Spectral Modularity

Conceptual Challenges, Algorithmic Enhancements and Systematic Benchmarking

V.R. Bockstael

Technische Universiteit Delft

TU Delft

Spectral Modularity: Conceptual Challenges, Algorithmic Enhancements and Systematic Benchmarking

V.R. Bockstael

Abstract Cluster analysis in high dimensional data is a difficult but desirable task. Many existing methods fail to cluster high dimensional data due to what is known as the curse of dimensionality. Therefore, sophisticated clustering methods are in wide development. Along these lines, spectral modularity maximization emerged from the theory of random matrices and graph modularity. The method is based on a filtering of the spectral decomposition of similarity matrices. Despite the recent success of this method, we uncover a fundamental challenge of spectral modularity: the spectral modularity breaks down as the number of groups in a data set grows. To mitigate this challenge, we propose two solutions: one solution based on a regularization and one solution based on a normalization. We perform a thorough empirical analysis of the clustering performance of the solutions and find that, not only do our methods resolve the breakdown of spectral modularity, but they also outperform existing clustering methods in a variety of settings.

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday, August 28, 2024 at 15:00.

Student number:	4590694	
Project duration:	November 13, 2023 – August 28, 2024	
Thesis committee:	Dr. N. Yorke-Smith,	TU Delft, responsible supervisor
	Dr. A.I. Băbeanu,	TU Delft, daily supervisor
	Dr. H. Wang,	TU Delft

This thesis is confidential and cannot be made public until August 28, 2024.

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Contents

1 Introduction

With today's digital society, the ability to store and collect large amounts of data has become easier than ever. The utility of this data is explored on a large scale in the supervised learning paradigm. The benefit of machine learning with such clear objectives is ubiquitous in scientific communities as well as industry. On the other hand, in many areas of research, the precise purpose of the collected data is still ambiguous. These problems are encapsulated in the field of exploratory data analysis.

An essential task in exploratory data analysis is clustering. Clustering can be described as grouping the objects in a dataset, so that objects within the same group are more similar [1]. Therefore, clustering is a computational tool that is used to find patterns in data that give insight into the structure of the underlying system in which the data is observed. Typically, these insights are used to determine further research directions.

Specifically, in this study, we put an emphasis on data that is high dimensional, which adds a significant array of difficulties to the clustering problem. These challenges that are a consequence of high dimensionality are commonly known as the curse of dimensionality [2, 3]. Application areas where research and business practice heavily rely on clustering of high dimensional data are item response analysis [4], financial market research [5], population genetics [6], single-cell sequencing [7], and many more.

MacMahon and Garlaschelli [8] developed a seminal method for clustering high dimensional multivariate data. The method utilizes theory of random matrices [9, 10, 11] and theory of network modularity [12] in an emergent way. Conceptually, this method is based on the filtering of noise from a matrix through the use of the spectral decomposition of the observed matrix compared with that of a null model random matrix. The filtering implicitly reduces the data dimensionality in a sophisticated way, as it only retains a smaller number of informative dimensions. This makes the method especially useful for clustering high dimensional data. From now on, in this thesis, we refer to this method as spectral modularity maximization.

Because of the reliance on the spectral decomposition, it is tempting to think that this spectral modularity maximization might be equivalent to spectral clustering. Although it aligns with the general philosophy of the spectral clustering paradigm [13, 14, 15, 16], spectral modularity maximization is significantly different from existing work in that setting. In particular, the modularity inspired objective in spectral modularity maximization, that is originally used for community detection in networks [12], gives a different perspective on the clustering problem. Many of the methods from the spectral clustering paradigm focus on clusterings that optimally separate clusters, while modularity based methods focus on clustering objects that have higher than expected similarity.

Within the spectral clustering paradigm, random matrix theory has been used fruitfully to explain the suitability of using a spectral decomposition in the context of high dimensional data [17, 18, 19, 9]. Furthermore, the concept of filtering the spectral decomposition that is done in spectral modularity borrows ideas from the wide variety of statistical applications of random matrix analysis, which has been previously applied in denoising single-cell data [20], improving memory-based recommendation systems [21], studying cross correlations of financial markets [22, 23, 24, 25, 26, 27, 28, 29], robust portfolio optimization [30, 31], and, more fundamentally, covariance estimation [32, 33, 34] and time series analysis [35].

While being a popular method, much of the work in response to its birth has been solely continuing with the data clustering based on the correlation matrices of the data. This is mainly because a fundamental random matrix ensemble, the Wishart matrix, conveniently resembles correlation matrices. Nevertheless, [36] have shown that the spectra of different random matrix ensembles, in particular similarity matrices of discrete data, may still exhibit similar universal behavior as Wishart matrices. More recently, [37] illustrates a random matrix perspective of sociological data, encompassing insights in the ability of random matrix based methods to uncover complex cluster structures that are characterized by overlaps and low internal uniformity [38]. To bridge this gap, this thesis concerns the extension of spectral modularity maximization beyond correlation matrices, and consequentially, the potency of spectral modularity maximization as a clustering algorithm.

1.1. Research Questions and Contributions

The aim of this thesis is to investigate the use of spectral modularity for clustering high dimensional multivariate data. The main research question of the study is therefore as follows:

Main Research Question. *To what extent is spectral modularity maximization a viable method for clustering multivariate data?*

In this thesis, we study the behavior of spectral modularity maximization, the underlying conceptual challenges, and the necessary enhancements. The main contributions can be expressed in both theoretical and methodological work. Specifically, we theoretically uncover a fundamental challenge of spectral modularity and introduce methodological enhancements to address this uncovered issue. Furthermore, we perform a thorough performance evaluation of the (enhanced) spectral modularity maximization methods. The contributions in this thesis are structurally aligned with the following defined research questions.

First, one may question whether the extension of spectral modularity from correlation matrices, as is done originally in [8], to similarity matrices is theoretically valid. While the theoretical aspects of this question are already addressed in [36] and [37], the feasibility of an application of spectral modularity as a general similarity based clustering method is not yet studied. Therefore, the first research question of this thesis is as follows:

Research Question 1. *How does a naive application of spectral modularity behave in the context of clustering multivariate data?*

In answering research question 1 in this thesis, we discover that clustering with naive spectral modularity maximization, hereafter denoted by SMM0, has some fundamental challenges. In particular, SMM0 appears to be biased towards creating clusterings with relatively large groups. This observation is obtained through the consideration of data with an increasingly large number of groups, a setting that is often neglected in studying the performance of clustering methods. In this setting, where there is a fine-grained group structure, the ability for spectral modularity maximization to detect all groups is significantly flawed. At the same time, existing competitive methods are still capable of detecting the group structure with high accuracy.

On the other hand, when the group structure is coarse, we do find that SMM0 can be competitively applied for high dimensional data and relatively difficult clustering problems. Therefore, it is worthwhile to investigate the reason for the breakdown of spectral modularity, as it may lead to insights to enhance the spectral modularity method. Our second research question is as follows:

Research Question 2. *Why does spectral modularity break for fine-grained group structures?*

It appears that the phenomenon that drives this breakdown of spectral modularity lies in the saturation of the informative group structure in the similarity matrix. To be precise, as the number of groups, K, grows, the number of pairs of objects that are in different clusters grows with order K^2, while the number of pairs of objects that are in the same cluster grows with only order K. Consequently, a filtering of the similarity matrix, that is done in the computation of spectral modularity, is less and less powerful in distinguishing internal similarities of a cluster and external similarities of objects in two different clusters.

Both the empirical observations and the theoretical understanding of the spectral modularity breakdown motivate the development of a particular class of enhancements that are able to overcome the breakdown. This brings us to our third research questions:

Research Question 3. *How do we overcome the breakdown of spectral modularity?*

In principle, the inconsistency caused by the breakdown is demonstrated in two ways: clusterings have fewer groups, and clusterings have heterogeneous group sizes. The combination of these two effects is obtained through inconsistent mergers of the relatively weak groups. Therefore, two enhancements can be proposed, both of which are based on mitigating the bias towards clusterings with fewer and heterogeneous groups. First, a regularization term can be added to the maximization objective of SMM0 that penalizes clusterings with heterogeneous cluster sizes. This way, the bias towards heterogeneous sized groups is extrinsically mitigated. This method neatly fits with the existing modularity maximization frameworks, such as the Louvain method [39]. However, the regularization parameter needs to be externally calibrated with a calibration scheme.

Contribution 1. *We introduce a regularized spectral modularity maximization (SMM1) method that can be maximized within the existing modularity maximization framework but requires a novel external parameter calibration scheme.*

The second enhancement is based on a normalization of the spectral modularity maximization objective. This way, the bias towards heterogeneous-sized groups is intrinsically removed. In addition, the method does not require parameter calibration like in SMM1. However, unlike SMM1, this method requires a novel maximization procedure that we provide.

Contribution 2. *We introduce a normalized spectral modularity maximization (SMM2) method that is completely parameter-free but requires a novel maximization procedure.*

Apart from these solutions, spectral modularity can be used to derive a parameter-free soft clustering method, which is a particularly coveted feature when a data set contains groups with soft boundaries, overlapping, and internally non-uniform groups.

Contribution 3. *We introduce a parameter-free soft clustering method that is based on spectral modularity.*

Finally, we investigate the performance of the spectral modularity based methods (SMM0, SMM1, and SMM2) in comparison with existing clustering methods. The emphasis of the empirical analysis is on data with internally non-uniform groups, as this setting poses difficult problems with existing methods. Our final research question is, therefore:

Research Question 4. *How do the naive and enhanced spectral modularity maximization methods perform for multivariate data clustering compared to existing methods?*

In order to answer this research question, we use high dimensional synthetic data that we generated with specific data generation processes related to the Gaussian Mixture Model, which resembles an easier setting where groups are internally uniform, and the Prototype Model, introduced in [38], which specifies internally non-uniform groups.

1.2. Outline

The body of this thesis is divided into four parts. The first part provides essential background on the context in which this thesis is written. The second part describes the main theoretical and methodological contributions. The third part exposes an empirical performance analysis of the contributed methods. The fourth part discusses the findings of this research, how they relate to existing work, and provides recommendations for future research.

In Part 1, we first introduce mathematical and methodological foundations related to cluster analysis in high dimensional data (Chapter 2). In particular, we discuss the concept of partitions and how they can be compared through a metric called variation of information [40]. Furthermore, we discuss a selection of existing clustering methods, including KMeans [41], KMedoids [42], and Spectral Clustering [13]. Finally, we discuss the challenges of clustering in high dimensional data [3]. Then, we elaborate on some important results from random matrix theory in the context of similarity matrices (Chapter 3). Here, we demonstrate the distribution of eigenvectors of random matrices [9] and the well known Marchenko-Pastur law [10]. Furthermore, we elaborate on the existence of spiked eigenvalues outside the bulk of the eigenvalue distribution [34] and how they are related to the information contained in the associated eigenvectors [43, 34]. Finally, we discuss how the number of spiked eigenvalues can be detected in the absence of a theoretical threshold, through the use of permutation based parallel analysis [44]. Finally, we provide the existing background on the spectral modularity (Chapter 4). Here, we discuss the intuition behind the Girvan-Newman modularity [12] and how this optimization objective can be maximized to provide clusterings of data. At last, we provide an explicit definition of the algorithm concerned with naive spectral modularity maximization.

In Part 2, the primary observation in this part is that the naive spectral modularity maximization has a fundamental flaw in that it breaks as the number of clusters is relatively large, relating to Research Question 1 (Chapter 5). The chapter is oriented towards Research Question 2. Consequently, we illustrate the concept and formalize this observation through the use of theoretical derivation and numerical analysis. The empirical observation of the spectral modularity breakdown is that the modularity maximization becomes increasingly biased towards large groups. Therefore, we propose two solutions to combat this induced bias, answering Research Question 3. First, we propose a regularized adaptation of the modularity objective, i.e. Contribution 1, that is jointly justified by an explicit regularization and a correction of the bias within the modularity matrix (chapter 6). A benefit of this method is that the adjustment is minimal, in the sense that we are still able to use the same modularity maximization methods as in the literature, such as Louvain [39]. A disadvantage of this solution is that, for a parameter-free method, we are required to calibrate a regularization parameter in a somewhat unstable and ad hoc way. Second, we propose a normalized adaptation of the modularity maximization, i.e., Contribution 2, (Chapter 7). This has the benefit that it fundamentally removes the bias, as opposed to regularizing the problem. However, a downside is that we are required to develop novel maximization algorithms. Finally, we introduce a natural extension of the spectral modularity maximization to encompass soft-clustering (Chapter 8), i.e., Contribution 3. This is done in such a way, that the method can be applied to an arbitrary hard partition as a post-processing step.

In Part 3, the objective is to thoroughly evaluate the performance of each of the contributed methods, by addressing Research Question 4. First, we describe the evaluation setting through the description of synthetic data generation processes that help with the construction of data that contains unambiguous ground-truth partitions (Chapter 9). Then, the contributed methods, together with a selection of existing methods, are evaluated in terms of their ability to recover the exact ground-truth partition and their ability to recover representative data profiles of the individual clusters (Chapter 10). Finally, we investigate the practical applicability of clustering methods with a selection of real empirical data sets (Chapter 11). Here, we investigate the performance of the clustering methods on a specific categorical data set that concerns soybeans [45]. Then, we use the well-known MNIST digits data set to demonstrate the spectral modularity breakdown in a real data setting. Finally, in the MNIST setting, challenges of the heuristic eigenvalue threshold procedure are demonstrated, as is the ability to detect the ground-truth representative data profiles of the digits.

In Part 4, the results demonstrated in this thesis are discussed. First, we position the contribution of this thesis by studying the relations with existing scientific work in this direction (Chapter 12). In particular, we highlight the related developments within the spectral clustering paradigm. In addition, we elaborate on the broader use of modularity inspired objectives for clustering multivariate data. Then, we discuss the contributions of this thesis and the associated limitations (Chapter 13). Here, we discuss the findings of our work related to theoretical and methodological contributions. Furthermore, we discuss the results from our performance evaluation of the contributed methods and the limitations associated with the performance analysis. Finally, recommended directions for future research are given (Chapter 14). Here, we provide insights that can inspire further theoretical analysis of spectral modularity breakdown. Furthermore, we discuss potential methodological enhancements to the regularization parameter calibration scheme and normalized spectral modularity maximization.

In Appendix A, three modular research proposals associated with results obtained in this thesis are discussed. First, we propose a compact article that introduces the spectral modularity breakdown and the two fixes in the context of correlation-based clustering. Second, we propose the extension of enhanced spectral modularity to similarity-based clustering, with a specific emphasis on the internally non-uniform groups, through studying the Categorical Mixed Prototype Model. Finally, we propose a research direction to study a novel method to separate the informative eigenvectors from the non-informative that is based on the observed limitations of the current permutation-based parallel analysis.

1.3. Notation

The following describes the most important notation details that are used throughout the thesis. In cases where notation deviates, it is explicitly clarified from the context.

$\mathcal{X}$ denotes the data space. d is used to denote a distance metric on $\mathcal{X}$. p denotes the number of dimensions. n denotes the number of objects. s is used to denote a similarity function on $\mathcal{X}$. $\mathbf{S}$ denotes a $n \times n$ similarity matrix. $\mathbf{X}$ denotes a $n \times p$ data matrix, with rows $\{\mathbf{x}_{i=1}^{n}\} \subseteq \mathcal{X}$. q denotes the ratio between number of dimensions p and number of objects n, i.e. $q = \frac{n}{p}$.

$\rho = \{C_1, \ldots, C_K\}$ denotes a partition of $\{1, \ldots, n\}$ with $C_1, \ldots, C_K$ being the groups. K denotes the number of groups. Specifically in the context of cluster analysis, we refer to ρ as a clustering if it is obtained by a clustering method, where the elements of ρ are referred to as clusters. Where $\hat{K}$ denotes the number of clusters, obtained from counting the number of spiked eigenvalues. In the context of a ground-truth partition, we denote the partition with ρ^* and the number of groups by K^*.

$\{\lambda_m\}_{m=1}^{n}$ denote the eigenvalues of $\mathbf{S}$ that are always ordered $\lambda_1 \geq \cdots \geq \lambda_n$. The associated eigenvectors are typically denoted with $\{\mathbf{v}^{(m)}\}_{m=1}^{n} \subset \mathbb{R}^n$. The ith element of eigenvector associated to the kth largest eigenvalue is denoted by $v_i^{(m)} \in \mathbb{R}$. Furthermore, to prevent ambiguity, the eigenvectors are considered to be normalized such that $||\mathbf{v}^{(m)}||_2 = 1$ for all $m \in \{1, \ldots, \hat{K}\}$

$\mathbf{B}$ generally denotes the spectral modularity matrix. $\mathbf{R}$ denotes the matrix of spectral modularity vectors, such that $\mathbf{B} = \mathbf{R}\mathbf{R}^\top$. The rows of $\mathbf{R}$ are denoted by $\{\mathbf{r}_i\}_{i=1}^{n}$.

Given some ρ of size K, the elements of the $K \times K$ matrix $\mathbf{G}$ denote the group affinity matrix, which is the sum of all pairwise modularities between objects of two clusters, i.e., $\mathbf{G}_{kh} = \sum_{i \in C_k} \sum_{j \in C_h} \mathbf{B}_{ij}$.

$\mathcal{M}$ denotes the space of Markov matrices, which are $n \times K$ matrices with elements in $[0, 1]$ such that the rows of the matrix sum up to one. $\mathbf{P} \in \mathcal{M}$ denotes a (soft) partition matrix. If the elements of $\mathbf{P}$ are in $\{0, 1\}$ the matrix can be associated to a hard partition. $\partial\mathcal{M}$ represents the boundary of $\mathcal{M}$, meaning the entries in the matrices are binary and thus associated to hard partitions. Formally, denoted

$$\mathcal{M} = \left\{ \mathbf{P} \in [0,1]^{n \times K} : \sum_{k=1}^{K} \mathbf{P}_{ik} = 1 \right\} \quad \text{and} \quad \partial\mathcal{M} = \left\{ \mathbf{P} \in \{0,1\}^{n \times K} : \sum_{k=1}^{K} \mathbf{P}_{ik} = 1 \right\}. \tag{1.1}$$

Part I

Essential Background and Concepts

2

High Dimensional Cluster Analysis

In this chapter, we introduce the essential background on cluster analysis in high dimensional data. The task of clustering is to extract non-overlapping groups from a dataset such that objects inside the same group are more similar than objects that are not in the same group. This objective is quite abstract, therefore, a particular clustering algorithm is typically defined for a specific quantification of this objective. In this way, the essence of cluster analysis is to find the best clustering of the data out of all possible clusterings.

A specific clustering refers to one way that the objects of a dataset can be clustered together, and is formally referred to as a partition of a set. The number of possible partitions of a set of objects is huge. Therefore, clustering is typically considered to be a computationally hard task. For this reason, many clustering algorithms are tailored to specific contexts. In this way, different clustering algorithms are likely to be able to deal with different levels and types of difficulty. For example, certain algorithms, like KMeans [41], require linear separability and convexity of the clusters, while other algorithms, like spectral clustering [13], do not.

Although clustering itself is already a hard problem, an additional layer of difficulty is unavoidable when considering high dimensional data. Apart from the added computational and interpretational difficulties that are natural to large optimization problems, there are surprising characteristics in high dimensional data that may pose fundamental issues when applying traditional clustering algorithms. The combination of these challenges is collectively known as the curse of dimensionality. Within all clustering paradigms, many adaptations of traditional methods exist that are designed to deal with these challenges, often through explicit dimensionality reduction or regularization.

In Section 2.1, the notion of a distance metric is defined to express closeness between objects. This is typically an important aspect when describing the quality of a specific partition. In Section 2.2, the partition is formally defined together with the notion of quality of a partition and how the discrepancy between partitions can be computed with the variation of information. Furthermore, we discuss how representative data profiles can be associated with a particular partition. In Section 2.3, a compact overview of a few prominent clustering paradigms is given. In Section 2.4, the challenges of clustering in high dimensional data are described. Furthermore, a compact overview of the typical approaches to circumventing these challenges is given.

2.1. Distance Metrics

In most clustering objectives, the closeness between objects within clusters and the farness between objects in different clusters are used. In order to obtain such measures between two objects in a dataset, one needs to define a specific distance metric that is valid on the space in which the data lies. We denote this data space with $\mathcal{X}$. This means that for some data set of size n, i.e., $\{\mathbf{x}_i\}_{i=1}^n$, the objects satisfy $\mathbf{x}_i \in \mathcal{X}$, for all $i \in \{1, \ldots, n\}$. Then, we define a particular function d to be a distance metric on $\mathcal{X}$. Specifically, the distance metric is a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that satisfies the following properties for all points $x, y, z \in \mathcal{X}$. First, d must be reflexive, i.e., the distance of objects to itself is zero; $d(x, x) = 0$ for all $x \in \mathcal{X}$. Second, d must be positive, i.e., $d(x, y) > 0$ for $x \neq y$ $x, y \in \mathcal{X}$. Third, d must be symmetric, i.e. $d(x, y) = d(y, x)$. Finally, d must satisfy the triangle inequality, i.e., $d(x, z) \leq d(x, y) + d(y, z)$.

The combination of a data space $\mathcal{X}$ and a distance metric d can be defined as a metric space $(\mathcal{X}, d)$, if and only if the distance metric d is well-defined on $\mathcal{X}$. Examples of generally considered valid metric spaces are

1. p-dimensional real space with **Euclidean** distance, i.e. $(\mathbb{R}^p, d)$, where
$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^{p} |x_j^l - x_j^l|^2}. \tag{2.1}$$
2. Ordinal discrete space endowed with **Manhattan** distance, i.e., $(\{1, \ldots, M\}^p, d)$ for some $M \in \mathbb{N}$, where
$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{p} |x_j^l - x_j^l|. \tag{2.2}$$
3. Nominal discrete space endowed with **Hamming** distance, i.e., $(\{1, \ldots, M\}^p, d)$ for some $M \in \mathbb{N}$, where
$$d(\mathbf{x}_i, \mathbf{x}_j) = d^{-1} \sum_{l=1}^{p} \mathbf{1}_{x_j^l \neq x_j^l}. \tag{2.3}$$

An example of an invalid metric space is the use of Manhattan or Euclidean distance in a nominal space, as subtraction and summation in categorical data are not defined.

In the context of clustering, we may refer to distances or similarities as being **internal** or **external**, where internal distance means the distance between two objects that are in the same group, and external distance is computed for two objects that are in two different groups.

2.2. Partition

In cluster analysis, we are generally interested in finding a group assignment for each object in a set according to some criteria. The mathematical object that is associated with this task is a partition. To be precise, a partition, denoted by ρ, is defined for a set of n indices, $\{1, \ldots, n\}$, associated with the objects $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$. Then, the set of sets

$$\rho = \{C_1, \ldots, C_K\}, \tag{2.4}$$

for $C_1, \ldots, C_K \subseteq \{1, \ldots, n\}$ and $K \in \{1, \ldots, n\}$ is a partition of $\{1, \ldots, n\}$ if it satisfies the following properties for all $k, h \in \{1, \ldots, K\}$. First, groups are non-empty, i.e., $C_k \neq \emptyset$. Second, groups are disjoint, i.e., $|C_k \cap C_h| = \emptyset$. Third, the union of the groups covers the entire set, i.e. $\cup_{k=1}^{K} C_k = \{1, \ldots, n\}$. Here, the number of non-empty and mutually exclusive subsets, or, in other words, the number of groups in ρ is denoted by $K = |\rho|$ and also referred to as the size of the partition. Furthermore, a K-partition refers to a partition of size K.

Throughout this thesis, we use the term 'a partition' to refer to the more general mathematical object, and the term 'a clustering' to refer to a partition that is obtained through cluster analysis methods. Furthermore, we use the term 'a group' to refer to some element of a partition in general and the term 'a cluster' to specifically refer to some element of a clustering.

One may recognize that the number of possible clusterings will grow fast with respect to the number of objects. The space of partitions is denoted by $\mathcal{P}$, and the size of this set is known as the nth Bell number. To illustrate the size of this quantity, consider the first 7 Bell numbers that are depicted in Table 2.1. Because of the large size of the space of partitions, it is not hard to see that finding an optimal partition for some non-trivial objective is computationally hard. This should expose the intuition of the algorithmic intricacies that are involved with cluster analysis in general, independent of the difficulty of the particular data set or the high dimensionality.

n	1	2	3	4	5	6	7
nth Bell number	1	2	5	15	203	877	4140

Table 2.1: Bell number: the number of possible partitions of a set of size n.

At last, from the definition of the partition, one may notice a pair of extreme partitions. Consider that for a partition of a set of size n, the maximum size of ρ is n and the smallest size is 1. Both extremes correspond to a trivial partition: one that groups all objects in a group, and one that groups all objects in a separate group. These partitions can, respectively, be written as:

1. **(Singleton Partition):** $\rho = \{\{1, \ldots, n\}\} \in \mathcal{P}$ with $|\rho| = 1$,
2. **(Partition of Singletons):** $\rho = \{\{i\} : i \in \{1, \ldots, n\} \in \mathcal{P}$ with $|\rho| = n$.

2.2.1. Group Representative Profiles

Given a partition $\rho = \{C_1, \ldots, C_K\}$ of data set in $\mathcal{X}$, one is often interested in determining the representative data profiles of the groups $\{C_k\}_{k=1}^K$. The data profiles can be expressed as K points in $\mathcal{X}$ that are representative of the objects in a particular group, and are denoted by $\{\boldsymbol{\mu}_k\}_{k=1}^K$. The procedure for inferring these group centers depends on the data space. For example, a natural procedure in $\mathbb{R}^p$ is to use the mean of the objects in a group, i.e., for all $k \in \{1, \ldots, K\}$

$$\boldsymbol{\mu}_k = \frac{1}{|C_k|} \sum_{i \in C_k} \mathbf{x}_i. \tag{2.5}$$

However, in categorical data, summations are not defined. Therefore, to obtain a representative profile of the groups, in that setting, we use the mode of the objects in a group instead. The mode gives the most frequently occurring value in a dataset, making it a natural choice for representing the group center. When dealing with categorical data, each attribute can take on a finite set of discrete values. In this way, the categorical mode is an intuitive method for identifying central objects in groups of categorical data, leveraging the frequency of attribute values to determine the most representative instances. The group representative profile, therefore, can be defined by the mode of each attribute within the group. Formally, for a particular group $C_k \in \rho$, the representative profile $\boldsymbol{\mu}_k$ is determined as follows:

$$\mu_k^l = \arg \max_{a \in 1, \ldots, N} \sum_{i \in C_k} \mathbf{1}_{\{x_i^l = a\}}. \tag{2.6}$$

where $\{1, \ldots, N\}$ represents the set of all possible values for the l-th attribute, and $\mathbf{1}_{\{x_i^l = a\}}$ is an indicator function that equals 1 if the l-th attribute of x_i is a, and 0 otherwise.

Note that representative profiles of groups may not always be well-defined. For example, the set of canonical concentric rings in $\mathbb{R}^2$, which is further discussed in Section 2.3 and are depicted in Figure 2.1, can not be effectively represented by a selection of K points in the data space $\mathbb{R}^2$. However, in this thesis, we only consider data sets where such representative profiles are meaningful.

2.2.2. Clustering Quality

The purpose of a clustering algorithm is to define a function that uses the data set to encode a measure of quality for the clustering and maximize this function. The quality function Q is a function from the space of partitions to the real numbers, i.e.,

$$Q : \mathcal{P} \mapsto \mathbb{R}. \tag{2.7}$$

Typically, a clustering ρ that satisfies $Q(\rho) > Q(\rho')$ for some other clustering ρ', should be favored over ρ'. A natural tendency in the specification of a specific quality function Q is to utilize the notion of internal and external distances. However, this typically leads to ambiguous quality functions, as it is unclear how to weigh the two quantities.

An alternative approach is to use the distance of objects within a cluster to the representative profile of the cluster. This is useful because the central points are by definition the center of the cluster, and any object can be associated with a closest cluster center. Thereby leading to relatively unambiguous expressions for the quality function.

An example for such a quality function Q that is defined for a Euclidean metric space, i.e., $(\mathbb{R}^p, d)$ with Euclidean metric d and some integer $p \geq 1$, is a function that uses the distance between the objects in a cluster and the center of the cluster. Here, the center of the cluster is obtained by averaging the elements of the cluster, i.e.,

$$Q(\rho) = -\sum_{C_k \in \rho} \sum_{i \in C_k} (\mathbf{x}_i - \boldsymbol{\mu}_k)^2 , \tag{2.8}$$

where $\{\boldsymbol{\mu}_k\}_{k=1}^K$ are the cluster representative data profiles of ρ defined in Equation 2.5.

2.2.3. Comparing Partitions

Using the notion of the quality function Q, it is clear that we can favor certain partitions by comparing their qualities. However, this does not demonstrate how much the two partitions differ. Indeed, two completely different partitions may have the same partition quality, and two almost identical partitions may have drastically different qualities.

Fortunately, if we have two partitions, $\rho, \rho' \in \mathcal{P}$, the exact discrepancy between these two partitions can be computed in numerous ways. Comparing two partitions can be useful when there is a particular important partition ρ^*, e.g., a known optimum to 2.7, and one is interested in the resemblance of a clustering ρ to the partition ρ^*.

The difference between the two partitions can be quantified by counting the overlapping partition relationships of objects. In these criteria, there are four quantities of interest that are related to the observations of two objects being in the same group or not: true positives, true negatives, false positives, and false negatives. Using these quantities, a plethora of evaluation metrics can be considered that are all based on counting pairs [46, 47, 48, 40].

However, many of these criteria are difficult to use in comparing multiple pairs of partitions, e.g., when comparing the resemblance of ρ^* and ρ with the resemblance of ρ^* and ρ'. This is difficult because the criteria based on counting pairs do not satisfy metric properties akin to those specified in Section 2.1. To this end, [40] use information theoretic principles to derive a metric on the space of partitions, that is referred to as Variation of Information (VI). For the definition of VI, there are two important information theoretical principles that are extended to the context of partitions.

First, the entropy associated with a partition ρ can be quantified using the uncertainty of a uniformly randomly drawn object belonging to a specific group. Specifically, given a data set of size n and a partition $\rho = \{C_1, \ldots, C_K\}$, the probability that an arbitrary object is in the group C_k for some $k \in \{1, \ldots, K\}$ is $\frac{|C_k|}{n}$. Then, the partition entropy is the entropy associated with the system of drawing these arbitrary objects and is given by

$$H(\rho) = -\sum_{C_k \in \rho} \frac{|C_k|}{n} \log \frac{|C_k|}{n}. \tag{2.9}$$

This entropy measure is always non-negative and takes a value of 0 if and only if there is no uncertainty. This means that we are certain that a uniformly randomly drawn object belongs to a specific group, which is only the case for the trivial partition $\rho = \{\{1, \ldots, n\}\}$.

The second information theoretical principle is the definition of the mutual information of two partitions, which is based on the mutual information between the groups of uniformly randomly drawn objects from the two partitions. This is defined as:

$$I(\rho, \rho') = -\sum_{C_k \in \rho} \sum_{C'_k \in \rho'} \frac{|C_k \cap C'_k|}{n} \log \frac{|C_k \cap C'_k|}{|C_k||C'_k|}. \tag{2.10}$$

The mutual information is always non-negative and symmetric. Furthermore, when $\rho = \rho'$, we have that $I(\rho, \rho') = H(\rho) = H(\rho')$. Then, VI can be defined as

$$VI(\rho, \rho') = H(\rho) + H(\rho') - 2I(\rho, \rho'). \tag{2.11}$$

For a detailed intuition and description of the mathematical properties of VI, the reader is referred to [40].

2.3. Clustering Methods

In light of the previous section, the ideal clustering problem can be roughly expressed by the maximization of a specific well-defined quality function Q over the space of all possible partitions $\mathcal{P}$, i.e.

$$\max_{\rho \in \mathcal{P}} Q(\rho). \tag{2.12}$$

Because of the size of the search space $\mathcal{P}$, this clearly is a difficult problem to solve naively. Therefore, many constraints, adaptations, or alternative definitions of the clustering problem are used to obtain tractable algorithms that can provide reasonable clusterings for specific contexts.

There are a few commonly used perspectives to describe the plethora of clustering algorithms: statistical, partitional, hierarchical, and spectral. The statistical clustering methods are based on the statistical inference of a clustering model. The partitional clustering methods are based on heuristic partitioning of the data set. The hierarchical clustering methods are based on a dendrogram that can be cut at various levels, thereby demonstrating group structure at different levels of granularity. Spectral methods are based on the eigenvectors obtained through the spectral decomposition of certain matrices that are related to the data.

These perspectives are not necessarily mutually exclusive but rather give an indication of the philosophy behind the method. For example, the spectral methods often depend on a partitional clustering algorithm as a subcomponent of the method. In [49] an overview of clustering algorithms is given. For completeness, a description of the most important algorithms for this thesis is given below.

2.3.1. Statistical

It is tempting to approach a clustering problem from a statistical perspective, as it may provide an insightful description of the processes behind the data. In this way, we assume there is an underlying K-partition ρ, possibly endowed with a prior probability distribution, in which each of the objects within different groups is associated with a single probability distribution. This gives us the statistical model that is referred to as a finite mixture model and can be formally expressed by

$$\mathbf{x}_i \sim \pi_k \text{ for all } i \in C_k, \tag{2.13}$$

$$\{C_1, \ldots, C_K\} = \rho \sim f_\rho. \tag{2.14}$$

Here, objects $\{\mathbf{x}_i\}_{i=1}^n$ that belong to group C_k for some $k \in \{1, \ldots, K\}$ follow distribution π_k.

An example of such a finite mixture model is a Gaussian mixture model (GMM). The GMM is defined by the model where the objects of a group are distributed with multivariate Gaussian distributions, i.e., for some $k \in \{1, \ldots, K\}$, we have $\pi_k = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\mu}_k \in \mathbb{R}^p$, and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{p \times p}$, The distribution parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ and the underlying K-partition ρ can be estimated by maximum likelihood estimation using the Expectation-Maximization framework.

The most important benefit of this statistical perspective, is that the clusterings obtained from model based clustering have high interpretability. In particular, being able to directly quantify uncertainties through the explicit definition of group-specific probability distributions is a valuable aspect that is difficult to find in other clustering methods.

On the other hand, there are computational and domain dependency drawbacks that make statistical cluster analysis infeasible for many applications. Specifically, the number of parameters to estimate can be incredibly large. For example, for a p dimensional GMM, all the $O(p^2)$ entries of the covariance matrices of the K multivariate Gaussian distributions need to be estimated. In addition, prior to clustering the data set, assumptions about the model of the statistical group distributions $\{\pi_k\}_{k=1}^K$ must be made. This can be problematic for two reasons. First, it is often the case that the model is not known, thereby creating an extra layer of bias through the model choice. Second, it may be difficult to define a reasonable model for the specific context, e.g., when the random variables have more obscure dependency structures.

Nevertheless, this statistical perspective is a highly desirable vantage point for clustering algorithms. Therefore, much research is concerned with developments in this area [50, 51]. In particular, Bayesian perspectives on the clustering problem have been recently embraced [52]. An additional important application of this perspective is Latent Dirichlet Allocation [53], which relaxes strict cluster allocations to allow soft clusterings. In Chapter 8, a soft clustering procedure is introduced that adheres to the methodological context of this thesis.

2.3.2. Partitional

Partitional algorithms are purposed at clustering a set of objects directly based on some inter object relation. Among those inter-object relations, the most commonly is distance. Other relations are density [54] or connectivity [55]. Whenever a quality function of Q is uses distances within the data space directly, we refer to these specific clustering problems as distance based. One commonly used class of distance based clustering problems is the KMeans [41], which finds clusters whose objects are minimally distanced to the cluster representative profile, or specifically the cluster center, akin to the discussion of Equation 2.5. To be precise, the KMeans problem is defined as follows:

$$\min_{\rho \in \mathcal{P}} \sum_{C_k \in \rho} \sum_{i \in C_k} ||\mathbf{x}_i - \boldsymbol{\mu}_k||_2^2, \tag{2.15}$$

where $||\cdot||_2$ is the Euclidean norm. This is equivalent to $\max_{\rho \in \mathcal{P}} Q(\rho)$, where Q is defined by the partition quality function in Equation 2.8.

The most commonly used algorithm that optimizes the KMeans problem is Lloyd's algorithm [56]. This particular algorithm is done by alternating between two steps. The method initializes with $\hat{K}$, obtained from an approximation of K, positions in the data space that act as the initial centers of the $\hat{K}$ clusters. In the first step, objects are assigned to one of the $\hat{K}$ clusters, decided by the closest representative object. In the second step, the $\hat{K}$ representative objects are recomputed using Equation 2.5 with the newly found clustering. These two steps are performed until the objective does not improve.

The KMeans clustering paradigm is a fundamental heuristic method in the context of cluster analysis, primarily because it reappears in different contexts. First, the KMeans problem can be equivalently described as a matrix factorization problem [57], which allows for a broader theoretical analysis of the problem. Second, KMeans is essentially equivalent to performing Expectation Maximization with an isotropic Gaussian Mixture Model [58], i.e., the covariance matrices of the distributions are proportional to the identity. The latter also demonstrates a limitation of the KMeans, as it assumes that the objects are spherically distributed with the same variance for each cluster. Therefore, if the distributions of objects in different clusters have different variances, or are not exactly spherical, KMeans will not perform well.

The KMeans problem can only be applied to a setting where the mean of a set of objects is meaningful. This is problematic in the case where data is nominal, where means are not defined due to the absence of a summation operation. For this setting, a related clustering problem exists called KMedoids [59]. In contrast to KMeans, KMedoids chooses existing representative objects from the data set instead of arbitrary representative points in the data space. This has the benefit that the KMedoids algorithm has greater interpretability of the clusters; an observed object explicitly defines them. Furthermore, instead of minimizing the squared Euclidean distance between the cluster centers and the objects in that cluster, like is done in KMeans, KMedoids can be used with arbitrary distance metrics. This makes it applicable for nominal data. The KMedoids problem can therefore be defined abstractly for some arbitrary distance metric d, including Euclidean, Manhattan, or Hamming, as

$$\min_{\rho \in \mathcal{P}} \sum_{C_k \in \rho} \min_{\mathbf{x} \in C_k} \sum_{i \in C_k} d(\mathbf{x}_i, \mathbf{x}). \tag{2.16}$$

A well-known heuristic algorithm for solving KMedoids is the partitioning around medoids algorithm, as introduced in [59]. The algorithm is initialized with $\hat{K}$ objects that function as initial medoids for the clusters. Then the objects are assigned to the cluster with the closest medoid. Then, a medoid object and a non-medoid object are swapped such that this specific swap improves the objective the most. This last step is repeated until no improvement can be made.

2.3.3. Hierarchical

Hierarchical clustering methods are based on clusterings that occur at different levels of granularity. The hierarchical methods provide a hierarchical structure in the form of a dendrogram. The methods generally start with a trivial partition and merge (or split) the clusters according to some criteria and a value t. In particular, for a given value $t > 0$, a clustering ρ, and a specific criterion, two clusters C_k and C_h in ρ are merged (or split) if the criterion, which is often based on distances between objects, is met.

The procedure to obtain the dendrogram can be performed in two ways. First, one can start from a trivial partition that contains a single cluster with all the elements and split the clusters as the threshold t grows. This approach is called divisive. Second, one can start from a trivial partition that contains n clusters of single element sets and merge clusters as the threshold grows. This approach is called agglomerative.

An example of a criterion is the single linkage criteria. In the agglomerative setting, these criteria determine the merging of two clusters if a single pair of objects in the different clusters has a distance smaller than the value t, i.e., C_k and C_h are merged if

$$d(\mathbf{x}_i, \mathbf{x}_j) < t \text{ for some } i \in C_k, j \in C_h. \tag{2.17}$$

Other criteria are complete linkage, which requires that all the distances between objects in the two different clusters are smaller than the value t, and average linkage, which requires that the average distances between objects in the two different clusters are smaller than the value t. Many more criteria exist [60]. A limitation of these methods is that the approach provides an entire dendrogram as opposed to a single, specific clustering. In order to obtain a clustering, one needs to choose a specific value of t. Furthermore, in itself, the hierarchical methods do not directly relate to the framework as described by the abstract objective in equation 2.7, as the hierarchical framework does not describe a cluster objective directly.

2.3.4. Spectral

Spectral clustering [13] borrows ideas from graph theory, where the eigenvectors of the Laplacian matrix are commonly used to perform clustering on the nodes of a graph. In the context of multivariate data, we can use the distances between objects to represent a related graph. In particular, we can interpret a similarity matrix $\mathbf{S}$, of which the elements represent the pairwise similarities between objects, as a weighted adjacency matrix of a graph with n nodes. Then, the Laplacian can be defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{S}, \tag{2.18}$$

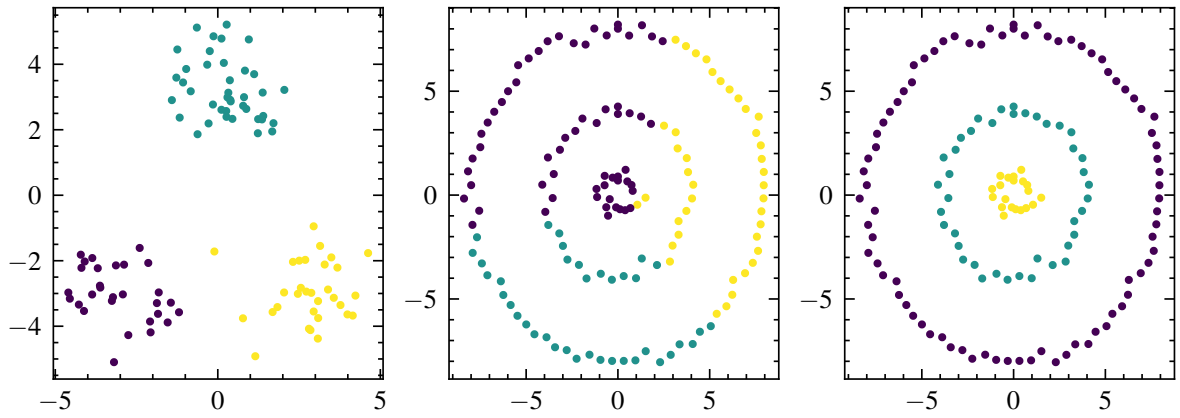


Figure 2.1: Low dimensional clustering examples. The leftmost figure displays KMeans clustering on a two-dimensional multivariate GMM with $K = 3$. The middle figure displays the KMeans clustering on the concentric ring example. The rightmost figure displays the clusterings obtained with spectral clustering.

where $\mathbf{D}$ is a diagonal matrix with the node degrees on the diagonal, i.e., for all $i \in \{1, \dots, n\}$, we have $\mathbf{D}_{ii} = \sum_j \mathbf{S}_{ij}$. The elements of $\mathbf{S}$ are obtained with a suitable similarity function $s : \mathcal{X} \times \mathcal{X} \to [0, 1]$ such that $\mathbf{S}_{ij} = s(\mathbf{x}_i, \mathbf{x}_j)$, of which the details are discussed in Chapter 3. In spectral graph theory, the Laplacian is a useful representation of the graph given that it is guaranteed to be symmetric positive definite. Furthermore, the multiplicity of the zero eigenvalue of the Laplacian corresponds to the number of connected components, and the sparsest cut of a graph can be approximated using the second eigenvector associated with the second-smallest eigenvalue of the Laplacian matrix, known as the Fiedler vector [61]. A more thorough discussion of spectral clustering methods and their relation to the spectral modularity methods is given in Chapter 12

The scope of spectral clustering is large, with many ongoing developments [15]. To this end, in [13] a specific spectral clustering algorithm is suggested to be the most generally applicable method. In particular, the use of a normalized version of the Laplacian $\mathbf{L}_{norm}$ as opposed to $\mathbf{L}$ poses to be more suitable in many cases. The normalized Laplacian from [62] takes the following form:

$$\mathbf{L}_{norm} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2}. \tag{2.19}$$

Therefore, for the remainder of the thesis, the use of spectral clustering (SC) refers to applying KMeans to the eigenvectors associated with the $\hat{K}$ smallest eigenvalues of $\mathbf{L}_{norm}$, unless stated otherwise.

Apart from the graph theoretical motivation, the Laplacian matrix gives access to a particularly useful Euclidean embedding of the data set $\{\mathbf{x}_i\}_{i=1}^n$ in an arbitrary data space $\mathcal{X}$. For an arbitrary metric space $(\mathcal{X}, d)$ and a suitable similarity function s, the eigenvectors of the Laplacian matrix induce a mapping of the objects from $\mathcal{X}$ to $\mathbb{R}^n$, in which the distances between all objects in $\mathcal{X}$ are equal to the Euclidean distance between the objects in the embedding in $\mathbb{R}^n$. In particular, this can be done in such a way that the number of dimensions of the embedding space can be much smaller than n. In this way, using the Laplacian matrix, the data can be represented in a lower dimensional Euclidean space [63]. The fact that the Laplacian eigenvectors provide a Euclidean embedding motivates the use of the KMeans algorithm on the dataset projected on the Laplacian eigenvectors, which is a common choice within the spectral clustering paradigm.

Finally, because of the construction of the similarity graph, the spectral clustering methods are typically capable of clustering non-linearly separable clusters, in contrast to more traditional methods like KMeans. A canonical example is that of clustering concentric rings, depicted in Figure 2.1. Visually, the groups are easy to distinguish, however, many clustering algorithms, such as KMeans and KMedoids, require linear separation of the clusters. Therefore, these methods are unable to cluster meaningfully in this context. On the other hand, the spectral clustering methods do cluster the rings conceptually correctly.

2.4. High Dimensional Data

The characteristics of high dimensional data can pose fundamental challenges. These challenges are often summarized as the curse of dimensionality, a term that is first described in [64]. In the classical setting, where most of the statistical methods are originally developed, one is often concerned with data matrices of size $n \times p$, where p is significantly smaller than n. However, in modern applications, it is often the case that p and n are of the same order of magnitude or that p is significantly larger than n. This modern setting, is what we refer to as the high dimensional regime.

2.4.1. Curse of Dimensionality

A relatively intuitive challenge is the computational complexity that is naturally accompanied by an increasing number of dimensions. Numerical computations and optimizations in these high dimensional spaces often require exponentially more steps. In particular, this can be illustrated by the growing state space of d-dimensional binary data, that is, of size 2^p. This phenomenon is associated with the common characterization that high dimensional spaces are sparse. For example, the sample covariance matrix, which is known to be consistent for small p and large n, requires $O(p^2)$ terms to be approximated. Therefore, for large p, such that $p \approx n$ the estimator is inconsistent [65]. This inconsistency of the sample covariance matrix, which is further discussed in Chapter 3, is a fundamental challenge of high dimensional data and is a core topic of random matrix theory [33]. In addition, a perhaps even more intuitive challenge is the lack of visualization of data that has high dimensions. Clearly, data with more than three dimensions cannot be visualized directly. This makes data analysis with high dimensional data particularly difficult.

Apart from these intuitive challenges, there are, however, more surprising characteristics of high dimensional geometry [3] that affect the ability to perform cluster analysis. In particular, as dimensions grow, the notion of distances becomes less meaningful, making many clustering algorithms unable to distinguish between internal and external distances. This phenomenon, known as distance concentration, refers to the pairwise dissimilarities converging to the same value as the dimensionality of the data increases. This makes it more and more difficult to distinguish between pairs of objects that are far away and pairs of objects that are near.

Furthermore, it is not difficult to see that as the number of dimensions grows, more and more randomness is added to the system. In principle, this is a relatively logical consequence of adding more dimensions, which we refer to as noise accumulation. Indeed, under natural conditions, one can expect that one extra dimension to provide for one more source of randomness. However, the challenge is, in fact, more daunting than this. In particular, events that appear to be rare in a single dimension become common with increasing dimensionality due to the increasing probability of a rare event in each of the dimensions. This means that the existence of many dimensions suggests that it is likely that the object has extreme attributes in at least a few directions. Interpreting these rare instances of the object's individual dimensions as meaningful. This way, one has a tendency to construct machine learning models that are based on false information by treating noise as information. This is a particular example of overfitting, which is generally considered malpractice in machine learning research.

2.4.2. Lower Dimensional Structure

While high dimensional geometry exhibits challenges that are hard to circumvent, data analysis in high dimensions is often still possible. In particular, data is often represented by a lower dimensional structure, as opposed to being evenly distributed in a high dimensional space. This means that while the data is high dimensional, there generally exists a latent representation of the data that is low dimensional.

In order to illustrate why this is often the case, one can think of data as an outcome of a complex system. In such a system, although the state space can be extremely large, the observed states are produced by the logical coherence of the system. Therefore, one can think of all the states that are coherent with the system as being represented in a lower dimensional latent state space.

In the context of cluster analysis, this holds observation of a latent representation even more. Namely, in the task of finding a clustering that describes the data, one can think of the clusters themselves as being a representation of the data. If the data cannot be effectively represented by a significantly smaller number of dimensions, then it is also not possible to talk about the existence of a clustering. This makes the assumption of a lower dimensional latent structure in which the data resides especially sensible in cluster analysis.

Upon retrieving the representative lower dimensional structures, classical statistics can be applied again. Therefore, the task of high dimensional statistics, and consequently, high dimensional clustering, is to retrieve these low dimensional latent representations. The most widely used technique in this setting is principal component analysis, which aims to find the directions of the data that explain the most variance, i.e., the principal components.

2.4.3. Clustering in High Dimensions

Because of the challenges in high dimensions, clustering with naive methods is not suitable. Therefore, many adaptations have been provided. Model based clustering often suffers from extremely large parameter spaces, and therefore regularization or dimensionality reductions [66, 67, 4] are used to mitigate the issues. In [68] the authors provide an overview of many heuristic based clustering methods for high dimensional data. Most of the methods are based on finding a subspace in which the data can be clustered more effectively and efficiently, using the sparsity and redundancy of the original data space.

Specifically, it is not difficult to see that the use of KMeans will become problematic when the dimensionality of the data is high. In particular, the distance concentration will make it difficult for any KMeans algorithm to distinguish between internal and external distances, leading to an increasing number of local optima. Therefore, many solutions are based on the use of a dimensionality reduction step. For example, principal component analysis is used in [69], multidimensional scaling is used in [70], and manifold algorithm, such as local linear embedding, local tangent space alignment, and Laplacian eigenmap are considered in [71]. In particular, if we use the Laplacian eigenmap, i.e., with the properties of the Laplacian Euclidean embedding, then the combination of this dimensionality reduction and KMeans gives us the exact spectral clustering algorithm described in Section 2.3.

Furthermore, for KMedoids in Euclidean data space, the same distance concentration phenomenon will likely happen. However, in a discrete setting, KMedoids is also challenged by the noise accumulation of high dimensional data. In particular, as the number of dimensions grows, the probability that good representative objects exist diminishes fast. Therefore, KMedoids will have difficulty clustering high dimensional data.

Unlike the other perspectives discussed in Section 2.3, spectral clustering methods are not heavily affected by the challenges of high dimensional data. In essence, all spectral clustering algorithms can be characterized by the fact that they utilize a selection of eigenvectors, often only the ones associated with the largest or smallest eigenvalues, to obtain an optimal clustering. It is not difficult to see how spectral clustering is suitable for high dimensional data, as its focus on a few eigenvectors shares resemblance with the projection to a lower dimensional latent representation.

3

Random Similarity Matrices

In this chapter, the well-known results from random matrix theory are discussed in relation to similarity matrices. Random matrix theory studies the statistical properties of matrices whose entries are random variables [72] and is commonly studied in the context of machine learning and statistics [9, 11]. Therefore, we explore some of these key concepts and results, focusing on similarity matrices and their properties. Of particular interest are random matrices that resemble the Wishart-type ensemble [73]. The theory of Wishart matrices finds its use in multivariate statistics, mainly through its relation to covariance matrices, which is widely applicable for research fields such as finance, signal processing, and machine learning. In studying random similarity matrices, they are important, as they resemble matrices that are symmetric and positive definite, a property that is natural to many similarity matrices.

A fundamental part of random matrix theory is the study of the eigenvalue distributions of random matrices. When discussing the eigenvalue distribution of Wishart matrices, one often refers to the distribution of the so-called bulk of the eigenvalues of the matrix. In particular, for the simplest matrix with Gaussian i.i.d. zero-mean entries, the eigenvalues of its sample covariance matrix follow a distribution known as the Marchenko-Pastur distribution [10]. Specifically, the distribution characterizes the behavior of eigenvalues and provides well-defined minimum and maximum eigenvalues of the random matrices.

In the context of statistics, it is not always the case that the purpose of the study lies in the understanding of pure random matrices. In fact, random matrix theory provides insights into the behavior of the random part of observations and, consequentially, the non-random part. In the context of cluster analysis, the non-random part refers to the group structure that is represented by higher internal similarity than external similarity in the similarity matrix. Fortunately, random matrix theory shows that in matrices that are not completely random, the eigenvalues can often be separated into a bulk component and a number of spiked eigenvalues that are distinctively outside the bulk. The well-separated spikes indicate a strong presence of information about the underlying system. On the other hand, if there are no spiked eigenvalues, it is suggestive that no such information is available. In this way, the separation of spiked eigenvalues and the bulk drives a phase transition in the detectability of the information.

In the context of cluster analysis, this distinctive feature of random matrices allows us to gain insights into the existence of group structures in data, reduce dimensions, and determine the number of groups that are present. The eigenvectors and eigenvalues of Hamming similarity matrices, Manhattan similarity matrices, and a large class of Kernel matrices all attain universal behavior that resembles that of the Wishart matrices, which inspires the practical usage of random matrix theory to separate the informative spectral components from the uninformative ones in the following chapters. In the absence of a theoretical threshold for separating the bulk eigenvalues from the spiked eigenvalues of general similarity matrices, we can use a shuffling-based parallel analysis method. This method compares the eigenvalues of a similarity matrix of shuffled data with the original data to determine which eigenvalues belong to informative eigenvectors and which do not.

In Section 3.1, the similarity matrices of interest and their properties are discussed. In Section 3.2, the distribution of eigenvalues of the Wishart matrices is examined. Furthermore, the relationship to the context of our defined similarity matrices is described. In Section 3.3, the phase transition related to the detectability of information within the random matrices is discussed. In Section 3.4, a practical approach to determining the existence of spiked eigenvalues is presented. In particular, this approach can be used in contexts where theoretical insights into the eigenvalue distributions of a completely random matrix are lacking.

3.1. Similarity Matrices

Similarity is typically defined as an inverse relation to a specific distance metric d. In particular, for some metric d there exists a function $s : \mathcal{X} \times \mathcal{X} \to [0, 1]$ that satisfies the following properties. First, an object $\mathbf{x} \in \mathcal{X}$ has unit similarity with itself, i.e., $s(\mathbf{x}, \mathbf{x}) = 1$. Second, for all objects $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, the similarity is greater or equal to zero, i.e. $s(\mathbf{x}, \mathbf{y}) \geq 0$. Third, the similarity function is symmetric, i.e., $s(\mathbf{x}, \mathbf{y}) = s(\mathbf{y}, \mathbf{x})$. Fourth, the similarity function, s, has an inverse relation with the distance metric d, i.e.

$$s(\mathbf{x}, \mathbf{y}) > s(\mathbf{w}, \mathbf{z}) \text{ if and only if } d(\mathbf{x}, \mathbf{y}) < d(\mathbf{w}, \mathbf{z}) \text{ for all } \mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w} \in \mathcal{X}. \tag{3.1}$$

For the theory of random matrices to apply, we require the similarity functions to be associated with a positive definite similarity matrix. A similarity matrix $\mathbf{S}$ is the $n \times n$ matrix associated with a set $\{\mathbf{x}_i\}_{i=1}^n$ of n objects in a data space, i.e., for all $i \in \{1, \ldots, n\}, \mathbf{x}_i \in \mathcal{X}$ and a similarity function s defined on the data space $\mathcal{X}$ that has the following correspondence:

$$\mathbf{S}_{ij} = s(\mathbf{x}_i, \mathbf{x}_j) \qquad \forall i, j \in \{1, \ldots, n\}. \tag{3.2}$$

It is easy to see that $\mathbf{S}$ is a symmetric matrix, as these properties directly transfer from the definition of the distance metric. In particular, we restrict our study to the similarity matrices that are symmetric positive (semi-)definite. A matrix $\mathbf{S}$ is positive definite if, for all non-zero $\mathbf{x} \in \mathbb{R}^n$, we have $\mathbf{x}^\top \mathbf{S} \mathbf{x} > 0$. If the inequality is not strict, the matrix is positive semi-definite. A property that is sometimes used as a definition for positive definiteness is that for a symmetric positive definite matrix $\mathbf{S}$, there exists a matrix $\mathbf{Q}$ that satisfies $\mathbf{S} = \mathbf{Q}\mathbf{Q}^\top$. For this reason, it is easy to see the necessity of the relationship between the positive definite similarity matrices, covariance matrices, which take the form $\mathbf{X}^\top \mathbf{X}$, and Gram matrices, which take the form $\mathbf{X}\mathbf{X}^\top$.

In [37] the positive definiteness of the Manhattan similarity matrix,

$$\mathbf{S}_{ij} = 1 - \frac{|\mathbf{x}_i - \mathbf{x}_j|}{\sup_{i,j \in \{1,\ldots,n\}} |\mathbf{x}_i - \mathbf{x}_j|}, \tag{3.3}$$

and the Hamming similarity matrix,

$$\mathbf{S}_{ij} = \frac{1}{p} |\{l \in \{1, \ldots, p\} \quad : \quad x_i^l = x_j^l\}|, \tag{3.4}$$

is shown.

Furthermore, a well known class of matrices that are often used to represent similarities between objects is the matrices that are associated with kernel functions. A kernel matrix $\mathbf{S}$ is defined by an implicit transformation $\phi : \mathcal{X} \to \mathbb{R}^p$ that satisfies the following:

$$\mathbf{S}_{ij} = s(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)\phi(\mathbf{x}_j)^\top. \tag{3.5}$$

These kernel matrices therefore coincide with the definition of symmetric positive definiteness similarity matrices. A large body of work focuses on the derivation of random matrix equivalents of arbitrary kernel matrices [18, 9]. In this thesis, we limit our study to a commonly used similarity metric for Euclidean distances, as is discussed in [13]. Specifically, we consider a negative exponential of the squared Euclidean distance, scaled by the number of dimensions. To be precise,

$$\mathbf{S}_{ij} = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||_2^2}{p}\right). \tag{3.6}$$

The similarity matrix of this form is symmetric and positive definite. Furthermore, [19] and [18] have shown that a similarity matrix of the form can be asymptotically approximated with a random matrix equivalent, thereby adhering to the universal behavior eigenvalues appear in bulk and spikes.

3.1.1. Spectral Decomposition

The linear algebraic theory that enables a lot of random matrix theory is the spectral decomposition of symmetric positive definite matrices. In particular, if $\mathbf{S}$ is a symmetric positive definite matrix, then there exists a unique decomposition based on an orthonormal matrix $\mathbf{V}$ of which the columns are the eigenvectors of $\mathbf{S}$, and a diagonal matrix $\mathbf{\Lambda}$ of which the diagonal elements are the eigenvalues of $\mathbf{S}$. Specifically, let $\mathbf{S}$ be a matrix of size $n \times n$, then the eigenvalue problem is denoted as

$$\mathbf{S}\mathbf{v}^{(m)} = \lambda_m \mathbf{v}^{(m)} \quad \text{for } m \in \{1, \ldots, n\}. \tag{3.7}$$

The solutions to $\{\mathbf{v}^{(m)}\}_{m=1}^n$ and $\{\lambda_m\}_{m=1}^n$ in the above problem are the eigenvectors and the eigenvalues, respectively. We consider an ordering of the eigenvalues such that

$$\lambda_1 > \cdots > \lambda_n > 0, \tag{3.8}$$

where we know that the eigenvalues are positive because the matrix $\mathbf{S}$ is positive definite. Then, the spectral decomposition of $\mathbf{S}$ is

$$\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top, \tag{3.9}$$

where we define the two matrices $\mathbf{V}$ and $\mathbf{\Lambda}$ as

$$\mathbf{\Lambda} = \mathrm{Diag}(\lambda_1, \ldots, \lambda_n) \text{ and } \mathbf{V} = \begin{pmatrix} \mathbf{v}^{(1)} & \ldots & \mathbf{v}^{(n)} \end{pmatrix}. \tag{3.10}$$

Using this decomposition, it is easy to see the decomposition property of symmetric positive definite matrices, i.e. for

$$\mathbf{Q} = \mathbf{V}\mathbf{\Lambda}^{1/2}, \tag{3.11}$$

we have that

$$\mathbf{S} = \mathbf{Q}\mathbf{Q}^\top. \tag{3.12}$$

If $\mathbf{X}$ is $n \times p$, with n larger than p, then the gram matrix $\mathbf{X}\mathbf{X}^\top$ is not positive definite. To be precise, it is positive semi-definite, which means that there exist zero eigenvalues. This is because $\mathbf{X}\mathbf{X}^\top$ is not a full rank matrix. However, the non-zero eigenvalues of $\mathbf{X}\mathbf{X}^\top$ and $\mathbf{X}^\top\mathbf{X}$ are equivalent. Therefore, if the matrix $\mathbf{S}$ is semi positive definite, a spectral decomposition can still be obtained in a similar manner.

3.2. Eigenvalue Distribution

A large part of the research in random matrix theory is concerned with the eigenvalues of random matrices. In our context, we are mainly concerned with the results related to Wishart matrices. Although Wishart matrices are defined more precisely, these matrices resemble symmetric positive definite matrices and are of interest for studying random similarity matrices. The precise definition of a Wishart matrix is a $n \times n$ random matrix of the form $\mathbf{X}\mathbf{X}^\top$, where $\mathbf{X}$ is a $n \times p$ matrix with independent and identically distributed Gaussian entries.

While Wishart matrices themselves are uniquely defined to be related to matrices with i.i.d. zero mean Gaussian entries, most of the results that we see can be extended to matrices with non-zero mean [74], non-Gaussian [75] and even dependent matrix entries [76]. These properties are of particular importance in the definition of general similarity matrices, as the type of data that we study is not necessarily Gaussian or even continuous. Fortunately, [36] empirically studied the relation between the eigenvalues of Hamming similarity matrices of categorical data and Wishart matrices. Additionally, [21] utilizes results originally derived for Wishart matrices to cosine similarity of high dimensional discrete data, based upon the observation that a finite rank perturbation of the Wishart matrix does not affect the spectrum of a large part of the Wishart matrix. This result is formally derived in [74]. In other words, most of the eigenvalues of a perturbed data matrix $\mathbf{X}$ are distributed as the eigenvalues of Wishart matrices, while only a few spiked eigenvalues are not.

If $\mathbf{X}$ is a $n \times p$ matrix of Gaussian elements $\mathbf{x}_i \sim N(\mathbf{0}, \mathbf{I}_{p\times p})$, then the matrix $\mathbf{S} = \frac{1}{p}\mathbf{X}\mathbf{X}^\top$ is a Wishart matrix. For fixed n and $p \to \infty$, the matrix $\mathbf{S}$ converges to $\mathbf{I}$, as the random vectors $\{\mathbf{x}_i\}_{i=1}^n$ have variance one and are uncorrelated. This is the setting of classical statistics. However, if $p \to \infty$ and $p \to \infty$, such that $q = \frac{n}{p}$, the convergence of $\mathbf{S}$ is no longer trivial and therefore inconsistently estimates the theoretical limit. The only eigenvalue of $\mathbf{I}$ is clearly one. However, in the high dimensional data asymptotic setting, the eigenvalues of $\mathbf{S}$ are not one. In particular, the counting measure of the eigenvalues of $\mathbf{S}$ has a specific limit [10], that is:

$$\frac{1}{n}|\{\lambda \in \{\lambda_m\}_{m=1}^n : \lambda \leq x\}| \to H(x). \tag{3.13}$$

If we define

$$\lambda_- := (1 - \sqrt{q})^2 \text{ and } \lambda_+ := (1 + \sqrt{q})^2, \tag{3.14}$$

then for all $x \in (\lambda_-, \lambda_+)$, the distribution of the eigenvalues is described by the following probability density function:

$$H'(x) = \frac{q^2}{2\pi x}\sqrt{(\lambda_+ - x)(x - \lambda_-)}. \tag{3.15}$$

The probability density is known as the Marchenko-Pastur law [10]. In Figure 3.1 the theoretical eigenvalue distribution of a Wishart matrix is demonstrated. The eigenvalue distributions for three standard Gaussian data matrices with different values for q. When $q > 1$, many eigenvalues are located at 0 due to the rank deficiency of the matrix. The shape of the distribution indicates that as q is smaller, which means the number of features p is much larger than the number of objects n, the eigenvalues of $\mathbf{S}$ approach the eigenvalues of $\mathbf{I}$, indicated by the vertical dashed line at 1. This is the case as for vanishing q, we approach a setting that resembles the classical statistical regime. If q is higher, we clearly see the inconsistency between the eigenvalues of the random matrix $\mathbf{S}$ and the theoretical limit of the classical statistical setting. For higher q, the eigenvalues spread around 1, and the bulk has increased width.

The bulk specified by the Marchenko-Pastur law is relatively sharp. In particular, the largest eigenvalue λ_1 converges almost surely to a fixed value [77]. Specifically, it converges strongly to the right edge of the bulk of the limiting distribution, i.e., with probability one, we have

$$\lambda_1 \xrightarrow{n,p\to\infty} \lambda_+ := (1 + \sqrt{q})^2. \tag{3.16}$$

This indicates that the edge of the bulk is consistently approximated by the limiting value λ_+. In Figure 3.2 this convergence is demonstrated. Each point represents the value of λ_1, depicted on the vertical axis, of a randomly sampled Wishart matrix with p indicated by the horizontal axis. For any q, as p grows, the value of λ_1 is closer to λ_+, which is indicated by the horizontal dashed line, with a vanishing variance. Note that the setting in this figure does not represent a fixed n regime, as n grows with p through the relation $n = qp$.

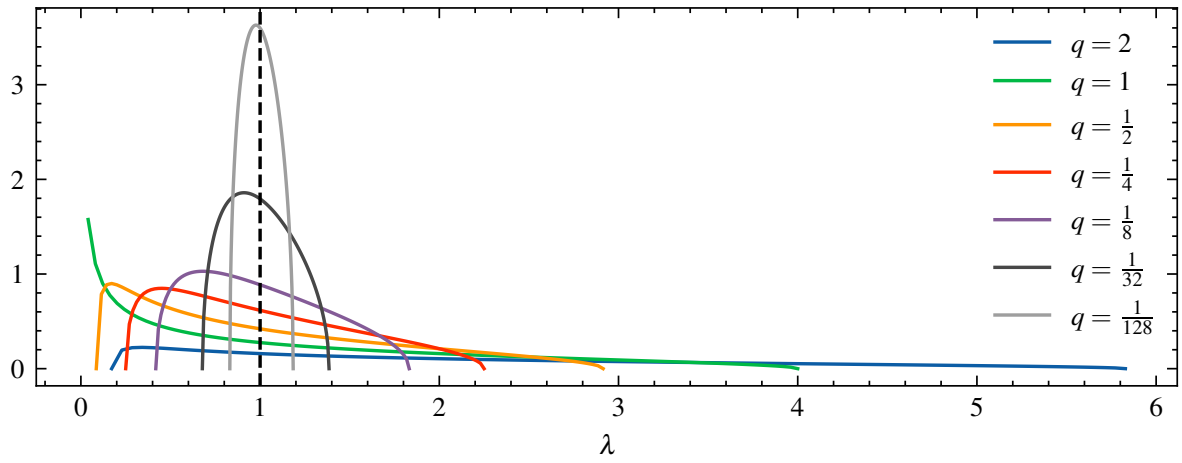


Figure 3.1: Theoretical eigenvalue distribution of Wishart matrix. The vertical dashed line represents the eigenvalues of the population covariance matrix, which corresponds to all eigenvalues being 1.

The exact distribution of the largest eigenvalue λ_1 appears not to be symmetric around the theoretical limit λ_+ [78]. In fact, the distribution can be described by the so-called Tracy-Widom law [79]. The Tracy-Widom law provides a hypothesis testing framework that allows the testing of the existence of eigenvalues outside the bulk. This framework is particularly important when the eigenvalues are close to the bulk edge. However, because little is known about this distribution outside the Wishart setting, we refrain from studying the exact distribution of λ_1 in the context of different similarity matrices that are not strictly Wishart. This can be partially substantiated by the fact that when p is not small (e.g., $p > 100$), the distribution of λ_1 is relatively narrow, and its exact shape arguably becomes less important for practical use.

While, in principle, the Marchenko-Pastur law is defined for sample covariance matrices, we can observe a similar bulky distribution of the eigenvalues in a selection of similarity matrices. In Figure 3.3 the eigenvalue distribution of a selection of random similarity matrices is demonstrated. In the leftmost panel, we see the eigenvalue distribution of a similarity matrix of the form described in Equation 3.6. The data matrix $\mathbf{X}$ contains i.i.d. standard Gaussian entries. In the middle panel, we see the eigenvalue distribution of a Hamming similarity matrix as described in Equation 3.4. The entries of the data matrix are uniformly sampled from $\{1, \ldots, 10\}$. In the rightmost panel, we see the eigenvalue distribution of a Manhattan similarity matrix as described in Equation 3.3. The entries of the data matrix are again uniformly sampled from $\{1, \ldots, 10\}$. In the histograms, we find that most of the eigenvalues are again located at a 'bulk' except for a single large eigenvalue. This single large eigenvalue is indicative of what we consider to be the global component, which is a non-bulk component of the spectrum of similarity matrix that appears in completely random matrices. Therefore, it indicates a sense of global similarity obtained through the structural definition of the metric space and underlying random processes. It is therefore not representative of any group structures, as will be further discussed in Section 3.4.

The distributions of the eigenvalues of these similarity matrices are not theoretically equivalent to that of the Wishart matrices. However, they appear to exhibit similar behavior. First, the eigenvalues are positive due to the positive definiteness of the similarity matrices. Second, the distribution of the eigenvalues displays a bulk that has a relatively strict edge. This resembles the behavior that we find in the limiting distribution of the eigenvalues of Wishart matrices, i.e., the Marchenko-Pastur law displayed in Figure 3.1. However, the eigenvalues of the similarity matrices exhibit a spiked eigenvalue outside the bulk, as displayed in the histograms of Figure 3.3 and not in the Marchenko-Pastur law, which is due to the global similarity structure of random processes and metric space.

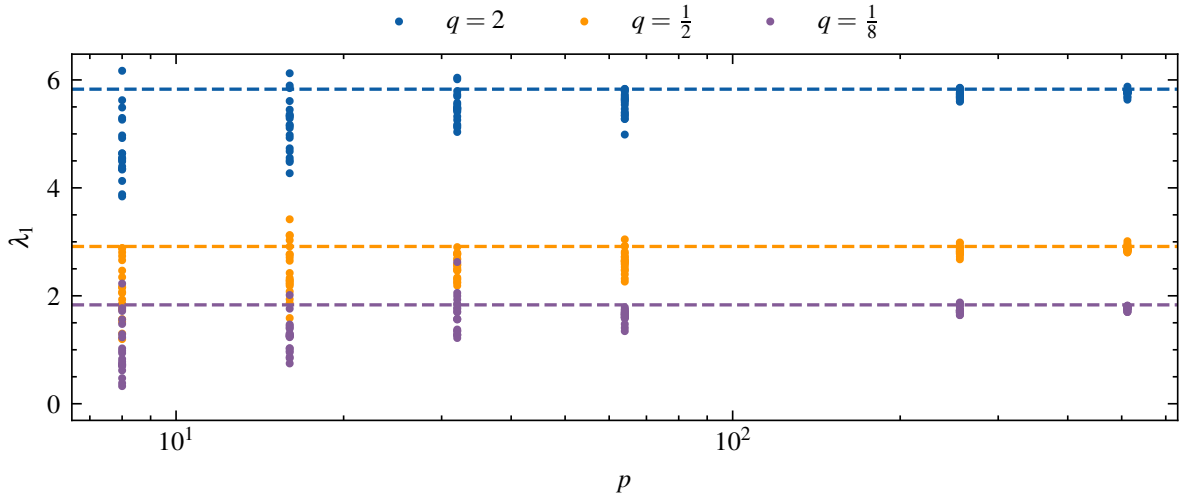


Figure 3.2: Convergence of empirical λ_1 of Wishart matrix. The scatter plot shows λ_1 of 20 samples of a $n \times n$ Wishart matrix $\mathbf{S} = \frac{1}{p}\mathbf{X}\mathbf{X}^\top$ with the rows of the $n \times p$ matrix $\mathbf{X}$ satisfying $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{p\times p})$ for different values of p indicated by the horizontal axis and q indicated by the color. The horizontal dashed line indicates the theoretical limit of the bulk edge, λ_+.

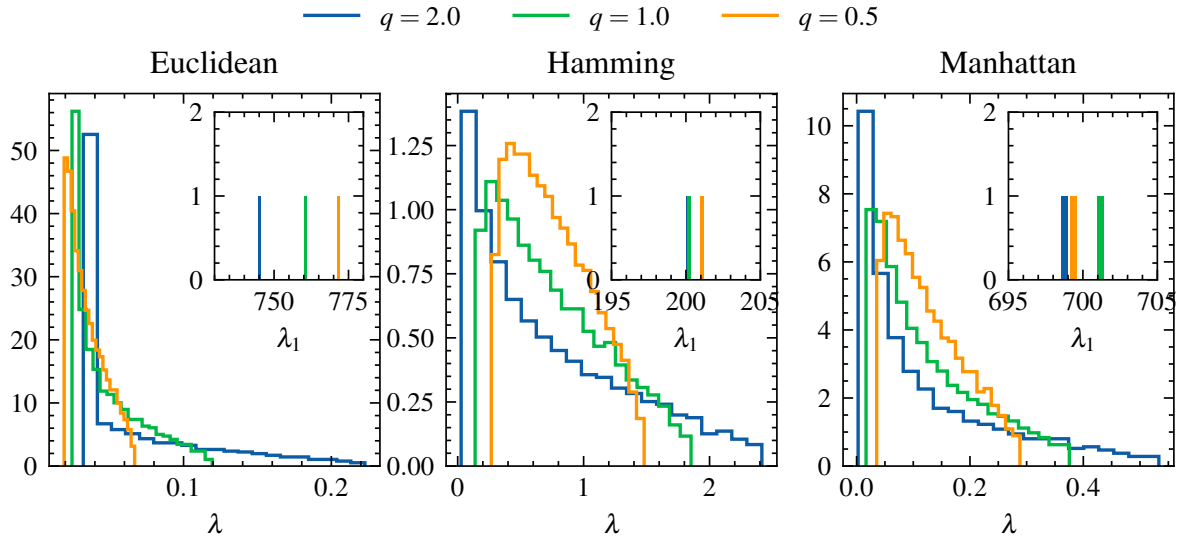


Figure 3.3: Eigenvalue distributions of random similarity matrices. In the leftmost panel, the entries of the data matrix are standard Gaussian and the similarity matrix is defined by Equation 3.6. In the middle panel, the entries of the data matrix are uniformly sampled on $\{1, \ldots, 10\}$ and the similarity matrix is defined by Equation 3.4. In the rightmost panel, the entries of the data matrix are also uniformly sampled on $\{1, \ldots, 10\}$ and the similarity matrix is defined by Equation 3.3. In all cases, $n = 800$ and $q = \frac{n}{p}$

3.3. Phase Transition

The presence of spiked eigenvalues is indicative of more information than purely random. The separation between the bulk and the spikes drives a phase transition in the detectability of this information. The theoretical framework in which the phase transition is studied is often referred to as the spiked eigenvalue model. The framework demonstrates the behavior of eigenvalues and eigenvectors in the presence of a dominant signal. In the setting of similarity matrices, this may correspond to the presence of group components and global similarity components. In this model, the spectrum of the random matrix is characterized by a bulk of eigenvalues that follow a certain distribution, e.g., the Marchenko-Pastur law for Wishart matrices, along with a few isolated eigenvalues, referred to as spikes, that deviate significantly from the bulk.

This spiked eigenvalue model has found applications in various fields such as signal processing, statistical inference, and machine learning. Understanding the behavior of eigenvalues in this model provides insights into the detection of signals in noisy data and the performance of estimation algorithms. Several theoretical results have been established for the spiked eigenvalue model, including phase transition phenomena, asymptotic behavior of eigenvalue distributions, and optimal detection thresholds for signal recovery [43].

To demonstrate the behavior of the spiked eigenvalue model, we consider a model for the columns of a $n \times p$ random matrix $\mathbf{X}$. In particular, let $\mathbf{x}^{(l)} \in \mathbb{R}^n$ denote the lth column of $\mathbf{X}$ for some $l \in \{1, \dots, p\}$. The columns $\{\mathbf{x}^{(l)}\}_{l=1}^{p}$ follow a zero mean multivariate Gaussian distribution, i.e., $\mathbf{x}^l \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_\theta)$ for some $n \times n$ matrix $\mathbf{\Sigma}_\theta$. If we consider the orthonormal matrix $\mathbf{U}$, then we specifically define the covariance matrix $\mathbf{\Sigma}_\theta$ of the multivariate distribution by

$$\mathbf{\Sigma}_\theta = \mathbf{U}\mathrm{Diag}(\theta, 1, 1, \dots, 1)\mathbf{U}^\top \text{ where } \theta > 1 \text{ and } \mathbf{x}^l \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_\theta) \text{ for } l \in \{1, \dots, p\}. \tag{3.17}$$

The reason for the use of this specific example is that it gives us access to a $n \times n$ matrix $\mathbf{S} = \frac{1}{p}\mathbf{X}\mathbf{X}^\top$, which has a known theoretical limiting distribution of the eigenvalues. To be precise, for $\theta = 1$, the eigenvalues follow a Wishart matrix as the elements of $\mathbf{X}$ are i.i.d. standard normal. Furthermore, as q vanishes, $\mathbf{S}$ will converge to $\mathbf{\Sigma}_\theta$ as in the classical regime. On the other hand, if $\theta > 1$, $\mathbf{\Sigma}_\theta$ has all but one eigenvalue set to 1, and one eigenvalue set to $\theta 1$. In essence, this means that for $\theta > 1$, there is one principal component, $\mathbf{u}^{(1)}$, in the theoretical limit of $\mathbf{S}$ that is more significant than the rest. In general, the eigenvalues of $\mathbf{S}$ of high dimensional data are not consistent with those of $\mathbf{\Sigma}_\theta$ due to the fact that q does not vanish. In particular, the eigenvalues of $\mathbf{S}$ are spread between λ_- and λ_+ as defined in Equation 3.14 and therefore the detection of the principal component of $\mathbf{u}^{(1)}$ is not trivial for all $\theta > 1$.

In Figure 3.4 the eigenvalue distributions of both $\mathbf{S}$ and $\mathbf{\Sigma}_\theta$ are displayed. The different colors represent different values of θ. The eigenvalues of $\mathbf{\Sigma}_\theta$ are all 1, indicated by the black vertical line, except for a single remaining eigenvalue which is exactly at θ, indicated by the shaded colored bars. All but one eigenvalue of $\mathbf{S}$ are in a bulk between λ_- and λ_+ that is identical to the eigenvalues of the Wishart matrix, which follow the Marchenko-Pastur distribution. This is the case for all values of θ, suggesting that the bulk behavior is not heavily influenced by the size of the spiked eigenvalue. The remaining spiked eigenvalue of $\mathbf{S}$, λ_1, is positioned at roughly θ, which is indicated by the colored bars. It is noteworthy that the largest eigenvalue of $\mathbf{S}$ is only near θ when θ is large. If θ is one, then λ_1 is near λ_+, which is significantly larger than 1. This is suggestive of the fact that for small θ, the largest eigenvalue of $\mathbf{S}$ is absorbed into the bulk, while for large θ it is not.

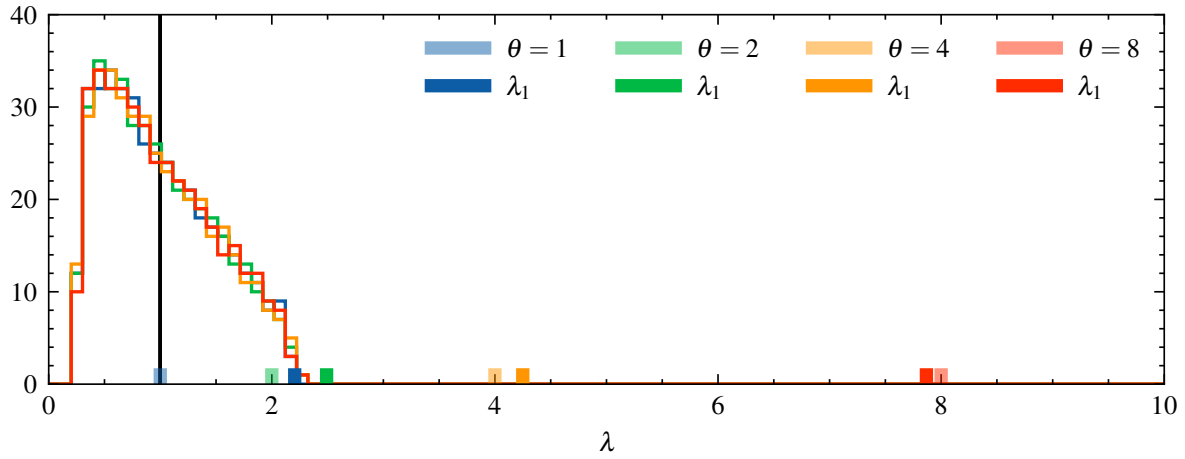


Figure 3.4: Spiked eigenvalue model. The empirical eigenvalue distributions of the spiked eigenvalue model for four different values of $\theta \in \{1, 2, 4, 8\}$ with $n = 400$ and $p = 1600$, i.e. $q = \frac{1}{4}$. The spikes are indicated with the fully colored bars, the values of θ are indicated with the bars with slightly faded colors at positions $\{1, 2, 4, 8\}$. The remaining eigenvalues of the population covariance matrix Σ_θ are indicated by the black vertical line positioned at 1. The bulk of the eigenvalues of the sample covariance matrix are indicated by the colored lines.

This clearly raises questions about the behavior of the eigenvector associated with λ_1, which we denote by $\mathbf{v}^{(1)}$. We know that in the limit of vanishing q the eigenvector would be identical to $\mathbf{u}^{(1)}$. Furthermore, if q is fixed but $\theta \to \infty$, the contribution of the principal component to the variance of $\mathbf{X}$ is so large, that the $\mathbf{v}^{(1)}$ is also identical to $\mathbf{u}^{(1)}$. On the other hand, if $\theta = 1$, the eigenvectors $\mathbf{v}^{(1)}$ and $\mathbf{u}^{(1)}$ are near orthogonal. This is because, the multivariate Gaussian distribution is exactly spherical, meaning that the probability of a particular sample matrix $\mathbf{X}$ is invariant to any rotations. Therefore, the eigenvector associated with λ_1 is completely independent to $\mathbf{u}^{(1)}$. Furthermore, in high dimensional settings, random vectors are typically near orthogonal [3], and therefore the principal component of $\mathbf{S}$ and $\mathbf{\Sigma}_\theta$, for $\theta = 1$, are near orthogonal too.

This suggests that there is a phase transition that depends on θ, and implicitly on the separation between the bulk eigenvalues and the spiked eigenvalues, that differentiates the two phases where the eigenvectors align and where the eigenvectors do not align. Because of the theoretical framework, the detection of these principal components can be studied explicitly [11]. In particular, we define the alignment between the two vectors by

$$\phi = |(\mathbf{v}^{(1)})^\top \mathbf{u}^{(1)}|. \tag{3.18}$$

Clearly, if $\phi = 0$, the vectors are orthogonal, and if $\phi = 1$ the vectors are identical up to a sign change. The intuition described above about the phase transition is proven in [43], where it is shown that for large θ ϕ is close to 1 and for small θ, ϕ is close to 0. This corresponds to the idea that the spiked eigenvalues, associated with the large θ case, are associated with eigenvectors that contain information about the data, while the eigenvalues in the bulk, associated with the small θ case, do not.

In Figure 3.5, we demonstrate the phase transition of the angular alignment of eigenvectors ϕ as a function of θ. This phase transition indeed shows the relevance of the right bulk edge that is present in the histograms in Figure 3.1 and Figure 3.3. The exact behavior of the eigenvalues around the edge is relatively complicated, as indicated by the smooth transition in Figure 3.5. The behavior of the eigenvalues around the bulk edge is beyond the scope of this thesis, however, it is studied in depth in [80]. Because we do not rely on any theoretical derivations of eigenvalue thresholds or phase transitions due to the deviation from the exact Wishart model, we use a heuristic tool that can be used to distinguish significant spiked eigenvalues from the bulk.

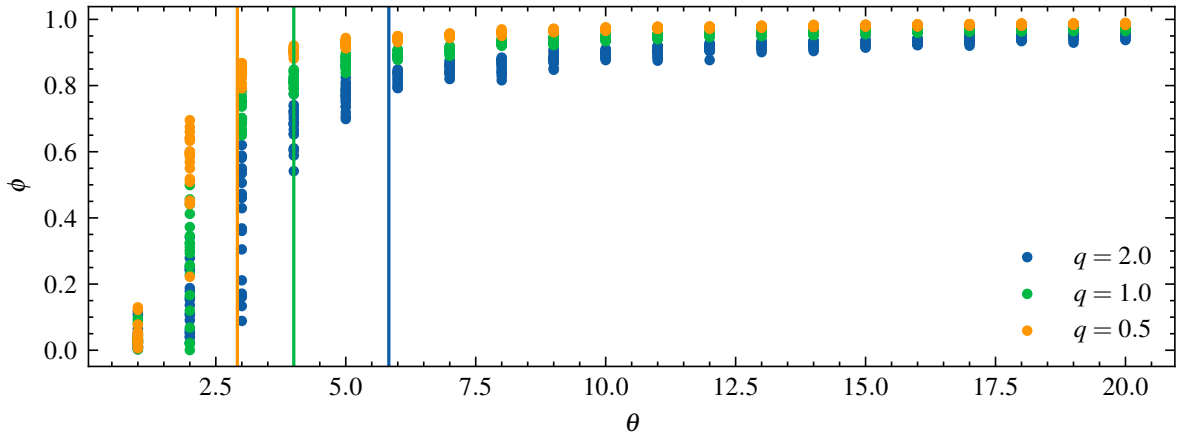


Figure 3.5: Phase transition of eigenvector alignment. The entries of the data matrix are distributed according to the model defined by Equation 3.17 with $n = 200$. The vertical axis displays the eigenvector alignment ϕ defined in Equation 3.18. If $\phi = 1$ the eigenvector associated with the largest eigenvalue of $\mathbf{X}\mathbf{X}^\top/p$ and $\mathbf{\Sigma}_\theta$ are identical. If $\phi = 0$ the eigenvectors are orthogonal. The horizontal axis displays the value for θ as defined in Equation 3.17. The vertical lines indicate the value of the right edge of the bulk, λ_+

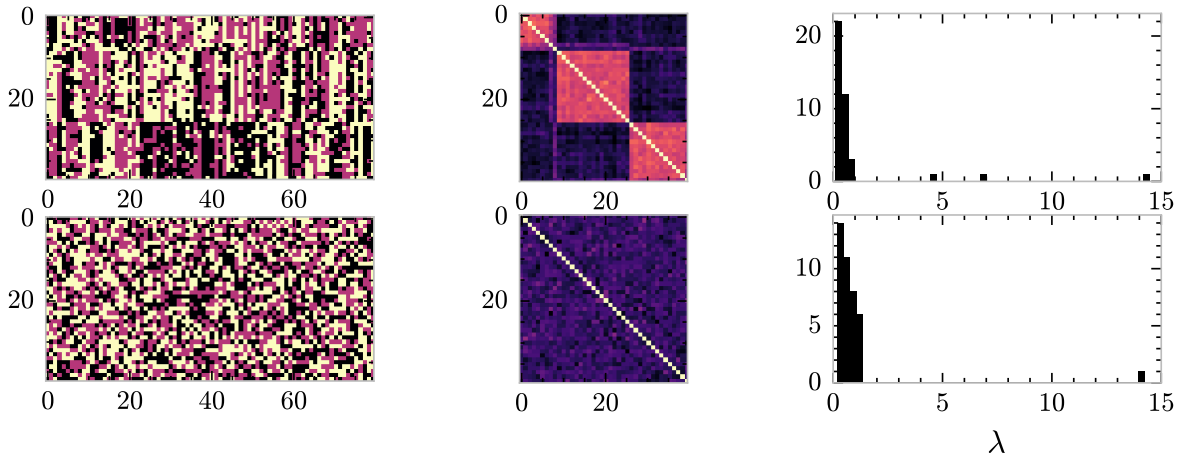


Figure 3.6: Data and shuffled data The top row represents the original data. The bottom row represents the shuffled data. The left column represents the 40×80 data matrices. The middle column represents the corresponding 40×40 similarity matrices. The right column represents the eigenvalue distributions of the similarity matrices.

3.4. Parallel Analysis

In a practical setting, we often encounter similarity matrices that contain information about group structure, but we do not have access to a theoretical derivation of the null model. Therefore, a bridge between the non-null and the null models, where no group structure exists, needs to be constructed. In particular, we must determine an eigenvalue threshold τ that significantly separates the eigenvalues of the bulk from the spiked eigenvalues. For Gaussian data, we can use the theoretical threshold implied by the Marchenko-Pastur law. However, for non-Gaussian data, theoretical thresholds are unknown. To surpass this, a commonly used heuristic is parallel analysis [44], which is done by comparing the eigenvalues of a reasonable null model with the actual observed eigenvalues. Along these lines, a practical approach to approximate τ is to use permutation based parallel analysis [81], which uses a data shuffling procedure to obtain a sample of the presumptive null model. In this way, one removes any structural groups, in order to observe how the spectrum of a random matrix that approximately follows the same marginal statistics behaves.

The specific shuffling procedure constructs a new matrix $\mathbf{X}'$ that is of identical size as $\mathbf{X}$, which shuffles the entries of each column l separately. In this way, if objects from a specific group have specific correlated features, these correlations are no longer present in the shuffled matrix. However, the global correlations among features that are present in all the objects in the data set are still maintained. Then, using the eigenvalues of the similarity matrix derived from $\mathbf{X}'$, we can distinguish the eigenvalues that belong to the bulk from those that belong to the spikes. To be specific, the shuffling procedure is performed N times. For each iteration, the similarity matrix is computed for the shuffled matrix. Then, the second-largest eigenvalue λ_2 is saved. After the N iterations, the average of all the observed second-largest eigenvalues is computed with

$$\bar{\tau} = \frac{1}{N}\sum_{r=1}^{N}\lambda_2^{(r)}, \tag{3.19}$$

where $\lambda_2^{(r)}$ denotes the second-largest eigenvalues of the rth shuffle iteration. Furthermore, we compute the standard deviation of the observed second-largest eigenvalues with

$$\sigma = \sqrt{\frac{1}{N}\sum_{r=1}^{N}(\lambda_2^{(r)} - \bar{\tau})^2}. \tag{3.20}$$

Then, a heuristic threshold that separates the eigenvalues associated with the bulk and the spikes is

$$\tau = \bar{\tau} + 2\sigma. \tag{3.21}$$

In Algorithm 1 we provide a formal description of the shuffling algorithm that determines a threshold τ, such that eigenvalues above τ are considered spiked eigenvalues. The number of shuffling operations and the confidence interval to determine τ are parameters of this algorithm. To limit the scope of this thesis, we choose a relatively conservative 2 standard deviations, such that the number of shuffling operations is not too influential.

In Figure 3.6 we illustrate the intuition behind the shuffling procedure on a data set with $n = 40$ and $p = 80$. The top row corresponds to the observed data set and the bottom row corresponds to the shuffled data set. In the left column of the figure, we display the two $n \times p$ data matrices of categorical entries. The different colored pixels represent different values. The rows representing objects are ordered such that objects of the same group are adjacent. In the observed data matrix, we clearly see a resemblance of the colors among objects of the same group. In the shuffled matrix, no such pattern can be seen. In the middle column of the figure, we display the $n \times n$ Hamming similarity matrices. Brighter colors indicate higher similarity. In the observed similarity matrix, we clearly see a group structure of 4 groups indicated by the diagonal blocks. Again, in the similarity matrix of the shuffled data, no such pattern can be recognized. Finally, in the right column, we see histograms depicting the eigenvalues of the observed similarity matrix and the shuffled similarity matrix.

The eigenvalue distribution of the observed similarity matrix clearly shows 4 visible spiked eigenvalues that are significantly separated from the bulk on the left of the histogram. Upon shuffling the data matrix and recomputing the eigenvalues of the associated similarity matrix, most of the spikes are no longer visible. Instead, there is a bulk and a single spiked eigenvalue. The bulk eigenvalues are associated with the random components, while the remaining spiked eigenvalue is associated with the global component. The global component can be recognized from the eigenvalue distributions of the completely random similarity matrices depicted in the histograms of Figure 3.3. The difference between the spectrum of the observed similarity matrix and the spectrum of the shuffled similarity matrix motivates the subtraction of the random component, i.e., the eigenvalues below the threshold τ, and the global component, the largest eigenvalue, from the observed spectrum. The remaining 3 eigenvalues are associated with the modularity component, which is the core concept in spectral modularity and is discussed in Section 4.3.

Because the eigenvectors are orthogonal, it is tempting to interpret the number of spiked eigenvalues as the number of groups. The reasoning behind this is as follows. The largest eigenvalue is associated with the global component, i.e., it represents the information that is present in all the objects. The second-largest spiked eigenvalue is orthogonal to the largest, and therefore contains information that distinguishes the similarities of a selection of objects from the remaining objects. The continuation of this reasoning leads to the conjecture that the $\hat{K}$th spiked eigenvalue, leads to the distinction of K groups. Then, if the $\hat{K}+1$th eigenvalue is not a spiked eigenvalue, i.e., it is absorbed in the bulk, the eigenvector associated with it is suggested to contain no further information about the group structure. Therefore, the number of spiked eigenvalues, i.e.

$$\hat{K} = |\{\lambda \in \{\lambda_{m=1}^{n}\} : \lambda > \tau\}|, \tag{3.22}$$

is used to approximate the number of groups K. It should be noted that, the relation between the number of spiked eigenvalues and the proposed number of groups is a relatively ad hoc assumption, of which the exact theoretical inclinations are not known and may lead to conceptual mismatches when the groups in a data set are explicitly ordered hierarchically.

A final observation from the comparison of the two similarity matrix spectra in the histograms of Figure 3.6, is the deviating bulk shapes. In principle, this leads to an important challenge of the heuristic. When spiked eigenvalues are close to the bulk, the shuffling procedure may not lead to a correct estimate of the threshold. In most of our study, the studied data matrices are specified such that this does not lead to problems, however in Chapter 11 we uncover that in realistic applications the shuffling procedure may provide suboptimal thresholds.

Algorithm 1 Shuffling Based Parallel Analysis

Input: $n \times p$ data matrix $\mathbf{X}$ with for all n rows, $\mathbf{x}_i \in \mathcal{X}$, similarity metric $s : \mathcal{X} \times \mathcal{X} \to [0, 1]$, number of shuffles N

Output: Eigenvalue threshold τ and number of spiked eigenvalues $\hat{K}$

1. **Non-random eigenvalues** We solve the eigenvalue problem for matrix $\mathbf{S}$, to obtain all the eigenvalues $\{\lambda_m\}_{m=1}^n$.
2. **Random eigenvalues** for $r = 1, \ldots, N$ do
   - Let $\mathbf{X}'$ be a $n \times p$ matrix in which the rows share the same data space as $\mathbf{X}$, i.e. $\mathcal{X}$.
   - For $l = 1, \ldots, p$, consider the lth column of, $\mathbf{X}$ denoted by $\mathbf{x}^{(l)}$. Then, shuffling the elements of $\mathbf{x}^{(l)}$ can be done by formally by considering the indices $\{1, \ldots, n\}$ being randomly reordered, i.e., uniformly sampled over the space of all $n!$ possible permutations. Let $j_1, \ldots, j_n$ denote such a randomly sampled permutation. Then,
   $$x_i^{(l)'} = x_{j_i}^{(l)} \text{ for all } i \in \{1, \ldots, n\}. \tag{3.23}$$
   - We compute the similarity matrix of the shuffled data set $\{\mathbf{x}'_i\}_{i=1}^n$, with
   $$\mathbf{S}'_{ij} = s(\mathbf{x}'_i, \mathbf{x}'_j). \tag{3.24}$$
   - We solve the eigenvalue problem for matrix $\mathbf{S}'$, to obtain the second-largest eigenvalue, $\lambda_2^{(m)}$, i.e.
3. **Comparison** We compute the sample mean of $\{\lambda_2^{(r)}\}$ and denote this by $\bar{\lambda_2}$, i.e.
   $$\bar{\tau} = \frac{1}{N} \sum_{r=1}^{N} \lambda_2^{(r)}. \tag{3.25}$$
   Furthermore, we compute the sample standard deviation, i.e.
   $$\sigma = \sqrt{\frac{1}{N} \sum_{r=1}^{N} (\lambda_2^{(r)} - \bar{\tau})^2}. \tag{3.26}$$
   Then, we determine a relatively conservative value for τ by increasing the threshold by 2 standard deviation above the mean.
   $$\tau = \bar{\tau} + 2\sigma. \tag{3.27}$$
   Finally, we can determine $\hat{K}$ by counting the number of eigenvalues of $\mathbf{S}$ that are larger than the threshold τ.
   $$\hat{K} = |\{\lambda \in \{\lambda_{m=1}^n\} : \lambda > \tau\}|. \tag{3.28}$$

4
Modularity

In this chapter, we discuss modularity as originally introduced in [12] through the lens of the spectra of random matrices, which we refer to as spectral modularity. This interpretation leads to the method that is introduced in [8] for the task of clustering time series based on correlation matrices. The essence of modularity is to quantify a relative definition of similarity instead of an absolute quantity. In particular, observed relations between objects are compared to an expectation of that relation in a completely random setting, the null model.

Modularity, as introduced [12], quantifies the quality of an estimation of the community structure in a graph and can be used as an objective in an optimization problem that finds a clustering that has a higher level of internal connection than expected and a lower level of external connection than expected. The maximization of modularity has since been a standard way of detecting communities in networks [82]. Modularity maximization is NP-Hard, but can be heuristically maximized with greedy methods such as Louvain.

For the purpose of clustering multivariate data with similarity matrices, spectral modularity can be used, which is based on the separation of informative eigenvectors from the uninformative ones. This is done by filtering out the global component, associated with the largest eigenvalue, and the random component, associated with the bulk eigenvalues, from the spectral decomposition of a symmetric positive definite similarity matrix, as discussed in Chapter 3.

Spectral modularity is also a quantification of the relative similarity between pairs of objects. Both positive and negative values are spanned by the values in a spectral modularity matrix, as opposed to similarity, which only spans positive numbers. It transforms the similarity of objects into a quantification, where an indifference of cluster memberships is expressed at 0. Therefore, a positive modularity between two objects is associated with a higher similarity than would be expected in a null model. In this way, the (spectral) modularity of a partition can be computed by summing over all the pairwise (spectral) modularity values within each group of the partition.

In Section 4.1 we discuss the traditional interpretation of modularity and how it can be used to detect communities in networks. In Section 4.2 we elaborate on the maximization procedure of the optimization problem that maximizes the modularity objective. In Section 4.3 we discuss the interpretation of modularity of multivariate data through the lens of random matrices and how it leads to spectral modularity.

4.1. Girvan-Newman Modularity

Networks and complex systems, or more formally, graph-structured data, are often composed of nodes (objects) and weighted edges (relations). For example, in a social network, objects are people, and edges are (weighted) friendships between them. In sets of objects with attributes, a graph can be constructed by setting the similarity between objects as weighted edges. The latter interpretation directly contextualizes our line of research. A common objective in the analysis of these systems is to detect and recover groups of nodes that have a group structure with high internal similarities and low external similarities. A well-known quantification of the nodal organization is Girvan-Newman modularity, which is commonly referred to as simply modularity.

The modularity objective can be seen as a quality function from the space of partitions to a real number $Q : \mathcal{P} \to \mathbb{R}$, akin to the definition in Equation 2.7. Specifically, if we consider an $n \times n$ adjacency matrix $\mathbf{A}$ of a particular graph representation of a dataset and the node degrees $\{d_i\}_{i=1}^n$, then the modularity of a partition $\rho = \{C_1, \dots, C_K\}$ is defined to be proportional to

$$Q(\rho) \propto \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \in C_k} \mathbf{A}_{ij} - \frac{d_i d_j}{2m}. \tag{4.1}$$

The elements in this particular sum can be thought of as a subtraction of a null model.

Specifically, for the adjacency matrix $\mathbf{A}$, we can rewire all the edges such that one end is fixed, and the other end is rewired to a random node. This way, we obtain a random graph where the existence of an edge is Bernoulli distributed with probability $\frac{d_i d_j}{2m}$, where m is the number of edges in the graph. Formally, if we denote the adjacency matrix of this random graph as $\tilde{\mathbf{A}}$, we have that it is a random matrix of which the elements are distributed according to

$$\tilde{\mathbf{A}}_{ij} \sim \text{Ber}\left(\frac{d_i d_j}{2m}\right). \tag{4.2}$$

If we write the node degrees as a vector, i.e., $\mathbf{d} = \begin{pmatrix} d_1 \dots d_n \end{pmatrix}^\top$, we can see that the expected value of this random matrix, which represents the randomized graph associated with $\mathbf{A}$, is

$$\mathbb{E}[\tilde{\mathbf{A}}] = \frac{\mathbf{d}\mathbf{d}^\top}{2m}. \tag{4.3}$$

Then, the modularity objective can be written in terms of a subtraction of the expectation of a random matrix associated with a null model of the particular observed graph, i.e.

$$Q(\rho) \propto \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \in C_k} \left(\mathbf{A} - \mathbb{E}[\tilde{\mathbf{A}}]\right)_{ij}. \tag{4.4}$$

This conveys that a random graph that retains information about global connectivity, i.e., the degrees of vertices remain the same, can be used to provide a relative quantification of the connectivity. In Figure 4.1, we illustrate the observed graph and a randomized graph. The nodes are positioned in a circle, such that the nodes in the lower half represent one group, indicated by the black color, and the nodes in the upper half represent another group, indicated by the gray color. In Figure 4.2, we display associated adjacency matrices. The left graph and left matrix, in Figure 4.1 and Figure 4.2 respectively, the observed graph and the observed adjacency matrix, $\mathbf{A}$, are illustrated. The black matrix elements represent the edges between two nodes. From the figures, we clearly see that the internal connectivity of the groups is significantly higher than the external connectivity, as indicated by the relatively few edges between the two groups. In the second-to-left graph and second-to-left matrix, in Figure 4.1 and Figure 4.2 respectively, we display the random graph and the associated random adjacency matrix $\tilde{\mathbf{A}}$ that follows from the random sampling of the model specified in Equation 4.2. Here, the degrees of the nodes are identical to the observed graph, but the clear distinction between the upper and lower half groups is no longer visible.

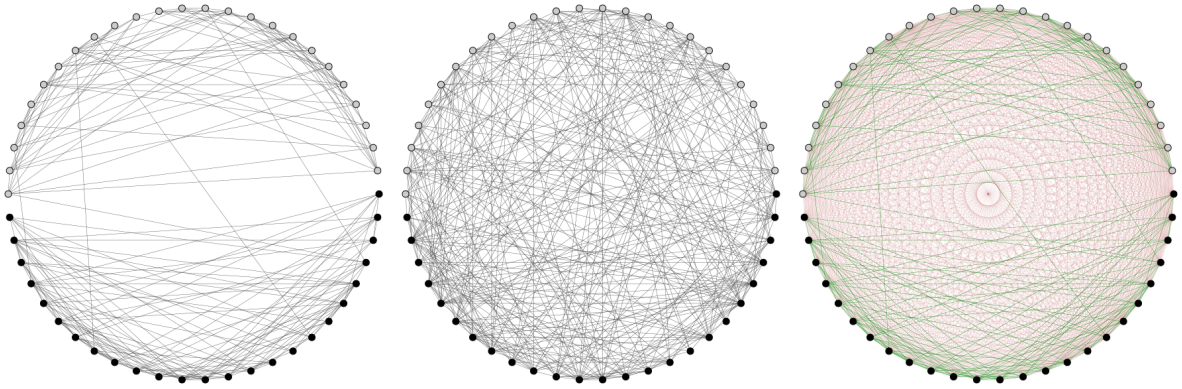


Figure 4.1: Graphs with and without clustered nodes. In the left figure, the graph has two strongly connected clusters, namely the upper half of the circle and the lower half of the circle. This means that there are many internal edges between nodes in the lower half and in the upper half, but there are only a few edges between the lower half and the upper half. The associated adjacency matrix is given in the left panel of Figure 4.2. In the middle figure, we see a sample adjacency matrix $\tilde{\mathbf{A}}$ of the associated random graph model defined in Equation 4.2, of which the expected value as defined in Equation 4.3 is given in the middle panel of Figure 4.2. In the right graph, the modularity graph is illustrated, where the edges are weighted with the pairwise modularity values. Red edges indicate negative modularity, and green edges indicate positive modularity. In the right panel of Figure 4.2, the associated adjacency matrix is given.

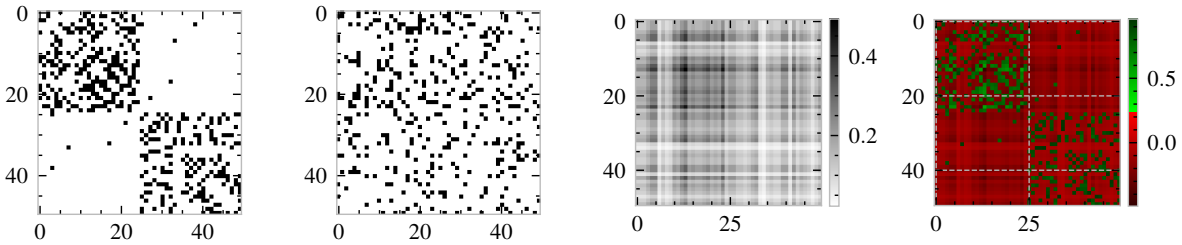


Figure 4.2: Adjacency matrices with and without clustered nodes. The left figure displays the adjacency matrix $\mathbf{A}$, where black indicates an edge between the two nodes and white indicates no edge. The left graph displayed in Figure 4.1 is associated with the same adjacency matrix. The second-to-left figure displays a sample of $\tilde{\mathbf{A}}$ as defined in Equation 4.2. The middle graph in Figure 4.1 is a sample of the identical model. The second-to-right figure displays $\mathbb{E}\left[\tilde{\mathbf{A}}\right]$. The right matrix shows the matrix that represents the pairwise modularity between two nodes.

In the second-to-right matrix in Figure 4.2, we see the associated expected value of the random graph model, i.e., $\mathbb{E}[\tilde{\mathbf{A}}]$. This matrix essentially reflects the degree of heterogeneity among nodes. Therefore, when used as a subtraction term in the modularity definition, the existence of certain edges in $\mathbf{A}$ is valued more than others. In particular, edges on which the nodes both have high degrees are more likely to be connected, therefore, these connections are considered less important for the detection of modular structures. At the same time, the edges on which the nodes both have low degrees are less likely to be connected, making these connections more important for the detection of the groups. This perspective on the weighing of the edge contributions to the modularity of a partition is reflected in the subtraction of $\mathbb{E}[\tilde{\mathbf{A}}]$.

In the right graph in Figure 4.1 and the right adjacency matrix in Figure 4.2, we display a combination of the two concepts. In essence, the graph that is visualized is a weighted graph with adjacency $\mathbf{A} - \mathbb{E}[\tilde{\mathbf{A}}]$ from Equation 4.4. The green edges and matrix elements correspond to positive pairwise modularity between nodes, while the red edges and matrix elements indicate negative pairwise modularity. In this way, objects from the two different groups that have a positive link are still repelled in the objective by the many negative links between neighbors of the objects. And nodes from the same group that have a negative link are still attracted by the many positive links among the neighbors.

4.2. Maximizing Modularity

The modularity maximization problem is defined as a maximization of Q from Equation 4.1 over the space of partitions:

$$\max_{\rho \in \mathcal{P}} Q(\rho). \tag{4.5}$$

There is an exact correspondence between this definition and the clustering objective defined in Section 2.3, specifically Equation 2.7. Therefore, Q neatly fits in the framework of quantifying the quality of a partition.

The modularity maximization problem is in fact an integer optimization problem on the space of all possible partitions, which is extremely large, as discussed in Section 2.2. In particular, it can be shown that modularity maximization is related to a 3-partition; therefore, this maximization problem is NP-Hard [83].

Numerous greedy approaches exist that are able to provide high quality graph partitions in many cases [82]. A particular successful method is Louvain [39]. The Louvain method consists of two phases. In the first phase, objects are moved around sets such that the total modularity is maximized until no move can be made to improve the modularity. In the second phase, all sets in the partition found in the first stage are used to construct a new graph, on which we continue our search by employing the first phase again. These alternating phases continue until no improvements can be made in the first step of the first phase.

A property of the modularity method that enables the Louvain method is the ability to explicitly express the change in the objective upon moving one object i from a specific cluster C_{k_i} to a cluster C_h for some partition $\rho = \{C_1, \ldots, C_k, \ldots, C_h, \ldots, C_K\}$. To be precise, if we denote $\Delta_{i \to C_h}$, we have that

$$\Delta_{i \to C_h} Q(\rho) = -\sum_{j \in C_{k_i}} \left(\mathbf{A}_{ij} - \frac{d_i d_j}{2m} \right) + \sum_{j \in C_k} \left(\mathbf{A}_{ij} - \frac{d_i d_j}{2m} \right). \tag{4.6}$$

Using this value for each iteration of the first phase, only $O(n)$ computations are required to determine the most optimal move. This makes the method relatively efficient; it has an empirical time complexity of $O(n \log n)$, but a theoretical proof is not known. Furthermore, due to its greedy nature, it is likely that the found local optimum is not a global optimum, sometimes leading to arbitrarily bad clusterings. This makes clustering with the Louvain method not ideal in certain settings [39].

4.3. Spectral Modularity

In the context of graph clustering, the definition of modularity uses random matrices in the form of expected values of null models to separate local connectivity behavior from its global connectivity behavior. When considering multivariate data, however, random matrix theory enables an alternative approach to separate the observation from the null model. In particular, we do not have to choose an implicit null model, as we directly extract the null model from the spectral information of the matrix.

Instead of using modularity to cluster nodes in a graph, we use it to cluster objects in a data set. Taking inspiration from the Girvan-Newman modularity matrix that defines the pairwise modularity between nodes, we can describe an alternative for this modularity matrix that makes use of random matrix theory. Modularity optimization in the context of multivariate data can be done by replacing the null model of a graph with a null model based on random matrix theory. Specifically, the behavior of eigenvalues that are present in a random matrix, without any specific group structure, can be used to filter out the informative eigenvalues from the uninformative ones. This way, the spectral decomposition of a similarity matrix is filtered such that only the spike eigenvalues that are associated with informative eigenvectors remain. Notably, this filtering procedure aligns well with the concept of modularity, thereby motivating the use of modularity maximization.

4.3.1. Spectral Modularity Matrix

For some symmetric positive definite matrix $\mathbf{S}$, we can write the spectral decomposition, as introduced in Section 3.1 in terms of its eigenvectors and eigenvalues, i.e., for some objects $i, j \in \{1, \ldots, n\}$

$$\mathbf{S}_{ij} = \sum_{m=1}^{n} \lambda_m v_i^{(m)} v_j^{(m)}, \tag{4.7}$$

where $\{\lambda_m\}_{m=1}^n$ are the eigenvalues of $\mathbf{S}$ and $\{\mathbf{v}\}_{m=1}^n$ are the associated eigenvectors, with $v_i^{(m)}$ denoting the ith entry of the mth eigenvector. Alternatively, we can write

$$\mathbf{S} = \sum_{m=1}^{n} \lambda_m \mathbf{v}^{(m)} (\mathbf{v}^{(m)})^\top. \tag{4.8}$$

An example of the eigenvalues associated with the symmetric positive definite matrix $\mathbf{S}$ is given in Figure 4.3. From the study of the random similarity matrices in Chapter 3, we know that comparing the eigenvalue spectrum of similarity matrices of randomly shuffled data gives us insights into the separation of the informative eigenvectors from the uninformative ones.

Global Component

In similarity matrices of data, it is common to assume that there exists a global component to the similarities between objects that is motivated by the existence of a dominant pattern in every object, irrespective of the groups. In the context of multivariate data, this may correspond to shared values among features of every object, irrespective of the group. But even without the consideration of such a dominant pattern, the constraints of the metric space make a global component unavoidable. For example, in a categorical space of $\{1, \ldots, M\}$, the probability of two pure random samples being equal is $\frac{1}{M^2}$, resulting in an expected Hamming similarity of $\frac{1}{M^2}$. As discussed in Section 3.2, in Figure 3.3 we see the metric space induced global component of three completely random similarity matrices, as indicated by the large eigenvalue depicted in the inset of the figures.

To see how the global component is defined in the presence of a ground-truth partition representative of the group structure in the data, we consider a p dimensional data set of n objects, for which we elaborate on the asymptotic behavior. If n is fixed and $p \to \infty$, we obtain an asymptotic similarity that is representative of the group structure and clear of any noise. Indeed, if the ground-truth partition is sensible, the internal similarities are higher than the external similarities. Using this asymptotic context, we see that if the asymptotic external similarity is non-zero, a global component must be well-defined. Indeed, objects of different groups that have positive asymptotic similarity can only exist if there is a dominant pattern or metric space constraints that produce a background level of external similarity. Conceptually, the global component is represented by the dominant component in the similarity matrix and is therefore associated with the eigenvector corresponding to the largest eigenvalue of the similarity matrix:

$$\mathbf{S}^{(g)} = \lambda_1 \mathbf{v}^{(1)} (\mathbf{v}^{(1)})^\top. \tag{4.9}$$

If the asymptotic external similarity vanishes, which may happen in certain cases, the global mode vanishes. In practice, the similarities between objects are not asymptotic and are in fact obtained from a finite dimensional data set, which means a global component of the similarity is present anyway.

In the second-to-right matrix of Figure 4.4, we see the global component $\mathbf{S}^{(g)}$, of which the associated eigenvalue is depicted in Figure 4.3. From the histograms, we see that upon shuffling the data and therefore removing the group structure, the global component of the shuffled similarity matrix is still observed, as indicated by the green colored histogram.

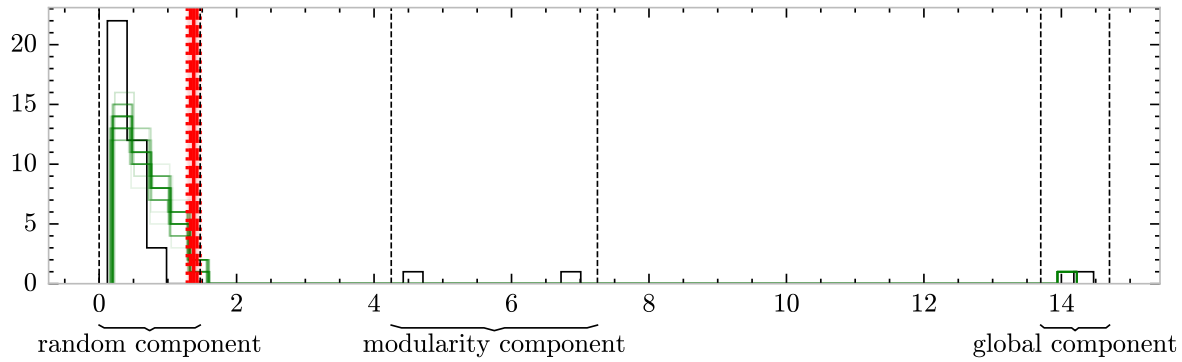


Figure 4.3: Spectral modularity histogram. The black solid line represents the eigenvalues of the observed similarity matrix, which is identical to the similarity matrix in Figure 4.4 and Figure 3.6. The green lines show the eigenvalue distributions of similarity matrices obtained from 50 shuffling procedures. The dashed lines are illustrative of the different components of the eigenvalues. The red vertical solid line represents $\bar{\tau}$, while the red dashed and dotted lines represent $\bar{\tau} \pm \sigma$ and $\bar{\tau} \pm 2\sigma$, respectively, for the standard deviation σ of the observed second-largest eigenvalues during the shuffling procedure. The threshold used to separate the bulk from the spikes is $\tau = \bar{\tau} + 2\sigma$

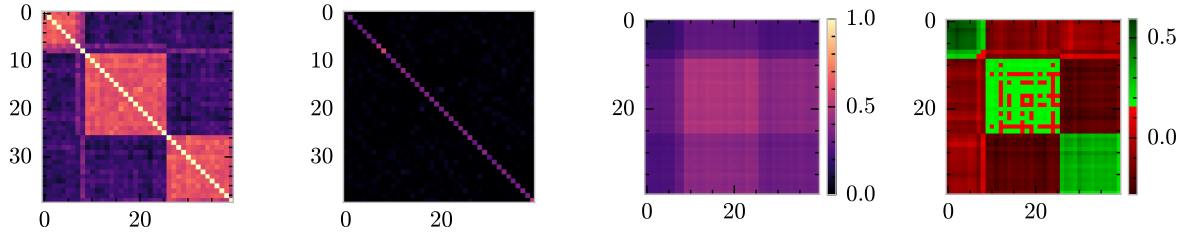


Figure 4.4: Random, global, and modularity matrix. The left figure displays a $n \times n$ similarity matrix, where the setting is identical to that of Figure 3.6 and Figure 4.3. The second-to-left figure displays the random component $\mathbf{S}^{(r)}$, associated with the eigenvalues in the random component of Figure 4.3. The second to right figure displays the global component $\mathbf{S}^{(g)}$, associated to the eigenvalues in the global component of Figure 4.3, associated to the eigenvalues in the modularity component of Figure 4.3. In these images, the values are given by the color bar, where brighter colors represent higher values. The rightmost figure displays the modularity matrix $\mathbf{B}$, where the red colors represent negative values and the green colors represent positive values.

Random Component

The finiteness of data makes similarity measurements subject to noise. From studying the eigenvalues and eigenvectors of random matrices in Section 3.3, we know that there exists a threshold τ, such that the eigenvalues below that threshold are associated with eigenvectors that do not contain information about the group structure, i.e., the bulk, while the eigenvalues that are above the threshold τ do correspond to informative eigenvectors, i.e., the spikes. Therefore, given this threshold τ that separates the spiked eigenvalues from the bulk, we can construct the following interpretation of the random component associated with the similarity matrix:

$$\mathbf{S}^{(r)} = \sum_{m \in \{1,\dots,n\}:\lambda_m < \tau} \lambda_m \mathbf{v}^{(m)} (\mathbf{v}^{(m)})^\top, \tag{4.10}$$

where the threshold τ can be obtained from a procedure as described in Algorithm 1, and of which $\bar{\tau}$ is depicted in red in Figure 4.3. Given that we use $\tau = \bar{\tau} + 2\sigma$, as discussed in Section 3.4 and indicated by the vertical dotted line to the right of τ, the threshold clearly separates the $\hat{K} = 3$ spikes from the bulk of the eigenvalues. In the second-to-left matrix of Figure 4.4, we see the matrix associated with the random component.

Modularity Component

Now that we have introduced the global component $\mathbf{S}^{(g)}$ and the random component $\mathbf{S}^{(r)}$, we can use these to define the spectral modularity matrix by subtracting the components associated with a null model from the similarity matrix, which reads

$$\mathbf{B} = \mathbf{S} - \mathbf{S}^{(g)} - \mathbf{S}^{(r)} = \sum_{m=2}^{\hat{K}} \lambda_m \mathbf{v}^{(m)} (\mathbf{v}^{(m)})^\top, \tag{4.11}$$

where the number of spiked eigenvalues, $\hat{K}$, is determined by the threshold procedure from Section 3.4. In Figure 4.3, the eigenvalues associated with the modularity component are displayed in between the random component and the global component. In the rightmost matrix of Figure 4.4, the associated spectral modularity matrix is displayed, where we recognize the presence of positive and negative elements similar to the Girvan-Newman modularity displayed in Figure 4.2. The spectral modularity objective, from now on denoted by $Q_0 : \mathcal{P} \to \mathbb{R}$, is

$$Q_0(\rho) = \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \in C_k} \mathbf{B}_{ij}, \tag{4.12}$$

where the equivalence to subtraction of a null model in the modularity framework from Equation 4.1 can be seen from that we subtract $\mathbf{S}^{(g)}$ and $\mathbf{S}^{(r)}$ from $\mathbf{S}$. The equivalence of the objective Q_0 to the Girvan-Newman objective, specified in Equation 4.1, makes it possible to maximize spectral modularity, which can be maximized with existing modularity maximization methods.

4.3.2. Spectral Modularity Vectors

Spectral modularity gives us access to a particular representation of objects $\{\mathbf{x}_i\}_{i=1}^n \subset \mathcal{X}$ in a lower dimensional subspace, the spectral modularity vectors. To be precise, we use the set $\{\mathbf{r}_i\}_{i=1}^n \subset \mathbb{R}^{\hat{K}-1}$ defined by

$$\mathbf{r}_i = \left(\sqrt{\lambda_2} v_i^{(2)} \quad \dots \quad \sqrt{\lambda_{\hat{K}}} v_i^{(\hat{K})} \right). \tag{4.13}$$

Then the spectral modularity vectors are related to the spectral modularity matrix $\mathbf{B}$, such that

$$\mathbf{B}_{ij} = \mathbf{r}_i \cdot \mathbf{r}_j, \tag{4.14}$$

for some $i, j \in \{1, \dots, n\}$. Therefore, the inner product of two objects, i, j, in this representation is especially meaningful. A positive inner product signifies a positive pairwise modularity between the two objects, and a negative product is equivalent to a negative pairwise modularity. A few instances of the spectral modularity vectors are illustrated in Figure 4.5.

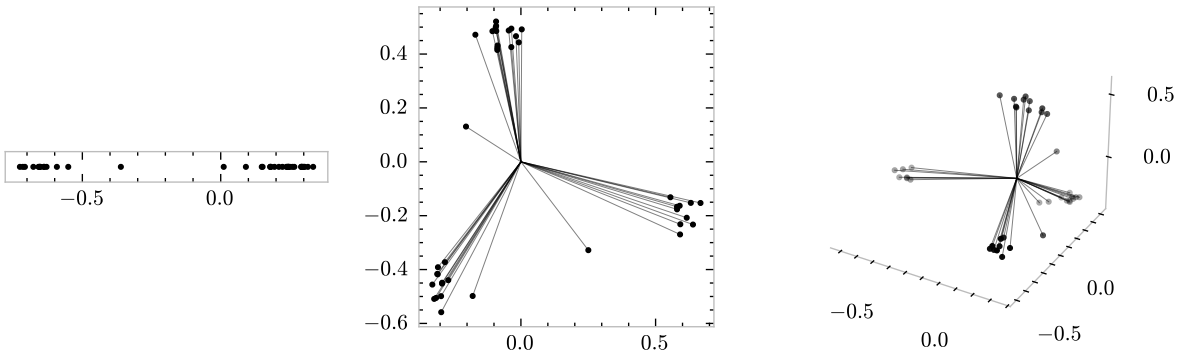


Figure 4.5: Spectral modularity vectors. The left figure displays the spectral modularity vectors of a data set with two groups represented on $\mathbb{R}$. In the middle figure, the spectral modularity vectors of a data set with 3 groups are presented in $\mathbb{R}^2$. In the rightmost figure, the spectral modularity vectors of a dataset with 4 groups are present in $\mathbb{R}^3$.

Part II

Theoretical and Methodological Developments

5

Spectral Modularity Breakdown

In this chapter, we demonstrate a fundamental challenge of spectral modularity maximization: the spectral modularity breakdown. The phenomenon occurs as the number of groups of objects that are present in a data set increases. In that setting, it appears that clusterings obtained with naive spectral modularity maximization have a bias toward constructing clusterings with fewer clusters than one would expect. In particular, naive spectral modularity maximization has a tendency to inconsistently merge clusters, whose objects are in fact significantly different.

To be precise, consider a set of objects that are clearly structured in different groups, such that the ground-truth number of groups, K, is correctly represented by the spiked eigenvalues, $\hat{K}$. This means that the K groups are well represented by the information in the eigenvectors of the similarity matrix. This ensures that the clustering methods based on these eigenvectors, including the spectral modularity method, should be capable of recovering the group structure with relatively high accuracy. However, it appears that naive spectral modularity maximization fails to do so, as it detects fewer groups.

Even if the clustering problem is easy, such that existing spectral clustering methods provide an almost exact recovery of the correct ground-truth partition, the naive spectral modularity method fails. Therefore, it is tempting to believe that there is a fundamental flaw in naive spectral modularity maximization.

It is natural to question which part of the spectral modularity maximization is responsible for the breakdown. In particular, three components can be studied: the spectral modularity matrix $\mathbf{B}$ from Equation 4.11, the spectral modularity objective Q_0 from Equation 4.12, and the maximization procedure. We start by studying the theoretical properties of the spectral modularity matrix in combination with the modularity objective function and find that the problem can be explained by these two components. First, in the computation of the spectral modularity matrix, the information representing the actual group structure becomes distorted by the number of pairs of objects that are in different groups. Second, the objective function is heavily influenced by this distortion, such that when it is maximized, inconsistent (locally) optimal partitions will be obtained. Because the combination of these two components intrinsically displays inconsistency, the maximization procedure is rendered free of any scrutiny.

In Section 5.1, we discuss the intuition behind the breakdown of naive spectral modularity maximization and provide an illustrative example. In Section 5.2, we provide a consistency condition, which becomes problematic as the number of groups grows. In Section 5.3, we see how increasing the number of groups brings us arbitrarily close to violating this condition by studying the asymptotic behavior of the spectral modularity matrix in a highly ideal setting. In Section 5.4, we see how this leads to the breakdown of spectral modularity by studying perturbations of the ideal setting.

5.1. Illustrative Example and Intuition

The intuition of spectral modularity breakdown can be demonstrated with the use of a ground-truth partition, i.e., a partition that is considered most meaningful. This ground-truth partition can be obtained by artificially constructing a data set with a group structure such that objects in the same group are significantly more similar than objects that are in different groups. For example, we can construct a data set that follows the distribution of a Gaussian Mixture Model (GMM) as discussed in Section 2.3, which is endowed with a specific partition that can be considered the ground truth.

In this setting, spectral modularity breakdown refers to the observation that naive spectral modularity maximization provides a different group structure than ground-truth partition, specifically in relation to underestimating the number of groups. On the other hand, within the context of random matrix theory, and assuming the groups are significantly different enough, one can identify the correct number of groups, K, from the number of spiked eigenvalues, $\hat{K}$, as discussed in Section 3.4, and we therefore assume $\hat{K} = K$ for the remainder of this chapter and use K to denote the quantity. This is important because, using the correct number of groups, a relatively standard spectral clustering algorithm such as the one demonstrated in Section 2.3 can recover the ground-truth partition with high accuracy.

To illustrate this, we consider a data set of n objects, i.e., $\{\mathbf{x}_i\}_{i=1}^n$, and let $\rho^* = \{C_1, \dots, C_K\}$ denote a symmetric K-partition of these n objects that satisfies

$$|C_1| = \cdots = |C_K| = M, \tag{5.1}$$

where M denotes the size of the groups. In particular, the number of objects n should be seen as a function of K with a constant M that satisfies the relation $n = mK$. This means that the number of objects in a group does not shrink when K grows, and each group remains significant as the number of groups grows. Note that the alternative of fixing n and implicitly varying $M = \frac{n}{K}$ instead trivially results in problematic clusterings, as the number of objects per group shrinks drastically. For example, if n is fixed and $K = n$, then every object is contained in its own group, which is equivalent to saying that there is no group at all. For a particular combination of n, M, and K, the ground-truth partition ρ^* can be written as

$$C_1 = \{1, \dots, m\}, C_2 = \{m+1, \dots, 2m\}, \dots C_K = \{(K-1)m + 1, \dots, Km\}. \tag{5.2}$$

Now, we assume that the distribution of the data set follows a GMM as introduced in Section 2.3. Formally, this means that for each $C_k \in \rho^*$ and for all $i \in C_k$

$$\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{I}), \tag{5.3}$$

where $\boldsymbol{\mu}_k \in \mathbb{R}^p$. In addition, the distances between the means of the Gaussian distribution, $\{\boldsymbol{\mu}_k\}_{k=1}^K$, are symmetric, such that for some $\alpha > 0$ we have

$$||\boldsymbol{\mu}_k - \boldsymbol{\mu}_h||_2 = \alpha, \text{ for all } k, h \in \{1, \dots, K\} \text{ with } k \neq h. \tag{5.4}$$

Then, the $n \times n$ similarity matrix $\mathbf{S}$ is obtained using the Euclidean distance transformed with the negative exponential as described in Equation 3.6. Furthermore, assume that $\alpha > 0$ is chosen such that there are K eigenvalues of $\mathbf{S}$ that are spikes, i.e., they are significantly larger than the eigenvalues in the bulk. This ensures that the number of groups in the ground-truth partition of the data is in the detectable regime as specified in Section 3.3.

In Figure 5.1 we see an example of the $n \times n$ similarity matrix $\mathbf{S}$ of this setting for $M = 30$, $K = 6$, $n = MK = 180$, and $\alpha = 0.15$, associated with this example in the left panel. Here, we see that there are indeed $K = 6$ brighter squares on the diagonal of the matrix, representing the higher internal similarity within the groups. In the right panel of the figure, we see the eigenvalue distribution of the similarity matrix $\mathbf{S}$ associated with this dataset. In the inset, we find that there is one large eigenvalue at approximately 160. Furthermore, there are 5 spiked eigenvalues that are somewhat closer to the bulk of the eigenvalues but still significantly spiked. Therefore, the histogram indeed shows $\hat{K} = 6$ spiked eigenvalues, which corresponds to the correct number of groups, $K = 6$, that are present in the data set as discussed in Section 3.4.

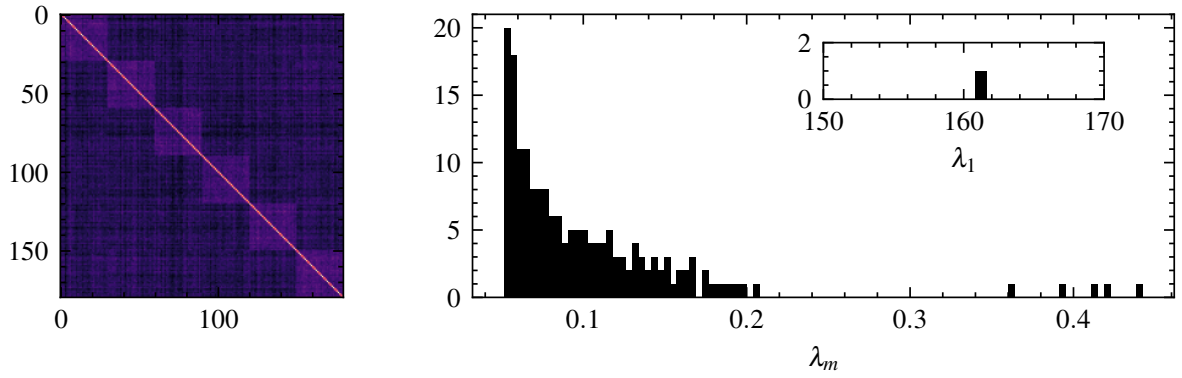


Figure 5.1: Similarity matrix and its eigenvalue distribution of Gaussian mixture model data. The data set is a sample of the GMM defined in Equation 5.3 with $\alpha = 0.15$, $p = 200$, $K = 6$, $n = 30 \cdot K$ and a symmetric ground-truth partition. The $n \times n$ similarity matrix is depicted in the left panel and is defined by Equation 3.6. Brighter colors are associated with higher values. The eigenvalues are demonstrated in the histogram in the right panel. The inset in the right panel shows the size of the largest eigenvalue.

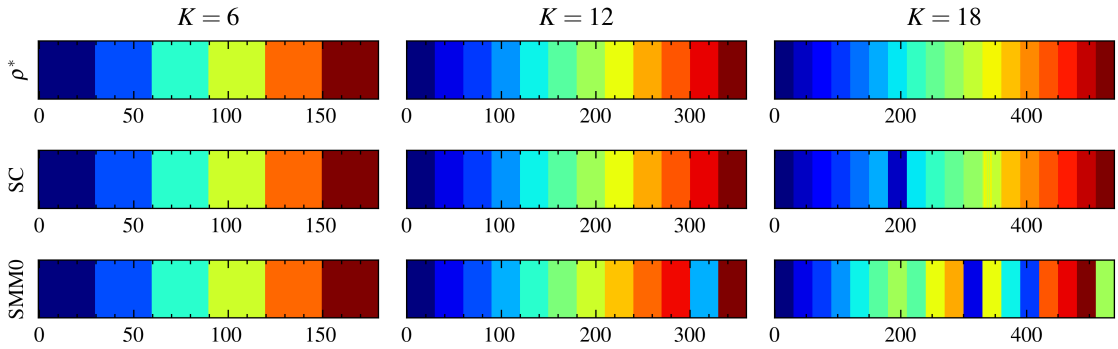


Figure 5.2: Breakdown of spectral modularity. The color bars demonstrate the clustering results of standard spectral clustering (SC) as specified in Section 2.3 and naive spectral modularity maximization (SMM0) as specified in Section 4.3 compared to the symmetric ground-truth partition ρ^* for different values of K. The data set follows the GMM model specified in Equation 5.3 with $\alpha = 0.15$ and $M = 30$. The different colors represent the different clusters.

To illustrate the breakdown, we consider this example data set for multiple values of K and perform clustering of the data set with two methods. First, we use the standard spectral clustering (SC) as described in Section 2.3. Second, we use the naive spectral modularity maximization (SMM0) as described in Section 4.3. In Figure 5.2, the clusterings of the example datasets, i.e., for $K = 6, K = 12, K = 18$, are demonstrated. There are three columns, where each contains the clustering of a synthetic data set for a given number of clusters, K. The first row represents the ground-truth ρ^*, the second row represents the clustering obtained with SC, and the third row represents the clustering obtained with SMM0. The clusters are indicated by objects having the same color. The horizontal position of each bar represents the objects. Here, we see that when $K = 6$, both methods recover the exact ground-truth partition, indicated by an exactly similar color order. When $K = 12$, we see that SMM0 merges two clusters (the clusters around 100 and 300, indicated by the same light blue color), while SC has again an exact recovery of the ground-truth partition. When $K = 18$, SMM0 merges more than a few clusters, and the ground-truth partition is hardly recovered. At the same time, SC almost exactly recovers the entire partition.

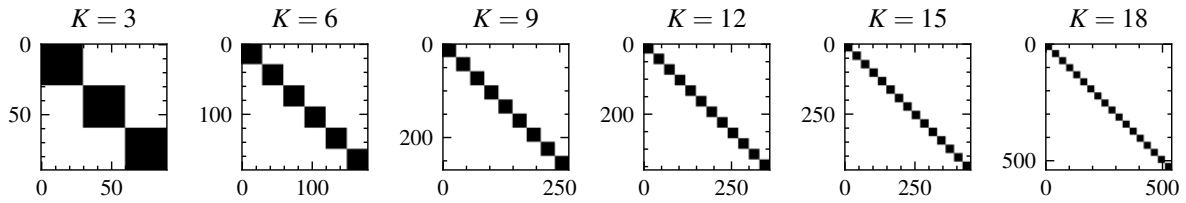


Figure 5.3: Relative importance of internal and external pairs. The $n \times n$ matrices indicate the group structure of a symmetric partition with $n = 30 \cdot K$. The black colors indicate two objects belonging to the same group. The white colors indicate two objects belonging to different groups.

The mechanism behind the inconsistency of spectral modularity has a link to a relatively intuitive combinatorial aspect. To be precise, the inconsistency can be explained by the growth differences in the number of internal pairs of objects within groups and external pairs of objects between groups. Mathematically, this phenomenon can be described as the ratio between the size of a set, which represents the number of internal pairs and grows with order K, and the Cartesian product of the set, which represents all external pairs and grows with order K^2. Because the size of the Cartesian product grows much faster, the set of information in the similarity matrix is dominated by the information of the similarities of external pairs of objects for high K.

This is illustrated in Figure 5.3, where we see the structural matrices representing the group structure that follows the symmetric partition with $M = 30$. The black color in the figures indicates that the objects are in the same group, while the white color indicates that the objects are not in the same group. From these figures, we see that as K grows, the proportion of white entries in the figures becomes increasingly bigger and essentially vanishes as $K \to \infty$.

The number of pairs of objects in the same group can be expressed as

$$\#\text{ internal pairs of objects } = K\frac{M(M-1)}{2}, \tag{5.5}$$

while the number of pairs of objects that are in different groups can be expressed as

$$\#\text{ external pairs of objects } = M^2\frac{K(K-1)}{2}, \tag{5.6}$$

so the ratio between the two quantities reads:

$$\frac{\#\text{ internal pairs of objects}}{\#\text{ external pairs of objects}} = \frac{M-1}{M}\frac{1}{K-1}, \tag{5.7}$$

which tends to zero for large K. This confirms the intuition obtained in Figure 5.3 and suggests that the information in the similarity matrix is dominated by the external object pairs.

This saturation of the similarity matrix becomes problematic, as the external between-group modularity should be smaller or equal to zero in order for the ground-truth partition ρ^* to be consistent. If the sum of pairwise modularities of objects two in two different groups is positive, a merge of the two groups of the ground-truth partition is favored over the ground-truth partition, resulting in an inconsistency of the model and modularity objective. However, the consistency of a ground-truth partition becomes increasingly difficult to realize as K grows because of how the relative contribution of external object pairs grows. This leads to a conceptual misalignment between, on the one hand, the theoretically correct ground-truth partition and, on the other hand, the spectral modularity based optimum.

5.2. Ground-Truth Consistency

The spectral modularity breakdown can be studied by comparing a non-ambiguously correct ground-truth partition that is obtained from a simplistic model for the data and an optimal partition that is obtained by maximization of the spectral modularity objective. This way, the consistency of a particular ground-truth partition ρ^* can be expressed in terms of the global optimality of the partition. In order for the ground-truth partition ρ^* to be consistent with Q_0, no other partition should have a higher value in the spectral modularity objective, which means that we require

$$Q_0(\rho) \leq Q_0(\rho^*) \quad \text{for all } \rho \in \mathcal{P}. \tag{5.8}$$

Then, using this definition, we write the following equivalent statement about the consistency of the ground-truth partition ρ^* with the spectral modularity objective Q_0:

$$\rho^* \text{ is consistent with } Q_0 \quad \iff \quad \rho^* \in \mathcal{O} := \{\hat{\rho} \in \mathcal{P} : Q_0(\hat{\rho}) = \max_{\rho \in \mathcal{P}} Q_0(\rho)\}, \tag{5.9}$$

where the set $\mathcal{O}$ denotes the set of all partitions that attain the maximal value of Q_0, thereby denoting the set of optimal solutions, which may be more than one solution. The expression on the left-hand side of the equation is shortly referred to as ρ^* being Q_0-consistent.

If there is a partition almost identical to ρ^* but with two of the groups merged that are favored by Q_0, then ρ^* is not optimal. Therefore, we can derive a necessary condition for the Q_0-consistency of ρ^* that ensures no such partition is favored. In particular, for a given ground-truth partition ρ^*, we use a $K \times K$ matrix $\mathbf{C}$, which we refer to as the group affinity matrix and is defined by

$$\mathbf{G}_{kh} = \sum_{i \in C_k} \sum_{j \in C_h} \mathbf{B}_{ij} \text{ for all } k, h \in \{1, \ldots, K\}, \tag{5.10}$$

where $\rho^* = \{C_1, \ldots, C_K\} \in \mathcal{P}$ and $\mathbf{B}$ is some spectral modularity matrix, of which we make the details precise in the next sections. Note that the spectral modularity objective function Q_0 can be concisely written in terms of the group affinity matrix $\mathbf{G}$, i.e.

$$Q_0(\rho^*) = \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \in C_k} \mathbf{B}_{ij} = \sum_{k=1}^{K} \mathbf{G}_{kk} = \mathrm{Tr}[\mathbf{G}]. \tag{5.11}$$

The goal of the group affinity matrix is to demonstrate a condition based on the merging of two groups of ρ^* that is favored by Q_0, ensuring that $\rho^* \notin \mathcal{O}$ and therefore making ρ^* inconsistent. In essence, global optimality requires that merging any two groups in a partition ρ does not lead to a higher value in the modularity objective. Specifically, the condition is shortly referred to by A and defined by

$$\mathrm{A} \quad \iff \quad \mathbf{G}_{kh} \leq 0 \text{ for all } k, h \in \{1, \ldots, K\} \text{ with } k \neq h. \tag{5.12}$$

To see that A is indeed a necessary condition, consider that one of the inequalities does not hold, i.e., if $\mathbf{G}_{kh} > 0$ for some $k, h \in \{1, \ldots, K\}$ and $k \neq h$, then ρ^* is not a global maximum, as we can improve the objective Q_0 by merging the two groups C_k and C_h, i.e., for ρ' identical to ρ^* with the two groups merged

$$Q(\rho') = Q(\rho^*) + \mathbf{G}_{kh} > Q(\rho^*). \tag{5.13}$$

Therefore, if A is broken, the ground-truth partition ρ^* is no longer among the set of optimal solutions, i.e., $\rho^* \notin \mathcal{O}$, and is therefore inconsistent with Q_0. We can formally denote this by

$$\neg\mathrm{A} \iff \mathbf{G}_{kh} > 0 \text{ for some } k, h \in \{1, \ldots, K\} \text{ with } k \neq h \implies \rho^* \text{ is not } Q_0\text{-consistent.} \tag{5.14}$$

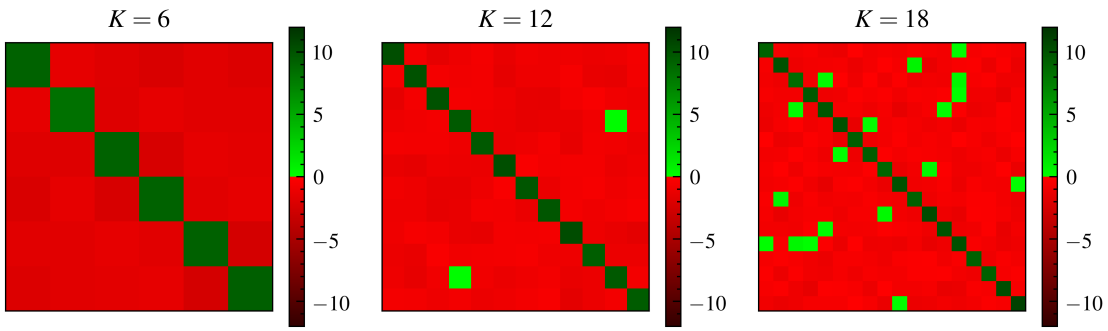


Figure 5.4: Group affinity matrix The data set is identical to the setting of 5.1. Green represents positive values. Red represents negative values. In the left figure, the diagonal elements of $\mathbf{G}$ are positive and the off-diagonal elements are negative; therefore, condition C is met. On the other hand, in the other figures, there are positive off-diagonal elements, suggesting a breaking of the condition and making the ground-truth partition inconsistent with Q_0.

In Figure 5.4, we demonstrate the breaking of the condition by showing the group affinity matrix $\mathbf{G}$. In the leftmost figure, with $K = 6$, we see that all the diagonal elements of the matrix are positive and the off-diagonal elements are negative. In the middle figure, where $K = 12$, we see that a single off-diagonal element is positive, even though it is still relatively close to 0. A modularity maximization procedure would merge the two clusters that have a positive off-diagonal. This demonstrates an inconsistency, as according to the ground-truth partition ρ^*, the clusters should not be merged. In the rightmost figure, where $K = 18$, we see a more extreme violation of A, where many of the off-diagonal elements of $\mathbf{G}$ are positive.

5.3. Idealized Asymptotic Behavior of Spectral Modularity

To describe the asymptotic behavior of spectral modularity, we use a model of ideal similarity matrices that is free of randomness. In particular, we demonstrate that the off-diagonal elements of the group affinity matrix $\mathbf{G}$ are close to 0 for ground-truth partitions with many groups. To achieve this, we derive an explicit expression for the spectral modularity matrix in the ideal setting, where we make use of piece-wise constant eigenvectors that are due to the homogeneity and symmetry of the model. Using the spectral modularity matrix, we then show that the off-diagonal elements of the group affinity matrix associated with a correct ground-truth partition converge to zero, i.e.,

$$\mathbf{G}_{kh} \uparrow 0 \text{ for some } k, h \in \{1, \ldots, K\} \text{ with } k \neq h. \tag{5.15}$$

Therefore, increasing the number of groups brings the ground-truth partition arbitrarily close to violating the condition A defined in Equation 5.12.

5.3.1. Toy Model A

The specific model of ideal similarity matrices is referred to as Toy Model A (TM-A), as it describes a rather simplistic view of similarity matrices without an underlying data matrix. Specifically, TM-A models a similarity matrix of a data set with K equal-sized clusters. Furthermore, the clusters are completely symmetric and homogeneous. This means that the similarity of two objects in any of the clusters takes a constant value, $a \in [0, 1]$, and the similarity of two objects in any two different clusters takes a constant value, $b \in [0, 1]$, with $b < a$.

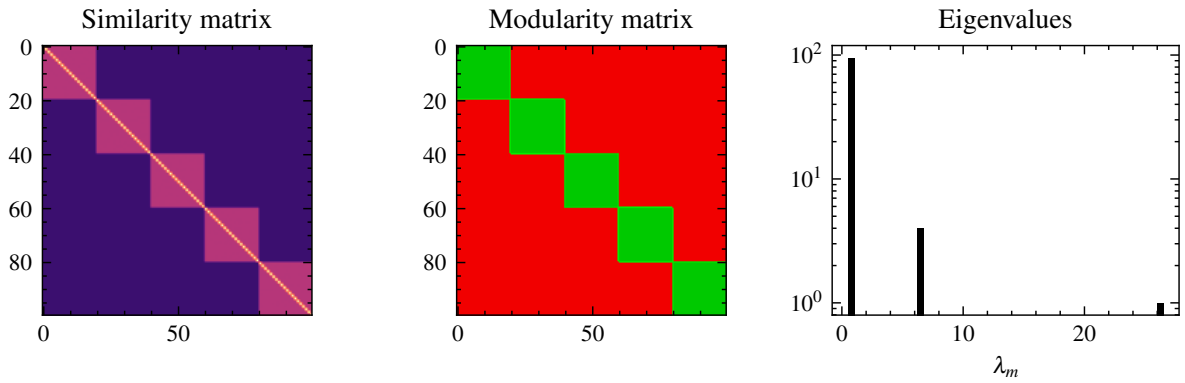


Figure 5.5: Illustration of toy model A. The leftmost figure shows the similarity matrix of TM-A with $K = 5$. The diagonal elements take value 1, the diagonal blocks take values a and the remaining elements of the matrix take value b. In the middle figure, we see the spectral modularity matrix of TM-A. The red color represents negative values, and the green color represents positive values. The rightmost figure displays a histogram of the eigenvalues of the similarity matrix. The vertical axis is logarithmically scaled and represents the multiplicity of the eigenvalue, while the vertical axis represents the value of λ_m. From this, we see that $\lambda_1 \approx 23$ is the highest, and $\lambda_2 = \cdots = \lambda_5 \approx 8$, while the remaining eigenvalues are almost zero.

Specifically, a similarity matrix $\mathbf{S}$ is defined according to TM-A if the similarity between objects i and j with $i, j \in \{1, \ldots, n\}$ is given by

$$\mathbf{S}_{ij} = \begin{cases} 1 & \text{if } i = j, \\ a & \text{if } i, j \in C_k \text{ for some } k \in \{1, \ldots, K\}, \\ b & \text{if } i \in C_h \text{ and } j \in C_k \text{ for some } k, h \in \{1, \ldots, K\} \text{ with } k \neq h, \end{cases} \tag{5.16}$$

for some $a, b \in (0, 1)$ with $a > b$ and symmetric K-partition $\rho^* = \{C_1, \ldots, C_K\}$. Furthermore, the ground-truth partition in the toy model is unambiguously represented by the pairs of objects that have higher similarity.

The model can be seen as an extreme version of the symmetric Gaussian mixture model data described in Section 5.1. This is because the Euclidean distances between Gaussian random variables are known to concentrate at a fixed value, as is discussed in Section 2.4. Therefore, as the dimensionality p grows, the similarity matrix will converge to that described by TM-A, which highlights the importance of TM-A for our asymptotic understanding. Figure 5.5 shows an instance of TM-A, along with the associated spectral modularity matrix and eigenvalue spectrum. We denote the eigenvalues of $\mathbf{S}$ by $\{\lambda_m\}_{m=1}^n$ and the eigenvectors by $\{\mathbf{v}^{(m)}\}_{m=1}^n$, with $v_i^{(m)}$ denoting the ith entry of the mth eigenvector.

Positive Definiteness

For consistency with the rest of this thesis, matrices $\mathbf{S}$ obtained from TM-A need to be positive semi-definite, which can be shown through an equivalence to Hamming similarity matrices that are guaranteed to be positive definite, as discussed in Section 3.1. To be precise, for a $n \times p$ matrix $\mathbf{X}$, the entries satisfy the following definition:

$$x_i^{(l)} = \begin{cases} 0 & \text{for } l \in \{1, \ldots, \lfloor p \cdot b \rfloor\}, \\ k+1 & \text{for } l \in \{\lceil p \cdot b \rceil + 1, \ldots, \lfloor p \cdot a \rfloor\}, \\ i & \text{for } l \in \{\lceil p \cdot a \rceil, \ldots, p\}, \end{cases} \tag{5.17}$$

for all $i \in \{1, \ldots, n\}$ and $l \in \{1, \ldots p\}$. The Hamming similarity matrix of $\mathbf{X}$ has the same similarity matrix as that obtained from TM-A, up to rounding errors that become negligible for high p. The justification behind the equivalence is that all objects have the same value for the first $\lfloor p \cdot (a - b) \rfloor$ features, the objects in the same group share the same values for the next approximately $\lfloor p \cdot (a - b) \rfloor$ features, and the remaining features are all different, which leads to internal similarities a and external similarities b.

Asymptotic Group Affinity Matrix

Accordingly, the group affinity matrix, as defined in Equation 5.10 of the ground-truth partition ρ^*, can be written with the spectral decomposition of the similarity matrix $\mathbf{S}$, where we denote $\{\lambda_m\}_{m=1}^n$s as the eigenvalues of $\mathbf{S}$ and $\{\mathbf{v}^{(m)}\}_{m=1}^n$ where $v_i^{(m)}$ denotes the ith entry of the mth eigenvector. Specifically, we find an expression for the off-diagonal elements, i.e., $k \neq h$, and $\mathbf{G}$ has

$$\mathbf{G}_{kh} \propto -\frac{1}{K}, \tag{5.18}$$

which will converge to zero from below. The derivation of this expression is based on the equivalence to the K largest eigenvectors of a block matrix, i.e.,

$$\mathbf{S}^\circ = (\mathbf{S} - (1-a)\mathbf{I}), \tag{5.19}$$

which is identical to $\mathbf{S}$ except with the diagonal set to a. Then, if $\mathbf{v}$ is an arbitrary eigenvector of $\mathbf{S}^\circ$ associated with the eigenvalue λ°, then

$$\mathbf{S}\mathbf{v} = (\mathbf{S}^\circ + (1-a)\mathbf{I})\mathbf{v} = \mathbf{S}^\circ\mathbf{v} + (1-a)\mathbf{v} = (\lambda^\circ + (1-a))\mathbf{v} = \lambda\mathbf{v}, \tag{5.20}$$

which shows that the eigenvalues are related through $\lambda_m = \lambda_m^\circ + (1-a)$.

5.3.2. Eigenvectors of Block Diagonal Matrix $\mathbf{S}^\circ$

We make use of the fact that the matrix $\mathbf{S}^\circ$ can be conveniently written as a Kronecker product, denoted by $\otimes$, of two simple symmetric positive definite matrices, i.e.

$$\mathbf{S}^\circ = \mathbf{M}^\circ \otimes \mathbf{J}^\circ, \quad \text{with} \quad \mathbf{M}^\circ := b\mathbf{J}_{K\times K} + (a-b)\mathbf{I}_{K\times K} \quad \text{and} \quad \mathbf{J}^\circ := \mathbf{J}_{M\times M}, \tag{5.21}$$

where $\mathbf{M}^\circ$ is a $K \times K$ matrix of b's and a's on the diagonal, and $\mathbf{J}^\circ$ is a $M \times M$ matrix of 1's. The only non-zero eigenvalue of $\mathbf{J}^\circ$ is M and is associated to a constant eigenvector $\mathbf{q}^{(1)} = \mathbf{1}\frac{1}{\sqrt{M}} \in \mathbb{R}^M$, while the eigenvalues of $\mathbf{M}^\circ$ are

$$\mu_1^\circ = b(K-1) + a \text{ and } \mu_2^\circ = \cdots = \mu_K^\circ = a - b. \tag{5.22}$$

The largest eigenvalue μ_1° is associated with satisfying the constant eigenvector $\mathbf{u}^{(1)} = \mathbf{1}\frac{1}{\sqrt{K}} \in \mathbb{R}^M$, i.e.,

$$\mathbf{M}^\circ\mathbf{u}^{(1)} = b\mathbf{J}_{K\times K}\mathbf{u}^{(1)} + (a-b)\mathbf{I}_{K\times K}\mathbf{u}^{(1)} = \mu_1^\circ\mathbf{u}^{(1)}. \tag{5.23}$$

Furthermore, the $K-1$ eigenvectors associated with the eigenvalue $a-b$ are the solutions for $\mathbf{u} \in \mathbb{R}^K$ in the following equation:

$$(\mathbf{M}^\circ - (a-b)\mathbf{I}_{K\times K})\,\mathbf{u} = b\mathbf{J}_{K\times K}\mathbf{u} = \mathbf{0}. \tag{5.24}$$

from which the multiplicity of the eigenvalue can be seen by the number of solutions. Because the matrix in the left-hand side of the equation is equal to $\mathbf{J}_{K\times K}$, which is a rank 1 matrix, we have by the rank-nullity theorem that there are $K-1$ orthonormal solutions to the above problem, which leads us to a $K-1$ multiplicity of the eigenvalue $a-b$.

Then, because of the positivity of the eigenvalues, $\mathbf{M}^\circ$ and $\mathbf{J}^\circ$ satisfy the positive (semi-)definiteness that is required for the spectral decompositions. This is convenient because the eigenvectors of a Kronecker product of two symmetric positive definite matrices can be written as the Kronecker product of the eigenvectors (Chapter 2, [84]), i.e.,

$$\mathbf{S}^\circ = (\underbrace{\mathbf{U}\mathbf{\Lambda}_1\mathbf{U}^\top}_{\mathbf{M}^\circ}) \otimes (\underbrace{\mathbf{Q}\mathbf{\Lambda}_2\mathbf{Q}^\top}_{\mathbf{J}^\circ}) = (\mathbf{U}\otimes\mathbf{Q})(\mathbf{\Lambda}_1\otimes\mathbf{\Lambda}_2)(\mathbf{U}\otimes\mathbf{Q})^\top, \tag{5.25}$$

where we use the mixed-product property of the Kronecker product. Combining the above, we can write the eigenvalues $\lambda_m^\circ = M\mu_m^\circ$ and the entries of the largest K eigenvectors of $\mathbf{S}^\circ$, and therefore also of $\mathbf{S}$, as

$$v_i^{(m)} = \frac{1}{\sqrt{M}}u_{k_i}^{(m)} \text{ for all } m \in \{1,\ldots,K\}, \tag{5.26}$$

where k_i refers to the group of object i, i.e., $i \in C_{k_i}$.

5.3.3. Off-Diagonal Elements of G

Using the notation for λ_m, we can write the off-diagonal elements of the group affinity matrix as

$$\mathbf{G}_{kh} = \sum_{i \in C_k} \sum_{j \in C_h} \sum_{m=2}^{K} \underbrace{(\lambda_m^\circ + (1-a))}_{\lambda_m} v_i^{(m)} v_j^{(m)}, \tag{5.27}$$

for some $k, h \in \{1, \dots, K\}$ with $k \neq h$. The right-hand side of the equation can be decomposed into two separate sums, i.e.,

$$\mathbf{G}_{kh} = \underbrace{\sum_{i \in C_k} \sum_{j \in C_h} \sum_{m=2}^{K} \lambda_m^\circ v_i^{(m)} v_j^{(m)}}_{*} + \underbrace{\sum_{i \in C_k} \sum_{j \in C_h} \sum_{m=2}^{K} (1-a) v_i^{(m)} v_j^{(m)}}_{**}. \tag{5.28}$$

For the first sum, we use that there are only K non-zero eigenvalues of $\mathbf{S}^\circ$, i.e.,$\lambda_m^\circ = 0$ for $m > K$, and that the K associated eigenvectors of are identical to those associated with $\mathbf{S}$. This way, we recognize the spectral decomposition of $\mathbf{S}^\circ$, which can be used to write

$$* = \sum_{i \in C_k} \sum_{j \in C_h} \left(\mathbf{S}_{ij}^\circ - \lambda_1^\circ v_i^{(1)} v_j^{(1)} \right) = M^2 \left(\underbrace{b}_{\mathbf{S}_{ij}^\circ} - M \left(\underbrace{b(K-1)+a}_{\mu_1^\circ} \right) \frac{1}{M} \frac{1}{K} \right) = -\frac{M^2(a-b)}{K}. \tag{5.29}$$

For the second term, we recognize the spectral decomposition of $\mathbf{M}^\circ$ using the eigenvectors denoted in Equation 5.26, which allows us to write

$$** = \sum_{i \in C_k} \sum_{j \in C_h} \left(\frac{1-a}{a-b} \sum_{m=2}^{K} \underbrace{\mu_m^\circ}_{a-b \text{ for } m>1} \underbrace{\frac{1}{\sqrt{M}} u_k^{(m)}}_{v_i^{(m)}} \underbrace{\frac{1}{\sqrt{M}} u_k^{(m)}}_{v_j^{(m)}} \right) = M^2 \left(\frac{1-a}{a-b} \frac{1}{M} \left(\mathbf{M}_{kh}^\circ - \mu_1^\circ \frac{1}{K} \right) \right) \tag{5.30}$$

$$= M \frac{1-a}{a-b} \left(\underbrace{b}_{\mathbf{M}_{kh}^\circ} - \left(\underbrace{b(K-1)+a}_{\mu_1^\circ} \right) \frac{1}{K} \right) = -\frac{M(1-a)}{K}. \tag{5.31}$$

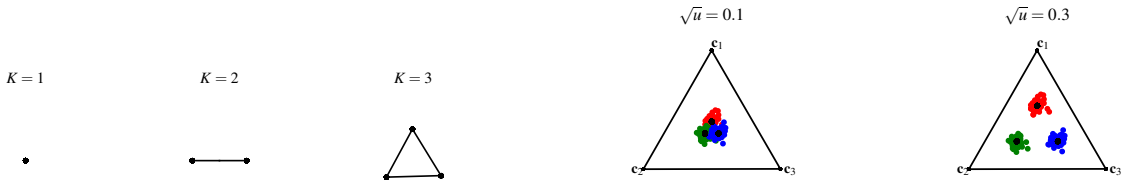
Then, combining the two, we have

$$\mathbf{G}_{kh} = * + ** = -\frac{M^2(a-b)}{K} - \frac{M(1-a)}{K} = -(M^2 \underbrace{(a-b)}_{>0} + M \underbrace{(1-a)}_{>0}). \tag{5.32}$$

Since, by definition of TM-A, we have $b < a < 1$ and clearly $M > 0$, the factor before $\frac{1}{K}$ is always positive. Therefore, from the above, it follows that the off-diagonal elements of the group affinity matrix in this idealized setting are proportional to $-\frac{1}{K}$ and converge to zero from below, i.e.,

$$\mathbf{G}_{kh} \propto -\frac{1}{K} \implies \mathbf{G}_{kh} \uparrow 0 \text{ as } K \to \infty. \tag{5.33}$$

This makes the necessary condition A increasingly difficult to satisfy for perturbations away from this ideal setting, which we will study in the next section.



(a) Regular $K-1$-simplex. **(b) Illustration of toy model B.**

Figure 5.6: (a) $K = 1$ associates to a 0-simplex, which is a dot. $K = 2$ associates to a 1-simplex, which is a line piece. $K = 3$ associates to a regular 2-simplex, which is a regular triangle. (b) The triangle represents a 2-simplex. The black dots represent the scaled corners of the simplex. The colored dots represent the perturbed spectral modularity vectors. The perturbations $\{\mathbf{z}_i\}_{i=1}^n$ are i.i.d. zero mean Gaussian with variance 0.1. The left panel shows the spectral modularity vectors of TM-B for $\sqrt{u} = 0.1$, and the right panel shows the spectral modularity vectors of TM-B for $\sqrt{u} = 0.3$.

5.4. Perturbing Spectral Modularity Vectors

Given that the partition affinity matrix in TM-A becomes arbitrarily close to violating the condition A, it is tempting to think that only slight deviations away from the idealized similarity matrix will break. Unfortunately, such a perturbation analysis is difficult to perform in the context of TM-A, as it would require element-wise expressions for the eigenvectors of the perturbed matrices. To circumvent this mathematical complexity, we can approach the problem from a different perspective by directly providing a model for the spectral modularity matrix and the associated spectral modularity vectors described in Section 4.3.

5.4.1. Toy Model B

Instead of considering a model for the $n \times n$ similarity matrices as is done in TM-A, we now consider a set of n vectors in $K-1$ dimensions that resemble the idealized spectral modularity vectors, which we can then perturb directly. Using these vectors, the entries of the spectral modularity matrix can be obtained by taking the inner product of the associated spectral modularity vectors.

Again, we consider an ideal setting of a symmetric ground-truth partition, where the group structure is homogeneous and symmetric, akin to the structure specified in TM-A. If we consider such an ideal setting where all groups are equally separated, are equally sized, and have equal densities, the only reasonable candidate structure for the spectral modularity vectors in TM-B is the regular $K-1$ simplex, which is visualized in Figure 5.6a. This can be seen by the fact that the maximum number of vectors in $\mathbb{R}^{K-1}$ that have pairwise negative dot products is K [85]. Furthermore, in order for the model to represent a completely symmetrical setting, the spectral modularity vectors, described in Section 4, of objects of the same group must all be identical, and the angles between any two objects from two different groups must be identical. Therefore, a regular $K-1$ simplex conceptually aligns with the requirement of the vectors specified in TM-B, as it contains the maximum number of points attainable in a $K-1$ dimensional real space with negative pairwise dot products that are also equally spaced apart. To be precise, consider $K > 0$ and let $\{\mathbf{c}_k\}_{k=1}^K \in \mathbb{R}_{K-1}$ be the corners of a regular $(K-1)$-simplex, then for all $k, h \in \{1, \dots, K\}$ with $k \neq h$,

$$\mathbf{c}_k \cdot \mathbf{c}_h = -\frac{1}{K-1}, \tag{5.34}$$

where we normalize the corners for convenience, such that $||\mathbf{c}_k||_2^2 = 1$ for all $k \in \{1, \dots, K\}$. This makes the dot product between $\mathbf{c}_k$ and $\mathbf{c}_h$ identical to the cosine of the angle between the vectors, which equivalently converges to $\frac{\pi}{2}$ as $K \to \infty$.

From this description of the regular $K-1$-simplex alone, we already recognize similar asymptotic behavior as that of TM-A and the combinatorial relationship described in Section 5.1. In particular, in model TM-B, the objects that are associated with one corner of the simplex belong to one group. Then, the spectral modularity of two objects from different groups is of order $-\frac{1}{K-1}$. This means that, similar to what we saw in the spectral modularity derivation of TM-A, the external modularity is negative but grows to zero from below as the number of groups, K, grows.

Using the corners of this regular $K-1$-simplex, we define the following model for spectral modularity vectors of a ground-truth symmetric K-partition:

$$\mathbf{r}_i = \sqrt{u}\mathbf{c}_k \text{ for all } i \in C_k, \text{ for all } k \in \{1, \dots, K\}, \tag{5.35}$$

where $\rho^* = \{C_1, \dots, C_K\}$ is the ground-truth partition, $u \in (0, 1)$, and $\{\mathbf{c}_k\}_{k=1}^K$ are the normalized corners of $K-1$ simplex. This notation allows us to write the elements of spectral modularity as

$$\mathbf{B}_{ij} = \mathbf{r}_i \cdot \mathbf{r}_j = \begin{cases} u & \text{if } i, j \in C_k \text{ for some } k \in \{1, \dots, K\}, \\ -\frac{u}{1-K} & \text{if } i \in C_h \text{ and } j \in C_k \text{ for some } k, h \in \{1, \dots, K\} \text{ with } k \neq h. \end{cases} \tag{5.36}$$

5.4.2. Ground-truth Violating Perturbations

Consider the perturbation of the ideal spectral modularity vectors with random vectors $\{\mathbf{z}_i\}_{i=1}^n \subset \mathbb{R}^{K-1}$ with independent and identically distributed entries that have a mean vector $\mathbf{0}$ and a constant variance for each of the $K-1$. The perturbed spectral modularity vectors are denoted by $\tilde{\mathbf{r}}_i := \mathbf{r}_i + \mathbf{z}_i$, of which a few instances are visualized in Figure 5.6b. Then, the perturbed spectral modularity matrix can be written as

$$\tilde{\mathbf{B}}_{ij} = \tilde{\mathbf{r}}_i \cdot \tilde{\mathbf{r}}_j = \underbrace{-\frac{u}{K-1}}_{\mathbf{B}_{ij}} + \mathbf{r}_i^\top \mathbf{z}_j + \mathbf{r}_j^\top \mathbf{z}_i + \mathbf{z}_i^\top \mathbf{z}_j, \tag{5.37}$$

for some $i \in C_k$ and $j \in C_h$ with $k, h \in \{1, \dots, K\}$ and $k \neq h$. Then, we can also obtain a definition for the perturbed group affinity matrix:

$$\tilde{\mathbf{G}}_{kh} := \sum_{i \in C_k} \sum_{j \in C_h} \tilde{\mathbf{B}}_{ij} = \underbrace{-\frac{uM^2}{K-1}}_{\mathbf{G}_{kh}} + \sum_{i \in C_k, j \in C_h} \left(\mathbf{r}_i^\top \mathbf{z}_j + \mathbf{r}_j^\top \mathbf{z}_i + \mathbf{z}_i^\top \mathbf{z}_j \right). \tag{5.38}$$

Because the goal of this chapter is to show that the spectral modularity breaks down as K grows, we are merely interested in showing that the necessary Q_0-consistency condition A, defined in 5.12, is broken as K grows. In order to reason about the condition probabilistically, we use the following event in the sample space of the perturbed group affinity matrix $\tilde{\mathbf{G}}$, i.e., the event that condition A holds for the perturbed group affinity matrix:

$$\mathcal{C} := \{\tilde{\mathbf{G}}_{kh} < 0 \text{ for all } k, h \in \{1, \dots, K\} \text{ with } k \neq h\}. \tag{5.39}$$

Then, for a given probability measure $\mathbb{P}$ that is defined for the perturbations of TM-B, $\mathbb{P}(\mathcal{C})$ specifies the probability that the condition A holds for the perturbed group affinity matrix $\tilde{\mathbf{G}}$. The essence of the spectral modularity breakdown is that this probability goes to zero relatively quickly as K grows.

In order to show this, we first consider that the off-diagonal elements in $\tilde{\mathbf{G}}$ are not mutually independent. This can be seen as the elements $\tilde{\mathbf{G}}_{12}, \tilde{\mathbf{G}}_{13}$, and $\tilde{\mathbf{G}}_{23}$ are not independent as they depend on the same perturbations associated with the objects in C_1, C_2, and C_3. Therefore, factoring out probabilities in terms of perturbation distributions is relatively difficult. To circumvent this mathematical difficulty of the matrix elements of $\tilde{\mathbf{G}}$, we use the fact that the set of distinct pairs is independent. To be precise, pick (approximately) $K/2$ pairs from the independent $K(K-1)/2$ pairs of groups by considering the pairs.

$$\{(1,2), (3,4), (5,6), ..., (K-1, K)\} \text{ for even } K, \tag{5.40}$$

and

$$\{(1,2), (3,4), (5,6), ..., (K-2, K-1)\} \text{ for odd } K. \tag{5.41}$$

For the remainder of this section, we assume that K is even, and therefore there are exactly $K/2$ such pairs. Then, in order to satisfy Q_0-consistency for the selected pairs, the condition must be satisfied at the least; therefore, we can use this quantity to provide an upper bound to the probability of $\mathcal{C}$, as the event of satisfying the condition for the selected (independent) pairs is contained in the event of satisfying the condition for all pairs.

We denote this event with $\mathcal{C}'$, i.e.,

$$\mathcal{C}' := \{\tilde{\mathbf{G}}_{k,k+1} < 0 \text{ for all } k \in \{1, 3, 5, \ldots, K-1\}\}. \tag{5.42}$$

Because $\mathcal{C}' \subseteq \mathcal{C}$, the probability of $\mathcal{C}$ is bounded by the probability of $\mathcal{C}$', i.e., $\mathbb{P}(\mathcal{C}) \leq \mathbb{P}(\mathcal{C}')$. Moreover, because of the independence of the pairs $(k, k+1)$ with $k \in \{1, 3, \ldots, K-1\}$ specified in the description of $\mathcal{C}'$, we can factor out the probabilities, i.e.,

$$\mathbb{P}(\mathcal{C}') = \prod_{k \in \{1,3,5,\ldots,K-1\}} \mathbb{P}(\tilde{\mathbf{G}}_{k,k+1} < 0). \tag{5.43}$$

Because of the symmetry in the model TM-B and the perturbations, the elements of $\mathbf{G}$ are identically distributed. Therefore, we can replace the probability $\mathbb{P}(\tilde{\mathbf{G}}_{k,k+1} < 0)$ with $\mathbb{P}(\tilde{\mathbf{G}}_{1,2} < 0)$ without loss of generality. Then, we obtain a relatively simple bound for the probability of the event $\mathcal{C}$ in terms of the probability of

$$\mathbb{P}(\mathcal{C}) \leq \left(\mathbb{P}(\{\tilde{\mathbf{G}}_{12} < 0\})\right)^{K/2}. \tag{5.44}$$

Now for a given K, consider a real random variable z_K that represents the sum of the elements described in Equation 5.45 for all $i \in C_1$ and $j \in C_2$, and consequently for the other pairs of groups $(k, k+1)$, i.e.,

$$z_K = \sum_{i \in C_1, j \in C_2} \mathbf{r}_i^\top \mathbf{z}_j + \mathbf{r}_j^\top \mathbf{z}_i + \mathbf{z}_i^\top \mathbf{z}_j. \tag{5.45}$$

Then, the probability $\mathbb{P}(\{\tilde{\mathbf{G}}_{12} < 0\})$ can be expressed as

$$\mathbb{P}\left(\left\{\underbrace{-\frac{uM^2}{K-1} + z_K}_{\tilde{\mathbf{G}}_{12}} < 0\right\}\right) = \mathbb{P}\left(\left\{z_K < \underbrace{\frac{uM^2}{K-1}}_{\omega_K}\right\}\right) = F_{z_K}(\omega_K). \tag{5.46}$$

Here, F_{z_K} is the cumulative distribution function (CDF) for the random variable z_K, and we use the notation of ω_K for brevity. Therefore, an upper bound for the probability of satisfying the condition can be expressed in terms of the distribution of the random variable z_K, i.e.

$$\mathbb{P}(\mathcal{C}) \leq \left(F_{z_K}(\omega_K)\right)^{K/2}. \tag{5.47}$$

5.4.3. Distribution of z_K

Unfortunately, given an arbitrary perturbation distribution for $\{\mathbf{z}_i\}_{i=1}^n$, the distribution function F_{z_K} is difficult to determine explicitly because of the dependencies in the sum. Instead, we shed some light on the mild requirements that the distribution F_{z_K} should satisfy in order for the spectral modularity breakdown to occur.

If we assume that the perturbations $\{\mathbf{z}_i\}_{i=1}^n$ are distributed such that the variance of z_K remains fixed, then it is tempting to think that the condition A is broken as K grows. Specifically, because the term ω_K approaches zero, there exists a value of K' where $F_{z_K}(\omega_K)$ will be smaller than 1 for all $K > K'$, which makes the exponent in the bound of Equation 5.47 converge to zero.

On the other hand, one can think of a trivial example of a distribution for z_K such that F_{z_K}. The example that corresponds to a situation without any perturbation and does not satisfy the above condition is the Heaviside step H, or in terms of probability density functions, the Dirac delta function δ at 0. Therefore, a question can be raised about the minimal assumption on the perturbations and the coinciding distribution of z_K that the spectral modularity indeed breaks down.

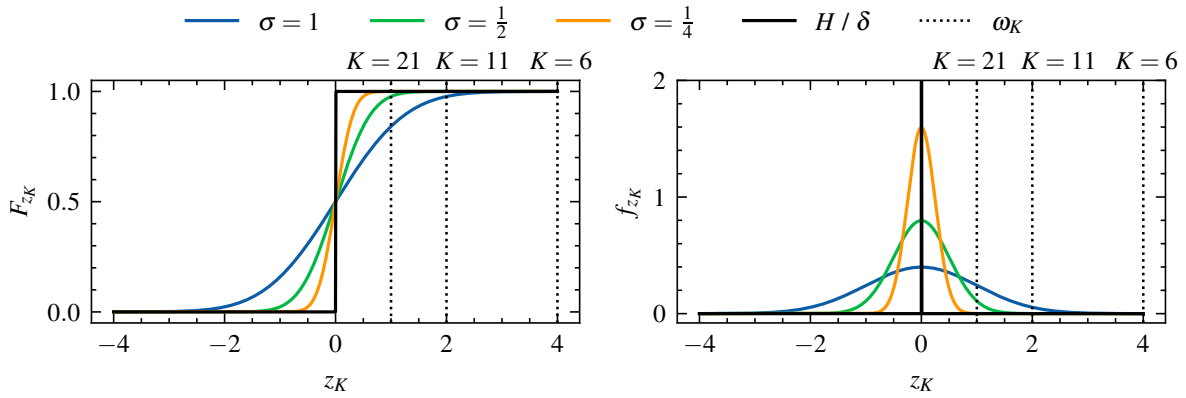


Figure 5.7: Pedagogical distribution of z_K. The left panel shows the cumulative distribution function for a selection of example distributions for z_K. The right panel shows the density functions for the same distributions. The black line represents the Heaviside function H in the left panel and the Dirac delta function δ in the right panel density. The colored lines represent Gaussian distributions with mean zero and $\sigma \in \{1, \frac{1}{2}, \frac{1}{4}\}$. The vertical dotted lines represent ω_K for $u = 0.2$ and $M = 10$.

While this example is indeed trivial, as it essentially corresponds to the situation without any perturbations at all, one can imagine that the same idea may be extended to a more general class of distributions that do not break the consistency condition A. In particular, these distributions would behave such that the mass moves faster to zero than the evaluation point ω_K does as K grows, for example, a distribution that satisfies

$$F_{z_K}(\omega_K) = p^{2/K}, \tag{5.48}$$

for some $p > 0$. Then it is clear that

$$\lim_{K\to\infty} \left(F_{z_K}(\omega_K)\right)^{K/2} = p > 0, \tag{5.49}$$

which makes the upper bound of the probability of satisfying the consistency condition A high, which therefore does not guarantee a spectral modularity breakdown.

This illustrates that spectral modularity breakdown only occurs for perturbations that are significant enough. However, as K grows, these distribution functions become more and more similar to the Dirac delta distribution, making their existence in the real world unlikely. Therefore, we make an assumption that is formally required to show the spectral modularity breakdown should be that distributions for z_K that satisfy

$$\lim_{K\to\infty} \left(F_{z_K}(\omega_K)\right)^{K/2} = 0. \tag{5.50}$$

This is achieved when $F_{z_K}(\omega_K)$ does not converge to 1. For a symmetric distribution around 0, this conceptually happens if the variance of z_K vanishes slower than the speed at which the evaluation point ω_K decreases to zero.

In Figure 5.7, we demonstrate a few pedagogical distributions that illustrate the triviality of the above-specified requirement on F_{z_K}. The colored lines represent the CDF and density function for three different Gaussian distributions, each with a different standard deviation. The black lines indicate the distribution associated with the point density at zero, i.e., the example distribution that does not show spectral modularity breakdown but is practically associated with having no perturbation at all. From the figure, we clearly see that if the standard deviations are supposedly small, i.e., more concentrated at zero, the distributions start to resemble the point density.

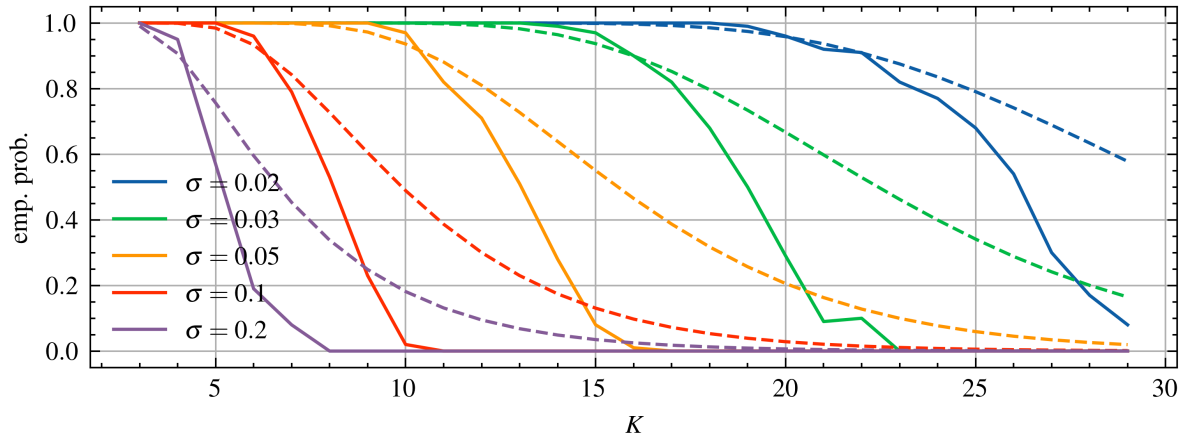


Figure 5.8: Empirical probability of satisfying the necessary Q_0-consistency condition A**.** The vertical axis represents the empirical probability. The horizontal axis represents the number of groups K. The empirical probability is obtained from Gaussian perturbations of the spectral modularity vectors for different values of σ. The dashed lines represent the theoretical upper bound for $\mathbb{P}(\mathcal{C})$ under the assumption that the sum z_K follows a Gaussian distribution.

5.4.4. Numerical Analysis of Breakdown

Now that we have obtained a theoretical understanding of the breakdown in terms of the sum of all perturbations z_K, what remains is studying the perturbations of the objects directly. However, because of the complicated dependence relations, this is difficult to do analytically. Therefore, we demonstrate how the perturbations of the spectral modularity vectors lead to breakdown with a numerical analysis.

Our numerical analysis is based on generating N random samples from TM-B with the perturbations $\{\mathbf{z}_i\}_{i=1}^n \in \mathbb{R}^{K-1}$, where $\mathbf{z}_i \sim \mathcal{N}(0, \sigma^2\mathbf{I})$. Then, for the choice of K and σ, we compute the N associated group affinity matrices, where we count the number of times the consistency condition specified in 5.12 is satisfied. Then we compute the empirical probability itself with $\frac{N_\mathcal{C}}{N}$, where $N_\mathcal{C}$ denotes the number of times $\tilde{\mathbf{G}}$.

In Figure 5.8, we demonstrate the spectral modularity breakdown of TM-B with the perturbation model described above. In essence, we observe that the empirical probability of the necessary condition being satisfied, i.e., event $\mathcal{C}$ shrinks to zero relatively fast and has a relatively sharp transition point. This observation is in alignment with our theoretical understanding of the spectral modularity breakdown, as the exponent in Equation 5.47 is of order K thereby enforcing an exponential decay after the transition point. Before the transition point, the number of groups and the perturbations are small enough that inconsistency in the group affinity matrix is not present. Additionally, it is clear that as the perturbations of the spectral modularity vectors are smaller, namely as σ decreases, the breakdown occurs only for larger K.

Furthermore, in the dashed lines of Figure 5.8 we display the theoretical bound, described by Equation 5.47, for some example distribution of z_K, namely $z_K \sim \mathcal{N}(0, \sigma^2)$. This is to indicate that if the sum z_K from Equation 5.45 satisfies a central limit theorem, such that the sum converges to a normal distribution, the quantity is indeed a theoretical upper bound to the probability satisfying Q_0-consistency condition A. Although the transition points of these dashed lines appear at roughly the same points as the solid lines, the decaying slopes are significantly less steep. This is for two reasons. Firstly, we underline that the quantity expressed in Equation 5.47 and displayed by the dashed line represents an upper bound of the probability. Secondly, the actual sum z_K does not necessarily follow a Gaussian distribution because of the dependencies in the sum, preventing the standard CLT results from applying.

6

Regularized Spectral Modularity

In this chapter, we introduce a regularization of the spectral modularity objective to mitigate the combinatorial saturation that causes the naive spectral modularity maximization to breakdown. In particular, we discuss the derivation of this method in two ways. First, we give an intuitive perspective on the correction for small positive modularity values that are the underlying cause of the inconsistent merges in spectral modularity breakdown, as discussed in Chapter 5. In this way, we see how subtracting a small constant from all entries of the spectral modularity matrix resolves the spectral modularity breakdown. Second, we show that this subtraction of a small constant is equivalent to adding an explicit regularization term to the standard spectral modularity objective, which penalizes the construction of heterogeneously sized groups. In this way, we see how the regularization reduces the bias towards clusterings with heterogeneously sized groups that is implicitly caused by the inconsistent merges.

The conceptual benefit of this regularization solution is that the adjustment to the spectral modularity framework lies only in changing the spectral modularity matrix by subtracting a small constant. This makes it possible to utilize existing modularity maximization algorithms. On the other hand, the regularization does require a parameter that needs to be calibrated separately.

In Section 6.1, the intuition behind the subtraction of a small constant from the spectral modularity matrix is given. In Section 6.2, the relation between this correction term and an explicit regularization of the original spectral modularity objective is given. In Section 6.3, a specific condition that reasonable clusterings of a data set should adhere to is introduced. This condition is based on random matrix theory and is used to calibrate the correction term. In Section 6.4, a practical calibration algorithm for the correction term is given.

6.1. Correction term

The primary observation that is obtained through studying the spectral modularity matrix of toy models in Chapter 5 is that the off-diagonal elements of the group affinity matrix, $\mathbf{S}$, converge to zero from below as the number of groups grows, even for easy clustering problems. Then, with only perturbations away from the ideal setting in the toy models, the necessary condition of negative off-diagonal elements in the group affinity matrix specified in Equation 5.12 is violated.

The saturation of the off-diagonal elements inspires the use of a small correction term. In particular, it is used to subtract the bias that is obtained from the combinatorial saturation as the number of groups grows. Therefore, to mitigate the spectral modularity breakdown, we employ the subtraction of a small constant.

Specifically, if $\mathbf{B}$ is a spectral modularity matrix, as is obtained through the procedure described in Section 4.3, then a corrected spectral modularity matrix is obtained by subtracting a constant $\epsilon \in \mathbb{R}$ from all the elements of the matrix, i.e.,

$$\mathbf{B}^{(\epsilon)} = \mathbf{B} - \epsilon \mathbf{J}. \tag{6.1}$$

Here $\mathbf{J}$ is a $n \times n$ matrix of ones. In its original form, spectral modularity essentially transforms the similarity measurements into modularity measurements, such that objects that are significantly similar have a spectral modularity above zero and objects that are significantly different have a spectral modularity below zero. Along these lines, by subtracting the ϵ correction term, we adapt the spectral modularity measure by increasing the threshold for significant similarity to be slightly larger than zero. This way, as the number of groups grows, the contribution of significantly different objects becomes more dominant, and therefore, the value of ϵ supposedly should be larger.

Because this adaptation of the modularity matrix is relatively simple, in that we solely replace the matrix $\mathbf{B}$ matrix with the corrected version $\mathbf{B}^{(\epsilon)}$, we are still able to use the well-established maximization algorithms that are used in the naive spectral modularity maximization as given in Chapter 4, such as Louvain. In fact, we can denote the ϵ-corrected objective as

$$Q^{(\epsilon)}(\rho) = \sum_{C_k \in \rho} \sum_{i \in C_k} \sum_{j \in C_k} \mathbf{B}^{(\epsilon)}_{ij}. \tag{6.2}$$

Here, it is clearly seen that in a comparison with the spectral modularity objective in Equation 4.1, the only different element is the modularity matrix. While the maximization procedure itself is unchanged, the constant correction term ϵ is not known and therefore needs to be calibrated, which we discuss in Section 6.3.

In Figure 6.1, we demonstrate the effect of the use of ϵ correction on the group affinity matrix. For a given spectral modularity matrix $\mathbf{B}$ and a value for ϵ, we compute $\mathbf{B}^{(\epsilon)}$ and display the group affinity matrix $\mathbf{G}^{(\epsilon)}$. The setting of this figure is the same as the setting of Figure 5.1 with $K = 18$. This means that using naive spectral modularity maximization, a relatively large number of clusters are inconsistently merged, as we see in the obtained clusterings in Figure 5.2, the structural matrices displayed in Figure 5.3, and the group affinity matrix in Figure 5.4.

In the leftmost figure, a value $\epsilon = -\frac{1}{200}$ is subtracted from the entries in the spectral modularity matrix. The resulting group affinity matrix for this value of ϵ has only positive elements. This will result in a trivial clustering, with all elements in a single cluster. In the second figure, a value of $\epsilon = -\frac{1}{1000}$ is subtracted from the entries in the spectral modularity matrix. The associated group affinity matrix shows that most of the off-diagonal elements are positive, but some are negative. In the third figure, we evaluate $\epsilon = 0$. Here, we have the original setting without any improvements. Indeed, we see that while most of the off-diagonal elements of the group affinity matrix are negative, some of the off-diagonal elements are positive. This way, a modularity maximization procedure will likely merge these clusters with positive elements in the group affinity matrix. Clearly, choosing $\epsilon < 0$ only worsens the behavior, as indicated by an increased number of positive off-diagonal elements for $\epsilon = -\frac{1}{1000}$ and $\epsilon = -\frac{1}{200}$. In the fourth figure, we see that a value of $\epsilon = \frac{1}{200}$ corrects the inconsistent group affinity matrix. This means that all the diagonal elements are positive and the off-diagonal elements are negative. Clearly, among the tested values for ϵ, this is the only value that does not break the Q_0-consistency condition A specified in Equation 5.12. In the fifth figure, however, we see that subtracting a too large value, i.e., $\epsilon = \frac{1}{100}$, will cause some diagonal elements to become negative. This will break the clusters associated with these diagonal elements. In the last figure, we see that for $\epsilon = \frac{1}{50}$, all the clusters will be broken, as indicated by the negativity of all the diagonal elements.

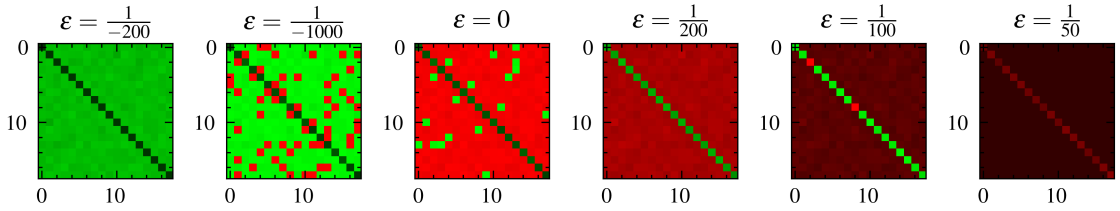


Figure 6.1: Group affinity matrix with different ϵ. The group affinity matrix $\mathbf{G}$ with $K = 18$ from Figure 5.4 with different values for ϵ, with the same color coding. Red specifies negative values, and green specifies positive values. The saturation of the color represents the absolute value. In the leftmost figure, all values are negative. In the rightmost figure, all values are negative.

6.2. Explicit Regularization

Although the interpretation of subtracting the constant correction term does align with our intuition obtained from Chapter 5, there is a second perspective that motivates the solution and places the solution in the more traditional explicit regularization paradigm. Namely, the subtraction of this constant can be equivalently written as an explicit regularization of the spectral modularity objective. In particular, rewriting the subtraction introduces a regularization term that penalizes heterogeneously sized groups. As seen in the empirical intuition demonstrated in Chapter 5, the spectral modularity breakdown causes clusterings that are obtained with naive SMM to have fewer clusters and be more heterogeneously sized than in the ground-truth. The heterogeneous sizes are explained by the fact that the positive off-diagonal group affinity matrix implies that a merge of two clusters is favorable according to the naive objective.

These inconsistent merges are the cause of large cluster size heterogeneity and a general under-approximation of the number of clusters. Therefore, reducing the bias towards cluster size heterogeneity while still maintaining the correct number of clusters is a promising approach to resolving the spectral modularity breakdown.

Consider that the ϵ-corrected modularity objective can be rewritten as

$$Q^{(\epsilon)}(\rho) = \sum_{C_k \in \rho} \sum_{i \in C_k} \sum_{j \in C_k} \mathbf{B}^{(\epsilon)}_{ij} = Q(\rho) - \epsilon \sum_{C_k \in \rho} |C_k|^2. \tag{6.3}$$

The latter term can be seen as the regularization term. In particular, the term penalizes clusterings with heterogeneously sized groups. Indeed, consider that the division is constrained to

$$\sum_{k=1}^{K} |C_k| = n. \tag{6.4}$$

Then, because of the quadratic term in the regularization, a maximum is attained at $C_1 = \{1, \ldots, n\}$ and $C_k = \emptyset$ for all $k \geq 2$. This term takes size n^2. On the other hand, a minimum is attained when $|C_k| \approx \frac{n}{K}$ for all $k \in \{1, \ldots, K\}$. This term takes size $\frac{n^2}{K}$. If any term $|C_k|$ is increased, it comes at the cost of a decrease in a different term $|C_h|$ for $h \neq k$. Because of the quadratic relationship, the increase in $|C_k|$ is larger than the decrease in $|C_h|$; therefore, the uniform distribution of objects among the K groups is indeed a minimum.

Taking this regularization parameter to an extreme may pose a limitation for clustering data that actually contains heterogeneously sized groups. However, subtle amounts of regularization are likely to prevent inconsistent merges from occurring. Especially when maintaining the correct number of groups within the clusterings.

6.3. Calibration Condition

The behavior of the objective maximization at asymptotic values of ϵ, i.e., large or small, is not difficult to study. In particular, if $\epsilon > \max_{i,j} \mathbf{B}_{ij}$, we have that $\mathbf{B}^{(\epsilon)}_{ij} < 0$ for all $i \neq j$. And, if $\epsilon < \min_{i,j} B_{ij}$, we have that $\mathbf{B}^{(\epsilon)}_{ij} > 0$ for all $i \neq j$. In the former case, it is clear that the optimal partition according to $Q^{(\epsilon)}$ is the partition of singletons, i.e.,

$$\hat{\rho}^{(\epsilon)} = \{\{1\}, \ldots, \{n\}\} \tag{6.5}$$

and in the latter case, the optimal partition is the singleton partition, i.e.,

$$\hat{\rho}^{(\epsilon)} = \{1 \ldots, n\}. \tag{6.6}$$

This illustrates the refining behavior of values of ϵ, and it clearly shows that finding a 'good' value for ϵ is far from trivial, as we are able to choose values for ϵ that are associated with the most extreme trivial partition choices. The way the optimal partitions, $\hat{\rho}^{(\epsilon)}$ change through moving from the minimal ϵ_- to the maximal ϵ_+, is not completely arbitrary; however, the exact behavior is difficult to describe, as the combination of many individual elements has a chaotic effect.

In practice, we do not know the ground-truth partition, and therefore there is no way to determine the actual size of the bias that we need to correct by subtraction. This makes it difficult to practically apply the algorithm, as one is required to choose a value for ϵ that constitutes the amount of subtraction. However, because we know that choosing ϵ at its two extremes gives the two extreme trivial partitions, where both are likely to have a large discrepancy with the ground-truth partition, it is probable that there exists a value for ϵ in between these two extremes for which the parameter is optimally chosen. We can make this assumption because if one of these extreme trivial partitions were present in the data, then methods from random matrix theory would apprehend this before the stage in which spectral modularity maximization is employed, e.g., by showing that there are no spiked eigenvalues outside the bulk and global component. Additionally, we know that there are as many different configurations of the optimal partition as there are distinct values in the modularity matrix. This makes it reasonable to expect enough flexibility obtained through the choice of ϵ, such that there exists at least an optimal value for it.

We can calibrate the parameter by comparing the approximated number of groups by detecting the number of spiked eigenvalues with the size of the clustering obtained through modularity maximization, where we make the assumption that $\hat{K} = K$. Let H be a modularity maximization algorithm that can be described by a function M that maps a modularity matrix to a clustering, i.e,

$$\mathrm{H} : \mathbb{R}^{n \times n} \to \mathcal{P}. \tag{6.7}$$

Then we denote the number of non-trivial sets in a clustering $\hat{\rho}$ by $\phi_1(\rho)$, where by non-trivial, we mean sets that are larger than size 1, i.e.

$$\phi_1(\hat{\rho}) = |\{C_k \in \rho : |C_k| > 1\}|. \tag{6.8}$$

Furthermore, if $\hat{K}$ denotes the number of desired clusters in $\hat{\rho}$, which is approximated by the number of spiked eigenvalues. Then, we want to find an approximate value for ϵ that satisfies the condition

$$\phi_1(\hat{\rho}^{(\epsilon)}) = \hat{K}, \quad \text{where} \quad \hat{\rho}^{(\epsilon)} = \mathrm{H}(\mathbf{B}^{(\epsilon)}). \tag{6.9}$$

In this setting, we only want to find the minimum value for ϵ such that the above condition is satisfied, i.e.,

$$\hat{\epsilon} = \min\{\epsilon \in [\epsilon_-, \epsilon_+] \quad : \quad \phi_1(\hat{\rho}^{(\epsilon)}) = \hat{K}\}. \tag{6.10}$$

This choice is mainly based on the philosophy that smaller adjustments to the original problem are favorable. Nevertheless, other estimates, such as a midpoint between the minimum value and the maximum value for which the condition is satisfied, may be meaningful too.

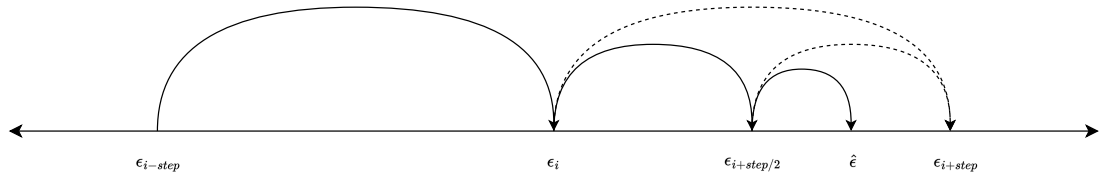


Figure 6.2: Illustration of ϵ calibration. The horizontal line represents the ordered ϵ search space $\mathcal{E}$. Arrows represent steps from a previous value for ϵ to a next value for ϵ. The solid arrows represent cases where the number of clusters is underestimated or correctly estimated. The dotted arrows indicate cases where the number of clusters is overestimated; therefore, the step is reverted and a smaller step is taken instead.

6.4. Parameter Search Algorithm

In order to find the value $\hat{\epsilon}$, it is possible to search in the finite set of all effective values. The number of effective values for $\hat{\epsilon}$ corresponds to the number of distinct values in the spectral modularity matrix $\mathbf{B}$. The search can be limited to this set because the optimal partitions associated with the use of ϵ and ϵ' can only be different when the positive and negative values of $\mathbf{B}^{(\epsilon)}$ and $\mathbf{B}^{(\epsilon')}$ are different.

Specifically, to solve the problem in Equation 6.10, we consider the search space for ϵ, denoted by $\mathcal{E}$, as follows:

$$\mathcal{E} = \{\mathbf{B}_{ij} | i \in \{1, \ldots, n\} \text{ and } j \in \{i+1, \ldots, n\}\}. \tag{6.11}$$

Then observe that $|\mathcal{E}| \leq \frac{n(n-1)}{2}$. Still, this can be cumbersome if the search space is not traversed efficiently. In particular, for each of the point evaluations, a modularity maximization procedure must be employed, which may take a considerable amount of time. For example, Louvain takes $O(n \log n)$ time. More importantly, we expect that there is not a big behavioral difference in the clustering when ϵ is only slightly different from another evaluation point.

Let $N = |\mathcal{E}|$ be the size of the ϵ search space. Then, we use the following ordering of elements as follows:

$$\epsilon_- = \epsilon_1 < \epsilon_2 < \cdots < \epsilon_{N-1} < \epsilon_N = \epsilon_+. \tag{6.12}$$

To efficiently evaluate the different values for ϵ, we take large steps instead of small steps that are dynamically adjusted when the obtained clusterings are significantly adjusted. Specifically, if $\hat{K}$ denotes the number of desired clusters, i.e., obtained through detecting the number of spiked eigenvalues as discussed in Chapter 3, In principle, there are three cases to consider that depend on the non-trivial size of a partition defined in Equation 6.8 and a desired number of clusters $\hat{K}$. First, if $\phi_1(\rho^{(\epsilon)}) < \hat{K}$, we know that the partition underestimates the number of clusters, and we therefore should increase ϵ. Second, if $\phi_1(\rho^{(\epsilon)}) > \hat{K}$, we know that the partition overestimated the number of clusters, and we therefore should decrease ϵ. Third, if $\phi_1(\rho^{(\epsilon)}) = \hat{K}$, we know that the partition has the right number of clusters, and therefore we want to find a value for $\hat{\epsilon}$ that is close to the value used to obtain $\rho^{(\epsilon)}$. In particular, we choose the minimum value for ϵ in which this condition is satisfied.

The calibration scheme that we use starts at the leftmost end of the search space $\mathcal{E}$. Therefore, initially, the value for the chosen ϵ is equal to ϵ_-. We define an initially relatively small step size $j \in \{1, \ldots, N\}$ that satisfies $j = 2^M$ for the largest possible integer M. Then, if we increase ϵ, we simply add j to i; therefore, if the current value for ϵ is ϵ_i, the next value is ϵ_{i+j}. If we decrease ϵ, we subtract j from i and half the step size to $j/2$ and add this to i. Therefore, in this case, ϵ_i changes to $\epsilon_{i-j/2}$.

Algorithm 2 ϵ **Calibration**: provides a value for ϵ

Input: $j = 2^M$, $i = 1$, $\hat{\epsilon} = \infty$, $\{\epsilon_i\}_{i=1}^N$, with for all $i \in \{1, \ldots, N\}, \epsilon_i \in \mathcal{E}$ and ordered according to 6.12, number of spiked eigenvalues $\hat{K}$
Output: $\hat{\epsilon}$
While the step size is greater than or equal to 1, i.e., $j \geq 1$,

- **Correct**: If $\phi_1(\hat{\rho}^{(\epsilon_i)}) = \hat{K}$: We set the value of $\hat{\epsilon}$ that satisfies the condition of 6.9, i.e., we set

$$\hat{\epsilon} = \epsilon_i. \tag{6.13}$$

  Then, we move to a smaller value of ϵ, i.e.,

$$\text{(decrease)} \quad i = i - j, j = j/2, i = i + j. \tag{6.14}$$

- **Overestimates**: If $\phi_1(\hat{\rho}^{(\epsilon_i)}) > \hat{K}$, we move to a smaller value for ϵ, i.e.,

$$\text{(decrease)} \quad i = i - j, j = j/2, i = i + j. \tag{6.15}$$

- **Underestimates, not previously correct**: If $\phi_1(\hat{\rho}^{(\epsilon_i)}) < \hat{K}$ and $\hat{\epsilon} = \infty$: We move to a larger value for ϵ, i.e.

$$\text{(increase)} \quad i = i + j. \tag{6.16}$$

- **Underestimates, previously correct**: If $\phi_1(\hat{\rho}^{(\epsilon_i)}) < \hat{K}$ and $\hat{\epsilon} < \infty$: The current value does not satisfy condition 6.9, but the previous value does. Therefore, we finish the while loop and return $\hat{\epsilon}$.

In Algorithm 2, we demonstrate the calibration algorithm. Using the initial value for i and j and a large value for $\hat{\epsilon}$, that indicates that no value has yet been found that satisfies the condition $\phi_1(\hat{\rho}^{(\epsilon)}) = \hat{K}$. In the algorithm, this is denoted as ∞, but in practice, an arbitrary large value suffices. The first case specifies the case where the condition is met; at that point, the current value ϵ is chosen for $\hat{\epsilon}$. The second case is where the number of clusters is overestimated. The third case is where the number of clusters is underestimated, but no satisfying value for $\hat{\epsilon}$ has been found yet. The fourth case is where a value for $\hat{\epsilon}$ was previously found, but the current values are no longer satisfactory. Therefore, we return the previously satisfactory value for $\hat{\epsilon}$.

In Figure 6.2, we demonstrate an illustration of the ϵ calibration scheme. The horizontal line represents the ordered ϵ search space $\mathcal{E}$. The arrows in the figure represent steps from a previous value for ϵ to a next value for ϵ. The solid arrows represent cases where the number of clusters is underestimated or correctly estimated. The dotted arrows indicate cases where the number of clusters is overestimated. In those cases, the step is reverted, and a smaller step of size $j/2$ is taken instead. In particular, from the initial position in the figure, a big step of size j is taken towards ϵ_i. Then, considering ϵ_{i+j}, we find that the number of clusters in the partition is overestimated; therefore, we halve the step size and consider $\epsilon_{i+j/2}$ instead. This procedure leads us to the value of $\epsilon_{i+j/2+j/4}$.

Because of the resemblance to binary search, which is obtained through the halving of the step sizes, we can observe that the calibration takes $O(\log N)$ steps in the worst case. Therefore, this procedure is relatively efficient, especially considering linearly searching the search space requires $O(N)$ steps. Furthermore, because N is $O(n^2)$, we have that if we use Louvain as our modularity maximization procedure of choice, the calibration procedure takes $O(\log N)O(n \log n) = O(n \log^2 n)$ time. On the other hand, with linear search, the entire time complexity takes $O(N)O(n \log n) = O(n^3 \log n)$ time. The most important difference between these time complexities is that the former is dominated by the time complexity of computing the spectral decomposition $O(n^3)$, while the latter is not.

7

Normalized Spectral Modularity

In this chapter, we introduce a second solution to mitigate the breakdown of spectral modularity maximization by employing a modification to the modularity objective. Spectral modularity maximization, in its naive form, maximizes the sum of all elements in the spectral modularity matrix $\mathbf{B}$ that are associated with the internal pairs of objects. In this way, adding an object to a big group generally has a larger increase in the objective function than adding an object to a small group, given that the individual pairwise modularities are of the same size. While in principle this should not be a problem if all the pairwise modularities are well-defined, the spectral modularity breakdown causes the naive maximization to be inconsistent. Therefore, we are interested in a normalization that discounts this bias towards creating large groups in the clustering. In Chapter 6, this mitigation is achieved by regularizing the objective to penalize clusterings with heterogeneously sized groups. In normalized spectral modularity maximization, we mitigate bias towards clusterings with heterogeneously sized groups by considering an alternative objective that does not favor clusterings with heterogeneously sized groups.

Because we redefine the objective, we cannot use existing maximization methods. In contrast to the explicit regularization solution from Chapter 6, we are required to develop new methods to maximize the normalized modularity. In particular, this maximization method is based on the angular orientations of the spectral modularity vectors of objects and cluster representative vectors. To do this, we first define a set of seed objects that are initially used as candidate positions for the group representative vectors. Using this set of seeds, a dynamic assignment phase can efficiently be used to provide a clustering of the entire data set.

In Section 7.1, we introduce the specific normalized objective and how it differs from the standard modularity and other normalizations of modularity. In Section 7.2, we present an interpretation of the magnitudes and orientations of spectral modularity vectors that helps to define a clustering method based on the normalized modularity objective. In Section 7.3, we define the set of seeds that are representative objects for the clusters and describe the procedure for finding the seeds. In Section 7.4, we describe the normalized spectral modularity maximization algorithm.

7.1. Normalized Objective

The essence of this solution lies in the adaptation of a normalization in the modularity objective defined in Equation 4.1. However, as the modularity objective is changed, using existing maximization methods becomes cumbersome. In particular, this is the case because existing methods, such as Louvain, make use of an explicit expression for the change of modularity by changing the clustering of a single object. However, this is no longer possible when we consider a normalized objective. In order to demonstrate the derivation of the normalized spectral modularity objective and why the existing maximization procedure cannot be used, like in the regularization-based solution discussed in Chapter 6, we show how the normalized objective Q_{norm} deviates from the standard modularity objective Q_0 and from an average modularity objective Q_{avg}.

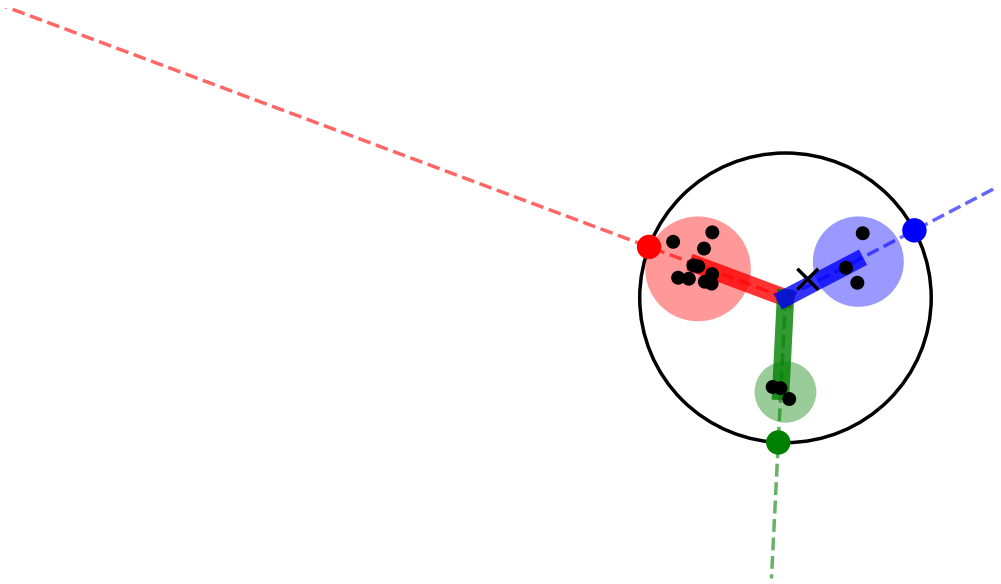


Figure 7.1: Spectral modularity vectors. The black dots represent the objects represented by their spectral modularity vectors in $\mathbb{R}^2$. The black circle is the unit circle. The different colors represent the different clusters. The dashed line represents the $\mathbf{z}_k$ associated with the standard modularity objective, Q_0. The dot on the unit circle represents $\frac{\mathbf{z}_k}{||\mathbf{z}_k||}$ associated with the normalized modularity objective, Q_{norm}. The thick solid line represents $\frac{\mathbf{z}_k}{|C_k|}$ associated with the average modularity objective, Q_{avg}

The new spectral objective can be obtained by recognizing the use of the orientations of cluster representative vectors in the traditional spectral modularity objective. In particular, given a K-partition ρ of $\{1,\ldots,n\}$ and the spectral modularity vectors $\{\mathbf{r}_i\}_{i=1}^n \subset \mathbb{R}^{\hat{K}-1}$, where $\hat{K}$ denotes the number of spiked eigenvalues, as discussed in Chapter 3, a set of K representative spectral modularity vectors can be obtained with the same philosophy that group representative data profiles, as discussed in Sec 2.2. The angular orientations of these representative spectral modularity vectors, hereafter referred to as 'cluster vectors' and denoted with $\{\mathbf{z}_k{}_{k=1}^K$, are fundamental in the spectral modularity maximization. In particular, the cluster vectors are defined by

$$\mathbf{z}_k = \sum_{j \in C_k} \mathbf{r}_j, \tag{7.1}$$

where $k \in \{1,\ldots,K\}$ and $C_k \in \rho$ for some partition ρ. In Figure 7.1 we display the spectral modularity vectors $\{\mathbf{r}_i\}_{i=1}^n$, with $\mathbf{r}_i \in \mathbb{R}^2$ for all $i \in \{1,\ldots,n\}$, associated to a small data set with three groups, i.e., $K = 3$, which is correctly determined by the number of spiked eigenvalues, such that $\hat{K} = 3$. The group sizes are deliberately heterogeneous in order to demonstrate the different behaviors of Q_0, Q_{avg}, and Q_{norm}. The colored dashed lines, thick solid lines, and dots represent different interpretations of the representative spectral modularity vectors that are used implicitly or explicitly in the different objectives.

Standard Modularity

Recall the standard modularity objective, Q_0, that is discussed in Section 4.3, i.e.

$$Q_0(\rho) = \sum_{C_k \in \rho} \sum_{i \in C_k} \sum_{j \in C_k} \mathbf{B}_{ij}. \tag{7.2}$$

This form is particularly useful, as it can be used to explicitly indicate a change in modularity. To be precise, let $\Delta_{i\to k}$ denote an operator on the argument of Q_0 that moves a single object i from C_h to cluster C_k and computes the difference, i.e., for some $\rho = \{C_1,\ldots C_h,\ldots C_k,\ldots,C_K\}$

$$\Delta_{i\to k} Q_0(\rho) = Q(\rho') - Q(\rho), \tag{7.3}$$

where $\{C_1,\ldots C_h',\ldots C_k',\ldots,C_K\}$ and $C_h' = C_h \setminus \{i\}$ and $C_k' = C_k \cup \{i\}$.

This quantity can be explicitly expressed as

$$\Delta_{i\to k}Q_0(\rho) = -\sum_{j\in C_h}\mathbf{B}_{ij} + \sum_{j\in C_k}\mathbf{B}_{ij}. \tag{7.4}$$

The computation of this quantity requires at most n summations. Therefore, an enumeration of all possible moves at each stage, like in Louvain, is tractable. Because the gain in the modularity objective Q_0 favors big groups, groups are merged inconsistently when the number of groups grows. This can be seen from the modularity change expression in equation 7.4, where if C_k is much larger than C_h, the expression is likely to be positive. Indeed, if $\mathbf{B}_{ij} = a \in [0,1]$ for all $j \in C_h \cup C_k$, meaning the individual pairwise modularities are all constant in both clusters, the term $\Delta_{i\to k}Q_0(\rho)$ will be positive simply because C_k is larger.

Now, consider the notation of the objective in terms of the spectral modularity vectors $\{\mathbf{r}_i\}_{i=1}^n$, then we can write

$$Q_0(\rho) = \sum_{C_k\in\rho}\sum_{i\in C_k}\sum_{j\in C_k}\mathbf{r}_i\cdot\mathbf{r}_j = \sum_{C_k\in\rho}\sum_{i\in C_k}\mathbf{r}_i\cdot\sum_{j\in C_k}\mathbf{r}_j = \sum_{C_k\in\rho}\sum_{i\in C_k}\mathbf{r}_i\cdot\mathbf{z}_k. \tag{7.5}$$

As we see in the right-hand side of the equality, modularity can be alternatively considered as a sum of the inner products of the spectral modularity vectors with their respective cluster vectors. Using this connotation, it is clear that large-magnitude cluster vectors are favored by the objective. The large magnitude of relatively large groups is seen in Figure 7.1, as indicated by the red dashed line.

Average Modularity

Therefore, it is tempting to find an adjustment to the objective such that it does not favor clusterings with relatively large groups. A simple way to think about this is by using averages. In the average modularity objective, we want to find the partition ρ that uses the same objective in Equation 7.5, but instead of using the term $\mathbf{z}_k = \sum_{j\in C_k}\mathbf{r}_j$, we use the average instead, i.e., $\frac{\mathbf{z}_k}{|C_k|}$. This gives us the following average modularity objective:

$$Q_{avg}(\rho) = \sum_{C_k\in\rho}\sum_{i\in C_k}\mathbf{r}_i\cdot\frac{\mathbf{z}_k}{|C_k|}, \tag{7.6}$$

which can be equivalently written as

$$Q_{avg}(\rho) = \sum_{C_k\in\rho}\sum_{i\in C_k}\sum_{j\in C_k}\frac{\mathbf{B}_{ij}}{|C_k|}. \tag{7.7}$$

Average modularity does not favor large groups. Because the cluster vectors are averaged, the number of objects in a group does not influence the magnitude of the terms $\{\frac{\mathbf{z}_k}{|C_k|}\}_{k=1}^K$. This is indicated by the roughly same size of the thick solid lines in Figure 7.1 that indicate these average cluster vectors, even though the cluster sizes are significantly different.

Furthermore, the Q_{avg} does have a relatively simple expression for $\Delta_{i\to k}Q_{avg}(\rho)$. Indeed, consider some $i \in C_h$ that we move to C_k. Then,

$$\Delta_{i\to k}Q_{avg}(\rho) = \sum_{r\neq i, j\in C_h}\mathbf{B}_{rj}\left(\frac{1}{|C_h|-1}-\frac{1}{|C_h|}\right) + \sum_{r\neq i, j\in C_k}\mathbf{B}_{rj}\left(\frac{1}{|C_k|+1}-\frac{1}{|C_k|}\right) \tag{7.8}$$

$$-\sum_{j\in C_h}\mathbf{B}_{ij}\frac{1}{|C_h|} + \sum_{j\in C_k}\mathbf{B}_{ij}\frac{1}{|C_k|+1}. \tag{7.9}$$

Because computing these sums is not too complicated, it is likely that an algorithm similar to Louvain that efficiently uses these modularity-change quantities is feasible.

Nevertheless, there is a fundamental problem with using Q_{avg}. The contributions of large-magnitude objects, which indicate larger absolute pairwise modularities, are not accounted for. Indeed, consider that the following two objects are in a cluster:

$$\mathbf{r}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } \mathbf{r}_2 = \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix}. \tag{7.10}$$

This gives that $\mathbf{z}_k = \begin{pmatrix} 0.6 & 0.1 \end{pmatrix}^\top$, where both vectors contribute equally to the direction of the cluster vector. Because $\mathbf{r}_1$ has a large magnitude, we want its influence on the cluster vector $\mathbf{z}_k$ to be more important than the small magnitude vector $\mathbf{r}_2$. Especially as any object has a relatively insignificant pairwise modularity with object 2, it is much less likely to be a representative object for that cluster. On the other hand, objects have significant pairwise modularities that can be positive or negative. With $\mathbf{r}_1$, this makes it much more representative of the cluster.

Unit Normalization

Fortunately, there is a more natural objective that does not have the problem that average modularity suffers from. Instead, we consider the unit normalization of $\{\mathbf{z}_k\}_{k=1}^K$, i.e., for Euclidean norm $||\cdot||_2$, we have

$$Q_{norm}(\rho) = \sum_{C_k \in \rho} \sum_{i \in C_k} \mathbf{r}_i \cdot \frac{\mathbf{z}_k}{||\mathbf{z}_k||_2}.$$

To illustrate the benefit of using Q_{norm} over Q_{avg} and especially Q_0, we zoom in on the upper half of Figure 7.1. In particular, we consider the addition of an object to one of the two clusters, i.e., the red or the blue cluster. The newly added has a spectral modularity vector in $\mathbb{R}^2$ at a radius of r and an angle of θ in $\mathbb{R}^2$. Specifically, if $\mathbf{r}$ denotes this spectral modularity vector representation, then

$$\mathbf{r} = (r\cos\theta, r\sin\theta) \in \mathbb{R}^2. \tag{7.11}$$

In Figure 7.1, an example for $\mathbf{r}$ is position at $r = 0.2, \theta = 0.5$, indicated with a cross. In Figure 7.2, we display the effect on the objective functions of adding the new object, for different values of r and θ, to any of the two clusters. The effect of adding the object to the red cluster is indicated by the red lines. The effect of adding the object to the blue cluster is indicated by the blue lines. In the top row, we display a zoomed perspective of the spectral modularity vectors from Figure 7.1. The three different colored half circles correspond to the object assignments made based on that objective for the position of the newly added object. The color in the half circle indicates the decision based on that objective for that position of $\mathbf{r}$. For example, if at some point on the half circle the color is blue, then if the newly added object is positioned at that point, the object is added to the blue cluster according to the specific objective. The half circles are supposed to be displayed exactly at radius r, but are slightly spread out to visualize all three of the objectives simultaneously. In the bottom three rows, we visualize the objectives explicitly as a function of θ. The colors displayed in the half circles in the top row are exactly the colors of the cluster with the highest value for the objective for that given value of θ. This is seen by the alignment of the intersections of the red and blue lines in the bottom figures and the color switch in the half circles in the top row.

When the radius of $\mathbf{r}$ is relatively large, i.e., when $r = 0.8$, we clearly see the tendency towards big groups of Q_0. The larger group, i.e., the red cluster, is more favorable to assign $\mathbf{r}$ to, even if it visually aligns the cluster vector of the blue cluster. Moreover, the other objective Q_{avg} and Q_{norm} have a lesser tendency to cluster $\mathbf{r}$ to the red cluster, as indicated by the larger blue section in the half circle.

When the radius is small, i.e., $r = 0.2$, as depicted in the left column of Figure 7.2, the same tendency prevails in Q_0. However, more importantly, Q_{avg} demonstrates an extreme favor to assign the object with the red cluster. This is because the objects with small magnitudes affect the orientation of the cluster vectors disproportionally. On the other hand, Q_{norm} is robust against this situation, as indicated by a similar decision point for large and small magnitudes.

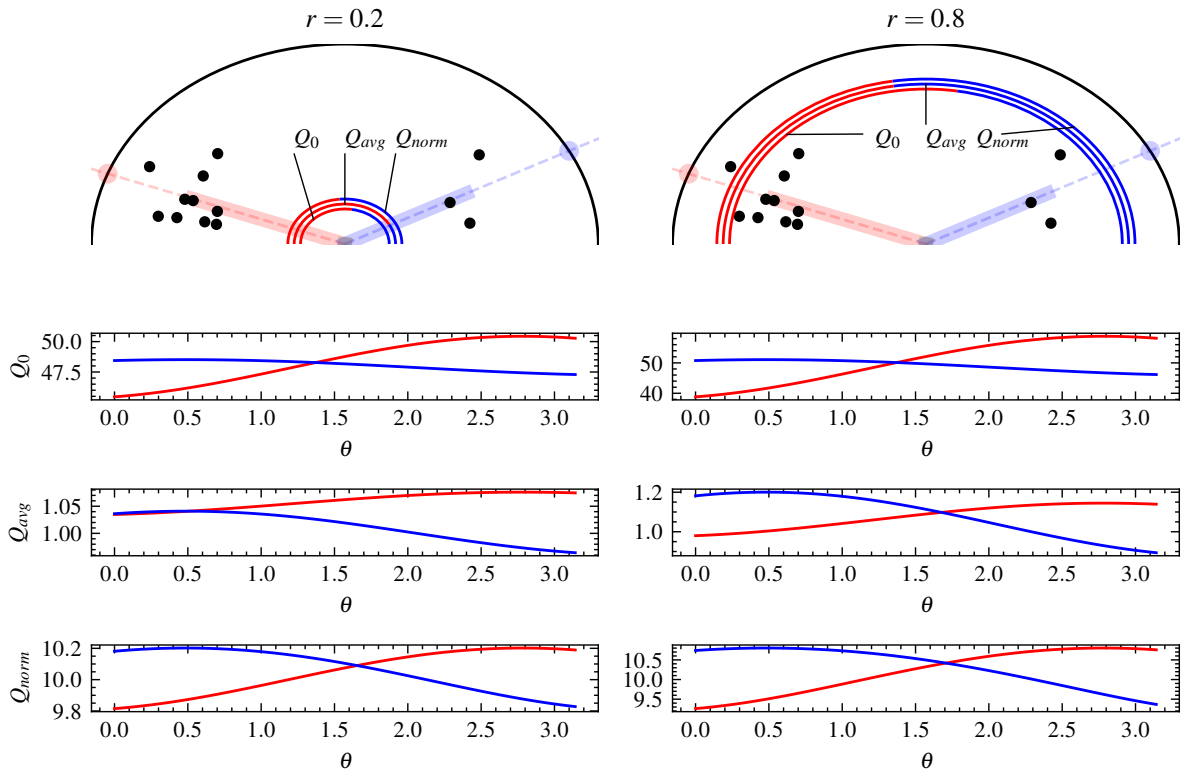


Figure 7.2: Comparison of spectral modularity objectives. The top row displays the same spectral modularity vectors as in figure 7.1, zoomed in on the top half circle. The dashed line represents $\{\mathbf{z}_k\}_{k\in\{red,blue\}}$ associated with Q_0, the thick solid line represents $\left\{\frac{\mathbf{z}_k}{|C_k|}\right\}_{k\in\{red,blue\}}$ associated with Q_{avg}, and the dot represents $\left\{\frac{\mathbf{z}_k}{||\mathbf{z}_k||}\right\}_{k\in\{red,blue\}}$, associated with Q_{norm} The bottom rows display the different objectives (Q_0, Q_{avg}, Q_{norm}) for the addition of an object with spectral modularity vector, $\mathbf{r} = r(\cos\theta, \sin\theta)$, to the respective cluster (red, blue) indicated by the respective colored lines. The colored half circles in the top row represent the cluster assignment decision (red, blue) based on the specified objective. The switch of color in these half circles is exactly at θ, where the blue and red lines intersect in the bottom rows for the associated objective.

While this specific example illustrates the different behaviors of the objectives, the actual reason for the normalization is to be applied in the context of a large number of groups, where the tendency towards a large group of Q_0 actually causes fundamental inconsistencies. Unfortunately, the spectral modularity vectors in dimensions larger than 3, cannot be visualized. Therefore, the relatively marginal difference in behavior between Q_0 and Q_{norm} in this low-dimensional setting should be taken with a grain of salt. However, this illustration does demonstrate the fundamental benefit of using Q_{norm} over Q_{avg}.

Although Q_{norm} has the potential to be robust against the challenges of spectral modularity breakdown, the objective is less amenable. In particular, for this specific objective function, the modularity change, i.e., $\Delta_{i\to k}Q_{norm}(\rho)$, cannot be expressed in the summation of $O(n)$ terms. Therefore, an alternative maximization approach is required.

7.2. Separation of Magnitude and Orientation

The trick that allows for an efficient maximization algorithm of Q_{norm} is to separate the orientations of the spectral modularity vectors from the magnitudes. To be precise, consider $\theta_{ij} \in [0, 2\pi)$ being the angle between $\mathbf{r}_i$ and $\mathbf{r}_j$. Then, the spectral modularity between i and j can be written as

$$\mathbf{B}_{ij} = \cos(\theta_{ij})||\mathbf{r}_i||\,||\mathbf{r}_j||. \tag{7.12}$$

From this, we see that the pairwise modularities can be decomposed into the cosines, which solely determine the sign of the pairwise modularity, and the magnitudes of $\mathbf{r}_i$ and $\mathbf{r}_j$, which have no influence on the sign of the pairwise modularity. Therefore, if the magnitude of an object i is large, the absolute pairwise modularities $|\mathbf{B}_{ij}|$ are likely to be large for all $j \in \{1, \dots, n\}$. On the other hand, if the magnitude is small, the absolute pairwise modularities are likely to be small. Therefore, the objects with large spectral modularity vectors are more representative of the objects in their cluster. This suggests that the algorithm should put an emphasis on these large-magnitude objects when clustering them.

Furthermore, if α_{ik} denotes the angle between the kth cluster vector, $\mathbf{z}_k$, of a partition and the spectral modularity vector of object i, then the normalized objective can be written as

$$Q_{norm}(\rho) = \sum_{C_k \in \rho} \sum_{i \in C_k} ||\mathbf{r}_i|| \frac{\mathbf{r}_i}{||\mathbf{r}_i||} \cdot \frac{\mathbf{z}_k}{||\mathbf{z}_k||} = \sum_{C_k \in \rho} \sum_{i \in C_k} ||\mathbf{r}_i|| \frac{\mathbf{r}_i}{||\mathbf{r}_i||} \cdot \frac{\mathbf{z}_k}{||\mathbf{z}_k||} = \sum_{C_k \in \rho} \sum_{i \in C_k} ||\mathbf{r}_i|| \cos(\alpha_{ik}). \tag{7.13}$$

Then, an important observation is that $\cos(\alpha_{ik})$ depends on the partition ρ through its dependence on $\mathbf{z}_k = \sum_{j \in C_k} \mathbf{r}_j$, but $||\mathbf{r}_i||$ is not dependent on the partition ρ. Hence, this procedure allows us to separate the information we obtain about the data through the spectral modularity vector magnitudes from the information we obtain from the spectral modularity vector orientations.

This is useful, as we cannot base our greedy decisions on the optimal moves of objects among subsets of the partition. Unlike in Q_0 and Q_{avg}, we cannot derive an expression for the delta modularity for a move due to the nonlinearity of the vector norms as depicted in Equation 7.4 and Equation 7.8, respectively. Despite this limitation, the separation of magnitudes and orientation gives us an alternative approach to optimizing the modularity objective Q_{norm}.

Instead of consistently updating the objective by moving objects around in the partition until a local optimum is found, which is not practical due to the complicated expression of ΔQ_{norm}, we have to consider a smaller search space. We do this by enforcing an ordering in which we cluster the objects. In particular, we order the objects by the magnitudes of the spectral modularity vectors.

Consider for some data set $\{\mathbf{x}_i\}_{i=1}^n$, with $\mathbf{x}_i \in \mathcal{X}$ for all $i \in \{1, \dots, n\}$, we have the spectral modularity vectors $\{\mathbf{r}_i\}_{i=1}^n$, with $\mathbf{r}_i \in \mathbb{R}^{\hat{K}-1}$, where $\hat{K}$ is the number of spiked eigenvalues and therefore the considered to be the number of groups. Then, we find an ordering of the indices $\{1, \dots, n\}$, i.e.,

$$j_1, \dots, j_n \in \{1, \dots, n\}, \text{ with } j_1 \neq j_2 \neq \dots \neq j_n, \tag{7.14}$$

such that,

$$||\mathbf{r}_{j_1}||_2^2 \geq \dots \geq ||\mathbf{r}_{j_i}||_2^2 \geq \dots \geq ||\mathbf{r}_{j_n}||_2^2. \tag{7.15}$$

By ordering spectral modularity vectors by their magnitudes, which relate to a sense of significance of their orientation, we ensure that the most meaningful decisions that influence the orientation of the cluster vector $\mathbf{z}_k$ are made early on. At the same time, the angular orientation of objects that are relatively insignificant, through their small magnitude, is made later.

7.3. Cluster Seeds

Given $\hat{K}$ detected spiked eigenvalues, we want to find $\hat{K}$ clusters, assuming that the ground-truth number of groups K is correctly approximated by the number of spiked eigenvalues $\hat{K}$. Therefore, there should be $\hat{K}$ vectors in $\mathbb{R}^{\hat{K}-1}$ that have negative pairwise dot products. This way, they are representative of distinct cluster vectors. However, the existence of $\hat{K}-1$ vectors with a negative dot product becomes increasingly rare when $\hat{K}$ grows, due to similar reasons behind the spectral modularity breakdown as suggested in Chapter 5. Therefore, for large $\hat{K}$, it is unlikely that any clustering will have $\hat{K}$ representative spectral modularity vectors that actually have pairwise negative dot products. Therefore, we initialize the clustering with a subset of $\hat{K}$ spectral modularity vectors in $\{\mathbf{r}_i\}_{i=1}^n$, which we call the cluster seeds. This specific initialization ensures that $\hat{K}$ clusters are constructed whenever there are $\hat{K}$ spiked eigenvalues.

Using the intuition from the separation of angles and magnitudes, there is a natural method to select candidate objects from the data set that are particularly important for representing the group structure. If we are able to find the objects with the largest magnitude, we will have a convenient starting point for the clustering procedure. Unfortunately, purely using the objects with the largest magnitude is susceptible to spectral modularity breakdown. Therefore, for obtaining the seeds, we still require the use of an ϵ parameter, akin to the regularization parameter defined in the regularized spectral modularity maximization (SMM1). Fortunately, we do not depend on the calibration scheme for the actual clustering but rather for the seed finding, which enables a more stable and compact procedure.

First, consider the following recursive set definition for the set of seeds S_m, where m represents the number of seeds that are found, and

$$S_0 = \{\}. \tag{7.16}$$

Consider the following set of indices of objects that represent candidate seeds, I_m. The spectral modularity vectors of candidates in I_m should all have negative dot products with the spectral modularity vectors of the seeds that are currently in the set S_m. Then, I_m is defined as

$$I_m = \{i \in \{1,\dots,n\} \setminus S_m \quad \text{such that} \quad \mathbf{r}_i \cdot \mathbf{r}_j < 0 \text{ for all} j \in S_m\}. \tag{7.17}$$

This makes $I_0 = \{1,\dots,n\}$. Now, consider the object s_m that represents one of the remaining candidates I_m with the largest magnitude, i.e.,

$$s_m = \arg\max_{i \in I_m} ||\mathbf{r}_i||_2^2. \tag{7.18}$$

Then, the recursive set definition for S_m depends on the availability of candidate objects. This means that if the candidate set I_m is empty, S_l is not defined for $l > m$. We define S_m as follows:

$$S_0 = \{\},\ S_{m+1} = S_m \cup \{s_m\} \text{ for all } m \geq 0 \text{ such that } I_m \neq \emptyset. \tag{7.19}$$

In other words, we want the set of seeds S to be a set of vectors such that the sum of the magnitudes of the vectors is maximal and the pairwise cosines are negative. Because for $\mathbb{R}^{\hat{K}-1}$, i.e., the space of the spectral modularity vectors $\{\mathbf{r}_i\}_{i=1}^n$, the maximum number of objects with pairwise negative dot product is K, we know that there must exist an $m \leq K$, such that S_m is defined but S_{m+1} is not. Let us denote this set with S. Because of the breakdown of spectral modularity, it is not guaranteed that $|S| = K$.

The set S above can be set can be equivalently defined as the solution to an optimization problem over the space of all possible subsets of $\{1,\dots,n\}$ with a constraint that requires pairwise negative dot products. The objective is to maximize the sum of all spectral modularity vector magnitudes of the seeds, i.e., for some set S, $\sum_{i \in S} ||\mathbf{r}_i||_2^2$. To be specific,

$$\max_{S \subseteq \{1,\dots,n\}} \sum_{i \in S} ||\mathbf{r}_i||_2^2, \tag{7.20}$$

$$\text{s.t. } \mathbf{r}_i \cdot \mathbf{r}_j < 0 \qquad \text{for all } i,j \in S \text{ that have } i \neq j. \tag{7.21}$$

We are specifically interested in finding a set of seeds that is of exactly size K, in alignment with the number of spiked eigenvalues. Therefore, with the same philosophy of the regularization parameter specified in Chapter 6, we use a correction term ϵ to correct for the breakdown in the seed finding stage. In essence, we are interested in finding a value for $\epsilon \in (\epsilon_-, \epsilon_+)$ such that the size of the set of seeds obtained from the following slightly adjusted optimization problem is equal to $\hat{K}$. To do this, we use the optimization problem defined in Equation 7.20 defined above and add a constraint that ensures that the number of seeds is equal to K. Then, in order to ensure the feasibility of the problem that is otherwise prevented by the breakdown of spectral modularity, we replace the pairwise negative dot product constraint with an adjusted constraint that uses the term ϵ. Finally, we add the ϵ term to the variables of the optimization problem, such that a value can be chosen for which the seed size constraint holds. This value can be uniquely obtained in two ways. We can find the minimal value for ϵ such that the problem is feasible, or we can find the maximal value for ϵ such that the problem is feasible. This should not make a difference, as the magnitudes themselves do not change; the maximum magnitude spectral modularity vectors will remain the same.

The adjusted optimization problem is then:

$$\max_{S \subseteq \{1,\dots,n\}, \epsilon \in [\epsilon_-, \epsilon_+]} \sum_{i \in S} ||\mathbf{r}_i||_2^2, \tag{7.22}$$

$$\text{s.t. } \mathbf{r}_i \cdot \mathbf{r}_j < \epsilon \qquad \text{for all } i, j \in S \text{ that have } i \neq j, \tag{7.23}$$

$$|S| = K. \tag{7.24}$$

Now, from this optimization problem, we can additionally redefine the recursive set definition, which immediately leads to a practical implementation to solve the above problem:

$$I_m^{(\epsilon)} = \{i \in \{1, \dots, n\} \setminus S_m \quad \text{such that} \quad \mathbf{r}_i \cdot \mathbf{r}_j < \epsilon \text{ for all } \quad j \in S_m\}, \tag{7.25}$$

$$S_{m+1}^{(\epsilon)} = S_m^{(\epsilon)} \cup \{s_m^{(\epsilon)}\} \quad \text{where} \quad s_m = \arg\max_{i \in I_m^{(\epsilon)}} ||\mathbf{r}_i||_2^2. \tag{7.26}$$

Then, using the recursive definition, we have

$$S_K = \min\{\epsilon \in [\epsilon_-, \epsilon_+] : |S^{(\epsilon)}| = K\}. \tag{7.27}$$

In Figure 7.3, we illustrate the seed-finding procedure. The four images represent the evaluation of the four spectral modularity vectors with the largest magnitudes as candidates for seeds. The outer ring is the unit circle. The inner ring is the ring with the magnitude of the current spectral modularity vector that is being considered. In the leftmost figure, the absolute largest magnitude is assigned to the cluster. This is always the case, as it is the first vector that is considered. In the second-to-left figure, the second-largest magnitude vector is considered; this vector does not have a negative dot product with the existing seed vector and is therefore ignored. The third figure from the left, the third-largest magnitude vector, is considered. This vector does have a negative dot product with the existing seed and is therefore assigned to its own cluster. In the last figure, the fourth-largest magnitude vector is assigned to its own cluster because it has negative dot products with both the red and green cluster vectors. Unfortunately, the seed-finding procedure can only be visualized for $K = 3$, such that the dimensions of the spectral modularity vectors are limited to 2. Therefore, the use of the epsilon term is not demonstrated.

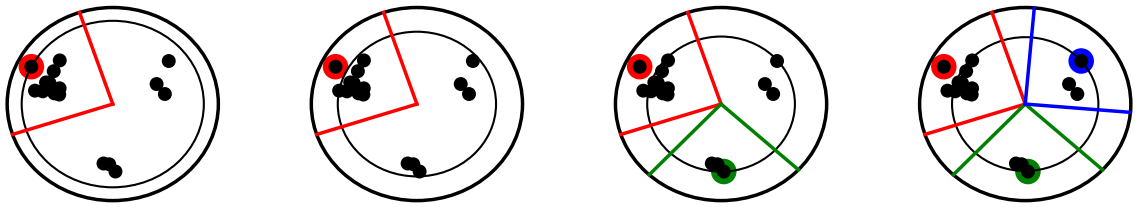


Figure 7.3: Illustration of seed finding procedure. The four figures represent the first four steps of the seed-finding procedure. In the leftmost figure, the vector with the largest magnitude is considered. In the rightmost figure, the vector with the fourth-largest magnitude is considered. The outer circles are unit circles. The inner circle is a circle with a radius corresponding to the magnitude of the vector that is being considered in that step. The black dots are the same spectral modularity vectors as in figure 7.1. The colors represent the different clusters. The colored dots represent the seed of the respective cluster. The orthogonal lines represent the cones around the vectors that should not intersect for the pairwise negative dot product condition to be satisfied.

7.4. Maximization Algorithm

Given the separation of orientations and magnitudes of the vectors, we obtain the ordering $j_1, \ldots, j_n$ that satisfies the magnitude sorting specified in Equation 7.15. In addition, given an initial set of seeds S that can produce the initial cluster vectors $\{\mathbf{z}_k\}_{k=1}^{K}$, what remains is to determine the remaining steps. To proceed, for a given index $j \in \{1, \ldots, n\}$ obtained from the ordering and such that $j \notin S$, we compute the current optimal assignment based on the largest cosine between $\mathbf{r}_j$ and the cluster vectors

$$k' = \arg\max_k \frac{\mathbf{r}_j}{||\mathbf{r}_j||_2^2} \cdot \mathbf{z}_j. \tag{7.28}$$

Based on this criteria, we assign object j to cluster $C_{k'}$. After this, the cluster vectors are recomputed, and the next index from the ordering is selected for the next cluster assignment.

Conceptually, the outline of the algorithm that can be used to maximize normalized spectral modularity can be divided into three phases:

1. Seed phase: First, we find the K objects in the data set that are each representative of a cluster.
2. Sort phase: Second, the remaining objects are ordered by the magnitudes of the spectral modularity vectors.
3. Assign phase: Finally, we repeatedly pick the first object that is not assigned yet (ordered by step 2) and compute the cosine between the object and the cluster vector. Then assign the object to the cluster with the largest cosine and update the cluster vector.

The complete algorithm is given in Algorithm 3. In Figure 7.4, an illustration of a selection of steps of the algorithm is given. The leftmost figure shows the initial state, where only three seed objects are assigned to their respective clusters. After this, the remaining objects are considered in an ordering that goes from the highest spectral modularity vector magnitude to the lowest magnitude. Then, the second object that is added is an object that belongs to the red cluster, as indicated in the middle figure. After 7 steps, the first object that does not belong to the red cluster is assigned. This is seen in the rightmost figure, where an object is added to the green cluster.

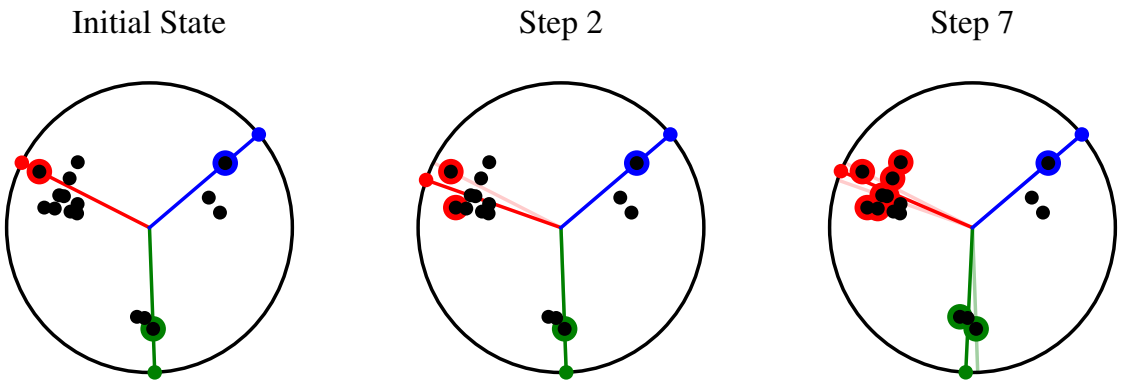


Figure 7.4: Illustration of maximization algorithm 3. The circle represents the unit circle. The dots represent the data set represented by their spectral modularity vectors in $\mathbb{R}^2$. The colors represent different clusters. The dots on the unit circle represent the normalized cluster vectors. The leftmost figure gives the initial state, where the seeds are assigned to their own cluster. The middle figure gives an illustration of the second step, where the object with the next largest magnitude of the spectral modularity vector is assigned to the best aligning cluster vector. In the rightmost figure, the first object that does not belong to the red cluster is assigned.

Algorithm 3 Normalized Spectral Modularity Maximization (SMM2)

Input: $\hat{K}-1$ dimensional spectral modularity vectors $\{\mathbf{r}_i\}_{m=1}^{n}$.
Output: partition $\rho = \{C_1, \ldots, C_{\hat{K}}\}$

1. **Seed Phase**: Obtain the seeds, S, by solving Equation 7.27. S is the set of $\hat{K}$ seeds that serve as initial positions for the latent representative objects.

$$\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{\hat{K}}\} = \left\{ \frac{\mathbf{r}_i}{||\mathbf{r}_i||_2^2} : i \in S \right\}. \tag{7.29}$$

   Initialize the clusters with the seeds, i.e., for $S = \{s_1, \ldots, s_{\hat{K}}\}$, for all $k \in \{1, \ldots, \hat{K}\}$ we have

$$C_k = \{s_k\}. \tag{7.30}$$

2. **Sort Phase**: Sort the indices by the magnitudes of the spectral modularity vectors, i.e., the quantities $||\mathbf{r}_i||_2^2$ for all i, to obtain an ordering of indices $j_1, \ldots, j_n$ that satisfies

$$||\mathbf{r}_{j_1}||_2^2 \geq \cdots \geq ||\mathbf{r}_{j_i}||_2^2 \geq \cdots \geq ||\mathbf{r}_{j_n}||_2^2. \tag{7.31}$$

3. **Assignment Phase**: For $m = j_1, \ldots j_n$, with $m \notin S$, do

   - Compute the current optimal assignment based on the largest cosine between $\mathbf{r}_m$ and the latent representative vectors.

$$k' = \arg\max_k \frac{\mathbf{r}_m}{||\mathbf{r}_m||_2^2} \cdot \frac{\mathbf{z}_k}{||\mathbf{z}_k||_2^2}. \tag{7.32}$$

   - Update the cluster

$$C_{k'} \leftarrow C_{k'} \cup \{m\}. \tag{7.33}$$

   - Update the latent representative vector.

$$\mathbf{z}_{k'} \leftarrow \sum_{j \in C_{k'}} \mathbf{r}_j. \tag{7.34}$$

8

Soft Spectral Modularity

In this chapter, we introduce a soft clustering method that is based on the spectral modularity vectors. The method uses the spectral modularity vector representations of objects in a data set to uncover different amounts of membership. In particular, given a hard clustering, the cluster representative spectral modularity vectors can be computed for each of the clusters. Then, objects can be proportionally assigned to the clusters with which they have a positive cosine. This way, the soft clustering procedure can be applied to partitions obtained from any hard clustering method.

In general, being able to quantify uncertainty or overlapping clusters is a highly desirable feature of clustering algorithms. Yet, not many of these methods outside of model-based clustering exist. For the main part of this thesis, cluster analysis is concerned with the finding of a partition that strictly assigns objects to a single cluster. However, such rigidity is not always a realistic expectation. Fundamentally, an object may belong partially to multiple clusters. Furthermore, the different levels of uncertainty between objects that are near the border of two clusters are ignored completely in clustering with these hard partitions.

Despite this clear benefit of soft clustering over hard clustering, the development of soft clustering is significantly smaller than its hard counterpart. For example, fuzzy KMeans [86] is a well-known method, but it requires a lot of parameter choices and is unlikely to perform well in high dimensions. In addition, clustering methods based on the inference of statistical models naturally provide soft clustering through probability distributions. However, fundamentally, these methods have computational and conceptual issues in high dimensional data, as discussed in Section 2.4, especially when a particular statistical model of the data is difficult to determine.

In Section 8.1, the mathematical definition of a soft partition is given. In Section 8.2, we uncover why the spectral modularity vectors are particularly interesting for a parameter-free soft clustering method by demonstrating challenges that exist with naive interpretations. In particular, we describe the problem of maximizing spectral modularity over all the possible soft partitions. Finally, in Section 8.3, a procedure based on the spectral modularity vector representations of objects is given that can be used to convert an arbitrary hard clustering into a soft clustering.

8.1. Soft Partition

A soft partition can be viewed as a mapping from objects to cluster assignments. Let $\{1, \dots, n\}$ be a set representing the indices of the data set. A soft partition of size K can be as a mapping from $\{1, \dots, n\}$ the points inside a regular K-simplex, denoted by $\mathcal{S}_K$, i.e.

$$\mathcal{S}_K = \left\{ \mathbf{w} \in [0,1]^K : \sum_{k=1}^{K} w^k = 1 \right\}. \tag{8.1}$$

In this interpretation, some object i with $\mathbf{w}_i \in \mathcal{S}_K$ associated with the object represents the amount of membership of the object i to the clusters. In particular, for some $k \in \{1, \dots, K\}$, the term w_i^k represents the amount of membership to cluster k.

Note that the definition of clusters can no longer be represented by subsets of $\{1, \ldots, n\}$ and therefore are only referred to by their index. Furthermore, this definition of soft partition specifies a particular symmetry in the order of the dimensions of the vectors in $\mathcal{S}_K$. To be precise, permutations of the order of dimensions associated with the cluster assignments, i.e., permutations of $\{1, \ldots, K\}$, are associated with the same soft partition.

In principle, a hard partition ρ can also be viewed from this perspective, and it conveniently highlights the symmetry of this mapping-based definition. A single cluster assignment of an object i in a hard partition can also be defined by a point in $\mathcal{S}_K$, namely a point $\mathbf{w}_i$, that satisfies

$$w_i^h = 1 \text{ for some } h \in \{1, \ldots, K\} \text{ and } w_i^k = 0 \text{ for all } k \in \{1, \ldots, K\} \text{ with } k \neq h. \tag{8.2}$$

Then, consider that there are equivalently defined cluster assignments for all objects consistent with the partition ρ, i.e.

$$\text{for all } i \in \{1, \ldots, n\} \text{ and } k \in \{1, \ldots, K\} : w_i^k = \begin{cases} 1 \text{ if } i \in C_k, \\ 0 \text{ otherwise .} \end{cases} \tag{8.3}$$

In this setting, it becomes clear that a permutation of the dimensions of $\mathcal{S}_K$ is equivalent to permuting the indices of the clusters $\{C_k\}_{k=1}^K$ of the partition ρ, which is invariant to these permutations as the partition is simply a set of sets.

Furthermore, soft clustering can be seen as a relaxation of hard clustering. The size of the space of soft partitions is, however, uncountable due to the cluster assignments taking values in the continuous interval $[0, 1]$. This makes it difficult to approach the soft clustering problem from the same perspective as hard clustering. Therefore, alternative methods are required.

To concretize this relatively abstract notion of a mapping of objects to cluster assignment, the soft partitions can be practically represented by a $n \times K$ matrix, which we refer to as the partition matrix. Specifically, for any, soft or hard, partition, there exists a representative matrix $\mathbb{P} \in [0, 1]^{n \times K}$ that satisfies

$$\mathbf{P}_{ik} = w_i^k \text{ with } \mathbf{w}_i \in \mathcal{S}_K, \tag{8.4}$$

for all $i \in \{1, \ldots, n\}$ and $k \in \{1, \ldots, K\}$. In particular, as a direct result, we have

$$\sum_{k=1}^{K} \mathbf{P}_{ik} = 1, \tag{8.5}$$

for all $i \in \{1, \ldots, n\}$. For hard partitions, the related partition matrices are additionally constrained to the space of $\{0, 1\}^{n \times K}$. For a given hard partition, this corresponds to the following notation. We can represent the partition as an ordered list of labels for each object, with $k_i \in \{1, \ldots, K\}$ for all $i \in \{1, \ldots, n\}$. Here, the labels $\{k_i\}_{i=1}^n$ are defined such that the objects that share the same label are in the same group. Then, this representation entices another representation that is particularly useful in this study. Consider the row vector $\mathbf{p}_i \in 0, 1^K$ that is the 'one-hot encoding' of the label k_i. Specifically,

$$\mathbf{p}_i = \left[\underbrace{0}_{1}, \cdots, \underbrace{0}_{k_i - 1}, \underbrace{1}_{k_i}, \underbrace{0}_{k_i + 1}, \cdots, \underbrace{0}_{K} \right]. \tag{8.6}$$

However, given a soft or hard partition, there is not a unique partition matrix due to the above-mentioned symmetries. The symmetries in the partition matrices are represented by the invariance to permutations of the columns of $\mathbf{P}$. In particular, for a given, soft or hard, partition, there are $K!$ equivalent partition matrices.

8.2. Ineffectual Relaxation

Because soft partitions can be seen as a relaxation of hard partitions, it is tempting to maximize modularity over all the possible soft partitions. While this is rather cumbersome to consider from the perspective of soft partitions being a mapping between objects and K dimensional cluster assignment, the partition matrices allow for an intuitive relaxation. In particular, we can define the space of all possible partition matrices, which we from now on refer to as the space of Markov matrices, denoted by $\mathcal{M}$, and formally define such that the matrices in $\mathcal{M}$ have positive elements and their rows sum to one, i.e.

$$\mathcal{M} = \{\mathbf{P} \in [0,1]^{n\times K} : \mathbf{P}\mathbf{1}_K = \mathbf{1}_n\}. \tag{8.7}$$

Alternative names that are commonly used to describe Markov matrices are 'stochastic matrices' and 'probability matrices', which resemble the practical setting in which these matrices are often used. However, to prevent ambiguity with 'random matrices' in general, we use the term Markov matrices to denote this class of matrices.

The space of hard partition matrices can be written in a similar form by constraining the cluster assignments to the set $\{0,1\}$, i.e.

$$\partial\mathcal{M} = \{\mathbf{P} \in \{0,1\}^{n\times K} : \mathbf{P}\mathbf{1}_K = \mathbf{1}_n\}, \tag{8.8}$$

where we use the notation $\partial\mathcal{M}$ as the space of hard partitions can be seen as the boundary of $\mathcal{M}$, on which we elaborate below. The hard and soft partitions are closely related in their natural descriptions; however, their traditional definitions are not particularly similar. The differences are that hard partitions assume mutual exclusivity, while in soft partitions, the partition is endowed with a membership function for each subset.

The maximization of the modularity objective as defined in Equation 4.1 can be equivalently written in terms of (hard) partition matrices and the search space of hard partition matrices $\partial\mathcal{M}$, i.e., note that

$$\max_{\rho\in\mathcal{P}} Q(\rho) = \max_{\rho\in\mathcal{P}} \sum_{C_k\in\rho}\sum_{i\in C_k}\sum_{j\in C_k} \mathbf{B}_{ij} = \max_{\mathbf{P}\in\partial\mathcal{M}} \sum_{k=1}^{K}\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{P}_{ik}\mathbf{P}_{jk}\mathbf{B}_{ij} = \max_{\mathbf{P}\in\partial\mathcal{M}} \mathrm{Tr}[\mathbf{P}^\top\mathbf{B}\mathbf{P}] \tag{8.9}$$

Therefore, if we consider that the maximum on the left-hand side is unique,

$$\rho^* = \arg\max_{\rho\in\mathcal{P}} Q(\rho) \text{ and } \mathbf{P}^* = \arg\max_{\mathbf{P}\in\partial\mathcal{M}} \mathrm{Tr}[\mathbf{P}^\top\mathbf{B}\mathbf{P}], \tag{8.10}$$

then the partition matrix $\mathbf{P}^*$ represents the partition ρ^*, i.e., they satisfy the relation specified in Equation 8.3. It should be noted that because of the symmetries in the partition matrices, the search space of the left maximization problem is larger without conceptually changing the space of partitions. Therefore, a naive maximization of the left objective is arguably more efficient. However, at this stage, we are interested in a relaxation to soft partition, which can be easily done in the partition matrix based objective.

A relaxation of the optimization problem in the right-hand side of Equation 8.10 to $\mathcal{M}$ gives us

$$\max_{\mathbf{P}\in\mathcal{M}} \mathrm{Tr}[\mathbf{P}^\top\mathbf{B}\mathbf{P}], \tag{8.11}$$

which makes it tempting to think that the relaxation of the search space of the clustering optimization problem from $\partial\mathcal{M}$ to $\mathcal{M}$ may give us an optimal partition in the interior of $\mathcal{M}$ such that it associates to a soft partition, as the search space including the soft partition matrices is much larger than only the space hard partition matrices.

However, in the case of the modularity objective Q_0 this is not true. In particular, the naive extension of modularity maximization to the space of soft partitions is non-trivial, which makes us consider an alternative approach to employing soft clustering in the next section. The ineffectual relaxation is, specifically, because the maxima that are attained when relaxing the search space of the optimization problem to include soft partitions are still hard partitions, due to the convexity of space $\mathcal{M}$ and function Q_0,

Convexity of $\mathcal{M}$

First, the space of $\mathcal{M}$ is convex, meaning that for some $t \in [0,1]$ and any two partition matrices, $\mathbf{P}, \mathbf{P}' \in \mathcal{M}$, we have that

$$\mathbf{P}^{(t)} := t\mathbf{P} + (1-t)\mathbf{P}' \in \mathcal{M}. \tag{8.12}$$

To see that this is indeed the case, consider that for some $i \in \{1,\dots,n\}$ and $K \in \{1,\dots,K\}$, we have $\mathbf{P}^{(t)}_{ik} = t\mathbf{P}_{ik} + (1-t)\mathbf{P}' \in [0,1]$. Furthermore, consider that $\sum_{k=1}^{K} \mathbf{P}^{(t)}_{ik} = t\sum_{k=1}^{K} \mathbf{P}_{ik} + (1-t)\sum_{k=1}^{K} \mathbf{P}'_{ik} = 1$. Therefore, we have that $\mathbf{P}^{(t)} \in \mathcal{M}$, demonstrating the convexity of $\mathcal{M}$.

Convexity of Q_0

Second, the modularity objective based on the partition matrices specified in Equation 4.12 is a convex function. This can be seen by the fact that the objective can be written as a sum of K convex functions of the form $\mathbf{x}^\top \mathbf{B}\mathbf{x}$, where $\mathbf{x}$ are the columns of the partition matrix $\mathbf{P}$. These K functions are convex because of the symmetric positive definiteness of $\mathbf{B}$, implying that the $n \times n$ Hessian matrices of these functions, which is $2\mathbf{B}$, are therefore also positive definite.

Extremal Boundary

Because the objective is a convex function on a convex set, the maxima are attained at the boundary of $\mathcal{M}$, which is $\partial\mathcal{M}$. To see this, consider that if $\mathcal{M}'$ is the largest open set within $\mathcal{M}$, i.e., the interior of $\mathcal{M}$, that is naturally obtained by changing the closed bounds of matrix entries to open bounds, we have

$$\mathcal{M}' = \mathrm{Int}\mathcal{M} = \{\mathbf{P} \in (0,1)^{n\times K} : \mathbf{P}\mathbf{1}_K = \mathbf{1}_n\} \tag{8.13}$$

Then, the boundary of $\mathcal{M}'$, written as $\partial\mathcal{M}$, written as $\partial\mathcal{M}'$ and defined by the closure of $\mathcal{M}'$ minus the interior, is the space of hard partition matrices, i.e., $\partial\mathcal{M}' = \mathrm{cl}.\mathcal{M}' \setminus \mathcal{M}'$.

8.3. Spectral Modularity Based Soft Clustering

To answer this need for an alternative approach, we propose here a procedure that is based on a hard partition ρ. To be precise, we use the relationship between the objects and the cluster vectors to determine soft cluster assignments for each of the objects. To highlight why this soft clustering procedure is a particular trait of spectral modularity and not of similarity, we first illustrate the philosophy of the procedure without the use of the spectral modularity vectors and show that it does not lead to a satisfactory method.

Consider a hard clustering ρ of $\{1,\dots,n\}$ that is of size K and a similarity metric s defined on the data space $\mathcal{X}$. To obtain a soft clustering from this hard partition, a naive procedure is to inspect the inferred representative profile $\hat{\boldsymbol{\mu}}_k \in \mathcal{X}$ of the data in the clusters of ρ. Then, one can construct a $n \times K$ matrix that contains for each object the similarity to all representative data profiles. Then, a natural assumption is that the amount of membership of an object i to the specific cluster k can be expressed proportionally by the similarity between the object i and the kth cluster profile. This would give a soft partition matrix generated by the following computation: for all $i \in \{1,\dots,n\}$ and $k \in \{1,\dots,K\}$ we have

$$\mathbf{P}_{ik} = \frac{s(\mathbf{x}_i, \hat{\boldsymbol{\mu}}_k)}{\sum_{k=1}^{K} s(\mathbf{x}_i, \hat{\boldsymbol{\mu}}_k)}. \tag{8.14}$$

However, this approach suffers from a fundamental problem. As the similarity metric is typically greater than zero, it will not be able to discount a global similarity level among all the objects. Therefore, all objects will likely have positive membership in all clusters. This is problematic, as it will significantly blur any actual strong cluster memberships that may be present. Any method that is based on this similarity based principle would then require parameter choices to determine a threshold that distinguishes significant similarity from insignificant similarity.

Algorithm 4 Spectral Modularity Based Soft Clustering

Input: hard partition $\rho = \{C_1, \ldots, C_K\}$, spectral modularity vectors $\{\mathbf{r}_{i=1}^n\}$, with for all $i \in \{1, \ldots, n\}$ $\mathbf{r}_i \in \mathbb{R}^{K-1}$

Output: soft partition matrix $\mathbf{P} \in \mathcal{M}_{n \times K}$

1. Initialize the partition matrix $\mathbf{P} \in \mathcal{M}_{n \times K}$ with for all $i \in \{1, \ldots, n\}$ and $k \in \{1, \ldots, K\}$:
$$\mathbf{P}_{ik} = \begin{cases} 0 \text{ if } i \in C_k \\ 1 \text{ otherwise} \end{cases} \tag{8.17}$$
2. Initialize the cluster vectors $\{\mathbf{z}_k\}_{k=1}^K$, with $\mathbf{z}_k = \sum_{i=1}^n \mathbf{P}_{ik}\mathbf{r}_i$.
3. Order the indices $\{1, \ldots, n\}$ according to the magnitudes of the spectral modularity vectors, as indicated in Equation 7.15 and denoted by $j_1, \ldots, j_n$.
4. For $i = j_1, \ldots, j_n$, do
   - Compute the cosines with all the K cluster vectors, i.e., for $k \in \{1, \ldots, K\}$, we compute $\cos\theta_{ik}$. Then, for all $k \in \{1, \ldots, K\}$
   $$\mathbf{P}_{ik} = \frac{(||\mathbf{r}_i|| \cos\theta_{ik})_+}{\sum_{h=1}^K (||\mathbf{r}_i|| \cos\theta_{ih})_+} \tag{8.18}$$
   - Update the cluster vectors with $\mathbf{z}_k = \sum_{i=1}^n \mathbf{P}_{ik}\mathbf{r}_i$

On the other hand, spectral modularity subtracts a global and random similarity component from the similarity matrix, as a way of separating the waves from the tides. Therefore, a large part of cluster memberships will be ensured to be negative and can be attributed to the actual presence of group structure as opposed to randomness. This makes the use of spectral modularity particularly suitable for parameter-free soft clustering.

Utilizing the spectral modularity vectors, described in Section 4.3, we can describe a weight assignment based on the cosines and magnitudes of spectral modularity vectors associated with objects and cluster vectors, akin to the maximization procedure discussed in Chapter 7.

If we have a hard partition $\rho = \{C_1, \ldots, C_K\}$ and consider the spectral modularity vectors $\{\mathbf{r}_i\}_{i=1}^n$, then we can use the same representative spectral modularity vectors as discussed in Chapter 7. To be specific, we have $\bar{\mathbf{z}}_k = \frac{\sum_{i \in C_k} \mathbf{r}_i}{||\sum_{i \in C_k} \mathbf{r}_i||}$ for all $k \in \{1, \ldots, K\}$, which represents the normalized version of the cluster vectors $\{\mathbf{z}_k\}_{k=1}^K$. Then the cosine of $\mathbf{r}_i$ and $\mathbf{z}_k$ is

$$\cos\theta_{ik} = \frac{\mathbf{r}_i \cdot \bar{\mathbf{z}}_k}{||\mathbf{r}_i||} \tag{8.15}$$

The beneficial aspect of using spectral modularity vector orientations, as opposed to similarity-based softening, is the natural distinction between positive and negative cosines. This way, we are able to clearly distinguish between object memberships that are meaningful and those that are not. In fact, consider the notation $(\cdot)_+$, for $(x)_+ = \max\{x, 0\}$, then

$$\mathbf{P}_{ik} = \frac{(||\mathbf{r}_i|| \cos\theta_{ik})_+}{\sum_h (||\mathbf{r}_i|| \cos\theta_{ih})_+} \tag{8.16}$$

This way, only the assignments that have positive cosines are considered proportional. Where the proportions are determined from the magnitude and the actual cosines. In Algorithm 4, the procedure is formally described.

Part III

Experimental Setup and Analysis

9 Synthetic Data Generation

In this chapter, we introduce the data generation processes (DGPs) that are used in our experimental evaluation of the clustering method. In order to evaluate the performance of clustering methods, it is beneficial to compare clusterings with a ground-truth partition that is known to be meaningful in its context. One might be inclined to use labeled empirical data sets to obtain such a ground-truth partition. However, this approach is unlikely to produce an insightful evaluation. A primary reason for this is that there is no guarantee that the provided labels are actually meaningfully representative of the data. With synthetically generated data, we can compare the enforced group structure with the clustering obtained from methods to quantify performance. Furthermore, there are only a small number of high-quality, labeled empirical data sets available that are ready to use. Finally, there is no controlled flexibility in empirical data sets with varying characteristics that can influence the effectiveness of specific clustering algorithms. With synthetically generated data, there is a clear representation of the ground-truth partition. Furthermore, the characteristics of synthetic datasets can be flexibly controlled to represent different regimes of interest in this study and different levels of clustering difficulty. To this end, we use two specific DGPs.

The first DGP is based on the well-known Gaussian Mixture Model (GMM) that we introduced briefly in Section 2.3 and have made anecdotal use of in Chapter 5. The GMM-based DGP is used to generate relatively easy data sets that have groups that are centered around the means. By varying the separation of the group centers, we can create different levels of difficult-to-cluster data sets. However, because in the GMM setting, the distances between objects and the associated group center are uniformly distributed, clustering is generally easy, unless the separation between groups becomes arbitrarily small.

The second DGP is based on the categorical mixed prototype model (CMPM) and provides a counterpart to the relatively easy setting of GMM. The CMPM-based DGP is used to generate data sets that have overlapping groups with heterogeneous levels of internal similarity, making the distances between objects and their group centers internally non-uniform in comparison to data generated with GMM. The CMPM data for a single object is generated by mixing samples from K distributions through a procedure called prototype mixing and is controlled by a weight vector. With a mixing parameter, we can transition from relatively easy problems that are closely related to GMM to more difficult problems where the boundaries of the clusters are soft. The weights are randomly drawn from two different distributions: a Dirichlet distribution where relatively few objects are close to the cluster centers, and a logit χ^2 distribution where relatively many objects are close to the cluster centers.

In Section 9.1, we give an overview of the DGP, its basic parameters, and its structure. In Section 9.2, we discuss the GMM-based DGP and how the clustering difficulty can be varied. In Section 9.3, we introduce the CMPM-based DGP, discuss prototype mixing, and again explain how the clustering difficulty can be varied in this DGP.

Symbol	Description	GMM Def.	CMPM Def. (F^0)	CMPM Def. (F^1)
α	group proximity	Equation 9.1	Equation 9.3	
γ	prototype diffusion	-	Equation 9.4	
β	prototype mixing	-	Equation 9.7	Equation 9.9
η	group size heterogeneity	-	Equation 9.8	

Table 9.1: Symbols used in the description of the DGP and their meaning, where the GMM column refers to the definition in the Gaussian Mixture Model and the CMPM column to the definitions in the Categorical Mixed Prototype Model. F^0 refers to the CMPM with the Dirichlet weight distribution, and F^1 refers to the CMPM with the logit χ^2 weight distribution.

9.1. Data Generation Process

At its core, the data generation process is a statistical model that specifies a distribution from which synthetic data is randomly sampled. The data sets are sampled conditionally on a few model parameters that control the shape of the distribution. Each generated dataset can be expressed by a n by p data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$, in which the rows represent the objects and the columns represent the features. Therefore, a single object $\mathbf{x} \in \mathcal{X}$ is a multivariate data point in the data space $\mathcal{X}$.

In order to inject group structure into synthetic data, we choose K positions, prior to sampling, in the data space that are going to be highly representative of objects in a specific group. Such a representative data profile is denoted by a point $\boldsymbol{\mu}_k \in \mathcal{X}$ for some group $k \in \{1, \ldots, K\}$. Therefore, using the representative data profiles, the sampled objects are obtained from a group-specific distribution, which we further specify in the next sections. This way, the K distributions characterize the existence of groups while still facilitating a significant amount of randomness. A ground-truth partition can be derived directly from this DGP, as the sampled objects unambiguously belong to the group with the closest representative profile of the chosen set $\{\boldsymbol{\mu}_k\}_{k=1}^K$.

The DGP has a selection of parameters that are required to control the generation of synthetic data. In both DGPs, the most important parameters for this study are the number of groups, denoted by K, and the group proximity, denoted by α, while less important are the number of objects and the number of dimensions. In the CMPM-based DGP, we additionally have the amount of prototype mixing, denoted by β, and the group size heterogeneity, denoted by η, which are not defined for the GMM. Table 9.1 provides an overview of the specific symbolic usage in the DGP context, where n, p, and K are defined in the same way as throughout the rest of the thesis.

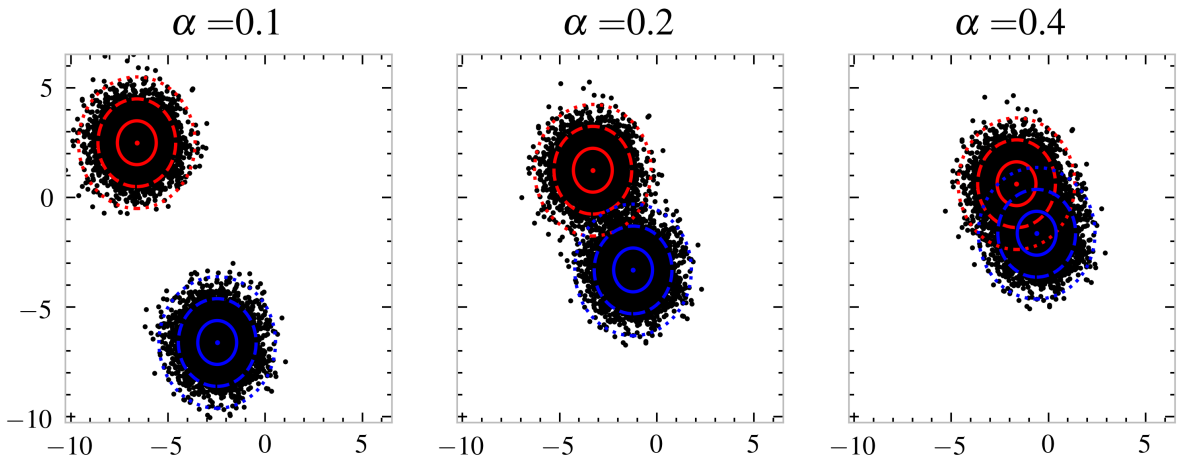


Figure 9.1: Scatter plot of two-dimensional Gaussian mixture data with group center The group proximity parameter α is different in each of the figures.

9.2. Gaussian Mixture Model

The setting of the Gaussian mixture model (GMM), as a special case of finite mixture models that we discussed in Section 2.3, is relatively well studied in the context of clustering. For example, [87] shows that a Laplacian-based spectral clustering algorithm is theoretically consistent under a GMM. Additionally, the Gaussian mixture model is often used for low dimensional clustering evaluation benchmarks, such as in [88, 89].

The GMM-based DGP generates data according to the following statistical model that is defined on the p dimensional real numbers, $\mathbb{R}^p$, as its data space. Specifically, if K be the number of groups. The objects in a specific group are distributed with a specific Gaussian distribution. In this setting, we enforce the size of the groups to be roughly identical. The distribution of the objects of a group k is a p-dimensional Gaussian distribution that is centered at $\boldsymbol{\mu}_k \in \mathbb{R}^p$ and has an identity covariance matrix $\mathbf{I}$. The group centers $\{\boldsymbol{\mu}_k\}_{k=1}^K$ are chosen such that

$$||\boldsymbol{\mu}_k - \boldsymbol{\mu}_h||_2^2 = \frac{1}{\alpha} \text{ for } k \neq h. \tag{9.1}$$

This means that for asymptotically large α, the group centers are identical, and for small α the group centers are far apart. Thus, α controls the distance of objects between groups. Without loss of generality, we use an identity covariance matrix, as the effect of the variance is only relative to the separation between the group centers. This leads to the following statistical model for $\mathbf{x}_i \in \mathbb{R}^p$:

$$\mathbf{x}_i \sim \sum_{k=1}^{K} \frac{1}{K} \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{I}) \qquad \text{for all } i \in \{1, \ldots, n\}, \tag{9.2}$$

where the $\frac{1}{K}$ term ensures that their probability of drawing a sample from each of the K distributions is equal, thus enforcing roughly equal group sizes in the ground-truth partition ρ^*. In Figure 9.1, we see a scatter plot of a two-dimensional Gaussian mixture model for different group proximity values. While $p = 2$ is not particularly high dimensional and therefore not strictly representative of the data that we generate with the GMM-based DGP, the visualization gives an illustration of the effect of the group proximity parameter α. In this setting, we use a Euclidean metric to compute distances, which is a natural choice for this kind of data. Furthermore, for two arbitrary points, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$, we consider the similarity metric defined in Equation 3.6.

9.3. Categorical Mixed Prototype Model

The CMPM-based DGP is defined in a categorical data space. The goal of mixed prototype-based DGP is to generate data that shares significant resemblance to multiple representative profiles, the prototypes, rather than a single one. Objects that are sampled by combining mixtures of prototypes can exhibit complex group structures with highly overlapping and low internally uniform groups, which makes them worthwhile to study.

For convenience, we encode the data space with integers, such that $\mathcal{X} := \{1, \ldots, K\}^p$. Furthermore, we choose K possible traits for each feature l to obtain the freedom of generating K maximally distinct prototypes, i.e., the possibility of having all pairs of prototypes attain Hamming similarity zero. Hamming similarity zero between two categorical objects is attained when the two objects have no features with a shared value. Let x_i^l be the lth feature of object i, i.e., $\mathbf{x}_i = (x_i^1, \ldots, x_i^p)$. Then, $\mathbf{x}_i \in \mathcal{X}$ and $x_i^l \in \{1, \ldots, K\}$.

In categorical data, this can be achieved by mixing the entries of samples from multiple distributions that are each centered on a prototype. The contribution of the K prototypes is indicated by a K dimensional weight vector $\mathbf{w}_i$, which is defined by a second class of probability distributions, the weight distribution. The weight distributions, denoted by F, are a probability distribution on the $K-1$ simplex, as introduced in Section 5.4 and depicted in Figure 5.6a, parameterized by a mixing parameter β and a heterogeneity parameter η. This distribution F is used to sample the weights $w_{i1}, w_{i2}, \ldots, w_{iK}$, which specify each prototype's contribution to the object i.

Furthermore, the K representative profiles, or specifically the prototypes, $\boldsymbol{\mu}_k \in \{1, \ldots, K\}^d$, are chosen such that they satisfy the following relationship:

$$s(\boldsymbol{\mu}_k, \boldsymbol{\mu}_h) \approx \alpha. \tag{9.3}$$

The relation between the similarity function and the group proximity parameter α is slightly different from the relation that is specified in the GMM-based DGP. For the categorical data specified in this section, the Hamming similarity between the prototypes is directly related to the proximity parameter. This is the case, as there is no ambiguous transformation between the Hamming similarity and the prototype generation procedure. Specifically, we can directly generate the prototypes to be at α similarity to each other. For the GMM, we encode this as the reciprocal of the Euclidean distance between the group centers. While the two quantities of α should not be compared across models, their directional behavior shares the same meaning. In practice, we generate the prototypes according to the algorithm specified in the appendix of [38], which is a convenient procedure for generating prototypes with a specific proximity.

The location of the distributions is determined by the K prototype. For the diffusion part of the distribution, we introduce a secondary parameter $\gamma \in (0, 1)$ specifying the amount of diffusion from the prototype, which is located at the center. We ensure that approximately (up to rounding) $\gamma \cdot d$ of the features are randomized, while the remaining features are the same as $\boldsymbol{\mu}_k$. Suppose that feature l for object i, denoted by x_i^l, is drawn from some distribution associated with prototype k. Then, we introduce diffusion by adding randomness. To be precise, we assume a $1 - \gamma$ probability that x_i^l will be equal to μ_k^l and a γ probability that it will be a uniformly random value from all the possibilities, i.e., $\{1, \ldots, K\}$. Then, we obtain the following statistical model for $\mathbf{x}_i \in \{1, \ldots, K\}^p$:

$$x_i^l \sim \sum_{k=1}^{K} w_{ik} \left(\gamma \underbrace{\mu_k^l}_{\text{location}} + (1-\gamma) \underbrace{\text{Cat}\left(\{1, \ldots, K\}; \frac{1}{K}, \ldots, \frac{1}{K}\right)}_{\text{diffusion}} \right), \tag{9.4}$$

$$w_{i1}, w_{i2} \ldots, w_{iK} \sim F(\beta, \eta). \tag{9.5}$$

If the amount of prototype mixing approaches zero, i.e., $\beta \to 0$, we obtain a DGP that is called "prototype generation", as is discussed in [38]. This specific procedure resembles the class of finite mixture models in the sense that there are only objects that are purely generated from a single distribution and not mixtures. This way, it is possible to see the class of mixed prototype models as a generalization of finite mixture models.

9.3.1. Weight Distributions

We consider the generation of data sets with the CMPM-based DGP for two different weight distributions. First, we discuss the Dirichlet weight distribution. In fact, specifically, the categorical distribution in the CMPM-based DGP shares some resemblances to the statistical model in Latent Dirichlet Allocation, rendering the results from this thesis more broadly applicable. The second weight distribution is a newly introduced model that is specifically aimed at providing a counterpart to the concentration shape of the Dirichlet distribution. The distribution is based on the logistic transformation of squares of normal random variables.

Dirichlet Weight Distribution (F^0)

The Dirichlet weight distribution has a probability density function that is defined by

$$f_K^0(w_1, \ldots, w_K; \beta_1, \ldots, \beta_K) = \frac{\Gamma(\sum_{k=1}^K \beta_k)}{\prod_{k=1}^K \Gamma(\beta_k)} \prod_{k=1}^K w_k^{\beta_k - 1}, \tag{9.6}$$

where Γ is the gamma function. Here, the constant term in front of the product is a normalization factor such that the probabilities integrate to 1.

Therefore, it can be conceptually supportive to understand the proportional expression of the density, i.e.,

$$f_K^0(w_1, \ldots, w_K; \beta_1, \ldots, \beta_K) \propto \prod_{k=1}^{K} w_k^{\beta_k - 1}. \tag{9.7}$$

Since we want an easy-to-vary parameterization of the distribution, we cannot use the Dirichlet distribution directly because it has too many parameters. Instead, for $k \in 1, \ldots, K$, we define

$$\beta_k \propto \beta \cdot K(1 - \eta/2)^k, \tag{9.8}$$

where we normalize β_k, such that $\sum_k \beta_k = \beta \cdot K$. This gives us a parameterization in terms of global concentration β and size heterogeneity η that satisfies $\beta_1 = \cdots = \beta_K$ when $\eta = 1$. This means that when the sizes are completely homogeneous (i.e., $\eta = 1$), the concentration of the distribution is symmetric. Sampling weights from the Dirichlet distribution can be done by directly using the explicit probability density.

In Figure 9.2, we see a two-dimensional histogram of the Dirichlet weight distribution for $K = 3$ and multiple values of η and β. The figures show the concentration of the weights around the center as β grows. In fact, when β is small, most of the mass of the distribution is concentrated at the corners of the simplex. This means that many objects exist that are close to their respective prototypes. Then, for $\beta = 1$ the distribution is approximately uniform. Finally, for $\beta = 4$ most of the mass is concentrated at the center of the simplex. At this stage, only a few objects resemble the pure prototype, as practically all objects are heavily mixed. When η, the group size heterogeneity is not zero, similar behavior in the weight distributions as a function of β occurs; however, the symmetry of the weights is much more skewed towards the corners.

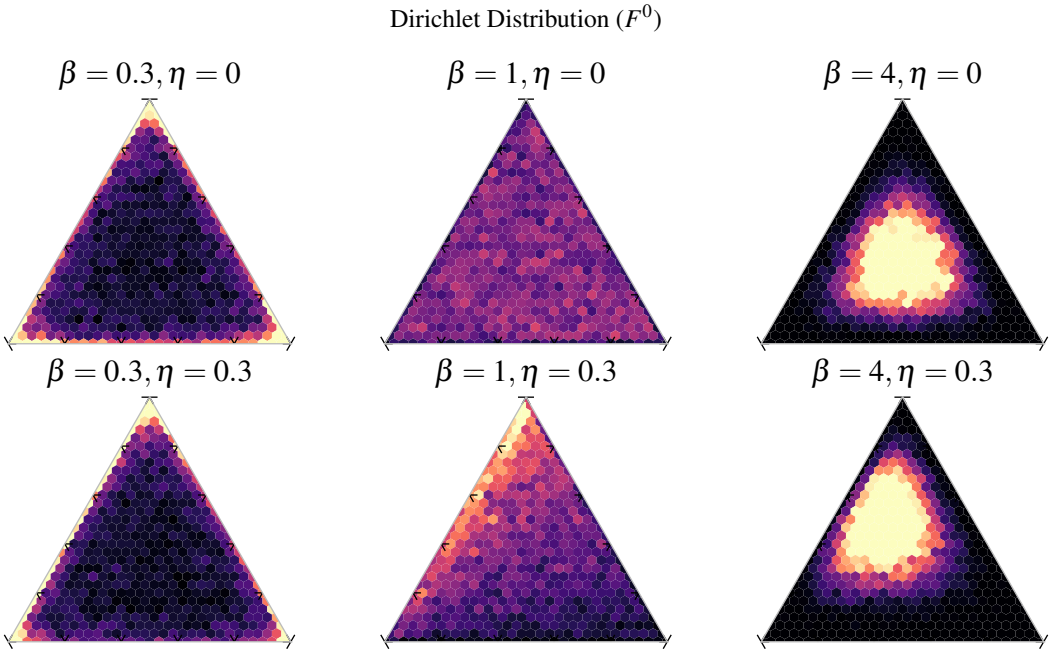


Figure 9.2: Two-dimensional hexagonal histogram of Dirichlet weight distribution (F^0) The different histograms represent different values of mixing β and group size heterogeneity η and $K = 3$.

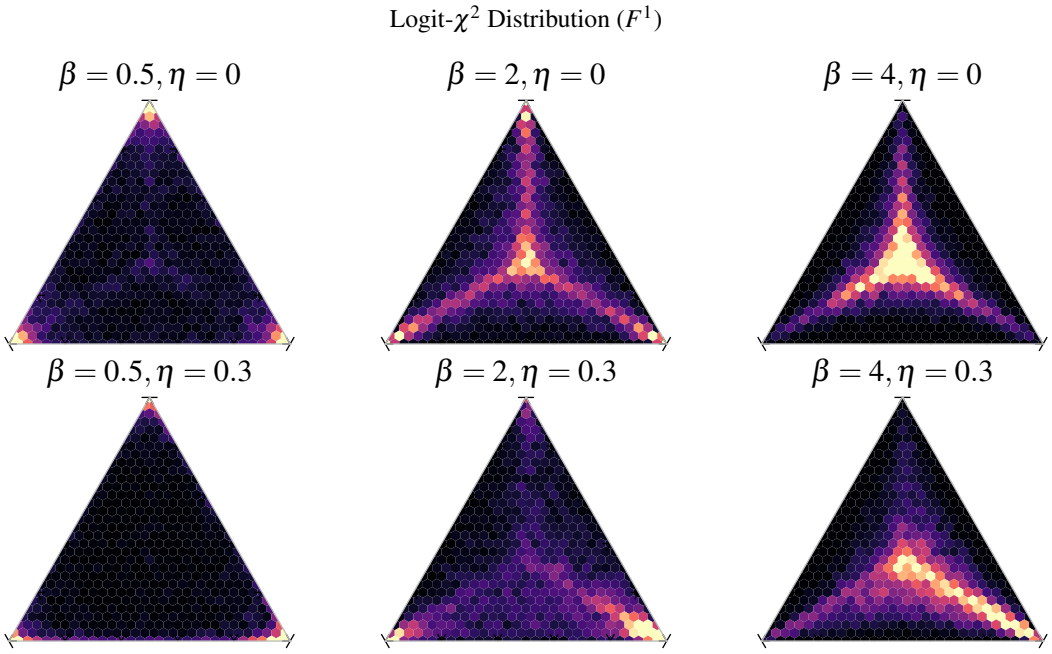


Figure 9.3: Two-dimensional hexagonal histogram of Logit χ^2 weight distribution (F^1). The different histograms represent different values of mixing β and group size heterogeneity η and $K = 3$.

Logit χ^2 Weight Distribution (F^1)

Another weight distribution on the $K-1$ simplex that we consider is the logit-χ^2 distribution, denoted by F^1. This distribution is inspired by the logit-normal distribution [90], which is defined by the distribution of a logistic transformation of Gaussian-distributed random variables. Although the Dirichlet distribution and the logit-normal distribution are not identical, they exhibit a similar concentration shape [90]. To enhance flexibility, especially in the shape of the concentration, we introduce the logit-χ^2 distribution, which is built upon the logit-normal distribution. The logit-χ^2 distribution is defined by the distribution of the logistic transformation of the squares of Gaussian random variables. The name "logit-χ^2" reflects the fact that we sample the objects by using a logistic transformation of squares of Gaussian distributed random variables.

The statistical model that is used to sample from the logit χ^2 distribution is

$$z_k \sim \mathcal{N}(\eta^{-k}, 1), \tag{9.9}$$
$$w_k \propto \exp\left(z_k^2/\beta\right), \tag{9.10}$$

where the weights $w_1, \ldots, w_K$ are again normalized such that $\sum_k w_k = 1$.

In Figure 9.3, we see a two-dimensional histogram of the logit χ^2 weight distribution for $K = 3$ and multiple values of η and β. The figures also show the concentration of the weights around the center as β grows in a similar way as observed in the Dirichlet distribution. However, as β grows, the decay of the proportion of objects that are relatively close to pure distributions is much slower. This is part of the reason for the study of this weight type. It simulates different tendencies to polarize into one pure distribution. Here, while more and more objects become oriented at the center, there still remain objects that are relatively purely associated with a single prototype. This specific aspect of this weight distribution is likely to encompass different behaviors in clustering algorithms than those studied in traditional clustering algorithms.

9.3.2. Mixing Illustration

In Figure 9.5 and Figure 9.6, we illustrate the mixing in the CMPM-based DGP and the resulting Hamming similarity matrices. Consider a setting with $K = 3$ prototypes that are maximally distanced, i.e., with proximity parameter $\alpha = 0$. For simplicity, we set the prototypes $\{\boldsymbol{\mu}_k\}_{k=1}^K$ to have identical entries for all features, i.e., for every group $k \in \{1, \ldots, K\}$ and feature $l \in \{1, \ldots, p\}$, we set $\mu_k^l = k$. In the figure, we display three $n \times p$ data matrices for $n = 300$ on the vertical axis and $p = 500$ on the horizontal axis. In the leftmost data matrix, we demonstrate a trivial data set where each of the entries is an exact copy of the prototype. This is associated with a data set with zero proximity α, zero mixing β and zero diffusion γ. In the leftmost similarity matrix in Figure 9.6, which corresponds to the same setting, we see that the similarity of objects that are within the same group, the diagonal blocks, is 1 and the similarity of objects that are in different groups, the off diagonal blocks, is 0.

Then, if we increase the amount of diffusion to $\gamma = 0.5$, we see in the rightmost panel of Figure 9.5 that the data matrix has become noisy, but in a uniform way. That means that the amount of noise added to the data matrix is roughly equal for every object. In the right most similarity matrix in Figure 9.6, we see how this addition of $\gamma = 0.5$ changes the similarity matrix. While the difference in similarities between objects of the same group and objects of different groups becomes smaller, the boundaries between groups are still clear.

If we increase β to 0.5 instead of γ, we obtain a different picture. In the middle panel of Figure 9.6, the data matrix is again noisy, but upon sorting the rows of the matrix by the closeness to its original group center without disrupting the visible group structure itself, we see the heterogeneous similarities of objects to prototypes. To be clear, the rows of the data matrix in the rightmost figure are sorted in the same way, yet they do not display the same level of heterogeneity. This difference shows the differentiating effect of diffusion γ (and implicitly group proximity α), and mixing β.

9.3.3. Density of states

Another observation from the middle similarity matrix in Figure 9.6 that describes the behavior of the CMPM-based DGP lies in the fact that sorting the rows of the data matrix by the closeness to its group center shows a transition of pairwise similarities from low to high. In the middle panel of Figure 9.6, we see gradients of pairwise similarities in the diagonal blocks. This demonstrates that as two objects are closer to the prototypes, their pairwise similarity is higher. On the other hand, as two objects are farther away from the prototypes, their pairwise similarity is lower. This contributes to the additional difficulty of disentangling the objects at the boundary. Additionally, this suggests that traditional methods based directly on distances or densities, e.g., KMedoids, are unlikely to perform well. On the other hand, this density of states phenomenon underlines the validity of clustering within this mixed prototype data. Even though boundaries are difficult to disentangle, as objects move away from the boundary, they are significantly more clustered together.

9.3.4. Statistical Distribution of Group Proximity

To effectively distinguish the difference between the group proximity parameter α and the mixing parameter β, it is useful to look at the histogram of similarities of objects to a group center $\boldsymbol{\mu}_k$ for some $k \in \{1, \ldots, K\}$. We display the histograms in Figure 9.4. In this figure, the $K = 3$ groups are equally sized, and we look at the similarities between all objects $\{\mathbf{x}_i\}_{i=1}^n$ and one prototype $\boldsymbol{\mu}_1$ that corresponds to one of the groups.

Initially, with small group proximity α and small mixing amounts β, as seen in the left-most column of the figure, the object-to-prototype similarity demonstrates two modes. One mode corresponds to the objects that are close to the center, i.e., that belong to the group, and one mode corresponds to the objects that are far away from the center, i.e., that belong to a different group. We see that the proportion of objects at the high similarity level is around a third of the total, and the objects at the lower similarity level are around two thirds. This corresponds with the fact that there are three equally sized groups, and we look at the similarities of the objects to one of the group centers.

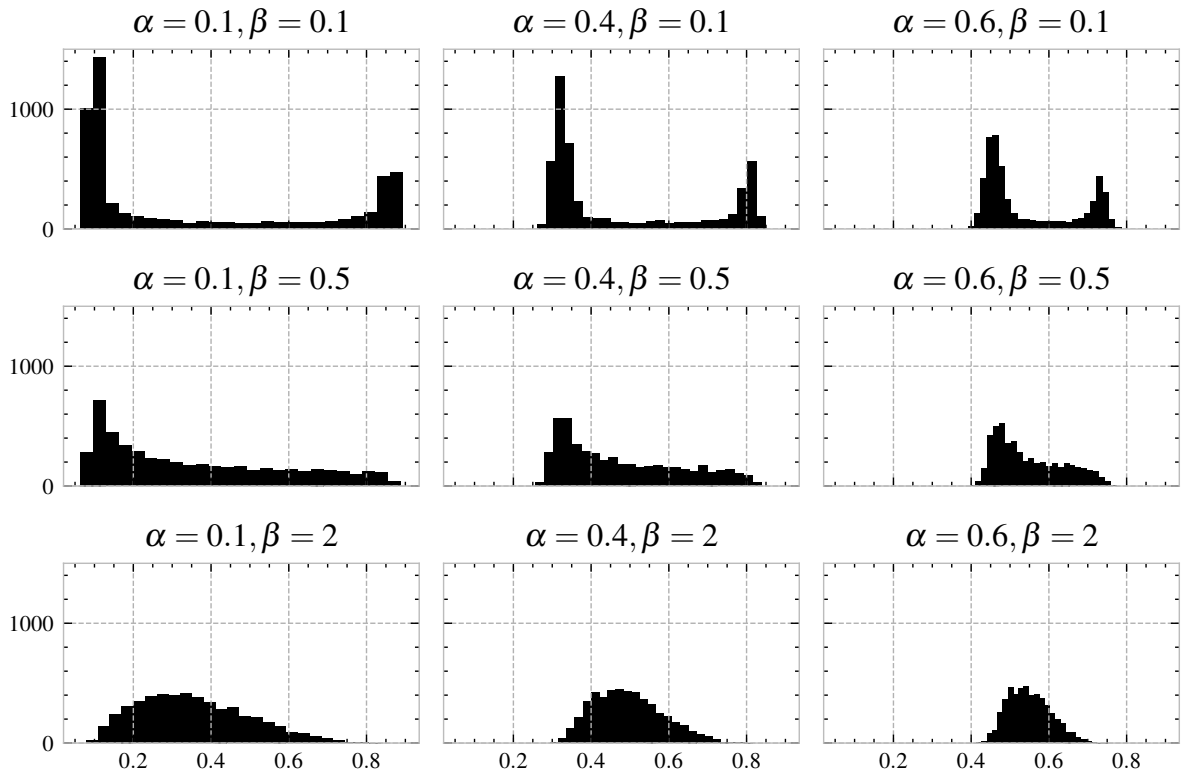


Figure 9.4: Histogram of Hamming similarity of objects to prototypes. The figure contains both the similarities between objects in C_1 and prototype $\boldsymbol{\mu}_1$ and similarities between objects in $C_2 \cup C_3$ and prototype $\boldsymbol{\mu}_1$. In the rows, we vary the prototype mixing β and in the columns, we vary the group proximity α.

We see that as the group proximity α grows, i.e., the prototypes become closer, the two modes also become closer and slightly wider; however, the bimodal shape remains until the histogram merges into a single relatively narrow symmetric single-mode distribution. In this final stage, which occurs as $\alpha \to 1$, which is not displayed in the figure, there is hardly any distinction between objects that belong to the group and objects that belong to a different group. This makes recovery of the partition through clustering methods theoretically impossible.

On the other hand, if β grows and α stays small, we see that the bimodal shape vanishes before reaching a trivially hard data set. While for low levels of β, the bimodal structure persists, for a slightly higher value of β the gap between the modes gets filled. At that point, there are no longer strictly separated modes, as is seen for $\beta = 0.5$ and even more for $\beta = 2$. While there is still a large difference in the object-prototype similarities, objects are no longer uniformly separated from their group centers. This creates a gradient of proximities from the center of the group to the boundary of the group, where the boundaries between two groups are no longer abrupt. This indicates that groups are not only overlapping more; the distribution of proximities between objects and their closest prototype is also more heterogeneous, or, in other words, less internally uniform. These two characteristics, low internal uniformity and high overlaps, are prominent in the CMPM-based synthetic data.

The combination of clear prototypical objects at the group centers with difficult-to-entangle group boundaries makes this a particularly difficult clustering task that is not theoretically impossible to cluster. This is an important reason to investigate the performance of clustering methods for the CMPM-based DGP.

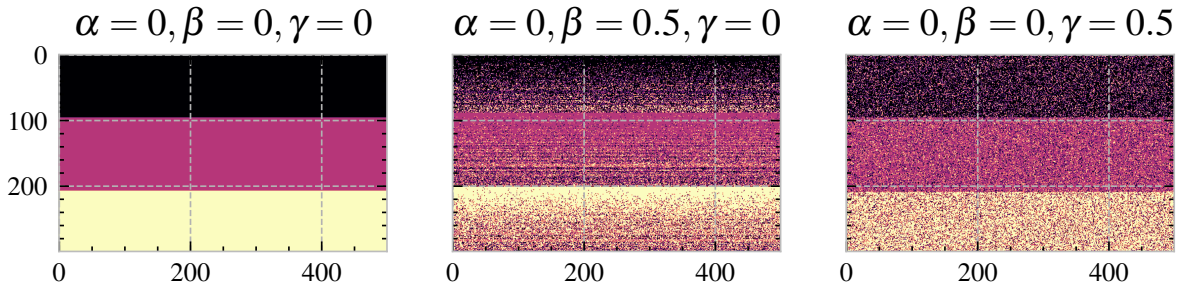


Figure 9.5: Categorical mixed prototype data matrices. The colors represent different categorical entries. The rows represent the objects. The columns represent features.

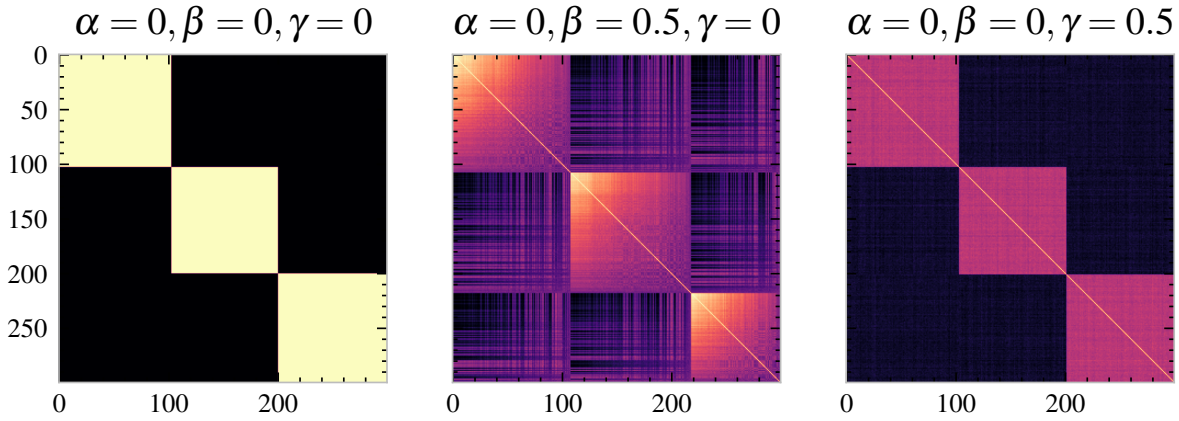


Figure 9.6: Hamming similarity matrices of data sets generated by the categorical mixed prototype data generation process for different parameters. The colors represent the level of similarity, with the brightest color having the highest similarity (1) and the darkest color having the lowest similarity (0).

10 Empirical Performance Analysis

In this chapter, we investigate the behavior of the spectral modularity based clustering methods that are introduced in this thesis. The experimental evaluation of the clustering methods is done by comparing their performance with existing clustering methods. In this evaluation, we use synthetic data to compare clustering methods with the ground-truth partition. Important properties of the data generation process (DGP) are controlled by several free parameters, which we vary in order to explore different regimes that are of interest. Specifically, we consider the synthetic data that is generated by the two DGPs defined in Chapter 9. We consider data that is generated with a Gaussian Mixture Model (GMM) based DGP. Furthermore, we consider data that is generated from a categorical mixed prototype model (CMPM) based DGP.

There are two performance evaluation criteria that we consider in our assessment of the clustering methods. Both assessments are based on the quantification of the discrepancy between clusterings and the ground-truth partition that is derived from the DGP. First, the ability of the methods to correctly recover the ground-truth partition from the data is measured with a partition-space distance metric between the clustering and the ground-truth. Second, we evaluate the ability to recover the ground-truth representative data profiles by measuring the discrepancy between the inferred profiles and the profiles associated with the ground-truth partition.

In our experiments, we examine naive spectral modularity maximization (SMM0) as described in Chapter 4. Furthermore, we investigate our two contributed enhancements of spectral modularity maximization. Specifically, we consider regularized spectral modularity maximization with partition based calibration (SMM1) as introduced in Chapter 6 and normalized spectral modularity maximization with the seed based search algorithm (SMM2) as introduced in Chapter 7. We compare our contributed algorithms with methods that are chosen such that we compare them to a rather simple baseline method but also to a competitive method. As a baseline clustering algorithm, we study the KMeans [41] or KMedoids [59] clustering algorithms (KM) depending on the data type (real data or nominal data, respectively). The competitive clustering method we study is a Laplacian-based spectral clustering algorithm (SC) [62]. These methods are further discussed in Section 2.3. This selection of clustering methods gives us four spectral methods, of which three are based on spectral modularity. For an overview of the methods, see Table 10.1. For the evaluation of the clustering methods in terms of group profile inference, we additionally study the soft-clustering variants of the methods using the procedure from Chapter 8.

In Section 10.1, we introduce the performance evaluation criteria of the clustering methods. We discuss the metrics for measuring the partition recovery and the profile inference. In Section 10.2, we study the partition recovery performance of the clustering methods on synthetic data from the GMM-based DGP. In Section 10.3, we study the partition recovery performance on synthetic data from the CMPM-based DGP. Finally, in Section 10.4, we demonstrate the performances in terms of profile inference in the context of CMPM data for each of the investigated methods and their soft-clustering variants.

Method	Color	GMM	CMPM
KMeans (KM).		•	
KMedoids (KM).			•
Spectral clustering (SC).		•	•
Naive ***spectral* modularity** maximization (SMM0).		•	•
Regularized ***spectral* modularity** maximization (SMM1).		•	•
Normalized ***spectral* modularity** maximization (SMM2)		•	•

Table 10.1: Methods used in the benchmark. The methods with the term ***spectral*** in the name are referred to as the spectral methods, while the methods with the term **modularity** are referred to as the spectral modularity methods. The SMM1 and SMM2 are the contributions of this thesis. The dots indicate whether the method is used for the specific DGP.

10.1. Performance Evaluation Criteria

The criteria with which we evaluate the performance of our clustering methods are based on the availability of a DGP-derived ground-truth partition, which we denote by ρ^*. We denote a partition that is obtained through a clustering method by $\hat{\rho}$. In the performance evaluation of this chapter, we use two criteria based on the recovery of information from ρ^*: one for the recovery of the ground-truth partition itself, and one for the recovery of ground-truth profiles.

In the experiments, we want to ensure that the synthetically generated data significantly contains relevant information representing the ground-truth partition; otherwise, evaluating the clustering performance can become ambiguous, as outlined in the introduction to Chapter 9. To be specific, we limit our interest to the performance of data sets where at least a partial recovery of the ground-truth partition is theoretically possible. This means that we limit the difficulty inducing parameters of the DGP such that the number of groups in the data is approximated correctly and corresponds to the number of detected spiked eigenvalues of the similarity matrix, as discussed in Section 3.4. By limiting the clustering difficulty, we ensure that not only the right number of clusters are constructed but also that all the eigenvalues used in the spectral modularity are informative, as they are not absorbed in the bulk. At the start of Section 10.2, we elaborate on this through the inspection of the effect of the number of detected spiked eigenvalues and the clustering performance.

In Figure 10.1, we schematically demonstrate a hypothetical scenario of clustering performances and the associated regime in which performance differences are meaningful. The colored lines represent the inverse performance of imaginative clustering methods as a function of clustering difficulty. The vertical axis represents a metric for inverse performance. This means that a lower value on the vertical axis is associated with better clustering performance. The horizontal axis represents a parameter that determines clustering difficulty, e.g., the group proximity parameter α or the mixing parameter β. The vertical dotted line represents the start of a phase transition, where to the right of this line datasets are theoretically too difficult to cluster. Any performance differences to the right of this vertical line are therefore not of interest. The horizontal dotted line represents the inverse performance of a trivial clustering of the data. To be specific, a clustering where all objects are together in a single group. This makes the use of any method with an inverse performance that is higher than this line worse than using no clustering algorithm at all. Any performance difference above this horizontal line is therefore also not of interest.

To further elaborate on the specific hypothetical scenario that is illustrated in Figure 10.1, we are able to unambiguously rank the methods from best to worst: black, red, and blue. This is despite the fact that there are regions where the red line is above the blue and black lines and regions where the black line is above the blue line. We neglect those regions because they occur in areas where the problems are theoretically too difficult or the clustering is already arbitrarily different from the ground-truth.

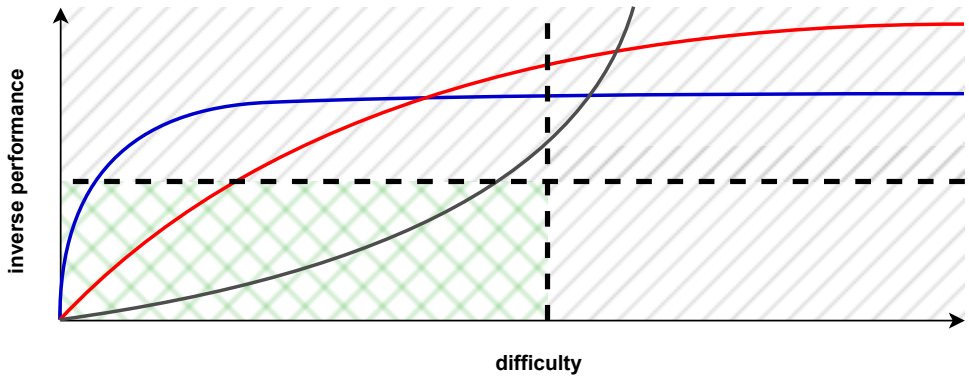


Figure 10.1: Illustrative diagram of an interesting regime. The colored curves represent different ways in which the inverse performance of clustering methods can depend on difficulty. The vertical axis represents the inverse performance. The horizontal axis represents a difficulty parameter (e.g., the group proximity parameter α or the mixing parameter β). The vertical dashed line represents a phase transition, where the existence of some ground-truth groups are theoretically no longer detectable. Clustering performance beyond that point is deemed not interesting, as the problem is theoretically too difficult. The horizontal dashed line represents a baseline level of variation of information obtained through a trivial singleton partition. Clustering performance above that line are deemed not interesting, as the clusterings are supposedly arbitrarily different from the ground-truth partition. The complement of these two regimes is represented by the light green area in the bottom left, which denotes the regime where performance differences are meaningful.

10.1.1. Partition recovery

For studying the clustering performance through the recovery of the ground-truth partition, we use a variation of information (VI) as discussed in Chapter 2. VI is a distance metric on the space of partitions and quantifies the discrepancy between two partitions. This gives us a measure of how much of the ground-truth partition is recovered by the clustering method. Specifically, we use the variation of information (VI) between partition ρ^* and clustering $\hat{\rho}$, i.e.

$$VI(\rho^*, \hat{\rho}). \tag{10.1}$$

The VI is an inverse performance measure because a higher VI represents a larger discrepancy between the two partitions, and a lower VI represents a smaller discrepancy. A VI of zero is obtained if and only if the partitions are exactly identical.

As a baseline for partition recovery, we compute the variation of information between the DGP-derived partition ρ^* and a particular trivial partition:

$$\rho_0 = \{\{1, 2, 3, \ldots, n\}\}. \tag{10.2}$$

This partition essentially mimics a clustering where all objects are clustered together, or equivalently, there is no group structure. If a method's clustering, $\hat{\rho}$, performs worse than ρ_0, then it is arguably better not to use the clustering, as we recover more from the original clustering by assuming there is no group structure. In the context of evaluating the partition recovery, the horizontal line in Figure 10.1 is represented by $VI(\rho^*, \rho_0)$.

10.1.2. Profile inference

Another way to quantify the performance of clustering methods is through their ability to recover group-representative data profiles from the data set. There are two reasons for evaluating the methods according to this criterion.

First, in settings with highly overlapping groups, strictly comparing the partitions directly can be problematic. Instead of strictly concerning segmentation of sets into well-separated components, like is done in clustering, the task of profile inference is rather aimed at the detection of the relevant and reoccurring profile patterns in the data set. Evaluating with profile inference is important in data sets where a strict segmentation of the data set is relatively ambiguous. In the synthetic data generation paradigm, the CMPM-based DGP satisfies this setting. Within this DGP, groups are generated such that their boundaries between groups are soft and objects are relatively close to multiple prototypes. With soft group boundaries, depending solely on the evaluation through misclusterings can become harmful as the clusterings become unstable due to the many objects that are close to the boundaries. Therefore, this evaluation criterion is important when considering CMPM-based DGPs.

Second, measuring the discrepancy between the soft-clustering and a ground-truth soft-partition directly is impossible because there is no trivial ground-truth. If we wish to evaluate the abilities of the soft-clustering extension that is introduced in Chapter 8, an evaluation through the use of a distance metric between two partitions is no longer suitable. Unlike in the evaluation of (hard-)partition recovery, there is no clear way to extract a ground-truth soft-partition from a DGP directly. However, it is tempting to believe that soft-clustering is capable of improving the clustering in one way or another. In particular, the inference of representative data profiles from soft clusterings is likely more accurate, as the inferred profiles are less influenced by weak cluster candidates that are on the boundary of multiple clusters. This demonstration of the benefit of soft partition is a secondary reason to evaluate the clustering methods using profile inference criteria.

In order to quantify the profile inference performance criteria of the clustering methods, we compute the distance between the representative data profile derived from the DGP ground-truth partition, denoted by $\boldsymbol{\mu}_k$, and the representative profile inferred from the clustering method, denoted by $\hat{\boldsymbol{\mu}}_k$, using the categorical mode as indicated in Equation 2.6.

For all $K = |\rho^*|$ ground-truth profiles, we find the closest inferred profile and compute the distances. The average of these distances is referred to as the profile precision. In addition, we introduce the profile recall; for all $|\hat{\rho}|$ inferred profiles, which may be different from $\hat{K}$, we find the closest ground-truth profile, compute the distances, and again take the average. Formally, we define profile precision and recall through the following definition:

$$\text{Precision} = \frac{1}{K}\sum_{k=1}^{K} \min_{h\in\{1,\dots,|\hat{\rho}|\}} s(\boldsymbol{\mu}_k, \hat{\boldsymbol{\mu}}_h) \text{ and } \text{Recall} = \frac{1}{|\hat{\rho}|}\sum_{h=1}^{|\hat{\rho}|} \min_{k\in\{1,\dots,K\}} s(\boldsymbol{\mu}_k, \hat{\boldsymbol{\mu}}_h). \tag{10.3}$$

The profile precision is indifferent to the situation where $|\hat{\rho}|$ is much larger than K. This undesirably prevents penalizing detected outliers, i.e., separated clusters with a single object. Solely evaluating profile precision can therefore be problematic in the case that the clustering hallucinates data profiles that are not representative of any group. In that case, this undesired behavior is not accounted for in the profile precision. For that reason, the profile recall is considered as well. While the precision quantifies how much of the inferred profiles is actually representative of the ground-truth profiles, the recall quantifies how much of the ground-truth profiles is recovered by the inferred profiles.

Precision and recall can be naturally computed for profiles in the original data space. However, the evaluation metrics can be applied for a second use case through the evaluation of the inference of the cluster representative spectral modularity vectors, as discussed in Chapter 7. While the inference of profiles in the data space evaluates the ability to recover relevant data patterns, the inference of spectral modularity vectors demonstrates the effective usage of the spectral modularity subspace by the methods. The focus on inference in the spectral modularity subspace highlights the performance of the clustering step, excluding the impact of the dimensionality reduction that precedes it. In this way, we separate these two concerns.

Let $\mathbf{z}_k \in \mathbb{R}^{\hat{K}-1}$ denote the group representative spectral modularity vectors from partition ρ^*. Furthermore, let $\hat{\mathbf{z}}_k \in \mathbb{R}^{\hat{K}-1}$ denote the cluster representative spectral modularity vectors from partition $\hat{\rho}$. Here, both the inferred vectors and the ground-truth vectors are computed by the following summation:

$$\mathbf{z}_k = \sum_{i \in C_k} \mathbf{r}_i \text{ for all } C_k \in \rho^* \qquad \text{and} \qquad \hat{\mathbf{z}}_k = \sum_{i \in C'_k} \mathbf{r}_i \text{ for all } C'_k \in \hat{\rho}, \tag{10.4}$$

which is identically defined as the group representative spectral modularity vectors that we considered in Chapter 7. Here, $\{\mathbf{r}_i\}_{i=1}^n$ are the spectral modularity vectors. Then, we are interested in the angular alignment of $\mathbf{z}_k$ and $\hat{\mathbf{z}}_h$. For this, we can use the cosine similarity as a similarity metric in the profile precision and recall defined above, i.e.

$$s(\mathbf{z}_k, \hat{\mathbf{z}}_h) = \frac{\hat{\mathbf{z}}_k \cdot \mathbf{z}_h}{||\hat{\mathbf{z}}_k|| \cdot ||\mathbf{z}_h||}. \tag{10.5}$$

Note that in the case of evaluating the profile inference of the soft-clustering methods, we multiply the objects in the summations in Equation 10.4 with the weighted memberships $\mathbf{P}_{ik}$ associated with the $n \times K$ soft-partition matrix $\mathbf{P}$ specified in Section 8.1. For the cluster representative spectral modularity vectors, we can shortly write the inferred profiles as the rows of the matrix $\mathbf{P}^\top \mathbf{R}$.

10.2. Partition Recovery in Gaussian Mixture Data

In our first experiment, we evaluate the methods for clustering data that is generated from a high dimensional Gaussian Mixture Model (GMM) based DGP. In order to vary the clustering difficulty of a particular synthetic data set, we vary three DGP parameters. We consider the effect of group proximity denoted by α, the number of groups denoted by K, and the number of dimensions denoted by p.

10.2.1. Effect of Group Proximity

First, we investigate the effect of the proximity of groups in the GMM data. Furthermore, to see how the performance of the clustering methods is reflected in the separation of the spiked eigenvalues and the bulk eigenvalues as discussed in Chapter 3, we also study the effect on the gap between the bulk and the spikes for varying group proximity, α. In doing this, we study values for the parameter α that are typically considered to be in the uninteresting regimes. In particular, relating the parameter α to the discussion of the illustrative difficulty axis in Figure 10.1, we make an exception to this philosophy to demonstrate the sufficiency of focusing on the solely interesting regime going forward.

Figure 10.2 illustrates the performance of the clustering methods in terms of partition recovery measured with VI as a function of group proximity α. The synthetic data sets of $n = 200$ objects and $p = 200$ features are generated for multiple values of K and a varying value of α. For 20 values of α, we generate 30 data sets for each.

In the top row of the figure, we see the mean inverse performance that is computed in terms of VI. Additionally, we display a horizontal dotted line to indicate the VI between the ground-truth partition and the trivial partition, ρ_0. This gives an indication of a baseline level of VI that is obtained through the trivial clustering of all objects in a single cluster. The vertical dashed line represents the start of the phase transition, where one of the spiked eigenvalues is absorbed in the bulk. To the right of the vertical dashed line, all the methods are unable to distinguish some of the spiked eigenvalues from the bulk eigenvalues. This is expected to quickly worsen the clustering performance of any method. Specifically, the intersection of the vertical line with the α axis indicates the smallest observed proximity value, where the eigenvalue bulk absorbs a spiked eigenvalue. When a spiked eigenvalue has been sufficiently merged with the bulk, it is theoretically impossible to cluster with the significant information that is contained in that corresponding eigenvector. This threshold aligns with the vertical dotted line specified in Figure 10.1.

In the middle row, we display the $\hat{K}$ that is approximated with shuffling based parallel analysis. The meaning of the dashed vertical line in the top row is explained well by the alignment on the horizontal axis with the leftmost point that is lower than K. This is equivalent to the smallest observed value for α where we encounter an occurrence of $\hat{K} \neq K$ in one of the generated synthetic data sets.

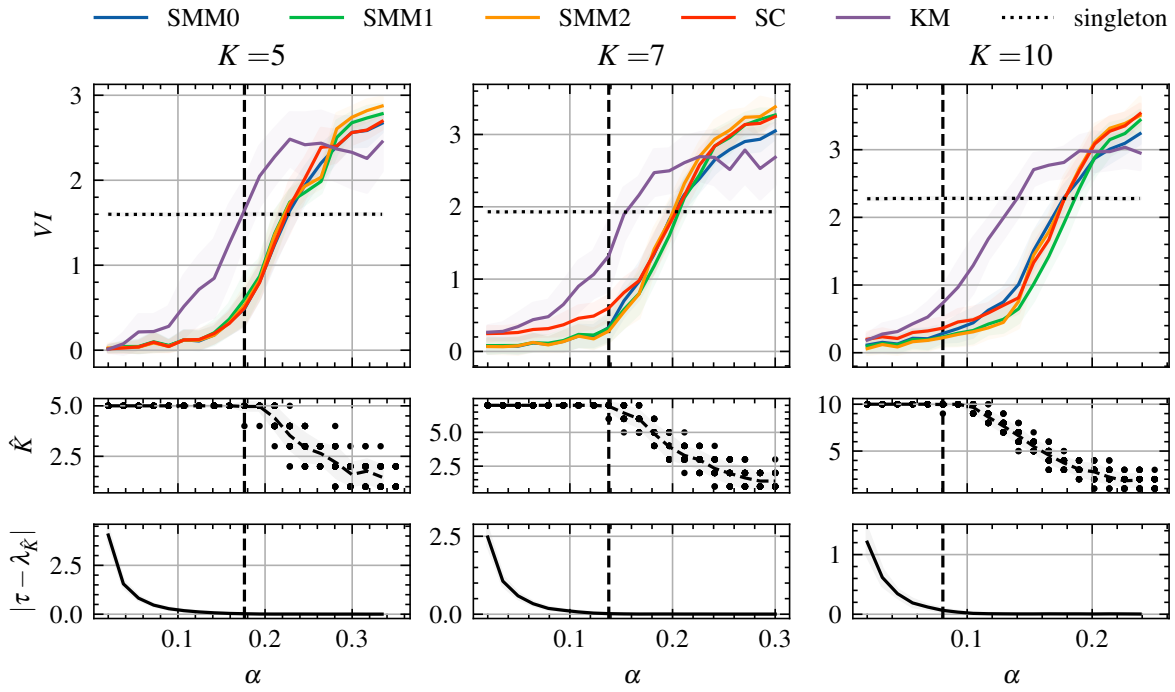


Figure 10.2: Effect of group proximity. In the top row, the inverse performance (VI) of the clustering methods is displayed for different values of K and varying α. The horizontal axis represents the group proximity parameter α, and the vertical axis represents the VI between the ground-truth partition and the clusterings obtained from the methods. The horizontal dotted line represents the VI between the ground-truth partition ρ^* and the trivial partition ρ. The vertical dashed line represents the start of the phase transition, where the bulk of the eigenvalues absorb one of the K spiked eigenvalues, indicating the start of the 'too-difficult-to-cluster' regime. The synthetic datasets have $n = 200, d = 200$. For each value of α, 20 datasets are generated that are used in the performance evaluation. Shaded areas denote one standard deviation around the means, which are represented by the solid lines. In the middle row, the number of spiked eigenvalues, $\hat{K}$, is displayed as a function of α. In the bottom row, the difference between the computed eigenvalue threshold τ and the $\hat{K}$th eigenvalue is displayed as a function of α.

The bottom row of the figure shows the distance between the shuffling based parallel analysis threshold τ and the smallest eigenvalue that is outside the bulk $\lambda_{\hat{K}}$. As is expected, the gap $|\tau - \lambda_{\hat{K}}|$ is already relatively small as α approaches the intersection with a vertical dashed line from the left. This observation aligns with our expectation of a demonstration of the strict detect-ability phase transition that is discussed in Chapter 3.

Clearly, the performance of clustering worsens rapidly when more and more spiked eigenvalues are absorbed into the bulk. Essentially, this means that the ground-truth partition of the synthetically generated data sets is theoretically impossible to recover. The evaluation of the clustering methods in such settings may therefore not be representative. Therefore, with the philosophy described in Section 10.1, the remainder of the figures in this chapter consider the interesting regime. This means that if we synthetically generate a data set with K groups, the number of detected spiked eigenvalues $\hat{K}$ of the similarity matrix satisfies $\hat{K} = K$.

Furthermore, regarding the displayed clustering performances, we observe a relatively large performance difference between the spectral methods (SC, SMM0, SMM1, SMM2) and KMeans. This corresponds with our expectation that the KMeans method fails in high dimensional ($p = 200$) setting. KMeans appears to be surpassing the remaining methods for high levels of α; however, this is the regime where the clustering with a trivial clustering would be significantly better than any of the suggested methods. Additionally, when focusing on the interesting regime alone, i.e., left to the dashed line and below the dotted line, we observe that among the spectral methods, the competitor method (SC) is slightly outperformed by the spectral modularity methods.

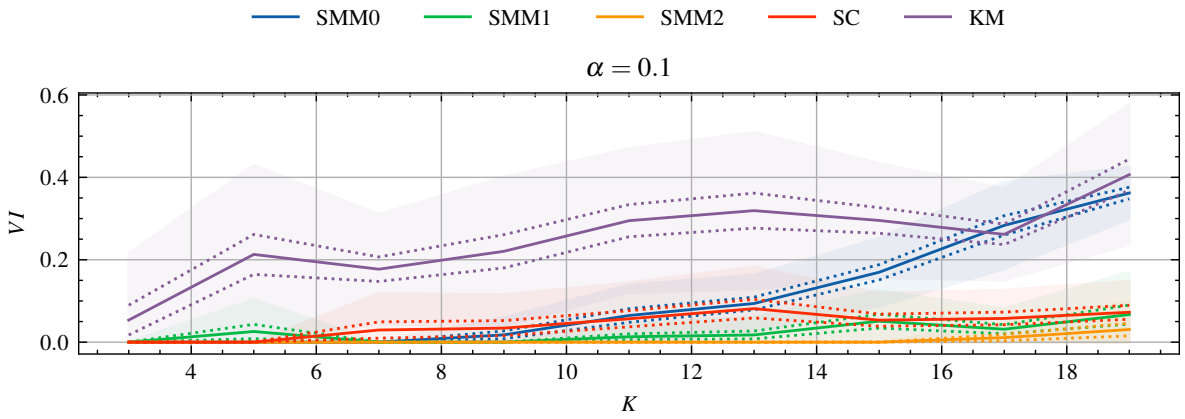


Figure 10.3: Effect of the number of groups. The vertical axis represents the variation of information between clusterings and the DGP-derived ground-truth partition. The horizontal axis represents a varying number of groups K. The number of objects is $n = 30 \cdot K$ each with $p = 200$ dimensions. For each value of K, 20 synthetic data sets are generated. The solid lines are the means. The dotted lines represent the standard error of the means. The shaded parts represent error, the standard deviations.

10.2.2. Effect of Number of Groups

From the values for K that are experimentally evaluated in Figure 10.2, we do not perceive a significantly deteriorating performance of SMM0 compared to the other spectral (modularity) methods. However, we know from Chapter 5 that when K is large, SMM0 fails to perform well. For example, in Figure 5.2, we clearly observe spectral modularity breakdown for $K = 18$ in the exactly identical setting of GMM data. For this reason, we study the effect of varying K on the clustering performance.

Figure 10.3 illustrates the VI between the ground-truth partition, ρ^*, and the clusterings obtained from the clustering methods, $\hat{\rho}$, for datasets generated from a GMM for a varying number of groups $K \in \{3, 5, \ldots, 17, 19\}$. For each value of K, we generate 20 datasets for which we perform the evaluation. The solid lines are the means of the computed VI's at that specific value for K, the dotted lines are the standard error of the mean, and the shaded areas display the standard deviations. The horizontal axis shows the number of groups, and the vertical axis depicts the VI. The value for $\alpha = 0.1$ is chosen such that the group centers are not too far apart such that the clustering becomes too easy, and not too close such that the clustering becomes too difficult. This means all the considered synthetic data sets belong to the interesting regime. The number of objects is set to be in a fixed ratio with K, i.e., $n = 30K$. This ensures that the diminishing representative power of groups, due to a limited number of objects within the group, i.e., due to shrinking n/K, is not mistaken for the combinatorial saturation discussed in Chapter 5. The latter is specifically caused by the absolute increase in the number of groups, independent of the ratio n/K; therefore, the experimental demonstration should reflect that.

In the figure, we observe the breakdown of spectral modularity through studying the performance differences of naive spectral modularity maximization (SMM0) and spectral clustering (SC). Additionally, we see that the breakdown is resolved in the contributed spectral modularity enhancements (SMM1, SMM2). Because in all tested cases of Figure 10.3, the clustering difficulty is relatively easy, the diminishing performance of SMM0 is purely due to the increasing of K and not from any difficulties with detecting spiked eigenvalues.

Furthermore, the effect of the number of groups appears to be slightly negatively influencing the performance of SC, while SMM2 and SMM1 are capable of highly accurate clustering for all values of K. This potentially points towards a structural improvement in the clustering of SMM1 and SMM2 over SC. The performance difference may be unrelated to the associated spectral modularity breakdown because the SC method used here always constructs clusterings with the right number of clusters, according to the number of spiked eigenvalues, in the same way that SMM1 and SMM2 do.

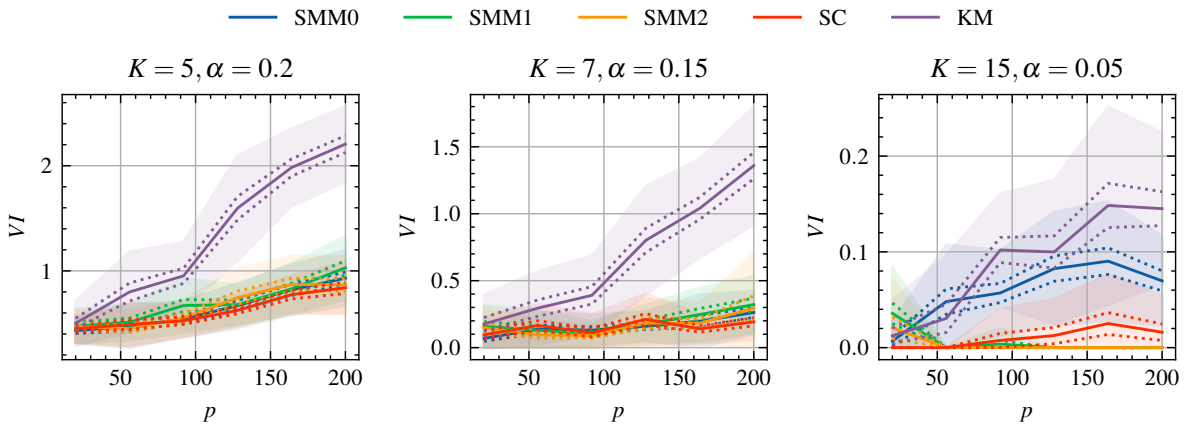


Figure 10.4: Effect of dimensionality The vertical axis represents the variation of information between clusterings and the DGP-derived ground-truth partition. The horizontal axis represents varying p. The shaded area is one standard deviation around the mean (clipped at zero), and the dotted line is one standard error of the mean.

10.2.3. Effect of Dimensionality

The benefit of spectral, including spectral modularity, methods in high dimensional data is due to the implicit dimension reduction that is employed. On the other hand, the KMeans method is known to be problematic with high dimensional data. Therefore, we investigate the effect of the dimensionality.

In Figure 10.4, we see the effect of the number of dimensions p on the clustering performances. The vertical axis shows the variation of information between the clusterings and the DGP ground-truth partition. The horizontal axis shows the dimensionality p. The values for α are again chosen such that the Kth eigenvalue is close enough to the threshold, such that it exhibits some level of clustering difficulty but is not theoretically impossible to cluster.

The performance of KMeans, which is based on distances in the data space, is increasingly worsening as the number of dimensions grows. At the same time, the spectral methods are much more robust against growing dimensions. Furthermore, we verify in the right panel of Figure 10.4 that the spectral modularity breakdown, indicated by the worse performance of SMM0 than the other spectral methods, happens independently of the number of dimensions. In particular, when K is high, e.g., $K = 15$, for a large range of values for p, the clusterings obtained with SMM0 have a higher VI with the ground-truth partition. In addition, there even appears to be a range of relatively small values of p, around $p = 50$, where KMeans performs slightly better than SMM0. Although the performance gap in this setting is relatively small, it aligns with the understanding that SMM0 suffers from high values for K due to the spectral modularity breakdown but is robust to high p, while KMeans suffers from high values for p due to the curse of dimensionality and is robust to high K.

10.3. Partition Recovery in Categorical Mixed Prototype Data

In the second experiment, we assess the partition recovery performance of the clustering methods using data generated from the categorical mixed prototype model. In order to gain additional depth in the performance evaluation, we briefly deviate from the study of average performances through the study of the entire statistical distribution of partition recovery performances for a selection of DGP parameterizations. After this, we return to varying the clustering difficulty of synthetic data sets. We vary three DGP parameters. We consider the amount of mixing denoted by β, the number of groups denoted by K, and the group size heterogeneity denoted by η.

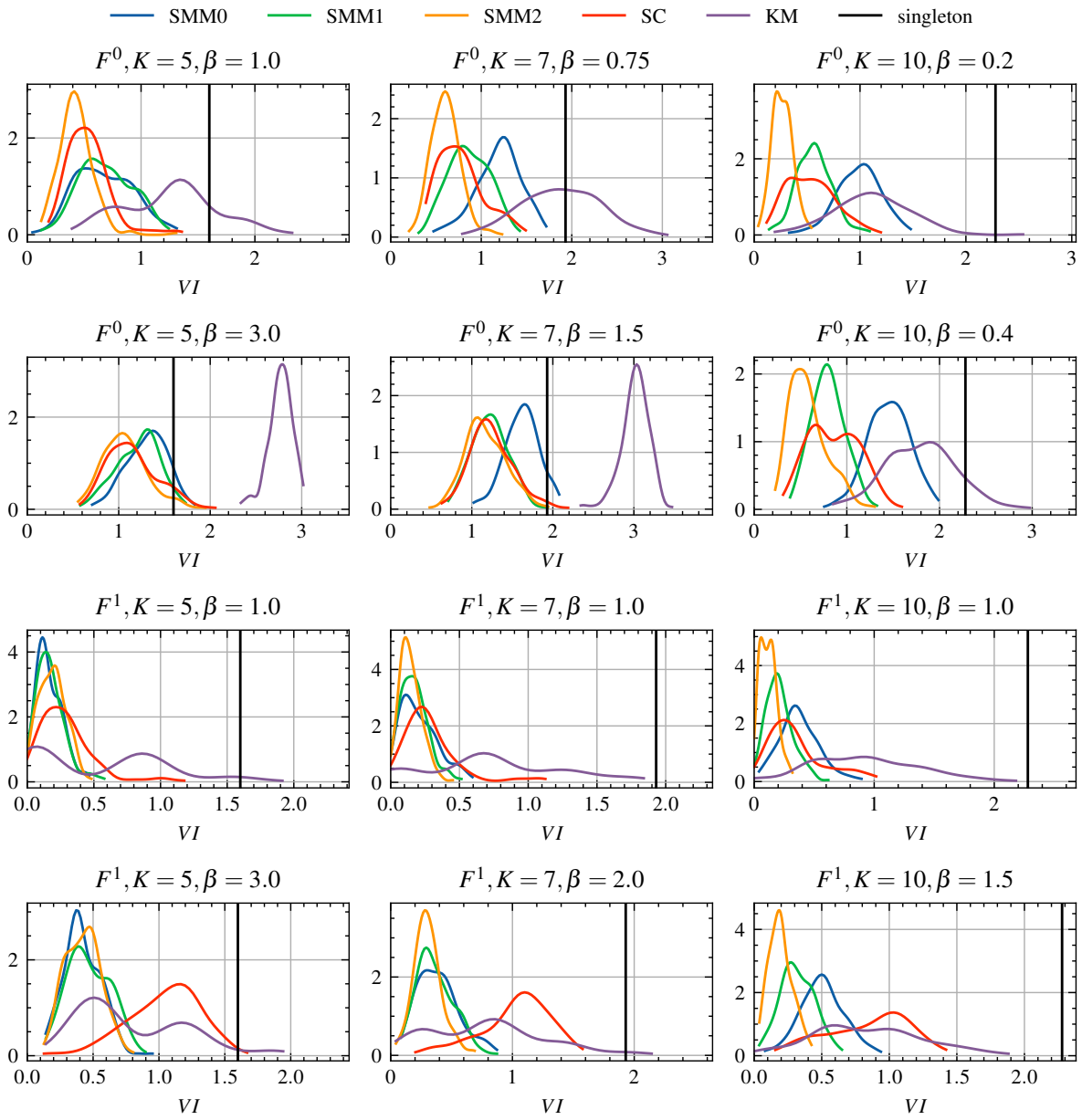


Figure 10.5: Partition recovery performance. The smoothed density histogram is derived from the variation of information between clusterings and the DGP derived ground-truth. 200 datasets are used in the histogram. The generated data sets have $n = 200$ and $p = 200$, group proximity $\alpha = 0.1$, group size heterogeneity $\eta = 0$, and prototype diffusion $\gamma = 0.05$. F^0 represents the Dirichlet weight distribution, and F^1 represents the Logit-χ^2 weight distribution. The smoothed histogram is obtained through kernel density estimation with Scott's rule [91].

10.3.1. Statistical Distribution of Performance

In the depiction of the clustering performances so far, we have only considered performance statistics, such as averages, standard deviations, and standard errors of the mean. However, the wide bands that are representative of standard deviations of the distribution of clustering performances and standard errors of the mean do not display the shape of the distribution of the clustering performances. These statistical abstractions may incorrectly suggest unimodal distributions of clustering performances. Therefore, before we focus on the effects of the parameter, we inspect the entire statistical distribution of clustering performance for a selection of parameterization to ensure that our conclusions based on the average performances are meaningful.

In Figure 10.5, we see histograms of the variation of information between the clusterings found by methods and ground-truth. For visibility, the smooth histograms are displayed with a kernel density estimation, where the bandwidth is determined using Scott's Rule [91]. The vertical axis represents the normalized frequency distribution of observed clustering performances with respect to the VI that is indicated on the horizontal axis. The clusterings are performed on synthetic CMPM data sets with different values of K and β. We display a histogram for both the Dirichlet weight distributions (F^0) and the logit-χ^2 weight distribution (F^1), both defined in Chapter 9. The vertical solid line represents the VI between the ground-truth and the trivial singleton.

In the figure, we see a significant performance overlap between spectral clustering (SC) and spectral modularity methods (SMM0, SMM1, and SMM2). Despite this overlap, SMM2 consistently appears to be the top performer. This conclusion is based on two observations. First, the peak of the statistical distribution of SMM2 is notably lower compared to other methods, indicating a better average performance. Second, the width of SMM2's histogram is relatively narrow, indicating a stable performance.

Furthermore, we observe diminishing performance of SMM0 in the Dirichlet weight distribution (F^0) as the number of groups is larger, hinting at the breakdown of the naive spectral modularity maximization. While SMM0's performance overlaps with other methods for $K = 5$, it significantly underperforms for $K = 7$ and $K = 10$. SMM1 and SMM2 consistently reduce the variation of information, making them competitive with spectral clustering. In some cases, SMM1 and SMM2 slightly outperform SC, as depicted in the upper right of Figure 10.5. Among the suggested solutions, SMM2 exhibits better average performance, especially for this specific data generation process. This is likely because the calibration step in SMM1 is relatively unstable, rendering SMM2 a more favorable choice.

In the F^1 weight distribution, depicted in the second row of Figure 10.5, SC is significantly surpassed by the spectral modularity methods. Notably, even the naive method, SMM0, demonstrates superior performances, especially with high values of β. This partially suggests that SMM0 demonstrates a lesser susceptibility to spectral modularity breakdown in F^1 weights, which we will further discuss upon studying the variation of K. Furthermore, the concentrated shape of the F^1 weight distribution highlights a flaw in the SC method. To be specific, the non-convexity of the weights is problematic for the KMeans clustering step that is employed in the Laplacian embedding.

Finally, now in a setting of categorical data, we again observe the comparatively poor performance of the baseline based method, i.e., KMedoids, suggesting its unsuitability for clustering data from the CMPM-based DGP. In addition, we observe that the performance of KMedoids is relatively unstable with the F^1 weight distribution, as indicated by the wide and multi-modal distribution of VI. This instability is due to the incidental presence or absence of good representative objects in the generated dataset, which is crucial for the performance of KMedoids, as discussed in Chapter 2. This highlights a fundamental flaw of the KMedoids method. In contrast, our contributed methods, SMM1 and SMM2, not only typically construct higher quality partitions, but they are also more stable.

10.3.2. Effect of Prototype Mixing

Now that we have seen from the statistical distribution of the clustering performances that most methods, except for KMedoids, exhibit a unimodal shape, we again limit the study of the clustering performances to the performance mean, the standard error of the mean (SEM), and the standard deviation.

In Figure 10.6, we extend our observations from studying the histograms by employing the clustering performance evaluation for varying beta for a selection of values of K. The group proximity α is 0.1, and the prototype diffusion γ is 0.05. The vertical axis represents the VI. The horizontal axis represents the prototype mixing parameter β. We demonstrate the SEM of the VI with the dotted lines and one standard deviation around the mean with shaded areas.

The insights obtained from studying the histogram roughly extend to a larger range of values for the mixing parameter β. The KMedoids method fails to compete with the other methods in almost all cases. The performance of SMM0 is poor when the number of groups is high ($K = 10$) and the weights are Dirichlet distributed (F^0). However, SMM1 and SMM2 correctly address this issue and even surpass (in the case of SMM2) the competitive method (SC).

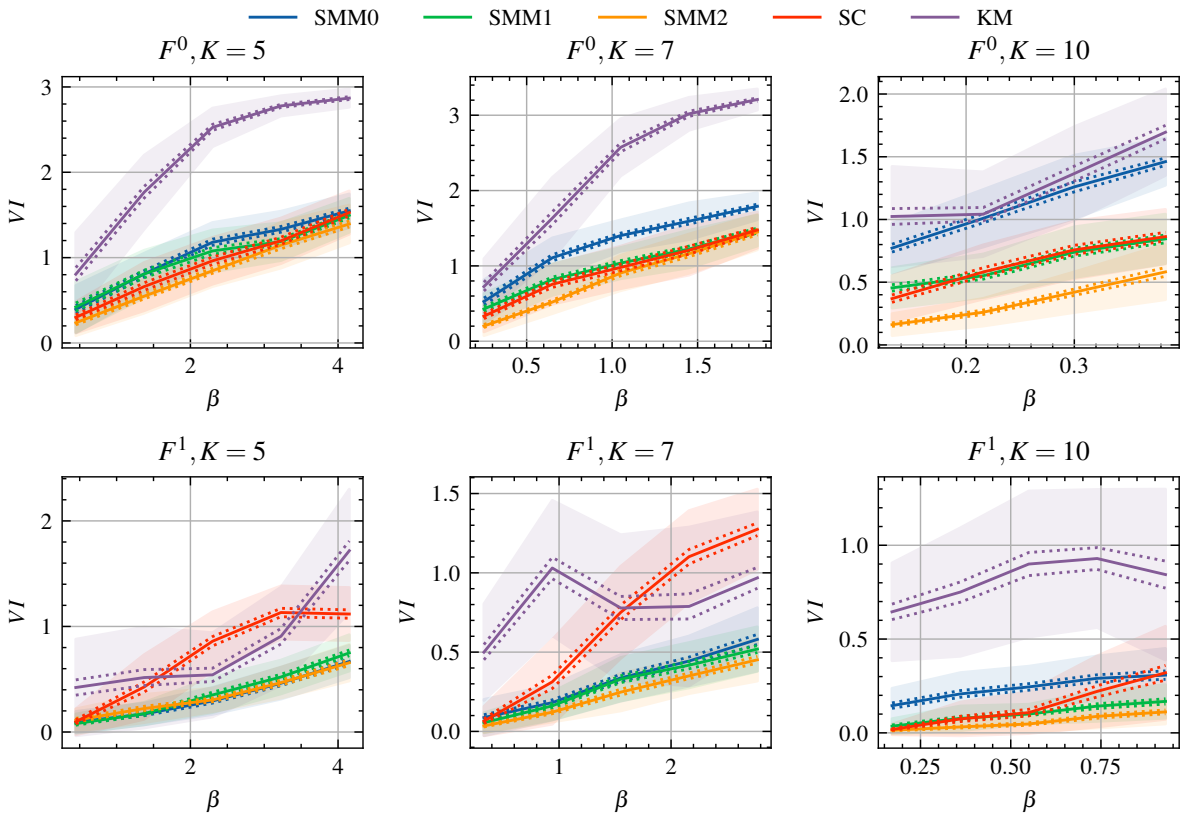


Figure 10.6: Effect of prototype mixing. The vertical axis represents the VI between the clustering and the ground-truth partition. The horizontal axis represents varying prototype mixing parameter β. For each combination of K and weight distribution, 5 equally separated values for β are chosen, with 40 generated datasets for each. The generated data sets have $n = 200$ and $p = 200$, group proximity $\alpha = 0.1$, group size heterogeneity $\eta = 0$, and prototype diffusion $\gamma = 0.05$.

Furthermore, we again see a drastic performance decrease for the SC in the logit χ^2 distributed (F^1) weights. This shows how the fundamental flaw of the SC method becomes more severe as the mixing amount β increases. As β is higher, the shape of the weight distribution is more concentrated, as indicated in Figure 9.3. However, unlike in the F^0 distributed weights, the concentration shape is not convex, as indicated in Figure 9.2. Therefore, for high β weights that are F^1 distributed, data projected onto the Laplacian subspace itself is not convex and not linearly separable. Combining this with the fact that the SC method uses a KMeans cluster step in the Laplacian subspace, the bad performance of SC in that setting can be well explained.

10.3.3. Effect of Number of Groups

Unlike in the setting of GMM-based data, in the CMPM-based data, we already observed spectral modularity breakdown through the deteriorating performance of SMM0 for $K = 7$, especially considering the F^0 weights. This is likely due to the relatively soft cluster boundaries that are present in the CMPM-based data, making the effect of combinatorial saturation more severe.

In Figure 10.7, we see the effect of varying the number of groups on the clustering performance. We do this by specifying a relatively simple setting with a prototype mixing amount that is not too high and not too low, i.e., $\beta = 1$, and adjusting the number of objects in the data set to be in a fixed ratio with K, i.e., $n = 30K$. This again ensures that the demonstration of spectral modularity breakdown is relatively isolated from other trivial sources of clustering difficulty.

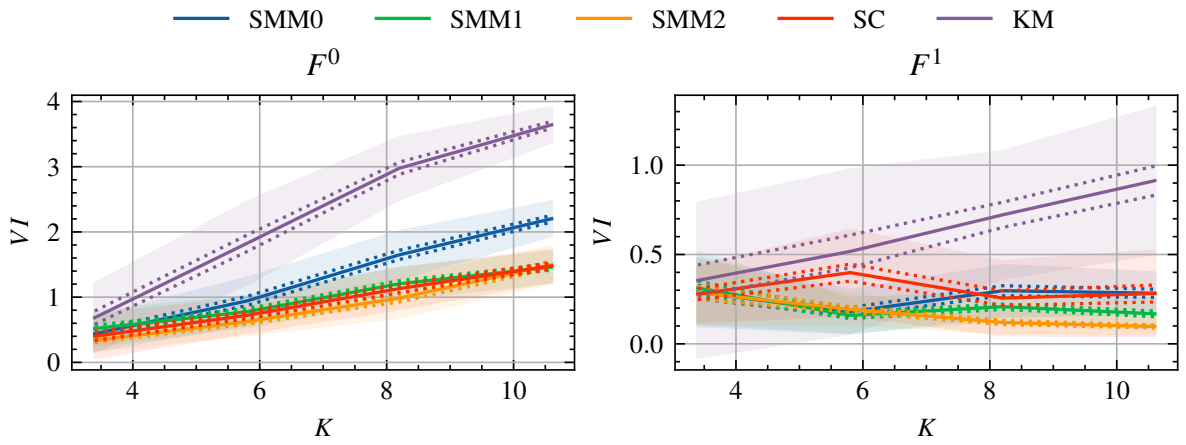


Figure 10.7: Effect of number of groups. The vertical axis represents the VI between the clusterings and the DGP-derived ground-truth partition. The horizontal axis represents the varying number of groups K. For both weight distributions, 5 equally separated values for K are chosen, for which 40 datasets are generated. The generated data sets have $n = 30 \cdot K$ and $p = 200$, prototype mixing $\beta = 1$, group proximity $\alpha = 0.1$, group size heterogeneity $\eta = 0$, and prototype diffusion $\gamma = 0.05$.

The observed clustering performance is in line with the understanding of the breakdown that happens in naive spectral modularity maximization (SMM0). With Dirichlet-distributed weights, the performance of SMM0 degrades more rapidly than that of the competitor method, spectral clustering (SC). However, SMM1 and SMM2 mitigate the issue and make spectral modularity based clustering perform competitively with SC.

For the F^1 weight distribution, the breakdown of SMM0 is less noticeable for the range of tested values. This aligns with our observation in the histogram of Figure 10.5. In the figure, it is difficult to see the breakdown because we can no longer compare SMM0 to SC as the latter method has difficulty with the F^1 distribution. Instead, we do see a slightly increasing improvement of our contributed solutions, SMM1 and SMM2, over SMM0, which is indicative of the spectral modularity breakdown.

The increased resilience that appears when generating data with F^1 distributed weights is due to the increased presence of objects that resemble a single prototype. Conversely, in F^0, as the mixing amount increases, objects only become more mixed, and contributions become less focused. This demonstrates that the type of entanglement of group boundaries, which contrasts F^1 and F^0 weights, affects the spectral modularity breakdown. Although SMM0 shows reduced susceptibility to spectral modularity breakdown in this setting, it is probable that breakdown may occur as K increases further, akin to occurrences in GMM, where the breakdown only occurs at a much higher value of K.

10.3.4. Effect of Group Size Heterogeneity

Up to this point, we have only studied the clustering performance of the data sets with ground-truth partitions that have roughly symmetrically sized groups. However, because the proposed spectral modularity breakdown solutions, SMM1 and SMM2, specifically tackle the breakdown by removing bias towards clusterings with heterogeneously sized groups, we investigate to what extent these solutions are suitable when the ground-truth partitions actually contain heterogeneously sized groups.

In Figure 10.8, we demonstrate the effect of group size heterogeneity. The horizontal axis represents the heterogeneity of the group sizes, where $\eta = 0$ represents completely homogeneously sized groups. The vertical axis shows the variation of information.

In general, we see that as η grows, the clustering tends to become more difficult. In contrast to previous observations, the performance of SMM0 appears to be relatively robust for varying this parameter compared to the other methods. Even to the extent that the naive method starts to perform better on average than SC and SMM2 for very heterogeneous sizes.

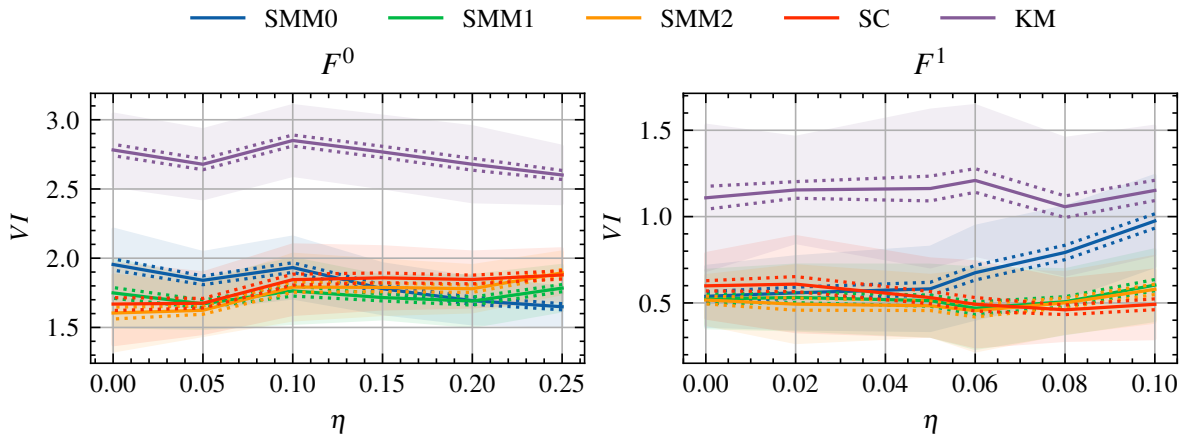


Figure 10.8: Effect of group size heterogeneity η in categorical mixed prototype data. The vertical axis represents the VI between clusterings and the DGP-derived ground-truth partition. The horizontal axis represents the group size heterogeneity η. For both weight distributions, 5 equally separated values for η are chosen, for which 40 datasets are generated. The data is generated with a CMPM based DGP with $n = 200$ objects each with $p = 200$ dimensions, $K = 7$ clusters, prototype mixing $\beta = 1$, group proximity $\alpha = 0.1$, and prototype diffusion $\gamma = 0.05$.

This indeed illustrates a potential flaw of the two suggested solutions, SMM1 and SMM2. Both solutions resolve the spectral modularity breakdown by combating the bias towards clusterings with heterogeneously sized and, typically, inconsistently merged groups. Therefore, if there are actual heterogeneously sized groups present in the ground-truth partition, their presence can be inconsistently penalized in the clustering. However, this behavior is only expected to occur for extremely heterogeneous group sizes. The performance difference between the naive methods and the resolved methods is not significant. Therefore, there is yet no reason to believe that it is better to use the SMM0 over SMM1 or SMM2.

On the other hand, when looking at F^1 distributed weights in the right panel of Figure 10.8, there appears to be an opposite effect. In this setting, SMM0 is significantly worse at clustering data with heterogeneously sized groups, as an effect of a more significant spectral modularity breakdown. Specifically, as the group size heterogeneity is large, the proportion of relatively small groups is large. In small groups, the existence of representative objects that are represented by the F^1 distributed weights is therefore less likely. This way, for heterogeneous group sizes and F^1 weights, there are underrepresented groups, making it much more likely that the naive maximization of spectral modularity will lead to inconsistent merges. Therefore, while the spectral modularity breakdown is generally less severe for datasets with F^1 distributed weights, if group sizes of the ground-truth partition are heterogeneously sized, the spectral modularity breakdown is considerably more severe.

10.4. Profile Inference in Categorical Mixed Prototype Data

In this third experiment, we evaluate the clustering methods according to their profile inference abilities in categorical mixed prototype data. We also investigate the added value of using soft clustering in this context. In particular, we study profile precision and profile recall in both the data space and in the spectral modularity space, as discussed in Section 10.1, for the methods specified in Table 10.1 and the associated soft variants by employing the procedure of Chapter 8. This gives us a total of four quantification for the profile inference performance, which we evaluate on ten (soft) clustering methods.

For this reason, apart from a general comparison of performance methods among themselves, there are three additional comparisons that are worthwhile to consider. First, the performance of a method in terms of profile precision can be compared with the profile recall. Second, the performance of the soft-clustering variants can be compared to the original, non-soft, clustering methods. Third, the performances of profile inference in data space can be compared with the performances of profile inference in terms of the spectral modularity vectors. To this end, we first study the statistical distribution of these quantities, after which we study the effect of the prototype mixing β on profile inference.

10.4.1. Statistical Distribution of Profile Inference Performance

In the same philosophy as Figure 10.5, we first study the statistical distributions of the profile inference performance. The shapes of the distributions indicate whether we can again further limit the study to averages of the performance metrics.

In Figure 10.9a, we display the statistical distribution of the data space profile inference performance, and in Figure 10.9b, we display the statistical distribution of the profile inference with spectral modularity vectors. The clusterings are performed on synthetic data generated with the CMPM-based DGP with the group proximity α set to 0.1 and prototype diffusion γ set to 0.05, a setting that is identical to the one studied in Figure 10.5. For brevity, we omit the study of data sets with F^1 distributed weights. However, these results are included in studying the effect of prototype mixing. The vertical axis represents the frequency of the observed profile precision and recall, which are indicated by the horizontal axis in the upper and lower rows, respectively.

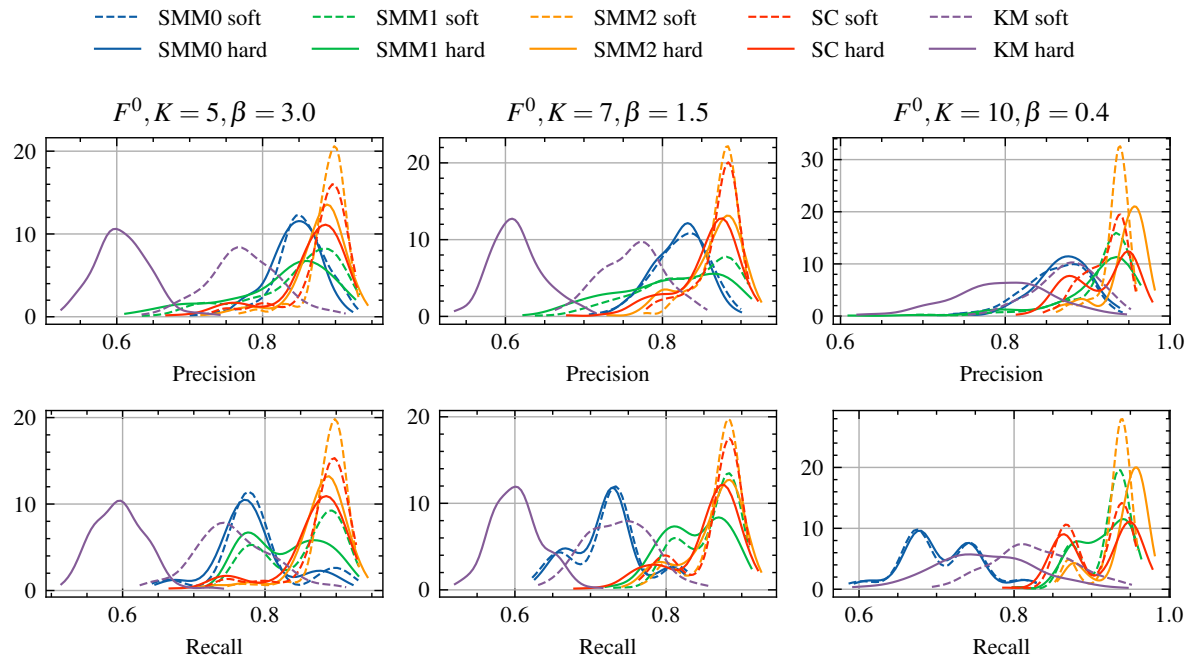
Comparing the shapes of statistical distributions of the profile precision and the profile recall in both Figure 10.9a and Figure 10.9b, we find that most of the peaks of the distributions are located at the same places and the widths of the distributions are roughly the same. An exception to this, however, is the SMM0 method. The SMM0 method performs significantly worse in terms of profile recall than in terms of profile precision. This is in agreement with the theory of spectral modularity breakdown because the method generally finds less than the ground-truth K representative profiles due to merging clusters inconsistently leading to clusterings with fewer than K clusters. This causes the few profiles it finds to not be representative of the ground-truth profiles. The severity of the breakdown is particularly observed when $K = 10$. Here, we see that the KMedoids method, which is considered to be unsuitable for the CMPM data setting, even performs better than SMM0 in terms of profile recall.

Furthermore, from both figures, we find that the soft variants, depicted by the dashed lines, of all methods generally have a higher or equal peak of the profile precision and recall histograms. The most notable improvement is found in the KMedoids method. The softening procedure applied to KMedoids clusterings improves the profile inference significantly; however, it is still the worst-performing method in most cases. Although in all displays of statistical distributions the effect can be seen, it is most prominently seen in the data space profile inference depicted in the two leftmost columns of Figure 10.9a. The large improvement found in soft-clustering of KMedoids is expected, as the hard clustering obtained with KMedoids is restrained by the existence of representative objects within a dataset. However, the soft clustering removes the discreteness of this restriction by allowing the inferred profiles to be a combination of objects. Nevertheless, the performance of softened KMedoids is still among the worst performing methods.

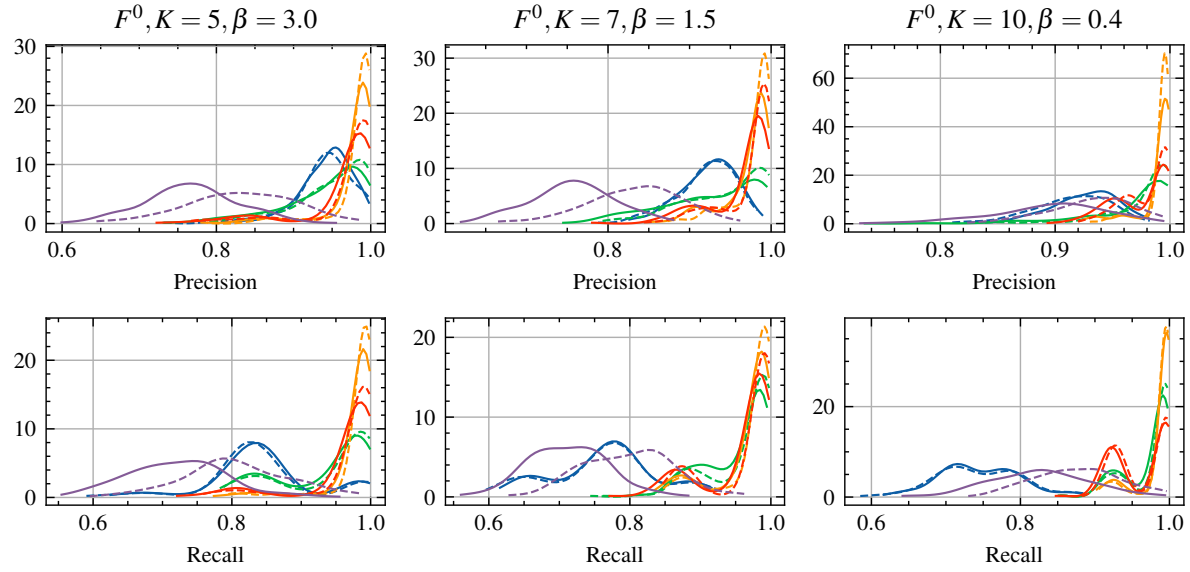
On the other hand, an exception to soft-clustering being a strict improvement over the original clustering method, can be found in the data space profile inference of SMM2. Especially when K is high, the peak of the soft variant of SMM2 is positioned at a slightly lower level of precision and recall. However, the soft-clustering variant of SMM2 stabilizes the profile inference performance, resulting in a slightly narrower distribution.

Considering the comparison of data space profile inference and spectral modularity profile inference, we find that, a similar ordering of the performances of the methods is obtained. However, a slight difference is that the soft clustering of SMM2 can be considered a strict improvement in the spectral modularity profile inference depicted in Figure 10.9b for all combinations of K and β. This is unlike in the data space profile inference, where there is a trade-off for $K = 10$ between the stability of the performance of SMM2 and the location of the peak, as seen in Figure 10.9a.

Overall, examining the ordering of the performance of the clustering methods in terms of the different profile inference performance quantities yields quite some overlap. Therefore, it is relatively clear that the best performing methods are SMM2 and its soft variant, as indicated by the location and width of the peaks of the histogram. Furthermore, specifically, SMM1 demonstrates unstable profile inference performance in both the data space and the spectral modularity space. This is indicated by the wide and multi-modal shapes of the distributions associated with the performance. In many cases, the SMM1 performs significantly worse than the competitor method SC when $K = 5$ or $K = 7$.



(a) Profile precision and recall in data space.



(b) Profile precision and recall with spectral modularity vectors.

Figure 10.9: Profile precision and recall The smoothed density histogram is obtained through kernel density estimation with Scott's rule [91]. 200 datasets are used in the histogram. The generated CMPM data sets have $n = 200$ and $p = 200$, group proximity $\alpha = 0.1$, group size heterogeneity $\eta = 0$, prototype diffusion $\gamma = 0.05$, and have F^0 distributed weights.

10.4.2. Effect of Prototype Mixing

Now that we have an intuition of the statistical distributions of the profile inference performance, we study the effect of prototype mixing on the profile inference. To do this, we consider the same setting as in Figure 10.6, where we evaluate the effect of prototype mixing on the partition recovery performance. Furthermore, we do include F^1 distributed weights, which have been omitted during the demonstration of the statistical distributions of profile inference performance.

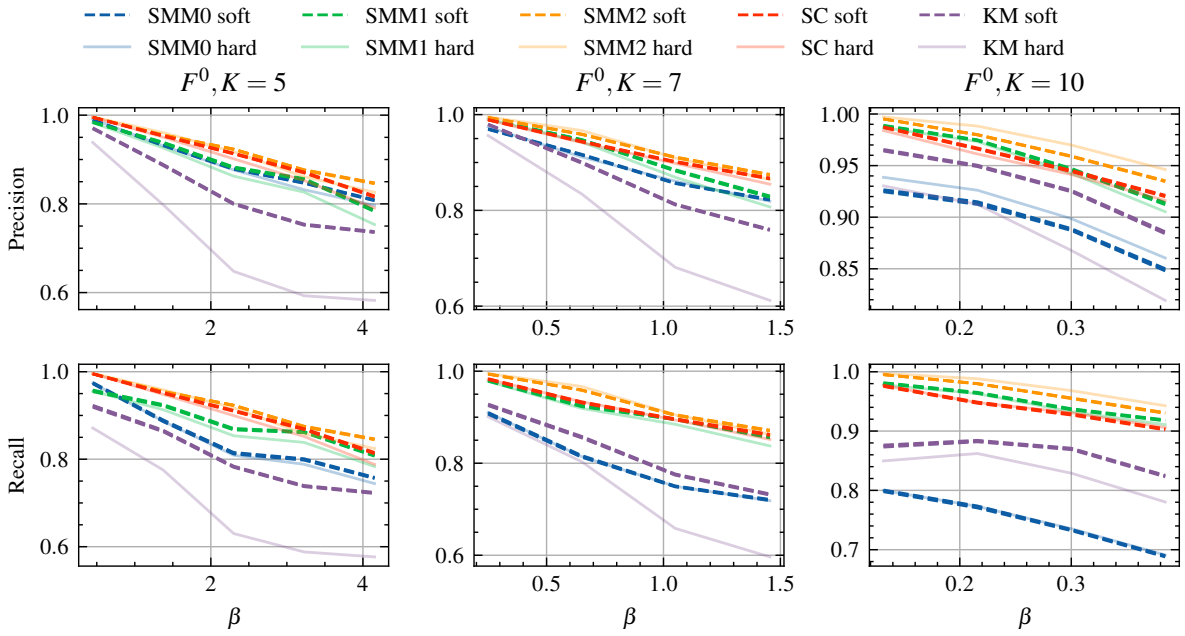
In Figure 10.10a we see the profile precision and recall for a varying prototype mixing amount β and Dirichlet distributed weights, F^0, and in Figure 10.11a we see the same type of results for data sets generated with the logit χ^2 weights, F^1. Furthermore, in Figure 10.10b we see the profile precision and recall in terms of the spectral modularity vectors for a varying prototype mixing amount β, and in Figure 10.11b we see the same type of results for F^1 weights. In all of the above figures, the vertical axes represent precision and recall, for the top row and bottom row, respectively, and the horizontal axis represents the prototype mixing β. The standard error of the mean for the soft clustering based profile inference performance is displayed by the width of the dashed lines. For visibility, the hard clustering based profile inference performance is made slightly opaque, and the standard error of the mean of the hard clustering is removed. From the study of the statistical distribution of the performances, we find that the soft variants typically perform better anyway. For reference, the statistical distributions that are discussed above in Figure 10.9a represent the same identical experimental setting but are limited to a small selection of values for β.

The most general observation is that as the prototype mixing increases, the profile precision and recall decrease, indicating that profile inference becomes more complicated. This is in agreement with the fact that partition recovery becomes more difficult as prototype mixing increases. The resemblance of the performance in terms of profile inference and partition recovery can be seen in a specific example that is easily recognizable in the settings with F^1 distributed weights. To be precise, consider the relatively abnormal shape of the performance of the KMedoids method in Figure 10.11a and Figure 10.11a. A similar shape can be recognized in the partition recovery performance of KMedoids on the identical data sets in Figure 10.6, where the performance curve is flipped vertically, because VI is an inverse performance metric.

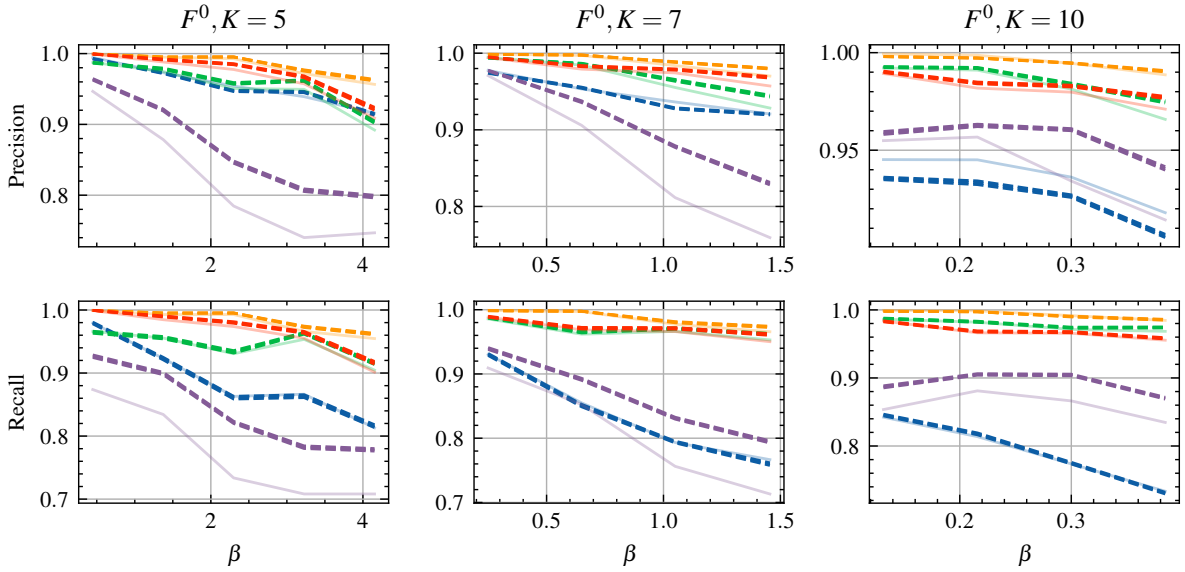
On the other hand, the difference between measuring performance through profile inference and partition recovery can also be demonstrated with an example in the setting of F^1 weights. In particular, whereas the partition recovery performance of SC is significantly worse for F^1 weights, as depicted in Figure 10.6, the discrepancy between SC and the other spectral methods is less noticeable for the profile inference performance depicted in Figure 10.11a and Figure 10.11b, especially considering the spectral modularity profile inference. However, despite the smaller performance gap, SMM1 and especially SMM2 still clearly exhibit superior performance in terms of profile inference for F^1 weights over SC.

Comparing the profile precision and profile recall of the methods, we again find that these quantities are significantly different for SMM0. In particular, in Figure 10.10a and in Figure 10.10b, we see that in the middle column, representing $K = 7$, the ordering of the performance of the soft SMM0 method and the soft KMedoids methods is different for profile recall and profile precision. This is in agreement with the comparison of the quantities done in the discussion of Figure 10.5. A second demonstration of the breakdown that is hard to observe in the statistical distribution is the clear outperformance of SMM0 by the soft variant of KMedoids in terms of both data space profile recall and precision, as depicted in Figure 10.10a.

However, for F^1 distributed weights, the deteriorating profile recall in SMM0 is not observed, as depicted in Figure 10.11a for data space profile inference and in Figure 10.11b. This underlines the intuition that we previously used to explain the deteriorating profile recall in studying the statistical distributions. In fact, the profile recall specifically suffers from spectral modularity breakdown, as we know from studying the effect of the number of groups K in Figure 10.7 that the breakdown is considerably less severe when data has F^1 distributed weights as opposed to F^0 distributed weights. For the remaining methods, we verify that the overall ordering of profile precision and recall is roughly equivalent, which is in agreement with what we learn from studying the statistical distributions.

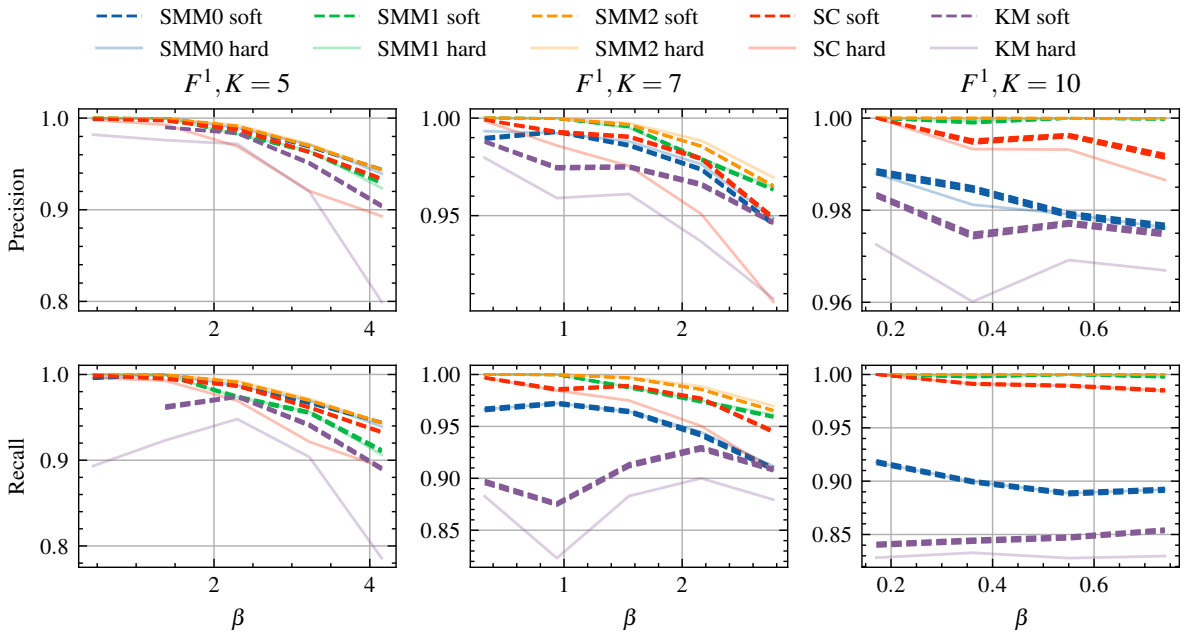


(a) Effect of prototype mixing on data space profile inference with F^0 weights.

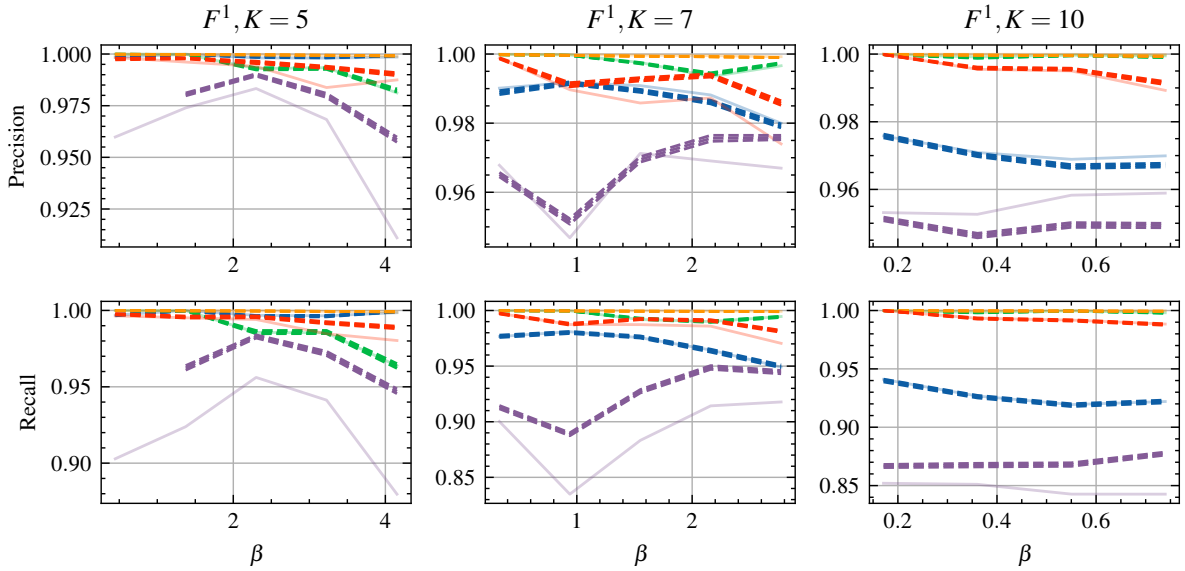


(b) Effect of prototype mixing on spectral modularity profile inference.

Figure 10.10: Effect of prototype mixing on profile inference with F^0 weights. The vertical axis represents profile precision and recall for the top and bottom rows, respectively. The horizontal axis represents the amount of prototype mixing β. The width of the dashed line represents the standard error of the mean. For each K, 5 equally separated values for β are chosen, for which 40 datasets are generated. The generated data sets have $n = 200$ and $p = 200$, group proximity $\alpha = 0.1$, group size heterogeneity $\eta = 0$, and prototype diffusion $\gamma = 0.05$.



(a) Effect of prototype mixing on data space profile inference with F^1 weights.



(b) Effect of prototype mixing on inferring representative spectral modularity vectors with F^1 weights.

Figure 10.11: Effect of prototype mixing on profile inference with F^1 weights. The vertical axis represents profile precision and recall for the top and bottom rows, respectively. The horizontal axis represents the amount of prototype mixing β. The width of the dashed line represents the standard error of the mean. For each K, 5 equally separated values for β are chosen, for which 40 datasets are generated. The generated data sets have $n = 200$ and $p = 200$, group proximity $\alpha = 0.1$, group size heterogeneity $\eta = 0$, and prototype diffusion $\gamma = 0.05$.

11
Beyond Synthetic Data

In this chapter, we investigate the use of spectral modularity in settings beyond synthetic data. In particular, we perform the clustering on a selection of real, empirically labeled data sets. Our evaluation of the clustering methods is then partially based on the partition that is induced by the provided labels in the data set. While the data labels can be used to derive a particular ground-truth group structure, it is not always the case that this ground-truth is represented by the data. Therefore, the performance of clustering methods that are observed in real data should be considered with care. This is generally true for any clustering evaluation based on real empirical data. Nevertheless, the application of the methods to real empirical data encourages the extension of the study from specific synthetic data settings to a more generally applicable field.

In settings outside of synthetic data, it is important to have a correct intuition of the metric space and a sensible null model for the random data matrix. In synthetic data settings, these choices are almost trivial, can be expressed theoretically, or can be confidently obtained through shuffling based parallel analysis. However, in empirical data, this can be especially difficult where features can be redundant or correlated.

Therefore, the goal of this chapter is to simultaneously demonstrate the potency and challenges of spectral modularity maximization outside of synthetic data, display the difficulty of evaluating clustering methods with labeled empirical data, and illustrate the importance of selecting a reasonable null model. To achieve this, we investigate the clustering performance of a relatively small categorical data set and of the well-known `MNIST` handwritten digit image data set.

In Section 11.1, we investigate a categorical data set containing attributes of soybeans. Here, we find that although the fine-grained structure of the ground-truth partition is not found, spectral modularity maximization is capable of capturing most of the underlying group structure. In Section 11.2, we rediscover the spectral modularity breakdown in the handwritten digits by reconstructing the data set to contain a varying number of groups K. Furthermore, we evaluate the behavior of shuffling based parallel analysis to determine the null model matrix of this data set.

11.1. Soybean

The soybean data set [45] is a set of descriptive attributes of soybeans obtained from the UCI data bank. The attributes are mostly categorical. The few ordinal exceptions, `date`, `leafspot-size`, and `germination`, and the objects with missing values are removed from the dataset to maintain an easy-to-manage 266 by 32 data matrix $\mathbf{X}$. The data matrix is visualized in the middle panel of Figure 11.1. The Hamming similarity matrix, as defined in Equation 3.4, of the data set is displayed in the right panel of the figure. Finally, the partition that was obtained from the provided class labels in the data set is given in the left most panel of the figure. Each color represents a different class, and the objects are ordered such that objects in the same class are next to each other. There are $K = 15$ classes that are specified.

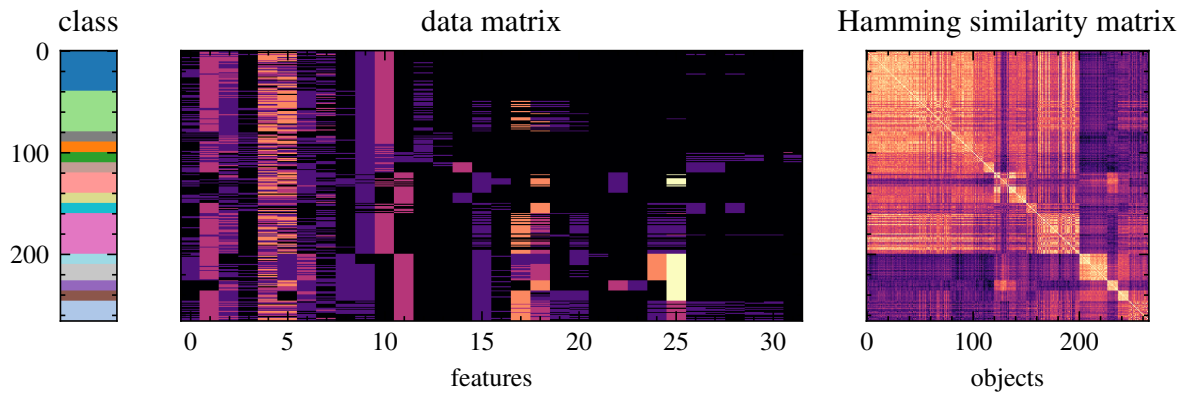


Figure 11.1: Data matrix, Hamming similarity matrix and partition of soybean data set. The leftmost panel represents the ground-truth partition that is obtained from the provided class labels. The middle matrix is an $n = 266$ by $d = 32$ data matrix $\mathbf{X}$, where the color values represent different categorical values. In the right panel, the $n \times n$ Hamming similarity matrix is displayed. The brightness of the color represents similarity between objects.

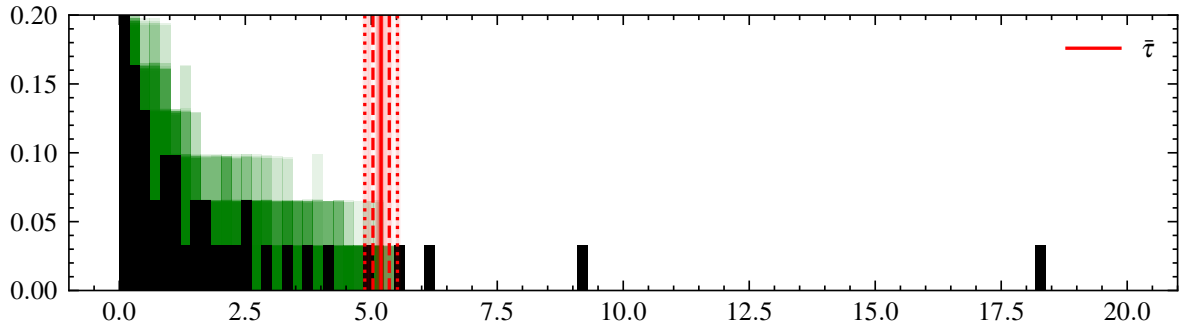


Figure 11.2: Eigenvalues of Hamming similarity matrix of soybean data set. The largest eigenvalue is omitted from the histogram. For visibility, the figure is cropped in a vertical direction.

In Figure 11.2, we see the eigenvalue distribution of the Hamming similarity matrix of the data set, in black, and the eigenvalue distributions of the Hamming similarity matrices after employing the data shuffling procedure from Algorithm 1, in green. From this, we also obtain an eigenvalue threshold, depicted by the red vertical bar and denoted by $\bar{\tau}$, that approximates the distinction between informative and uninformative eigenvalue-eigenvector pairs. The threshold is the average of the second-largest eigenvalue obtained from 50 shuffling procedures. The dashed line represents one standard deviation, and the dotted line represents two standard deviations. While the labels of the data set give us an indication of the number of groups being $K = 15$, the number of clusters approximated using shuffling based parallel analysis is $\hat{K} = 4$, depending on the confidence interval assumed in the threshold procedure.

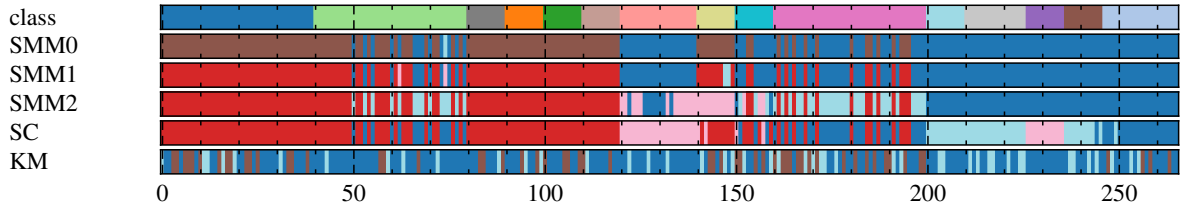


Figure 11.3: Clustering on the soybean data set. Colors represent the different clusters. The color coding is only relative to the rest of the partition and shares no additional meaning in relation to the other clusterings.

This discrepancy between $\hat{K} = 4$ and $K = 15$ does not necessarily indicate that the shuffling based parallel analysis is malfunctioning. In fact, it underlines the malpractice of naively using generic labeled data to evaluate clustering methods. For example, while the provided classes in the dataset for objects range 0 to 200, as found in Figure 11.1, are encoded as 10 different groups in supervised partition, their pairwise similarity levels are relatively high for both internal pairs and external pairs. This is seen in the structure of the sub-matrix obtained from indices $[0, 200] \times [0, 200]$ in the similarity matrix of Figure 11.1. For this reason, it is difficult to evaluate the clustering performances on the soybean data set by comparing them to the ground-truth partition.

In Figure 11.3, we see the resulting clustering of the data set by the methods KM, SC, SMM0, SMM1, SMM2, and SC, compared with the ground-truth partition derived from the class labels. In this setting, we observe little effect of the spectral modularity breakdown because the clusterings obtained with SMM0 and SMM1 are almost identical. This is expected, as the number of detected spiked eigenvalues, $\hat{K} = 4$, is relatively small. We see that the KM method constructs a clustering that shares little resemblance with the supervised partition. The spectral clustering (SC) method and SMM2 are almost identical, except that SC disentangles objects in the range 200 to 250 and SMM2 disentangles some objects in the range 50 to 70 and in the range 160 to 200.

11.2. Handwritten Digits (MNIST)

The MNIST dataset, which contains images of handwritten digits, is a popular tool for evaluating machine learning algorithms. While traditionally a computer vision task, flattening the 8×8 images to high dimensional feature vectors gives a reasonable evaluation framework for clustering methods in high dimensional data. In doing so, we obtain additional insight into the behavior of the clustering methods outside the synthetic data setting in a slightly more quantitative way than can be done for the bean data set. Therefore, it is important to note that for the precise purpose of recognizing or clustering images, it is advised to use the appropriate tools for image recognition. The demonstration with digits here is mainly to illustrate the behavior of the clustering methods and their accompanying random matrix based spectral tools in a setting that is not as ideal as the synthetically generated data sets from Chapter 9.

The MNIST data set can be reconstructed with $K \in \{3, \dots, 10\}$ such that we can investigate the behavior of the clustering methods for varying K in a setting outside synthetic data. We use the Manhattan distance $d(x, y) = |x-y|$ as a metric for the data space of the flattened digit images, i.e., $[0, 255]^{64}$. Then the accompanied similarity metric is as defined in Equation 3.3. In Figure 11.4, we see an example visualization of three digits. The three left most images are 8×8 images obtained from the MNIST_64 data set.

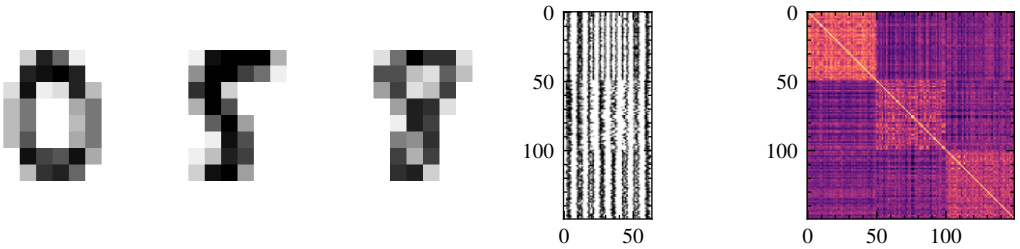


Figure 11.4: MNIST digits data set. The three left most images are examples of the handwritten digits. The first matrix represents a flattened 150 by 64 data matrix that is obtained from 150 images of dimension 8 by 8. The rightmost matrix is the Manhattan similarity matrix of this data set.

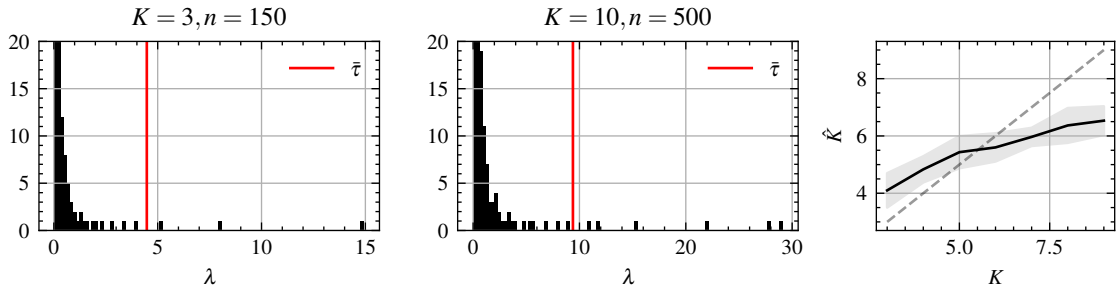


Figure 11.5: Eigenvalue histograms of MNIST data. The left most figure contains a histogram of the eigenvalues (without λ_1) for a dataset containing three equally sized groups of size 50. The middle figure contains a histogram of the eigenvalues (without λ_1) for a data set of 10 equally sized groups of size 50. The red vertical line is the threshold determined by shuffling based parallel analysis $\bar{\tau}$. The rightmost figure contains the number of spiked eigenvalues $\hat{K}$ obtained through the shuffling based parallel analysis as a function of the number of digits present in the data set K. The size of the data set containing K different digits is $50 \cdot K$ and the statistical mean is the result of 30 iterations. The shaded area is the standard deviation.

In Figure 11.5, we see an example of eigenvalue histograms for the similarity matrix with a selection of $K = 3$ digits $(0, 5, 8)$ in the left part, and in the middle we see the similarity matrix for the case where we use all the digits, i.e., $K = 10$. In the rightmost figure, we see a figure containing the number of spiked eigenvalues $\hat{K}$ as a function of the number of digits, K included in the data set. This line indicates that for small K, the number of spikes is over-approximated, while for large, K the number of spikes is under-approximated by the shuffling based parallel analysis.

11.2.1. Partition Recovery

In the left panel of Figure 11.6, we see the inverse performance (VI) of the clustering methods. Most importantly, we find that two observations from the synthetic empirical analysis can be recognized. First, the KM method performs relatively poorly compared to all the other methods. Second, the performance of the naive spectral modularity maximization (SMM0) worsens as the number of groups K grows. Finally, the spectral methods perform roughly equally, except for small K where SMM0 and SMM1 outperform SC and SMM2, and for high K where SC is slightly better.

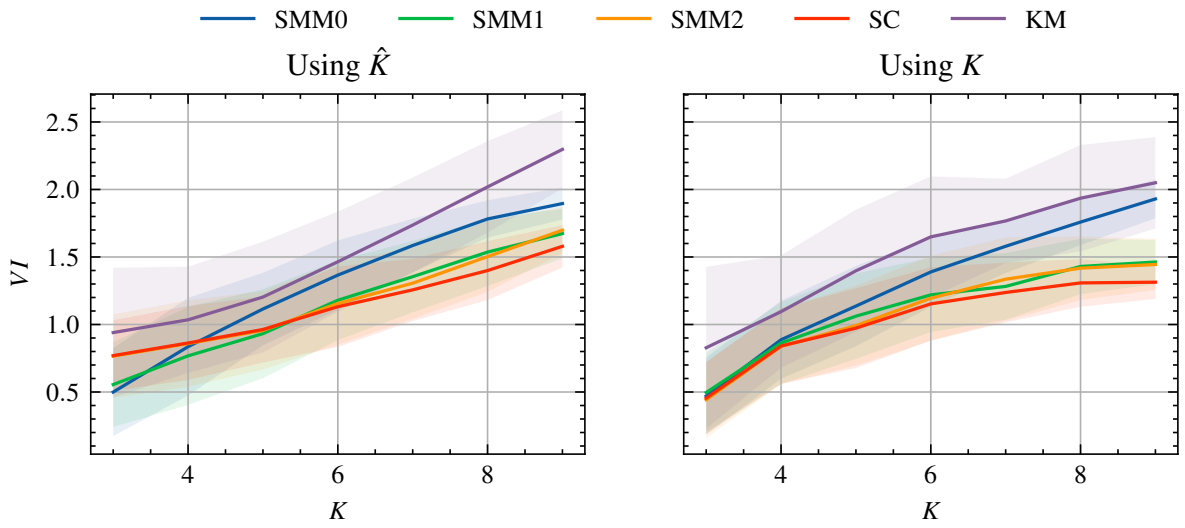


Figure 11.6: Performance of clustering methods on MNIST data. The clustering performance is computed with the variation of information between the clustering and the ground-truth partition. In the left, the performance of the clustering methods (SMM0, SMM1, SMM2, SC, and KM) is evaluated as a function of the number of included digits K. Here, $\hat{K}$ is determined by the standard shuffling based parallel analysis. In the right figure, the same evaluation is done, but $\hat{K}$ is replaced with the oracle value K.

Because we know from studying the eigenvalue histogram that the shuffling based parallel analysis may not be effectively determining the number of groups in the data set, we verify the intuition by performing the experiment with an oracle estimator K, as seen in the right panel of Figure 11.6. Instead of using an approximation $\hat{K}$ to determine the informative eigenvalues, we directly use the known number of groups K. The performance of the clustering methods using the oracle estimator for K, does indeed improve slightly, indicating that the shuffling based parallel analysis may not be optimal. However, the ordering of the methods based on their clustering performance stays roughly the same. The most important observation, however, is that the spectral modularity breakdown is still exhibited, as indicated by the deteriorating performance of SMM0 as K grows. This further demonstrates the universality of the challenge of naive spectral modularity maximization.

11.2.2. Digit Recognition

In order to illustrate the behavior of under-approximating the number of spiked eigenvalues and hence the number of groups, we look at the inferred profiles from a single clustering. In Figure 11.7, we find the result of profile inference on the clustering of normalized spectral modularity maximization (SMM2). The data set that is considered contains all the $K = 10$ digit classes, where each group has 100 objects, totaling to $n = 1000$ objects. In the left panel, we see the clustering with the shuffling based approximate for the number of clusters and informative eigenvalues $\hat{K} = 7$. The inferred profiles, with mode based inference introduced in Equation 2.6, clearly resemble existing digits. The digits that are recognized are $4, 5, 6, 1, 9, 0, 7$, where the cluster representative profile of 9 is very similar to 3. This ambiguity between digit 9 and 3 is confirmed by the fact that the cluster associated with profile 9 contains many values for 9, 3, 8, and the cluster that belongs to the cluster representative profile 1, appears to contain many instances of 1 and 2.

(a) Using $\hat{K}$

(b) Using K

Figure 11.7: Cluster profiles of MNIST digits data set. The left uses the $\hat{K}$ eigenvalues obtained from threshold with shuffling based PA. The right uses an oracle value of $K = 10$ to determine the threshold of the eigenvalue spectrum. For each row, the left image is the inferred profile, and the right images contain example objects that are in the clusters represented by the profile. The number of objects in this visualization of the clusters is capped at 90.

It is tempting to think that replacing the approximate $\hat{K}$ for the oracle value K will disentangle the clusters that contain multiple digits. However, this is not entirely true, as is seen in the right panel of Figure 11.7. While in this oracle-based method, the digit 2 is recognizable in the representative profiles, the remainder of the undiscovered digits remain unrecognizable. To be specific, there is no profile that resembles the digits 8 or a 3. Instead, the clusters that are associated with the digits 1 and 7 are both split, resulting in two clusters with a profile resembling 1 and two clusters with a representative profile resembling 7.

Part IV

Discussion and Conclusions

12
Related Work

In this chapter, we examine relevant literature to highlight the contributions of this study within the research field. In particular, certain fundamental aspects of the studied methods, i.e., naive (SMM1), regularized (SMM1), and normalized (SMM2) spectral modularity maximization, bear resemblance to existing work within the fields of spectral clustering and Girvan-Newman modularity maximization.

First, given its reliance on spectral decomposition, the methodological equivalence of spectral modularity methods with the existing spectral clustering paradigm is evident. However, the conceptual usage of the spectral decomposition of similarity matrices as a modularity measure, as introduced in [8], is relatively unique. Specifically, most of the existing spectral clustering work [92, 93, 62, 15] is based on the eigenvectors of the Laplacian matrix instead of the eigenvectors of the similarity matrix, which is done in spectral modularity.

Second, the use of modularity based objectives, akin to [12], for clustering multivariate data is not new. Although existing methods [94, 95] typically rely on the Girvan-Newman modularity discussed in Chapter 4, there are shared aspects with the introduced spectral modularity methods. Along these lines, the Girvan-Newman modularity suffers from a similar fundamental limitation, known as the resolution limit [96], which may appear to have some conceptual overlap to spectral modularity breakdown. Furthermore, existing normalizations and vector interpretations of the Girvan-Newman modularity demonstrate equivalence to the enabling concepts of the SMM2 method.

In Section 12.1, we elaborate on the relation of spectral modularity to the spectral clustering paradigm. Specifically, we discuss the use of Laplacian matrices in contrast to similarity matrices, which is highlighted by the resemblance of SMM2 to a specific existing similarity-based spectral clustering method [97]. In Section 12.2, we discuss existing approaches to multivariate data clustering using Girvan-Newman modularity and how the developments in that direction resemble aspects of the methods in this thesis.

12.1. Spectral Clustering

Spectral clustering, as discussed in Chapter 2, is a clustering paradigm that makes use of the most important eigenvectors of matrices associated with a dataset or graph. There are many variations of spectral clustering, typically relating to a slightly different graph partitioning objective. The simplest objective is to minimize the cut of a graph, which essentially boils down to finding a partition that minimizes the similarity between the clusters. While this is one of the simplest algorithms, the obtained clusters are generally inconsistently split into uneven sizes.

To mitigate this, [92] optimize a ratio cut objective, [93] study a min-max cut objective, and [62, 14] study a normalized ratio cut, which is generally suggested [13] to be the most applicable in typical settings and is the spectral clustering algorithm we considered throughout the thesis. For an elaborate overview of the recent advances in spectral clustering, the reader is referred to [15].

12.1.1. Similarity and Laplacian

In contrast to many of the above specified Laplacian based spectral clustering methods, spectral modularity maximization is based directly on the eigenvectors of the similarity matrix. While early in the development of spectral clustering methods the eigenvectors of the adjacency matrix were used [98], much of the proceeded developments were done on the eigenvectors of the Laplacian matrix, which is originally introduced in [61].

There is no consensus on the suitability of clustering data using the eigenvectors of similarity matrices. In [13] the author argues for the use of eigenvectors of the Laplacian as opposed to those of the similarity matrices because the smallest eigenvalue eigenvector pairs of the Laplacian are more meaningful. Specifically, because the zero eigenvalues of the Laplacian matrix represent the connected components of a clustered graph in an ideal setting.

On the other hand, [12] argues that the analogy of Laplacian based spectral clustering with graph cut sizes is fundamentally problematic in reflecting the concept of network communities. The argumentation suggests that a good division of a network into communities is not merely one in which the number of links between groups is small, but rather one in which the number of links between groups is smaller than average. Hence, this argues for the use of modularity based community detection as opposed to Laplacian based community detection. Whether the same argument is reasonable in the context of multivariate data is an open question.

In the context of graphs, an adjacency matrix is not guaranteed to be positive definite, which makes the Laplacian matrix particularly convenient. The Laplacian matrix is a convenient mathematical object as it satisfies symmetric positive (semi-)definiteness, and its eigenvectors approximate a Euclidean embedding of the data. The former is important to ensure that the trace maximization problems that are encountered through graph clustering definition, like normalized cuts, can be solved with the eigenvectors of the Laplacian matrix.

However, for many choices of similarity measures, the induced similarity matrices are naturally already symmetric positive definite, which makes it common for spectral clustering based methods to be based on a symmetric positive definite similarity matrix instead. This makes a Laplacian transformation arguably redundant, especially if one is not strictly interested in the Euclideanity of the Laplacian embedding. Along these lines, in [18, 99] the authors study the kernel matrices of multivariate data that exhibit a ground-truth group structure. The exact benefit of this matrix, as opposed to the Laplacian, is that under a Gaussian assumption of the matrix elements, the matrix conveniently aligns with the theoretical structure of random matrix theory. This way, the behavior of spectral clustering on similarity matrices can be improved by studying the theoretical properties of the eigenvalues.

12.1.2. Related Similarity-Based Method

A neat property of Laplacian based spectral clustering, is the ability to use KMeans clustering in the Laplacian embedding, which is motivated by its Euclideanity. However, because clustering based on the eigenvectors of a similarity matrix does not satisfy Euclideanity, alternative methods need to be derived. For example, [97] introduces a method that uses the eigenvectors of a similarity matrix. In particular, the first K eigenvectors are used, where K is determined by Kaiser's Criterion. Then, if two objects projected on the subspace that is spanned by the K eigenvectors have a cosine similarity greater than a specified threshold, the objects are clustered together. The threshold is calibrated such that K clusters are found. Although it is not identical, the approach does share commonalities with both SMM1 and SMM2, due to the use of angular orientations and a threshold calibration procedure.

12.2. Girvan-Newman Modularity Maximization

Although [8] introduced the redefinition of modularity using eigenvectors and random matrix theory, i.e., spectral modularity, it is not the only attempt at using the modularity definition from [12] in the context of clustering multivariate data. For example, in [94] the authors propose an extension to the modularity objective that relates to Relational Analysis. Furthermore, [95] uses a different normalization of the modularity matrix, one that borrows the specific normalization from the well established normalized Laplacian matrix defined in [62].

Apart from these modularity based methods, there is additional related work that resembles aspects of the developments within this thesis. First, the resolution limit is known to be a fundamental problem, which at first glance may appear to be related to the spectral modularity breakdown. Second, normalizations of modularity are studied to obtain alternative modularity maximization algorithms [82]. Third, modularity based vector representations of objects have been studied in the context of graph clustering algorithms; however, this is only done for the Girvan-Newman modularity definition. Finally, the extension of modularity maximization to obtain soft partitions

12.2.1. Resolution Limit and Spectral Modularity breakdown

In a simple graph setting, the Girvan-Newman modularity based community detection methods suffer from a failure mode that is called the resolution limit [96]. Modularity maximization may fail to identify smaller clusters. This phenomenon appears among many network classes and is an important limitation of naive modularity maximization. The resemblance between the spectral modularity breakdown and the resolution limit lies in the inability to uncover fine-grained structures that alternative existing methods are able to detect. Therefore, aspects of both phenomena are likely to be relatable.

On the other hand, because spectral modularity operates on multivariate data, little is known about the presence of such a resolution limit in spectral modularity. In particular, the resolution limit is often studied for imbalanced community configurations, which demonstrate the failure of detecting smaller-sized communities. However, even in completely homogeneous size and symmetric group separation, we observe a significant breakdown in Chapter 10, which is particularly severe for data with internally non-uniform groups, i.e., the CMPM data that we introduced in Chapter 9.

12.2.2. Modularity Normalization

There are other normalizations of modularity that are related to the spectral modularity normalizations in Chapter 7. Most are introduced in the context of graph clustering. In [100] two specific penalizations of the modularity objective are given.

First, the authors introduce a balanced variant of the modularity objective that resembles the Q_{avg} objective we described in Chapter 7 and of which we know may pose problematic decisions in the context of clustering multivariate data with spectral modularity.

Second, the authors introduce another normalization that is based on the same procedure as [95], which relates the modularity matrix to a Laplacian. This way, clustering can be done with KMeans on the eigenvectors, akin to Laplacian based spectral clustering, which suggests that the normalizations from [95] may uncover a more intricate relationship between the modularity matrices and Laplacian matrices.

12.2.3. Modularity Vector Representations

In the context of graph clustering, [101] also uses a similar notion of a lower dimensional representation of the objects based on the Girvan-Newman modularity matrix. However, this representation is obtained from the eigenvectors of the Girvan-Newman modularity matrix and not the spectral modularity matrix. In addition, the authors use KMeans to cluster the data in the lower dimensional representation. However, as discussed in Section 12.1, the KMeans objective is not meaningful because the eigenvectors of the Girvan-Newman modularity matrix do not provide a Euclidean embedding.

To address this conceptual issue, [102] consider a method that is based on the inner products as opposed to the distances in KMeans. These inner products are meaningful as they represent the pairwise Girvan-Newman modularity, similar to the inner products of spectral modularity vectors, as discussed in Chapter 7, which makes the philosophy of the method resemble that of SMM2.

13 Discussion

In this chapter we discuss the developed methods, their challenges, and their performance. The aim of this study is to investigate the viability of spectral modularity, and specifically the ability to cluster high dimensional multivariate data.

Although we learn that the naive spectral modularity maximization is an inconsistent method, the theoretical results obtained from studying the spectral modularity breakdown incite the development of two solutions that are aimed at resolving the breakdown. These enhancements show direct improvement over the naive spectral modularity maximization and existing clustering methods through a thorough empirical performance analysis. However, the performance analysis has limitations related to the scope of the experimentation being limited to a small selection of evaluation criteria, synthetically generated data and few competitive clustering methods. Furthermore, there are some aspects of the spectral modularity methods, related to the computational efficiency and the parallel analysis procedure, that may hinder practical application.

In Section 13.1, we summarize and discuss the theoretical and methodological developments of this thesis. Specifically, we highlight our findings related to the spectral modularity breakdown, and the properties of the enhancements. In Section 13.2, we elaborate on the performance of our contributed methods in terms of partition recovery and profile inference. In Section 13.3, we describe the limitations of our employed performance analysis. Finally, in Section 13.4, we discuss the limitations of spectral modularity as a whole.

13.1. Theoretical and Methodological Developments

The framework of spectral modularity relies on two steps. First, we filter out informative eigenvalue eigenvector pairs of any symmetric positive definite similarity matrix from non-informative ones, as described in Chapter 3. Second, we perform modularity maximization on a constructed spectral modularity matrix, composed of only informative eigenvalues and eigenvectors, as described in Chapter 4. From studying this framework, there are three main takeaways.

First, we find that naive spectral modularity maximization suffers from a fundamental challenge: spectral modularity breakdown. This phenomenon is examined throughout this thesis through an analytical study of an ideal setting and the effect of perturbations in this ideal setting. However, there remain open questions related to the tractability of the perturbation results. Second, we are able to mitigate the breakdown with two related solutions, SMM1 and SMM2. Both solutions have advantages, disadvantages, and open questions that may lead to significant improvements or enlightening alternatives. Third, we can use spectral modularity to derive a soft clustering method that is capable of transforming arbitrary hard partitions into soft partitions without choosing parameters.

13.1.1. Spectral Modularity Breakdown

The behavior of a naive maximization of the spectral modularity objective is studied in Chapter 5, and answers to Research Question 1. Here, we study naive spectral modularity maximization through a theoretical analysis of the spectral modularity objective from Section 4.3. However, the method breaks down as the number of groups that are present in the data is high. This makes the application of naive spectral modularity maximization problematic.

A natural follow-up question concerns the cause of this spectral modularity breakdown, as is posed in Research Question 2. To understand the breakdown, we provide an intuition of the breakdown in Section 5.1 through a combinatorial reasoning that shows the shrinking ratio of the $O(K)$ internal pairs to the $O(K^2)$ external pairs. Because of the relatively natural link to this combinatorial phenomenon, it is tempting to believe that the breakdown has some universal behavior. To be precise, the breakdown may extend to different tasks within high dimensional statistical domains, where the size of the underlying hidden representation is large.

The formal analysis of the spectral modularity breakdown comprises a necessary consistency condition that is based on the spectral modularity objective, the spectral modularity matrix, and a non-ambiguous ground-truth partition. In particular, we design two mathematical models that are related to the construction of spectral modularity matrices, to show that the non-ambiguous ground-truth partition associated with these models will likely break this necessary consistency condition from Section 5.2. In this way, the spectral modularity objective becomes inconsistent with these two rather fundamental models. In the first model, Toy Model A (TM-A), we demonstrate that for a completely symmetric and homogeneous setting, the spectral modularity matrix becomes arbitrarily close to violating the consistency condition, as discussed in Section 5.3. Then, increasing K makes the spectral modularity objective more susceptible to ground-truth violating perturbations, breaking the consistency of the spectral modularity matrix. In the second model, Toy Model B (TM-B), we demonstrate using a model for the spectral modularity vectors that with only small deviations from the idealized setting, the consistency condition is broken, as discussed in Section 5.4 In particular, we demonstrate an upper bound on the probability of a ground-truth partition satisfying the necessary condition, and show that as K grows, the upper bound will converge to zero, meaning the probability of satisfying the necessary consistency condition will converge to zero.

Intractable Perturbations

Ideally, one would perturb these spectral modularity vectors and express a probability bound in terms of the analytical distribution of these vectors. However, this is difficult to do because of the dependence among the perturbations. Therefore, we resort to the analysis of the sum of the inner products of spectral modularity vectors, denoted by z_K. Here, we learn that as long as z_K has some sufficient amount of variance in relation to K, the breakdown is guaranteed.

However, a counterexample to guaranteed breakdown exists, specifically if the amount of variance of z_K vanishes too fast as K grows. If this is the case, the upper bound may never go to zero, which makes the guaranteed breakdown argument fail. However, this setting would require adversarially chosen perturbation distributions that are only associated with tiny perturbation amounts. Therefore, these situations are unlikely to occur in realistic settings, where randomness is generally a fundamental component of the system.

Because an explicit distribution of z_K determined from the distribution of the perturbations $\{\mathbf{z}_i\}_{i=1}^n$ is difficult for non-trivial distributions, an interesting question is whether a specific central limit theorem, that allows certain dependence structures, may apply in this setting. As is illustrated in the numerical analysis of spectral modularity breakdown in Section 5.4, a normal distributed z_K, as a potential result of such a conjectured central limit theorem, would suffice to show the existence of the spectral modularity breakdown.

13.1.2. Spectral Modularity Enhancements

In order to mitigate the spectral modularity breakdown, we study enhancements to the spectral modularity objective. In fact, we provide two solutions, thereby answering Research Question 3. Both solutions (Contribution 1 and Contribution 2) reduce the bias towards clusters with few big, by combining the following two aspects. First, both solutions ensure that the clusterings contain the right number of groups, that is, under a reasonable setting, well approximated by the number of spiked eigenvalues, as discussed in Chapter 3. Second, the solutions reduce the bias towards heterogeneous groups, such that we prevent inconsistent merges that are caused by the spectral modularity breakdown, as discussed in Chapter 5.

Regularization

In Chapter 6, we provide a solution (SMM1) that employs an explicit regularization of the spectral modularity objective by penalizing heterogeneously sized groups. A benefit of this method, is that its adaptation of the objective is rather minimal. In fact, its adaptation solely consists of the subtraction of a small constant from the modularity matrix. Given that the change only adapts the modularity matrix, ensures that the remainder of the existing modularity maximization framework can still be used effectively. In practice, this means that we are able to use the existing modularity maximization algorithms, such as Louvain [39], which poses as a benefit of this particular solution.

However, a fundamental disadvantage of SMM1 is the necessity of calibrating the regularization parameter ϵ. This requires the use of rather cumbersome approximation heuristics, for which we propose a potential improvement in Section 14.2.

Normalization

In Chapter 7, we provided a secondary solution (SMM2) that employs a more fundamental change of the spectral modularity objective through a specific normalization. Instead of maximizing the summation of all internal sums of pairwise modularities, as is done in the naive spectral modularity maximization (SMM0), we maximize a normalized sum. Because the objective has changed significantly, we cannot use existing modularity maximization methods. Instead, we use the orientations and magnitudes of the spectral modularity vectors, which enables a simple maximization procedure. While this method does not require additional parameter calibration like in SMM1, due to the relative novelty of this procedure, it is likely that improvements, such as we discuss in Section 14.3 and in Section 14.4 can be made.

13.1.3. Soft Clustering Method

A relatively natural interpretation of the orientations of spectral modularity vectors, inspires a way to determine an expression for the uncertainties of cluster memberships. Therefore, in Chapter 8, we introduce a soft clustering algorithm (Contribution 3), where the orientations of spectral modularity vectors are used to determine the membership distribution of objects to the different clusters. This soft clustering can be performed on a hard partition that is obtained from an arbitrary clustering algorithm. On the one hand, this makes it possible to gain detailed insights into the group structure of a data set while not solely depending on the spectral modularity framework. On the other hand, as the softening is relatively separated from the (arbitrary) clustering procedure, it is tempting to think that a procedure that incorporates the two phases, such as we propose in Section 14.5, will give more meaningful results.

13.2. Experimental Setup and Analysis

In Chapter 10, we study the performance of the clustering methods in terms of partition recovery and profile inference, as discussed in Section 10.1, of SMM0, SMM1, and SMM2 in comparison to existing methods, such as KMeans, KMedoids, and a specific spectral clustering method (SC), answering to Research Question 4. Our theoretical analysis of naive spectral modularity maximization (SMM0) in Chapter 5 suggests that a breakdown of the method occurs as the number of groups grows, and the severity of the breakdown is observed in a multitude of ways.

From the combination of all results in Section 10.2 and in Section 10.3, we find that SMM2 and its soft variant have the best performance in most of the tested scenarios. In particular, the conclusions we can draw from the experiments are threefold. First, the performance is significantly deteriorated due to spectral modularity breakdown, which is successfully resolved by SMM1 and SMM2, although SMM1 sometimes demonstrates inferior performance. Second, in the setting of internally non-uniform groups, modeled by CMPM, the enhanced spectral modularity methods, especially SMM2, demonstrate a significant superior performance over SC. Third, in terms of profile inference, the soft clustering variants typically improve the profile inference performance.

13.2.1. Spectral Modularity Breakdown

In Chapter 10, we verify these observations through empirical analysis of the performance of the naive spectral modularity method. Although SMM0 generally performs better than the baseline methods (KMeans and KMedoids), which suffer from the curse of dimensionality and general instability, SMM0 performs significantly worse than a well-established competitive clustering method (SC). The trend with which the clustering performance worsens is indicatively observed when the performance is measured as a function of the number of groups. In fact, for a small number of groups, SMM0 has relatively similar performance to its competitors and improvements; however, as the number of groups grows, the performance gap increases a lot.

For high dimensional Gaussian mixture based data described in Section 9.2, we find in Section 10.2 that SMM0 performs generally better than KMeans. An exception to this is when the number of groups is large or when the number of dimensions is low, where KMeans performs equally well. This can be explained in two ways. First, the performance of SMM0 deteriorates to the level of KMeans because of the spectral modularity breakdown. Second, the performance of KMeans is still relatively good, due to the low dimensionality. When the number of groups is large, the competitive method, spectral clustering, and the contributed solutions perform better than SMM0.

For categorical mixed prototype data described in Section 9.3, we again find in Section 10.3 that SMM0 performs better in the baseline method (KMedoids), but worse than the competitor (SC). In addition, the performance of SMM0 is severely and significantly worsened when the number of groups grows. To be precise, the spectral modularity breakdown appears to be much more sensitive in this mixed prototype setting. This is due to the softness of the group boundaries in the categorical mixed prototype data. Because the boundaries of groups in the CMPM based data are relatively soft, meaning that the internal similarities are not homogeneous, increasing K is much more likely to cause inconsistent clusterings. This suggests that there are different levels of severity of the effect of breakdown.

Furthermore, in the context of real empirical data, we also observe the effect of breakdown. In the handwritten digits data set that we study in Chapter 11, the spectral modularity breakdown is also observed when we include a high number of digits (e.g. $K = 10$). However, the amount of real empirical data studied is too limited to draw strong conclusions.

In terms of partition recovery, we find that the breakdown disappears in SMM1 and SMM2. In particular, as the number of groups grows, the partition recovery performance does not degrade compared to the competitive spectral clustering method. This is observed in both the GMM-based data and the CMPM-based data, where the breakdown appears to be more severe.

On the other hand, a potential limitation of both SMM1 and SMM2 is observed when varying cluster size heterogeneity. Because both the solutions to mitigate the spectral modularity breakdown that are used in SMM1 and SMM2 are based on tackling the bias towards a few big groups, they may overcompensate in the actual presence of actual heterogeneous group structure. The observed effect of this is, however, relatively small.

13.2.2. Internally Non-Uniform Groups

Both the enhanced spectral modularity methods, SMM1 and SMM2, demonstrate significant superior performance over SC, which is unsuitable for data with internally non-uniform groups. In this setting, the groups of objects in the Laplacian embedding are not convex or linearly separable. These two conditions are desired for KMeans clustering to perform reasonably. If the conditions are not met, KMeans will typically converge to an ill-defined optimum. Because the weights of the prototype contributions of objects in mixed prototype data may be distributed in a non-convex way, this specific clustering algorithm is likely to fail. This phenomenon is repeatedly observed in the study of clustering with data generated with Logit Chi squared (F^1) weights, as introduced in Section 9.2.

13.2.3. Soft Clustering Improves Profile Inference

In terms of profile inference, as discussed in Section 10.4, a similar performance ordering of soft variants of the methods is found. In particular, the soft variant of SMM2 is generally considered as the top performer, which is explained by the maximization algorithm of SMM2, which makes use of the orientations of cluster representative vectors. The superior performance of SMM2 is highlighted even more when considering the profile inference in terms of the space of the spectral modularity vectors. This latter performance quantification is almost identical to the maximization objective of SMM2.

In fact, the soft clustering variants of any method generally improve profile inference. This is expected, as the softness of the clustering assignments makes objects that are on the boundaries of two clusters, and consequently not representative of a data profile that strongly represents any of the two clusters, less important for the profile inference. However, an exception to this is SMM2. In particular, because SMM2 already approximately maximizes the profile inference as an objective, the extension to soft-clustering does not improve the objective further. Possibly this behavior can be explained by the same arguments that are used to express the inability of spectral modularity maximization on the space of soft partitions in Section 8.2. Here, because of the convexity of the search space and the convexity of the function, a maximization is attained at the boundary. It is possible that the objective in SMM2 has the same behavior.

13.3. Limitations of Performance Analysis

The limitations of the performance analysis can be divided into three aspects. First, the approach to evaluating the ability to recover partitions or to infer profiles may have some flaws. Second, despite the diversity of the generated data, there are still important characteristics in realistic data that are not covered by synthetically generated data. Third, the number of methods with which the performance of spectral modularity maximization is compared is rather small. Finally, the results extending beyond synthetic data leave some open questions regarding the eigenvalue detection threshold.

13.3.1. Limitations of Evaluation Metrics

A limitation of the current evaluation in terms of partition recovery is that it does not ensure that the chosen ground-truth partitions are in fact of higher quality in terms of the spectral modularity objective. One can imagine that if the ground-truth partition is ill-defined, i.e., it does not represent the data reasonably, the comparison to the ground-truth partition is problematic. Even though we ensure that the ground-truth partitions from the synthetic data sets are not completely ill posed, by making the difficulty parameters of the DGPs not too high such that the spiked eigenvalues remain detected, this does not entirely solve the issue nor guarantee optimality of the ground-truth partition. Despite the measures taken, it may still be the case that a clustering $\hat{\rho}$ has a higher spectral modularity than the ground-truth partition ρ^*, i.e., $Q_0(\hat{\rho}) > Q_0(\rho^*)$.

This is a problem that is present for any evaluation of any clustering algorithm. This limitation of the empirical analysis may be partially solved by evaluating the ground-truth partition and clustering with Q_0, such that we obtain an idea of the respective qualities. Additionally, one can investigate the use of different additional cluster quality functions that are completely separate from the spectral modularity definition and see how the performance analysis can be cross validated with those metrics.

13.3.2. Limitations of Data

For practical reasons, the scope of the thorough empirical analysis is limited to synthetically generated data. Although, we extend some experimentation to a small set of real empirical data in Chapter 11, the conclusions that can be drawn are far less significant than those from the synthetic data benchmarking. Within the DGPs that are responsible for the synthetic data, multiple restrictive choices have been made. Therefore, the scope of the benchmark is somewhat limited, in the amount, and variety of data sets. Especially, settings that are more related to practical problems, as most of the understanding obtained throughout this study is obtained through synthetic data. For example, we have not studied heterogeneous diffusion amounts of the group specific probability distributions, meaning different groups have different variances or internal uniformities. Density based methods, which we did not study here but are also able to deal with non-convex and non-linear separable groups, are known to fail in this regime [54]. Therefore, it is worthwhile to consider how spectral modularity based methods are affected by heterogeneous diffusion.

13.3.3. Limitations of Compared Methods

At the same time, the number of methods that are used to compare them is limited. For the sake of space and time, we limited the formal validation of the benchmark to a relatively standard spectral clustering method, that is expected to behave reasonably well in high dimensional data. As clustering spans a large field, many high dimensional data clustering methods exist. It is, therefore, worthwhile to investigate the performance of this method with respect to more methods that are suitable for high dimensional clustering than just spectral clustering.

For example, one could investigate how methods from this thesis can be applied to Laplacian based spectral clustering adapted for the domain where it fails (F^1), as is likely to be the case for the method introduced in [97]. In addition, although KMedoids is a commonly used clustering method for discrete data, more baseline methods can be investigated. For example, KModes, as introduced in [103] are likely capable of performing better at the mixed prototype data than KMedoids. Furthermore, hierarchical clustering methods are relatively standard, and their behavior is not known in the context of CMPM based data. Also, in the scope of modularity maximization, alternative methods are known to improve the performance of Louvain. For example, the Leiden algorithm, introduced in [39], which is known to resolve some limitations of Louvain, may be used in both SMM0 and SMM1.

Furthermore, we have not thoroughly studied methods that are specifically aimed at soft clustering due to lack of known soft-clustering methods that work well on categorical data, which is required for the unambiguous mode based profile inference procedure. However, for real data where Euclidean of the metric space is satisfied, Archetypal Analysis [104] can be used as a comparison method or fuzzy KMeans. Furthermore, [105] have studied fuzzy KMedoids and [106] have studied Archetypal Analysis for categorical data. In addition, in the context of graphs, soft modularity maximization methods exist that may be used as a replacement for the Louvain method. For example, in [107], the authors introduce an algorithm that maximizes modularity with iterative propagation of modularity guided membership degrees, and in [108] use an alternative modularity objective that approaches modularity from a probabilistic partition.

13.4. Limitations of Spectral Modularity

Despite the fact that spectral modularity has previously been shown to be a promising method [8], and the fact that we have successfully circumvented the spectral modularity breakdown with our contributed algorithmic enhancements, there are still some fundamental limitations to the spectral modularity methods as a whole. First, at this stage, it appears that the computational bottleneck of the spectral modularity methods lies in the computation of the eigenvalues and eigenvectors. This makes the application to settings where the number of objects is larger impractical because of the size of the similarity matrix. Second, the current approach to estimating the number of groups may not always be appropriate.

13.4.1. Computational Efficiency

A fundamental limitation of all the contributed methods, that is natural to any of the spectral methods, lies in the time complexity, which is severely dominated by the computation of eigenvectors and eigenvalues. While modularity maximization algorithms, such as Louvain, are designed to maximize the modularity of large graphs, the extension to large similarity matrices in the spectral modularity paradigm is not trivial. In particular, the spectral modularity approach requires a spectral decomposition, and therefore the eigenvectors and eigenvalues of the matrix need to be computed, which generally takes $O(n^3)$ time.

The time complexities of variety modularity maximization algorithms are not considered in great detail. However, it is unlikely that the time complexities of these heuristics will be more problematic than the spectral computations. Indeed, for a simplistic perspective, consider that the empirical time complexity of Louvain is $O(n \log n)$ and the calibration scheme utilized in SMM1 takes, in the worst case $O(\log n)$ time. Therefore, this is clearly dominated by $O(n^3)$ which is required for the computation of the spectral modularity. Furthermore, the sort phase in SMM2 takes $O(n^2)$ time in the worst case, the seed phase takes at most $O(n)$ time, and the assignment phase takes $O(n \cdot K)$ time. Therefore, SMM2 is also dominated by the $O(n^3)$ required for the spectral decomposition.

Therefore, most of the computational efficiency can be gained by improving the efficiency of the spectral decomposition. A fundamental component is the computation of the top-K eigenvectors and eigenvalues that are required to construct the spectral modularity matrix. Fortunately, in the context of spectral clustering, this problem is addressed by an approximation of the Laplacian embedding [109], potentially affecting the quality of the clustering performance. Furthermore, for the determination K, we currently use shuffling based parallel analysis as described in Algorithm 1, and therefore the second-largest eigenvalue needs to be computed many times. Alternative methods for approximating K may influence the time complexity.

13.4.2. Problematic Approximation of K

As demonstrated in Chapter 11, it is not trivial that shuffling based parallel analysis approximates a sensible number of spiked eigenvalues, let alone the right number of groups. From the experimentation on the `MNIST` digits dataset, it is not entirely clear, whether the spiked eigenvalues are theoretically too close or that the method to obtain the threshold is invalid. The shuffling based parallel analysis to determine the number of clusters in the handwritten digits, has both an under- and over-approximating behavior. This indicates that the shuffling based parallel analysis may not be ideal for this setting.

First, consider that the number of ground-truth groups K is lower than the number of spiked eigenvalues $\hat{K}$ is higher. It is probable that some spiked eigenvalues are associated with patterns that are not related to a group structure, but rather through trivial correlations of the features. In the example setting of the `MNIST` digits, such an over-approximation can be partially explained by trivial correlations of the feature space obtained from encoding the 8 by 8 images in 64 dimensions, which is not invariant through horizontal or vertical translations of the digits. While this demonstrates a problematic application of this specific encoding, it also exposes the puzzle of disentangling features, space redundancies, and group structure from the spectral information of similarity matrices, which is known to be especially difficult in the context of internally non-uniform groups [37].

Second, consider that the number of ground-truth groups K is higher than the number of spiked eigenvalues $\hat{K}$ is lower. An intuition of how this may happen can be obtained by studying Figure 3.6. Here, the shuffling procedure slightly widens the bulk of the eigenvalues. While in the example, the spiked eigenvalues are not influenced by this wider bulk, if the spiked eigenvalues appear close to the bulk edge, as may happen in realistic data, the widened bulk may complicate the detection of the spiked eigenvalues.

However, while studying the `MNIST` digits, if we choose to consider more eigenvalues as 'spikes', the clustering performance does improve only slightly. Specifically, using an oracle value for the number of groups in the clustering methods improves the resulting clusterings slightly, in two ways. First, the clusterings using the oracle value K have only a slightly lower variation of information between the clusterings and the ground-truth. Second, the oracle based profile inference only detects one new profile, while it also constructs copies of profiles that were already inferred by the non-oracle method. This makes it difficult to be conclusive about shuffling based parallel analysis.

Existing Approaches

There are many existing methods to infer a threshold τ that are either more or less suitable than the shuffling based parallel analysis. The use of a specific procedure heavily depends on the context in which it is used. For example, shuffling is believed to be relatively well-behaved in the context of discrete data [37, 36], making it suitable for most of the settings we studied here. As seen in the experiments with multivariate Gaussian data in Chapter 10, the shuffling appeared to give an accurate approximation for the number of groups. This relation between parallel analysis and random matrix theory is studied in [110]. In [37] the author proposes the use of restricted randomness, i.e., a resampling procedure of categorical data instead of shuffling, which may pose as a more robust method of parallel analysis.

In the context of principal component analysis, much work has been done in the direction of detecting informative signals from uninformative signals. Along these lines, [111] promise to give arguably more accurate estimates of the number of spiked eigenvalues by using random matrix theory derived matching analysis. [112] proposes a parallel analysis method that, instead of shuffling features, flips the sign. This approach promises to be more robust in the case of heterogeneous noise within different clusters. However, this approach is only meaningful for ordinal data.

Theoretical Thresholds

When data is assumed to be Gaussian, the use of the Marchenko-Pastur law to determine the threshold may be appropriate, as is done in the original introduction of spectral modularity in [8]. It is therefore recommended to extend the theoretical threshold framework to a broader class of multivariate data. For example, for the setting of categorical data, a definition for a null model to obtain an exact threshold is not known yet. An important research direction is the study of the random Hamming similarity matrix. The reason for this is twofold.

First, the current shuffling based parallel analysis is a relatively well-suited method for categorical data because of the finiteness of the data space. This can give valuable insights into the alignment of the parallel analysis and the theoretical distribution. In positive results, it is tempting to think that parallel analysis is indeed a reasonable method to approximate K.

Second, a relatively simple extension of the Marchenko-Pastur law with block dependencies can be used, rendering the theoretical derivation simple. In particular, the complete disjunctive table of a categorical data matrix in itself does not have independent columns, however, [**bryson_marchenko-pastur_2021**] has shown that the MP law can be used under block dependencies.

Results related to theoretically deduced thresholds are inclined to coincide with algorithmic efficiency. In particular, as most of the computational burden appears to be in the computation of eigenvalues, eliminating the need to compute eigenvalues of shuffled matrices many times, greatly reduces the total computation time.

14

Recommended Developments

In this chapter, we discuss recommended developments that are beyond the scope of this thesis, but may improve current methods, resolve certain limitations, or provide additional insights into the open questions. Along these lines, there are at least three promising perspectives, that are yet to be studied relating to, the severity of spectral modularity breakdown, possible enhancements to regularized spectral modularity maximization (SMM1), and alternatives to normalized spectral modularity maximization (SMM2).

First, the severity of spectral modularity breakdown by studying the hierarchical structure of the inconsistent merges, discussed in Chapter 5. In particular, if the clusters in naive spectral modularity maximization were to merge hierarchically, which is unlikely, the breakdown is easily solved by existing methods. In [8] a multi-level approach to spectral modularity is introduced, that recursively splits clusters, which may appear to resolve the problem of inconsistent merges through positive off-diagonal elements in the group affinity matrix $\mathbf{G}$, as discussed in Section 5.2. However, while the group affinity matrix is demonstrative of the breaking of Q_0-consistency, in practice, the breakdown of spectral modularity may not occur by conveniently merging the clusters in a hierarchical pattern. This makes it unlikely that a multilevel approach resolves the spectral modularity breakdown, however, this is still an open question.

Second, the calibration of the regularization parameter in SMM1, which is described in Chapter 6, may be improved by making use of the seed interpretation from Chapter 7. Of the two contributed solutions to the spectral modularity breakdown, SMM1 is considerably less stable, and its clustering performance is therefore often surpassed by SMM2. A large part of this performance mismatch is likely to be explained by the instability of the calibration procedure. Using this procedure based on seeds promises to make the calibration, and therefore the performance of SMM1, more stable.

Third, there are a number of alternative interpretations of the normalized spectral modularity objective discussed in Chapter 7. In particular, the maximization procedure may be improved by adapting the assignment phase of the algorithm to resemble that of Lloyd's algorithm [56], which may give higher optima. Furthermore, the normalized spectral modularity objective Q_{norm} can be maximized with a gradient projection algorithm, which may provide a more flexible theoretical framework. Finally, an adaptation to SMM2 may give access to an intrinsic soft clustering method, which may provide more meaningful soft clusterings of the data compared to the method described in Chapter 8.

In Section 14.1, we describe the multi-level approach to spectral modularity [8] and highlight how it is unlikely to resolve the spectral modularity breakdown. In Section 14.2, we describe the seed-based alternative to Algorithm 2 and how it may result in more stable performance. In the remaining sections, we discuss the enhancements and alternatives to SMM2. In Section 14.3, we discuss the improved assignment phase of Algorithm 3. In Section 14.4, we discuss how Q_{norm} can be maximized with a gradient projection algorithm. In Section 14.5, we discuss how the intrinsic soft clustering method can be obtained within the scope of SMM2.

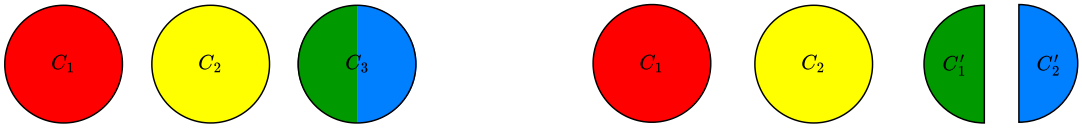


Figure 14.1: Illustration of ideal application of multi-level spectral modularity. The ground-truth partition $\rho^* = \{R, G, B, Y\}$ is indicated with the colors. Specifically, R is encoded with red, G is encoded with green, B is encoded with blue, and Y is encoded with yellow. On the left, we see the initial clustering $\rho = \{C_1, C_2, C_3\}$. In the right, we see the clustering of performing the second level of spectral modularity maximization, i.e., $\rho' = \{C_1, C_2, C_1', C_2'\}$. This clustering is exactly identical to the partition ρ.

14.1. Multi-level Spectral Modularity Breakdown

Upon the introduction of the spectral modularity as a way to cluster time series in [8], an important aspect of its potential was the recursive employment of the spectral modularity to deal with hierarchical group structures. The multi-level approach recursively applies the spectral modularity maximization to the subsets of the data that are found by the clustering. This way from a partition $\rho = \{C_1, \dots, C_K\}$. The spectral modularity procedure is applied to the data sub-matrices that belong to $C_1, \dots, C_K$ to obtain K partitions to create a refinement of ρ. Theoretically, the multi-level method will stop clustering when the similarity matrix of a specific cluster C_k has only one spiked eigenvalue, as this would correspond to no further significant group structure.

To illustrate the procedure, consider as an example a data set of four groups, i.e., $\rho^* = \{R, G, B, Y\}$, in which the maximization algorithm only finds three of the four clusters $\rho = \{C_1, C_2, C_3\}$. Specifically, $C_1 = R$, $C_2 = Y$ and $C_3 = G \cup B$. Then the multi level spectral modularity procedure would cluster the data again within each of the clusters. The goal of this procedure is to separate C_3 into two clusters, such that we retrieve a clustering that is equivalent to ρ^*. In Figure 14.1, the partitions in this example are illustrated.

At first glance, it may appear that the inconsistent merging caused by spectral modularity breakdown is to be corrected by the recursive employment of spectral modularity maximization. In particular, because the theoretical setting in which we suggest the breaking of the consistency of the modularity is based on depicting an inconsistency with the objective that favors inconsistent merges. Therefore, a multi-level spectral modularity approach may then, in turn, separate the two merged clusters.

However, in practice, modularity maximization algorithms rarely reach the ground-truth partition before (inconsistently) merging two clusters. Specifically, the breakdown of spectral modularity is likely to happen before entire clusters are merged. This way, the clusters are likely to be merged and broken in a non-hierarchical manner. The objects of some broken clusters will be distributed among other clusters.

To illustrate this, we consider again a ground-truth partition with $\rho^* = \{R, G, B, Y\}$. In particular, assume that there is a small positive spectral modularity between R and G and between groups R and B, but a relatively strong negative modularity between B and G. Then, it is unlikely that objects from B and G will be merged together. However, because of the small positive spectral modularity in the other two pairs, there will likely be some partial merges. Specifically, objects from R can be distributed among the two separate clusters that contain elements of G and B. This way, we obtain a clustering with 3 clusters, i.e., $\rho = \{C_1, C_2, C_3\}$. Here let $R = R_1 \cup R_2$ for disjoint sets R_1 and R_2. Then, consider $C_1 = R_1 \cup G$, $C_2 = Y$, and $C_3 = R_2 \cup B$. Then a multi-level approach may split the two merged groups, resulting in a partition with 5 clusters. But it will never reconstruct the original group R, and therefore will not recover the ground-truth partition. In Figure 14.2, the partitions of this example are illustrated.

This suggests that spectral modularity breakdown is unlikely to be resolved by this multi-level spectral modularity approach. However, further investigation in this direction is required to make this claim with certainty.

Figure 14.2: Illustration of non-ideal application of multi-level spectral modularity The ground-truth partition $\rho^* = \{R, G, B, Y\}$ is indicated with the colors. Specifically, R is encoded with red, G is encoded with green, B is encoded with blue, and Y is encoded with yellow. On the left, we see the initial clustering $\rho = \{C_1, C_2, C_3\}$. In the right, we see the clustering of performing the second level of spectral modularity maximization, i.e., $\rho' = \{C_2, C_1', C_2', C_1'', C_2''\}$. This clustering is not identical to the partition ρ.

On the other hand, the recursive application of spectral modularity maximization may still be of use in the context of explicit hierarchical groups, which we do not study in this thesis. Furthermore, an open question is whether the multi-level approach is effective in a setting where too few eigenvalues are spiked. Then, within each cluster, new spiked eigenvalues may appear, upon recursively applying spectral modularity. With this, the hierarchical structures may be detected effectively. The methods that we introduced, SMM1 and SMM2, can be equivalently used in the multi-level application of spectral modularity.

14.2. Improved Calibration Condition

The calibration procedure for the regularization parameter in chapter 6 has some limitations. First, the calibration procedure requires explicitly running the maximization algorithm for each evaluation of ϵ. Although theoretically the time complexity is dominated by the computation of the spectral decomposition, in practice the efficiency of the maximization algorithm implementation may be problematic.

Second, because of the relatively unexpected behavior that a greedy modularity maximization algorithm may have, the satisfaction of the calibration condition, from Section 6.3, can be rather unstable. For example, whenever a cluster is constructed that contains two elements, the cluster is counted as significant in the procedure. However, this grouping of two objects is, of course, not a particularly significant group. Therefore, to mitigate this, measures to gain robustness, such as increasing the number of elements before a group is counted as non-trivial, can be taken. This does, however, require parameter choices.

Fortunately, the seed finding procedure, discussed in Section 7.3, promises to give an alternative approach to determining the regularization parameter ϵ for SMM2. In fact, the procedure is likely to be more suitable as its calibration is less time-consuming and more stable. In particular, in the seed finding procedure, a similar threshold value is calibrated to ensure that there are $\hat{K}$ spectral modularity vectors of which the pairwise inner products are smaller than the threshold. This way, we would calibrate ϵ with a different condition that is based on the number of seeds, rather than the number of clusters. Because this condition is computationally much simpler, this procedure will be less time-consuming. Furthermore, the number of seeds behaves more predictably, which will lead to more stable approximations of the regularization parameter.

It is not entirely clear what specific value for ϵ is optimal. To be precise, the seed based calibration gives a larger region for values of ϵ, as it is not entirely clear that we want to use the smallest value for ϵ. This is in contrast to using the partition size calibration, where it is only reasonable to use the smallest value for ϵ that satisfies the constraint. A convenient solution may be to use a mid-point value between the smallest and largest value for ϵ that can be obtained through seed based calibration.

14.3. Alternating Assignment Algorithm

The SMM2 algorithm introduced in this thesis makes use of the ordering of the objects based on the magnitudes of the spectral modularity vectors. To be precise, the assignments of objects to clusters are considered only once, so objects with large magnitudes are considered earlier. Furthermore, the algorithm depends on the choice of seed objects, that are influential for the remainder of the clustering of the data. Although SMM2 often appears as the top-performing method in our performance analysis, it quite likely that the dependencies on the magnitude orderings and seeds may lead to problematic counterexamples. Therefore, it is recommended to consider alternative, tractable procedures that are free of these two dependencies.

One algorithm that potentially satisfies this may take inspiration from Lloyd's algorithm [56] that is used to solve the KMeans problem [89] and described in Section 2.3. In particular, the maximization goal of SMM2 can be alternatively approximated by alternating between two steps: an assignment step and an update step. Consider an initial position for cluster vectors $\{\mathbf{z}_k\}_{k=1}^{\hat{K}}$, which may be obtained from the seed procedure or randomly chosen. In the assignment step, each object, in an arbitrary order, is assigned to the cluster with the best aligned cluster vector, like in the assignment step of SMM2. In the update step, which is done when all objects are assigned, the cluster vectors are recomputed, like in the update step of SMM2. These two steps can be alternated until no improvement in the objective is obtained.

The essential difference between this Lloyd inspired method and SMM2 is that the objects are considered multiple times and the cluster vectors are adjusted multiple times. Therefore, the method does not depend on the ordering of magnitudes or the chosen seeds. Clearly, this method will require significantly more computational time than SMM2, which may pose potential problems. Also, it is unclear whether this method performs significantly better than SMM2. Therefore, further investigation of this method and its components in relation to SMM2 is recommended.

14.4. Gradient Projection

Using the matrix notation of the spectral modularity cluster profiles $\{\mathbf{z}_k\}_{k=1}^{\hat{K}} \subset \mathbb{R}^{\hat{K}-1}$ and the soft partition matrix, an alternative scheme to maximize the objective of SMM2. Specifically, we can use gradient based methods using the gradient of the objective w.r.t. the variable matrices. In particular, the gradient of $\operatorname{Tr}\left[\mathbf{P}\mathbf{R}\mathbf{Z}^\top\right]$ w.r.t. to $\mathbf{P}$ is $\mathbf{R}\mathbf{Z}^\top$ and the gradient w.r.t. $\mathbf{Z}$ is $\mathbf{P}\mathbf{R}$.

$$\max_{\mathbf{P}\in\mathcal{M},\{\mathbf{z}_k\}_{k=1}^{\hat{K}}\subset\mathcal{U}} \sum_{i=1}^{n} \mathbf{r}_i\mathbf{p}_i\mathbf{Z} = \max_{\mathbf{P}\in\mathcal{M},\{\mathbf{z}_k\}^{k=1}\subset\mathcal{U}} \operatorname{Tr}\left[\mathbf{R}\mathbf{P}\mathbf{Z}\right]. \tag{14.1}$$

Therefore, we can use a gradient projection procedure [113]. This method is a well-known heuristic for solving nonlinear, constrained optimization problems. The approach is based on gradient descent, or ascent, in our case, followed by a projection of the variable onto the constraint space. In particular, for some small value of $\alpha > 0$ and a fixed number of iterations N, we perform the following steps N times:

$$\mathbf{P} \leftarrow \mathbf{P} + \alpha\mathbf{R}\mathbf{Z}, \tag{14.2}$$
$$\text{project } \mathbf{P} \text{ onto } \mathcal{M}, \tag{14.3}$$
$$\mathbf{Z} \leftarrow \mathbf{Z} + \alpha\mathbf{R}\mathbf{P}, \tag{14.4}$$
$$\text{project } \mathbf{z}_k \text{ onto } \mathcal{U}_{K-1} \text{ for all } k \in \{1,\dots,K\}. \tag{14.5}$$

The projection of a matrix $\mathbf{P} \in \mathbb{R}^{n\times\hat{K}}$ onto $\mathcal{M}$ can be done efficiently by following the algorithm described in [**chen_projection_nodate**]. The projection of the vectors $\{\mathbf{z}_k\}_{k=1}^{\hat{K}}$ onto $\mathcal{U}$ can be done by simply normalizing the vectors.

At this point, little can be said about the theoretical convergence of the projected gradient method. Furthermore, this approach requires the choice of a parameter α and the specification of the number of iterations N. Finally, this specific gradient based method does not efficiently utilize symmetries in the partition matrix representation of partitions, which makes the method not ideal. However, the iterative improvement of the objective through the gradient projection may give additional insight into the behavior of the objective function.

14.5. Intrinsic Soft Clustering

The soft clustering method that is introduced in this thesis is based on spectral modularity vectors and is used to construct a related soft clustering given a hard clustering. Therefore, it is universally applicable given an arbitrary hard partition. On the one hand, this is a fundamental benefit of the method, but on the other hand, it may not make optimal use of the available information in the data. However, the soft clustering procedure can be incorporated with the maximization procedure of SMM2 as described in Algorithm 3 to possibly obtain an even more meaningful soft clustering of the data.

In particular, in the assignment phase of SMM2, it can be extended to encompass soft cluster assignments. We can replace the strict assignment of objects to clusters with a proportionally distributed membership of objects over the clusters. In the original form, the strict assignment is due to assignment to a maximum of the cosines in Chapter 8. However, for a single object, there may be more than one positive cosine associated with a cluster. Therefore, we can proportionally assign the object's membership to the clusters based on the size of the cosine relative to all the other positive cosines.

The necessary changes are based on the notation of a soft partition matrix $\mathbf{P}$, that is updated in each step of the assignment phase of Algorithm 3 with

$$\mathbf{P}_{ik} = \frac{(||\mathbf{r}_i|| \cos\theta_{ik})_+}{\sum_{h=1}^{\hat{K}} (||\mathbf{r}_i|| \cos\theta_{ih})_+}. \tag{14.6}$$

Furthermore, we can update the $\hat{K}$ cluster vectors in Algorithm 3, with

$$\mathbf{z}_k = \sum_{i=1}^{n} \mathbf{P}_{ik} \mathbf{r}_i. \tag{14.7}$$

Investigating whether this intrinsic soft clustering performs better in terms of profile inference is an open question. Furthermore, for the development of further enhancements in the direction of soft-clustering, it is advised to investigate other soft clustering evaluation methods rather than solely relying on profile inference.

15
Conclusion

The aim of this research is to demonstrate the challenges, required enhancements, and potency of spectral modularity maximization. To do this, we addressed research questions related to the performance of naive spectral modularity maximization, the challenges that arise with it, and how these challenges can be circumvented. The main theoretical contribution is the demonstration of a fundamental challenge of naive spectral modularity maximization, that appears when encountering clusterings with a relatively large number of groups. The main methodological contributions lie in the two proposed solutions that mitigate these problems and in the thorough investigation of their performance.

Cluster analysis gives insights into structures that are present in data, which can lead to further research directions. These insights are especially important when the datasets consist of objects with many dimensions. Specifically, because high dimensional data sets are typically difficult to interpret directly, However, many clustering algorithms fail in the context of high dimensional data, because of an abundance of non-representative information. Fortunately, spectral modularity cleverly filters informative structures from uninformative ones. This way, spectral modularity is suitable for high dimensional cluster analysis.

Using spectral modularity in the context of multivariate data is shown to be a relatively trustworthy method to uncover hidden group structure in the data. Especially when the group structure is relatively course, a naive maximization of spectral modularity, which we call SMM0, is capable of clustering the high dimensional data. However, when the group structure is fine-grained, i.e., there are many groups, the naive method faces a fundamental challenge. In particular, the naive method has a bias towards clusterings with a smaller number of groups, as a consequence of inconsistent merges. Fortunately, there are two solutions that can mitigate the challenges. First, we can use an explicit penalization of the bias, which we call SMM1. Second, we can remove the bias all together with a particular normalization, which we call SMM2. Both methods, have methodological advantages and disadvantages. However, empirically, SMM2 shows superior performance in most of our experiments.
Although the study of spectral modularity, and its enhancements, has shown significant potency as a clustering method for high dimensional data, there are still limitations to the method. A fundamental problem is the computational burden of the exact spectral decomposition. Although spectral modularity methods are designed for data in high dimensions, computationally, the requirement of eigenvalues and eigenvectors for data with many objects may present practical difficulties. Furthermore, the currently used procedure to approximate the number of clusters is only suitable in a relatively ideal setting. This may make the application of this procedure to real data suboptimal.

Finally, in addition to the improvements to the algorithmic efficiency and practical applicability of spectral modularity in general, there are recommended developments that are likely to improve the performance and understanding of the spectral modularity methods. First, the study of a multi-level spectral modularity approach will likely demonstrate the severity of the spectral modularity breakdown. Second, an improvement to the calibration of the regularization parameter in SMM1 will likely improve its clustering performance significantly. Third, enhancements to the maximization algorithm in SMM2 may lead to methods that attain higher performance, more flexibility, and intrinsic uncertainty quantification.

A

Prospective Papers

The scientific contribution of this thesis can be divided into three potential papers. First, we introduce the concept of spectral modularity breakdown in a setting that closely resembles the setting of the original introduction. Second, we extend the spectral modularity framework and evaluation from correlation matrices to more general similarity matrices, focusing on the mixed prototype data. Third, we extend the study towards methods for the detection of the number of clusters, inspired by the findings of Chapter 11.

The ordering of the articles, employed here, mostly aligns with the development timeline of the concepts. However, it may appear that the third paper related to the detection of the number of clusters is fundamental and should be worked on prior to the second paper, where we focus on spectral modularity methods in relatively complicated data. Therefore, the current ordering below does not necessarily reflect the most optimal order of work distribution.

A.1. Breakdown in Spectral Modularity of Correlation Matrices

The purpose of this paper is to stick relatively close to the setting in which the spectral modularity method is first introduced [8]. This means that instead of focusing on clustering with similarity matrices, we stick to the context of correlation matrices of time-series data. Therefore, we make use of the eigenvalue distributions of empirical correlation matrices of multivariate Gaussian distributions.

First, we demonstrate the theoretical understanding of spectral modularity breakdown, as is mainly discussed in Chapter 4. In order to do this, we follow the same argumentation with the help of a model of correlation matrices, that is similar to Toy Model A, and a model of spectral modularity vectors, similar to Toy Model B. The empirical analysis that is briefly done in Chapter 4 may be further extended with the purpose of further demonstrating the breakdown for different perturbations.

An important aspect of this demonstration in relation to the introduction of spectral modularity in [8] is the unsuitability of the recursive multi-level approach we discussed in Section 14.1. Also, the comparison of the spectral modularity breakdown and the resolution limit is considered with more care.

Second, we introduce the two solutions to overcome the spectral modularity breakdown that are introduced in this thesis. To this end, we discuss the regularized spectral modularity method (SMM1) as defined in Chapter 6 and the normalized spectral modularity method (SMM2) as defined in Chapter 7. For brevity, we leave out the soft clustering method discussed in Chapter 8 as this method is of higher importance in the mixed prototype data that is studied in Article A.2.

The performance of the methods is evaluated in the context of correlation matrices of time series. In particular, for synthetic data generation, we make use of multivariate Gaussian distributions with properly chosen covariance structures and/or perturbations of dynamical profiles. To be specific, we consider the perspective of a single Gaussian n-dimensional multivariate distribution covariance matrix that contains the correlation structure indicating the correlation of n individual time series. Then, p copies of this random variable provide us with a $n \times p$ data matrix, from which the sample correlation matrix approaches the correlation structure associated with the covariance matrix of the distribution if $p \to \infty$. Alternatively, we can consider the K prototypical time series $\{\mu_k\}_{k=1}^{K} \subset \mathbb{R}^p$ to obtain a mixture of multivariate Gaussian distributions where the kth distribution is centered at μ_k, and the n samples are standard perturbations around the respective prototypical time series.

Potentially, we may extend the mixture of multivariate Gaussian distributions with cluster-unique covariance matrices associated with the perturbations. However, this latter model may be problematic when inspecting the eigenvalues to determine $\hat{K}$ as it may lead to additional eigenvalues similar to what we observe in the clustering of the MNIST digits in Chapter 11. This aspect relates to the paper on the eigenvalue threshold discussed in Article A.3.

Additionally, we examine the clusterings obtained from our refined methods in the context of real empirical financial time-series data, as is done [8].

A.1.1. Abstract

Spectral modularity based clustering methods emerge from the theory of random matrices and graph modularity. By filtering the spectral decomposition of empirical correlation matrices, mesoscopic group structures of more strongly correlated objects can be detected. However, in this paper, we uncover a fundamental challenge of the spectral modularity that occurs as the number of groups is significantly large. To be precise, as the number of groups increases, the information in the correlation matrices is dominated by the inter-cluster correlations. Therefore, the spectral modularity becomes less and less effective in distinguishing the inter- and intra-cluster correlations. In order to resolve this breakdown of spectral modularity, we propose two solutions: one solution based on regularization and one solution based on normalization. We perform an empirical analysis of the clustering performance of the two solutions and find that, not only do our methods resolve the breakdown of spectral modularity, they also outperform existing clustering methods in a variety of settings.

A.2. Clustering Mixed Prototype Data with Spectral Modularity

The purpose of this paper is to extend the spectral modularity methods to more general similarity matrices. This is in correspondence with the main purpose of this thesis. However, because the spectral modularity breakdown and the enhancements to resolve the breakdown have already been discussed in Article A.1, the main part of this paper relies on the investigation of the spectral modularity method in the context of mixed prototype model data from Chapter 9.

Specifically, studying the setting of mixed prototype data is especially important for three reasons. First, the mixed prototype model has a rather deep contextual meaning in settings such as sociological and psychological data. This is because the mixed prototype model resembles a larger class of data that contains prototypical data profiles but not necessarily strict cluster assignments. Therefore, the study of this setting extends to a much broader class of cluster-like data. Second, a preliminary study shows that many existing clustering methods, such as hierarchical clustering, KMeans, KMedoids, and spectral clustering, have difficulty dealing with data that resembles mixed prototype models. Third, because the mixed prototype data exhibits strong overlaps, internal non-uniformities, and soft cluster boundaries, the spectral modularity breakdown is much more severe. This is mainly because the fluctuations of external spectral modularities are much larger than in settings where the clusters are typically well-separated, like in Gaussian mixture data.

Fortunately, the spectral modularity methods and especially the enhanced methods (SMM1 from Chapter 6, SMM2 from Chapter 7, and the soft clustering variant from Chapter 8) that overcome the spectral modularity breakdown are conceptually more aligned with the context of mixed prototype data. This proposition is verified in the results of this thesis in Chapter 10, and therefore will be compactly presented in the paper. In particular, certain limitations associated with the performance analysis as discussed in Chapter 13, such as the limited number of methods, will be taken into account in the paper. Several forms of empirical data with known ground-truths, like the handwritten digits and the soybeans data set from Chapter 11 are taken into account. Furthermore, human genome data, as studied in [6] may also be demonstrative of the practical applications of spectral modularity. Finally, we may examine empirical cultural data in the setting of [37, 38] because of the well understood resemblance to the synthetically generated mixed prototype data. This ground truth is not known, and therefore the evaluation can only be done qualitatively without going into too much detail on the sociological context.

A.2.1. Abstract

Cluster analysis typically assumes that objects in a data set can be strictly divided into distinct clusters. However, in practice, objects may be represented by multiple latent data profiles, which makes strict cluster assignments an ill-defined problem. In this setting, i.e., the mixed prototype setting, existing (soft) clustering methods have difficulty clustering the data reasonably, especially when the distribution of the objects in the latent prototype space is relatively concentrated at the center. Fortunately, (enhanced) spectral modularity based methods, which are based on the filtering of the spectral decomposition of correlation matrices, promise to be well suited in this setting. Here, we investigate the extension of existing spectral modularity to more general similarity matrices, in particular those representing mixed prototype data. To evaluate the spectral modularity methods, we employ systemic benchmarking of synthetically generated mixed prototype data and a selection of real empirical data. We find that problems associated with naive spectral modularity methods are amplified in the mixed prototype setting, underlining the significance of the use of the recent enhancements to spectral modularity. In particular, the conceptual overlap between the spectral modularity and the mixed prototype enables the enhanced spectral modularity methods to extract valuable information from the data set even when the mixing of prototypes is strong, which is a regime where existing methods tend to fail.

A.3. Eigenvalue Thresholds for Spectral Modularity

The purpose of this paper is to demonstrate the issue with established methods to detect the number of clusters. In particular, the shuffling based approach may not identify the right number of spiked eigenvalues. When considering the synthetically generated data, as is done for the main part of this thesis, the number of spiked eigenvalues is representative of the number of groups. However, when we consider realistic data sets, as in Chapter 11, we find that this equivalence becomes more ambiguous.

Determining the right number of clusters is an important part of cluster analysis. Methods based on detecting the number of spiked eigenvalues demonstrate robust and trustworthy behavior. While theoretical thresholding procedures exist, the methods that are more widely applicable are based on shuffling based parallel analysis. However, most of these methods are oriented to the setting of K independent data profiles that are representative of the associated cluster. In practice, the objects may have structurally correlated features that disappear through shuffling, providing additional spiked eigenvalues that represent these redundant features. Furthermore, cluster profiles may have structurally correlated features that do not disappear through shuffling, which absorbs the spiked eigenvalue into the bulk.

Taking the exposition of the challenge and preliminary intuition into consideration, this line of work requires more development in comparison to the Article A.1 and Article A.2.

Bibliography

[1] Christian Hennig. “What are the true clusters?” In: *Pattern Recognition Letters*. Philosophical Aspects of Pattern Recognition 64 (Oct. 2015), pp. 53–62. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2015.04.009.

[2] Iain M. Johnstone. *High Dimensional Statistical Inference and Random Matrices*. arXiv:math/0611589. Nov. 2006. DOI: 10.48550/arXiv.math/0611589.

[3] Christophe Giraud. *Introduction to High-Dimensional Statistics*. Aug. 2021. ISBN: 978-1-00-315874-5. DOI: 10.1201/9781003158745.

[4] Yang Tang, Ryan P. Browne, and Paul D. McNicholas. “Model Based Clustering of High-Dimensional Binary Data”. en. In: *Computational Statistics & Data Analysis* 87 (July 2015). arXiv:1404.3174 [stat], pp. 84–101. ISSN: 01679473. DOI: 10.1016/j.csda.2014.12.009.

[5] Pierpaolo D’Urso et al. “Clustering of financial time series”. In: *Physica A: Statistical Mechanics and its Applications* 392.9 (May 2013), pp. 2114–2129. ISSN: 0378-4371. DOI: 10.1016/j.physa.2013.01.027.

[6] Julia Gimbernat-Mayol et al. “Archetypal Analysis for population genetics”. en. In: *PLOS Computational Biology* 18.8 (Aug. 2022). Publisher: Public Library of Science, e1010301. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1010301.

[7] Maria Mircea et al. “Phiclust: a clusterability measure for single-cell transcriptomics reveals phenotypic subpopulations”. en. In: *Genome Biology* 23.1 (Jan. 2022), p. 18. ISSN: 1474-760X. DOI: 10.1186/s13059-021-02590-x.

[8] Mel MacMahon and Diego Garlaschelli. “Community Detection for Correlation Matrices”. en. In: *Physical Review X* 5.2 (Apr. 2015), p. 021006. ISSN: 2160-3308. DOI: 10.1103/PhysRevX.5.021006.

[9] Romain Couillet and Zhenyu Liao. *Random Matrix Methods for Machine Learning*. en. 1st ed. Cambridge University Press, July 2022. ISBN: 978-1-00-912849-0 978-1-00-912323-5. DOI: 10.1017/9781009128490.

[10] V. A. Marčenko and L. A. Pastur. “DISTRIBUTION OF EIGENVALUES FOR SOME SETS OF RANDOM MATRICES”. en. In: *Mathematics of the USSR-Sbornik* 1.4 (Apr. 1967). Publisher: IOP Publishing, p. 457. ISSN: 0025-5734. DOI: 10.1070/SM1967v001n04ABEH001994.

[11] Debashis Paul and Alexander Aue. “Random matrix theory in statistics: A review”. In: *Journal of Statistical Planning and Inference* 150 (July 2014), pp. 1–29. ISSN: 0378-3758. DOI: 10.1016/j.jspi.2013.09.005.

[12] M. E. J. Newman. “Finding community structure in networks using the eigenvectors of matrices”. en. In: *Physical Review E* 74.3 (Sept. 2006). arXiv:physics/0605087, p. 036104. ISSN: 1539-3755, 1550-2376. DOI: 10.1103/PhysRevE.74.036104.

[13] Ulrike Von Luxburg. “A tutorial on spectral clustering”. en. In: *Statistics and Computing* 17.4 (Dec. 2007), pp. 395–416. ISSN: 0960-3174, 1573-1375. DOI: 10.1007/s11222-007-9033-z.

[14] Andrew Ng, Michael Jordan, and Yair Weiss. “On Spectral Clustering: Analysis and an algorithm”. In: *Advances in Neural Information Processing Systems*. Vol. 14. MIT Press, 2001.

[15] Hongjie Jia et al. “The latest research progress on spectral clustering”. en. In: *Neural Computing and Applications* 24.7 (June 2014), pp. 1477–1486. ISSN: 1433-3058. DOI: 10.1007/s00521-013-1439-2.

[16] T. Shen. *The Mathematics Behind Spectral Clustering And The Equivalence To PCA*. arXiv:2103.00733 [cs, stat]. Feb. 2021. DOI: 10.48550/arXiv.2103.00733.

[17] Romain Couillet and Florent Benaych-Georges. “Understanding big data spectral clustering”. en. In: *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. Cancun, Mexico: IEEE, Dec. 2015, pp. 29–32. ISBN: 978-1-4799-1963-5. DOI: 10.1109/CAMSAP.2015.7383728.

[18] Romain Couillet and Florent Benaych-Georges. “Kernel spectral clustering of large dimensional data”. en. In: *Electronic Journal of Statistics* 10.1 (Jan. 2016). ISSN: 1935-7524. DOI: 10.1214/16-EJS1144.

[19] Noureddine El Karoui. “The spectrum of kernel random matrices”. In: *The Annals of Statistics* 38.1 (Feb. 2010). arXiv:1001.0492 [math, stat]. ISSN: 0090-5364. DOI: 10.1214/08-AOS648.

[20] Luis Aparicio et al. “A Random Matrix Theory Approach to Denoise Single-Cell Data”. In: *Patterns* 1.3 (June 2020), p. 100035. ISSN: 2666-3899. DOI: 10.1016/j.patter.2020.100035.

[21] Farhan Khawar and Nevin L. Zhang. “Cleaned Similarity for Better Memory-Based Recommenders”. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR’19. New York, NY, USA: Association for Computing Machinery, July 2019, pp. 1193–1196. ISBN: 978-1-4503-6172-9. DOI: 10.1145/3331184.3331310.

[22] V. Plerou et al. “A Random Matrix Approach to Cross-Correlations in Financial Data”. In: *Physical Review E* 65.6 (June 2002). arXiv:cond-mat/0108023, p. 066126. ISSN: 1063-651X, 1095-3787. DOI: 10.1103/PhysRevE.65.066126.

[23] Thomas Guhr and Bernd Kaelber. “A New Method to Estimate the Noise in Financial Correlation Matrices”. en. In: *Journal of Physics A: Mathematical and General* 36.12 (Mar. 2003). arXiv:cond-mat/0206577, pp. 3009–3032. ISSN: 0305-4470. DOI: 10.1088/0305-4470/36/12/310.

[24] M. Potters, J. P. Bouchaud, and L. Laloux. *Financial Applications of Random Matrix Theory: Old Laces and New Pieces*. arXiv:physics/0507111. July 2005. DOI: 10.48550/arXiv.physics/0507111.

[25] Ivailo I. Dimov et al. *Hidden Noise Structure and Random Matrix Models of Stock Correlations*. arXiv:0909.1383 [cond-mat, q-fin]. Dec. 2009.

[26] Joël Bun, Jean-Philippe Bouchaud, and Marc Potters. “Cleaning large correlation matrices: tools from random matrix theory”. en. In: *Physics Reports* 666 (Jan. 2017). arXiv:1610.08104 [cond-mat, q-fin, stat], pp. 1–109. ISSN: 03701573. DOI: 10.1016/j.physrep.2016.10.005.

[27] Laurent Laloux et al. “Noise Dressing of Financial Correlation Matrices”. en. In: *Physical Review Letters* 83.7 (Aug. 1999). arXiv:cond-mat/9810255, pp. 1467–1470. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.83.1467.

[28] Maxim Kazakov and Valery A. Kalyagin. “Spectral Properties of Financial Correlation Matrices”. en. In: *Models, Algorithms and Technologies for Network Analysis*. Ed. by Valery A. Kalyagin, Petr A. Koldanov, and Panos M. Pardalos. Cham: Springer International Publishing, 2016, pp. 135–156. ISBN: 978-3-319-29608-1. DOI: 10.1007/978-3-319-29608-1_9.

[29] Ali Namaki et al. “Analysis of the Global Banking Network by Random Matrix Theory”. In: (July 2020). DOI: 10.3389/fphy.2020.586561.

[30] Olivier Ledoit and Michael Wolf. “Improved estimation of the covariance matrix of stock returns with an application to portfolio selection”. In: *Journal of Empirical Finance* 10.5 (Dec. 2003), pp. 603–621. ISSN: 0927-5398. DOI: 10.1016/S0927-5398(03)00007-0.

[31] Ioannis Anagnostou et al. “Uncovering the mesoscale structure of the credit default swap market to improve portfolio risk modelling”. In: *Quantitative Finance* 21.9 (Sept. 2021). arXiv:2006.03014 [q-fin], pp. 1501–1518. ISSN: 1469-7688, 1469-7696. DOI: 10.1080/14697688.2021.1890807.

[32] Olivier Ledoit and Michael Wolf. “Nonlinear shrinkage estimation of large-dimensional covariance matrices”. In: *The Annals of Statistics* 40.2 (Apr. 2012). Publisher: Institute of Mathematical Statistics, pp. 1024–1060. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/12-AOS989.

[33] Olivier Ledoit and Michael Wolf. “Spectrum estimation: A unified framework for covariance matrix estimation and PCA in large dimensions”. In: *Journal of Multivariate Analysis* 139 (July 2015), pp. 360–384. ISSN: 0047-259X. DOI: 10.1016/j.jmva.2015.04.006.

[34] Jinho Baik, Gérard Ben Arous, and Sandrine Péché. “Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices”. In: *The Annals of Probability* 33.5 (Sept. 2005). Publisher: Institute of Mathematical Statistics, pp. 1643–1697. ISSN: 0091-1798, 2168-894X. DOI: 10.1214/009117905000000233.

[35] Paolo Barucca, Mario Kieburg, and Alexander Ossipov. “Eigenvalue and Eigenvector Statistics in Time Series Analysis”. en. In: *EPL (Europhysics Letters)* 129.6 (Apr. 2020). arXiv:1904.05079 [cond-mat], p. 60003. ISSN: 1286-4854. DOI: 10.1209/0295-5075/129/60003.

[36] Aashay Patil and M. S. Santhanam. “Random matrix approach to multivariate categorical data analysis”. en. In: *Physical Review E* 92.3 (Sept. 2015). arXiv:1503.06559 [physics], p. 032130. ISSN: 1539-3755, 1550-2376. DOI: 10.1103/PhysRevE.92.032130.

[37] Alexandru-Ionuț Băbeanu. “A random matrix perspective of cultural structure: groups or redundancies?” In: *Journal of Physics: Complexity* 2 (June 2021). Publisher: IOP ADS Bibcode: 2021JPCom...2b5008B, p. 025008. ISSN: 2632-072X. DOI: 10.1088/2632-072X/abc859.

[38] Alexandru-Ionu□ Băbeanu and Diego Garlaschelli. “Evidence for mixed rationalities in preference formation”. In: *Complexity* 2018 (2018). Publisher: Hindawi Limited, pp. 1–19.

[39] V. A. Traag, L. Waltman, and N. J. van Eck. “From Louvain to Leiden: guaranteeing well-connected communities”. en. In: *Scientific Reports* 9.1 (Mar. 2019). Number: 1 Publisher: Nature Publishing Group, p. 5233. ISSN: 2045-2322. DOI: 10.1038/s41598-019-41695-z.

[40] Marina Meilă. “Comparing Clusterings by the Variation of Information”. en. In: *Learning Theory and Kernel Machines*. Ed. by Gerhard Goos et al. Vol. 2777. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 173–187. ISBN: 978-3-540-40720-1 978-3-540-45167-9. DOI: 10.1007/978-3-540-45167-9_14.

[41] Bao Chong. “K-means clustering algorithm: a brief review”. en. In: *Academic Journal of Computing & Information Science* 4.5 (Sept. 2021). Publisher: Francis Academic Press. DOI: 10.25236/AJCIS.2021.040506.

[42] Hae-Sang Park and Chi-Hyuck Jun. “A simple and fast algorithm for K-medoids clustering”. In: *Expert Systems with Applications* 36.2, Part 2 (Mar. 2009), pp. 3336–3341. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2008.01.039.

[43] Debashis Paul. “Asymptotics of Sample Eigenstructure for a Large Dimensional Spiked Covariance Model”. In: *Statistica Sinica* 17.4 (2007). Publisher: Institute of Statistical Science, Academia Sinica, pp. 1617–1642. ISSN: 1017-0405.

[44] J. L. Horn. “A RATIONALE AND TEST FOR THE NUMBER OF FACTORS IN FACTOR ANALYSIS”. eng. In: *Psychometrika* 30 (June 1965), pp. 179–185. ISSN: 0033-3123. DOI: 10.1007/BF02289447.

[45] R.L. Chilausky R.S. Michalski. *Soybean (Large)*. 1980. DOI: 10.24432/C5JG6Z.

[46] William M. Rand. “Objective Criteria for the Evaluation of Clustering Methods”. In: *Journal of the American Statistical Association* 66.336 (Dec. 1971). Publisher: Taylor & Francis _eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1971.10482356, pp. 846–850. ISSN: 0162-1459. DOI: 10.1080/01621459.1971.10482356.

[47] E. B. Fowlkes and C. L. Mallows. “A Method for Comparing Two Hierarchical Clusterings”. In: *Journal of the American Statistical Association* 78.383 (Sept. 1983). Publisher: Taylor & Francis _eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1983.10478008, pp. 553–569. ISSN: 0162-1459. DOI: 10.1080/01621459.1983.10478008.

[48] Lawrence Hubert and Phipps Arabie. “Comparing partitions”. en. In: *Journal of Classification* 2.1 (Dec. 1985), pp. 193–218. ISSN: 1432-1343. DOI: 10.1007/BF01908075.

[49] Dongkuan Xu and Yingjie Tian. “A Comprehensive Survey of Clustering Algorithms”. en. In: *Annals of Data Science* 2.2 (June 2015), pp. 165–193. ISSN: 2198-5812. DOI: 10.1007/s40745-015-0040-1.

[50] Paul D. McNicholas. “Model-Based Clustering”. en. In: *Journal of Classification* 33.3 (Oct. 2016), pp. 331–373. ISSN: 1432-1343. DOI: 10.1007/s00357-016-9211-9.

[51] Man-Suk Oh and Adrian E Raftery. “Model-Based Clustering With Dissimilarities: A Bayesian Approach”. en. In: *Journal of Computational and Graphical Statistics* 16.3 (Sept. 2007), pp. 559–585. ISSN: 1061-8600, 1537-2715. DOI: 10.1198/106186007X236127.

[52] S. Wade. “Bayesian cluster analysis”. en. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 381.2247 (May 2023), p. 20220149. ISSN: 1364-503X, 1471-2962. DOI: 10.1098/rsta.2022.0149.

[53] David Blei, Andrew Ng, and Michael Jordan. “Latent Dirichlet Allocation”. In: *Advances in Neural Information Processing Systems*. Vol. 14. MIT Press, 2001.

[54] Joerg Sander. “Density-Based Clustering”. en. In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, 2010, pp. 270–273. ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8_211.

[55] Poornachandra Sarang. “Connectivity-Based Clustering”. en. In: *Thinking Data Science: A Data Science Practitioner’s Guide*. Ed. by Poornachandra Sarang. Cham: Springer International Publishing, 2023, pp. 185–195. ISBN: 978-3-031-02363-7. DOI: 10.1007/978-3-031-02363-7_10.

[56] S. Lloyd. “Least squares quantization in PCM”. In: *IEEE Transactions on Information Theory* 28.2 (Mar. 1982). Conference Name: IEEE Transactions on Information Theory, pp. 129–137. ISSN: 1557-9654. DOI: 10.1109/TIT.1982.1056489.

[57] Christian Bauckhage. *k-Means Clustering Is Matrix Factorization*. arXiv:1512.07548 [stat]. Dec. 2015. DOI: 10.48550/arXiv.1512.07548.

[58] Kenichi Kurihara and Max Welling. “Bayesian k-Means as a ”maximization-expectation” algorithm”. eng. In: *Neural Computation* 21.4 (Apr. 2009), pp. 1145–1172. ISSN: 0899-7667. DOI: 10.1162/neco.2008.12-06-421.

[59] Leonard Kaufman and Peter Rousseeuw. *Finding Groups in Data: An Introduction To Cluster Analysis*. Journal Abbreviation: Wiley, New York. ISBN 0-471-87876-6. Publication Title: Wiley, New York. ISBN 0-471-87876-6. Jan. 1990. ISBN: 978-0-471-87876-6. DOI: 10.2307/2532178.

[60] Frank Nielsen. “Hierarchical Clustering”. en. In: *Introduction to HPC with MPI for Data Science*. Ed. by Frank Nielsen. Cham: Springer International Publishing, 2016, pp. 195–211. ISBN: 978-3-319-21903-5. DOI: 10.1007/978-3-319-21903-5_8.

[61] Miroslav Fiedler. “Algebraic connectivity of graphs”. en. In: *Czechoslovak Mathematical Journal* 23.2 (1973), pp. 298–305. ISSN: 0011-4642, 1572-9141. DOI: 10.21136/CMJ.1973.101168.

[62] Jianbo Shi and Jitendra Malik. “Normalized Cuts and Image Segmentation”. en. In: *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* 22.8 (2000).

[63] Mikhail Belkin and Partha Niyogi. “Laplacian Eigenmaps for Dimensionality Reduction and Data Representation”. en. In: *Neural Computation* 15.6 (June 2003), pp. 1373–1396. ISSN: 0899-7667, 1530-888X. DOI: 10.1162/089976603321780317.

[64] Richard Bellman. *Dynamic Programming*. en. Google-Books-ID: wdtoPwAACAAJ. Princeton University Press, 1957. ISBN: 978-0-691-07951-6.

[65] Olivier Ledoit and Michael Wolf. *Honey, I Shrunk the Sample Covariance Matrix*. en. SSRN Scholarly Paper. Rochester, NY, June 2003. DOI: 10.2139/ssrn.433840.

[66] Noirrit Kiran Chandra, Antonio Canale, and David B. Dunson. *Escaping the curse of dimensionality in Bayesian model based clustering*. en. arXiv:2006.02700 [stat]. Nov. 2022.

[67] Charles Bouveyron and Camille Brunet-Saumard. “Model-based clustering of high-dimensional data: A review”. In: *Computational Statistics & Data Analysis* 71 (Mar. 2014), pp. 52–78. ISSN: 0167-9473. DOI: 10.1016/j.csda.2012.12.008.

[68] Mamta Mittal et al. “Clustering approaches for high-dimensional databases: A review”. en. In: *WIREs Data Mining and Knowledge Discovery* 9.3 (2019), e1300. ISSN: 1942-4795. DOI: 10.1002/widm.1300.

[69] Qin Xu et al. “PCA-guided search for K-means”. In: *Pattern Recognition Letters* 54 (Dec. 2014). DOI: 10.1016/j.patrec.2014.11.017.

[70] J. Fernando Vera and Rodrigo Macías. “On the Behaviour of K-Means Clustering of a Dissimilarity Matrix by Means of Full Multidimensional Scaling”. en. In: *Psychometrika* 86.2 (June 2021), pp. 489–513. ISSN: 1860-0980. DOI: 10.1007/s11336-021-09757-2.

[71] Lai Wei, Weiming Zeng, and Hong Wang. *K-means clustering with manifold*. Pages: 2099. Aug. 2010. DOI: 10.1109/FSKD.2010.5569712.

[72] Giacomo Livan, Marcel Novaes, and Pierpaolo Vivo. *Introduction to Random Matrices - Theory and Practice*. en. Vol. 26. arXiv:1712.07903 [cond-mat, physics:math-ph]. 2018. DOI: 10.1007/978-3-319-70885-0.

[73] JOHN WISHART. “THE GENERALISED PRODUCT MOMENT DISTRIBUTION IN SAMPLES FROM A NORMAL MULTIVARIATE POPULATION”. In: *Biometrika* 20A.1-2 (Dec. 1928), pp. 32–52. ISSN: 0006-3444. DOI: 10.1093/biomet/20A.1-2.32.

[74] Florent Benaych-Georges and Raj Rao Nadakuditi. “The singular values and vectors of low rank perturbations of large rectangular random matrices”. In: *Journal of Multivariate Analysis* 111 (Oct. 2012), pp. 120–135. ISSN: 0047-259X. DOI: 10.1016/j.jmva.2012.04.019.

[75] Alexander Soshnikov. *A note on universality of the distribution of the largest eigenvalues in certain sample covariance matrices*. arXiv:math/0104113. June 2001. DOI: 10.48550/arXiv.math/0104113.

[76] Jennifer Bryson, Roman Vershynin, and Hongkai Zhao. “Marchenko–Pastur law with relaxed independence conditions”. In: *Random Matrices: Theory and Applications* 10.04 (Oct. 2021). Publisher: World Scientific Publishing Co., p. 2150040. ISSN: 2010-3263. DOI: 10.1142/S2010326321500404.

[77] Stuart Geman. “A Limit Theorem for the Norm of Random Matrices”. In: *The Annals of Probability* 8.2 (Apr. 1980). Publisher: Institute of Mathematical Statistics, pp. 252–261. ISSN: 0091-1798, 2168-894X. DOI: 10.1214/aop/1176994775.

[78] Iain M. Johnstone. “On the distribution of the largest eigenvalue in principal components analysis”. In: *The Annals of Statistics* 29.2 (Apr. 2001). Publisher: Institute of Mathematical Statistics, pp. 295–327. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/aos/1009210544.

[79] Craig A. Tracy and Harold Widom. *Distribution functions for largest eigenvalues and their applications*. arXiv:math-ph/0210034. Nov. 2002. DOI: 10.48550/arXiv.math-ph/0210034.

[80] Jiashun Jin, Zheng Tracy Ke, and Wanjie Wang. *Phase Transitions for High Dimensional Clustering and Related Problems*. arXiv:1502.06952 [math, stat]. June 2016. DOI: 10.48550/arXiv.1502.06952.

[81] Edgar Dobriban. “Permutation methods for factor analysis and PCA”. In: *The Annals of Statistics* 48.5 (Oct. 2020). Publisher: Institute of Mathematical Statistics, pp. 2824–2847. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/19-AOS1907.

[82] Mingming Chen, Konstantin Kuzmin, and Boleslaw K. Szymanski. “Community Detection via Maximization of Modularity and Its Variants”. en. In: *IEEE Transactions on Computational Social Systems* 1.1 (Mar. 2014). arXiv:1507.00787 [physics], pp. 46–65. ISSN: 2329-924X. DOI: 10.1109/TCSS.2014.2307458.

[83] U. Brandes et al. *Maximizing Modularity is hard*. arXiv:physics/0608255. Aug. 2006. DOI: 10.48550/arXiv.physics/0608255.

[84] Yorick Hardy and Willi-Hans Steeb. “Kronecker Product”. en. In: *Matrix Calculus, Kronecker Product and Tensor Product*. 3rd ed. WORLD SCIENTIFIC, Apr. 2019, pp. 107–178. ISBN: 9789811202513 9789811202520. DOI: 10.1142/9789811202520_0002.

[85] Robin Chapman (https://mathoverflow.net/users/4213/robin-chapman). *Largest number of vectors with pairwise negative dot product*. https://mathoverflow.net/q/31440 Published: MathOverflow. Nov. 2010.

[86] Sławomir T. Wierzchoń and Mieczysław A. Kłopotek. “Algorithms of Combinatorial Cluster Analysis”. en. In: *Modern Algorithms of Cluster Analysis*. Ed. by Slawomir Wierzchoń and Mieczyslaw Kłopotek. Cham: Springer International Publishing, 2018, pp. 67–161. ISBN: 978-3-319-69308-8. DOI: 10.1007/978-3-319-69308-8_3.

[87] Ulrike von Luxburg, Mikhail Belkin, and Olivier Bousquet. “Consistency of spectral clustering”. en. In: *The Annals of Statistics* 36.2 (Apr. 2008). arXiv:0804.0678 [math, stat]. ISSN: 0090-5364. DOI: 10.1214/009053607000000640.

[88] Marek Gagolewski. “A framework for benchmarking clustering algorithms”. In: *SoftwareX* 20 (Dec. 2022), p. 101270. ISSN: 2352-7110. DOI: 10.1016/j.softx.2022.101270.

[89] Pasi Fränti and Sami Sieranoja. “K-means properties on six clustering benchmark datasets”. en. In: *Applied Intelligence* 48.12 (Dec. 2018), pp. 4743–4759. ISSN: 1573-7497. DOI: 10.1007/s10489-018-1238-7.

[90] J. Aitchison and S. M. Shen. “Logistic-Normal Distributions: Some Properties and Uses”. In: *Biometrika* 67.2 (1980). Publisher: [Oxford University Press, Biometrika Trust], pp. 261–272. ISSN: 0006-3444. DOI: 10.2307/2335470.

[91] David W. Scott. “On optimal and data-based histograms”. In: *Biometrika* 66.3 (Dec. 1979), pp. 605–610. ISSN: 0006-3444. DOI: 10.1093/biomet/66.3.605.

[92] L. Hagen and A.B. Kahng. “New spectral methods for ratio cut partitioning and clustering”. en. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 11.9 (Sept. 1992), pp. 1074–1085. ISSN: 02780070. DOI: 10.1109/43.159993.

[93] Chris Ding et al. “Spectral min-max cut for graph partitioning and data clustering”. en. In: (May 2001).

[94] Lazhar Labiod, Nistor Grozavu, and Younèns Bennani. “Clustering Categorical Data Using an Extended Modularity Measure”. en. In: *Neural Information Processing. Models and Applications*. Ed. by Kok Wai Wong, B. Sumudu U. Mendis, and Abdesselam Bouzerdoum. Vol. 6444. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 310–320. ISBN: 978-3-642-17533-6 978-3-642-17534-3. DOI: 10.1007/978-3-642-17534-3_38.

[95] Rong Wang. “Normalizing Modularity Matrices for Data Clustering”. In: *2011 Seventh International Conference on Computational Intelligence and Security*. Dec. 2011, pp. 1328–1330. DOI: 10.1109/CIS.2011.295.

[96] Santo Fortunato and Marc Barthélemy. “Resolution limit in community detection”. In: *Proceedings of the National Academy of Sciences* 104.1 (Jan. 2007). Publisher: Proceedings of the National Academy of Sciences, pp. 36–41. DOI: 10.1073/pnas.0605965104.

[97] Zheng Tian, XiaoBin Li, and YanWei Ju. “Spectral clustering based on matrix perturbation theory”. en. In: *Science in China Series F: Information Sciences* 50.1 (Feb. 2007), pp. 63–81. ISSN: 1862-2836. DOI: 10.1007/s11432-007-0007-8.

[98] W. E. Donath and A. J. Hoffman. “Lower Bounds for the Partitioning of Graphs”. In: *IBM Journal of Research and Development* 17.5 (Sept. 1973). Conference Name: IBM Journal of Research and Development, pp. 420–425. ISSN: 0018-8646. DOI: 10.1147/rd.175.0420.

[99] Florent Benaych-Georges and Romain Couillet. “Spectral analysis of the Gram matrix of mixture models”. fr. In: *ESAIM: Probability and Statistics* 20 (2016), pp. 217–237. ISSN: 1262-3318. DOI: 10.1051/ps/2016007.

[100] Marianna Bolla. “Penalized versions of the Newman-Girvan modularity and their relation to normalized cuts and k-means clustering”. In: *Physical Review E* 84.1 (July 2011). Publisher: American Physical Society, p. 016108. DOI: 10.1103/PhysRevE.84.016108.

[101] Scott White and Padhraic Smyth. “A Spectral Clustering Approach To Finding Communities in Graphs”. en. In: *Proceedings of the 2005 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, Apr. 2005, pp. 274–285. ISBN: 978-0-89871-593-4 978-1-61197-275-7. DOI: 10.1137/1.9781611972757.25.

[102] Xiao Zhang and M. E. J. Newman. “Multiway spectral community detection in networks”. en. In: *Physical Review E* 92.5 (Nov. 2015). arXiv:1507.05108 [cond-mat, physics:physics], p. 052808. ISSN: 1539-3755, 1550-2376. DOI: 10.1103/PhysRevE.92.052808.

[103] Zhexue Huang and M.K. Ng. “A fuzzy k-modes algorithm for clustering categorical data”. en. In: *IEEE Transactions on Fuzzy Systems* 7.4 (Aug. 1999), pp. 446–452. ISSN: 10636706. DOI: 10.1109/91.784206.

[104] Adele Cutler and Leo Breiman. “Archetypal Analysis”. In: *Technometrics* 36.4 (1994). Publisher: [Taylor & Francis, Ltd., American Statistical Association, American Society for Quality], pp. 338–347. ISSN: 0040-1706. DOI: 10.2307/1269949.

[105] Mousa AL-Akhras. “An Efficient Fuzzy K-Medoids Method”. In: *World Applied Sciences Journal* 10 (Jan. 2010), pp. 574–583.

[106] Sohan Seth and Manuel J. A. Eugster. *Probabilistic Archetypal Analysis*. arXiv:1312.7604 [stat]. Apr. 2014. DOI: 10.48550/arXiv.1312.7604.

[107] Hengyuan Zhang et al. “Fuzzy community detection via modularity guided membership-degree propagation”. In: *Pattern Recognition Letters* 70 (Jan. 2016), pp. 66–72. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2015.11.008.

[108] Jian Liu. “Fuzzy modularity and fuzzy community structure in networks”. en. In: *The European Physical Journal B* 77.4 (Oct. 2010), pp. 547–557. ISSN: 1434-6036. DOI: 10.1140/epjb/e2010-00290-3.

[109] Peter Macgregor. *Fast and Simple Spectral Clustering in Theory and Practice*. arXiv:2310.10939 [cs]. Oct. 2023. DOI: 10.48550/arXiv.2310.10939.

[110] Edoardo Saccenti and Marieke E. Timmerman. “Considering Horn’s Parallel Analysis from a Random Matrix Theory Point of View”. en. In: *Psychometrika* 82.1 (Mar. 2017), pp. 186–209. ISSN: 0033-3123, 1860-0980. DOI: 10.1007/s11336-016-9515-z.

[111] Zheng Tracy Ke, Yucong Ma, and Xihong Lin. “Estimation of the Number of Spiked Eigenvalues in a Covariance Matrix by Bulk Eigenvalue Matching Analysis”. In: *Journal of the American Statistical Association* 118.541 (Jan. 2023), pp. 374–392. ISSN: 0162-1459. DOI: 10.1080/01621459.2021.1933497.

[112] David Hong, Yue Sheng, and Edgar Dobriban. *Selecting the number of components in PCA via random signflips*. arXiv:2012.02985 [math, stat]. May 2024. DOI: 10.48550/arXiv.2012.02985.

[113] Youwei Liang. *Gradient Projection for Solving Quadratic Programs with Standard Simplex Constraints*. arXiv:2006.06934 [math]. July 2020. DOI: 10.48550/arXiv.2006.06934.