph colored by clusters partitioned based on co-citation and co-reference patterns

of [19], which combines $M = W'W'^T = (m_{ij}) \in \mathbb{R}^{n\times n}$ and $N = W'^T W' = (n_{ij}) \in \mathbb{R}^{n\times n}$. m_{ij} gives the sum of weight products $w'_{ik}w'_{jk}$ where vertices v_i and v_j share common out-links (edges from v_i and v_j) to vertex $v_k \in \mathcal{V}$, and n_{ij} gives the sum of weight products $w'_{ki}w'_{kj}$ where vertices v_i and v_j share common in-links (edges to v_i and v_j) from vertex $v_k \in \mathcal{V}$. The symmetrized adjacency matrix is then defined as

$$W = M + N = (w_{ij}) = (m_{ij} + n_{ij}) \in \mathbb{R}^{n\times n}.$$

Let $G = (\mathcal{V}, \mathcal{E})$ be the graph associated with W. Since its adjacency matrix W is symmetric, G is an undirected graph.

For the graph in Figure 3.1, the symmetrized adjacency matrix is

$$W = \begin{pmatrix} \boxed{\begin{matrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{matrix}} & \begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} & \boxed{\begin{matrix} 6 & 6 \\ 6 & 6 \end{matrix}} & \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix} & \boxed{\begin{matrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{matrix}} \end{pmatrix},$$

indicating that the associated undirected graph has three connected components aligning with the clusters based on co-citation and co-reference patterns.

3.1.3 Minimum-cut and clustering formulation

Given the symmetrized adjacency matrix $W = (w_{ij}) \in \mathbb{R}^{n\times n}$ and a set of vertices $A \subset \mathcal{V}$, we denotes its complement $\mathcal{V} \setminus A$ by $\bar{A}$ and define

$$\mathcal{W}(A, \bar{A}) = \sum_{v_i \in A, v_j \in \bar{A}} w_{ij}^2 = \sum_{v_i \in A, v_j \in \bar{A}} \sum_{k=1}^{n} \left(w'_{ik} w'_{jk} + w'_{ki} w'_{kj}\right)^2 .$$

We formulate both the minimum-cut problem (two clusters) and the general clustering problems (k clusters).

In order to find the clusters based on co-citation and co-reference patterns in the directed graph G', we make use of the symmetrized undirected graph G, whose weighted edges contain connectivity information by construction.

Constrained minimum-cut problem

For the minimum-cut problem, we aim to find a partition of $\mathcal{V}$ into clusters A and its complement $\bar{A}$ such that

$$\mathcal{W}(A, \bar{A}) \quad \text{is minimized.} \tag{3.1}$$

That is, the shared out-link and in-link patterns are minimized between the two clusters. This problem is considered under the following additional constraints:

- *Membership constraint*: It is required that a given set of vertices $\mathcal{V}^+ \subset \mathcal{V}$ are contained in one cluster and another given set of vertices $\mathcal{V}^- \subset \mathcal{V}$ are contained in the other cluster.

- *Cardinality constraint*: It is required that each of the clusters has a prescribed minimum number of vertices given by $\bar{n}$.

Constrained clustering problem

For the general clustering problem, given a number of clusters k, we aim to find a partition of $\mathcal{V}$ into clusters $A_1, \ldots, A_k$ such that

$$\sum_{l=1}^{k} \mathcal{W}(A_l, \bar{A}_l) \quad \text{is minimized,} \tag{3.2}$$

under similar additional constraints:

- *Membership constraint*: It is required that for each m-th cluster of the disconnected graph A_m, a given set of vertices $\mathcal{V}^m \subset \mathcal{V}$ is in that component, $m = 1, \ldots, k$.

- *Cardinality constraint*: It is required that each of the clusters has a prescribed minimum number of vertices given by $\bar{n}$.

3.1.4 Graph Laplacian and algebraic connectivity

Given the symmetrized adjacency matrix $W = (w_{ij}) \in \mathbb{R}^{n\times n}$ and according degree matrix D, The Laplacian matrix,

$$L = \mathrm{Lap}(W) = D - W$$

is symmetric, as defined in Section 2.1.1.

The following theorem reveals the relation between the connectivity of the graph G and the eigenvalues and eigenspace of the Laplacian matrix L.

Theorem 3.1.1 ([3], [5])**.** *Let $W \in \mathbb{R}^{n\times n}$ be the adjacency matrix of an undirected graph and L the corresponding Laplacian matrix. Let $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ be the eigenvalues of L. Then the multiplicity k of the eigenvalue 0 of L equals the number of connected components on vertex subsets $A_1, \ldots, A_k$ in the graph. Then $A_1, \ldots, A_k$ partition $\mathcal{V}$ and the eigenspace of eigenvalue 0 is spanned by their indicator vectors $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$.*

In particular, when $k = 2$, the entries of the corresponding eigenvector orthogonal to $\mathbb{1}$ assume only two different values of different signs, marking the membership to the two connected components.

In other words, in order to find connected components within a graph, we need to rely on the spectral properties of the Laplacian matrix associated with the symmetrized adjacency matrix. In particular, the multiplicity of zero eigenvalues will provide us with the number of connected components, and the eigenvectors will contain information on their belonging to a certain connected component.

3.2 Constrained graph minimum-cut problem via matrix differential equations

This section introduces a two-level constrained graph minimum-cut approach proposed in [1] motivated by Theorem 3.1.1. As in [1], we first describe the method for the unconstrained problem and then extend it to constrained problems.

3.2.1 Two-level method for the minimum-cut problem

The goal is to apply the method proposed in [1] to the symmetrized graph G, which represents the properties of the directed graph G'. This method features two-level iterations:

I. (Inner iteration) Given $\varepsilon > 0$, look for a perturbation matrix $E(\varepsilon) = (e_{ij}) \in \mathbb{R}^{n \times n}$ of unit norm, such that the second smallest eigenvalue λ_2 of $Lap(W + \varepsilon E)$ is minimized.

   To ensure that the graph associated with the perturbed adjacency matrix $W + \varepsilon E$ is a valid undirected graph with the same edge set $\mathcal{E}$ as G, it is required that E is symmetric, $W + \varepsilon E \geq 0$ (entry-wise) and E shares the sparsity pattern of W.

II. (Outer iteration) Look for the smallest value among positive values of ε, call it $\varepsilon^\star$, such that the second smallest eigenvalue λ_2 of $\mathrm{Lap}(W + \varepsilon E)$ equals 0. Then two clusters of G, and thus G', are determined by the signs of the entries of the eigenvector of $\mathrm{Lap}(W + \varepsilon^\star E(\varepsilon^\star))$ corresponding to λ_2.

Below, we summarize a set of results that allow for the inner iteration, that is, the problem position leading to the matrix differential equations, and their solution. For the sake of brevity, we omit the proofs. They can be found in [1].

Inner iteration: constrained gradient flow for the functional F_ε

To compute $E(\varepsilon)$ for the given $\varepsilon > 0$, we minimize the functional

$$F_\varepsilon(E) = \lambda_2(\mathrm{Lap}(W + \varepsilon E)). \tag{3.3}$$

We denote by $\|\cdot\| = \|\cdot\|_F$ the Frobenius norm on Euclidean space and by $\langle X, Y\rangle = \mathrm{trace}(X^T Y)$ the inner product associated to the chosen norm.
Given a set of edges $\mathcal{E}$, we define the orthogonal projection from $\mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ as follows:

$$A = (a_{ij}) \mapsto P_{\mathcal{E}}(A) = (p_{ij}) \quad \text{such that} \quad p_{ij} = \begin{cases} a_{ij}, & \text{if } (v_i, v_j) \in \mathcal{E} \\ 0, & \text{otherwise.} \end{cases}$$

Given a fixed graph adjacency matrix W and an $\varepsilon > 0$, we define an ε-feasible perturbation matrix as $E = (e_{ij}) \in \mathbb{R}^{n\times n}$ such that $W + \varepsilon E$ is associated with a valid perturbed undirected graph as mentioned above, i.e., E is such that

(i) $\|E\| = 1$; (ii) $E = E^T$; (iii) $E = P_{\mathcal{E}}(E)$; (iv) $W + \varepsilon E \geq 0$.

Consider a smooth path of ε-feasible matrices $E(t)$. Let $L(t) = \mathrm{Lap}(W + \varepsilon E(t))$ be the corresponding Laplacian matrix, $\lambda_2(t)$ the second smallest eigenvalue of $L(t)$ and $x(t)$ the corresponding eigenvector with $\|x(t)\| = 1$. The following lemma describes the gradient on a path of ε-feasible matrices $E(t)$.

Lemma 3.2.1 ([1]). *For every ε-feasible matrix E, a matrix $Z = (z_{ij}) \in \mathbb{R}^{n\times n}$ is the derivative of some path of ε-feasible matrices passing through E if and only if the following conditions are satisfied*

(i) $\langle E, Z\rangle = 0$; (ii) $Z = Z^T$; (iii) $Z = P_{\mathcal{E}}(Z)$; (iv) $P_{\mathcal{E}_0}(Z) \geq 0$,

where $\mathcal{E}_0 := \{(i, j) \in \mathcal{E} : w_{ij} + \varepsilon e_{ij} = 0\}$.

To determine the admissible direction $\dot{E}$ of steepest descent of $F_\varepsilon(E)$ from E, we aim to minimize $\frac{d}{dt}F_\varepsilon(E(t)) = \dot{\lambda}_2(t)$. Using a standard perturbation result for eigenvalues (see [26]), we have (omitting argument t)

$$\dot{\lambda}_2(t) = \langle xx^T, \dot{L} \rangle. \tag{3.4}$$

We are able now to write the derivative of the eigenvalue λ_2, which is the gradient of F_ε in the space of symmetric matrices that have the sparsity pattern of W, in terms of $\dot{E}$. This is formalized in the following Lemma:

Lemma 3.2.2 ([1])**.** *Rearranging (3.4) gives*

$$\dot{\lambda}_2(t) = \varepsilon \left\langle G_\varepsilon(E(t)), \dot{E}(t) \right\rangle, \qquad \textit{with} \qquad G_\varepsilon(E) = P_{\mathcal{E}}\left(\operatorname{Sym}\left((x \bullet x)\mathbb{1}^T\right) - xx^T\right), \tag{3.5}$$

where $\operatorname{Sym}(A) = \frac{1}{2}(A + A^T)$ *and* $x \bullet x = (x_i^2) \in \mathbb{R}^n$. *The matrix* $G_\varepsilon(E)$ *is symmetric and has the sparsity pattern determined by the set of edges* $\mathcal{E}$.

Equation (3.5) indicates that the admissible direction $\dot{E} = Z$ of the steepest descent is given by

$$\arg\min_Z \langle G, Z \rangle \quad \text{s.t.} \quad \text{(i-iv) in Lemma 3.2.1 and } \langle Z, Z \rangle = 1, \tag{3.6}$$

where G denotes $G_\varepsilon(E)$. The authors of [1] proposed an equivalent optimization question

$$\arg\min_Z \langle Z, Z \rangle \quad \text{s.t.} \quad \text{(i-iv) in Lemma 3.2.1 and } \langle G, Z \rangle = -1, \tag{3.7}$$

which yields the same Karush–Kuhn–Tucker (KKT) conditions as (3.6) except for the normalization sign. Solving (3.7) gives the following results on ε-feasible gradient flow (curve following the direction of steepest descent) of F_ε.

Theorem 3.2.3 ([1]). *On an interval where $\mathcal{E}_0(t)$ does not change, the gradient flow of F_ε under (i-iv) in Lemma 3.2.1 is the system of differential equations (omitting argument t)*

$$\dot{E} = -P^+ G_\varepsilon(E) - \kappa P^+ E \quad \textit{with} \quad \kappa = \frac{\langle -G_\varepsilon(E), P^+ E\rangle}{\|P^+ E\|^2}, \tag{3.8}$$

where $P^+ = P_{\mathcal{E}\setminus\mathcal{E}_0}$. Thus by construction, for a smooth path $E(t)$ of unit Frobenius norm satisfying this system, $\dot{\lambda}_2(t) \le 0$. Moreover, $\dot{\lambda}_2(t) = 0$ if and only if $\dot{E} = 0$.

In other words, this theorem states that there is a path for $E(t)$ such that the second smallest eigenvalue λ_2 decreases, and that finding the stationary points of $E(t)$ is equivalent to finding the stationary points of $\lambda_2(t)$.

Outer iteration Newton-bisection method

Once a minimizer of the above functional is found for a fixed ε, we want to "adjust" the size of the perturbation to ultimately find $\varepsilon^\star$, such that $\lambda_2(W + \varepsilon^\star E(\varepsilon^\star)) = 0$.

Let $E(\varepsilon)$ be the minimizer of F_ε resulting from the inner iteration. Assume that for $\varepsilon < \varepsilon^\star$, the eigenvalue $\lambda_2(W + \varepsilon E(\varepsilon)) > 0$ is simple. Then $E(\varepsilon)$ is a piecewise smooth function of ε. In order to find the zeros of this function, we use Newton's method if $\varepsilon < \varepsilon^\star$ to obtain a fast iterative method to converge to $\varepsilon^\star$ from the left. Otherwise, if $\varepsilon > \varepsilon^\star$, we can use a bisection technique to get closer to $\varepsilon^\star$.

To apply Newton's method, we need to be able to express the derivative of $f(\varepsilon) = F_\varepsilon(E(\varepsilon))$. The Lemma below provides us with such a result.

Lemma 3.2.4. *Assume that the second smallest eigenvalue of* $\mathrm{Lap}(W + \varepsilon E(\varepsilon))$ *is simple. Moreover, assume $E(\varepsilon)$ to be a smooth function of ε in some interval, and the set of zero weight edges $\mathcal{E}_0$ related to $E(\varepsilon)$ is independent of ε in that interval.*

Then, the function $f(\varepsilon) = F_\varepsilon(E(\varepsilon))$ is differentiable with

$$\frac{d}{d\varepsilon} f(\varepsilon) = -\left\| P^+ G_\varepsilon(E(\varepsilon)) \right\| \left\| P^+ E(\varepsilon) \right\| - \frac{1}{\varepsilon^2} \frac{\| P^+ G_\varepsilon(E(\varepsilon)) \|}{\| P^+ E(\varepsilon) \|} \left\| P_{\mathcal{E}_0} W \right\|^2 . \tag{3.9}$$

Once the value $\varepsilon^\star$ is found, we have identified the perturbation that sends the second smallest eigenvalue of the Laplacian matrix to zero; that is, we have two disconnected components of the graph. We are now left with the task of identifying the nodes that belong to the two partitions. Supported by Theorem 3.1.1, we assign the partition A, $\bar{A}$ based on the signs of entries of the eigenvector x to the second smallest eigenvalue λ_2 of $\mathrm{Lap}(W + \varepsilon^\star E(\varepsilon^\star))$. That is, to choose

$$\begin{cases} v_i \in A & \text{if } x_i \geq 0 \\ v_i \in \bar{A} & \text{if } x_i < 0. \end{cases}$$

for each $v_i \in \mathcal{V}$.

Algorithms

In order to summarize the method and clarify its steps, we present the pseudocode relative to the inner iteration in Algorithm 1 and the outer iteration in Algorithm 2. The same pseudocode applies to the constrained problems and general clustering problems examined below.

Algorithm 1: Inner iteration: constrained gradient flow to minimize F_ε

```
Data: $W$ (symmetrized adjacency matrix), $\varepsilon$ (distance), $E^0$ (initial $E$), $t_0$, $t_f$
      and $h_0$ (start time, ending time, and step size)
Result: $f(\varepsilon) = F_\varepsilon(E(\varepsilon))$, $E(\varepsilon)$, $f'(\varepsilon)$, $Z$ (gradient flow at $E(\varepsilon)$)
if $E^0$ is not given then
    Initialize $E_0$ by the free gradient $-G_\varepsilon(0)/\|G_\varepsilon(0)\|$
end
begin
    Set $E(0) = E^0$
    Compute $\lambda_2(0)$ the second smallest eigenvalue of Lap$(W + \varepsilon E(0))$
    Compute $x(0)$ the eigenvector associated with $\lambda_2(0)$
    Compute $F_\varepsilon(E(0))$ and gradient flow $Z(0)$
    Set $t = 0$, $h = h_0$
    while $t < t_f$ do
        if $h$ is smaller than a threshold then
            Compute $f'(\varepsilon)$
            return $F_\varepsilon(E(t)), E(t), f'(\varepsilon)$ and $Z(t)$
        end
        Set $t_1 = t + h$
        Compute $E(t_1)$ given $E(t), Z(t)$ ;   /* Modified Explicit Euler */
        Compute $F_\varepsilon(E(t_1))$
        if $F_\varepsilon(E(t)) > F_\varepsilon(E(t_1))$ then              /* Step accepted */
            Compute gradient flow $Z(t_1)$
            if $F_\varepsilon(E)$ has approximately reached a stationary value then
                Compute $f'(\varepsilon)$
                return $F_\varepsilon(E(t_1)), E(t_1), f'(\varepsilon)$ and $Z(t_1)$
            end
            if the previous iteration was accepted then
                Set $h = 2h$
            end
            Set $t = t_1$
        else                                                  /* Step rejected */
            Set $h = h/2$
        end
        Compute $f'(\varepsilon)$
        return $F_\varepsilon(E(t_1)), E(t_1), f'(\varepsilon)$ and $Z(t_1)$
    end
end
```

Algorithm 2: Outer iteration: Newton-bisection for distance approximation

```
Data: $W$ (symmetrized adjacency matrix), $k_{\max}$ (maximum number of
      iterations), tol (tolerance) $\varepsilon_0, \varepsilon_{lb}$ and $\varepsilon_{ub}$ (starting values for the lower
      and upper bounds for $\varepsilon^\star$ )
Result: $\varepsilon^\star$ (approximate computed distance), partitioning result
begin
    Compute $f(\varepsilon_0), E(\varepsilon_0), f'(\varepsilon_0), Z$ by the inner iteration
    Set $k = 0$
    while $k \leq k_{\max}$ do
        if $f(\varepsilon_k) <$ tol then                          /* Bisection step */
            Set $\varepsilon_{ub} = \min(\varepsilon_{ub}, \varepsilon_k)$
            Set $\varepsilon_{k+1} = (\varepsilon_{lb} + \varepsilon_{ub})/2$
        else                                                      /* Newton step */
            Set $\varepsilon_{lb} = \max(\varepsilon_{lb}, \varepsilon_k)$
            Set $\varepsilon_{k+1} = \varepsilon_k - \frac{f(\varepsilon_k)}{f'(\varepsilon_k)}$
        end
        if $\varepsilon_{k+1} \notin (\varepsilon_{lb}, \varepsilon_{ub})$ then
            Set $\varepsilon_{k+1} = (\varepsilon_{lb} + \varepsilon_{ub})/2$
        end
        if $k = k_{\max}$ or $\varepsilon_{ub} - \varepsilon_{lb} < tol$ then
            Calculate partitioning result
            return $\varepsilon_{k+1}$, partitioning result
        end
        Approximate initial guess $E^0$ from $E(\varepsilon_k)$ and $Z$
        Set $k = k + 1$
        Compute $f(\varepsilon_k), E(\varepsilon_k), f'(\varepsilon_k), Z$ by the inner iteration with initial $E^0$
    end
    Calculate partitioning result
    return $\varepsilon_{k+1}$, partitioning result
end
```

3.2.2 Constrained minimum-cut problems

The two-level approach to the constrained minimum-cut problem has the same structure as the unconstrained minimum-cut problem. However, extra terms must be added to the functional $F_\varepsilon(E)$ defined in Equation (3.3) so that the modified $F_\varepsilon(E) = 0$ if and only if the perturbed graph is disconnected and the constraints are satisfied.

Before illustrating this change, let us introduce the following notation.

Let $x = (x_i) \in \mathbb{R}^n$ be the eigenvector to the second smallest eigenvalue λ_2 of $\mathrm{Lap}(W + \varepsilon E)$. Let $x^- = \left(x_i^-\right)$ with $x_i^- = \min(x_i, 0)$ and $x^+ = \left(x_i^+\right)$ with $x_i^+ = \max(x_i, 0)$. Then we denote the averages of entries of x^- and x^+ by

$$\langle x^- \rangle = \frac{1}{n^-}\sum_{i=1}^{n} x_i^-, \quad \langle x^+ \rangle = \frac{1}{n^+}\sum_{i=1}^{n} x_i^+,$$

where n^- and n^+ are the numbers of negative and nonnegative components of x, respectively.

Augmented functional F_ε under the membership and cardinality constraints

I. **Augmented functional for the membership-constrained minimum-cut problem**

   Given the constraint that two sets of vertices $\mathcal{V}^+, \mathcal{V}^- \subset \mathcal{V}$ are contained in two different clusters in the partitioned graph, the functional $F_\varepsilon(E)$ to be minimized reads

$$F_\varepsilon(E) = \lambda_2(\mathrm{Lap}(W + \varepsilon E)) + \frac{\alpha}{2}\sum_{v_i \in \mathcal{V}^-} \left(x_i - \langle x^- \rangle\right)^2 + \frac{\alpha}{2}\sum_{v_i \in \mathcal{V}^+} \left(x_i - \langle x^+ \rangle\right)^2, \tag{3.10}$$

   where α is a weight to be chosen. The sign of the eigenvector x is chosen so that $F_\varepsilon(E)$ takes the smaller value of the two possible values in each inner iteration.

   Theorem 3.1.1 ensures that the augmented functional $F_\varepsilon(E) = 0$ if and only if the constraints are satisfied.

II. **Augmented functional for the cardinality-constrained minimum-cut problem**

The goal is to find the minimum-cut where each partition has at least $\bar{n}$ vertices. Here, we can use the functional F_ε from the membership constraint. However, $\mathcal{V}^-$ and $\mathcal{V}^+$ are not given but are chosen depending on perturbation matrix E; that is, we define $\mathcal{V}^-$ and $\mathcal{V}^+$ as the collection of vertices to the indices of the smallest and largest $\bar{n}$ components of eigenvector x, respectively.

Gradient of F_ε under the membership and cardinality constraints

Let

$$\mathbb{1}^- = \left(\mathbb{1}_i^-\right) \in \mathbb{R}^n \quad \text{with} \quad \mathbb{1}_i^- = \begin{cases} 1 & \text{if } x_i < 0 \\ 0 & \text{otherwise,} \end{cases}$$

$$\mathbb{1}^+ = \left(\mathbb{1}_i^+\right) \in \mathbb{R}^n \quad \text{with} \quad \mathbb{1}_i^+ = \begin{cases} 1 & \text{if } x_i \geq 0 \\ 0 & \text{otherwise,} \end{cases}$$

and, with e_i denoting the i-th standard unit vector, define

$$v = v^+ + v^- \quad \text{with} \quad v^\pm = -\sum_{v_i \in \mathcal{V}^\pm} \left(x_i - \langle x^\pm \rangle\right) \left(e_i - \frac{1}{n^\pm}\mathbb{1}^\pm\right).$$

Lemma 3.2.5 ([1]). *In the above situation, for $F_\varepsilon(E)$ defined in Equation (3.10), we have*

$$\frac{d}{dt}F_\varepsilon(E) = \varepsilon \left\langle G_\varepsilon(E), \dot{E} \right\rangle, \qquad \textit{with}$$

$$G_\varepsilon(E) = P_{\mathcal{E}}\left(\mathrm{Sym}\left(\left(x \bullet (x + \alpha z)\mathbb{1}^T - x(x+\alpha z)^T\right)\right)\right),$$

where $z = (L - \lambda_2 I)^\dagger v$ († denotes Moore-Penrose pseudoinverse) and $x \bullet y = (x_i y_i)$ for $x, y \in \mathbb{R}^n$. The matrix $G_\varepsilon(E)$ is symmetric and has the sparsity pattern determined by the set of edges $\mathcal{E}$.

Applying Lemma (3.2.5) to the Newton-bisection methods completes the outer iteration algorithm for constrained problems.

3.3 Constrained clustering problem via matrix differential equations

This section extends the two-level graph minimum-cut approach to the graph clustering problem. Again, we first describe the method for the unconstrained problem and then extend it to constrained problems.

3.3.1 Two-level method for the clustering problem

Given the number of clusters k, with $3 \leq k \leq n$, in the two-level method, we replace the second smallest eigenvalue in Section 3.2.1 by the $k-$th smallest eigenvalue λ_k of the Laplacian matrix relative to the perturbed graph $W + \varepsilon E$. Theorem 3.1.1 guarantees that $\lambda_k = 0$ if and only if the perturbed graph associated with adjacency matrix $\mathrm{Lap}(W + \varepsilon E)$ has k connected components $A_1, \ldots, A_k$.

The same Theorem also suggests how to determine the clusters $A_i, \ldots, A_k$, as it states that the eigenspace of the eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$.

Let $X \in \mathbb{R}^{n \times k}$ be the matrix which contains the first k eigenvectors of $\mathrm{Lap}(W + \varepsilon^\star E(\varepsilon^\star))$ (k eigenvectors according to the smallest k eigenvalues). That is,

$$X = \Big[X_1, \ldots, X_k \Big] = (x_{ij}),$$

where X_k denotes the k-th eigenvector of $\mathrm{Lap}(W + \varepsilon^\star E(\varepsilon^\star))$ (eigenvector according to the k-th smallest eigenvalue).

The standard way to determine the clusters is to apply k-means algorithms on the rows of X (see [5]). However, for the sake of deriving the gradient flow of the augmented functional F_ε, we relaxed the problem by requiring that for each k,

$$\|X_k\| = 1 \quad \text{with} \quad \sum_{k=1}^{n} x_{ij} \geq 0,$$

and determine the clusters by the index of the maximum entry in each row of X. That is, to assign

$$v_i \in A_p \quad \text{where} \quad p = \arg\max_{1 \leq p \leq k} x_{ip}, \tag{3.11}$$

for each $v_i \in \mathcal{V}$.

Inner iteration: constrained gradient flow for the functional F_ε

For clustering problem with k clusters, to compute $E(\varepsilon)$, we minimize the functional

$$F_\varepsilon(E) = \lambda_k(\mathrm{Lap}(W + \varepsilon E). \tag{3.12}$$

Consider the same admissible path $E(t)$ as for the minimum-cut problem in Section 3.2.1, under the same denotations, by similar proof as in [1], the gradient flow of F_ε is given in the following theorem.

Theorem 3.3.1. *On an interval where $\mathcal{E}_0(t)$ does not change, the gradient flow of F_ε under (i-iv) in Lemma 3.2.1 is the system of differential equations (omitting argument t)*

$$\begin{aligned} \dot{E} &= -P^+ G_\varepsilon(E) - \kappa P^+ E \quad \textit{with} \quad \kappa = \frac{\langle -G_\varepsilon(E), P^+ E\rangle}{\|P^+ E\|^2} \\ \textit{and } G_\varepsilon(E) &= P_{\mathcal{E}}\left(\mathrm{Sym}\left((X_k \bullet X_k)\,\mathbb{1}^T\right) - X_k X_k^T\right) \end{aligned} \tag{3.13}$$

where $P^+ = P_{\mathcal{E}\setminus\mathcal{E}_0}$. Thus by construction, for a smooth path $E(t)$ of unit Frobenius norm satisfying this system, $\dot{\lambda}_k(t) \leq 0$. Moreover, $\dot{\lambda}_k(t) = 0$ if and only if $\dot{E} = 0$. The matrix $G_\varepsilon(E)$ is symmetric and has the sparsity pattern determined by the set of edges $\mathcal{E}$.

Outer iteration Newton-bisection method

The same Newton-bisection technique applies with $G_\varepsilon(E(\varepsilon))$ in $\dfrac{d}{d\varepsilon}f(\varepsilon)$ modified as in Theorem 3.3.1 and the clustering results determined as in Equation (3.11).

3.3.2 Extension for constrained clustering problems

As in the minimum-cut problem, we need to add constraints to the functional $F_\varepsilon(E)$ defined in Equation (3.12) so that the modified $F_\varepsilon(E) = 0$ if and only if the graph is clustered into k connected components and the prescribed constraints are satisfied.

Augmented functional F_ε under the membership and cardinality constraints

I. **Augmented functional for the membership-constrained clustering problem**

   Given the constraint that k sets of vertices $\mathcal{V}_m$ for $m = 1, \ldots, k$ are in k different clusters in the clustered graph, we augment $F_\varepsilon(E)$ as

$$F_\varepsilon(E) = \lambda_k(\mathrm{Lap}(W + \varepsilon E)) + \alpha \sum_{1 \le m \le k} \sum_{v_i \in \mathcal{V}_m} \left(\max_{1 \le j \le k} x_{ij} - x_{im} \right), \tag{3.14}$$

   where α is a weight to be chosen. In particular, $V_1, \ldots, V_m$ are rearranged in each inner iteration so that $F_\varepsilon(E)$ takes the smallest value.

II. **Augmented functional for the cardinality-constrained clustering problem**

   The goal is to find k clusters with the condition that each cluster has at least $\bar{n}$ vertices. Here, we use the functional F_ε from the membership constraint. However, $\mathcal{V}_m$ are not given but are chosen depending on perturbation matrix E greedily: for each m, we determine $\mathcal{V}_m$ as the collection of vertices to the indices of the largest $\bar{n}$ components of eigenvector X_m that have not been assigned to some $\mathcal{V}_j$ for $1 \leq j \leq m$.

Gradient of F_ε under the membership and cardinality constraints

Given an m with $1 \leq m \leq k$, for an $v_i \in \mathcal{V}_m$, let m_i be the index of the maximum entry in the i-th row of X, i.e.,

$$m_i = \arg\max_{1\leq j\leq k} x_{ij}.$$

Then with e_i denoting the ith standard unit vector, we define

$$Z_{mi} = (K_m e_i \bullet X_m^T)\mathbb{1}^T - (K_{m_i} e_i \bullet X_{m_i}^T)\mathbb{1}^T - K_m e_i X_m^T + K_{m_i} e_i X_{m_i}^T \in \mathbb{R}^{n\times n},$$

where $K_j = (L - \lambda_j I)^\dagger$ denotes the Moore-Penrose pseudoinverse of $(L - \lambda_j I)$ for each $j = 1, \ldots, k$.

Lemma 3.3.2. *In the above situation, for $F_\varepsilon(E)$ defined in Equation (3.15), we have*

$$\frac{d}{dt} F_\varepsilon(E) = \varepsilon \left\langle G_\varepsilon(E), \dot{E} \right\rangle, \qquad \textit{with}$$
$$G_\varepsilon(E) = P_{\mathcal{E}}\left(\mathrm{Sym}\left((X_k \bullet X_k)\,\mathbb{1}^T - X_k X_k^T + \sum_{1\leq m\leq k} \sum_{v_i\in\mathcal{V}_m} Z_{mi}\right)\right).$$

The matrix $G_\varepsilon(E)$ is symmetric and has the sparsity pattern determined by the set of edges $\mathcal{E}$.

$$F_\varepsilon(E) = \lambda_k(\mathrm{Lap}(W + \varepsilon E)) + \alpha \sum_{1\leq m\leq k} \sum_{v_i\in\mathcal{V}_m} \left(\max_{1\leq j\leq k} x_{ij} - x_{im}\right), \tag{3.15}$$

Proof. For each m for $m = 1, \ldots, k$, we have

$$\frac{d}{dt} \sum_{v_i\in\mathcal{V}_m} \left(\max_{1\leq j\leq k} x_{ij} - x_{im}\right) = \sum_{v_i\in\mathcal{V}_m} \left(\frac{d}{dt}\max_{1\leq j\leq k} x_{ij} - \dot{x}_{im}\right).$$

With the pseudoinverses of $(L - \lambda_j I)$ defined above and the standard results for eigenvector derivatives [27], we obtain that

$$\frac{d}{dt} \max_{1 \le j \le k} x_{ij} = -e_i^T K_{m_i} \dot{L} X_{m_i}, \quad \dot{x}_{im} = -e_i^T K_m \dot{L} X_m,$$

so that

$$\begin{aligned} \frac{d}{dt} \sum_{v_i \in \mathcal{V}_m} \left(\max_{1 \le j \le k} x_{ij} - x_{im} \right) &= \sum_{v_i \in \mathcal{V}_m} \left(e_i^T K_m \dot{L} X_m - e_i^T K_{m_i} \dot{L} X_{m_i} \right) \\ &= \sum_{v_i \in \mathcal{V}_m} \langle K_m e_i X_m^T - K_{m_i} e_i X_{m_i}^T, \dot{L} \rangle. \end{aligned}$$

Then proceed as in the proof of Lemma 3.2.2 given in [1] to obtain the result. □

Applying Lemma (3.3.2) to the Newton-bisection methods completes the outer iteration algorithm for constrained problems.

In the following Chapter, we illustrate some numerical examples of all the methods above, applied to both undirected graphs (to validate the method, comparing results with [1]), and directed graphs.

Chapter 4

Numerical experiments

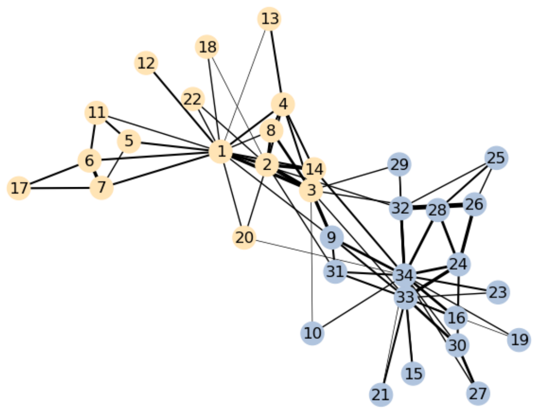
Numerical experiments are based on the implementation of the algorithms presented in the previous Chapter. The methods are coded entirely in Python (version 3.8) and the experiments are performed on a personal laptop. In the following, we present three examples. Example 1 is used to validate the codes and to compare the results with those of [1]. Indeed, the analyzed graph is undirected. In Example 2, we consider a community-structured graph generated according to [18]. In Example 3, we analyze a directed graph representing a subset of the United States Airline network.

Example 1 (Zachary's Karate club)

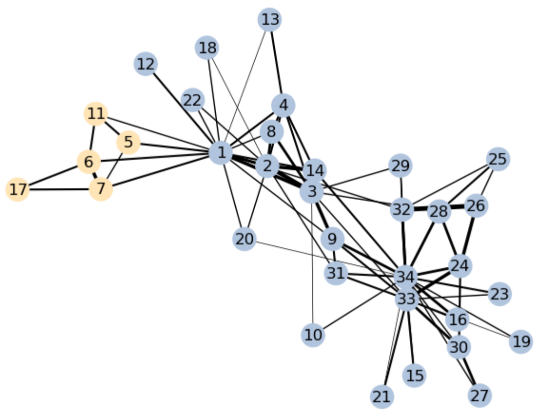
Zachary's Karate club is a graph describing the relation between 34 members of a karate club [28]. We aim to replicate the results from [1] to validate our code in minimum-cut undirected graphs under constraints.

(i) **Unconstrained minimum-cut**: With no constraints, we obtain two clusters of size 16 and 18 using the partition algorithm, as shown in Figure 4.1a. The approximate computed distance is $\varepsilon^\star = 10.9545$.

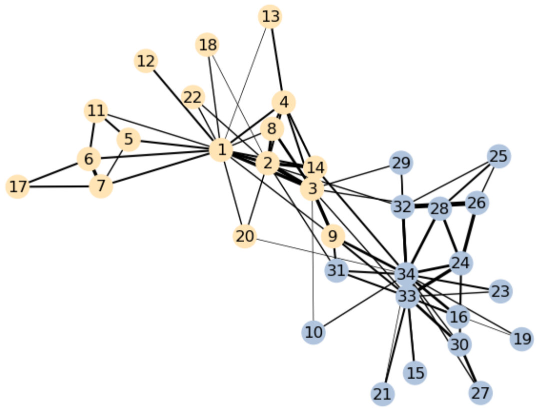
(ii) **Minimum-cut under membership constraints:** We aim to have a partition



(a) Zachary's karate club graph colored by unconstrained minimum-cut result.



(b) Zachary's karate club graph colored by minimum-cut result when requiring that vertices 14 and 34 to belong to the same cluster.



(c) Zachary's karate club graph colored by minimum-cut result when requiring each cluster to have at least 17 vertices.

Figure 4.1: Example 1: Zachary's karate club minimum-cut.

of the graph with specific vertices in each cluster. Four examples are shown in Table 4.1. Here, tol $= 10^{-5}$ and initial weight $\alpha = 3$. The results show that it is easiest to require vertex 9 to change its cluster, whereas it is most difficult to force vertex 14 to another cluster. As an example, Figure 4.1b shows the minimum-cut result when vertices 14 and 34 are in the same cluster.

(iii) **Minimum-cut under cardinality constraints:** Requiring to have a partition of the graph with $\bar{n} = 17$ vertices in each cluster, we obtain a partition as in Figure 4.1c with approximate computed distance $\varepsilon^\star = 69.0994$ with tol $= 10^{-5}$

Constraints	$\varepsilon^\star$
vertices 9 and 1 in the same partition	9.4212
vertices 14 and 34 in the same partition	14.6566
vertices 20 and 34 in the same partition	14.1860
vertices 32 and 1 in the same partition	13.5686

Table 4.1: Computed values of approximate computed distance $\varepsilon^\star$ for each membership constraint, Example 1 (ii).

and initial weight $\alpha = 0.3$.

Example 2 (Generated benchmark)

Here, we study a benchmark weighted directed graph with built-in community structure generated by the algorithm proposed in [29]. The digraph has 3 built-in communities, where vertices 13 and 14 belong to two communities, as shown in Figure 4.2. We compared the unconstrained minimum-cut result with the modularity-based LinkRank approach proposed in [22], as introduced in Section 2.2.2.

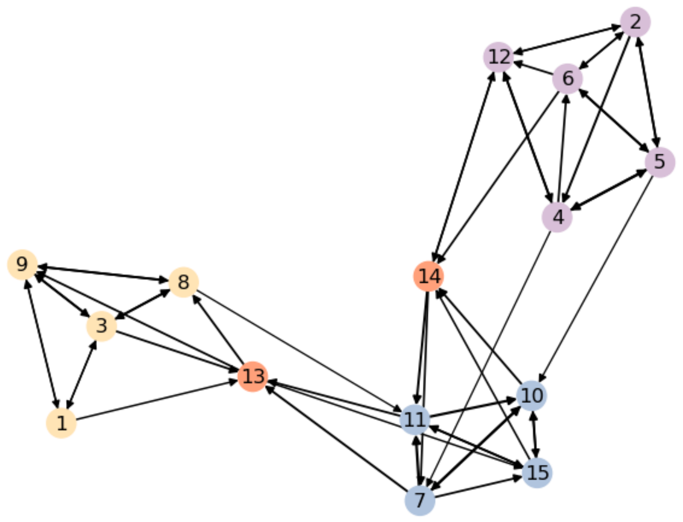


Figure 4.2: Generated benchmark digraph colored by built-in community, where vertices 13 and 14 belong to both communities near them.

Minimum-cut

(i) **Unconstrained minimum-cut:** With no constraints, we obtain two clusters of size 5 and 10, as shown in Figure 4.3a. The approximate computed distance is $\varepsilon^\star = 58.0345$. One community is detected, while the other two are included in the same cluster.

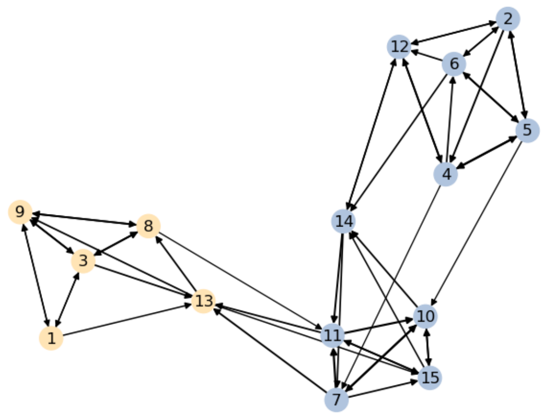
(ii) **LinkRank**: Applying the modularity-based method, we obtain the same clusters as our methods, as shown in Figure 4.3a.

(iii) **Minimum-cut under membership constraints:** We consider the case that vertex 7 to be in the same cluster as vertex 10, or vertex 7 to be not in the same cluster as vertex 4. By setting a tol $= 10^{-5}$ and initial weight $\alpha = 3$, we obtain approximate computed distances $\varepsilon^\star = 79.8658$ and $\varepsilon^\star = 77.7294$, respectively. Figure 4.3b and 4.3c present the constrained minimum-cut results. The applied constraint forced the algorithm to detect a different built-in community from the unconstrained case.

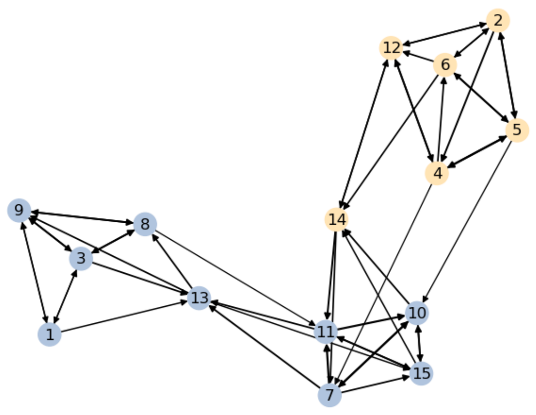
(iv) **Minimum-cut under cardinality constraints:** Requiring to have a partition of the graph with at least $\bar{n} = 6$ vertices in each cluster, we obtain a partition as in Figure 4.3d with approximate computed distance $\varepsilon^\star = 146.9121$ by setting tol $= 10^{-5}$ and initial weight $\alpha = 9$.

Clustering

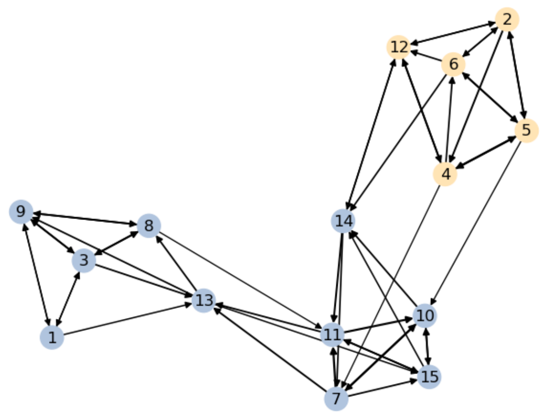
(i) **Unconstrained clustering with $k = 3$:** With no constraints applied, we obtain three clusters of size 4, 5, and 6, as shown in Figure 4.4a. The approximate computed distance is $\varepsilon^\star = 78.0150$, and the clusters align with the built-in communities.



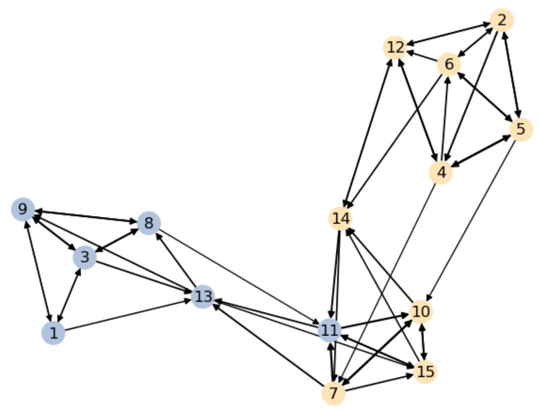
(a) Generated benchmark digraph colored by unconstrained minimum-cut and LinkRank clustering results.



(b) Generated benchmark digraph colored by minimum-cut result when requiring vertices 7 and 10 to belong to the same cluster.



(c) Generated benchmark digraph colored by minimum-cut result when requiring vertices 4 and 7 not to belong to the same cluster.

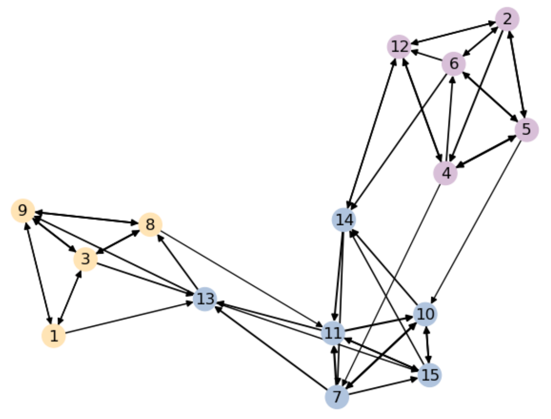


(d) Generated benchmark digraph colored by minimum-cut result when requiring each cluster to have at least 6 vertices.

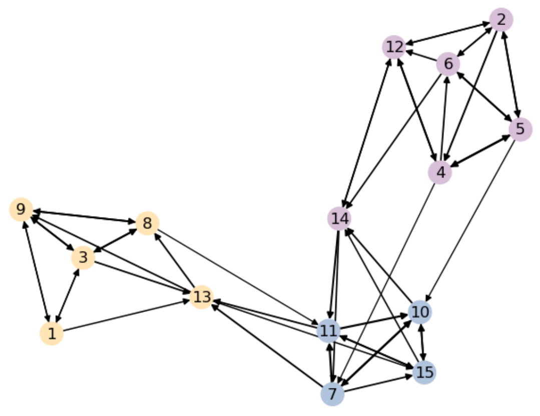
Figure 4.3: Example 2: Generated benchmark digraph minimum-cut.

(ii) **Clustering with $k = 3$ under membership constraints:** We require vertices 13 and 14 not in the same component, with tol $= 10^{-5}$ and initial weight $\alpha = 3$. Figure 4.4b presents the clustering results with the applied constraint, and the approximate computed distance is $\varepsilon^\star = 107.5529$.

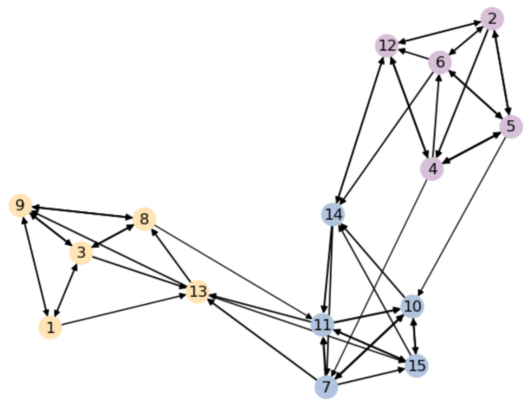
(iii) **Clustering with $k = 3$ under cardinality constraints:** Requiring to have a clustering of the graph with $\bar{n} = 5$ vertices in each cluster, we obtain the partition in Figure 4.4c with approximate computed distance $\varepsilon^\star = 111.3178$ by setting tol $= 10^{-5}$ and initial weight $\alpha = 3$.



(a) Generated benchmark digraph colored by unconstrained clustering result.



(b) Generated benchmark digraph colored clustering result when requiring vertices 13 and 14 not to belong to the same cluster.



(c) Generated benchmark digraph colored by clustering result when requiring each cluster to have at least 5 vertices.

Figure 4.4: Example 2: Generated benchmark digraph clustering.

Example 3 (United States Airline network)

Here, we study a directed graph representing the 50 domestic flight routes with the largest total number of passengers in the United States in 2019 based on Bureau of Transportation Statistics data [30]. Figure 4.5 presents the directed graph, where the nodes denote cities in the United States, and the edges denote flight routes weighted by the number of total passengers over 2019 (divided by $500,000$ for normalization).

Still, we compared the unconstrained minimum-cut result with the modularity-based proposed.

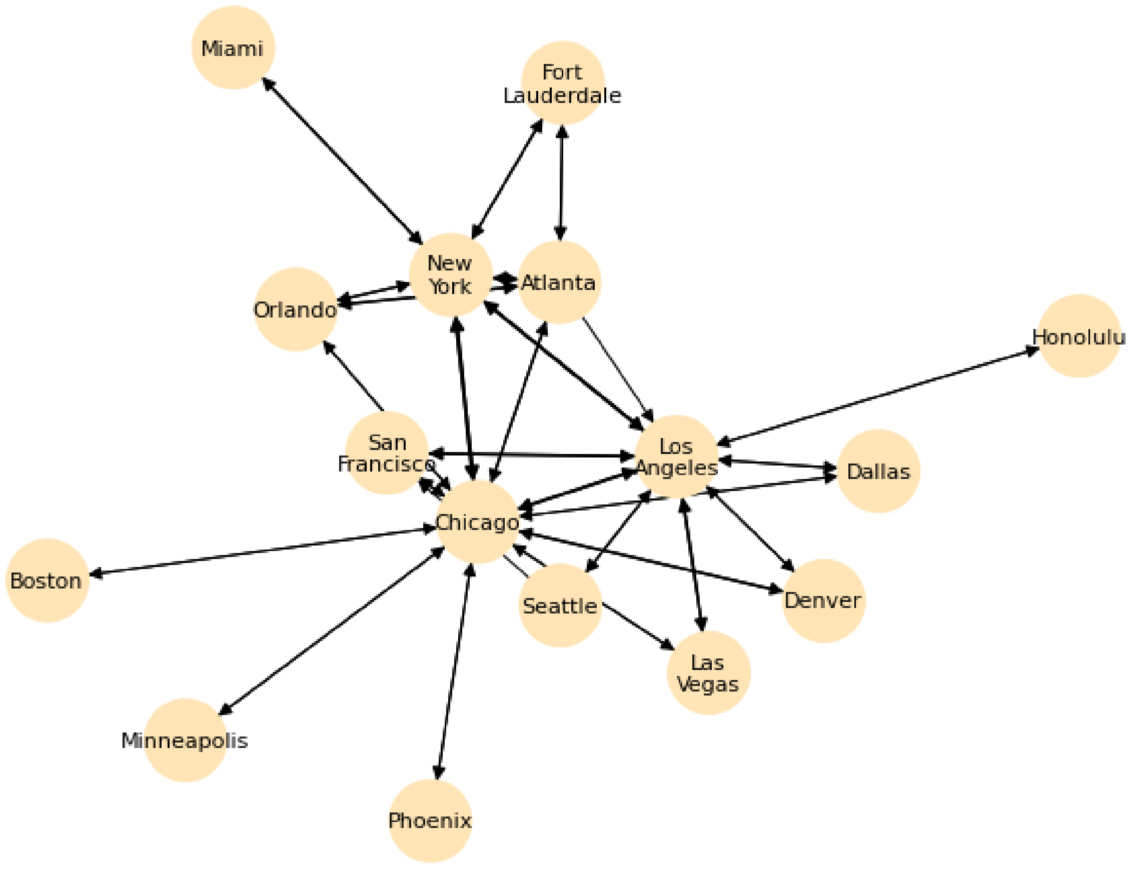


Figure 4.5: United States Airline network.

Minimum-cut

(i) **Unconstrained minimum-cut:** With no constraints, we obtain two clusters of size 1 and 15, as shown in Figure 4.3a. The approximate computed distance is $\varepsilon^{\star} = 58.5066$. Such unbalanced results appear since the optimization goal defined in Equation (3.1) didn't consider the size of partitions. As introduced in [5], we can explicitly request the sets $A_1, \ldots, A_k$ to be "reasonably large" by modifying the optimization goal and using the normalized Laplacian matrices in the two-level method [1].

(ii) **LinkRank**: Applying the modularity-based method, which detects a more balanced cluster than our methods, as shown in Figure 4.6b.

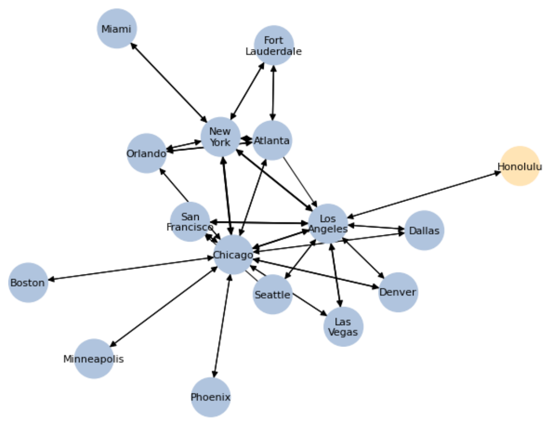
(iii) **Minimum-cut under membership constraints:** We require vertices New York and Los Angeles to be not in the same cluster, to partition the vertices by their similarity in flight patterns with the two metropolitan cities. By setting a tol $= 10^{-5}$ and initial weight $\alpha = 3$, we obtain the constrained minimum-cut result presented in Figure 4.6c, with approximate computed distances $\varepsilon^\star = 293.6063$.

(iv) **Minimum-cut under cardinality constraints:** Requiring to have a partition of the graph with $\bar{n} = 5$ vertices in each cluster, we obtain a partition as in Figure 4.6d with approximate computed distance $\varepsilon^\star = 263.2201$ by setting tol $= 10^{-5}$ and initial weight $\alpha = 0.8$.

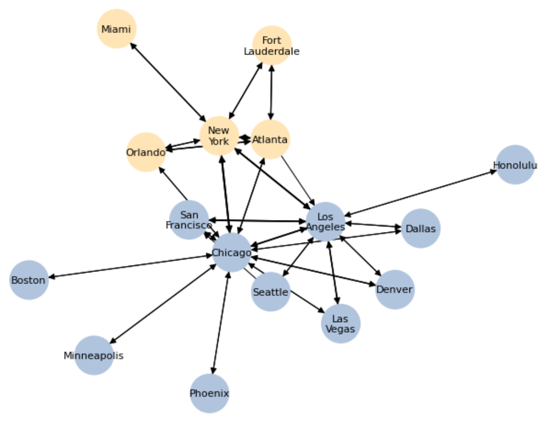
We observe that clusters our method detected when clusters are forced to be relatively balanced to be close to the result of the modularity-based method. We analyze that the difference appears in that the modularity method simulates directed graphs as web pages with interlinks in the World Wide Web and finds a group of webpages (vertices) where a random surfer is more likely to stay as a cluster based on the definition of modularity in Equation (2.2), while our methods divide vertices with similar connectivity patterns into a cluster based on the symmetrization technique applied.

Clustering

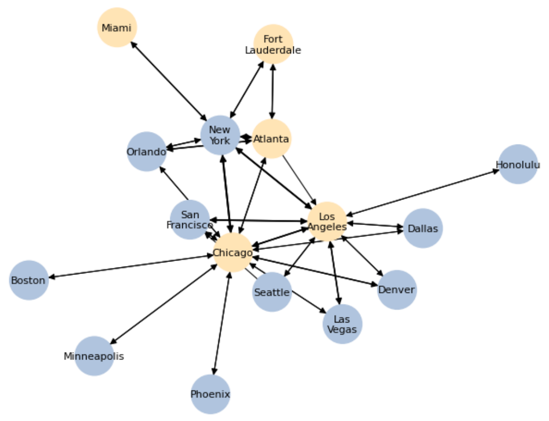
(i) **Unconstrained clustering with $k = 3$:** With no constraints applied, we obtain three clusters of size 1, 3, and 12, as shown in Figure 4.7a. The approximate computed distance is $\varepsilon^\star = 123.4132$. Similar to the minimum-cut problem, we



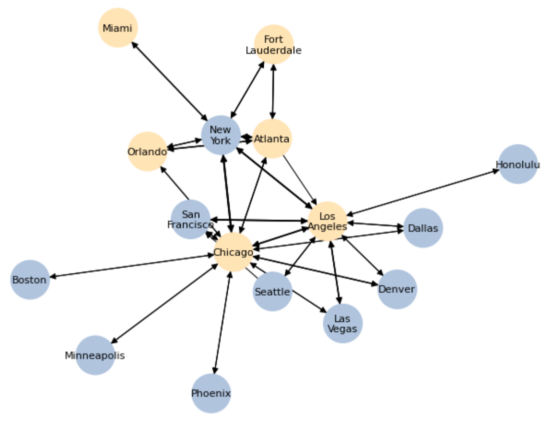
(a) Airline digraph colored by unconstrained minimum-cut result.



(b) Airline digraph colored by colored by LinkRank clustering results.



(c) Airline digraph colored by minimum-cut result when requiring vertices New York and Los Angeles not to belong to the same cluster.



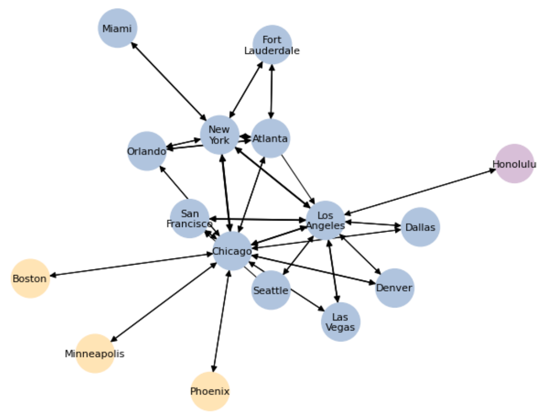
(d) Airline digraph colored by minimum-cut result when requiring each cluster to have at least 7 vertices.

Figure 4.6: Example 3: Airline network minimum-cut.

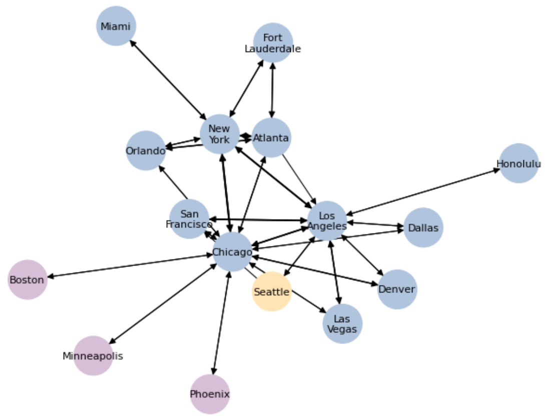
can obtain clusters of more balanced sizes by using normalized clusters in the two-level algorithm.

(ii) **Clustering with $k = 3$ under membership constraints:** We require vertices Atlanta and Chicago not in the same component, with tol $= 10^{-5}$ and initial weight $\alpha = 10$. Figure 4.7b presents the clustering results with the applied constraint, and the approximate computed distance is $\varepsilon^\star = 363.9693$.

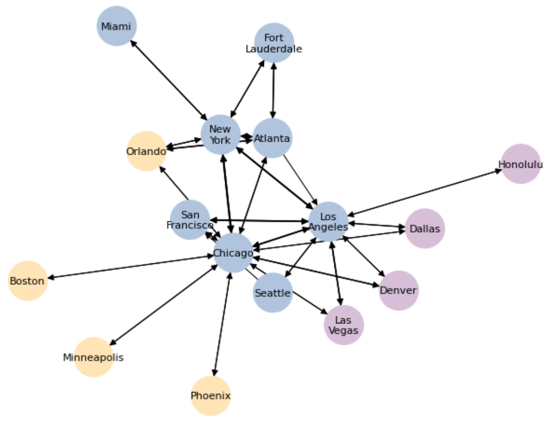
(iii) **Clustering with $k = 3$ under cardinality constraints:** Requiring to have a clustering of the graph with $\bar{n} = 4$ vertices in each cluster, we obtain the partition in Figure 4.7c with approximate computed distance $\varepsilon^\star = 450.6817$ by setting $\mathrm{tol} = 10^{-5}$ and initial weight $\alpha = 10$.



(a) Airline digraph colored by unconstrained clustering.



(b) Airline digraph colored clustering result when requiring vertices Atlanta and Chicago not to belong to the same component.



(c) Airline digraph colored clustering result when requiring each component to have at least 4 vertices.

Figure 4.7: Example 3: Airline network clustering.

Chapter 5

Conclusions

We generalize an algorithmic approach to solving constrained minimum-cut problems for undirected graphs to the constrained minimum-cut problems for directed graphs through a graph symmetrization technique which introduces the information about co-citation and co-reference patterns in directed graphs via the weights on the edges of the symmetrized graphs.

We follow a two-level iterative approach that solves constrained minimum-cut problems as matrix nearness problems: it minimizes the distance between the adjacency matrix of a graph and a set of perturbed graphs, imposing the constraints by forcing a functional of the adjacency matrix to take its minimal value. The main computational cost can be found in the solution of eigenvalue problems of symmetric positive semi-definite matrices, which is determined by available numerical linear algebra routines, making this algorithm capable of exploiting the sparsity of large graphs.

Moreover, we generalize this approach from constrained minimum-cut problems to general constrained clustering problems with more than 2 clusters by exploiting the eigenvalues and eigenspace of the perturbed adjacency matrix.

We compare our methods in unconstrained cases with a modularity-based directed graph clustering method which aims to detect communities in the directed graph by flow-based patterns instead of connectivity patterns (co-citation and co-reference) for our methods. Through numerical experiments, we observe that both patterns could co-exist in directed graphs, and the communities detected could align or defer.

Regarding future studies, we aim to tackle the limitations of our proposed approach and extend the scope of this study, as elaborated below.

I. As demonstrated in the numerical examples, utilizing unnormalized Laplacians can result in unbalanced communities. We aim to modify the two-level algorithm to make use of the normalized Laplacian matrices and investigate the result.

II. The two-level approach features versatility in constraints: it can easily incorporate constraints other than cardinality and membership constraints, including must-link (requiring pairs of vertices must be in the same cluster), cannot-link (requiring pairs of vertices must not be in the same cluster), and combinations of multiple constraints through modifying the augmented functionals to be minimized. We aim to extend our scope and incorporate more constraints for directed graph clustering problems.

III. Though performing remarkably well on extensive problems from literature, the two-level algorithm cannot guarantee finding the global minimum of non-smooth, non-convex optimization problems or NP-hard combinatorial optimization problems. [1] presents an example where it gets stuck in a local minimum, and we aim to explore similar caveats in directed cases and analyze different algorithms' performance.

Bibliography

[1] E. Andreotti, D. Edelmann, N. Guglielmi, and C. Lubich, "Constrained graph partitioning via matrix differential equations," *SIAM Journal on Matrix Analysis and Applications*, vol. 40, pp. 1–22, 01 2019.

[2] B. Mohar, "Laplace eigenvalues of graphs—a survey," *Discrete Mathematics*, vol. 109, no. 1, pp. 171–183, 1992.

[3] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[4] F. R. K. Chung, *Spectral Graph Theory.* American Mathematical Society, 1997.

[5] U. von Luxburg, "A tutorial on spectral clustering," 2007.

[6] J. MacQueen, "Some methods for classification and analysis of multivariate observations," 1967.

[7] X. Wang, B. Qian, and I. Davidson, "On constrained spectral clustering and its applications," *Data Mining and Knowledge Discovery*, vol. 28, pp. 1–30, sep 2012.

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

[9] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[10] S. Basu, M. Bilenko, and R. J. Mooney, "A probabilistic framework for semi-supervised clustering," KDD '04, (New York, NY, USA), p. 59–68, Association for Computing Machinery, 2004.

[11] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, (San Francisco, CA, USA), p. 577–584, Morgan Kaufmann Publishers Inc., 2001.

[12] N. Guglielmi and C. Lubich, "Differential equations for roaming pseudospectra: Paths to extremal point and boundary tracking," *SIAM Journal on Numerical Analysis*, vol. 49, no. 3/4, pp. 1194–1209, 2011.

[13] N. Guglielmi, D. Kressner, and C. Lubich, "Low rank differential equations for hamiltonian matrix nearness problems," *Numerische Mathematik*, vol. 129, 02 2015.

[14] N. Guglielmi and C. Lubich, "Matrix stabilization using differential equations," *SIAM Journal on Numerical Analysis*, vol. 55, no. 6, pp. 3097–3119, 2017.

[15] N. Guglielmi, C. Lubich, and V. Mehrmann, "On the nearest singular matrix pencil," *SIAM Journal on Matrix Analysis and Applications*, vol. 38, no. 3, pp. 776–806, 2017.

[16] E. Andreotti, D. Edelmann, N. Guglielmi, and C. Lubich, "Measuring the stability of spectral clustering," *Linear Algebra and its Applications*, vol. 610, pp. 673–697, 2021.

[17] F. D. Malliaros and M. Vazirgiannis, “Clustering and community detection in directed networks: A survey,” *Physics Reports*, vol. 533, no. 4, pp. 95–142, 2013.

[18] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[19] V. Satuluri and S. Parthasarathy, “Symmetrizations for clustering directed graphs,” in *Proceedings of the 14th International Conference on Extending Database Technology*, EDBT/ICDT ’11, (New York, NY, USA), p. 343–354, Association for Computing Machinery, 2011.

[20] D. Zhou, T. Hofmann, and B. Schölkopf, “Semi-supervised learning on directed graphs,” in *Advances in Neural Information Processing Systems* (L. Saul, Y. Weiss, and L. Bottou, eds.), vol. 17, MIT Press, 2004.

[21] E. A. Leicht and M. E. J. Newman, “Community structure in directed networks,” *Phys. Rev. Lett.*, vol. 100, p. 118703, Mar 2008.

[22] Y. Kim, S.-W. Son, and H. Jeong, “Finding communities in directed networks,” *Physical Review E*, vol. 81, no. 1, 2010.

[23] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking : Bringing order to the web,” in *The Web Conference*, 1999.

[24] F. Meng, H. Li, S. Zhu, B. Luo, C. Huang, B. Zeng, and M. Gabbouj, “Constrained directed graph clustering and segmentation propagation for multiple foregrounds cosegmentation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 11, pp. 1735–1748, 2015.

[25] Y. C. Yang, Z. Qi, H. Liu, and J. He, “Constrained clustering based on the link structure of a directed graph,” in *Pacific Asia Conference on Information Systems*, 2015.

[26] T. Kato, *Perturbation Theory for Linear Operators.* 1966.

[27] C. D. Meyer and G. W. Stewart, “Derivatives and perturbations of eigenvectors,” *SIAM Journal on Numerical Analysis*, vol. 25, no. 3, pp. 679–691, 1988.

[28] W. Zachary, “An information flow model for conflict and fission in small groups1,” *Journal of anthropological research*, vol. 33, 11 1976.

[29] A. Lancichinetti and S. Fortunato, “Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities,” *Physical Review E*, vol. 80, no. 1, 2009.

[30] “Data bank 28dm - t-100 domestic market data - u.s. air carriers traffic and capacity data,” Mar 2020.