Statistical Inference for Clustering

Lucy L. Gao

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Daniela Witten, Chair

Marina Meila

Ali Shojaie

Program Authorized to Offer Degree:
Department of Biostatistics

University of Washington

Abstract

Statistical Inference for Clustering

Lucy L. Gao

Chair of the Supervisory Committee:
Dr. Daniela Witten
Statistics and Biostatistics

In this dissertation, we develop new methods for statistical inference in the context of single-view and multi-view clustering. In the first two chapters, we consider the multi-view data setting, where multiple data sets are collected from a common set of features. We propose tests of independence between the cluster membership variables in each data view that can be applied to any combination of multivariate and network data views. In the third chapter, we propose a test of no difference in means between two clusters obtained from hierarchical clustering.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisor, Daniela Witten, for her unwavering and wholehearted dedication to my success; I really could not have asked for a better mentor. I would also like to thank my mentor, Julie Zhou, for putting me on the path of research, and my collaborator, Jacob Bien for his unfailing enthusiasm and encouragement. I am very grateful to my committee members, Marina Meila and Ali Shojaie, for their input on this dissertation. Finally, I sincerely thank my family and friends for their support and love.

DEDICATION

to my parents, Jane J. Ye and Hong Gao

Chapter 1

INTRODUCTION

Cluster analysis is the task of dividing observations or features into representative subgroups, or *clusters*. This is a fundamental task in data analyses across virtually all applied domains. For example, cluster analysis is often used in cancer genomics to identify subgroups of tumors that may correspond to hitherto unknown cancer subtypes [89, 100, 112, 13, 63, 108].

Over the years, many clustering methods have been proposed for a variety of data types and data settings [116]. A prominent contemporary data setting is the *multi-view* data setting [104, 65], where multiple data types, or *views*, (e.g. gene expression data and DNA copy number data) are available on a single, common set of observations (e.g. tumor samples). Many recent papers have proposed clustering methods in this multi-view data setting [9, 94, 98, 58, 56, 69, 33]. These multi-view clustering methods have enjoyed a burst of popularity in scientific applications, likely because scientific phenomena often have dynamics that cannot be fully captured by any single data type. For example, cancer has been linked to differences across several different "omes", including the genome, transcriptome, and proteome [110, 37, 3].

Much of the single-view and multi-view clustering literature focuses on issues related to cluster *estimation*. For example, virtually all clustering methods will return either an estimated clustering of the observations, or estimated cluster membership probabilities for the observations. However, we may instead be interested in performing *inference* on the estimated clusters. For example, instead of simply estimating cluster memberships of tumor samples on the basis of their gene expression data, we may additionally wish to determine whether these estimated clusters truly correspond to different patterns of gene expression.

In this dissertation, we develop new methods for statistical inference in the context of clustering problems in the single-view and multi-view data settings. The first two chap-

ters address the problem of inference for multi-view clustering methods; the first chapter considers multivariate data views, where the lth data view contains p_l features, and the second chapter considers network data views, where the lth data view is a network with edges describing a type of relationship or interaction between the observations. The third chapter addresses the problem of inference for agglomerative hierarchical clustering, which is a classic single-view clustering method.

In Chapter 2, we consider the setting where we have several multivariate data views on a single, common set of observations. In this setting, many papers on multi-view clustering methods assume that there is a single clustering underlying all of the data views, and propose methods that "borrow strength" across data views to more accurately estimate that clustering [9, 98, 58, 56, 69]. This implicitly assumes that there exists a single clustering underlying all of the data views, or at least that the clusterings underlying each data view are closely related. Before applying a multi-view clustering method that makes this assumption, it is important to be able to evaluate whether the assumption holds. Some papers take another approach, where they define separate clusterings of the observations with respect to different data views, and propose methods for estimating the relationship between the clusterings within each view [94, 33]. However, these papers do not provide a way to do inference on the relationship between the clusterings within each view.

Thus, we extend the finite mixture model [76] for clustering a single data view to the setting of multiple data views, without assuming that the cluster membership variables in each data view are dependent or identical. Within this framework, we propose a test of the null hypothesis that the cluster membership variables within each data view are independent. This allows us to evaluate the assumption that there exists a relationship between the clusterings of the observations defined with respect to each data view.

In Chapter 3, we consider the setting where we have two network data views on a single, common set of nodes. Much like the setting where there are several multivariate data views on a common set of observations, many papers on multi-view clustering methods for network data sets assume that the clusterings underlying each network are closely related [38, 88, 102, 12, 101]. This motivates us to develop methods for statistical inference on the relationship between the clusterings within each network, so that we can evaluate the

assumption that the clusterings of the nodes defined with respect to each network are related. We extend the stochastic block model [45] for clustering a single network data view to the setting of multiple networks, without assuming that the networks are closely related. Within this framework, we propose a test of the null hypothesis that the cluster membership variables within each data view are independent. We also consider a related setting, where we have a network on a set of nodes as well as covariate measurements on each node. This can be viewed as a network data view and a multivariate data view, and we adapt our methodology to propose a test of the null hypothesis that cluster memberships defined with respect to the network and cluster memberships defined with respect to the covariates are independent.

In Chapter 4, we consider the setting where we have a single data view (data set), and we cut the agglomerative hierarchical clustering dendrogram of the data set to get K estimated clusters. We define the "mean" of an estimated cluster to be the centroid of the mean vectors corresponding to the observations in each estimated cluster, and ask: is there a difference in means between two estimated clusters? This is a challenging question, because the null hypothesis depends on the data. Thus, it is inappropriate to apply classical techniques for testing for a difference in means between two groups, such as Hotelling's T-test to answer this question, because they do not account for this hypothesis selection. To answer this question, we make use of the extensive recent literature on *selective inference* [29, 70, 107, 61, 118, 49, 50, 20, 51] in order to develop a test of the null hypothesis that two estimated clusters have no difference in means, which properly accounts for the fact that we estimated the clusters from the data.

Chapter 2

ARE CLUSTERINGS OF MULTIPLE DATA VIEWS INDEPENDENT?

This work is published in *Biostatistics* [34].

2.1 Introduction

Complex biological systems consist of diverse components with dynamics that may vary over time, and so these sytems often cannot be fully characterized by any single type of data, or at any single snapshot in time. Consequently, it has become increasingly common for researchers to collect multiple data sets, or *views*, for a single set of observations.

Multiple-view data has been applied extensively to characterize disease, such as in The Cancer Genome Atlas Project [15]. In contrast, The Pioneer 100 (P100) Wellness Project [90] collected multiple-view data from healthy participants to characterize wellness, and to optimize wellness of the participants through personalized healthcare recommendations. One way to do this is to identify subgroups of similar participants using cluster analysis, and then tailor recommendations to each subgroup.

In recent years, many papers have proposed clustering methods in the multiple-view data setting [9, 98, 58, 56, 69, 33]. The vast majority of these methods "borrow strength" across the data views to obtain a more accurate clustering of the observations than would be possible based on a single data view. Implicitly, these methods assume that there is a single *consensus* clustering shared by all data views.

The P100 data contains many data views; multiple data types (e.g. clinical data and proteomic data) are available at multiple timepoints. Thus, it is tempting to apply consensus clustering methods to identify subgroups of the P100 participants. However, before doing so, it is important to check the assumption that there exists a single consensus clustering. If instead different views reflect unrelated aspects of the participants, then there is no "strength to be borrowed" across the views, and it would be better to perform a separate

clustering of the observations in each view. Before attempting cluster analysis of the P100 data, it is critical that we determine which combinations of views have "strength to be borrowed", and which combinations do not.

This raises the natural question of how associated the underlying clusterings are in each view. Suppose we cluster the P100 participants twice, once using their baseline clinical data, and once using their baseline proteomic data. *Can we tell from the data whether the two views' underlying clusterings are related or unrelated?* Answering this question provides useful information:

> *Case 1:* If the underlying clusterings appear related, then this increases confidence that the clusterings are scientifically meaningful, and offers some support for performing a consensus clustering of the P100 participants that integrates baseline clinical and proteomic views.
>
> *Case 2:* If the underlying clusterings appear unrelated, we must consider two explanations.
>
> 1. Perhaps clinical and proteomic views measure different properties about the participants, and therefore identify complementary (or "orthogonal") clusterings. If so, then a consensus clustering is unlikely to provide meaningful results, and may cause us to lose valuable information about the subgroups underlying the individual data views.
> 2. Perhaps the subgroups underlying the data views are indeed related, but they appear unrelated due to noise. If so, then we might be skeptical of any results obtained on these very noisy data, whether from consensus clustering or another approach.

In Case 2, it would not be appropriate to perform consensus clustering.

To determine from the data whether the two views' clusterings are related or unrelated, it is tempting to apply a clustering procedure (e.g. k-means) to each view, then apply well-studied tests of independence of categorical variables (e.g. the χ^2-test for independence,

the G-test for independence, or Fisher's exact test) to the estimated cluster assignments. However, such an approach relies on an assumption that the estimated cluster assignments are independent and identically distributed samples from the joint distribution of the cluster membership variables, which is not satisfied in practice. Thus, there is a need for an approach which takes into account the fact that the clusterings are estimated from the data.

The rest of this chapter is organized as follows. In Section 2.2, we propose a mixture model for two-view data. In Section 2.3, we use this model to develop a test of the null hypothesis that clusterings on two views of a single set of observations are independent. We explore the performance of our proposed hypothesis test via numerical simulation in Section 2.4. In Section 2.5, we connect and compare our proposed hypothesis test to the aforementioned approach of applying the G-test for independence to the estimated cluster assignments, and draw connections between this approach and the mutual information statistic [79]. In Section 2.6, we apply our method to the clinical, proteomic, and metabolomic datasets from the P100 study. In Section 2.7, we provide a discussion, which includes the extension to more than two views.

2.2 A mixture model for multiple-view data

2.2.1 Model specification

In what follows, we consider the case of two data views. We will discuss the extension to more than two views in Section 2.7.

Suppose we have p_1 and p_2 features in the first and second data view, respectively. For a single observation, let $X^{(1)} \in \mathbb{R}^{p_1}$ and $X^{(2)} \in \mathbb{R}^{p_2}$ denote the random vectors corresponding to the two data views and let $Z^{(1)} \in \{1, \ldots, K^{(1)}\}$ and $Z^{(2)} \in \{1, \ldots, K^{(2)}\}$ be unobserved random variables, indicating the latent group memberships of this observation in the two data views. Here, $K^{(1)}$ and $K^{(2)}$ represent the number of clusters in the two data views, which we assume for now to be known (we will consider the case in which they are unknown in Section 2.2.4). We assume that $X^{(1)}$ and $X^{(2)}$ are conditionally independent given the pair of cluster memberships, $(Z^{(1)}, Z^{(2)})$; this assumption is common in the multi-view

clustering literature (see e.g. [9], [94], [58], [69], [33]). Further, suppose that

$$f(X^{(l)} \mid Z^{(l)} = k) = \phi^{(l)}\left(X^{(l)}; \theta_k^{(l)}\right) \quad \text{for} \quad 1 \le k \le K^{(l)},\ 1 \le l \le 2, \tag{2.1}$$

$$P(Z^{(1)} = k, Z^{(2)} = k') = \Pi_{kk'} \quad \text{for} \quad 1 \le k \le K^{(1)}, 1 \le k' \le K^{(2)}, \tag{2.2}$$

where $\phi^{(l)}(\cdot;\theta)$ denotes a density function with parameter θ, and $\Pi \in \Delta^{K^{(1)} \times K^{(2)}} \equiv \{S \in \mathbb{R}^{K^{(1)} \times K^{(2)}} : S_{kk'} \ge 0, \sum_{k=1}^{K^{(1)}} \sum_{k'=1}^{K^{(2)}} S_{kk'} = 1\}$. Equations (2.1)–(2.2) are an extension of the finite mixture model [76] to the case of two data views. We further assume that each cluster has positive probability, i.e. $P(Z^{(1)} = k) > 0$ and $P(Z^{(2)} = k') > 0$, and so $\Pi 1_{K^{(2)}} \in \Delta_+^{K^{(1)}}$ and $\Pi^T 1_{K^{(1)}} \in \Delta_+^{K^{(2)}}$, where $\Delta_+^K \equiv \{s \in \mathbb{R}^K : s_k > 0, \sum_{k=1}^{K} s_k = 1\}$.

Let $\theta^{(1)} \equiv (\theta_1^{(1)}, \ldots, \theta_{K^{(1)}}^{(1)})$ and $\theta^{(2)} \equiv (\theta_1^{(2)}, \ldots, \theta_{K^{(2)}}^{(2)})$. The joint density of $X^{(1)}$ and $X^{(2)}$ is

$$\begin{aligned} f(X^{(1)}, X^{(2)}; \theta^{(1)}, \theta^{(2)}, \Pi) &= \sum_{k,k'} \Pi_{kk'} f(X^{(1)}, X^{(2)} \mid Z^{(1)} = k, Z^{(2)} = k') \\ &= \sum_{k,k'} \Pi_{kk'} f(X^{(1)} \mid Z^{(1)} = k) f(X^{(2)} \mid Z^{(2)} = k') \\ &= \sum_{k,k'} \Pi_{kk'} \phi^{(1)}\left(X^{(1)}; \theta_k^{(1)}\right) \phi^{(2)}\left(X^{(2)}; \theta_{k'}^{(2)}\right), \end{aligned} \tag{2.3}$$

where the second equality follows from conditional independence of $X^{(1)}$ and $X^{(2)}$ given $Z^{(1)}$ and $Z^{(2)}$, and the last equality follows from (2.1).

The matrix Π governs the statistical dependence between the two data views. It will be useful for us to parameterize Π in terms of a triplet $(\pi^{(1)}, \pi^{(2)}, C)$ that separates the single-view information from the cross-view information.

Proposition 1 *Suppose $\pi^{(1)} \in \Delta_+^{K^{(1)}}$ and $\pi^{(2)} \in \Delta_+^{K^{(2)}}$. Then,*

$$\left\{\Pi \in \Delta^{K^{(1)} \times K^{(2)}} : \Pi 1_{K^{(2)}} = \pi^{(1)},\ \Pi^T 1_{K^{(1)}} = \pi^{(2)}\right\} = \left\{\mathrm{diag}(\pi^{(1)}) C \mathrm{diag}(\pi^{(2)}) : C \in \mathcal{C}_{\pi^{(1)}, \pi^{(2)}}\right\},$$

where $\mathcal{C}_{\pi^{(1)}, \pi^{(2)}} = \{C \in \mathbb{R}^{K^{(1)} \times K^{(2)}} : C_{kk'} \ge 0,\ C\pi^{(2)} = 1_{K^{(1)}},\ C^T \pi^{(1)} = 1_{K^{(2)}}\}$.

A proof of Proposition 1 is given in Appendix A.1.

Proposition 1 indicates that any matrix $\Pi \in \Delta^{K^{(1)} \times K^{(2)}}$ with $\Pi 1_{K^{(2)}} \in \Delta_+^{K^{(1)}}$ and $\Pi^T 1_{K^{(1)}} \in \Delta_+^{K^{(2)}}$ can be written as the product of its row sums $\pi^{(1)}$, its column sums $\pi^{(2)}$,

and a matrix C. Therefore, we can rewrite the joint probability density (2.3) as follows:

$$f(X^{(1)}, X^{(2)}; \theta^{(1)}, \theta^{(2)}, \Pi) = \sum_{k,k'} \pi_k^{(1)} C_{kk'} \pi_{k'}^{(2)} \phi^{(1)} \left(X^{(1)}; \theta_k^{(1)} \right) \phi^{(2)} \left(X^{(2)}; \theta_{k'}^{(2)} \right)$$
$$\equiv f(X^{(1)}, X^{(2)}; \theta^{(1)}, \theta^{(2)}, \pi^{(1)}, \pi^{(2)}, C). \quad (2.4)$$

In what follows, we will parametrize the density of $X^{(1)}$ and $X^{(2)}$ in terms of $\theta^{(1)}, \theta^{(2)}, \pi^{(1)}, \pi^{(2)}$, and C, rather than in terms of $\theta^{(1)}, \theta^{(2)}$, and Π.

The following proposition characterizes the marginal distributions of $X^{(1)}$ and $X^{(2)}$.

Proposition 2 *Suppose $X^{(1)}$ and $X^{(2)}$ have joint distribution (2.4). Then for $l = 1, 2$, $X^{(l)}$ has marginal density given by*

$$f(X^{(l)}; \theta^{(l)}, \pi^{(l)}) = \sum_{k=1}^{K^{(l)}} \pi_k^{(l)} \phi_k^{(l)} \left(X^{(l)}; \theta_k^{(l)} \right). \quad (2.5)$$

Proposition 2 follows from (2.1) – (2.2). Proposition 2 shows that for $l = 1, 2$, $X^{(l)}$ marginally follows a mixture model with parameters $\theta^{(l)}$ and cluster membership probabilities $\pi^{(l)}$. Note that the marginal density of $X^{(1)}$ does not depend on $\theta^{(2)}, \pi^{(2)}$, and C, and similarly, the marginal density of $X^{(2)}$ does not depend on $\theta^{(1)}, \pi^{(1)}$, and C; this fact will be critical to our approach to parameter estimation in Section 2.2.3.

The model described in this section is closely related to several multiple-view mixture models proposed in the literature: see e.g. [94], [56], [69], and [33]. However, the focus of those papers is cluster estimation: they do not provide a statistical test of association, and for the most part, impose additional structure on the probability matrix Π in order to encourage similarity between the clusters estimated in each data view. By constrast, the focus of this chapter is inference: testing for dependence between the clusterings in different data views. The model described in this section is a step towards that goal.

2.2.2 Interpreting Π

In Figures 2.1(i)–(iii), $n = 15$ independent pairs $\{(X_i^{(1)}, X_i^{(2)})\}_{i=1}^n$ are drawn from the model (2.1)–(2.2), for three choices of Π. The left-hand panel represents the $p_1 = 2$ features in the first data view, and the right-hand panel represents the $p_2 = 2$ features in the second

data view. For $l = 1, 2$, the observations $\{X_i^{(l)}\}_{i=1}^n$ in the lth data view belong to two clusters, where the latent variables $\{Z_i^{(l)}\}_{i=1}^n$ characterize cluster membership in the lth data view. Light and dark gray represent the clusters in the first view, and circles and triangles represent the clusters in the second view.

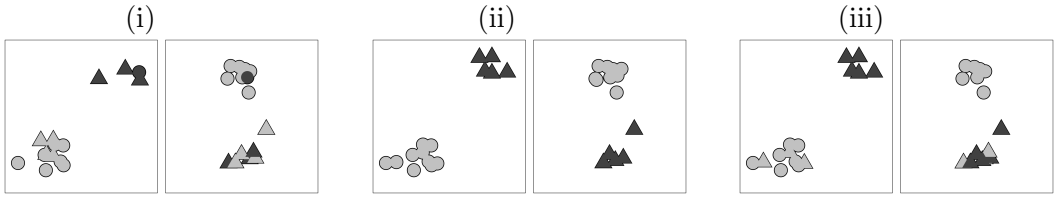

Figure 2.1: Clusters in the first view are represented with dark and light shades of gray, and clusters in the second view are represented with circles and triangles. (i) The clusterings in the two views are independent, i.e. Π has rank one, so the shade of gray (dark or light) and shape (circle or triangle) are unassociated. (ii) The clusterings in the two views are the same, i.e. Π is diagonal (up to permutation of rows), so the shade of gray (dark or light) and shape (circle or triangle) are perfectly correlated. (iii) The clusterings in the two view are somewhat dependent, i.e. Π is neither diagonal nor rank one.

Figures 2.1(i)–(ii) correspond to two special cases of Π that are easily interpretable. In Figure 2.1(i), Π has rank one, i.e. $\Pi = \pi^{(1)}[\pi^{(2)}]^T$, so that the clusterings in the two data views are independent. Thus, whether an observation is light or dark appears to be roughly independent of whether it is a circle or a triangle. In Figure 2.1(ii), $K^{(1)} = K^{(2)}$ and Π is diagonal (up to a permutation of the rows), so that the clusterings in the two data views are identical. Thus, all of the circles are light and all of the triangles are dark. Another special case is when Π is block diagonal (up to a permutation) with K_B blocks. Then, the clusterings of the two data views agree about the presence of K_B "meta-clusters" in the data. For example, one clustering might be a refinement of the other, or if one view has clusters A, B, C, D, and the other has clusters a, b, c, d, it could be that $A \cup B = a$ and $C = b \cup c$ and $D = d$.

In general, Π will be neither exactly rank one nor exactly (block) diagonal; Figure 2.1(iii) provides such an example. Furthermore, $\hat{\Pi}$ (an estimator for Π) almost certainly will be neither. Nonetheless, examination of $\hat{\Pi}$ can provide insight into the relationships between the two clusterings. For example, if $\hat{\Pi}$ is far from rank one, then this suggests that the

clusterings in the two data views may be dependent. We will formalize this intuition in Section 2.3.

2.2.3 Estimation

Estimation Procedure and Algorithm

Given n independent pairs $(X_1^{(1)}, X_1^{(2)}), \ldots, (X_n^{(1)}, X_n^{(2)})$ drawn from the model (2.1)–(2.2), the log-likelihood takes the form

$$\ell(\theta^{(1)}, \theta^{(2)}, \pi^{(1)}, \pi^{(2)}, C) = \sum_{i=1}^{n} \log f(X_i^{(1)}, X_i^{(2)}; \theta^{(1)}, \theta^{(2)}, \pi^{(1)}, \pi^{(2)}, C), \tag{2.6}$$

where $f(\cdot, \cdot; \theta^{(1)}, \theta^{(2)}, \pi^{(1)}, \pi^{(2)}, C)$ is defined in (2.4). A custom expectation-maximization (EM; [27], [75]) algorithm could be developed to solve (2.6) for a local optimum (a global optimum is typically unattainable, as (2.6) is non-concave). We instead take a simpler approach. Proposition 2 implies that for $l = 1, 2$, we can estimate $\theta^{(l)}$ and $\pi^{(l)}$ by maximizing the marginal likelihood for the lth data view, given by

$$\ell(\theta^{(l)}, \pi^{(l)}) = \sum_{i=1}^{n} \log f(X_i^{(l)}; \theta^{(l)}, \pi^{(l)}), \tag{2.7}$$

where $f(\cdot; \theta^{(l)}, \pi^{(l)})$ is defined in (2.5). Each of these maximizations can be performed using standard EM-based software for model-based clustering of a single data view. Let $\hat{\theta}^{(1)}, \hat{\pi}^{(1)}, \hat{\theta}^{(2)}$, and $\hat{\pi}^{(2)}$ denote the maximizers of (2.7). Next, to estimate C, we maximize the joint log-likelihood (2.6) evaluated at $\hat{\theta}^{(1)}, \hat{\pi}^{(1)}, \hat{\theta}^{(2)}$, and $\hat{\pi}^{(2)}$, subject to the constraints imposed by Proposition 1:

$$\hat{C} \equiv \underset{C \in \mathcal{C}_{\hat{\pi}^{(1)}, \hat{\pi}^{(2)}}}{\arg\min} \left[-\ell(\hat{\theta}^{(1)}, \hat{\theta}^{(2)}, \hat{\pi}^{(1)}, \hat{\pi}^{(2)}, C) \right], \tag{2.8}$$

where $\mathcal{C}_{\hat{\pi}^{(1)}, \hat{\pi}^{(2)}} = \{C \in \mathbb{R}^{K^{(1)} \times K^{(2)}} : C_{kk'} \geq 0,\ C\hat{\pi}^{(2)} = 1_{K^{(1)}},\ C^T\hat{\pi}^{(1)} = 1_{K^{(2)}}\}$. Equation 2.8 is a convex optimization problem, which we solve using a combination of exponentiated gradient descent [57] and the Sinkhorn-Knopp algorithm [31], as detailed in Appendix A.3. Details of our approach for fitting the model (2.1)–(2.2) are given in Algorithm 1.

Algorithm 1 Procedure for fitting the model (2.1)–(2.2)

1. Maximize the marginal likelihoods (2.7) in order to obtain the marginal MLEs $\hat{\theta}^{(1)}$, $\hat{\pi}^{(1)}$ and $\hat{\theta}^{(2)}$, $\hat{\pi}^{(2)}$. This can be done using standard software for model-based clustering.

2. Define matrices $\hat{\phi}^{(1)} \in \mathbb{R}^{n \times K^{(1)}}$ and $\hat{\phi}^{(2)} \in \mathbb{R}^{n \times K^{(2)}}$ with elements

$$\hat{\phi}^{(1)}_{ik} = \phi^{(1)}\left(X^{(1)}_i; \hat{\theta}^{(1)}_k\right) \quad \text{and} \quad \hat{\phi}^{(2)}_{ik'} = \phi^{(2)}\left(X^{(2)}_i; \hat{\theta}^{(2)}_{k'}\right). \tag{2.9}$$

3. Fix a step size $s > 0$. Theorem 5.3 from [57] gives conditions on s that guarantee convergence.

4. Let $\hat{C}^1 = 1_{K^{(1)}} 1^T_{K^{(2)}}$. For $t = 1, 2, \ldots$ until convergence:

   (a) Define $M_{kk'} = \hat{C}^t_{kk'} \exp\{sG_{kk'} - 1\}$, where $G_{kk'} = \sum_{i=1}^n \frac{\hat{\phi}^{(1)}_{ik}\hat{\phi}^{(2)}_{ik'}}{[\hat{\phi}^{(1)}_i]^T \mathrm{diag}(\hat{\pi}^{(1)})\hat{C}^t \mathrm{diag}(\hat{\pi}^{(2)})\hat{\phi}^{(2)}_i}$.

   (b) Let $u^0 = 1_{K^{(2)}}$ and $v^0 = 1_{K^{(1)}}$. For $t' = 1, 2, \ldots$, until convergence:

      i. $u^{t'} = \frac{1_{K^{(2)}}}{M^T \mathrm{diag}(\hat{\pi}^{(1)})v^{t'-1}}, \quad v^{t'} = \frac{1_{K^{(1)}}}{M \mathrm{diag}(\hat{\pi}^{(2)})u^{t'}},$

      where the fractions denote element-wise vector division.

   (c) Let u and v be the vectors to which $u^{t'}$ and $v^{t'}$ converge. Let $\hat{C}^{t+1}_{kk'} = u_k M_{kk'} v_{k'}$.

5. Let $\hat{C}$ denote the matrix to which $\hat{C}^t$ converges, and let $\hat{\Pi} = \mathrm{diag}(\hat{\pi}^{(1)})\hat{C}\mathrm{diag}(\hat{\pi}^{(2)})$.

Justification of Estimation Procedure

The estimation procedure described in Algorithm 1 does not maximize the joint likelihood (2.6); nonetheless, we will argue that it is an attractive approach.

To begin, in Step 1 of Algorithm 1, we estimate $\theta^{(1)}$ and $\pi^{(1)}$ by maximizing the marginal likelihood (2.7). This decision leads to computational advantages, as it enables us to make use of efficient software for clustering a single data view, such as the `mclust` package [96] in `R`. We can further justify this decision using conditional inference theory. Equation 3.6 in [92] extends the definition of ancillary statistics to a setting with nuisance parameters. We show that $X^{(2)}$ is ancillary (in the extended sense of [92]) for $\theta^{(1)}, \pi^{(1)}$, and C by using the definition of conditional densities, and Proposition 2, to rewrite (2.4) as

$$f(X^{(1)}, X^{(2)}; \theta^{(1)}, \theta^{(2)}, \pi^{(1)}, \pi^{(2)}, C) = f(X^{(1)} \mid X^{(2)}; \theta^{(1)}, \theta^{(2)}, \pi^{(1)}, \pi^{(2)}, C) f(X^{(2)}; \theta^{(2)}, \pi^{(2)}).$$

Thus, [92] argues that we should use only $X^{(2)}$, and not $X^{(1)}$, to estimate $\theta^{(2)}$ and $\pi^{(2)}$. In Step 1 of Algorithm 1, we are doing exactly this.

In Steps 3–5 of Algorithm 1, we maximize $\ell(\hat{\theta}^{(1)}, \hat{\theta}^{(2)}, \hat{\pi}^{(1)}, \hat{\pi}^{(2)}, \cdot)$, giving $\hat{C}$, which is a pseudo maximum likelihood estimator for C in the sense of [36]. This decision also leads to computational advantages, as it enables us to make use of efficient convex optimization algorithms in estimating C. Results in [36] suggest that when $\hat{\theta}^{(1)}$, $\hat{\theta}^{(2)}$, $\hat{\pi}^{(1)}$, and $\hat{\pi}^{(2)}$ are good estimates, $\hat{C}$ is so as well.

2.2.4 Selection of the number of clusters

In Section 2.2, our discussion assumed that $K^{(1)}$ and $K^{(2)}$ are known. However, this is rarely the case in practice. Recall that we estimate $\theta^{(1)}$ and $\pi^{(1)}$ by maximizing the marginal likelihood (2.7), which amounts to performing model-based clustering of $X^{(1)}$ only. Thus, to select the number of clusters $K^{(1)}$, we can make use of an extensive literature (reviewed in e.g. [81]) on choosing the number of clusters when clustering a single data view. For example, we can use AIC or BIC to select $K^{(1)}$ and $K^{(2)}$.

2.3 Testing whether two clusterings are independent

2.3.1 A brief review of pseudo likelihood ratio tests

Let $\ell(\alpha, \beta, \gamma)$ be the log-likelihood function for a random sample, where $\mathcal{A}$ is the parameter space of α. Given a null hypothesis $H_0 : \alpha = \alpha_0$ for some $\alpha_0 \in \mathcal{A}$, an alternative hypothesis $H_1 : \alpha \neq \alpha_0$, and an estimator $\hat{\gamma}$, the pseudo likelihood ratio statistic [97] is defined to be $\log \tilde{\Lambda} \equiv \sup_{\alpha,\beta} \ell(\alpha, \beta, \hat{\gamma}) - \sup_{\beta} \ell(\alpha_0, \beta, \hat{\gamma})$. Let α^* be the true parameter value for α. If α^* is an interior point of $\mathcal{A}$, then under some regularity conditions, if H_0 holds, then $2 \log \tilde{\Lambda} \xrightarrow{d} \chi^2_r$, where r is the dimension of $\mathcal{A}$ [21].

2.3.2 A pseudo likelihood ratio test for independence

In this subsection, we develop a test for the null hypothesis that $H_0 : C = 1_{K^{(1)}} 1^T_{K^{(2)}}$, or equivalently, that $H_0 : \Pi = \pi^{(1)}(\pi^{(2)})^T$: that is, we test whether $Z^{(1)}$ and $Z^{(2)}$ are independent, i.e. whether the cluster memberships in the two data views are independent. We could use a likelihood ratio test statistic to test H_0,

$$\begin{aligned}\log \Lambda &\equiv \sup_{\theta^{(1)},\theta^{(2)},\pi^{(1)},\pi^{(2)},C} \ell(\theta^{(1)}, \theta^{(2)}, \pi^{(1)}, \pi^{(2)}, C) - \sup_{\theta^{(1)},\theta^{(2)},\pi^{(1)},\pi^{(2)}} \ell(\theta^{(1)}, \theta^{(2)}, \pi^{(1)}, \pi^{(2)}, 1_{K^{(1)}} 1^T_{K^{(2)}}) \\ &= \sup_{\theta^{(1)},\theta^{(2)},\pi^{(1)},\pi^{(2)},C} \ell(\theta^{(1)}, \theta^{(2)}, \pi^{(1)}, \pi^{(2)}, C) - [\ell(\hat{\theta}^{(1)}, \hat{\pi}^{(1)}) + \ell(\hat{\theta}^{(2)}, \hat{\pi}^{(2)})], \qquad (2.10)\end{aligned}$$

where the second equality follows from noticing that substituting $C = 1_{K^{(1)}} 1^T_{K^{(2)}}$ into (2.6) yields

$$\ell(\theta^{(1)}, \theta^{(2)}, \pi^{(1)}, \pi^{(2)}, C) = \ell(\theta^{(1)}, \pi^{(1)}) + \ell(\theta^{(2)}, \pi^{(2)}), \qquad (2.11)$$

where $\ell(\theta^{(l)}, \pi^{(l)})$ for $l = 1, 2$ are defined in (2.7), and recalling the definition of $\hat{\theta}^{(1)}$, $\hat{\pi}^{(1)}$, $\hat{\theta}^{(2)}$, and $\hat{\pi}^{(2)}$ as the maximizers of (2.7). However, (2.10) requires maximizing $\ell(\theta^{(1)}, \theta^{(2)}, \pi^{(1)}, \pi^{(2)}, C)$, which would require a custom EM algorithm; furthermore, the resulting test statistic will typically involve the difference between two local maxima (since each term in (2.10) requires fitting an EM algorithm). This leads to erratic behavior, such as negative values of $\log \Lambda$.

Therefore, instead of taking the approach in (2.10), we develop a pseudo likelihood ratio test, as in Section 2.3.1. We use the marginal MLEs, $\hat{\theta}^{(1)}, \hat{\pi}^{(1)}$, and $\hat{\theta}^{(2)}, \hat{\pi}^{(2)}$, instead of

performing the joint optimization in (2.10). This leads to the test statistic

$$\begin{aligned}\log\tilde{\Lambda} &\equiv \sup_{C\in\mathcal{C}_{\hat{\pi}^{(1)},\hat{\pi}^{(2)}}} \ell(\hat{\theta}^{(1)},\hat{\theta}^{(2)},\hat{\pi}^{(1)},\hat{\pi}^{(2)},C) - \ell(\hat{\theta}^{(1)},\hat{\theta}^{(2)},\hat{\pi}^{(1)},\hat{\pi}^{(2)},1_{K^{(1)}}1_{K^{(2)}}^T) \\ &= \ell(\hat{\theta}^{(1)},\hat{\theta}^{(2)},\hat{\pi}^{(1)},\hat{\pi}^{(2)},\hat{C}) - [\ell(\hat{\theta}^{(1)},\hat{\pi}^{(1)}) + \ell(\hat{\theta}^{(2)},\hat{\pi}^{(2)})] \qquad (2.12) \\ &= \sum_{i=1}^{n} \log\left[\frac{(\hat{\phi}_i^{(1)})^T \mathrm{diag}(\hat{\pi}^{(1)})\hat{C}\mathrm{diag}(\hat{\pi}^{(2)})\hat{\phi}_i^{(2)}}{(\hat{\phi}_i^{(1)})^T\hat{\pi}^{(1)}(\hat{\pi}^{(2)})^T\hat{\phi}_i^{(2)}}\right]. \qquad (2.13)\end{aligned}$$

where $\hat{C}$ in (2.12) is defined in (2.8), $\mathcal{C}_{\hat{\pi}^{(1)},\hat{\pi}^{(2)}}$ is defined in Proposition 1, $\hat{\phi}^{(1)}$ and $\hat{\phi}^{(2)}$ are defined in (2.9), and the last equality follows from (2.6), (2.7), and (2.9). In addition to taking advantage of the computationally efficient estimation procedure described in Section 2.2.3, the pseudo likelihood ratio test statistic does not exhibit the erratic behavior exhibited by the likelihood ratio test statistic. This stability comes from all three terms in (2.12) involving the same local maxima (as opposed to different local maxima).

2.3.3 Approximating the null distribution of $\log\tilde{\Lambda}$

The discussion in Section 2.3.1 suggests that under $H_0 : C = 1_{K^{(1)}}1_{K^{(2)}}^T$, one might expect that $2\log\tilde{\Lambda} \xrightarrow{d} \chi_r^2$, where $r = (K^{(1)}-1)(K^{(2)}-1)$ is the dimension of $\mathcal{C}_{\pi^{(1)},\pi^{(2)}}$. However, this approximation performs poorly in practice, due to violations of the regularity conditions in [21]. Furthermore, we will often be interested in data applications in which n is relatively small. Hence, we propose a permutation approach. We observe from (2.11) that under H_0, the log-likelihood is identical under any permutation of the order of the samples in each view. Hence, we take B random permutations of the samples $X_i^{(2)}$ from the second view, and compare the observed value of $\log\tilde{\Lambda}$ to its empirical distribution in these permutation samples. Details are given in Algorithm 2. Since $\hat{\phi}^{(1)}$, $\hat{\phi}^{(2)}$, $\hat{\pi}^{(1)}$, and $\hat{\pi}^{(2)}$ are invariant to permutation, for each permutation we need only to estimate C. This is another advantage of our test over the likelihood ratio test discussed in Section 2.3.2, which would require repeating the EM algorithm in every permutation. Even when we reject the null hypothesis, the clusters could be only weakly dependent; thus, it is helpful to measure the strength of association between the views. Recalling from Section 2.2.2 that $\mathrm{rank}(\Pi) = 1$ implies independence of the clusterings in the two data views, we propose to calculate the effective

Algorithm 2 A Permutation Approach for Testing $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$

1. Compute $\log \tilde{\Lambda}$ according to (2.13) using the original data, $X^{(1)}$ and $X^{(2)}$.

2. For $b = 1, \ldots, B$, where B is the number of permutations:

   (a) Permute the observations in $X^{(2)}$ to obtain $X^{(2,*b)}$.

   (b) Compute $\log \tilde{\Lambda}^{*b}$ according to (2.13) based on $X^{(1)}$ and $X^{(2,*b)}$.

3. The p-value for testing $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$ is given by $\frac{1}{B} \sum_{b=1}^{B} 1_{\left\{\log \tilde{\Lambda} \leq \log \tilde{\Lambda}^{*b}\right\}}$.

rank [111] of $\hat{\Pi}$, defined in Algorithm 1 – the ratio of the sum of the singular values of $\hat{\Pi}$, and the largest singular value of $\hat{\Pi}$. The effective rank of a matrix is bounded between 1 and its rank, and the matrix is far from rank-1 when its effective rank is far from 1. For example, in Figure 2.1(iii), the effective rank of Π is 1.5, and is upper bounded by 2. Thus, the effective rank of $\hat{\Pi}$ is bounded between 1 and $\min\{K^{(1)}, K^{(2)}\}$, and $\hat{\Pi}$ is far from rank-1 when its effective rank is far from 1.

2.4 Simulation results

To investigate the Type I error and power of our test, we generate data from (2.1)–(2.2), with

$$\Pi = \frac{1-\delta}{K^2} 1_K 1_K^T + \frac{\delta}{K} I_K, \tag{2.14}$$

for $K = 6$ and for a range of values of $\delta \in [0, 1]$, where $\delta = 0$ corresponds to independent clusterings, and $\delta = 1$ corresponds to identical clusterings. We draw the observations in the lth data view from a Gaussian mixture model, for which the kth mixture component is a $N_p(\mu_k^{(l)}, \Sigma^{(l)})$ distribution, with $p = 10$, and with $\mu_k^{(l)}$ given in Appendix A.4.

We simulate 2000 data sets for $\Sigma^{(1)} = \Sigma^{(2)} = \sigma^2 I_p$ for a range of values of σ and n, and evaluate the power of the pseudo likelihood ratio test of $H_0 : C = 1_{K_1} 1_{K_2}^T$ described in Section 2.3.2 at nominal significance level $\alpha = 0.05$, when the number of clusters is

correctly and incorrectly specified. To perform Step 1 of Algorithm 1, we use the package `mclust` in `R` to fit Gaussian mixture models with a common $\sigma^2 I_p$ covariance matrix (the "EII" covariance structure in `mclust`). We use $B = 200$ permutation samples in Step 2 of Algorithm 2. Simulations in this section were conducted using the `simulator` package [10] in `R`. Results are shown in Figure 2.2.

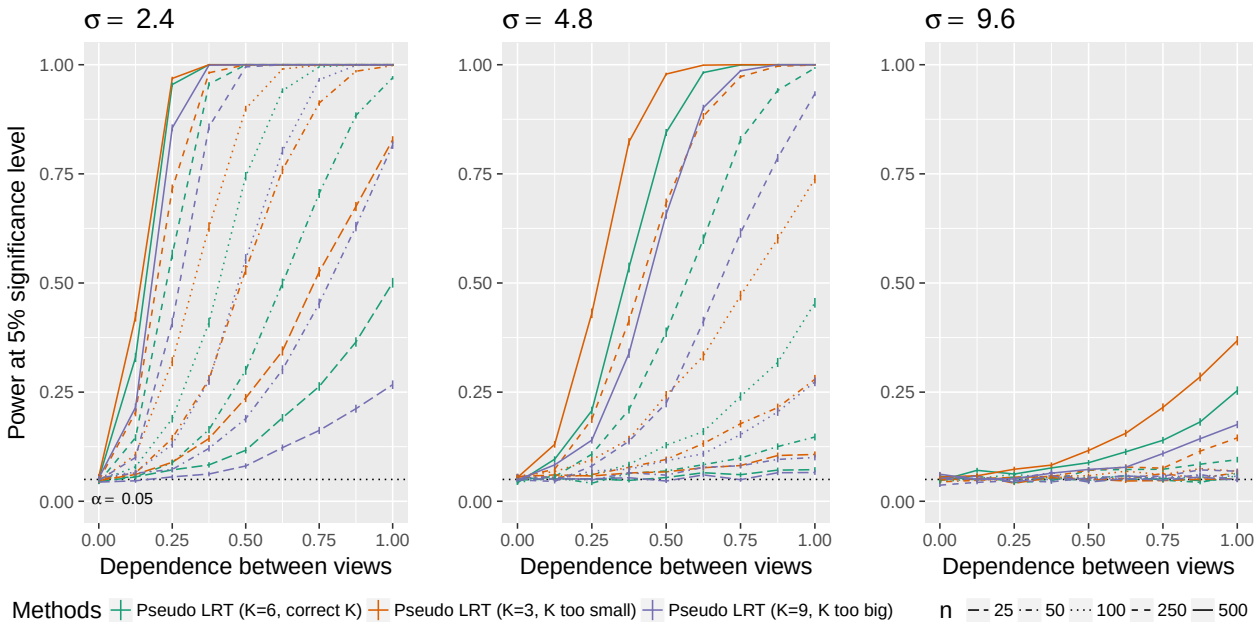

Figure 2.2: Power of the pseudo likelihood ratio test of $H_0 : C = 1_{K_1} 1_{K_2}^T$ with $p = 10$, $K = 6$ and $\sigma \in \{2.4, 4.8, 9.6\}$ in the simulation setting described in Section 2.4. The x-axis displays δ, defined in (2.14), and the y-axis displays the power.

The pseudo likelihood ratio test controls the Type I error close to the nominal $\alpha = 0.05$ level, even when the number of clusters is misspecified. Power tends to increase as δ (defined in (2.14)) increases, and tends to decrease as σ increases. Compared to using the correct number of clusters, using too many clusters yields lower power, but using too few clusters can sometimes yield higher power (e.g. in the middle panel of Figure 2.2). This is because, when the signal-to-noise ratio is low, the true clusters are not accurately estimated; thus, combining several true clusters into a single "meta-cluster" can sometimes, but not always, lead to improved agreement between clusterings across the two data views. We explore the

impact of the choice of K on the performance of the pseudo likelihood ratio test in Appendix A.5.1.

Additional values of K and p are investigated in Appendix A.5.2.

2.5 Connection to the G-test for independence and mutual information

Let $\hat{M}^{(1)} = (\hat{M}_1^{(1)}, \ldots, \hat{M}_n^{(1)})$ and $\hat{M}^{(2)} = (\hat{M}_1^{(2)}, \ldots, \hat{M}_n^{(2)})$ denote the results of applying a clustering procedure to $X^{(1)}$ and $X^{(2)}$ respectively. In this notation, $\hat{M}_i^{(1)} \in \{1, \ldots, K^{(1)}\}$ and $\hat{M}_i^{(2)} \in \{1, \ldots, K^{(2)}\}$ denote the estimated cluster assignment for the ith observation in the two views. To test whether $Z^{(1)}$ and $Z^{(2)}$ are independent, we could naively apply tests on $\hat{M}^{(1)}$ and $\hat{M}^{(2)}$ for whether two categorical variables are independent. For instance, we could use the G-test statistic for independence (Chapter 3.2 in [1]), given by

$$G^2(\hat{M}^{(1)}, \hat{M}^{(2)}) = 2 \sum_{k=1}^{K^{(1)}} \sum_{k'=1}^{K^{(2)}} \hat{N}_{kk'} \log \left[(n\hat{N}_{kk'})/(\hat{N}_{k\cdot}\hat{N}_{\cdot k'}) \right], \tag{2.15}$$

where $\hat{N}_{kk'} = |\{i \in \{1, \ldots, n\} : \hat{M}_i^{(1)} = k, \hat{M}_i^{(2)} = k'\}|$, $\hat{N}_{\cdot k'} = \sum_k \hat{N}_{kk'}$, and $\hat{N}_{k\cdot} = \sum_{k'} \hat{N}_{kk'}$. Under the model $\hat{N}_{kk'} \overset{\text{ind}}{\sim} \text{Poisson}\,(\mu_{kk'}), \log \mu_{kk'} = \alpha_k + \beta_{k'} + \gamma_{kk'}$, the G-test statistic for independence (2.15) is a likelihood ratio test statistic for testing the null hypothesis of independence, i.e. for testing H_0: $\gamma_{kk'} = 0$ for all k, k'. Thus, under H_0: $\gamma_{kk'} = 0$,

$$G^2(\hat{M}^{(1)}, \hat{M}^{(2)}) \xrightarrow{d} \chi^2_{(K^{(1)}-1)(K^{(2)}-1)}. \tag{2.16}$$

The G-test statistic for independence (2.15) relies on an assumption which is violated in our setting, namely that $\{(\hat{M}_i^{(1)}, \hat{M}_i^{(2)})\}_{i=1}^n$ are independent and identically distributed samples from the distribution of $(Z^{(1)}, Z^{(2)})$. It is nonetheless a natural approach to the problem of comparing two views' clusterings. In fact, the mutual information of [79] for measuring the similarity between two clusterings of a single $n \times p$ dataset can be written as a scaled version of the G-test statistic; when applied to instead measure the similarity between $\hat{M}^{(1)}$ and $\hat{M}^{(2)}$, the mutual information $I(\hat{M}^{(1)}, \hat{M}^{(2)})$ is given by

$$I(\hat{M}^{(1)}, \hat{M}^{(2)}) = G^2(\hat{M}^{(1)}, \hat{M}^{(2)})/2n. \tag{2.17}$$

While the proposed pseudo likelihood ratio test statistic (2.13) for testing independence of $Z^{(1)}$ and $Z^{(2)}$ does not resemble the simple G-test statistic for independence in (2.15), we show here that they are in fact quite related.

Let $\hat{r}_i^{(1)} = \frac{\hat{\phi}_i^{(1)}}{1_{K^{(1)}}^T \hat{\phi}_i^{(1)}}$ and $\hat{r}_i^{(2)} = \frac{\hat{\phi}_i^{(2)}}{1_{K^{(2)}}^T \hat{\phi}_i^{(2)}}$ be the vectors giving the soft-clustering assignment weights (or "responsibilities") for the ith observation in the two views, where $\hat{\phi}_i$ is defined in (2.9). We rewrite the pseudo likelihood ratio test statistic (2.13) as

$$\log \tilde{\Lambda}\left(\hat{\Pi}, \{\hat{r}_i^{(1)}, \hat{r}_i^{(2)}\}_{i=1}^n\right) = \sum_{i=1}^n \log\left[\frac{(\hat{r}_i^{(1)})^T \hat{\Pi} \hat{r}_i^{(2)}}{(\hat{r}_i^{(1)})^T (\hat{\Pi} 1_{K^{(2)}})(1_{K^{(1)}}^T \hat{\Pi}) \hat{r}_i^{(2)}}\right], \tag{2.18}$$

where $\hat{\Pi}$ is defined in Algorithm 1. In the following proposition, we consider replacing the "soft" cluster assignments $\hat{r}_i^{(1)}$ and $\hat{r}_i^{(2)}$ with "hard" cluster assignments, and replacing the estimate $\hat{\Pi}$ derived from the "soft" cluster assignments with an estimate derived from "hard" cluster assignments, in (2.18). In what follows,

$$\hat{M}_i^{(1)} \equiv \underset{k \in \{1,2,\ldots,K^{(1)}\}}{\arg\max}\ \hat{r}_{ik}^{(1)}, \quad \hat{M}_i^{(2)} \equiv \underset{k' \in \{1,2,\ldots,K^{(2)}\}}{\arg\max}\ \hat{r}_{ik'}^{(2)}. \tag{2.19}$$

Proposition 3 *Let $\hat{M}^{(1)}$ and $\hat{M}^{(2)}$ be the estimated model-based cluster assignments in each data view defined by* (2.19)*. Let $\hat{N}$ be the matrix with entries $\hat{N}_{kk'}$ containing the number of observations assigned to cluster k in view 1 and cluster k' in view 2. Then,*

$$\log \tilde{\Lambda}(\hat{N}/n, \{e_{\hat{M}_i^{(1)}}, e_{\hat{M}_i^{(2)}}\}_{i=1}^n) = nI(\hat{M}^{(1)}, \hat{M}^{(2)}) = G^2(\hat{M}^{(1)}, \hat{M}^{(2)})/2, \tag{2.20}$$

where $\log \tilde{\Lambda}(\cdot, \cdot)$ is defined in (2.18)*, and e_t is the unit vector that contains a 1 in the tth element.*

Proposition 3 follows by algebra, and says that replacing the soft cluster assignments in the pseudo likelihood ratio test statistic of Section 2.3 with hard cluster assignments yields *exactly* the G-test statistic for independence (2.15) (and the mutual information given in (2.17))! In fact, in the special case of fitting multiple-view Gaussian mixtures with common covariance matrix $\sigma^2 I_{p_1}$ in the first view and $\sigma^2 I_{p_2}$ in the second view, we will show that as $\sigma \to 0$, and the soft cluster assignments converge to hard cluster assignments, the pseudo likelihood ratio test statistic converges to the G-test for independence. In what follows, $\log \tilde{\Lambda} \equiv \log \tilde{\Lambda}\left(\hat{\Pi}, \{\hat{r}_i^{(1)}, \hat{r}_i^{(2)}\}_{i=1}^n\right)$, as in (2.13) and (2.18).

Proposition 4 *Let $\sigma^2 > 0$. Suppose that to compute $\log \tilde{\Lambda}$, we fit the model (2.1)–(2.2), for $\phi^{(1)}$ and $\phi^{(2)}$ densities of Gaussian distributions with covariance matrices $\sigma^2 I_{p_1}$ and $\sigma^2 I_{p_2}$ respectively. Let $\tilde{M}^{(1)}$ and $\tilde{M}^{(2)}$ denote the results of applying k-means clustering on the two data views. Then, as $\sigma^2 \to 0$, $\log \tilde{\Lambda} \to nI(\tilde{M}^{(1)}, \tilde{M}^{(2)}) = G^2(\tilde{M}^{(1)}, \tilde{M}^{(2)})/2$.*

Proposition 4 is proven in Appendix A.2. When $\sigma^2 > 0$, the pseudo likelihood ratio test statistic, the G-test statistic, and the mutual information are not equivalent. We can thus think of the pseudo likelihood ratio test statistic as reflecting the uncertainty associated with the clusterings obtained on the two views, and the G-test statistic and the mutual information as ignoring the uncertainty associated with the clusterings. This suggests that the pseudo likelihood ratio test of Section 2.3.2 outperforms the G-test for independence when the sample size is small and/or there is little separation between the clusters.

To confirm this intuition, we return to the simulation set-up described in Section 2.4, and compare the performances of the pseudo likelihood ratio test (2.13) and the G-test for independence (2.15) for testing $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$. We obtain p-values for (2.15) using the χ^2 approximation from (2.16), and using a permutation approach, where we take B permutations of the elements of $\hat{M}^{(2)}$, and compare the observed value of (2.15) to its empirical distribution in these permutation samples. The results are shown in Figure 2.3; we see that the two tests yield similar power when the sample size is larger and/or the value of σ is smaller, and that the pseudo likelihood ratio test yields higher power than the G-test for independence when the sample size is smaller and/or the value of σ is larger. We note that the χ^2 approximation for the G-test from (2.16) does not control the Type I error. Additional values of p and K, additional values of $\Sigma^{(l)}$, and non-Gaussian finite mixture models are investigated in Appendices A.6.1, A.6.2, and A.6.3, respectively; the results are similar to those described in this section.

2.6 Application to the Pioneer 100 Wellness Project [90]

2.6.1 Introduction to the scientific problem

In the P100 Wellness Project [90], multiple biological data types were collected at multiple timepoints for 108 healthy participants. For each participant, whole genome sequences

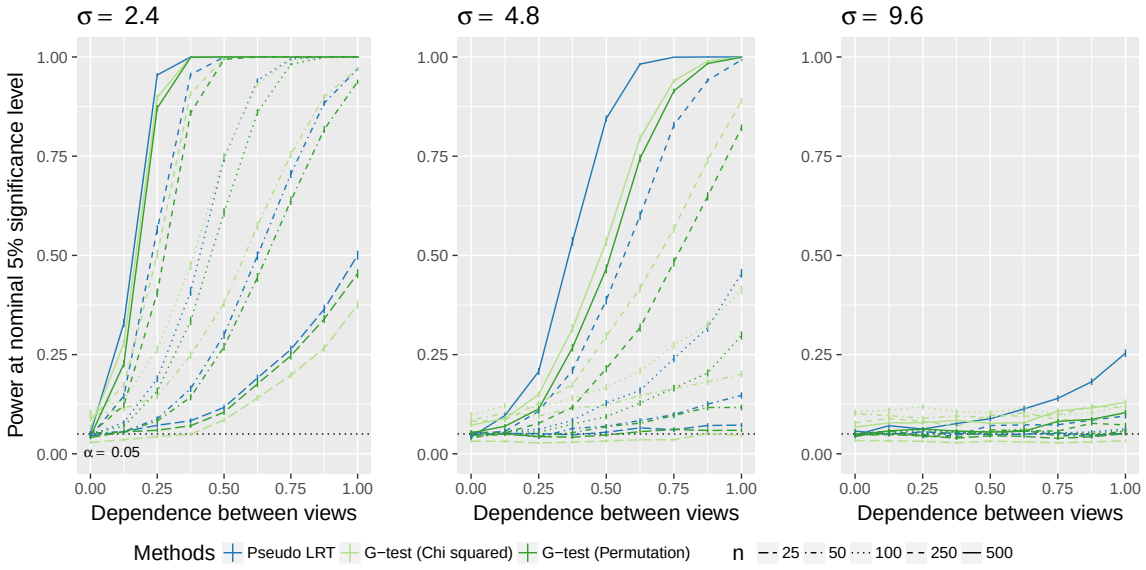

Figure 2.3: For the simulation study described in Section 2.5, power of the pseudo likelihood ratio test and the G-test of independence for $p = 10$, $K = 6$ and $\sigma \in \{2.4, 4.8, 9.6\}$, with δ, defined in (2.14), on the x-axis and power on the y-axis.

were measured, activity tracking data were collected daily over nine months, and clinical laboratory tests, metabolomes, proteomes, and microbiomes were measured at three-month, six-month, and nine-month timepoints. The P100 study aims to optimize wellness of the participants through personalized healthcare recommendations. In particular, clinical biomarkers measured at baseline were used to make personalized health recommendations.

As an alternative approach, we could identify subgroups of individuals with similar clinical profiles using cluster analysis, and then develop interventions tailored to each subgroup. It is tempting to identify these subgroups using not just clinical data at baseline, but also other types of data (e.g. proteomic data) at other timepoints. We could do this by applying a multi-view consensus clustering method (e.g. [98]). However, such an approach assumes that there is a single true clustering underlying all data types at all timepoints. Therefore, before applying a consensus clustering approach, we should determine whether there is any evidence that the clusterings underlying the data types and/or timepoints are at all related (in which case consensus clustering may lead to improved estimation of the clusters) or

whether the clusterings are completely unrelated (in which case one would be better off simply performing a separate clustering of the observations in each view). In what follows, we will use the hypothesis test developed in Section 2.3 to determine whether clusterings of P100 participants based on clinical, proteomic, and genomic data are dependent across timepoints, and across data types.

2.6.2 Data analysis

At each of the three timepoints, 207 clinical measurements, 268 proteomic measurements, and 642 metabolomic measurements were available for $n = 108$ observations. In the following, we define a data view to be a single data type at a single timepoint. In each view, we removed features missing in more than 25% of participants, and removed participants missing more than 25% of features. Next, features in each view with standard deviation 0 were removed. The remaining missing data were imputed using nearest neighbors imputation in the `impute` package in `R` [41]. Features in each view were then adjusted for gender using linear regression. Finally, the remaining features were scaled to have standard deviation 1. As in Section 2.4, we consider the model (2.1)–(2.2) under the assumption that each component in the mixture is drawn from a Gaussian distribution. For each data view, we fit the model using the `mclust` package in `R`, with a common $\sigma^2 I$ covariance matrix (the "EII" covariance structure in `mclust`). To test $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$, we compute p-values using the permutation approximation discussed in Section 2.3.3 with $B = 10^5$. Based on the results in Appendix A.5.1, we choose the number of clusters in each view by BIC under the constraint that the number of clusters is greater than one.

We now compare the clusterings in the clinical data at the first and third timepoints, the clustering in the proteomic data at the first and third timepoints, and the clusterings in the metabolomic data at the first and third timepoints. The sample sizes and results are reported in Table 2.1. For each data type, the clusters found at each timepoint are displayed in Figure 2.4.

We find strong evidence that for each data type, the clusterings at the first and third timepoints are not independent. We further measure the strength of dependence through

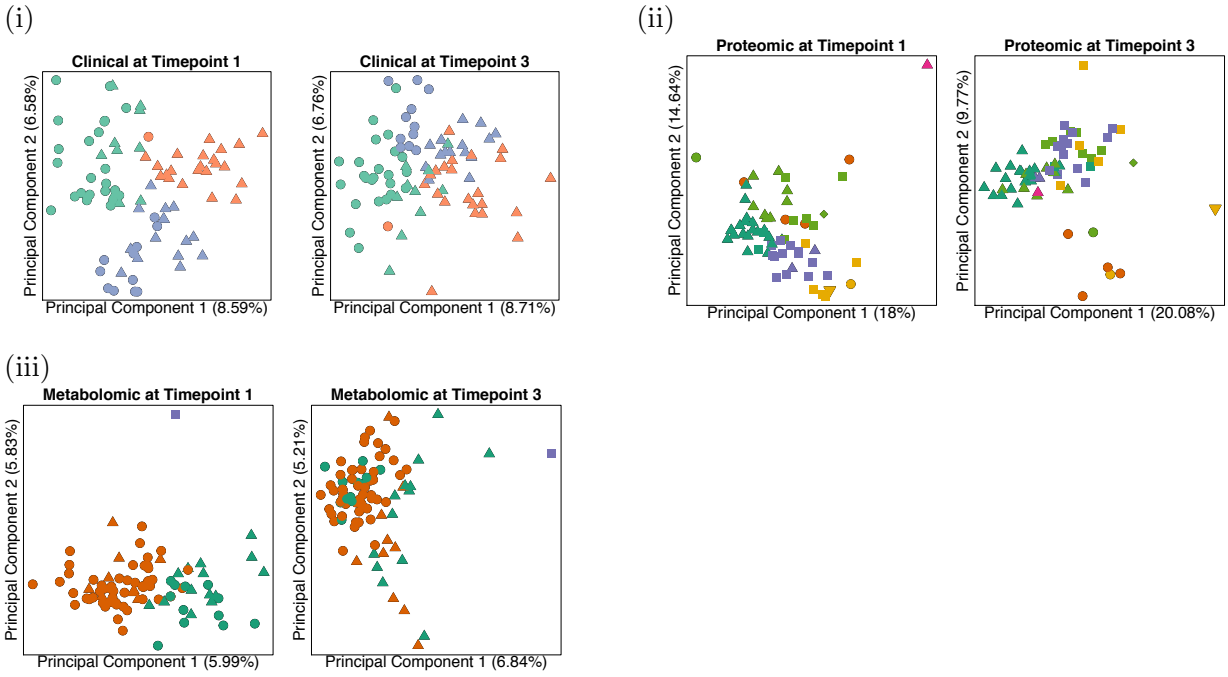

Figure 2.4: For three different data types, a comparison of the clustering at the first timepoint (represented with colors) with the clustering at the third timepoint (represented with shapes). In each data type, there is strong evidence of dependence (p-value < 0.0001). The data types are (i) clinical measurements, (ii) proteomic measurements, and (iii) metabolomic measurements.

the effective rank of $\hat{\Pi}$, as described in Section 2.3.3. For the clusterings in the clinical data, the effective rank of $\hat{\Pi}$ is 1.63, and is upper bounded by 2. For the clusterings in the proteomic data, the effective rank of $\hat{\Pi}$ is 1.90, and is upper bounded by 5. For the clusterings in the metabolomic data, the effective rank of $\hat{\Pi}$ is 1.2, and is upper bounded by 3. These results suggest that the strengths of association for the clusterings estimated on the clinical data, the proteomic data, and the metabolomic data, are strong, moderate, and weak respectively. The fact that the clusterings estimated on some data types are strongly dependent over time provides evidence that they are scientifically meaningful. Furthermore, it suggests that performing consensus clustering on some data types (e.g. clinical data and proteomic data) across timepoints may be reasonable.

We now focus on comparing clusterings in the clinical, proteomic, and metabolomic data

View 1	View 2	n	p_1	p_2	p-value
Clinical at Timepoint 1	Clinical at Timepoint 3	83	204	198	< 0.0001
Proteomic at Timepoint 1	Proteomic at Timepoint 3	66	249	257	< 0.0001
Metabolomic at Timepoint 1	Metabolomic at Timepoint 3	88	641	640	< 0.0001
Clinical at Timepoint 1	Proteomic at Timepoint 1	70	204	249	0.236
Clinical at Timepoint 2	Proteomic at Timepoint 2	60	205	254	0.091
Clinical at Timepoint 3	Proteomic at Timepoint 3	66	198	257	0.950
Clinical at Timepoint 1	Metabolomic at Timepoint 1	98	204	641	0.034
Clinical at Timepoint 2	Metabolomic at Timepoint 2	89	205	641	0.073
Clinical at Timepoint 3	Metabolomic at Timepoint 3	81	198	640	0.328
Proteomic at Timepoint 1	Metabolomic at Timepoint 1	72	249	641	0.402
Proteomic at Timepoint 2	Metabolomic at Timepoint 2	67	254	641	0.004
Proteomic at Timepoint 3	Metabolomic at Timepoint 3	73	257	640	0.020

Table 2.1: Results from the test of $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$ developed in Section 2.3.1 applied to clinical, proteomic, and metabolomic data at the first and third timepoints, and applied to pairs of data views defined by different data types. Sample sizes n, dimensions in each view p_1 and p_2, and p-values obtained using the permutation approximation from Section 2.3.3 are reported.

at a single timepoint. The sample sizes and results are reported in Table 2.1.

The results provide modest evidence that proteomic and metabolomic data at a given timepoint are dependent, and provide weak evidence that clinical and metabolomic data are dependent. However, on balance, the evidence that the clusterings are dependent across data types is weaker than we might expect. This suggests to us that the underlying subgroups defined by the three data types are in fact quite different, and that we should be very wary of performing a consensus clustering type approach across data types, or any analysis strategy that assumes that all three data types are getting at the same set of underlying clusters.

2.7 Discussion

Most existing work on multiple-view clustering has focused on the problem of *estimation*: namely, on exploiting the availability of multiple data views in order to cluster the observations more accurately. In this paper, we have instead focused on the relatively unexplored problem of *inference*: we have proposed a hypothesis test to determine whether clusterings based on multiple data views are independent or associated.

In Section 2.6, we applied our test to the P100 Wellness Study [90]. We found strong evidence that clusterings based on clinical data and proteomic data persist over time, i.e. that the subgroups defined by the clinical data and the proteomic data are similar at different timepoints. This suggests that if we wish to identify participant subgroups based on (say) clinical data, then it may be worthwhile to apply a consensus clustering approach to the clinical data from multiple timepoints. However, we found only modest evidence that clusterings based on different data types are dependent! This suggests that we should be cautious about identifying participant subgroups by applying consensus clustering across multiple data types, as the clusterings underlying the distinct data types may be quite different.

Throughout this paper, we compared clusterings on $L = 2$ data views. We may also wish to compare clusterings across $L > 2$ views. Let $X^{(l)} \in \mathbb{R}^{p_l}$ for $1 \leq l \leq L$ be the random vectors corresponding to the L views. Suppose $X^{(l)}$ are generated according to (2.1) for $1 \leq l \leq L$, where $(Z^{(1)}, \ldots, Z^{(L)})$ are unobserved multinomial random variables with probabilities given by $P(Z^{(1)} = k_1, \ldots, Z^{(L)} = k_L) = \Pi_{k_1 \ldots k_L}$, for $1 \leq k_l \leq K^{(l)}$ and $1 \leq l \leq L$, where the sum of Π over all indices is 1 and $\Pi_{k_1 \ldots k_L} \geq 0$. Results analogous to Propositions 1 and 2 hold in this setting. Thus, we can estimate the parameters in the extended model much as we did in Section 2.2.3, replacing the Sinkhorn-Knopp algorithm for matrix balancing with a tensor balancing algorithm (see e.g. [103]). To test the null hypothesis that $Z^{(1)}, \ldots, Z^{(L)}$ are mutually independent, we can develop a pseudo likelihood ratio test much as we did in Section 2.3, where instead of permuting the observations in $X^{(2)}$ in Step 2(a) of Algorithm 2, we permute the observations in $X^{(2)}, \ldots, X^{(L)}$. Alternatively, one can simply test for pairwise independence between clusterings, instead of testing for

mutual independence between clusterings on all views, as we did in Section 2.6.

An R package titled `multiviewtest` is available on CRAN. Code to reproduce the data analysis in Section 2.6, and to reproduce the simulations in Sections 2.4 and 2.5 and in Appendices A.5 and A.6, are available online at `https://github.com/lucylgao/independent-clusterings-code`.

Chapter 3

TESTING FOR ASSOCIATION IN MULTI-VIEW NETWORK DATA

3.1 Introduction

A network consists of the pairwise relationships (edges) between objects of interest (nodes). For example, nodes could correspond to proteins, with edges representing physical interactions, or nodes could correspond to people, with edges representing social interactions. Of the many models for network data [28, 46, 44], one of the best known is the stochastic block model [45], which assumes that nodes belong to latent communities, and that the probability of an edge between a pair of nodes is a function only of the community memberships for the two nodes.

It is often the case that multiple sets of edges are available on a common set of nodes, as is shown in Figure 3.1(i). Consider a pair of protein-protein interaction networks in which the nodes correspond to proteins [26]. In one network, the edges represent physical interactions (*binary interactions*), and in the other, they represent co-membership in a protein complex (*co-complex associations*). Another often-encountered scenario involves a single network, with a set of covariates corresponding to each node, as is shown in Figure 3.1(ii). For instance, we might have a social network along with p demographic covariates for each member of the network. Both Figures 3.1(i) and 3.1(ii) are examples of the multi-view data setting [104, 65], and we will refer to the two networks in Figure 3.1(i), or the network and the covariates corresponding to the nodes in Figure 3.1(ii), as two data views.

Extensions of network models to the multi-view data setting [30, 38, 35, 12, 95, 23] often assume that the data views are closely related. For example, extensions of the stochastic block model typically assume that the latent communities within each network view are closely related [38, 88, 102, 12, 101].

In this chapter, we propose a test of the assumption that the latent communities are

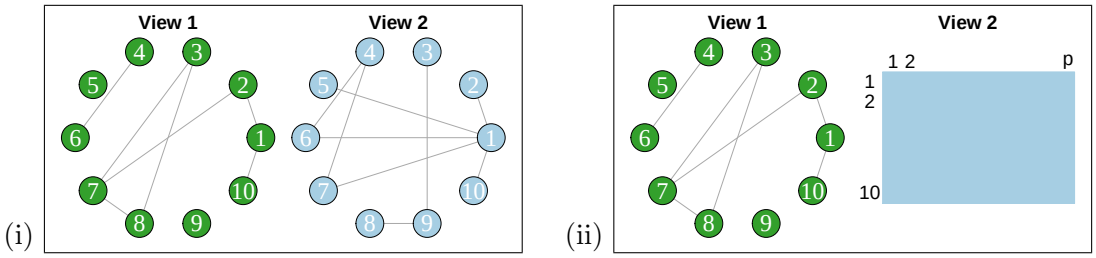

Figure 3.1: Two examples of multi-view data involving a network. (i) Two network views on $n = 10$ nodes. (ii) A network view and an $n \times p$ multivariate view on $n = 10$ nodes.

related. Why is this important? First of all, we should check whether two data views are in fact associated before we fit a model that relies on this assumption. Second, the relationship between the views may itself be of interest, and the test that we propose will allow us to assess this relationship. For example, such a tool can help shed light on whether the two distinct definitions of protein interactions capture similar versus complementary latent structures. Likewise, it can provide insight about whether peoples' social interactions and demographics are related. We investigated a similar problem for two multivariate data views in Chapter 2, but did not consider the case where one or both views are networks.

To this end, we extend the stochastic block model to the multi-view network setting (Figure 3.1(i)) *without* assuming that the network views are closely related. We then ask: are the latent communities within each network view associated? Similarly, for the case of a network view and a multivariate view (Figure 3.1(ii)), we model the network view with a stochastic block model and model the multivariate view with a finite mixture model [76], without assuming that the views are closely related. We then ask: are the latent communities within the network data view and the latent clusters within the multivariate data view associated?

The rest of the chapter is organized as follows. We review the stochastic block model in Section 3.2. We extend the stochastic block model to two network data views in Section 3.3, and develop a test for association between the latent communities within each view in Section 3.4. We develop a related test for the case of network view and a multivariate view in Section 3.5. We review related literature in Section 3.6, and explore the performance of our

tests via numerical simulation in Section 3.7. In Section 3.8, we apply the test from Section 3.4 to protein networks from the HINT database [25]. Section 2.7 provides a discussion.

3.2 The stochastic block model [45]

In this section, we briefly review the stochastic block model (SBM) proposed by [45] for a single network; see [74] for a detailed review.

3.2.1 Model and notation

Let $X \in \{0,1\}^{n \times n}$ be the adjacency matrix of an undirected, unweighted network with n nodes and no self-loops, so that X is symmetric and $X_{ii} = 0$ for $i = 1, 2, \ldots, n$. We assume that the nodes are partitioned into K communities, with unobserved memberships given by a latent random vector $Z = (Z_1, \ldots, Z_n)$ with independent and identically distributed (i.i.d.) elements and $\mathbb{P}(Z_i = k) \equiv \pi_k$ for $\pi \in \Delta_+^K \equiv \left\{\pi \in \mathbb{R}^K : 1_K^T \pi = 1, \pi_k > 0\right\}$. Conditional on Z, the edges are independently drawn from a Bernoulli distribution, with $\mathbb{P}[X_{ij} = 1 \mid Z] = \theta_{Z_i Z_j}$ for a symmetric matrix $\theta \in [0,1]^{K \times K}$. It follows that

$$f(X \mid Z) = \prod_{i=1}^{n} \prod_{j=1}^{i-1} \left(\theta_{Z_i Z_j}\right)^{X_{ij}} \left(1 - \theta_{Z_i Z_j}\right)^{1 - X_{ij}}, \quad \mathbb{P}(Z = z) = \prod_{i=1}^{n} \pi_{z_i}. \tag{3.1}$$

3.2.2 Approximate pseudo-likelihood function

As a result of (3.1), the log-likelihood function for the SBM is given by

$$\ell(\theta, \pi; X) \equiv \log \left(\sum_{z_1=1}^{K} \cdots \sum_{z_n=1}^{K} \left(\prod_{i=1}^{n} \prod_{j=1}^{i-1} \left(\theta_{z_i z_j}\right)^{X_{ij}} \left(1 - \theta_{z_i z_j}\right)^{1 - X_{ij}} \right) \left(\prod_{i=1}^{n} \pi_{z_i} \right) \right). \tag{3.2}$$

Equation (3.2) is computationally intractable, because it involves summing over K^n terms. Therefore, [4] developed an approximate *pseudo-likelihood* function, in the sense of [8]. We briefly review this approach; see Appendix B.1 for a detailed review.

Let $\hat{Z} \in \{1, \ldots, K\}^n$ be the results of applying spectral clustering with perturbations [4] to X. Define $\hat{b} \in \mathbb{R}^{n \times K}$ with rows $\hat{b}_i$ and $\hat{b}_{im} \equiv \sum_{j=1}^{n} X_{ij} \mathbb{1}\{\hat{Z}_j = m\}$, and let $d = X 1_n$. Here, $\hat{b}_{im}$ is the number of edges connecting the ith node to the mth estimated community in $\hat{Z}$, and d contains the degrees of the n nodes. Let $\hat{R}$ be the confusion matrix between $\hat{Z}$

and Z, and define the $K \times K$ matrix $\eta = (\text{diag}(\theta\hat{R}1_K))^{-1}\theta\hat{R}$, with rows $\eta_1, \ldots, \eta_K \in \Delta_+^K$. Let $g(\cdot; N, q)$ denote the probability mass function of a Multinomial$(N, q_1, \ldots, q_K)$ random variable. [4] treated $\hat{Z}$ and η as fixed and showed that

$$\hat{b} \mid d, Z \mathrel{\dot\sim} \prod_{i=1}^{n} g\left(\hat{b}_i; d_i, \eta_{Z_i}\right), \tag{3.3}$$

where $\dot\sim$ denotes "approximately distributed as". Ignoring any dependence between Z and d, and marginalizing over Z in (3.3) to approximate the conditional distribution of $\hat{b}$ given d, yields the following log-pseudo-likelihood function:

$$\ell_{PL}(\eta, \pi; \hat{b} \mid d) \equiv \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \pi_k g(\hat{b}_i; d_i, \eta_k)\right). \tag{3.4}$$

This can be viewed as the log-likelihood function of a finite mixture model (FMM; [76]) with K components, of which the kth component has prior probability π_k and density function $g(\hat{b}_i; d_i, \eta_k)$.

3.3 A stochastic block model for two network data views

In this section, we extend the SBM to the setting of two network data views, and derive approximate pseudo-likelihood functions for the proposed multi-view SBM.

3.3.1 Model and notation

Suppose that we have two network views on a common set of n nodes, as in Figure 3.1(i), e.g. a binary network and a co-complex network on n proteins. We assume that the networks are undirected, unweighted, and have no self-loops. Let $X^{(1)}, X^{(2)} \in \{0,1\}^{n\times n}$ be the symmetric adjacency matrices of the two networks, where $X_{ii}^{(l)} = 0$ for $i = 1, 2, \ldots, n$ and $l = 1, 2$.

We model $X^{(1)}$ with a SBM (Section 3.2.1) with $K^{(1)}$ communities, and $X^{(2)}$ with a SBM with $K^{(2)}$ communities. It follows from (3.1) that for $l = 1, 2$,

$$f(X^{(l)} \mid Z^{(l)}) = \prod_{j=1}^{n}\prod_{i=1}^{j-1}\left(\theta^{(l)}_{Z_i^{(l)}Z_j^{(l)}}\right)^{X_{ij}^{(l)}}\left(1-\theta^{(l)}_{Z_i^{(l)}Z_j^{(l)}}\right)^{1-X_{ij}^{(l)}}, \quad \mathbb{P}(Z^{(l)} = z^{(l)}) = \prod_{i=1}^{n}\pi^{(l)}_{z_i^{(l)}}, \tag{3.5}$$

for a symmetric matrix $\theta^{(l)} \in [0,1]^{K^{(l)} \times K^{(l)}}$ and $\pi^{(l)} \in \Delta_+^{K^{(l)}}$. Here, for $l = 1, 2$, $Z^{(l)}$ represents the latent community memberships for the n nodes within the lth network data view. We assume that the n pairs $\{(Z_i^{(1)}, Z_i^{(2)})\}_{i=1}^n$ are i.i.d. and that $X^{(1)} \perp X^{(2)} \mid Z^{(1)}, Z^{(2)}$.

The following result from Chapter 2 allows us to parameterize the joint distribution of $Z^{(1)}$ and $Z^{(2)}$.

Proposition 5 *Consider two categorical random variables A and B with K and K' levels, respectively, and with $\mathbb{P}(A = k) = \pi_k$ and $\mathbb{P}(B = k') = \pi'_{k'}$, for $\pi \in \Delta_+^K$ and $\pi' \in \Delta_+^{K'}$. Then, there exists a unique matrix $C \in \mathcal{C}_{\pi,\pi'}$ such that*

$$\mathbb{P}(A = k, B = k') = \pi_k \pi'_{k'} C_{kk'},$$

where $\mathcal{C}_{\pi,\pi'} \equiv \{C \in \mathbb{R}^{K \times K'} : \; C_{kk'} \geq 0, \; C\pi' = 1_K, \; C^T\pi = 1_{K'}\}$.

It follows from applying Proposition 5 to each of the n pairs of categorical variables $\{(Z_i^{(1)}, Z_i^{(2)})\}_{i=1}^n$ that there exists a unique $K^{(1)} \times K^{(2)}$ matrix $C \in \mathcal{C}_{\pi^{(1)},\pi^{(2)}}$ such that

$$\mathbb{P}(Z^{(1)} = z^{(1)}, Z^{(2)} = z^{(2)}) = \prod_{i=1}^n \mathbb{P}(Z_i^{(1)} = z_i^{(1)}, Z_i^{(2)} = z_i^{(2)}) = \prod_{i=1}^n \pi^{(1)}_{z_i^{(1)}} \pi^{(2)}_{z_i^{(2)}} C_{z_i^{(1)} z_i^{(2)}}, \quad (3.6)$$

where the first equality follows from the independence of the n pairs $\{(Z_i^{(1)}, Z_i^{(2)})\}_{i=1}^n$.

3.3.2 Approximate pseudo-likelihood function

The log-likelihood function of model (3.5)–(3.6) is given by

$$\ell(\theta^{(1)}, \theta^{(2)}, \pi^{(1)}, \pi^{(2)}, C; X^{(1)}, X^{(2)}) \equiv \quad (3.7)$$

$$\log \left(\sum_{z_1^{(1)}=1}^{K^{(1)}} \cdots \sum_{z_n^{(1)}=1}^{K^{(1)}} \sum_{z_1^{(2)}=1}^{K^{(2)}} \cdots \sum_{z_n^{(2)}=1}^{K^{(2)}} \left(\prod_{l=1}^{2} \prod_{i=1}^{n} \prod_{j=1}^{i-1} \left(\theta^{(l)}_{z_i^{(l)} z_j^{(l)}} \right)^{X_{ij}^{(l)}} \left(1 - \theta^{(l)}_{z_i^{(l)} z_j^{(l)}} \right)^{1 - X_{ij}^{(l)}} \right) \left(\prod_{i=1}^{n} \pi^{(1)}_{z_i^{(1)}} \pi^{(2)}_{z_i^{(2)}} C_{z_i^{(1)} z_i^{(2)}} \right) \right).$$

Equation (3.7) is computationally intractable, because it involves summing over $(K^{(1)}K^{(2)})^n$ terms. Thus, we will derive an approximate pseudo-likelihood function for model (3.5)–(3.6), along the lines of [4].

For $l = 1, 2$, let $\hat{Z}^{(l)} \in \{1, 2, \ldots, K^{(l)}\}^n$ be the results of applying spectral clustering with perturbations [4] to $X^{(l)}$, and let $\hat{b}^{(l)}$ be the $n \times K^{(l)}$ matrix defined by

$$\hat{b}^{(l)}_{im} = \sum_{i=1}^{n} X^{(l)}_{ij} \mathbb{1}\{\hat{Z}^{(l)}_j = m\}, \quad 1 \leq i \leq n, 1 \leq m \leq K^{(l)}, \tag{3.8}$$

and let $d^{(l)} = X^{(l)} 1_n$. Here, $\hat{b}^{(l)}_{im}$ is the number of edges connecting the ith node to the mth estimated community in the lth network, and $d^{(l)}$ contains the degrees of the n nodes in the lth network. We write

$$\begin{aligned} & f(\hat{b}^{(1)}, \hat{b}^{(2)} \mid d^{(1)}, d^{(2)}, Z^{(1)}, Z^{(2)}) \qquad (3.9) \\ = \frac{f(\hat{b}^{(1)}, \hat{b}^{(2)}, d^{(1)}, d^{(2)} \mid Z^{(1)}, Z^{(2)})}{f(d^{(1)}, d^{(2)} \mid Z^{(1)}, Z^{(2)})} &= \frac{f(\hat{b}^{(1)}, d^{(1)} \mid Z^{(1)}) f(\hat{b}^{(2)}, d^{(2)} \mid Z^{(2)})}{f(d^{(1)} \mid Z^{(1)}) f(d^{(2)} \mid Z^{(2)})} = \prod_{l=1}^{2} f(\hat{b}^{(l)} \mid d^{(l)}, Z^{(l)}), \end{aligned}$$

where the first and third equalities follow from the definition of a conditional density, and the second equality follows from the fact that $X^{(1)} \perp X^{(2)} \mid Z^{(1)}, Z^{(2)}$ and $X^{(1)} \perp Z^{(2)} \mid Z^{(1)}$ and $X^{(2)} \perp Z^{(1)} \mid Z^{(2)}$ (Section 3.3.1). Let $\hat{R}^{(l)}$ be the confusion matrix between $\hat{Z}^{(l)}$ and $Z^{(l)}$ and let $\eta^{(l)} = (\mathrm{diag}(\theta^{(l)} \hat{R}^{(l)} 1_{K^{(l)}}))^{-1} \theta^{(l)} \hat{R}^{(l)}$. As in [4], we treat $\hat{Z}^{(l)}$ and $\eta^{(l)}$ as fixed, and apply (3.3) in Section 3.2.2 to approximate $f(\hat{b}^{(l)} \mid Z^{(l)}, d^{(l)})$ for $l = 1, 2$ in (3.9), which yields

$$f(\hat{b}^{(1)}, \hat{b}^{(2)} \mid d^{(1)}, d^{(2)}, Z^{(1)}, Z^{(2)}) \approx \prod_{l=1}^{2} \prod_{i=1}^{n} g(\hat{b}^{(l)}_i; d^{(l)}_i, \eta^{(l)}_{Z^{(l)}_i}). \tag{3.10}$$

Ignoring any dependence between $(d^{(1)}, d^{(2)})$ and $(Z^{(1)}, Z^{(2)})$ and marginalizing over the latent community memberships $Z^{(1)}$ and $Z^{(2)}$ in (3.10) to approximate the conditional distribution of $\hat{b}^{(1)}$ and $\hat{b}^{(2)}$ given $d^{(1)}$ and $d^{(2)}$ yields the following log-pseudo-likelihood function:

$$\begin{aligned} & \ell_{PL}(\eta^{(1)}, \eta^{(2)}, \pi^{(1)}, \pi^{(2)}, C;\ \hat{b}^{(1)}, \hat{b}^{(2)} \mid d^{(1)}, d^{(2)}) \\ \equiv \sum_{i=1}^{n} \log & \left(\sum_{k=1}^{K^{(1)}} \sum_{k'=1}^{K^{(2)}} \pi^{(1)}_k \pi^{(2)}_{k'} C_{kk'} g(\hat{b}^{(1)}_i; d^{(1)}_i, \eta^{(1)}_k) g(\hat{b}^{(2)}_i; d^{(2)}_i, \eta^{(2)}_{k'}) \right). \end{aligned} \tag{3.11}$$

This closely resembles the log-likelihood function of the finite mixture model for two multivariate data views from Section 2.2.1 of Chapter 2.

3.4 Are two network views' community memberships associated?

Recall from (3.6) that $\mathbb{P}(Z^{(1)} = z^{(1)}, Z^{(2)} = z^{(2)}) = \prod_{i=1}^{n} \pi^{(1)}_{z_i^{(1)}} \pi^{(2)}_{z_i^{(2)}} C_{z_i^{(1)} z_i^{(2)}}$, where $C \in \mathcal{C}_{\pi^{(1)},\pi^{(2)}}$, defined in Proposition 1. It follows from the definition of $\mathbb{P}(Z^{(l)} = z^{(l)})$ in (3.5) that

$$\mathbb{P}(Z^{(1)} = z^{(1)}, Z^{(2)} = z^{(2)}) = \mathbb{P}(Z^{(1)} = z^{(1)})\mathbb{P}(Z^{(2)} = z^{(2)})$$

if and only if $C = 1_{K^{(1)}} 1^T_{K^{(2)}}$. Thus, testing the null hypothesis of independence between the latent community memberships $Z^{(1)}$ and $Z^{(2)}$ amounts to testing $H_0 : C = 1_{K^{(1)}} 1^T_{K^{(2)}}$.

3.4.1 The P^2-LRT statistic

To test $H_0 : C = 1_{K^{(1)}} 1^T_{K^{(2)}}$, one might consider using a likelihood ratio test. The likelihood ratio test statistic is of the form

$$\max_{\theta^{(1)},\theta^{(2)},\pi^{(1)},\pi^{(2)},C} \ell(\eta^{(1)},\eta^{(2)},\pi^{(1)},\pi^{(2)},C;X^{(1)},X^{(2)}) - \max_{\eta^{(1)},\eta^{(2)},\pi^{(1)},\pi^{(2)}} \ell(\theta^{(1)},\theta^{(2)},\pi^{(1)},\pi^{(2)},1_{K^{(1)}}1^T_{K^{(2)}};X^{(1)},X^{(2)}),$$

where the log-likelihood function ℓ is defined in (3.7). Unfortunately, recall from Section 3.3.2 that (3.7) is computationally intractable because it involves summing over $(K^{(1)}K^{(2)})^n$ terms. We could replace the log-likelihood functions ℓ with log-pseudo-likelihood functions ℓ_{PL}, defined in (3.11). This leads to a test statistic of the form

$$\begin{aligned}\log \Lambda \equiv & \max_{\eta^{(1)},\eta^{(2)},\pi^{(1)},\pi^{(2)},C} \ell_{PL}(\eta^{(1)},\eta^{(2)},\pi^{(1)},\pi^{(2)},C;\ \hat{b}^{(1)},\hat{b}^{(2)} \mid d^{(1)},d^{(2)}) - \\ & \max_{\eta^{(1)},\eta^{(2)},\pi^{(1)},\pi^{(2)}} \ell_{PL}(\eta^{(1)},\eta^{(2)},\pi^{(1)},\pi^{(2)},1_{K^{(1)}}1^T_{K^{(2)}};\ \hat{b}^{(1)},\hat{b}^{(2)} \mid d^{(1)},d^{(2)}). \qquad (3.12)\end{aligned}$$

However, ℓ_{PL} is a non-concave function of its arguments, and so no algorithms are available to exactly compute the two terms in (3.12) — they can at best be approximated via local maxima. Taking the difference between two local maxima can lead to unintuitive and undesirable behavior; for example, $\log \Lambda$ might be negative.

To overcome this problem, we take a different approach, motivated by the fact that each data view $X^{(l)}$ marginally follows a SBM with parameters $\theta^{(l)}$ and $\pi^{(l)}$ (Section 3.3.1). Rather than estimating the parameters $\eta^{(1)}, \eta^{(2)}, \pi^{(1)}, \pi^{(2)}$ and C by maximizing the log-pseudo-likelihood function for the multi-view SBM (3.11), we first estimate $\eta^{(1)}, \pi^{(1)}$ and

$\eta^{(2)}, \pi^{(2)}$ by maximizing the log-pseudo-likelihood function for the SBM (3.4) for each data view separately. Since (3.4) has exactly the same form as the log-likelihood function of a FMM (Section 3.3.2), it can be maximized using the expectation-maximization (EM; [27]) algorithm for fitting FMMs [75]. We then plug these estimates into (3.12), yielding the test statistic

$$\log \tilde{\Lambda} \equiv \max_{C \in \mathcal{C}_{\hat{\pi}^{(1)}, \hat{\pi}^{(2)}}} \ell_{PL}(\hat{\eta}^{(1)}, \hat{\eta}^{(2)}, \hat{\pi}^{(1)}, \hat{\pi}^{(2)}, C;\ \hat{b}^{(1)}, \hat{b}^{(2)} \mid d^{(1)}, d^{(2)}) - \ell_{PL}(\hat{\eta}^{(1)}, \hat{\eta}^{(2)}, \hat{\pi}^{(1)}, \hat{\pi}^{(2)}, 1_{K^{(1)}} 1_{K^{(2)}}^T;\ \hat{b}^{(1)}, \hat{b}^{(2)} \mid d^{(1)}, d^{(2)}). \tag{3.13}$$

Computing (3.13) requires maximizing the first term with respect to C, i.e. to compute

$$\hat{C} \equiv \operatorname*{arg\,max}_{C \in \mathcal{C}_{\hat{\pi}^{(1)}, \hat{\pi}^{(2)}}} \ell_{PL}(\hat{\eta}^{(1)}, \hat{\eta}^{(2)}, \hat{\pi}^{(1)}, \hat{\pi}^{(2)}, C;\ \hat{b}^{(1)}, \hat{b}^{(2)} \mid d^{(1)}, d^{(2)}), \tag{3.14}$$

where $\mathcal{C}_{\cdot,\cdot}$ is defined in Proposition 5. Because the objective of (3.14) is a concave function of C, $\hat{C}$ can be obtained using techniques from convex optimization. (In particular, we use the exponentiated gradient descent algorithm [57] developed in Chapter 2 for maximizing concave functions of C under the constraint that $C \in \mathcal{C}_{\hat{\pi}^{(1)}, \hat{\pi}^{(2)}}$.) This means that (3.13) completely overcomes the challenges associated with the test statistic (3.12); for example, (3.13) cannot be negative.

We refer to $\log \tilde{\Lambda}$ in (3.13) as a *pseudo-pseudo-likelihood ratio test* (P^2LRT) statistic. In the name P^2LRT, the term "pseudo" is used in two different senses: the first is because we use the pseudo-likelihood function ℓ_{PL} in place of the likelihood function, and the second is because we do not perform a full joint maximization over $(\eta^{(1)}, \eta^{(2)}, \pi^{(1)}, \pi^{(2)}, C)$ [36, 66].

We summarize the procedure for computing the P^2-LRT statistic in Algorithm 3.

3.4.2 Approximating the null distribution

Under the null hypothesis that the community memberships $Z^{(1)}$ and $Z^{(2)}$ are independent, we can write the joint density of the network data views $X^{(1)}$ and $X^{(2)}$ as

$$\begin{aligned} f(X^{(1)}, X^{(2)}) &= \mathbb{E}_{Z^{(1)}, Z^{(2)}}[f(X^{(1)}, X^{(2)} \mid Z^{(1)}, Z^{(2)})] \\ &= \mathbb{E}_{Z^{(1)}, Z^{(2)}}[f(X^{(1)} \mid Z^{(1)}) f(X^{(2)} \mid Z^{(2)})] \\ &= \mathbb{E}_{Z^{(1)}}[f(X^{(1)} \mid Z^{(1)})] \mathbb{E}_{Z^{(2)}}[f(X^{(2)} \mid Z^{(2)})] = f(X^{(1)}) f(X^{(2)}), \end{aligned}$$

Algorithm 3 Computing the P^2-LRT statistic $\log \tilde{\Lambda}$ defined in (3.13)

1. For $l = 1, 2$:

   i. Compute $d^{(l)} = X^{(l)} 1_n$.

   ii. Apply spectral clustering with perturbations [4] to $X^{(l)}$ to obtain $\hat{Z}^{(l)}$, and compute $\hat{b}^{(l)}$ according to (3.8).

   iii. Maximize $\ell_{PL}(\eta^{(l)}, \pi^{(l)}; \hat{b}^{(l)} \mid d^{(l)})$, where ℓ_{PL} is defined in (3.4), and denote the maximizers by $\hat{\eta}^{(l)}$ and $\hat{\pi}^{(l)}$. This can be done using the EM algorithm for fitting FMMs [75].

2. Compute $\hat{C}$ according to (3.14):

   i. Define matrices $\hat{g}^{(1)} \in \mathbb{R}^{n \times K^{(1)}}$ and $\hat{g}^{(2)} \in \mathbb{R}^{n \times K^{(2)}}$ with elements

   $$\hat{g}^{(1)}_{ik} = g\left(\hat{b}^{(1)}_i; d^{(1)}_i, \hat{\eta}^{(1)}_k\right) \quad \text{and } \hat{g}^{(2)}_{ik'} = g\left(\hat{b}^{(2)}_i; d^{(2)}_i, \hat{\eta}^{(2)}_{k'}\right).$$

   ii. Fix a step size $s > 0$.

   iii. Let $\hat{C}^1 = 1_{K^{(1)}} 1^T_{K^{(2)}}$. For $t = 1, 2, \ldots$ until convergence:

      a. Define $O_{kk'} = \hat{C}^t_{kk'} \exp\{sG_{kk'} - 1\}$, where $G_{kk'} = \sum_{i=1}^n \frac{\hat{g}^{(1)}_{ik} \hat{g}^{(2)}_{ik'}}{[\hat{g}^{(1)}_i]^T \mathrm{diag}(\hat{\pi}^{(1)}) \hat{C}^t \mathrm{diag}(\hat{\pi}^{(2)}) \hat{g}^{(2)}_i}$.

      b. Let $u^0 = 1_{K^{(2)}}$ and $v^0 = 1_{K^{(1)}}$. For $t' = 1, 2, \ldots$, until convergence:

      $$u^{t'} = \frac{1_{K^{(2)}}}{O^T \mathrm{diag}(\hat{\pi}^{(1)}) v^{t'-1}}, \quad v^{t'} = \frac{1_{K^{(1)}}}{O \mathrm{diag}(\hat{\pi}^{(2)}) u^{t'}},$$

      where the fractions denote element-wise vector division.

      c. Let u and v be the vectors to which $u^{t'}$ and $v^{t'}$ converge. Let $\hat{C}^{t+1}_{kk'} = u_k O_{kk'} v_{k'}$.

   iv. Let $\hat{C}$ denote the matrix to which $\hat{C}^t$ converges.

3. Compute $\log \tilde{\Lambda}$ according to (3.13), where ℓ_{PL} is defined in (3.11).

where the second equality follows from the fact that $X^{(1)} \perp X^{(2)} \mid Z^{(1)}, Z^{(2)}$ and $X^{(1)} \perp Z^{(2)} \mid Z^{(1)}$ and $X^{(2)} \perp Z^{(2)} \mid Z^{(1)}$ (Section 3.3.1). Thus, under $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$, the network data views $X^{(1)}$ and $X^{(2)}$ are independent, so the joint distribution of $X^{(1)}$ and $X^{(2)}$ is invariant under permutation of the node labels $\{1, 2, \ldots, n\}$ in either network.

It follows that we can approximate the null distribution of the P^2-LRT statistic $\log \tilde{\Lambda}$ defined in (3.13) by taking B random permutations of the node labels in the second network, and comparing the observed value of $\log \tilde{\Lambda}$ to its empirical distribution in the permuted data sets. Since $\hat{\eta}^{(1)}, \hat{\eta}^{(2)}$, $\hat{\pi}^{(1)}$, and $\hat{\pi}^{(2)}$ are invariant to permutation, we only need to compute $\hat{C}$ for each permutation. This is another advantage of the P^2-LRT statistic $\log \tilde{\Lambda}$ in (3.13) over $\log \Lambda$ defined in (3.12): if we had used $\log \Lambda$, then we would need to estimate $\eta^{(1)}, \eta^{(2)}, \pi^{(1)}, \pi^{(2)}$, and C for each permutation. Details of the testing procedure are provided in Algorithm 4.

Algorithm 4 P^2LRT for testing $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$

---

1. Apply Algorithm 3 to compute $\hat{b}^{(1)}, \hat{b}^{(2)}, d^{(1)}, d^{(2)}, \hat{\eta}^{(1)}, \hat{\eta}^{(2)}, \hat{\pi}^{(1)}, \hat{\pi}^{(2)}, \hat{C}$, and the P^2LRT statistic $\log \tilde{\Lambda}$ defined in (3.13).

2. For $m = 1, \ldots, M$, where M is the number of permutations:

   i. Apply the same permutation to the rows of $\hat{b}^{(2)}$ and the elements of $d^{(2)}$ to compute $\hat{b}^{(2,*m)}$ and $d^{(2,*m)}$.

   ii. Apply Step 2 of Algorithm 3 with $\hat{b}^{(2)}$ and $d^{(2)}$ replaced with $\hat{b}^{(2,*m)}$ and $d^{(2*,m)}$ to compute $\hat{C}^{(*m)}$.

   iii. Replace $\hat{b}^{(2)}$, $d^{(2)}$, and $\hat{C}$ with $\hat{b}^{(2,*m)}$, $d^{(2,*m)}$, and $\hat{C}^{(*m)}$ in (3.13) to compute $\log \tilde{\Lambda}^{(*m)}$.

3. The p-value for testing $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$ is given by $\sum_{m=1}^{M} 1_{\{\log \Lambda \leq \log \Lambda^{(*m)}\}} / M$.

3.5 Extension to a network view and a multivariate view

In this section, we develop a test of association between latent communities in a network view and latent clusters in a multivariate view.

3.5.1 Model and notation

We now propose an extension of the SBM to an undirected network view, $X \in \{0,1\}^{n\times n}$, and a multivariate view, $Y \in \mathbb{R}^{n\times p}$. We assume that the network is undirected with no self-loops, so that X is symmetric and $X_{ii} = 0$ for $i = 1, 2, \ldots, n$. We model X with a SBM (Section 3.2.2) with $K^{(1)}$ communities and we model the rows of Y with a finite mixture model [76] with $K^{(2)}$ clusters, so that

$$f(X \mid Z^{(1)}) = \prod_{j=1}^{n}\prod_{i=1}^{j-1}(\theta_{Z_i^{(1)}Z_j^{(1)}})^{X_{ij}}(1-\theta_{Z_i^{(1)}Z_j^{(1)}})^{1-X_{ij}}, \quad f(Y \mid Z^{(2)}) = \prod_{i=1}^{n}\phi(Y_i;\gamma_{Z_i^{(2)}}), \tag{3.15}$$

where $\phi(\cdot;\gamma)$ is a density parameterized by γ, and for $l = 1, 2$, the latent random vector $Z^{(l)} = (Z_1^{(l)}, \ldots, Z_n^{(l)})$ has i.i.d. elements with $\mathbb{P}(Z_i^{(l)} = k) = \pi_k^{(l)}$ for $\pi^{(l)} \in \Delta_+^{K^{(l)}}$. Here, $Z^{(1)}$ represents the latent community memberships in the network view, and $Z^{(2)}$ represents the latent cluster memberships in the multivariate view. We assume that the n pairs $\{(Z_i^{(1)}, Z_i^{(2)})\}_{i=1}^n$ are i.i.d., and that $X \perp Y \mid Z^{(1)}, Z^{(2)}$. Thus, as in Section 3.3.1, it follows from Proposition 1 that there exists $C \in \mathcal{C}_{\pi^{(1)},\pi^{(2)}}$ such that

$$\mathbb{P}(Z^{(1)} = z^{(1)}, Z^{(2)} = z^{(2)}) = \prod_{i=1}^{n}\pi_{z_i^{(1)}}^{(1)}\pi_{z_i^{(2)}}^{(2)}C_{z^{(1)}z^{(2)}}. \tag{3.16}$$

3.5.2 Approximate pseudo-likelihood function

The multi-view log-likelihood function of model (3.15)–(3.16) is computationally intractable. Thus, we will derive a multi-view log-pseudo-likelihood function for model (3.15)–(3.16). We begin by approximating the conditional density of $\hat{b}$ and Y given d, where $\hat{b}$ contains the number of edges connecting each of the n nodes in the network to each of the K estimated

communities in the network, and d contains the node degrees:

$$\hat{b}, Y \mid Z^{(1)}, Z^{(2)}, d \dot{\sim} \prod_{i=1}^{n} g(\hat{b}_i; d_i, \eta_{Z_i^{(1)}}) \phi(Y_i; \gamma_{Z_i^{(2)}}). \tag{3.17}$$

The derivation of (3.17) is very similar to the derivation of (3.10) in Section 3.3.2. Ignoring any dependence between d and $(Z^{(1)}, Z^{(2)})$, and marginalizing over $Z^{(1)}$ and $Z^{(2)}$ in (3.17) to approximate the conditional distribution of $\hat{b}$ and Y given d, yields

$$\ell_{PL}(\eta, \gamma, \pi^{(1)}, \pi^{(2)}, C; \hat{b}, Y \mid d) = \sum_{i=1}^{n} \log \left(\sum_{k,k'} \pi_k^{(1)} \pi_{k'}^{(2)} C_{kk'} g(\hat{b}_i; d_i, \eta_k) \phi(Y_i; \gamma_{k'}) \right). \tag{3.18}$$

We observe that the log-pseudo-likelihood function in (3.18) closely resembles (3.11).

3.5.3 Testing independence between $Z^{(1)}$ and $Z^{(2)}$

We now propose a test for the null hypothesis that the latent community memberships $Z^{(1)}$ and the latent cluster memberships $Z^{(2)}$ in model (3.15)–(3.16) are independent. As in Section 3.4, this amounts to testing $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$.

Recall that the network X marginally follows a SBM, and let $\hat{\eta}$ and $\hat{\pi}$ be the maximizers of $\ell_{PL}(\eta, \pi^{(1)}; \hat{b} \mid d)$, where ℓ_{PL} is the log-pseudo-likelihood function for the SBM given by (3.4). As in Section 3.4.1, we can compute $\hat{\eta}$ and $\hat{\pi}^{(1)}$ by using the EM algorithm for fitting FMMs [75]. Recall that the rows of the multivariate view Y marginally follow a FMM, and let $\hat{\gamma}$ and $\hat{\pi}^{(2)}$ be the maximizers of the log-likelihood function for the multivariate view, obtained via EM. We consider the P^2LRT statistic given by

$$\log \tilde{\Lambda} \equiv \underset{C \in \mathcal{C}_{\hat{\pi}^{(1)}, \hat{\pi}^{(2)}}}{\arg\max} \ell_{PL}(\hat{\eta}, \hat{\gamma}, \hat{\pi}^{(1)}, \hat{\pi}^{(2)}, C; \hat{b}, Y \mid d) - \ell_{PL}(\hat{\eta}, \hat{\gamma}, \hat{\pi}^{(1)}, \hat{\pi}^{(2)}, 1_{K^{(1)}} 1_{K^{(2)}}^T; \hat{b}, Y \mid d), \tag{3.19}$$

where ℓ_{PL} is the log-pseudo-likelihood function defined in (3.18), and $\mathcal{C}_{\cdot,\cdot}$ is defined in Proposition 1. Once again, we can perform the maximization over C in (3.19) using techniques from convex optimization. The details of the exponentiated gradient descent algorithm that we use are similar to Step 2 of Algorithm 3.

As in Section 3.4.2, we approximate the null distribution of $\log \tilde{\Lambda}$ by taking B random permutations of the rows of the multivariate data view $X^{(2)}$, and comparing the observed

value of $\log \Lambda$ to its empirical distribution in the permuted data sets. The details of the testing procedure are similar to Algorithm 4.

3.6 Related literature

Many papers have extended the SBM to the multiple network data view setting, under the assumption that a single set of communities is shared across all networks [38, 88, 85, 5] or a subset of networks [102]. The model proposed in Section 3.3.1 does not rely on this assumption. Most of the previous work that avoids the assumption of shared communities has focused on estimation of the community structure; Section 4 of [54] reviews these papers in detail. By contrast, the primary goal of this chapter is not estimation, but rather to develop a test of association between the communities underlying each network view (Section 3.4).

A related problem in functional neuroimaging is to test whether the communities underlying brain networks of a group of healthy patients are the same as the communities underlying brain networks of a group of diseased patients; see [86], and the references contained therein. However, these tests cannot be used to determine whether the communities underlying two network data views are the same, as the test statistics and/or p-values cannot be computed in the two network data view setting.

In Section 3.4, we proposed a test of the null hypothesis that the communities underlying two network views are independent. By contrast, [115] proposed a test of the null hypothesis that the two network views are *conditionally* independent given the communities underlying the two views.

In the case of a network view and a multivariate view, several papers have assumed that the communities underlying the network view and the clusters underlying the multivariate view are the same, and exploit this assumption to improve parameter estimation [12, 101, 117]. Our proposed model in Section 3.5.1 does not rely on this assumption. Another body of work estimates the relationship between community memberships and node covariates, but does not consider inference on this relationship [119, 83, 121].

In Section 3.5.3, we proposed testing for a specific type of relationship between the network view and the multivariate view: we test for association between the communities

underlying the network view and the clusters underlying the multivariate view. Several papers have considered testing for other types of relationships between the network view and the multivariate view [109, 30, 87]. For example, [30] tests for association between the multivariate view and the latent node-specific factors underlying the network view, within the context of a latent factor model.

3.7 Simulation results

In this section, we evaluate the performance of the test proposed in Section 3.4 in terms of power and Type I error across a variety of simulated scenarios. All simulations in this section were conducted using the `simulator` package [10].

3.7.1 SBM for two network data views

We will evaluate the performance of four tests of $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$:

1. The P^2LRT proposed in Section 3.4, using the true values of $K^{(1)}$ and $K^{(2)}$,

2. The P^2LRT proposed in Section 3.4, using estimated values of $K^{(1)}$ and $K^{(2)}$,

3. The G-test for testing dependence between two categorical variables (Chapter 3.2 in [1]) applied to the estimated community assignments for each view, using the true values of $K^{(1)}$ and $K^{(2)}$, and

4. The G-test, using estimated values of $K^{(1)}$ and $K^{(2)}$.

We estimate $K^{(1)}$ and $K^{(2)}$ by applying the method of [60] to $X^{(1)}$ and $X^{(2)}$, respectively. In all four tests, we approximate the null distribution with a permutation approach, as in Algorithm 4, using $M = 200$ permutation samples.

To evaluate these four tests, we generate data from model (3.5) –(3.6), with $n = 1000$, $K^{(1)} = K^{(2)} = K = 6$, $\pi^{(1)} = \pi^{(2)} = 1_K/K$, and

$$C = (1 - \Delta)1_K 1_K^T + \Delta \cdot \mathrm{diag}(K 1_K), \tag{3.20}$$

for $\Delta \in [0, 1]$. Here, $\Delta = 0$ corresponds to independent communities and $\Delta = 1$ corresponds to identical communities. We set $\theta^{(1)} = \theta^{(2)} = \theta$, with

$$\theta_{kk'} = \omega + (2r - 1)\omega \mathbb{1}\{k = k'\}, \tag{3.21}$$

for $r > 0$, and ω chosen so that the expected edge density of the network equals s, to be specified. Two nodes in the same community are $2r$ times more likely to be connected than two nodes in different communities; thus, r describes the strength of the communities. We simulate 2000 data sets for a range of values of s, Δ, and r, and evaluate the power of the four tests described above. Results are shown in Figure 3.2.

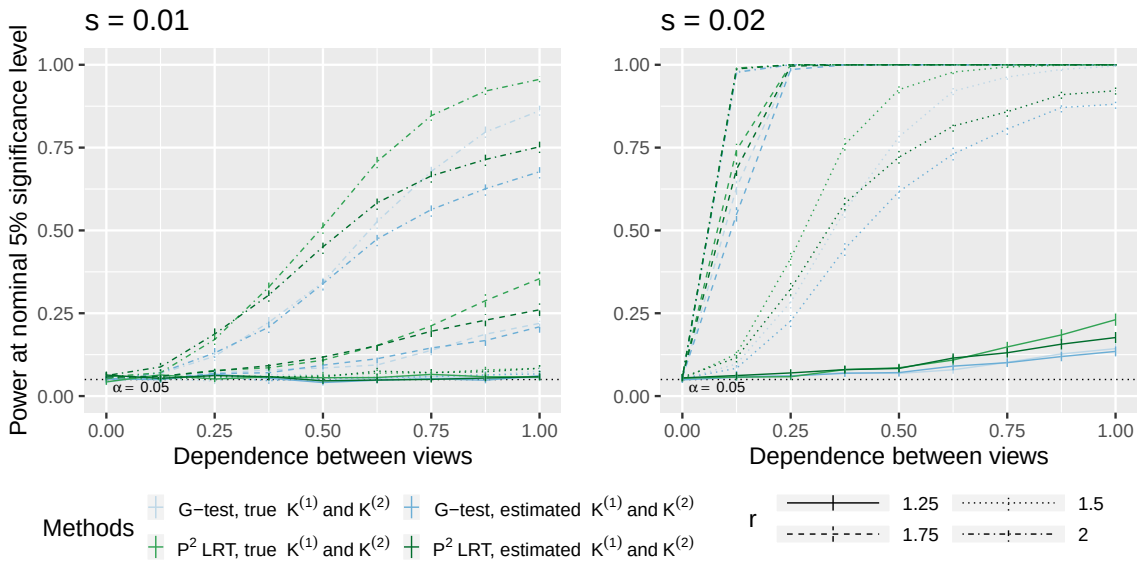

Figure 3.2: Power of the P^2LRT and the G-test with both views drawn from a SBM, varying the dependence between views (Δ), the strength of the communities (r), the expected edge density (s), and how the number of communities is selected. Details are in Section 3.7.1.

For all tests, power tends to increase as Δ, which controls the dependence between views, increases. Power also tends to increase as the strength of the communities (r) increases, and as the expected edge density (s) increases. Estimating $K^{(1)}$ and $K^{(2)}$ tends to yield lower power than using the true values of $K^{(1)}$ and $K^{(2)}$. All tests control the Type I error, but the P^2LRTs uniformly yield higher power than the G-tests.

3.7.2 Degree-corrected SBM for two network data views

Under the SBM, nodes within the same community have the same expected degree. In this subsection, we generate each network view from the more flexible degree-corrected stochastic block model (DCSBM, [52]), that allows nodes within the same community to have different expected degrees. We generate n vectors $(Z_i^{(1)}, Z_i^{(2)}, \delta_i^{(1)}, \delta_i^{(2)})$ i.i.d. for $i = 1, 2, \ldots, n$, with $Z_i^{(1)}$ and $Z_i^{(2)}$ categorical with $K^{(1)}$ and $K^{(2)}$ levels, respectively, and $(Z_i^{(1)}, Z_i^{(2)}) \perp (\delta_i^{(1)}, \delta_i^{(2)})$. Here, $\delta^{(1)}$ and $\delta^{(2)}$ represent *popularities* for the nodes in the two views; more popular nodes have higher expected degrees. We generate each view with

$$X^{(l)} \mid Z^{(l)}, \delta^{(l)} \sim \prod_{j=1}^{n} \prod_{i=1}^{j-1} \left(\delta_i^{(l)} \delta_j^{(l)} \theta^{(l)}_{Z_i Z_j}\right)^{X_{ij}^{(l)}} \left(1 - \delta_i^{(l)} \delta_j^{(l)} \theta^{(l)}_{Z_i^{(l)} Z_j^{(l)}}\right)^{1 - X_{ij}^{(l)}}, \quad l = 1, 2. \quad (3.22)$$

We set n, $K^{(1)}$, $K^{(2)}$, $\pi^{(1)}, \pi^{(2)}$, C, $\theta^{(1)}$, and $\theta^{(2)}$ as in Section 3.7.1 and take $\mathbb{P}(\delta_i^{(l)} = 2.5) = 0.2$, $\mathbb{P}(\delta_i^{(l)} = 0.625) = 0.8$, and $\delta_i^{(1)} \perp \delta_i^{(2)}$. We simulate 2000 data sets, varying the dependence between views (Δ), the expected edge density (s), and the strength of the communities (r); these parameters are defined in Section 3.7.1. Once again, we evaluate the power and Type I error of the four tests described in Section 3.7.1. Results are shown in Figure 3.3, and are similar to Section 3.7.1.

In this subsection, we assumed that the node popularities ($\delta^{(1)}$ and $\delta^{(2)}$) are independent. This can sometimes be an unrealistic assumption in practice. Consider two social network views (e.g. Facebook and LinkedIn) on a common set of people; people with a lot of friends on Facebook may also tend to be those with a lot of connections on LinkedIn. If $\delta^{(1)}$ and $\delta^{(2)}$ are dependent, then $X^{(1)}$ and $X^{(2)}$ could be dependent even when the communities are independent, which could inflate the Type I error rate. To investigate this effect, in Appendix B.2, we generate data from a multi-view DCSBM with $\delta^{(1)}$ and $\delta^{(2)}$ dependent, and apply the P^2LRT using a range of values of $K^{(1)}$ and $K^{(2)}$. We find that the Type I error rate is controlled, both when we estimate the number of communities and when we choose a fixed number of communities (as long as the number of communities is not grossly overspecified); Appendix B.2 gives intuition for why this is the case.

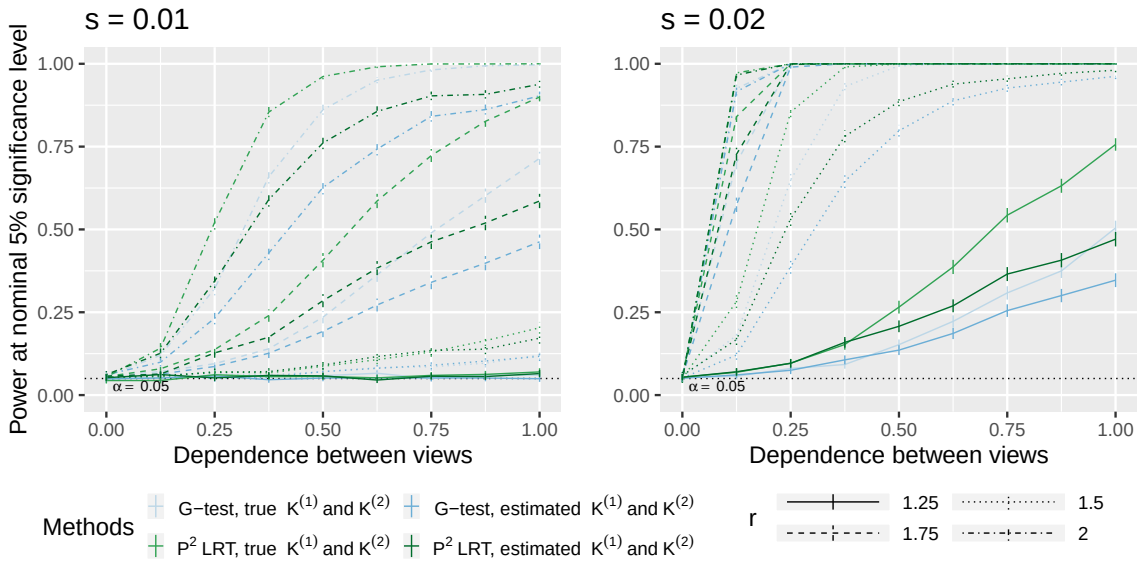

Figure 3.3: Power of the P^2LRT and the G-test with both views drawn from a DCSBM, varying the dependence between views (Δ), the strength of the communities (r), the expected edge density (s), and how the number of communities is selected. Details are in Section 3.7.2.

3.7.3 SBM for a network view and a multivariate view

We will evaluate the performance of four tests of $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$ in the setting in which one of the views is multivariate:

1. The P^2LRT proposed in Section 3.5.3, using the true values of $K^{(1)}$ and $K^{(2)}$,

2. The P^2LRT, using estimated values of $K^{(1)}$ and $K^{(2)}$,

3. The G-test applied to the estimated community assignments in the network view and the estimated cluster memberships in the multivariate view, using the true value of $K^{(1)}$ and $K^{(2)}$, and

4. The G-test, using the estimated values of $K^{(1)}$ and $K^{(2)}$,

where $K^{(1)}$ is estimated by applying the method of [60] to $X^{(1)}$, and $K^{(2)}$ is estimated using BIC. In all four tests, we approximate the null distribution with a permutation approach,

as in Algorithm 4, using $M = 200$ permutation samples.

We generate data from model (3.15)–(3.16); we generate data from a degree-corrected version of model (3.15)–(3.16) in Appendix B.2.1. We set $n = 500$, and $K^{(1)} = K^{(2)} = K = 3$. Let $\pi^{(1)} = \pi^{(2)} = 1_K/K$, and let C be given by (3.20). Let θ be given by (3.21), so that the expected edge density is $s = 0.015$. We draw the multivariate data view from a Gaussian mixture model, for which the kth mixture component is a $N_{10}(\mu_k, \sigma^2 I_{10})$ distribution. The $p \times K$ mean matrix for the multivariate data view is given by $\mu = \begin{bmatrix} 0 \cdot 1_5 & 0 \cdot 1_5 & \sqrt{12} \cdot 1_5 \\ 2 \cdot 1_5 & -2 \cdot 1_5 & 0 \cdot 1_5 \end{bmatrix}$. We simulate 2000 data sets for $n = 500$ and a range of values of Δ, r, and σ. Results are shown in Figure 3.4.

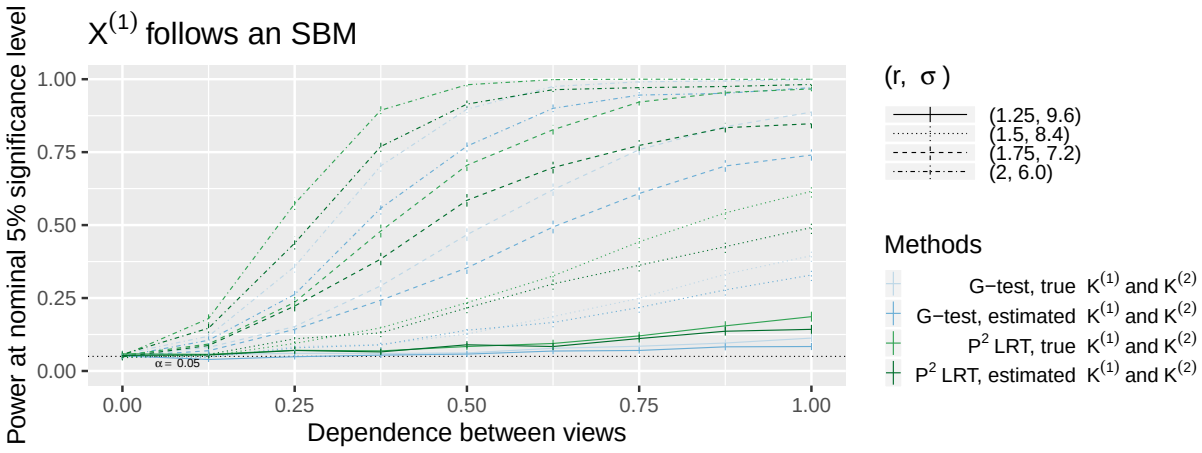

Figure 3.4: Power of the P^2LRT and the G-test with the multivariate view drawn from a Gaussian mixture model and the network view drawn from a SBM, varying the dependence between views (Δ), the strength of the communities (r), the variance of the clusters (σ), and how the number of communities and the number of clusters is selected. The expected edge density (s) is fixed at 0.015. Details are in Section 3.7.3.

The P^2LRT and the G-test both control the Type I error rate. Power tends to increase as the dependence between views (Δ) increases. Power also tends to increase as the strength of the communities (r) increases and the variance of the clusters (σ) decreases. As in Sections 3.7.1 and 3.7.2, the P^2LRT uniformly yields higher power than the G-test.

3.8 Application to protein-protein interaction data

In this section, we focus on two types of protein-protein interaction data. A binary interaction is a physical interaction between proteins, and a co-complex association is a pair of proteins that are part of the same complex. These two data views provide complementary information, in the sense that physical interactions can occur between a pair of proteins that are not in the same complex, and not all proteins in complexes physically interact.

[25] combined and filtered eight protein-protein interaction databases to create the HINT (High-quality INteractomes) database. We consider the *H. sapiens* protein-protein interaction data sets from HINT, and ask: are the communities within the binary interaction network and the communities within the co-complex association network associated?

We remove self-interactions from both networks, and consider only those proteins that appear in both networks. This yields $43,874$ binary interactions and $88,960$ co-complex associations among a common set of $n = 9,037$ proteins. We apply the P^2LRT of $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$ developed in Section 3.4, using $M = 10^4$ in Step 3 of Algorithm 4. As in Section 3.7, we estimate the number of communities in each view by applying the method of [60] to each view separately, which estimates 14 communities in (coincidentially) both data views. Our test yields a p-value of 0.012, and thus provides some evidence against the null hypothesis that communities of proteins defined with respect to binary interactions and communities of proteins defined with respect to co-complex associations are independent. Figure 3.5 displays $\hat{\pi}^{(1)}$ and $\hat{\pi}^{(2)}$ (defined in Section 3.4.1), and $\hat{C}$ (defined in equation 3.14). A color version of Figure 3.5 can be found in the electronic version of the article. Large values of $C_{kk'} = \frac{\mathbb{P}(Z_i^{(1)}=k, Z_i^{(2)}=k')}{\mathbb{P}(Z_i^{(1)}=k)\mathbb{P}(Z_i^{(2)}=k')}$ indicate nodes that are much more likely to belong to the kth community in the binary view and the k'th community in the co-complex view than they would under the assumption of independent communities. We find that the largest values of $\hat{C}_{kk'}$ (in particular, $\hat{C}_{2,4}$, $\hat{C}_{5,3}$, $\hat{C}_{6,6}$) correspond to small values of $\hat{\pi}_k^{(1)}$ and $\hat{\pi}_{k'}^{(2)}$. This means that while the kth community in the binary view and the k'th community in the co-complex view share more nodes than we would expect by chance, the total number of shared nodes is quite small in absolute terms. For instance, we estimate that six nodes jointly belong to the sixth community in the binary view and the sixth community in the co-

complex view, and we estimate that 57 nodes and 95 nodes belong to the sixth community in the binary view and the co-complex view, respectively.

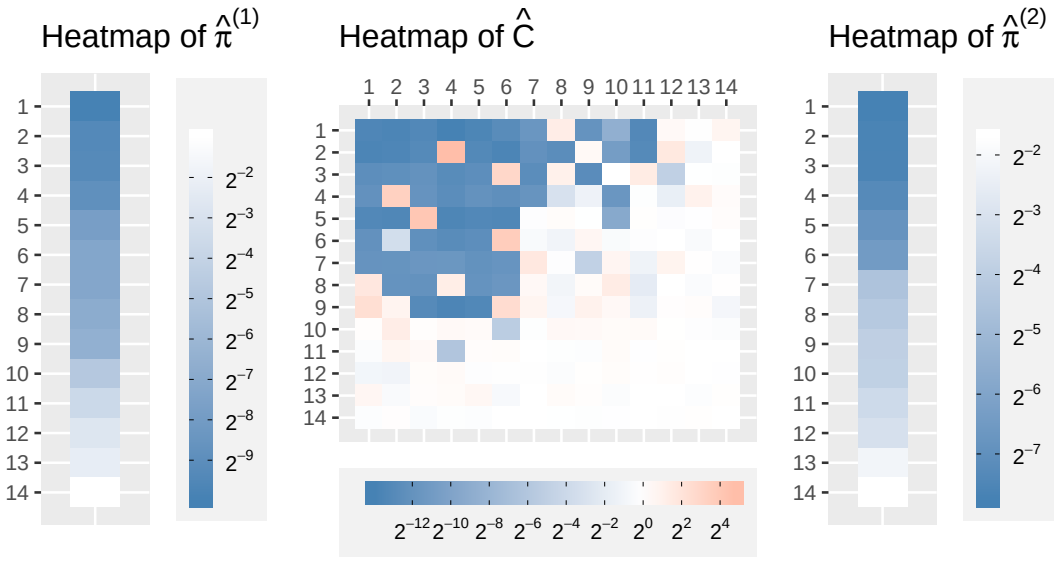

Figure 3.5: Heatmaps of $\hat{\pi}^{(1)}$ and $\hat{\pi}^{(2)}$, defined in Section 3.4.1, and of $\hat{C}$, defined in (3.14), for the HINT database described in Section 3.8.

3.9 Discussion

In this chapter, we considered testing whether communities defined with respect to two networks on a common set of nodes are related. We extended this test to the setting of one network and one multivariate data set on a common set of nodes. The proposed tests control the Type I error rate, and yield higher power than applying the G-test to the estimated community/cluster memberships in each data view.

In this chapter, we considered only undirected, unweighted network views. There is a body of work that extends the single-view SBM to directed and/or weighted networks; see e.g. [114] and [2]. It may be of future interest to extend the methodology developed in this chapter to allow for directed and/or weighted networks.

The tests developed in this chapter are implemented in the R package `multiviewtest`, which is available on CRAN. Code to reproduce the simulations in Section 3.7, Appendix

B.2, and Appendix B.2.1, and code to reproduce the data analysis in Section 2.6, are available online at `https://github.com/lucylgao/mv-network-test-code/`. The data sets used in Section 2.6 are the *H. sapiens* binary and co-complex interactomes openly available in the HINT database [24] at `http://hint.yulab.org`.

Chapter 4

SELECTIVE INFERENCE FOR HIERARCHICAL CLUSTERING

4.1 Introduction

Many popular methods have been developed to estimate clusters in a single data set, including but not limited to k-means clustering [72], model-based clustering [76], agglomerative hierarchical clustering [39], spectral clustering [80, 84] and convex clustering [43, 67]. However, the problem of statistical inference for the clusters estimated from these methods has been relatively unexplored. The existing body of work has largely concentrated on developing hypothesis tests that measure the stability of estimated clusterings with respect to data set perturbation [105, 99], or developing tests that measure the goodness of fit of clustering models. For instance, one body of work tests the goodness of fit of models with K clusters against models without clusters [17, 68, 18, 47, 55], or against models with $K' \neq K$ clusters [64, 19, 73, 77, 53, 16, 113]. These tests are sometimes applied recursively to assess the validity of each split in a dendrogram obtained from agglomerative hierarchical clustering [68, 47, 55, 16].

In this chapter, we consider a different inferential problem: testing for a difference in means between observations belonging to two different estimated clusters. Consider the following matrix Gaussian model for n independent observations of q features:

$$\mathbf{X} \sim \mathcal{MN}_{n\times q}(\boldsymbol{\mu}, \mathbf{I}_n, \sigma^2\mathbf{I}_q), \tag{4.1}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{n\times q}$ is unknown, and $\sigma^2 > 0$ is known. Let $\mathbf{x}$ be an observed realization from (4.1), which we cluster to obtain $\mathcal{C}(\mathbf{x}) = \{\hat{\mathcal{C}}_1, \hat{\mathcal{C}}_2, \ldots, \hat{\mathcal{C}}_K\}$, where

$$\hat{\mathcal{C}}_1 \cup \hat{\mathcal{C}}_2 \cup \cdots \cup \hat{\mathcal{C}}_K = \{1, 2, \ldots, n\}, \qquad \hat{\mathcal{C}}_k \cap \hat{\mathcal{C}}_{k'} = \emptyset, \ \forall\ k \neq k'.$$

Let μ_i denote the ith row of the mean matrix $\boldsymbol{\mu}$. We define the centroid of the kth cluster,

$\hat{\mathcal{C}}_k$, to be the mean vector of $\{\mu_i : i \in \hat{\mathcal{C}}_k\}$:

$$\bar{\mu}_{\hat{\mathcal{C}}_k} = \frac{1}{|\hat{\mathcal{C}}_k|} \sum_{i \in \hat{\mathcal{C}}_k} \mu_i, \quad k = 1, 2, \ldots, K. \tag{4.2}$$

Suppose that we wish to use $\mathbf{x}$ to quantify the evidence that the centroids of two clusters (say, $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$) are different. This amounts to testing

$$H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}} \quad \text{v.s.} \quad H_1 : \bar{\mu}_{\hat{\mathcal{C}}_k} \neq \bar{\mu}_{\hat{\mathcal{C}}_{k'}}. \tag{4.3}$$

For $k = 1, 2, \ldots, K$, let

$$\bar{x}_{\hat{\mathcal{C}}_k} = \frac{1}{|\hat{\mathcal{C}}_k|} \sum_{i \in \hat{\mathcal{C}}_k} x_i \quad \text{and} \quad \bar{X}_{\hat{\mathcal{C}}_k} = \frac{1}{|\hat{\mathcal{C}}_k|} \sum_{i \in \hat{\mathcal{C}}_k} X_i, \tag{4.4}$$

where x_i is the ith row of $\mathbf{x}$, and X_i is the ith row of $\mathbf{X}$. One might think to use the Wald test of $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$, where the p-value is given by

$$\mathbb{P}_{H_0}\left(\|\bar{X}_{\hat{\mathcal{C}}_k} - \bar{X}_{\hat{\mathcal{C}}_{k'}}\|_2 \geq \|\bar{x}_{\hat{\mathcal{C}}_k} - \bar{x}_{\hat{\mathcal{C}}_{k'}}\|_2 \right), \tag{4.5}$$

and $\|\bar{X}_{\hat{\mathcal{C}}_k} - \bar{X}_{\hat{\mathcal{C}}_{k'}}\|_2 \overset{H_0}{\sim} \left(\sigma\sqrt{\frac{1}{|\hat{\mathcal{C}}_k|} + \frac{1}{|\hat{\mathcal{C}}_{k'}|}}\right) \cdot \chi_q$. However, this ignores the fact that H_0 depends on the observed realization $\mathbf{x}$ through the estimated clusters $\mathcal{C}(\mathbf{x})$. We can expect a surprisingly large difference in the empirical centroids $\{\bar{x}_{\hat{\mathcal{C}}_k}\}_{k=1}^K$ of clusters $\{\hat{\mathcal{C}}_k\}_{k=1}^K$, even when the true centroids $\{\bar{\mu}_{\hat{\mathcal{C}}_k}\}_{k=1}^K$ are all the same, since $\hat{\mathcal{C}}_1, \hat{\mathcal{C}}_2, \ldots, \hat{\mathcal{C}}_K$ were obtained via clustering $\mathbf{x}$. This is illustrated in Figure 4.1(a). In short, the problem is that the null hypothesis $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$ is a function of the data. Since the Wald test does not account for this hypothesis selection procedure, it is extremely anti-conservative, as is shown in Figure 4.1(b).

In this chapter, we will develop a *selective inference* framework for testing the selected null hypothesis in (4.3). This framework exploits ideas from the extensive recent literature on selective inference within the context of regression, changepoint detection, and outlier detection [29, 70, 107, 61, 118, 49, 50, 20, 51]. The key idea is as follows: since we chose to test $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$ because $\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x})$, we can account for this hypothesis selection procedure by defining a p-value that conditions on the event $\left\{\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{X})\right\}$, where $\mathbf{X}$ is the random variable defined in (4.1), and $\mathcal{C}(\cdot)$ is a function that maps data to a clustering. This will yield a correctly sized test, as is shown in Figure 4.1(c).

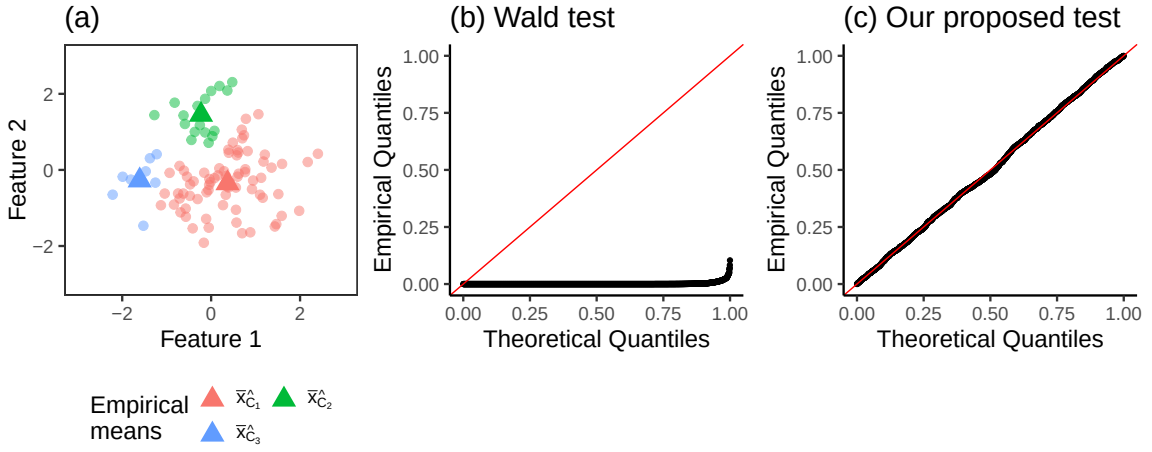

Figure 4.1: (a) A single draw from model (4.1) with $n = 100$, $q = 2$, $\sigma^2 = 1$, and $\boldsymbol{\mu} = \mathbf{0}_{n\times q}$, so that $\bar{\mu}_{\mathcal{G}} = 0_q$ for any $\mathcal{G} \subseteq \{1, 2, \ldots, n\}$. We cut the average-linkage hierarchical clustering dendogram of $\mathbf{x}$ to obtain three clusters ($\hat{\mathcal{C}}_1, \hat{\mathcal{C}}_2$ and $\hat{\mathcal{C}}_3$), and use color to indicate membership in the three clusters. For 2000 draws from model (4.1) with $n = 100$, $q = 2$, $\sigma^2 = 1$, and $\boldsymbol{\mu} = \mathbf{0}_{n\times q}$, QQ-plots of the theoretical Uniform$(0, 1)$ distribution against the p-values obtained from (b) the Wald test, defined in (4.5), and (c) our proposed test.

We are not the first to develop a selective inference framework to perform inference on means after clustering. For instance, [62] focuses on developing a selective inference framework for inference on means after *biclustering*, i.e. methods that cluster both the rows and columns of the data matrix. Another selective inference framework can be found in Chapter 3 of [14], which develops a test of the null hypothesis defined in (4.3) when convex clustering [43, 67, 106] is used to obtain estimated clusters. However, convex clustering is a relatively new technique that is not yet widely used in practice. We would like to instead develop a selective inference framework that can be applied to well-known and more widely used clustering methods, like agglomerative hierarchical clustering. Thus, in this chapter, we propose a selective inference framework that can be used to develop tests of the null hypothesis defined in (4.3), when agglomerative hierarchical clustering is used to obtain estimated clusters.

The rest of the chapter is organized as follows. In Section 4.2, we develop a general framework for testing the null defined in (4.3). In Section 4.3, we apply this framework

to derive exact tests of the null hypothesis that two clusters obtained from hierarchical clustering with single, average, or centroid linkage have the same mean. In Section 4.4, we propose an approximate Monte Carlo test based on the framework defined in (4.3) that can be applied to hierarchical clustering with any linkage; in particular, it can be applied to complete-linkage hierarchical clustering. In Section 4.5, we explore the performance of our testing procedure through numerical simulations. We conclude in Section 4.6 with a discussion.

4.2 Selective inference for clustering

In this section, we will introduce a general selective inference framework for testing the null hypothesis of no difference in means between two estimated clusters. This framework applies to estimated clusters obtained using any clustering method.

4.2.1 A test of no difference in means between two clusters

Let $\hat{\nu}$ be an n-vector of contrast coefficients defined by

$$\hat{\nu}_i = \begin{cases} \frac{1}{|\hat{\mathcal{C}}_k|}, & \text{if } i \in \hat{\mathcal{C}}_k, \\ -\frac{1}{|\hat{\mathcal{C}}_{k'}|}, & \text{if } i \in \hat{\mathcal{C}}_{k'}, \\ 0, & \text{if } i \notin \hat{\mathcal{C}}_k \cup \hat{\mathcal{C}}_{k'} \end{cases}. \tag{4.6}$$

Since $\boldsymbol{\mu}^T \hat{\nu} = \bar{\mu}_{\hat{\mathcal{C}}_k} - \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$, where $\bar{\mu}_{\hat{\mathcal{C}}_k}$ is defined in (4.2), testing (4.3) is equivalent to testing

$$H_0 : \boldsymbol{\mu}^T \hat{\nu} = 0_q \quad \text{v.s.} \quad H_1 : \boldsymbol{\mu}^T \hat{\nu} \neq 0_q. \tag{4.7}$$

In order to test for a difference in means between two clusters while conditioning on the fact that the clusters were estimated from the data, it is tempting to define the p-value to be

$$\mathbb{P}_{H_0}\left(\|\mathbf{X}^T \hat{\nu}\|_2 \geq \|\mathbf{x}^T \hat{\nu}\|_2 \mid \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{X}) \right). \tag{4.8}$$

Note that $\mathbf{x}^T \hat{\nu} = \bar{x}_{\hat{\mathcal{C}}_k} - \bar{x}_{\hat{\mathcal{C}}_{k'}}$, where $\bar{x}_{\hat{\mathcal{C}}_k}$ is defined in (4.4). Thus, (4.8) approximately asks, "Among all realizations of $\mathbf{X}$ that estimate the clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ we obtained from the

observed realization $\mathbf{x}$, how many realizations have a difference in means between $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ at least as large as the difference in means between $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ in $\mathbf{x}$?" Unfortunately, (4.8) is difficult to compute, because the conditional distribution of $\|\mathbf{X}^T\hat{\nu}\|_2$ given $\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{X})$ does not have a simple form. Instead, we define the p-value to be

$$p = \mathbb{P}_{H_0}\left(\|\mathbf{X}^T\hat{\nu}\|_2 \geq \|\mathbf{x}^T\hat{\nu}\|_2 \mid \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{X}),\ \boldsymbol{\pi}_{\hat{\nu}}^{\perp}\mathbf{X} = \boldsymbol{\pi}_{\hat{\nu}}^{\perp}\mathbf{x},\ \mathrm{dir}(\mathbf{X}^T\hat{\nu}) = \mathrm{dir}(\mathbf{x}^T\hat{\nu})\right), \quad (4.9)$$

where $\boldsymbol{\pi}_{\hat{\nu}}^{\perp} = \mathbf{I}_n - \frac{\hat{\nu}\hat{\nu}^T}{\|\hat{\nu}\|_2^2}$ is the orthogonal projection matrix onto the subspace orthogonal to $\mathrm{span}(\hat{\nu})$, and $\mathrm{dir}(w) = \frac{w}{\|w\|_2}$ if $w \neq 0$, and $\mathrm{dir}(w) = 0$ otherwise. In (4.9), we condition on the additional event $\{\mathrm{dir}(\mathbf{X}^T\hat{\nu}) = \mathrm{dir}(\mathbf{x}^T\hat{\nu})\}$ because the independence between $\|\mathbf{X}^T\hat{\nu}\|_2$ and $\mathrm{dir}(\mathbf{X}^T\hat{\nu})$ under H_0 ensures that computing (4.9) simply involves a truncated χ_q distribution. (We note that this would not be the case if we were to condition on the more general event $\{\mathbf{X}^T\hat{\nu} \in \mathrm{span}(\mathbf{x}^T\hat{\nu})\}$.) Conditioning on the additional event $\{\boldsymbol{\pi}_{\hat{\nu}}^{\perp}\mathbf{X} = \boldsymbol{\pi}_{\hat{\nu}}^{\perp}\mathbf{x}\}$ in (4.9) helps ensure that the support of the truncated χ_q distribution can be simply expressed.

Theorem 1 *Let $\phi \sim (\sigma\|\hat{\nu}\|_2) \cdot \chi_q$. For p defined in* (4.9)*,*

$$p = \mathbb{P}\left(\phi \geq \|\mathbf{x}^T\hat{\nu}\|_2 \mid \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\phi))\right), \quad (4.10)$$

where

$$\mathbf{x}'(\phi) = \mathbf{x} + \left(\frac{\phi - \|\mathbf{x}^T\hat{\nu}\|_2}{\|\hat{\nu}\|_2^2}\right)\hat{\nu}\left(\mathrm{dir}(\mathbf{x}^T\hat{\nu})\right)^T. \quad (4.11)$$

We prove Theorem 1 in Appendix C.1. Results similar to Theorem 1 have been used to develop selective inference frameworks for linear regression with groups of variables (Theorem 3.1 in [70], Lemma 1 in [118]), and to develop selective inference frameworks for change-point detection (Theorem 1 in [51]). It follows from Theorem 1 that efficiently computing the p-value defined in (4.9) amounts to efficiently characterizing the set

$$\mathcal{S} = \left\{\phi \geq 0 : \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\phi))\right\}, \quad (4.12)$$

since then for $\phi \sim (\sigma\|\hat{\nu}\|_2) \cdot \chi_q$, $p = \mathbb{P}\left(\phi \geq \|\mathbf{x}^T\hat{\nu}\|_2 \mid \phi \in \mathcal{S}\right)$.

4.2.2 Interpreting $\mathbf{x}'(\phi)$ *and* $\mathcal{S}$

Recall that $\mathbf{x}^T\hat{\nu} = \bar{x}_{\hat{\mathcal{C}}_k} - \bar{x}_{\hat{\mathcal{C}}_{k'}}$, where $\bar{x}_{\hat{\mathcal{C}}_k}$ is the kth cluster's empirical mean, as defined in (4.4). It follows that the ith row of $\mathbf{x}'(\phi)$ defined by (4.11) is given by

$$[\mathbf{x}'(\phi)]_i = \begin{cases} x_i + \left(\frac{|\hat{\mathcal{C}}_k|}{|\hat{\mathcal{C}}_k|+|\hat{\mathcal{C}}_{k'}|}\right)\left(\phi - \|\bar{x}_{\hat{\mathcal{C}}_k} - \bar{x}_{\hat{\mathcal{C}}_{k'}}\|_2\right) \text{dir}\left(\bar{x}_{\hat{\mathcal{C}}_k} - \bar{x}_{\hat{\mathcal{C}}_{k'}}\right), & \text{if } i \in \hat{\mathcal{C}}_k, \\ x_i - \left(\frac{|\hat{\mathcal{C}}_{k'}|}{|\hat{\mathcal{C}}_k|+|\hat{\mathcal{C}}_{k'}|}\right)\left(\phi - \|\bar{x}_{\hat{\mathcal{C}}_k} - \bar{x}_{\hat{\mathcal{C}}_{k'}}\|_2\right) \text{dir}\left(\bar{x}_{\hat{\mathcal{C}}_k} - \bar{x}_{\hat{\mathcal{C}}_{k'}}\right), & \text{if } i \in \hat{\mathcal{C}}_{k'}, \\ x_i, & \text{if } i \notin \hat{\mathcal{C}}_k \cup \hat{\mathcal{C}}_{k'}. \end{cases} \tag{4.13}$$

Thus, $\mathbf{x}'(\phi)_i$ is equal to x_i outside of clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$, and is otherwise equal to x_i perturbed by a vector with length proportional to $\phi - \|\bar{x}_{\hat{\mathcal{C}}_k} - \bar{x}_{\hat{\mathcal{C}}_{k'}}\|_2$, and with the same direction as $\bar{x}_{\hat{\mathcal{C}}_k} - \bar{x}_{\hat{\mathcal{C}}_{k'}}$, where $\bar{x}_{\hat{\mathcal{C}}_k}$. Thus, we can interpret $\mathbf{x}'(\phi)$ as a perturbed version of $\mathbf{x}$, where observations in clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ have been "pulled apart" or "pushed together" along the line through 0_q in the direction of $\bar{x}_{\hat{\mathcal{C}}_k} - \bar{x}_{\hat{\mathcal{C}}_{k'}}$, so that the perturbed data set $\mathbf{x}'(\phi)$ preserves direction of the difference in means between the clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$. In other words, the clusters cannot be "pushed past each other". Furthermore, $\mathcal{S} = \{\phi \in \mathbb{R} : \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\phi))\}$ describes the set of perturbed data sets $\mathbf{x}'(\phi)$ where $\mathcal{C}$ estimates the clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$.

We illustrate this interpretation with an example, where $\mathcal{C}$ is the function that results from performing average-linkage hierarchical clustering, and cutting the dendogram to obtain three clusters. Figure 4.2(a) displays a single realization $\mathbf{x}$ from (4.1), where we use color to indicate the clusters defined by $\mathcal{C}(\mathbf{x})$. Suppose that we are testing for no difference in means between the blue and orange clusters. Here, $\mathbf{x} = \mathbf{x}'(\phi)$ for $\phi = \|\bar{x}_{\hat{\mathcal{C}}_k} - \bar{x}_{\hat{\mathcal{C}}_{k'}}\|_2 = 4$. Figure 4.2(b) and Figure 4.2(c) display $\mathbf{x}'(\phi)$ for $\phi = 0$ and $\phi = 8$, respectively, where we again use color to indicate the clusters defined by $\mathcal{C}(\mathbf{x})$. Observe that Figure 4.2(b) displays a perturbed version of $\mathbf{x}$ displayed in Figure 4.2(a), where the blue and orange clusters have been "pushed together". There is now no mean difference between the blue and orange clusters, and average-linkage hierarchical clustering would no longer assign the blue and orange observations to different clusters. By contrast, observe that Figure 4.2(c) displays a perturbed version of $\mathbf{x}$ displayed in Figure 4.2(a), where the blue and orange clusters have been "pulled apart". The mean difference between the blue and orange clusters has been ex-

aggerated, and average-linkage hierarchical clustering would still assign the blue and orange observations to different clusters. Figure 4.2(d) displays $\mathcal{S} = \{\phi \geq 0 : \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\phi))\}$. We see that $4, 8 \in \mathcal{S}$, while $0 \notin \mathcal{S}$.

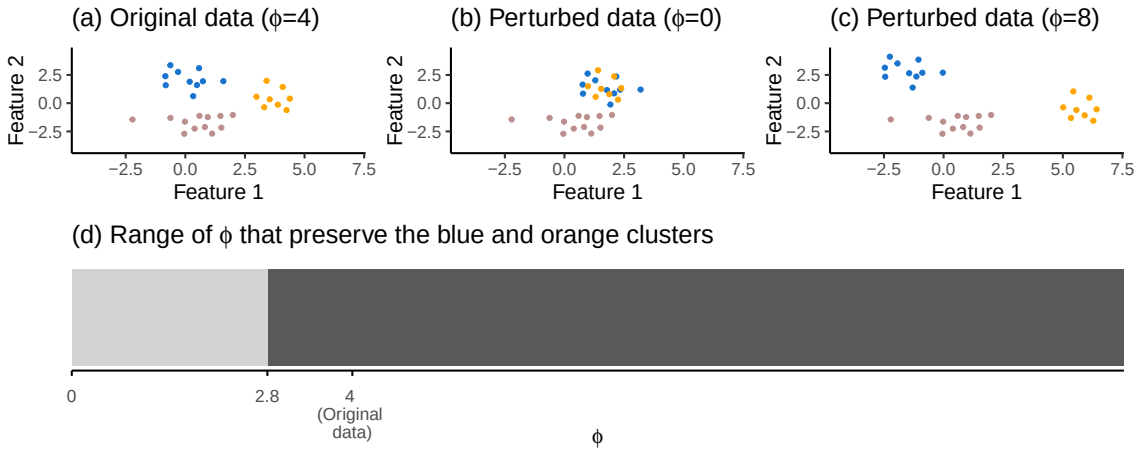

Figure 4.2: In the three panels on top, the observations belonging to $\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x})$ are displayed in blue and orange for three data sets: (a) the original data set $\mathbf{x}$, with $\phi = \|\mathbf{x}^T \hat{\nu}\|_2 = 4$, (b) a perturbed data set $\mathbf{x}'(\phi)$ with $\phi = 0$, and (c) a perturbed data set $\mathbf{x}'(\phi)$ with $\phi = 8$. In panel (d), the values of ϕ such that $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ appear in the clustering of $\mathbf{x}'(\phi)$ are displayed in dark grey.

4.3 Computing $\mathcal{S}$ for hierarchical clustering

In this section, we consider computing $\mathcal{S}$ defined in (4.12), when the clustering method $\mathcal{C}$ corresponds to a clustering obtained from agglomerative hierarchical clustering.

4.3.1 A brief review of agglomerative hierarchical clustering

Agglomerative hierarchical clustering produces a hierarchical sequence of clusterings, where the first clustering in the sequence contains n clusters, each containing a single observation, and the $(l+1)$th clustering in the sequence is created by merging the two most similar (or least dissimilar) clusters in the lth clustering in the sequence, for $l = 1, \ldots, n-1$ [39, 40]. This requires a measure of dissimilarity between two clusters. First, we define a measure of

dissimilarity between two observations in a data set $\mathbf{x}$, $d(\{i\},\{j\};\mathbf{x})$, for $i \neq j$. We assume throughout this chapter that $d(\{i\},\{j\};\mathbf{x})$ depends on x_i and x_j through $x_i - x_j$ only. For example, we could define $d(\{i\},\{j\};\mathbf{x}) = \|x_i - x_j\|_2^2$. Next, we choose a *linkage*, which extends this notion of dissimilarity to groups of observations. For example, *single-linkage* hierarchical clustering defines the dissimilarity between two groups of observations in a data set $\mathbf{x}$ to be $d(\mathcal{G},\mathcal{H};\mathbf{x}) = \min_{i\in\mathcal{G}, j\in\mathcal{H}} d(\{i\},\{j\};\mathbf{x})$.

The following algorithm describes the agglomerative hierarchical clustering procedure.

Algorithm 5 Agglomerative hierarchical clustering of a data set $\mathbf{x}$

- Let $\mathcal{C}^{(1)}(\mathbf{x}) = \{\{1\},\{2\},\ldots,\{n\}\}$.
- For $l = 1,\ldots,n-1$:
  1. Let $\left(\mathcal{G}_1^{(l)}(\mathbf{x}), \mathcal{G}_2^{(l)}(\mathbf{x})\right) = \underset{\mathcal{H}_1,\mathcal{H}_2\in\mathcal{C}^{(l)}(\mathbf{x}),\mathcal{H}_1\neq\mathcal{H}_2}{\arg\min} d(\mathcal{H}_1,\mathcal{H}_2;\mathbf{x})$. (We assume throughout this chapter that the minimizer is unique.)
  2. Merge $\mathcal{G}_1^{(l)}(\mathbf{x})$ and $\mathcal{G}_2^{(l)}(\mathbf{x})$ at the height of $d\left(\mathcal{G}_1^{(l)}(\mathbf{x}), \mathcal{G}_2^{(l)}(\mathbf{x});\mathbf{x}\right)$ in the dendrogram of $\mathbf{x}$, and update $\mathcal{C}^{(l+1)}(\mathbf{x}) = \mathcal{C}^{(l)}(\mathbf{x}) \cup \left\{\mathcal{G}_1^{(l)}(\mathbf{x}) \cup \mathcal{G}_2^{(l)}(\mathbf{x})\right\} \setminus \left\{\mathcal{G}_1^{(l)}(\mathbf{x}), \mathcal{G}_2^{(l)}(\mathbf{x})\right\}$.

In Algorithm 5, $\left(\mathcal{G}_1^{(l)}, \mathcal{G}_2^{(l)}\right)$ denotes the pair of clusters that are merged at the lth iteration of the hierarchical clustering procedure on $\mathbf{x}$, for $l = 1,2,\ldots,n-1$, and $\mathcal{C}^{(1)}(\mathbf{x}),\ldots,\mathcal{C}^{(n)}(\mathbf{x})$ denotes the hierarchical sequence of clusterings of $\mathbf{x}$.

Most commonly used linkages (e.g. single, complete, average, centroid) satisfy the *Lance-Williams update formula* [59, 82]. This means that there exist parameters $\alpha_1, \alpha_2, \beta$, and γ such that

$$d(\mathcal{G}_1 \cup \mathcal{G}_2, \mathcal{H};\mathbf{x}) = \alpha_1 d(\mathcal{G}_1,\mathcal{H};\mathbf{x}) + \alpha_2 d(\mathcal{G}_2,\mathcal{H};\mathbf{x}) + \beta d(\mathcal{G}_1,\mathcal{G}_2;\mathbf{x}) + \gamma|d(\mathcal{G}_1,\mathcal{H};\mathbf{x}) - d(\mathcal{G}_2,\mathcal{H};\mathbf{x})|, \tag{4.14}$$

where the parameters may depend on the cluster sizes $|\mathcal{G}_1|, |\mathcal{G}_2|$, and $|\mathcal{H}|$. We will show in Section 4.3.3 that the p-value defined in (4.9) can be computed using $\mathcal{O}(n^3)$ operations, for

any linkage that can be written in terms of a Lance-Williams update with $\gamma = 0$, such as average or centroid linkage [82].

4.3.2 Key results

Recall that we defined $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ to be two clusters obtained by applying a clustering method $\mathcal{C}$ to the original data set $\mathbf{x}$. Furthermore, recall from Section 4.2 that efficiently testing $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$ using (4.9) amounts to efficiently characterizing the set $\mathcal{S}$, where $\mathcal{S}$ is defined in (4.12) to be $\mathcal{S} = \{\phi \in \mathbb{R} : \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\phi))\}$, and $\mathbf{x}'(\phi)$ is defined in (4.11) to be a perturbed version of the original data set $\mathbf{x}$. In this subsection, we will present some important results that will help us characterize the set $\mathcal{S}$, when the clustering method $\mathcal{C}$ corresponds to a clustering obtained from agglomerative hierarchical clustering.

We first state a preliminary result involving the dissimilarities between groups of observations in any perturbed data set $\mathbf{x}'(\phi)$, which holds for any clustering method $\mathcal{C}$.

Lemma 1 *For any $\phi \geq 0$, and for any two sets $\mathcal{H}_1$ and $\mathcal{H}_2$ that are both contained in the estimated cluster $\hat{\mathcal{C}}_k$, both contained in the estimated cluster $\hat{\mathcal{C}}_{k'}$, or both contained in $\{1, 2, \ldots, n\} \backslash \hat{\mathcal{C}}_k \cup \hat{\mathcal{C}}_{k'}$,*

$$d(\mathcal{H}_1, \mathcal{H}_2; \mathbf{x}'(\phi)) = d(\mathcal{H}_1, \mathcal{H}_2; \mathbf{x}),$$

where $\mathbf{x}'(\phi)$ is the data set obtained by perturbing observations belonging to $\hat{\mathcal{C}}_k \cup \hat{\mathcal{C}}_{k'}$ in the observed data set $\mathbf{x}$ according to (4.13)*.*

Lemma 1 follows directly from the assumption that the measure of dissimilarity $d(\{i\}, \{j\}; \mathbf{x})$ depends on x_i and x_j through $x_i - x_j$ only, and the definition of $\mathbf{x}'(\phi)$ in (4.11) and (4.13). Furthermore, Lemma 1 says that any perturbed data set $\mathbf{x}'(\phi)$ preserves the dissimilarities between any two groups of observations both belonging to the estimated cluster $\hat{\mathcal{C}}_k$, between any two groups of observations both belonging to the estimated cluster $\hat{\mathcal{C}}_{k'}$, and between any two groups of observations that do not belong to $\hat{\mathcal{C}}_k \cup \hat{\mathcal{C}}_{k'}$. Surprisingly, this simple result is the key to deriving a sufficient and necessary condition for a perturbed data set $\mathbf{x}'(\phi)$ preserving $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$, when $\mathcal{C}$ corresponds to a clustering obtained from agglomerative hierarchical clustering.

Lemma 2 *Suppose that $\mathcal{C} = \mathcal{C}^{(n-K+1)}$, i.e. the clustering method used is the function that results from cutting the hierarchical clustering dendrogram to obtain K clusters. Then, $\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\phi))$ if and only if*

$$\underset{\mathcal{H}_1,\mathcal{H}_2 \in \mathcal{C}^{(l)}(\mathbf{x}), \mathcal{H}_1 \neq \mathcal{H}_2}{\arg\min} d(\mathcal{H}_1, \mathcal{H}_2; \mathbf{x}'(\phi)) = \left(\mathcal{G}_1^{(l)}(\mathbf{x}), \mathcal{G}_2^{(l)}(\mathbf{x})\right), \quad \forall\ l = 1, 2, \ldots, n-K, \tag{4.15}$$

where $(\mathcal{G}_1^{(l)}(\mathbf{x}), \mathcal{G}_2^{(l)}(\mathbf{x}))$ is the pair of clusters that was merged at the lth step of the hierarchical clustering algorithm applied to $\mathbf{x}$. Furthermore, $\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\phi))$ if and only if

$$\mathcal{C}^{(l)}(\mathbf{x}) = \mathcal{C}^{(l)}(\mathbf{x}'(\phi)), \quad \forall\ l = 1, 2, \ldots, n-K+1. \tag{4.16}$$

The proof of Lemma 2 is in Appendix C.2. The result in Lemma 2 is a bit surprising: it says that the only way a pair of clusters ($\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$) in $\mathcal{C}^{(n-K+1)}(\mathbf{x})$ can appear in $\mathcal{C}^{(n-K+1)}(\mathbf{x}'(\phi))$ is if *all* of the merges that occur during the first $n-K$ steps of the hierarchical clustering algorithm are completely identical, i.e. $\mathcal{C}^{(l)}(\mathbf{x}) = \mathcal{C}^{(l)}(\mathbf{x}'(\phi))$ for all $l = 1, 2, \ldots, n-K+1$. We now ask: is it possible for hierarchical clustering on $\mathbf{x}$ and $\mathbf{x}'(\phi)$ result in identical clusters in the first $n-K$ steps, but merge these clusters at different heights? The following result answers this question in the negative.

Lemma 3 *Suppose that $\mathcal{C} = \mathcal{C}^{(n-K+1)}$, i.e. the clustering method used is the function that results from cutting the hierarchical clustering dendrogram to obtain K clusters. Then, for any $\phi \geq 0$,*

$$d\left(\mathcal{G}_1^{(l)}(\mathbf{x}), \mathcal{G}_2^{(l)}(\mathbf{x}); \mathbf{x}'(\phi)\right) = d\left(\mathcal{G}_1^{(l)}(\mathbf{x}), \mathcal{G}_2^{(l)}(\mathbf{x}); \mathbf{x}\right), \quad \forall\ l = 1, 2, \ldots, n-K, \tag{4.17}$$

where $\left(\mathcal{G}_1^{(l)}(\mathbf{x}), \mathcal{G}_2^{(l)}(\mathbf{x})\right)$ is the pair of clusters that was merged at the lth step of the hierarchical clustering algorithm applied to $\mathbf{x}$, and $\mathbf{x}'(\phi)$ is the data set obtained by perturbing observations belonging to $\hat{\mathcal{C}}_k \cup \hat{\mathcal{C}}_{k'}$ in the observed data set $\mathbf{x}$ according to (4.13).

Recall that $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ are two estimated clusters in $\mathcal{C}(\mathbf{x}) = \mathcal{C}^{(n-K+1)}(\mathbf{x})$. Thus, Lemma 3 follows from Lemma 1, and the fact that clusters cannot become unmerged once they are merged. Lemma 3 says that the dissimilarity between any two sets of observations that were

merged in $\mathbf{x}$ during the lth step of the hierarchical clustering procedure on $\mathbf{x}$ is preserved by *any* perturbed data set $\mathbf{x}'(\phi)$.

Recall from Lemma 2 that a perturbed data set $\mathbf{x}'(\phi)$ preserves $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ if and only if $\mathbf{x}'(\phi)$ preserves which clusters are merged in the first $n-K$ steps of the hierarchical clustering algorithm. Furthermore, Lemma 3 told us that it is impossible for hierarchical clustering on $\mathbf{x}$ and $\mathbf{x}'(\phi)$ to result in identical clusters in the first $n-K$ steps, but merge these clusters at different heights. Thus, a perturbed data set $\mathbf{x}'(\phi)$ preserves $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ if and only if $\mathbf{x}'(\phi)$ preserves which clusters are merged, *and* the height at which they are merged, in the first $n-K$ steps of the hierarchical clustering algorithm. In other words, $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ are preserved if and only if the dendogram of $\mathbf{x}'(\phi)$ and the dendogram of $\mathbf{x}$ are identical (up to re-ordering of the leaves) below the $(n-K)$th merge. This is illustrated in Figure 4.3, which displays the dendrogram obtained from average-linkage hierarchical clustering, cut at the red dotted line to yield three clusters, for the original data set $\mathbf{x}$, and perturbed data sets $\mathbf{x}'(\phi)$ with $\phi = 4$ and $\phi = 1$. Observe that the perturbed data set with $\phi = 4$ displayed in Figure 4.3(b) and the original data set with $\phi = 5.4$ displayed in Figure 4.3(a) have identical dendrograms below the red dotted line, and so $\mathbf{x}'(4)$ and $\mathbf{x}'(5.4)$ preserve the clusters $\hat{\mathcal{C}}_k = \{1,5,2,4\}$ and $\hat{\mathcal{C}}_{k'} = \{6,7,8,9,10\}$. By contrast, the perturbed data set with $\phi = 1$ does not have an identical dendrogram below the red dotted line, and so $\mathbf{x}'(1)$ does not preserve the clusters $\hat{\mathcal{C}}_k = \{1,5,2,4\}$ and $\hat{\mathcal{C}}_{k'} = \{6,7,8,9,10\}$.

We now rewrite (4.15) in a convenient form. Observe that (4.15) is true if and only if

$$d(\mathcal{A},\mathcal{B};\mathbf{x}'(\phi)) > d\left(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x});\mathbf{x}'(\phi)\right), \tag{4.18}$$
$$\forall\ l = 1,2,\ldots,n-K,\ \forall\ \mathcal{A},\mathcal{B} \in \mathcal{C}^{(l)}(\mathbf{x}),\ \mathcal{A} \neq \mathcal{B},\ (\mathcal{A},\mathcal{B}) \neq \left(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x})\right).$$

Applying Lemma 3 to the right-hand-side of the inequalities in (4.18) yields

$$d(\mathcal{A},\mathcal{B};\mathbf{x}'(\phi)) > d\left(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x});\mathbf{x}\right), \quad \forall\ \mathcal{A},\mathcal{B} \in \mathcal{C}^{(l)}(\mathbf{x}),\ \mathcal{A} \neq \mathcal{B},\ (\mathcal{A},\mathcal{B}) \neq \left(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x})\right).$$

Thus, we can conveniently characterize $\mathcal{S} = \{\phi \geq 0 : \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\phi))\}$ defined in (4.12) as follows.

Theorem 2 *Suppose that $\mathcal{C} = \mathcal{C}^{(n-K+1)}$, i.e. the clustering method used is the function*

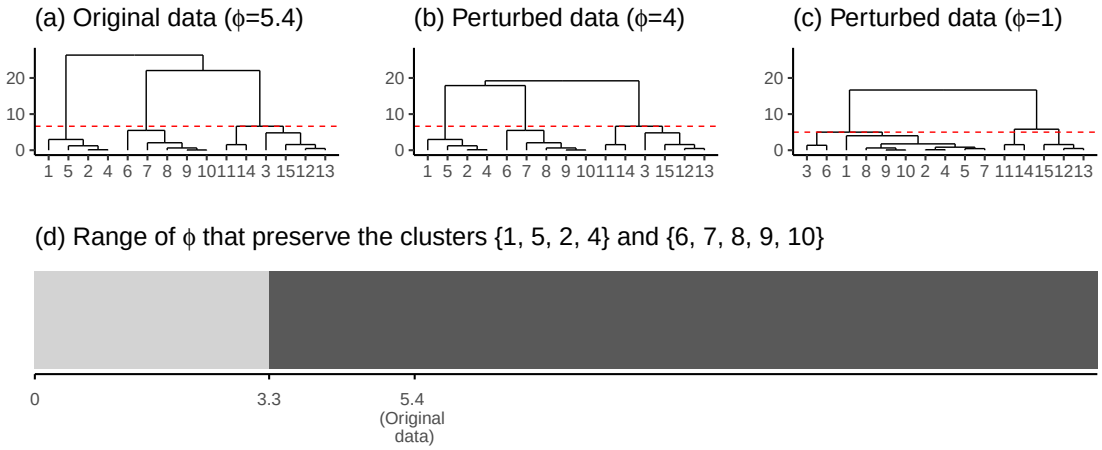

Figure 4.3: In (a)–(c), the dendrogram obtained from average-linkage hierarchical clustering, cut to yield three clusters, is displayed for three data sets: (a) the original data set $\mathbf{x}$, with $\phi = \|\mathbf{x}^T\hat{\nu}\|_2 = 5.4$, (b) a perturbed data set $\mathbf{x}'(\phi)$ with $\phi = 4$, and (c) a perturbed data set $\mathbf{x}'(\phi)$ with $\phi = 1$. In panel (d), the values of ϕ such that $\hat{\mathcal{C}}_k = \{1, 5, 2, 4\}$ and $\hat{\mathcal{C}}_{k'} = \{6, 7, 8, 9, 10\}$ appear in the clustering of $\mathbf{x}'(\phi)$ are displayed in dark grey; this is the set $\mathcal{S}$ defined in (4.12).

that results from cutting the hierarchical clustering dendrogram to obtain K clusters. Then,

$$\mathcal{S} = \bigcap_{l=1}^{n-K} \bigcap_{\substack{\mathcal{A},\mathcal{B}\in\mathcal{C}^{(l)}(\mathbf{x}),\mathcal{A}\neq\mathcal{B},\\ (\mathcal{A},\mathcal{B})\neq\left(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x})\right)}} \left\{\phi \geq 0 : d(\mathcal{A},\mathcal{B};\mathbf{x}'(\phi)) > d\left(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x});\mathbf{x}\right)\right\}, \quad (4.19)$$

where $\left(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x})\right)$ is the pair of clusters that was merged at the lth step of the hierarchical clustering algorithm applied to $\mathbf{x}$. Furthermore, (4.19) *is the intersection of $\mathcal{O}(n^3)$ sets.*

It follows from Theorem 2 that we can compute $\mathcal{S}$ by solving $\mathcal{O}(n^3)$ inequalities in ϕ. Whether these inequalities can be efficiently solved in closed form depends on the choice of the measure of dissimilarity between pairs of observations, and the choice of linkage. When the measure of dissimilarity between pairs of observations is defined to be squared Euclidean distance, it turns out that the dissimilarity between any pair of observations in the perturbed data set $\mathbf{x}'(\phi)$ can be written as a quadratic function of ϕ.

Lemma 4 *Suppose that $\mathcal{C} = \mathcal{C}^{(n-K+1)}$, i.e. the clustering method used is the function that results from cutting the hierarchical clustering dendrogram to obtain K clusters, and we define the dissimilarity between a pair of observations in data set $\mathbf{x}$ to be $d(\{i\},\{j\};\mathbf{x}) = \|x_i - x_j\|_2^2$. Then, for all $i \neq j$,*

$$d(\{i\},\{j\};\mathbf{x}'(\phi)) = a_{ij}\phi^2 + b_{ij}\phi + c_{ij}, \tag{4.20}$$

for $a_{ij} = \left(\frac{\hat{\nu}_i - \hat{\nu}_j}{\|\hat{\nu}\|_2}\right)^2$, $b_{ij} = 2\left(\sqrt{a_{ij}}\langle dir(\mathbf{x}^T\hat{\nu}), x_i - x_j\rangle - a_{ij}\|\mathbf{x}^T\hat{\nu}\|_2\right)$, $c_{ij} = \|x_i - x_j - \sqrt{a_{ij}}(\mathbf{x}^T\hat{\nu})\|_2^2$.

Lemma 4 follows from applying algebra to the definition of $\mathbf{x}'(\phi)$ in (4.11). This result will help us efficiently solve the inequalities in (4.19) in closed form for three commonly used linkages: single, average, and centroid.

4.3.3 Three commonly used linkages

We will now apply the results derived in Section 4.3.2 in order to show that the set $\mathcal{S}$ defined in (4.12) can be efficiently computed, when the measure of dissimilarity between pairs of observations is squared Euclidean distance, and the linkage used is centroid, average, or single linkage.

Centroid and average linkage

Average-linkage hierarchical clustering defines the dissimilarity between two groups of observations to be $d(\mathcal{G},\mathcal{H};\mathbf{x}) = \frac{1}{|\mathcal{G}||\mathcal{H}|}\sum_{i\in\mathcal{G}, j\in\mathcal{H}} d(\{i\},\{j\};\mathbf{x})$, and centroid-linkage hierarchical clustering defines the dissimilarity between two groups of observations $\mathcal{G}$ and $\mathcal{H}$ to be the dissimilarity between the two centroids, $\frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}} x_i$ and $\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} x_i$. It turns out that centroid and average linkage both satisfy the Lance-Williams update formula defined in (4.14) with $\gamma = 0$ [82]. In other words,

$$d(\mathcal{G}_1 \cup \mathcal{G}_2, \mathcal{H}; \mathbf{x}'(\phi)) = \alpha_1 d(\mathcal{G}_1, \mathcal{H}; \mathbf{x}'(\phi)) + \alpha_2 d(\mathcal{G}_2, \mathcal{H}; \mathbf{x}'(\phi)) + \beta d(\mathcal{G}_1, \mathcal{G}_2; \mathbf{x}'(\phi)), \tag{4.21}$$

where $\alpha_1 = \left(\frac{|\mathcal{G}_1|}{|\mathcal{G}_1|+|\mathcal{G}_2|}\right)$, $\alpha_2 = 1 - \alpha_1$, and $\beta = 0$ for average linkage, and $\alpha_1 = \left(\frac{|\mathcal{G}_1|}{|\mathcal{G}_1|+|\mathcal{G}_2|}\right)$, $\alpha_2 = 1 - \alpha_1$, and $\beta = \alpha_1\alpha_2$ for centroid linkage. Furthermore, recall from Lemma 4 that when the dissimilarity between pairs of observations is given by squared Euclidean distance,

$d(\{i\},\{j\},\mathbf{x}'(\phi))$ is a quadratic function of ϕ for all $i \neq j$. The following result follows from the fact that linear combinations of quadratic functions of ϕ are quadratic functions of ϕ.

Proposition 6 *Suppose that we define the dissimilarity between a pair of observations in data set* $\mathbf{x}$ *to be* $d(\{i\},\{j\};\mathbf{x}) = \|x_i - x_j\|_2^2$*, and we use either centroid or average linkage to define the dissimilarity between groups of observations. Then,*

1. $d(\mathcal{A},\mathcal{B};\mathbf{x}'(\phi))$ *is a quadratic function of* ϕ *for all sets* $\mathcal{A},\mathcal{B}$*.*

2. *Given the coefficients corresponding to the quadratic functions* $d(\mathcal{G}_1,\mathcal{H};\mathbf{x}'(\phi))$*,* $d(\mathcal{G}_2,\mathcal{H};\mathbf{x}'(\phi))$*, and* $d(\mathcal{G}_1,\mathcal{G}_2;\mathbf{x}'(\phi))$*, we can compute the coefficients corresponding to the quadratic function* $d(\mathcal{G}_1 \cup \mathcal{G}_2,\mathcal{H};\mathbf{x}'(\phi))$ *in constant time, using* (4.21)*.*

Now, recall from Theorem 2 that

$$\mathcal{S} = \bigcap_{l=1}^{n-K} \bigcap_{\substack{\mathcal{A},\mathcal{B}\in\mathcal{C}^{(l)}(\mathbf{x}),\mathcal{A}\neq\mathcal{B},\\ (\mathcal{A},\mathcal{B})\neq\left(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x})\right)}} \left\{\phi \geq 0 : d(\mathcal{A},\mathcal{B};\mathbf{x}'(\phi)) > d\left(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x});\mathbf{x}\right)\right\} \equiv \bigcap_{l=1}^{n-K} \mathcal{S}^{(l)}, \tag{4.22}$$

where $\left(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x})\right)$ is the pair of clusters that was merged in $\mathbf{x}$ at the lth step of the hierarchical clustering algorithm (Algorithm 5). We can now apply Proposition 6 to obtain a computationally efficient procedure to compute $\mathcal{S}$. This procedure is described in Algorithm 6.

Since $\mathcal{S}$ is the intersection of $\mathcal{O}(n^3)$ sets (Theorem 2), and Algorithm 6 computes each of these sets using a constant number of operations, it follows that computing $\mathcal{S}$ requires $\mathcal{O}(n^3)$ operations. We can also use Proposition 6 and Algorithm 6 to compute $\mathcal{S}$ using $\mathcal{O}(n^3)$ operations for other linkages that satisfy (4.21), i.e. the Lance-Williams update formula defined in (4.14) with $\gamma = 0$. For example, weighted, median, and Ward linkage also satisfy (4.21) [82].

Single linkage

Single-linkage hierarchical clustering defines the dissimilarity between two groups of observations to be $d(\mathcal{G},\mathcal{H};\mathbf{x}) = \min_{i\in\mathcal{G},j\in\mathcal{H}} d(\{i\},\{j\};\mathbf{x})$. Single linkage satisfies the Lance-Williams

Algorithm 6 Computing $\mathcal{S}$ defined in (4.12) for average-linkage or centroid-linkage hierarchical clustering

1. Compute the set $\mathcal{S}^{(1)}$ defined in (4.22):

   (a) Compute the coefficients corresponding to the $\binom{n}{2}$ quadratic functions of the form $d(\{i\}, \{j\}; \mathbf{x}'(\phi))$ with $i \neq j$, using Lemma 4.

   (b) Using these coefficients, compute each of the $\binom{n}{2} - 1$ sets intersected in $\mathcal{S}^{(1)}$ in constant time by solving a quadratic equation. Each of these sets is either an interval, or the union of two intervals.

2. For $l = 2, \ldots, n - K$, compute the set $\mathcal{S}^{(l)}$ defined in (4.22):

   (a) Compute the coefficients corresponding to the $\binom{n-l+1}{2}$ quadratic functions of the form $d(\mathcal{A}, \mathcal{B}; \mathbf{x}'(\phi))$ with $\mathcal{A}, \mathcal{B} \in \mathcal{C}^{(l)}(\mathbf{x})$ and $\mathcal{A} \neq \mathcal{B}$. Since Algorithm 5 says that

   $$\mathcal{C}^{(l)}(\mathbf{x}) = \mathcal{C}^{(l-1)}\mathbf{x} \cup \left\{\mathcal{G}_1^{(l-1)}(\mathbf{x}) \cup \mathcal{G}_2^{(l-1)}(\mathbf{x})\right\} \setminus \left\{\mathcal{G}_1^{(l-1)}(\mathbf{x}), \mathcal{G}_2^{(l-1)}(\mathbf{x})\right\},$$

   it follows that there are two types of quadratic functions: those that involve $\mathcal{G}_1^{(l-1)}(\mathbf{x}) \cup \mathcal{G}_2^{(l-1)}(\mathbf{x})$, and those that do not. There are $\binom{n-l}{2}$ of the former, and their coefficients were computed in the previous iteration. There are $n - l$ of the latter, and each of their coefficients can be computed in constant time, using (4.21), by Proposition 6.

   (b) Using these coefficients, compute each of the $\binom{n-l+1}{2} - 1$ sets intersected in $\mathcal{S}^{(l)}$ in constant time by solving a quadratic equation. Each of these sets is either an interval, or the union of two intervals.

3. Compute the set $\mathcal{S} = \bigcap_{l=1}^{n-K} \mathcal{S}^{(l)}$.

update formula defined in (4.14), but with $\gamma \neq 0$ [82]. Thus, Proposition 6 does not apply for single-linkage hierarchical clustering, and we cannot apply the procedure described in

Algorithm 6 to compute $\mathcal{S}$. Fortunately, there is nevertheless a simple way to characterize $\mathcal{S}$. Recall from Theorem 2 that

$$\mathcal{S} = \bigcap_{l=1}^{n-K} \bigcap_{\substack{\mathcal{A},\mathcal{B}\in\mathcal{C}^{(l)}(\mathbf{x}),\mathcal{A}\neq\mathcal{B},\\ (\mathcal{A},\mathcal{B})\neq\left(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x})\right)}} \left\{\phi \geq 0 : d(\mathcal{A},\mathcal{B};\mathbf{x}'(\phi)) > d\left(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x});\mathbf{x}\right)\right\}. \quad (4.23)$$

Applying the definition of single-linkage hierarchical clustering to (4.23) yields

$$\mathcal{S} = \bigcap_{l=1}^{n-K} \bigcap_{\substack{\mathcal{A},\mathcal{B}\in\mathcal{C}^{(l)}(\mathbf{x}),\mathcal{A}\neq\mathcal{B},\\ (\mathcal{A},\mathcal{B})\neq\left(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x})\right)}} \bigcap_{i\in\mathcal{A}} \bigcap_{j\in\mathcal{B}} \left\{\phi \geq 0 : d(\{i\},\{j\};\mathbf{x}'(\phi)) > d\left(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x});\mathbf{x}\right)\right\}. \quad (4.24)$$

This straightforward characterization of $\mathcal{S}$ seems easy to work with, because it only involves dissimilarities between two observations in the perturbed data set $\mathbf{x}'(\phi)$), rather than dissimilarities between two sets of observations in the perturbed data set $\mathbf{x}'(\phi)$). However, the characterization in (4.24) does not help us compute $\mathcal{S}$ efficiently, because there are more than $\mathcal{O}(n^3)$ sets intersected in (4.19). Fortunately, we are able to show that most of the sets in (4.19) can be safely eliminated from the intersection.

Lemma 5 *Suppose that $\mathcal{C}$ is the function that results from cutting the single-linkage hierarchical clustering dendrogram to get K clusters. Then, for $\mathcal{S}$ defined in* (4.12)*,*

$$\mathcal{S} = \bigcap_{\substack{\mathcal{A},\mathcal{B}\in\mathcal{C}^{(n-K)}(\mathbf{x}),\mathcal{A}\neq\mathcal{B},\\ (\mathcal{A},\mathcal{B})\neq\left(\mathcal{G}_1^{(n-K)}(\mathbf{x}),\mathcal{G}_2^{(n-K)}(\mathbf{x})\right)}} \bigcap_{i\in\mathcal{A}} \bigcap_{j\in\mathcal{B}} \left\{\phi \geq 0 : d(\{i\},\{j\};\mathbf{x}'(\phi)) > d\left(\mathcal{G}_1^{(n-K)}(\mathbf{x}),\mathcal{G}_2^{(n-K)}(\mathbf{x});\mathbf{x}\right)\right\},$$

where $\left(\mathcal{G}_1^{(n-K)}(\mathbf{x}),\mathcal{G}_2^{(n-K)}(\mathbf{x})\right)$ is the pair of clusters that merged in $\mathbf{x}$ at the $(n-K)$th step of the hierarchical clustering algorithm, and the intersection is taken over $\mathcal{O}(n^2)$ sets.

Lemma 5 says that we only need to take the intersection over the $\mathcal{O}(n^2)$ sets with $l = n-K$ in (4.24), and is proved in Appendix C.3. Thus, we can compute $\mathcal{S}$ by solving $\mathcal{O}(n^2)$ inequalities in ϕ that involve $d(\{i\},\{j\};\mathbf{x}'(\phi))$, for $i \neq j$. To solve these inequalities, we recall from Lemma 4 that $d(\{i\},\{j\};\mathbf{x}'(\phi))$ is a quadratic function of ϕ, when $i \neq j$, and the dissimilarity between pairs of observations is measured using squared Euclidean distance. Applying this result to Lemma 5 yields the following characterization of $\mathcal{S}$.

Proposition 7 *Suppose that we define the dissimilarity between pairs of observations to* $d(\{i\},\{j\};\mathbf{x}) = \|x_i - x_j\|_2^2$*, and we define* $\mathcal{C}$ *to be the map that results from cutting the single-linkage hierarchical clustering dendrogram to get* K *clusters. Then,*

$$\mathcal{S} = \bigcap_{\substack{\mathcal{A},\mathcal{B}\in\mathcal{C}^{(n-K)}(\mathbf{x}),\mathcal{A}\neq\mathcal{B},\\ (\mathcal{A},\mathcal{B})\neq\left(\mathcal{G}_1^{(n-K)}(\mathbf{x}),\mathcal{G}_2^{(n-K)}(\mathbf{x})\right)}} \bigcap_{i\in\mathcal{A}} \bigcap_{j\in\mathcal{B}} \left\{\phi \geq 0 : a_{ij}\phi^2 + b_{ij}\phi + c_{ij} \geq d\left(\mathcal{G}_1^{(n-K)}(\mathbf{x}),\mathcal{G}_2^{(n-K)}(\mathbf{x});\mathbf{x}\right)\right\},$$

where $\left(\mathcal{G}_1^{(n-K)}(\mathbf{x}),\mathcal{G}_2^{(n-K)}(\mathbf{x})\right)$ *is the pair of clusters that merged in* $\mathbf{x}$ *at the* $(n-K)$*th step of the hierarchical clustering algorithm,* a_{ij}, b_{ij}*, and* c_{ij} *are defined in Lemma 4, and the intersection is taken over* $\mathcal{O}(n^2)$ *sets.*

Proposition 7 says that $\mathcal{S}$ can be computed by solving $\mathcal{O}(n^2)$ quadratic inequalities in ϕ, where each inequality yields either a single interval, or a union of two intervals. Thus, computing $\mathcal{S}$ requires $\mathcal{O}(n^2)$ operations.

4.4 Approximating the p-value when analytically characterizing $\mathcal{S}$ is intractable

We have now shown that $\mathcal{S}$ can be efficiently computed for single-linkage, average-linkage, and centroid-linkage hierarchical clustering, where the pairwise dissimilarity between observations is given by squared Euclidean distance. Given $\mathcal{S}$, it follows from the fact that $p = \mathbb{P}(\phi \geq \|\mathbf{x}^T\hat{\nu}\|_2 \mid \phi \in \mathcal{S})$ for $\phi \sim (\sigma\|\hat{\nu}\|)\cdot\chi_q$ (Section 4.2.1) that the p-value defined in (4.9) can be efficiently computed. However, we may be interested in computing the p-value defined in (4.9), using other linkages for which $\mathcal{S}$ cannot be efficiently computed. For example, we may be interested in computing $\mathcal{S}$ for complete-linkage hierarchical clustering, where $d(\mathcal{G},\mathcal{H};\mathbf{x}) = \max_{i\in\mathcal{G},j\in\mathcal{H}} d(\{i\},\{j\};\mathbf{x})$. Alternatively, we may be interested in defining the pairwise dissimilarity between observations using a function other than Euclidean distance, such as non-squared Euclidean distance.

Thus, we will develop a Monte Carlo approximation to the p-value p. Recalling from (4.12) that $\mathcal{S} = \{\phi \geq 0 : \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\phi))\}$, we can rewrite $p = \mathbb{P}(\phi \geq \|\mathbf{x}^T\hat{\nu}\|_2 \mid \phi \in \mathcal{S})$ as

$$p = \mathbb{P}\left(\phi \geq \|\mathbf{x}^T\hat{\nu}\|_2 \mid \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\phi))\right) = \frac{\mathbb{E}\left[\mathbb{1}\{\phi \geq \|\mathbf{x}^T\hat{\nu}\|_2, \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\phi))\}\right]}{\mathbb{E}\left[\mathbb{1}\{\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\phi))\}\right]}. \quad (4.25)$$

Recalling that $\phi \sim (\sigma\|\hat{\nu}\|) \cdot \chi_q$, we could sample $\phi_1, \ldots, \phi_N \overset{i.i.d.}{\sim} (\sigma\|\hat{\nu}\|_2) \cdot \chi_q$, and use the naive approximation

$$p \approx \frac{\sum_{i=1}^{N} \mathbb{1}\{\phi_i \geq \|\mathbf{x}^T\hat{\nu}\|_2, \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\phi_i))\}}{\sum_{i=1}^{N} \mathbb{1}\{\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\phi_i))\}}. \tag{4.26}$$

However, when $\|\mathbf{x}^T\hat{\nu}\|_2$ is in the tail of the $(\sigma\|\hat{\nu}\|_2) \cdot \chi_q$ distribution, the right-hand side of (4.26) provides a poor approximation of the p-value for finite values of N. Thus, we adopt an importance sampling approach, along the lines of a selective inference procedure for the group lasso proposed by [118]. We sample $\omega_1, \ldots, \omega_N \overset{i.i.d}{\sim} N(\|\mathbf{x}^T\hat{\nu}\|_2, \sigma^2\|\hat{\nu}\|_2^2)$, and use the approximation

$$p \approx \frac{\sum_{i=1}^{N} \pi_i \mathbb{1}\{\omega_i \geq \|\mathbf{x}^T\hat{\nu}\|_2, \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\omega_i))\}}{\sum_{i=1}^{N} \pi_i \mathbb{1}\{\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(\mathbf{x}'(\omega_i))\}}, \tag{4.27}$$

for $\pi_i = \frac{f_1(\omega_i)}{f_2(\omega_i)}$, where f_1 is the density of a $(\sigma\|\hat{\nu}\|_2) \cdot \chi_q$ random variable, and f_2 is the density of a $N(\|\mathbf{x}^T\hat{\nu}\|_2, \sigma^2\|\hat{\nu}\|_2^2)$ random variable.

The main advantage of the approximate Monte Carlo test described in this section is that it can be applied to test $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$ defined in (4.3) for estimated clusters obtained from *any* clustering method $\mathcal{C}$. Of course, when we use single, average, or centroid linkage hierarchical clustering, and the pairwise dissimilarity between observations is given by squared Euclidean distance, it is generally preferable to use the exact test described in Section 4.3.

4.5 Simulation results

In this section, we will evaluate the Type I error rate and power of the test of $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$ proposed in Section 4.2.1, where the clustering method $\mathcal{C}$ is the map that results from performing hierarchical clustering and cutting the dendrogram to yield K clusters, for $K = 3$.

4.5.1 Type I error rate under a global null

To evaluate the Type I error rate of the test proposed in Section 4.2.1, we generate data from model (4.1) with $\mu = \mathbf{0}_{n\times q}$, so that there is no difference in means between any two groups of observations. We simulate 2000 data sets for $n = 150$, $\sigma \in \{1, 2, 10\}$, and $q \in \{2, 10, 100\}$. For each data set, we perform hierarchical clustering with average, centroid, single, or complete linkage, cut the dendrogram to yield three clusters, then test $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$ for an arbitrary pair of clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$. For average, centroid, and single-linkage hierarchical clustering, we compute the p-value exactly, using the methods described in Section 4.3. For complete-linkage hierarchical clustering, we compute the p-value approximately, using the Monte Carlo method described in Section B.1.

Figure 4.4 displays plots of the quantiles of the empirical p-values resulting from the test proposed in Section 4.2.1 against the quantiles of the theoretical Uniform$(0, 1)$ distribution. Across all choices of σ, all choices of dimension q, and all choices of linkage, the empirical p-values in Figure 4.4 approximately follow a Uniform$(0, 1)$ distribution. Thus, the test proposed in Section 4.2.1 controls the Type I error rate.

4.5.2 Power under a model with three clusters

Recall that $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ are two estimated clusters in $\mathcal{C}(\mathbf{x})$. We can consider at least two notions of power for the test of $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$ proposed in Section 4.2.1.

First, suppose that we generate data from model (4.1), with three unique values in the set $\{\mu_1, \ldots, \mu_n\}$, so that the observations belong to three true clusters. We could consider the power of the test to detect a difference in means between two *true* clusters. However, we would only test the null hypothesis that there is no difference in means between two true clusters when two estimated clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ are exact matches for two true clusters. Thus, this notion of *conditional* power is only concerned with the performance of the test when two estimated clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ are exact matches for two true clusters. However, it is somewhat unrealistic to assume that we estimate two clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ that match (or even nearly match) two true clusters, especially if we do not use the correct number of clusters in the hierarchical clustering algorithm, i.e. if $K \neq 3$. Furthermore, in practice,

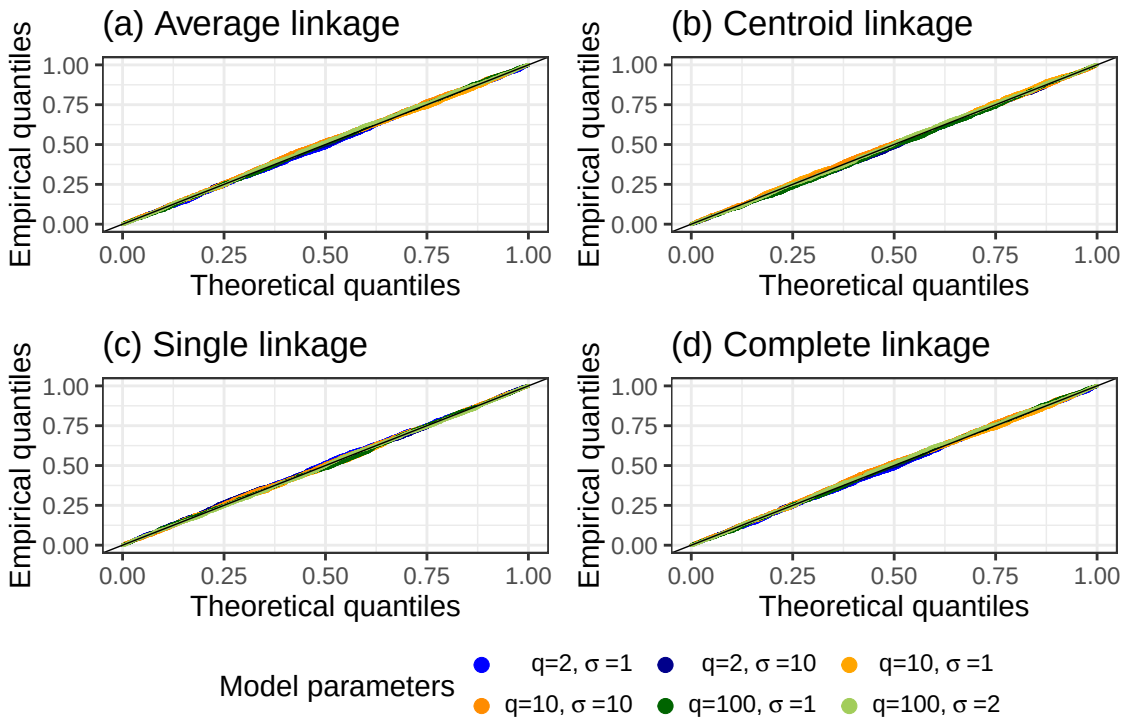

Figure 4.4: For 2000 draws from model (4.1) with $\boldsymbol{\mu} = \mathbf{0}_{n\times q}$, $n = 150$, $q \in \{2, 10, 100\}$, and $\sigma \in \{1, 2, 10\}$, QQ-plots of the p-values obtained from the test proposed in Section 4.2.1, when $\mathcal{C}$ is the map that results from cutting the dendrogram resulting from (a) average-linkage, (b) centroid-linkage, (c) single-linkage, and (d) complete-linkage clustering, to get three clusters.

there are typically several reasonable ways to divide up the observations into clusters, calling into question the idea of a "true" clustering. Thus, we may not want to use a definition of power that depends on the existence of true clusters as well as the exact recovery of these true clusters.

Thus, we consider a second notion of power: the power of the test to detect a difference in means between two groups of observations that have a difference in means. This definition of power remains relevant even when the data-driven null hypothesis is not the same as testing the null hypothesis that two true clusters have the same means, and when the notion of "true" clusters is not well-defined.

Conditional power and detection probability

We start by considering the first notion of power, i.e. the power of the test to detect a difference in means between two *true* clusters. We generate data from model (4.1) with $n = 30$, and

$$\mu_1 = \cdots = \mu_{n/3} = \begin{bmatrix} -\frac{\delta}{2} \\ 0_{q-1} \end{bmatrix}, \ \mu_{n/3+1} = \cdots = \mu_{2n/3} = \begin{bmatrix} 0_{q-1} \\ \frac{\sqrt{3}\delta}{2} \end{bmatrix}, \ \mu_{2n/3+1} = \cdots = \mu_n = \begin{bmatrix} \frac{\delta}{2} \\ 0_{q-1} \end{bmatrix}, \tag{4.28}$$

for $\delta > 0$, so that $n/3 = 10$ observations are assigned to each of three clusters, and the Euclidean distance between the means of all pairs of clusters is equal to δ. For each simulated data set, we perform hierarchical clustering with average, centroid, single, or complete linkage, cut the dendrogram to yield three clusters, then test $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$ for an arbitrary pair of clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$, with significance level $\alpha = 0.05$. For average, centroid, and single-linkage hierarchical clustering, we compute the exact p-value defined in (4.9), using the methods described in Section 4.3. For complete-linkage hierarchical clustering, we approximate the p-value using the Monte Carlo method described in Section B.1, with $N = 2000$ Monte Carlo samples. We simulate $500,000$ data sets for $\sigma = 1$, $q = 10$, and $\delta \in \{4, 4.5, 5, 5.5, 6, 6.5, 7\}$.

We now formally define what we mean by the power of the test to detect a difference in means between two true clusters. We consider the conditional probability of rejecting $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$, given that the estimated clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ exactly match two true clusters in $\{\{1, 2, \ldots, 10\}, \{11, \ldots, 20\}, \{21, \ldots, 30\}\}$:

$$\text{Conditional power} = \frac{\#\text{ data sets where we reject } H_0 \text{ and } \hat{\mathcal{C}}_k \text{ and } \hat{\mathcal{C}}_{k'} \text{ match two true clusters}}{\#\text{ data sets where } \hat{\mathcal{C}}_k \text{ and } \hat{\mathcal{C}}_{k'} \text{ match two true clusters}}. \tag{4.29}$$

We call this conditional probability the *conditional power*. Since (4.29) conditions on the event that $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ exactly match two true clusters, we may also be interested in how often this event occurs. Thus, we also consider the *detection probability*, defined to be the probability that estimated clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ exactly match two true clusters:

$$\text{Detection probability} = \frac{\#\text{ data sets where } \hat{\mathcal{C}}_k \text{ and } \hat{\mathcal{C}}_{k'} \text{ match two true clusters}}{500,000}. \tag{4.30}$$

Figure 4.5 displays the conditional power and detection probability as a function of the distance between the true clusters (δ). Observe that for all four linkages, the conditional

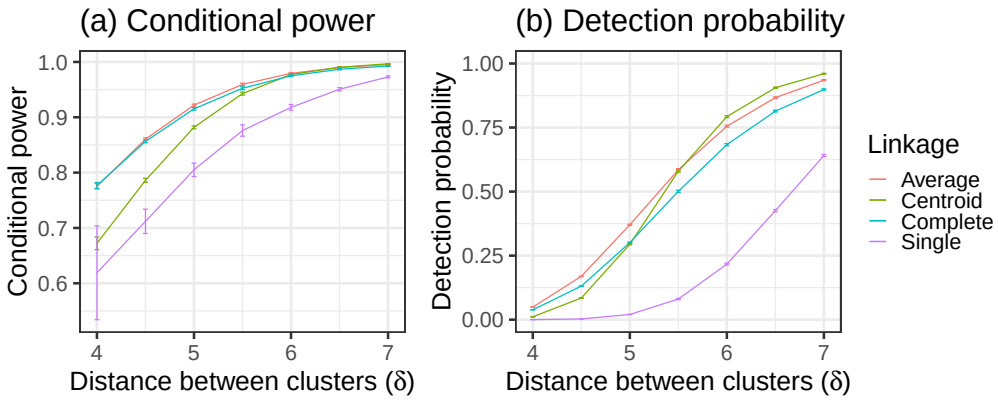

Figure 4.5: For the simulation study described in Section 4.5.2, (a) conditional power (defined in (4.29)) of the test proposed in Section 4.2 vs. the difference in means between the true clusters (δ) and (b) detection probability (defined in (4.30)) vs. the difference in means between the true clusters (δ), for hierarchical clustering with four linkages.

power and detection probability increase, as the distance between the true clusters (δ) increases. Overall, average and complete linkage have the highest conditional power, single linkage has the worst conditional power, and centroid linkage has conditional power between single linkage and average/complete linkage. Average, centroid, and complete linkage have similar detection probabilities, and all three linkages have much higher detection probability than single linkage.

Power as a function of effect size

We now consider the second notion of power, i.e. the power of the test to detect a difference in means between two groups of observations that truly have a difference in means. We generate data from model (4.1) with $n = 150$, and μ given by (4.28). Once again, for each simulated data set, we perform hierarchical clustering with average, centroid, single, or complete linkage to get three clusters, then test $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$ for an arbitrary pair of clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$, with significance level $\alpha = 0.05$, using the exact p-value for average,

centroid, and single linkage, and approximating the p-value with $N = 2000$ Monte Carlo samples for complete linkage. We simulate $10,000$ data sets for $\sigma = 1$, $q = 10$, and 17 evenly spaced values of δ in the interval $[3, 7]$.

We define the *effect size* to be the distance between the means of the two estimated clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$, scaled by the variance parameter σ:

$$\Delta = \|\bar{\mu}_{\hat{\mathcal{C}}_k} - \bar{\mu}_{\hat{\mathcal{C}}_{k'}}\|_2/\sigma, \tag{4.31}$$

and consider the power (i.e. the probability of rejecting $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$) of the test as a function of the effect size Δ. In order to smooth our estimates of power, we fit a regression spline using the `gam` function in the `R` package `mgcv`. We display the power of the test as a function of the effect size Δ when $\min\{|\hat{\mathcal{C}}_k|, |\hat{\mathcal{C}}_{k'}|\} \geq 10$ in Figure 4.6, and the power of the test as a function of the effect size Δ when $\min\{|\hat{\mathcal{C}}_k|, |\hat{\mathcal{C}}_{k'}|\} < 10$ in Appendix C.4. We stratify our results in this way because the power of the test is drastically different when $\min\{|\hat{\mathcal{C}}_k|, |\hat{\mathcal{C}}_{k'}|\}$ is small or large.

It may come as a surprise that the x-axis on Figure 4.6(c) starts at 4.5, while the x-axis on Figure 4.6(a) and Figure 4.6(d) starts at 0. This is because single-linkage hierarchical clustering has a well-known tendency to produce clusters with wildly different sizes (e.g. $|\hat{\mathcal{C}}_1| = 1, |\hat{\mathcal{C}}_2| = 1, |\hat{\mathcal{C}}_3| = 148$) unless it successfully detects the true clusters. Thus, the condition $\min\{|\hat{\mathcal{C}}_k|, |\hat{\mathcal{C}}_{k'}|\} \geq 10$ is only satisfied for single-linkage hierarchical clustering when the effect size Δ is greater than 4.5. Similarly, centroid-linkage hierarchical clustering also tends to produce clusters with wildly different sizes unless it successfully detects the true clusters, and so the x-axis on Figure 4.6(b) starts at 3 rather than 0.

Observe from Figure 4.6 that for all four linkages, the power to reject $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$ increases as the effect size Δ increases, where $\Delta = (\bar{\mu}_{\hat{\mathcal{C}}_k} - \bar{\mu}_{\hat{\mathcal{C}}_{k'}})/\sigma$. Thus, our test is more likely to detect a larger difference in means between estimated clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ than a smaller difference in means. All four linkages have similar power where the x-axes overlap.

4.6 Discussion

In this chapter, we proposed a general selective inference framework for testing the null hypothesis that there is no difference in means between two estimated clusters. We applied

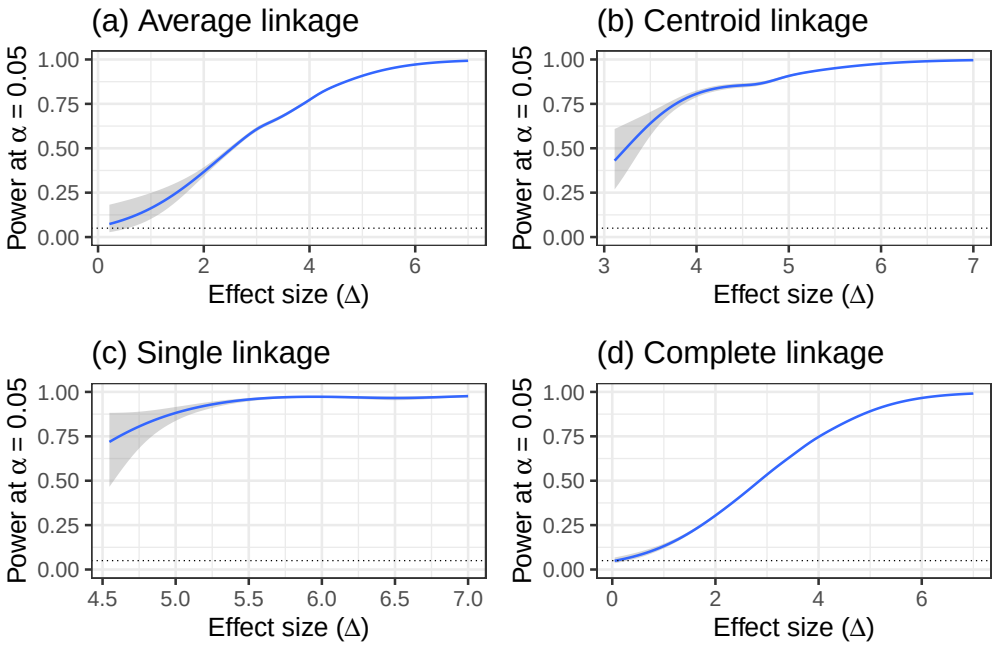

Figure 4.6: For the simulated data sets such that $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ both have at least 10 observations in them in the simulation study described in Section 4.5.2, power as a function of effect size Δ, defined in (4.31), when $\mathcal{C}$ is the function that results from cutting the dendrogram resulting from (a) average-linkage, (b) centroid-linkage, (c) single-linkage, and (d) complete-linkage hierarchical clustering, to get three clusters.

this framework to agglomerative hierarchical clustering, and showed that we could efficiently obtain an exact p-value for three commonly used linkages (single, average, and centroid), and an approximate p-value for a fourth commonly used linkage (complete), while properly accounting for the fact that the null hypothesis is a function of the data.

Throughout this chapter, we used squared Euclidean distance to define a measure of dissimilarity between two observations. The approximate p-value described in Section B.1 can be applied to clusters obtained from any clustering method, and so it could be applied to accomodate any definition of dissimilarity between two observations, and any linkage. It is relatively straightforward to modify the results in Section 4.3 to accomodate using Manhattan distance to define the dissimilarity between two observations using Manhattan distance, though the resulting testing procedure is much more computationally expensive

unless the number of features is quite small. It may also be of future interest to develop a set of results along the lines of Section 4.3 that accommodate the use of other definitions of dissimilarity between two observations, such as non-squared Euclidean distance, or to the use of other linkages, such as minimax linkage [11].

Other avenues of future work include applying our test to the problem of choosing the number of clusters in agglomerative hierarchical clustering. A basic proposal would be to start at the top of the dendrogram, and repeatedly apply our test to each split in the dendrogram until we fail to reject the null hypothesis, in combination with some sort of multiple testing correction procedure. For example, we could take advantage of a number of recently proposed procedures that control the false discovery rate for hierarchically ordered hypotheses [78, 7, 120, 42, 71, 91].

Chapter 5

DISCUSSION

5.1 Summary

In this dissertation, we propose new methodology to solve three inferential problems in single-view and multi-view clustering. In Chapter 2, we consider the setting where multiple multivariate data views are available on a common set of observations. Within the framework of an extended multi-view finite mixture model, we propose a pseudo likelihood ratio test (PLRT) of the null hypothesis that the cluster membership random variables within each data view are independent. We demonstrate through simulations that our proposed PLRT has higher power than the simple alternative approach of applying classic categorical data analysis techniques like the χ^2-test for independence or the G-test for independence to the estimated cluster memberships within each data view. We apply our proposed PLRT to clinical, metabolomic, and proteomic data sets, collected at three different time points, from the Pioneer 100 (P100) Wellness Study [90]. We found strong evidence that clusters of study participants defined with respect to a single data type (e.g. clinical) persist over time, but only weak evidence that clusters of study participants persist with respect to different data types, even at a single time point.

In Chapter 3, we consider the setting where multiple network data views are available on a common set of nodes. We extend the single-view stochastic block model to the multi-view data setting, and propose a pseudo pseudolikelihood ratio test (P^2-LRT) of the null hypothesis that the community membership random variables within each network are independent. We also adapt this methodology to tackle a related problem in another multi-view data setting, i.e. the setting where we have both a network and covariates available for a common set of nodes. Again, we demonstrate through simulation studies that our proposed P^2LRT has higher power than the simple alternative approach of applying classic categorical data analysis techniques to the estimated community memberships within each data

view.

In Chapter 4, we pivot to the simpler setting where we have a single data view (data set), and consider the scenario where we cut an agglomerative hierarchical clustering dendrogram to yield a set of estimated clusters. Under a simple matrix-variate normal model for the data set, we develop a selective inference framework for testing the null hypothesis that two estimated clusters obtained from any clustering procedure have the same mean, that properly accounts for the fact that the null hypothesis depends on the data. We then apply this framework to develop computationally efficient procedures for testing the null hypothesis that two estimated clusters obtained from hierarchical clustering with four commonly used linkages (single, average, centroid, and complete) have the same mean. We demonstrate through simulation studies that our proposed tests control the Type I error rate.

5.2 Limitations and future work

We developed the tests proposed in Chapters 2 and 3 under the assumption that the observed data views are conditionally independent given their cluster/community membership random variables. This assumption may not hold in relevant practical applications. For example, if the degree of the nodes in two network data views are correlated, even after conditioning on their community memberships, then the conditional independence assumption does not hold. We can imagine a scenario where the degrees of the nodes are correlated; consider multi-view social network data, where we might expect individuals who are popular (i.e. well-connected) in one social network to also be popular in another social network. While we showed through simulations in Chapter 3 that our proposed test was relatively robust against the conditional independence assumption, a desirable extension to our methodology would relax the assumption that the observed data views are conditionally independent given their cluster/community membership random variables. For example, we could attempt to develop tests of independence under the assumption that the observed data views are merely conditionally *uncorrelated* given their cluster/community membership random variables.

Another limitation of the tests proposed in Chapters 2 and 3 is that we develop them under the assumption that there is no missing data in any view. This is arguably an even

more problematic assumption than in the single-view data setting, because the concept of multi-view data opens up many new patterns of missingness. For example, there may be many observations that have only a single data view available, especially if one data view is more convenient or cheap to collect than the other data view. For example, one could imagine a scenario where we are interested in jointly analyzing a clinical data view and a genetic data view. Here, collecting clinical data on a patient may only require performing rapid and inexpensive laboratory tests, while collecting genetic data requires the use of genome sequencing technology. We tackle this type of missingness in the real data application in Chapter 2 by estimating the parameters specific to the lth data view ($\pi^{(l)}$ and $\theta^{(l)}$) using all of the observations that have no missing data in the lth data view, for $l = 1, 2$, and estimating the parameters joint to the two views (C) using all of the observations that have no missing data in both views. However, an interesting extension of Chapters 2 and 3 would incorporate imputation of missing values into the model and testing procedures.

A limitation of the tests proposed in Chapter 4 is that they were developed under the assumption that the observations in the data set are normally distributed. This assumption may affect the performance of the test in real data sets, where the normality assumption may not be satisfied. Future work will include simulations that evaluate the robustness of the tests against departures from the normality assumptions. Another limitation of the tests proposed in Chapter 4 is that they were developed under the assumption that the observations in the data set have covariance matrix $\sigma^2\mathbf{I}_q$, with known variance parameter σ^2. In practice, we will need to estimate σ^2 and future work will also include simulations that evaluate the performance of the tests when we replace σ^2 with an estimate. These simulation results will be especially relevant when we eventually apply the testing framework proposed in Chapter 4 to real data applications. It is relatively straightforward to adapt the tests proposed in Chapter 4 under the assumption that the observations in the data set have covariance matrix Σ, where Σ is a known positive definite matrix, although in practice, we need to estimate Σ.

BIBLIOGRAPHY

[1] Alan Agresti. *Categorical data analysis*, volume 482. John Wiley & Sons, 2003.

[2] Christopher Aicher, Abigail Z Jacobs, and Aaron Clauset. Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248, 2014.

[3] Javier A Alfaro, Ankit Sinha, Thomas Kislinger, and Paul C Boutros. Onco-proteogenomics: cancer proteomics joins forces with genomics. *Nature Methods*, 11(11):1107, 2014.

[4] Arash A Amini, Aiyou Chen, Peter J Bickel, and Elizaveta Levina. Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics*, 41(4):2097–2122, 2013.

[5] Pierre Barbillon, Sophie Donnet, Emmanuel Lazega, and Avner Bar-Hen. Stochastic block models for multiplex networks: an application to a multilevel network of researchers. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 180(1):295–314, 2017.

[6] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[7] Yoav Benjamini and Ruth Heller. False discovery rates for spatial signals. *Journal of the American Statistical Association*, 102(480):1272–1281, 2007.

[8] Julian Besag. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 24(3):179–195, 1975.

[9] Steffen Bickel and Tobias Scheffer. Multi-view clustering. In *ICDM*, volume 4, pages 19–26, 2004.

[10] Jacob Bien. The simulator: An engine to streamline simulations. *arXiv preprint arXiv:1607.00021*, 2016.

[11] Jacob Bien and Robert Tibshirani. Hierarchical clustering with prototypes via minimax linkage. *Journal of the American Statistical Association*, 106(495):1075–1084, 2011.

[12] Norbert Binkiewicz, Joshua T Vogelstein, and Karl Rohe. Covariate-assisted spectral clustering. *Biometrika*, 104(2):361–377, 2017.

[13] A Rose Brannon, Anupama Reddy, Michael Seiler, Alexandra Arreola, Dominic T Moore, Raj S Pruthi, Eric M Wallen, Matthew E Nielsen, Huiqing Liu, Katherine L Nathanson, et al. Molecular stratification of clear cell renal cell carcinoma by consensus clustering reveals distinct subtypes and survival patterns. *Genes & cancer*, 1(2):152–163, 2010.

[14] Frederick Campbell. *Statistical Machine Learning Methodology and Inference for Structured Variable Selection*. PhD thesis, Rice University, 2018.

[15] Cancer Genome Atlas Research Network. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061–1068, 2008.

[16] Purvasha Chakravarti, Sivaraman Balakrishnan, and Larry Wasserman. Gaussian mixture clustering using relative tests of fit. *arXiv preprint arXiv:1910.02566*, 2019.

[17] Hanfeng Chen, Jiahua Chen, and John D Kalbfleisch. A modified likelihood ratio test for homogeneity in finite mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):19–29, 2001.

[18] Jiahua Chen, Pengfei Li, et al. Hypothesis test for normal mixture models: The EM approach. *The Annals of Statistics*, 37(5A):2523–2542, 2009.

[19] Jiahua Chen, Pengfei Li, and Yuejiao Fu. Inference on the order of a normal mixture. *Journal of the American Statistical Association*, 107(499):1096–1105, 2012.

[20] Shuxiao Chen and Jacob Bien. Valid inference corrected for outlier removal. *Journal of Computational and Graphical Statistics*, pages 1–12, 2019.

[21] Yong Chen and Kung-Yee Liang. On the asymptotic behaviour of the pseudolikelihood ratio test statistic with boundary problems. *Biometrika*, 97(3):603–620, 2010.

[22] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Systems*, pages 2292–2300, 2013.

[23] Silvia D'Angelo, Thomas Brendan Murphy, and Marco Alfò. Latent space modelling of multidimensional networks with application to the exchange of votes in eurovision song contest. *The Annals of Applied Statistics*, 13(2):900–930, 2019.

[24] Jishnu Das and Haiyuan Yu. High-quality interactomes (HINT). http://hint.yulab.org, 2012. Accessed: 01-22-19.

[25] Jishnu Das and Haiyuan Yu. HINT: High-quality interactomes and their applications in understanding human disease. *BMC Systems Biology*, 6(1):92, 2012.

[26] Javier De Las Rivas and Celia Fontanillo. Protein–protein interactions essentials: key concepts to building and analyzing interactome networks. *PLoS Computational Biology*, 6(6):e1000807, 2010.

[27] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–38, 1977.

[28] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Proceedings of the Hungarian Academy of Sciences*, pages 17–61, 1960.

[29] William Fithian, Dennis Sun, and Jonathan Taylor. Optimal inference after model selection. *arXiv preprint arXiv:1410.2597*, 2014.

[30] Bailey K Fosdick and Peter D Hoff. Testing and modeling dependencies between a network and nodal attributes. *Journal of the American Statistical Association*, 110(511):1047–1056, 2015.

[31] Joel Franklin and Jens Lorenz. On the scaling of multidimensional matrices. *Linear Algebra and its Applications*, 114/115:717–735, 1989.

[32] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1 of *Springer Series in Statistics*. Springer-Verlag New York, 2001.

[33] Evelina Gabasova, John Reid, and Lorenz Wernisch. Clusternomics: Integrative context-dependent clustering for heterogeneous datasets. *PLoS Computational Biology*, 13(10):e1005781, 2017.

[34] Lucy L Gao, Jacob Bien, and Daniela Witten. Are clusterings of multiple data views independent? *Biostatistics*, (forthcoming), 02 2019.

[35] Isabella Gollini and Thomas Brendan Murphy. Joint modeling of multiple network views. *Journal of Computational and Graphical Statistics*, 25(1):246–265, 2016.

[36] Gail Gong and Francisco J. Samaniego. Pseudo maximum likelihood estimation: theory and applications. *The Annals of Statistics*, 9(4):861–869, 1981.

[37] Jens K Habermann, Ulrike Paulsen, Uwe J Roblick, Madhvi B Upender, Lisa M McShane, Edward L Korn, Danny Wangsa, Stefan Krüger, Michael Duchrow, Hans-Peter Bruch, et al. Stage-specific alterations of the genome, transcriptome, and proteome during colorectal carcinogenesis. *Genes, Chromosomes and Cancer*, 46(1):10–26, 2007.

[38] Qiuyi Han, Kevin Xu, and Edoardo Airoldi. Consistent estimation of dynamic and multi-layer block models. In *Proceedings of the 32nd International Conference on Machine Learning - Volume 37*, pages 1511–1520, 2015.

[39] John A Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc., 1975.

[40] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[41] Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu. *impute: Imputation for microarray data,*, 2017. R package version 1.50.1.

[42] Ruth Heller, Elisabetta Manduchi, Gregory R Grant, and Warren J Ewens. A flexible two-stage procedure for identifying gene sets that are differentially expressed. *Bioinformatics*, 25(8):1019–1025, 2009.

[43] Toby Dylan Hocking, Armand Joulin, Francis Bach, and Jean-Philippe Vert. Clusterpath an algorithm for clustering using convex fusion penalties. 2011.

[44] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

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

[46] Paul W Holland and Samuel Leinhardt. An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50, 1981.

[47] Hanwen Huang, Yufeng Liu, Ming Yuan, and JS Marron. Statistical significance of clustering using soft thresholding. *Journal of Computational and Graphical Statistics*, 24(4):975–993, 2015.

[48] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[49] Sangwon Hyun, Max G'Sell, Ryan J Tibshirani, et al. Exact post-selection inference for the generalized lasso path. *Electronic Journal of Statistics*, 12(1):1053–1097, 2018.

[50] Sangwon Hyun, Kevin Lin, Max G'Sell, and Ryan J Tibshirani. Post-selection inference for changepoint detection algorithms with application to copy number variation data. *arXiv preprint arXiv:1812.03644*, 2018.

[51] Sean Jewell, Paul Fearnhead, and Daniela Witten. Testing for a change in mean after changepoint detection. *arXiv preprint arXiv:1910.04291*, 2019.

[52] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

[53] Hiroyuki Kasahara and Katsumi Shimotsu. Testing the number of components in normal mixture regression models. *Journal of the American Statistical Association*, 110(512):1632–1645, 2015.

[54] Bomin Kim, Kevin H Lee, Lingzhou Xue, and Xiaoyue Niu. A review of dynamic network models with latent variables. *Statistics Surveys*, 12:105–135, 2018.

[55] Patrick K Kimes, Yufeng Liu, David Neil Hayes, and James Stephen Marron. Statistical significance for hierarchical clustering. *Biometrics*, 73(3):811–821, 2017.

[56] Paul Kirk, Jim E. Griffin, Richard S. Savage, Zoubin Ghahramani, and David L. Wild. Bayesian correlated clustering to integrate multiple datasets. *Bioinformatics*, 28(24):3290–3297, 2012.

[57] Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

[58] Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1413–1421, 2011.

[59] Godfrey N Lance and William Thomas Williams. A general theory of classificatory sorting strategies: 1. hierarchical systems. *The Computer Journal*, 9(4):373–380, 1967.

[60] Can M Le and Elizaveta Levina. Estimating the number of communities in networks by spectral methods. *arXiv preprint*, 2015. arXiv:1507.00827.

[61] Jason D Lee, Dennis L Sun, Yuekai Sun, Jonathan E Taylor, et al. Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, 44(3):907–927, 2016.

[62] Jason D Lee, Yuekai Sun, and Jonathan E Taylor. Evaluating the statistical significance of biclusters. In *Advances in neural information processing systems*, pages 1324–1332, 2015.

[63] Brian D Lehmann, Joshua A Bauer, Xi Chen, Melinda E Sanders, A Bapsi Chakravarthy, Yu Shyr, and Jennifer A Pietenpol. Identification of human triple-negative breast cancer subtypes and preclinical models for selection of targeted therapies. *The Journal of clinical investigation*, 121(7):2750–2767, 2011.

[64] Pengfei Li and Jiahua Chen. Testing the order of a finite mixture. *Journal of the American Statistical Association*, 105(491):1084–1092, 2010.

[65] Yifeng Li, Fang-Xiang Wu, and Alioune Ngom. A review on machine learning principles for multi-view biological data integration. *Briefings in Bioinformatics*, 19(2):325–340, 2018.

[66] Kung-Yee Liang and Steven G Self. On the asymptotic behaviour of the pseudolikelihood ratio test statistic. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 785–796, 1996.

[67] Fredrik Lindsten, Henrik Ohlsson, and Lennart Ljung. *Just relax and come clustering!: A convexification of k-means clustering*. Linköping University Electronic Press, 2011.

[68] Yufeng Liu, David Neil Hayes, Andrew Nobel, and James Stephen Marron. Statistical significance of clustering for high-dimension, low–sample size data. *Journal of the American Statistical Association*, 103(483):1281–1293, 2008.

[69] Eric F. Lock and David B. Dunson. Bayesian consensus clustering. *Bioinformatics*, 29(20):2610–2616, 2013.

[70] Joshua R Loftus and Jonathan E Taylor. Selective inference in regression models with groups of variables. *arXiv preprint arXiv:1511.01478*, 2015.

[71] Gavin Lynch and Wenge Guo. On procedures controlling the FDR for testing hierarchically ordered hypotheses. *arXiv preprint arXiv:1612.04467*, 2016.

[72] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[73] Ranjan Maitra, Volodymyr Melnykov, and Soumendra N Lahiri. Bootstrapping for significance of compact clusters in multidimensional datasets. *Journal of the American Statistical Association*, 107(497):378–392, 2012.

[74] Catherine Matias and Stéphane Robin. Modeling heterogeneity in random graphs through latent space models: a selective review. *ESAIM: Proceedings and Surveys*, 47:55–74, 2014.

[75] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[76] Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2000.

[77] Geoffrey J McLachlan and Suren Rathnayake. On the number of components in a gaussian mixture model. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(5):341–355, 2014.

[78] Devan V Mehrotra and Joseph F Heyse. Use of the false discovery rate for evaluating clinical safety data. *Statistical methods in medical research*, 13(3):227–238, 2004.

[79] Marina Meilă. Comparing clusterings – an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.

[80] Marina Meila and Jianbo Shi. Learning segmentation by random walks. In *Advances in Neural Information Processing Systems*, pages 873–879, 2001.

[81] Boris Mirkin. Choosing the number of clusters. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):252–260, 2011.

[82] Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

[83] Mark EJ Newman and Aaron Clauset. Structure and inference in annotated networks. *Nature Communications*, 7:11863, 2016.

[84] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2002.

[85] Subhadeep Paul and Yuguo Chen. Consistent community detection in multi-relational data through restricted multi-layer stochastic blockmodel. *Electronic Journal of Statistics*, 10(2):3807–3870, 2016.

[86] Subhadeep Paul and Yuguo Chen. A random effects stochastic block model for joint community detection in multiple networks with applications to neuroimaging. *arXiv preprint*, 2018. arXiv:1805.02292.

[87] Leto Peel, Daniel B Larremore, and Aaron Clauset. The ground truth about metadata and community detection in networks. *Science Advances*, 3(5):e1602548, 2017.

[88] Tiago P Peixoto. Inferring the mesoscale structure of layered, edge-valued, and time-varying networks. *Physical Review E*, 92(4):042807, 2015.

[89] Charles M Perou, Therese Sørlie, Michael B Eisen, Matt Van De Rijn, Stefanie S Jeffrey, Christian A Rees, Jonathan R Pollack, Douglas T Ross, Hilde Johnsen, Lars A Akslen, et al. Molecular portraits of human breast tumours. *nature*, 406(6797):747–752, 2000.

[90] Nathan D. Price, Andrew T. Magis, John C. Earls, Gustavo Glusman, Roie Levy, Christopher Lausted, Daniel T. McDonald, Ulrike Kusebauch, Christopher L. Moss, Yong Zhou, Shizhen Qin, Robert L. Moritz, Kristin Brogaard, Gilbert S. Omenn, Jennifer C. Lovejoy, and Leroy Hood. A wellness study of 108 individuals using personal, dense, dynamic data clouds. *Nature Biotechnology*, 35(8):747–756, 2017.

[91] Aaditya Ramdas, Jianbo Chen, Martin J Wainwright, and Michael I Jordan. Dagger: A sequential algorithm for fdr control on dags. *arXiv preprint arXiv:1709.10250*, 2017.

[92] Nancy Reid. The roles of conditioning in inference. *Statistical Science*, 10(2):138–157, 1995.

[93] R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag Berlin Heidelberg, 1998.

[94] Simon Rogers, Mark Girolami, Walter Kolch, Katrina M. Waters, Tao Liu, Brian Thrall, and H. Steven Wiley. Investigating the correspondence between transcriptomic and proteomic expression profiles using coupled cluster models. *Bioinformatics*, 24(24):2894–2900, 2008.

[95] Michael Salter-Townshend and Tyler H McCormick. Latent space models for multiview network data. *The Annals of Applied Statistics*, 11(3):1217, 2017.

[96] Luca Scrucca, Michael Fop, T. Brendan Murphy, and Adrian E Raftery. mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1):289, 2016.

[97] Steven G. Self and Kung-Yee Liang. Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. *Journal of the American Statistical Association*, 82(398):605–610, 1987.

[98] Ronglai Shen, Adam B. Olshen, and Marc Ladanyi. Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, 25(22):2906–2912, 2009.

[99] Hidetoshi Shimodaira and Yoshikazu Terada. Selective inference for testing trees and edges in phylogenetics. *Frontiers in Ecology and Evolution*, 7:174, 2019.

[100] Therese Sørlie, Charles M Perou, Robert Tibshirani, Turid Aas, Stephanie Geisler, Hilde Johnsen, Trevor Hastie, Michael B Eisen, Matt Van De Rijn, Stefanie S Jeffrey, et al. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences*, 98(19):10869–10874, 2001.

[101] Natalie Stanley, Thomas Bonacci, Roland Kwitt, Marc Niethammer, and Peter J

Mucha. Stochastic block models with multiple continuous attributes. *Applied Network Science*, 4(1):1–22, 2019.

[102] Natalie Stanley, Saray Shai, Dane Taylor, and Peter J Mucha. Clustering network layers with the strata multilayer stochastic block model. *IEEE Transactions on Network Science and Engineering*, 3(2):95–105, 2016.

[103] Mahito Sugiyama, Hiroyuki Nakahara, and Koji Tsuda. Tensor balancing on statistical manifold. *arXiv preprint arXiv:1702.08142*, 2017.

[104] Shiliang Sun. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7-8):2031–2038, 2013.

[105] Ryota Suzuki and Hidetoshi Shimodaira. Pvclust: an r package for assessing the uncertainty in hierarchical clustering. *Bioinformatics*, 22(12):1540–1542, 2006.

[106] Kean Ming Tan and Daniela Witten. Statistical properties of convex clustering. *Electronic journal of statistics*, 9(2):2324, 2015.

[107] Jonathan Taylor and Robert J Tibshirani. Statistical learning and selective inference. *Proceedings of the National Academy of Sciences*, 112(25):7629–7634, 2015.

[108] The Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61, 2012.

[109] Amanda L Traud, Eric D Kelsic, Peter J Mucha, and Mason A Porter. Comparing community structure to characteristics in online collegiate social networks. *SIAM Review*, 53(3):526–543, 2011.

[110] Sooryanarayana Varambally, Jianjun Yu, Bharathi Laxman, Daniel R Rhodes, Rohit Mehra, Scott A Tomlins, Rajal B Shah, Uma Chandran, Federico A Monzon, Michael J Becich, et al. Integrative genomic and proteomic analysis of prostate cancer reveals signatures of metastatic progression. *Cancer Cell*, 8(5):393–406, 2005.

[111] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Yonina C. Eldar and Gitta Kutyniok, editors, *Compressed Sensing: Theory and Applications*, chapter 5, pages 210–268. Cambridge University Press, Cambridge, 2012.

[112] Arvind K Virmani, Jeffrey A Tsou, Kimberly D Siegmund, Linda YC Shen, Tiffany I Long, Peter W Laird, Adi F Gazdar, and Ite A Laird-Offringa. Hierarchical clustering of lung cancer cell lines using DNA methylation markers. *Cancer Epidemiology and Prevention Biomarkers*, 11(3):291–297, 2002.

[113] Michael Vogt and Matthias Schmid. Clustering with statistical error control. *Scandinavian Journal of Statistics*, (forthcoming).

[114] Yuchung J Wang and George Y Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987.

[115] Junhao Xiong, Cencheng Shen, Jesüs Arroyo, and Joshua T Vogelstein. Graph independence testing. *arXiv preprint*, 2019. arXiv:1906.03661.

[116] Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, 2015.

[117] Bowei Yan and Purnamrita Sarkar. Covariate regularized community detection in sparse graphs. *Journal of the American Statistical Association*, 0(ja):1–29, 2020.

[118] Fan Yang, Rina Foygel Barber, Prateek Jain, and John Lafferty. Selective inference for group-sparse linear models. In *Advances in Neural Information Processing Systems*, pages 2469–2477, 2016.

[119] Jaewon Yang, Julian McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *Proceedings of the IEEE 13th International Conference on Data Mining*, pages 1151–1156, 2013.

[120] Daniel Yekutieli. Hierarchical false discovery rate–controlling methodology. *Journal of the American Statistical Association*, 103(481):309–316, 2008.

[121] Yuan Zhang, Elizaveta Levina, and Ji Zhu. Community detection in networks with node features. *Electronic Journal of Statistics*, 10(2):3153–3178, 2016.

Appendix A

SUPPLEMENTARY MATERIALS FOR CHAPTER 2

A.1 Proof of Proposition 1

Suppose $\pi^{(1)} \in \Delta_+^{K^{(1)}}$ and $\pi^{(2)} \in \Delta_+^{K^{(2)}}$, where $\Delta_+^K \equiv \{s \in \mathbb{R}^K : s_k > 0, \sum_{k=1}^{K} s_k = 1\}$. Let $\mathcal{C}_{\pi^{(1)},\pi^{(2)}} = \{C \in \mathbb{R}^{K^{(1)} \times K^{(2)}} : C_{kk'} \geq 0,\ C\pi^{(2)} = 1_{K^{(1)}},\ C^T\pi^{(1)} = 1_{K^{(2)}}\}$. First, we show that

$$\left\{\Pi \in \Delta^{K^{(1)} \times K^{(2)}} : \Pi 1_{K^{(2)}} = \pi^{(1)},\ \Pi^T 1_{K^{(1)}} = \pi^{(2)}\right\} \subseteq \left\{\mathrm{diag}(\pi^{(1)}) C \mathrm{diag}(\pi^{(2)}) : C \in \mathcal{C}_{\pi^{(1)},\pi^{(2)}}\right\}.$$

Suppose $\Pi \in \Delta^{K^{(1)} \times K^{(2)}} = \{S \in \mathbb{R}^{K^{(1)} \times K^{(2)}} : S_{kk'} \geq 0, \sum_{k=1}^{K^{(1)}} \sum_{k'=1}^{K^{(2)}} S_{kk'} = 1\}$ such that $\Pi \cdot 1_{K^{(2)}} = \pi^{(1)}$ and $\Pi^T \cdot 1_{K^{(1)}}$. Define $C \in \mathbb{R}_+^{K^{(1)} \times K^{(2)}}$ by $C_{kk'} = \frac{\Pi_{kk'}}{\pi_k^{(1)} \pi_{k'}^{(2)}}$. The denominator is nonzero because $\pi^{(1)} \in \Delta_+^{K^{(1)}}$ and $\pi^{(2)} \in \Delta_+^{K^{(2)}}$. Further, $C_{kk'} \geq 0$ and $\Pi = \mathrm{diag}(\pi^{(1)}) C \mathrm{diag}(\pi^{(2)})$. Now,

$$C\pi^{(2)} = \begin{bmatrix} \sum_{k'} C_{1k'} \pi_{k'}^{(2)} \\ \vdots \\ \sum_{k'} C_{K^{(1)}k'} \pi_{k'}^{(2)} \end{bmatrix} = \begin{bmatrix} \sum_{k'} \frac{\Pi_{1k'}}{\pi_1^{(1)}} \\ \vdots \\ \sum_{k'} \frac{\Pi_{K^{(1)}k'}}{\pi_{K^{(1)}}^{(1)}} \end{bmatrix} = 1_{K^{(1)}}$$

since $\Pi 1_{K^{(2)}} = \pi^{(1)}$ implies that $\sum_{k'} \Pi_{kk'} = \pi_k^{(1)}$. Similarly, we can show $C^T\pi^{(1)} = 1_{K^{(2)}}$.

Next, we show that

$$\left\{\Pi \in \Delta^{K^{(1)} \times K^{(2)}} : \Pi 1_{K^{(2)}} = \pi^{(1)},\ \Pi^T 1_{K^{(1)}} = \pi^{(2)}\right\} \supseteq \left\{\mathrm{diag}(\pi^{(1)}) C \mathrm{diag}(\pi^{(2)}) : C \in \mathcal{C}_{\pi^{(1)},\pi^{(2)}}\right\}.$$

Suppose $C \in \mathbb{R}^{K^{(1)} \times K^{(2)}}$ such that $C_{kk'} \geq 0$, $C\pi^{(2)} = 1_{K^{(1)}}$ and $C^T\pi^{(1)} = 1_{K^{(2)}}$. Define

$$\Pi = \mathrm{diag}(\pi^{(1)}) C \mathrm{diag}(\pi^{(2)}).$$

Then $\Pi_{kk'} = \pi_k^{(1)} C_{kk'} \pi_{k'}^{(2)}$. Since $\pi_k^{(1)} > 0$, $C_{kk'} \geq 0$ and $\pi_{k'}^{(2)} > 0$, it follows that $\Pi_{kk'} \geq 0$.

Now,

$$\begin{aligned}\Pi 1_{K^{(2)}} &= \mathrm{diag}(\pi^{(1)}) C \mathrm{diag}(\pi^{(2)}) 1_{K^{(2)}} \\ &= \mathrm{diag}(\pi^{(1)}) C \pi^{(2)} \\ &= \mathrm{diag}(\pi^{(1)}) 1_{K^{(1)}} = \pi^{(1)}.\end{aligned}$$

Similarly, we can show $\Pi^T 1_{K^{(1)}} = \pi^{(2)}$. Further, since $\pi^{(1)} \in \Delta_+^{K^{(1)}}$, $\sum_{kk'} \Pi_{kk'} = 1_{K^{(1)}}^T \Pi 1_{K^{(2)}} = 1_{K^{(1)}}^T \pi^{(1)} = 1$, so $\Pi \in \Delta^{K^{(1)} \times K^{(2)}}$.

Hence, we have proved Proposition 1.

A.2 Proof of Proposition 4

Let $\sigma^2 > 0$. Suppose throughout that we fit the model (2.1)–(2.2), for $\phi^{(1)}(\cdot;\theta)$ the density of a $N_{p_1}(\theta, \sigma^2 I_{p_1})$ random variable and $\phi^{(2)}(\cdot;\theta)$ the density of a $N_{p_2}(\theta, \sigma^2 I_{p_2})$ random variable. This amounts to applying Gaussian mixture model-based clustering with common covariance matrix $\sigma^2 I$ to each view.

Let $\hat{\pi}^{(1)}$, $\hat{\pi}^{(2)}$, $\hat{\theta}^{(1)}$, and $\hat{\theta}^{(2)}$ denote the maximizers of (2.7), and let

$$\hat{r}_i^{(1)} = \frac{\hat{\phi}_i^{(1)}}{1^T \hat{\phi}_i^{(1)}}, \quad \hat{r}_i^{(2)} = \frac{\hat{\phi}_i^{(2)}}{1^T \hat{\phi}_i^{(2)}}, \tag{A.1}$$

where $\hat{\phi}^{(1)}$ and $\hat{\phi}^{(2)}$ are defined in (2.9).

Since Gaussian mixture model-based clustering with common covariance matrix $\nu^2 I$ converges to k-means clustering in each view as $\nu^2 \to 0$ (see Section 14.3.7 in [32] for details), as $\sigma^2 \to 0$,

$$\hat{r}_{ik}^{(1)} \to \mathbb{1}\{\tilde{M}_i^{(1)} = k\}, \quad \hat{r}_{ik'}^{(2)} \to \mathbb{1}\{\tilde{M}_i^{(2)} = k'\}, \quad \hat{\pi}_k^{(1)} \to \frac{\tilde{N}_{k\cdot}}{n}, \quad \hat{\pi}_{k'}^{(2)} \to \frac{\tilde{N}_{\cdot k'}}{n}, \tag{A.2}$$

where $\tilde{M}_i^{(1)}$ and $\tilde{M}_i^{(2)}$ are the estimated k-means cluster assignments of the ith observation in each view, $\tilde{N}_{k\cdot} = \sum_{k'} \tilde{N}_{kk'}$, and $\tilde{N}_{\cdot k'} = \sum_{k} \tilde{N}_{kk'}$, for $\tilde{N}_{kk'} = |\{i \in \{1,\ldots,n\} : \tilde{M}_i^{(1)} = k, \tilde{M}_i^{(2)} = k'\}|$.

We now rewrite (2.8) as

$$\begin{aligned}
\hat{C} &= \underset{C\in\mathcal{C}_{\hat{\pi}^{(1)},\hat{\pi}^{(2)}}}{\arg\min} \left[-\sum_{i=1}^{n} \log \left(\sum_{k=1}^{K^{(1)}} \sum_{k'=1}^{K^{(2)}} \hat{\pi}_k^{(1)} C_{kk'} \hat{\pi}_{k'}^{(2)} \hat{\phi}_{ik}^{(1)} \hat{\phi}_{ik'}^{(2)} \right) \right] \\
&= \underset{C\in\mathcal{C}_{\hat{\pi}^{(1)},\hat{\pi}^{(2)}}}{\arg\min} \left[-\sum_{i=1}^{n} \log \left(\sum_{k=1}^{K^{(1)}} \sum_{k'=1}^{K^{(2)}} \hat{\pi}_k^{(1)} C_{kk'} \hat{\pi}_{k'}^{(2)} \hat{\phi}_{ik}^{(1)} \hat{\phi}_{ik'}^{(2)} \right) - \sum_{i=1}^{n} \log \left(1^T \hat{\phi}_i^{(1)} 1^T \hat{\phi}_i^{(2)} \right) \right] \\
&= \underset{C\in\mathcal{C}_{\hat{\pi}^{(1)},\hat{\pi}^{(2)}}}{\arg\min} \left[-\sum_{i=1}^{n} \log \left(\sum_{k=1}^{K^{(1)}} \sum_{k'=1}^{K^{(2)}} \hat{\pi}_k^{(1)} C_{kk'} \hat{\pi}_{k'}^{(2)} \hat{r}_{ik}^{(1)} \hat{r}_{ik'}^{(2)} \right) \right],
\end{aligned}$$

where the second equality holds because the quantities $\hat{\phi}_i^{(1)}$ and $\hat{\phi}_i^{(2)}$ defined in (2.9) do not depend on C, and the third equality follows from the definition of $\hat{r}_i^{(1)}$ and $\hat{r}_i^{(2)}$ in (A.1). It follows that

$$\hat{C} = \underset{C\in\mathcal{C}_{\hat{\pi}^{(1)},\hat{\pi}^{(2)}}}{\arg\min} \; g^{\sigma}(C), \tag{A.3}$$

where

$$g^{\sigma}(C) \equiv -\sum_{i=1}^{n} \log \left(\sum_{k=1}^{K^{(1)}} \sum_{k'=1}^{K^{(2)}} \hat{\pi}_k^{(1)} C_{kk'} \hat{\pi}_{k'}^{(2)} \hat{r}_{ik}^{(1)} \hat{r}_{ik'}^{(2)} \right).$$

By (A.2), as $\sigma^2 \to 0$, g^{σ} converges pointwise to g, where

$$\begin{aligned}
g(C) &\equiv -\sum_{i=1}^{n} \log \left(\sum_{k=1}^{K^{(1)}} \sum_{k'=1}^{K^{(2)}} \frac{\tilde{N}_{k.}}{n} C_{kk'} \frac{\tilde{N}_{.k'}}{n} \mathbb{1}\{\tilde{M}_i^{(1)} = k\} \mathbb{1}\{\tilde{M}_i^{(2)} = k'\} \right) \\
&= -\sum_{i=1}^{n} \sum_{k=1}^{K^{(1)}} \sum_{k'=1}^{K^{(2)}} \mathbb{1}\{\tilde{M}_i^{(1)} = k\} \mathbb{1}\{\tilde{M}_i^{(2)} = k'\} \log \left(\frac{\tilde{N}_{k.}}{n} C_{kk'} \frac{\tilde{N}_{.k'}}{n} \right) \\
&= -\sum_{k=1}^{K^{(1)}} \sum_{k=1}^{K^{(2)}} \tilde{N}_{kk'} \log \left(\frac{\tilde{N}_{k.}}{n} C_{kk'} \frac{\tilde{N}_{.k'}}{n} \right).
\end{aligned}$$

Applying the method of Lagrange multipliers, we find that $\tilde{C} \equiv \underset{C\in\mathcal{C}_{\hat{\pi}^{(1)},\hat{\pi}^{(2)}}}{\arg\min} \; g(C)$ satisfies

$$\tilde{C}_{kk'} = \frac{n\tilde{N}_{kk'}}{\tilde{N}_{k.}\tilde{N}_{.k'}}. \tag{A.4}$$

By Exercise 7.23(c) in [93], $\{g^{\sigma}(C)\}_{\{\sigma>0\}}$ is essentially bounded. By Theorem 7.17 in [93], the epigraphical limit of g^{σ} is g. Finally, g^{σ} and g are continuous and proper. Hence, by

Theorem 7.33 in [93],

$$\tilde{C} = \underset{C \in \mathcal{C}_{\hat{\pi}^{(1)}, \hat{\pi}^{(2)}}}{\arg\min} \; g(C) = \lim_{\sigma \to 0} \underset{C \in \mathcal{C}_{\hat{\pi}^{(1)}, \hat{\pi}^{(2)}}}{\arg\min} \; g^{\sigma}(C). \tag{A.5}$$

By (A.3), (A.4) and (A.5), as $\sigma^2 \to 0$ we have

$$\hat{C}_{kk'} \to \tilde{C}_{kk'} = \frac{n\tilde{N}_{kk'}}{\tilde{N}_{k\cdot}\tilde{N}_{\cdot k'}}. \tag{A.6}$$

By (A.2) and (A.6), and the definition of $\hat{\Pi}$ in Algorithm 1,

$$\hat{\Pi}_{kk'} \to \frac{\tilde{N}_{kk'}}{n}. \tag{A.7}$$

Applying (A.2) and (A.7) to (2.18) yields the result.

A.3 Exponentiated gradient descent for solving (2.8)

After a transformation of the optimization problem, (2.8) can be efficiently solved using exponentiated gradient descent [57], a first-order method specially designed for optimization over the simplex. This is a form of mirror descent, with provable convergence results [6]. While there is no analytic solution for the update performed at each iteration of the exponentiated gradient descent, each update can be performed by applying the Sinkhorn-Knopp algorithm, a matrix balancing algorithm with provable convergence results and linear convergence rates [31].

Define

$$\ell(\theta^{(1)}, \theta^{(2)}, \Pi) = \sum_{i=1}^{n} \log f(X_i^{(1)}, X_i^{(2)}; \theta^{(1)}, \theta^{(2)}, \Pi)$$

where $f(\cdot, \cdot; \theta^{(1)}, \theta^{(2)}, \Pi)$ is defined in (2.3). Consider the following optimization problem:

$$\begin{aligned} \underset{\Pi}{\text{minimize}} \quad & -\ell(\hat{\theta}^{(1)}, \hat{\theta}^{(2)}, \Pi) \\ \text{subject to} \quad & \Pi 1_{K^{(2)}} = \hat{\pi}^{(1)} \\ & \Pi^T 1_{K^{(1)}} = \hat{\pi}^{(2)} \\ & \Pi_{kk'} \geq 0. \end{aligned} \tag{A.8}$$

By Proposition 1, we can equivalently write (A.8) as follows:

$$\begin{aligned} \underset{C}{\text{minimize}} \quad & -\ell(\hat{\theta}^{(1)}, \hat{\theta}^{(2)}, \hat{\pi}^{(1)}, \hat{\pi}^{(2)}, C) \\ \text{subject to} \quad & \text{diag}(\hat{\pi}^{(1)}) C \text{diag}(\hat{\pi}^{(2)}) 1_{K^{(2)}} = \hat{\pi}^{(1)} \\ & \text{diag}(\hat{\pi}^{(2)}) C^T \text{diag}(\hat{\pi}^{(1)}) 1_{K^{(1)}} = \hat{\pi}^{(2)} \\ & C_{kk'} \geq 0 \end{aligned}$$

which is equivalent to

$$\begin{aligned} \underset{C}{\text{minimize}} \quad & -\ell(\hat{\theta}^{(1)}, \hat{\theta}^{(2)}, \hat{\pi}^{(1)}, \hat{\pi}^{(2)}, C) \\ \text{subject to} \quad & \text{diag}(\hat{\pi}^{(1)}) C \hat{\pi}^{(2)} = \hat{\pi}^{(1)} \\ & \text{diag}(\hat{\pi}^{(2)}) C^T \hat{\pi}^{(1)} = \hat{\pi}^{(2)} \\ & C_{kk'} \geq 0 \end{aligned}$$

which is in turn equivalent to

$$\begin{aligned} \underset{C}{\text{minimize}} \quad & -l(\hat{\theta}^{(1)}, \hat{\theta}^{(2)}, \hat{\pi}^{(1)}, \hat{\pi}^{(2)}, C) \\ \text{subject to} \quad & C \hat{\pi}^{(2)} = 1_{K^{(1)}} \\ & C^T \hat{\pi}^{(1)} = 1_{K^{(2)}} \\ & C_{kk'} \geq 0, \end{aligned}$$

which is (2.8), the optimization problem we must solve to estimate $\hat{C}$. Hence, to find $\hat{C}$, we can solve (A.8); let $\hat{\Pi}$ be the minimizer of (A.8). Then, $\hat{C}$ can be found by

$$\hat{C}_{kk'} = \frac{\hat{\Pi}_{kk'}}{\hat{\pi}_k^{(1)} \hat{\pi}_{k'}^{(2)}}. \tag{A.9}$$

The motivation for this transformation of (2.8) is that the transformed problem (A.8) can be efficiently solved using an algorithm described in [22].

We will describe the exponentiated gradient algorithm to solve the general problem

$$\begin{aligned} \underset{\Pi}{\text{minimize}} \quad & g(\Pi) \\ \text{subject to} \quad & \sum_{k'} \Pi_{kk'} = \hat{\pi}_k^{(1)} \\ & \sum_{k} \Pi_{kk'} = \hat{\pi}_{k'}^{(2)} \\ & \Pi_{kk'} \geq 0 \end{aligned} \quad . \tag{A.10}$$

To solve (A.10), we apply the update

$$\hat{\Pi}^{t+1} = \begin{cases} \underset{\Pi}{\arg\min} & g(\hat{\Pi}^t) + \langle \nabla g(\hat{\Pi}^t), \Pi - \hat{\Pi}^t \rangle + \dfrac{1}{s} \displaystyle\sum_{kk'} \Pi_{kk'} \log(\Pi_{kk'}/\hat{\Pi}^t_{kk'}) \\ \text{subject to} & \displaystyle\sum_{k'} \Pi_{kk'} = \hat{\pi}^{(1)}_k \\ & \displaystyle\sum_{k} \Pi_{kk'} = \hat{\pi}^{(2)}_{k'} \\ & \Pi_{kk'} \geq 0. \end{cases} \tag{A.11}$$

This is similar to the proximal gradient method, but instead of $\|\Pi - \hat{\Pi}^t\|_F^2/(2s)$ we use the Bregman divergence,

$$\frac{1}{s} \sum_{kk'} \Pi_{kk'} \log(\Pi_{kk'}/\hat{\Pi}^t_{kk'}).$$

An advantage of this choice is that the positivity constraint is automatically enforced. For more on this, see [6]. The optimality conditions for the problem (A.11) are

$$[\nabla g(\hat{\Pi}^t)]_{kk'} + [1 + \log(\hat{\Pi}^{t+1}_{kk'}/\hat{\Pi}^t_{kk'})]/s + \lambda_k + \eta_{k'} = 0, \tag{A.12}$$

for $1 \leq k \leq K^{(1)}$, $1 \leq k' \leq K^{(2)}$, where λ_k and $\eta_{k'}$ are Lagrange multipliers for the row sum and column sum constraints, respectively. This implies that

$$\hat{\Pi}^{t+1}_{kk'} = \hat{\Pi}^t_{kk'} \exp\{-s\lambda_k - s\eta_{k'} - s[\nabla g(\hat{\Pi}^t)]_{kk'} - 1\}.$$

The gradient is given by

$$\nabla g(\Pi) = -\sum_{i=1}^{n} \frac{\hat{\phi}^{(2)}_i [\hat{\phi}^{(1)}_i]^T}{[\hat{\phi}^{(1)}_i]^T \Pi \hat{\phi}^{(2)}_i}. \tag{A.13}$$

In the special case of (A.8), writing $G_{kk'} = [\nabla g(\hat{\Pi}^t)]_{kk'}$, the update

$$\hat{\Pi}^{t+1}_{kk'} = \hat{\Pi}^t_{kk'} \exp\{sG_{kk'} - 1\} \exp\{-s\lambda_k\} \exp\{-s\eta_{k'}\}$$

can be written as

$$\hat{\Pi}^{t+1} = \mathrm{diag}(v) M \mathrm{diag}(u),$$

where $u_k = \exp\{-s\lambda_k\}$, $v_{k'} = \exp\{-s\eta_{k'}\}$, and $M_{kk'} = \hat{\Pi}^t_{kk'} \exp\{sG_{kk'} - 1\}$.

Algorithm 7 Exponentiated Gradient Descent for Solving (A.8)

1. Choose a fixed step size s and compute $\hat{\phi}^{(1)}$ and $\hat{\phi}^{(2)}$ according to (2.9).

2. For $t = 1, 2, \ldots$, until convergence,

   (a) Define

   $$M_{kk'} = \hat{\Pi}^t_{kk'} \exp\{sG_{kk'} - 1\}$$

   where

   $$G_{kk'} = \sum_{i=1}^{n} \frac{\hat{\phi}^{(1)}_{ik} \hat{\phi}^{(2)}_{ik'}}{[\hat{\phi}^{(1)}_i]^T \hat{\Pi}^t \hat{\phi}^{(2)}_i}.$$

   (b) (Sinkhorn-Knopp algorithm) Let $u^0 = 1_{K^{(2)}}$ and $v^0 = 1_{K^{(1)}}$. For $t' = 1, 2, \ldots$, until convergence,

   - $u^{t'} = \frac{\hat{\pi}^{(2)}}{(\hat{\Pi}^{t+1})^T v^{t'-1}}$, where the fraction denotes element-wise vector division
   - $v^{t'} = \frac{\hat{\pi}^{(1)}}{\hat{\Pi}^{t+1} u^{t'}}$, where the fraction denotes element-wise vector division

   (c) Let u and v denote the vectors to which $u^{t'}$ and $v^{t'}$ converge. Update

   $$\hat{\Pi}^{t+1}_{kk'} = u_k M_{kk'} v_{k'}.$$

Since $\hat{\Pi}^{t+1}$ must satisfy the row and column sum constraints, u and v must be chosen accordingly. As in [22], we can apply the Sinkhorn Theorem and Sinkhorn-Knopp algorithm to find u and v. By the Sinkhorn Theorem, u and v are unique modulo scalar multiplication of u with a positive number and scalar division of v by that same positive number. The Sinkhorn-Knopp algorithm (alternatively rescaling the columns so that the rows sum to $\hat{\pi}^{(1)}$ then rescaling the rows so that the columns sum to $\hat{\pi}^{(2)}$) can be applied to $M_{kk'}$ to find u and v. Hence, to perform the update, we simply multiply $\hat{\Pi}^t_{kk'}$ by $\exp\{sG_{kk'} - 1\}$ and then apply the Sinkhorn-Knopp algorithm to the updated matrix so that the row and column sum constraints are satisfied. Algorithm 7 provides the details. Using Proposition 1, we can then use our bijection between Π to C (A.9) to obtain Algorithm 1 from Algorithm 7.

A.4 Mean matrices for simulations in Chapter 2.4

In the simulations described in Chapter 2.4, data are generated from (2.1)–(2.2) with

$$\Pi = \frac{1-\delta}{K^2} 1_K 1_K^T + \frac{\delta}{K} I_K \tag{A.14}$$

for $K = 6$. In the lth data view, the observations are drawn from a multivariate Gaussian mixture model, for which the kth component in the mixture (corresponding to the kth cluster) is a $N_p(\mu_k^{(l)}, \sigma^2 I_p)$ distribution, with $p = 10$. The $p \times K$ mean matrices for the two data views are of the form

$$\mu^{(1)} = \begin{bmatrix} 2 \cdot 1_5 & 0_5 & 2 \cdot 1_5 & -2 \cdot 1_5 & 0_5 & -2 \cdot 1_5 \\ 0_5 & 2 \cdot 1_5 & -2 \cdot 1_5 & 0_5 & -2 \cdot 1_5 & 2 \cdot 1_5 \end{bmatrix},$$

$$\mu^{(2)} = \begin{bmatrix} -2 \cdot 1_6 & 0_6 & -2 \cdot 1_6 & 2 \cdot 1_6 & 0_4 & 2 \cdot 1_4 \\ 0_4 & -2 \cdot 1_4 & 2 \cdot 1_4 & 0_4 & 2 \cdot 1_6 & -2 \cdot 1_6 \end{bmatrix}.$$

A.5 Supplementary simulations to Section 2.4

A.5.1 Selection of the number of clusters

Recall from Chapter 2.4 that using too few clusters in the pseudo likelihood ratio test sometimes yields better power than using the correct number of clusters. In this section, we demonstrate that using too few clusters can either increase or decrease the power, depending on the situation.

We generate data from (2.1)–(2.2) with $\Pi = \frac{1-\delta}{K^2} 1_K 1_K^T + \frac{\delta}{K} I_K$ for $K = 6$. In the lth data view, the observations are drawn from a bivariate Gaussian mixture model, for which the kth component in the mixture (corresponding to the kth cluster) is a $N_2(\mu_k^{(l)}, 0.4^2 I_p)$ distribution. We simulate 2000 datasets for a range of values of n, and for two choices of $\mu^{(l)}$:

Choice 1:

$$\mu^{(1)} = \begin{bmatrix} 2 & 2 & -2 & -2 \\ -2 & -1 & 1 & 2 \end{bmatrix}, \quad \mu^{(2)} = \begin{bmatrix} -2 & -2 & 2 & 2 \\ -2 & -1 & 1 & 2 \end{bmatrix}, \tag{A.15}$$

Choice 2:

$$\mu^{(1)} = \begin{bmatrix} 2 & 2 & -2 & -2 \\ -2 & -1 & 1 & 2 \end{bmatrix}, \quad \mu^{(2)} = \begin{bmatrix} 2 & -2 & -2 & 2 \\ 2 & -2 & -1 & 1 \end{bmatrix}. \tag{A.16}$$

We evaluate the power of the pseudo likelihood ratio test of $H_0 : C = 1_{K_1} 1_{K_2}^T$ at nominal significance level $\alpha = 0.05$, when the number of clusters is incorrectly and correctly specified. Results are displayed in Figure A.1.

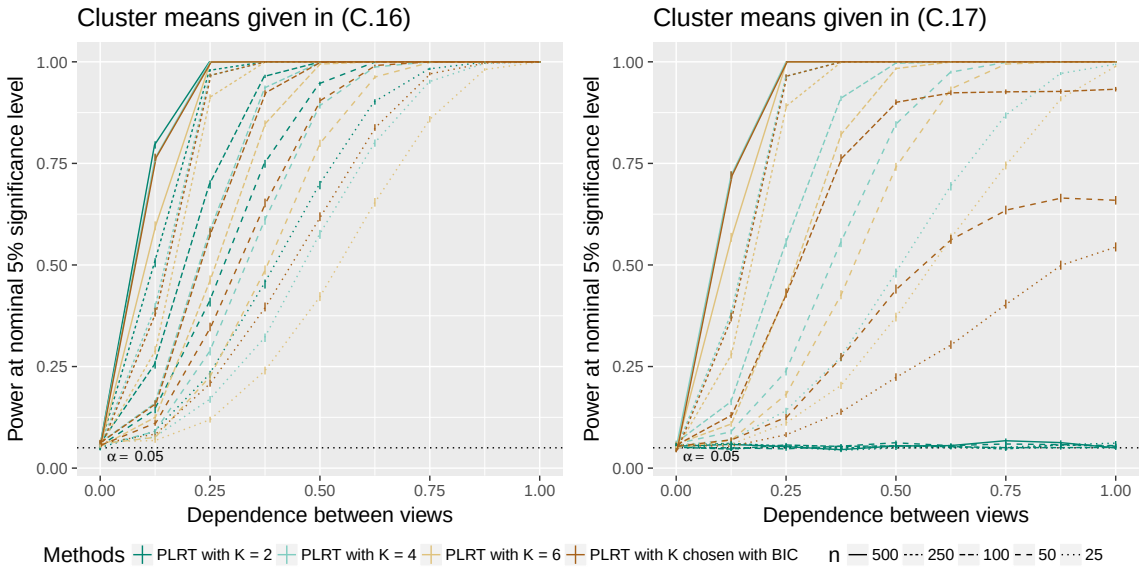

Figure A.1: Power of the pseudo likelihood ratio test of $H_0 : C = 1_{K_1} 1_{K_2}^T$ with $p = 2$, $K = 4$, and $\sigma = 0.4$ in the two simulation settings described in Appendix A.5.1. The x-axis displays δ, defined in (A.14), and the y-axis displays the power.

Observe from the left panel of Figure A.1 that when $\mu^{(l)}$ is given in (A.15), using too few clusters ($K = 2$) yields higher power than the correct number of clusters ($K = 4$). This is because under (A.15), the clusterings on each view contain two natural "meta-clusters", formed by combining clusters whose means are close; see Figure A.2(i) for an illustration. Because the clusters on each view are not well-separated, it is easier to instead cluster the data into the two "meta-clusters", which are highly in agreement when the four clusters are in agreement. For example, when $\Pi = I_4/4$, the Π matrix corresponding to the "meta-

clustering" is given by $I_2/2$. Thus, testing for independence assuming just two clusters yields better power than correctly assuming four clusters.

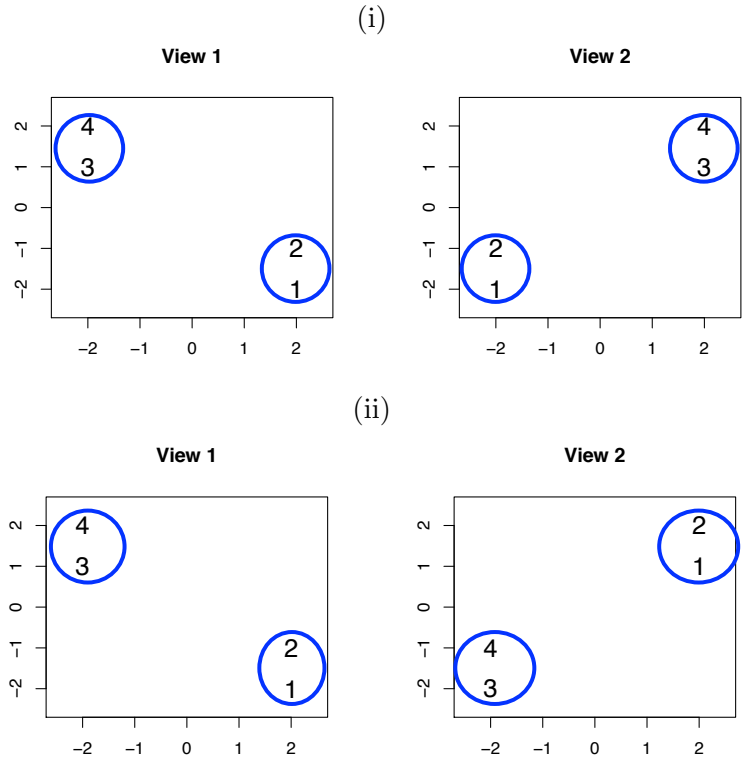

Figure A.2: (i) For the simulation study described in Appendix A.5.1, the cluster means and "meta-clusters" under (A.15), where the kth cluster mean on each view is indicated by the number k, and the two "meta-clusters" on each view are circled in blue. (ii) For the simulation study described in Appendix A.5.1, the cluster means and "meta-clusters" under (A.16), where the kth cluster mean on each view is indicated by the number k, and the two "meta-clusters" on each view are circled in blue.

By contrast, observe from the right panel of Figure A.1 that when $\mu^{(l)}$ is given in (A.16), using too few clusters ($K = 2$) yields much lower power than the correct number of clusters ($K = 4$). Again, under (A.16), the clusterings on each view contain two natural "meta-clusters" (see Figure A.2(ii)), and it is easier to cluster the data into the two "meta-clusters". However, under (A.16), even when the four clusters are highly in agreement, the meta-clusters are not highly in agreement. For example, when $\Pi = I_4/4$, the Π matrix corresponding to the "meta-clustering" is given by $1_2 1_2^T/4$. Thus, testing for independence

assuming just two clusters yields worse power than correctly assuming four clusters.

We also observe in both panels of Figure A.1 that using too many clusters yields slightly lower power than using the correct number of clusters under both (A.15) and (A.16); this result is similar to the results from the simulation setting described in Section 4 and Appendix A.5.2. Furthermore, in both panels of Figure A.1, corresponding to the two choices of $\mu^{(l)}$, choosing the clusters using BIC on each view yields slightly lower power than using the correct number of clusters in small and moderate sample sizes, and performs as well as using the correct number of clusters when the sample size is large.

A.5.2 Additional values of K and p

We simulate two data views with $p_1 = p_2 = p = 10$, and $K^{(1)} = K^{(2)} = K = 3$. In the lth data view, the observations are drawn from a multivariate Gaussian mixture model, for which the kth component in the mixture (corresponding to the kth cluster) is a $N_p(\mu_k^{(l)}, \sigma^2 I_p)$ distribution.

The means of the components in the mixture model, written as a $p \times K$ matrix, are

$$\mu^{(1)} = \begin{bmatrix} 2 \cdot 1_5 & 0_5 & 2 \cdot 1_5 \\ 0_5 & 2 \cdot 1_5 & -2 \cdot 1_5 \end{bmatrix}, \quad \mu^{(2)} = \begin{bmatrix} -2 \cdot 1_6 & 0_6 & -2 \cdot 1_6 \\ 0_4 & -2 \cdot 1_4 & 2 \cdot 1_4 \end{bmatrix}.$$

Additionally, we simulate two data views with $p_1 = p_2 = p = 100$, and $K^{(1)} = K^{(2)} = K$, for $K = 3$ and $K = 6$. In the lth data view, the observations are drawn from a multivariate Gaussian mixture model, for which the kth component in the mixture (corresponding to the kth cluster) is a $N_p(\mu_k^{(l)}, \sigma^2 I_p)$ distribution. For $K = 3$, the means of the components in the mixture model, written as a $p \times K$ matrix, are

$$\mu^{(1)} = \begin{bmatrix} 2 \cdot 1_{50} & 0_{50} & 2 \cdot 1_{50} \\ 0_{50} & 2 \cdot 1_{50} & -2 \cdot 1_{50} \end{bmatrix},$$
$$\mu^{(2)} = \begin{bmatrix} -2 \cdot 1_{60} & 0_{60} & -2 \cdot 1_{60} \\ 0_{40} & -2 \cdot 1_{40} & 2 \cdot 1_{40} \end{bmatrix}.$$

For $K = 6$, the means are

$$\mu^{(1)} = \begin{bmatrix} 2 \cdot 1_{50} & 0_{50} & 2 \cdot 1_{50} & -2 \cdot 1_{50} & 0_{50} & -2 \cdot 1_{50} \\ 0_{50} & 2 \cdot 1_{50} & -2 \cdot 1_{50} & 0_{50} & -2 \cdot 1_{50} & 2 \cdot 1_{50} \end{bmatrix},$$

$$\mu^{(2)} = \begin{bmatrix} -2 \cdot 1_{60} & 0_{60} & -2 \cdot 1_{60} & 2 \cdot 1_{60} & 0_{40} & 2 \cdot 1_{40} \\ 0_{40} & -2 \cdot 1_{40} & 2 \cdot 1_{40} & 0_{40} & 2 \cdot 1_{60} & -2 \cdot 1_{60} \end{bmatrix}.$$

To investigate the type I error and power of our test, we generate data according to (2.1)–(2.2), with a range of Π defined in (A.14).

We simulate 2000 datasets for a range of values of n and σ, and evaluate the power of the pseudo likelihood ratio test of $H_0 : C = 1_{K_1} 1_{K_2}^T$ at nominal significance level $\alpha = 0.05$, when the number of clusters is correctly and incorrectly specified. Results are shown in Figures A.3, A.4, and A.5.

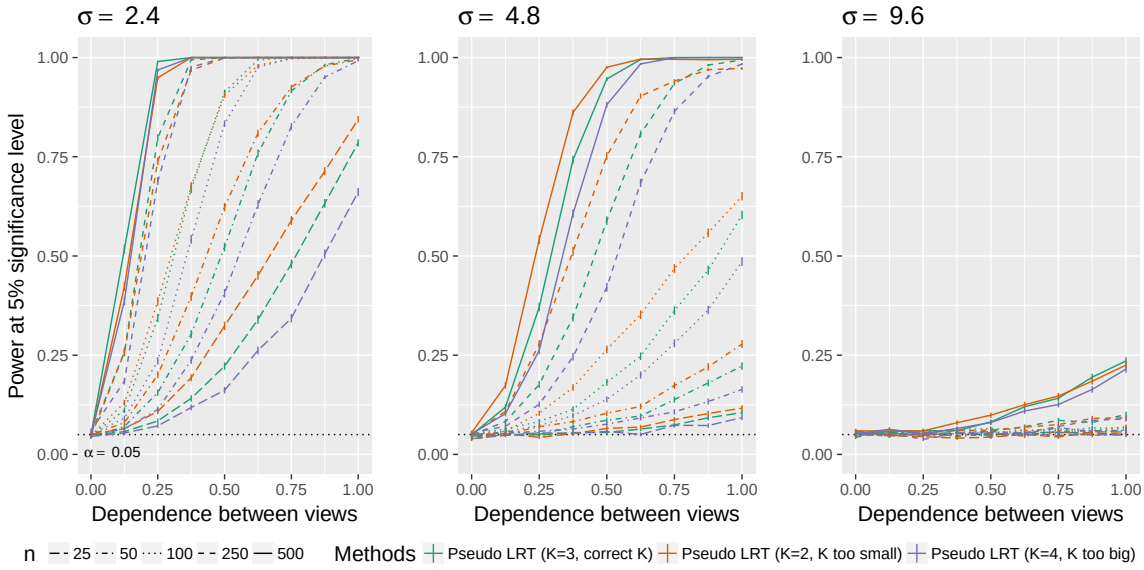

Figure A.3: Power of the pseudo likelihood ratio test of $H_0 : C = 1_{K_1} 1_{K_2}^T$ with $p = 10$, $K = 3$ and $\sigma \in \{2.4, 4.8, 9.6\}$ in the simulation setting described in Appendix A.5.2. The x-axis displays δ, defined in (A.14), and the y-axis displays the power.

Results are similar to the simulation study described in Section 2.4.

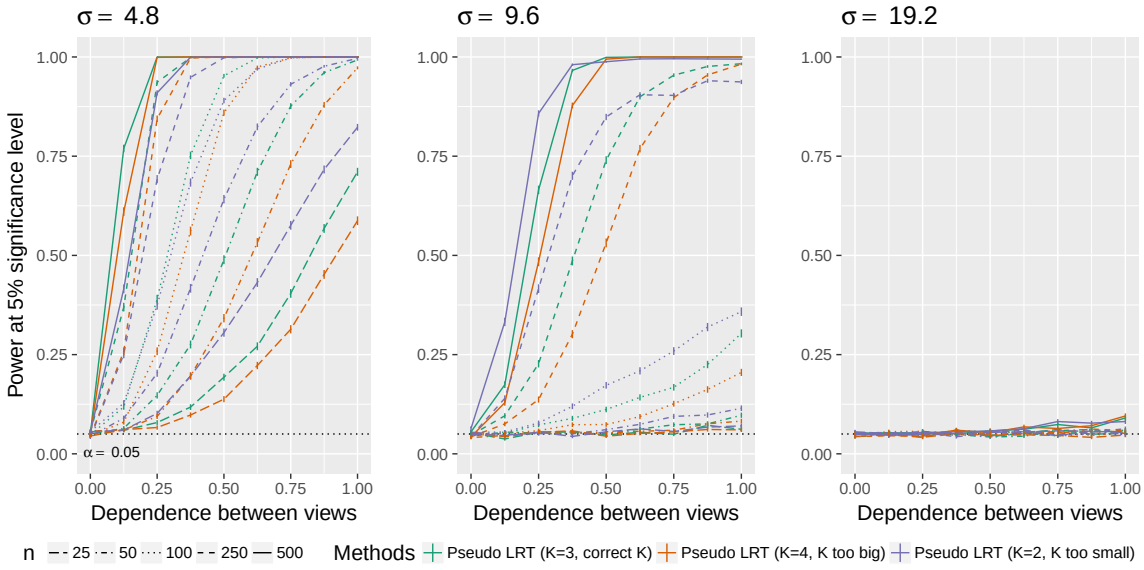

Figure A.4: Power of the pseudo likelihood ratio test of $H_0 : C = 1_{K_1} 1_{K_2}^T$ with $p = 100$, $K = 3$ and $\sigma \in \{4.8, 9.6, 19.2\}$ in the simulation setting described in Appendix A.5.2. The x-axis displays δ, defined in (A.14), and the y-axis displays the power.

A.6 Supplementary simulations to Section 2.5

In this section, we consider the G-test for independence in Section 2.5 using $\hat{M}^{(1)}$ and $\hat{M}^{(2)}$ defined in (2.19). As in Section 2.5, we perform a simulation study in order to compare the performances of the pseudo likelihood ratio test for testing the null hypothesis $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$ and the G-test for independence. We obtain p-values for $G^2(\hat{M}^{(1)}, \hat{M}^{(2)})$ (2.15) using the χ^2 approximation from (2.16), as well as a permutation approach, where we take B permutations of the elements of $\hat{M}^{(2)}$, and compare the observed value of $G^2(\hat{M}^{(1)}, \hat{M}^{(2)})$ to its empirical distribution in these permutation samples.

A.6.1 Additional values of p and K

We return to the simulation set-up described in Section 2.4 and Appendix A.4, with $\Sigma^{(1)} = \Sigma^{(2)} = \sigma^2 I_p$, and investigate a range of values of p and K. In addition to the G-test for independence and the pseudo likelihood ratio test, we also compute the adjusted Rand Index (ARI) of [48] in order to compare the results of model-based clustering (implemented as in

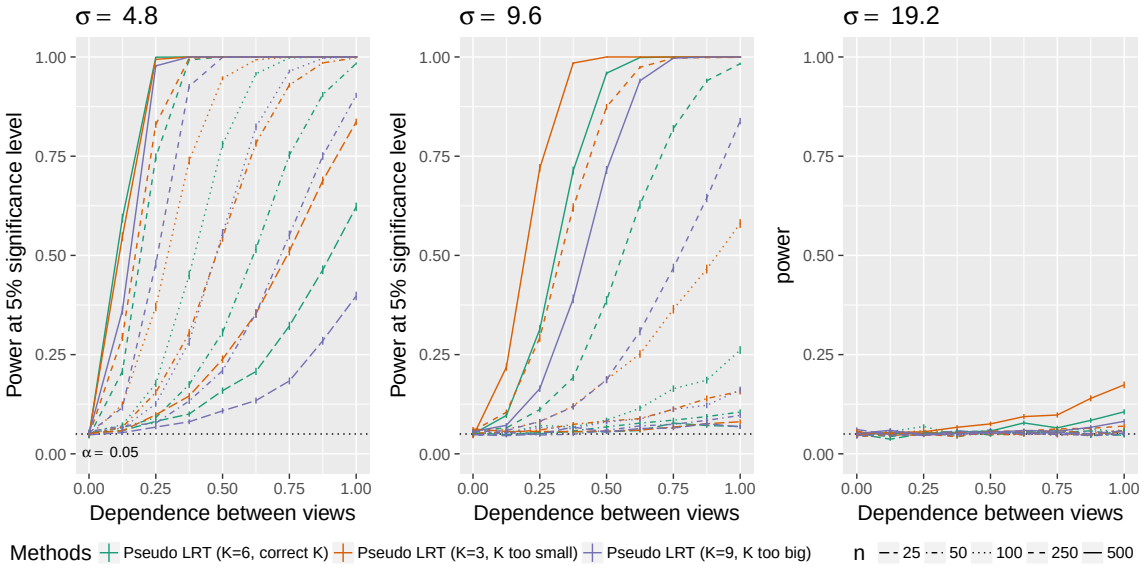

Figure A.5: Power of the pseudo likelihood ratio test of $H_0 : C = 1_{K_1} 1_{K_2}^T$ with $p = 100$, $K = 6$ and $\sigma \in \{4.8, 9.6, 19.2\}$ in the simulation setting described in Appendix A.5.2. The x-axis displays δ, defined in (A.14), and the y-axis displays the power.

Sections 2.2.3 and 2.3) on each view; p-values for the ARI are obtained using a permutation approach. We compare the performance of the pseudo likelihood ratio test, the G-test for independence, and the ARI for testing the null hypothesis $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$. The results are in Figures A.6, A.7, and A.8. Results are similar to results from the $K = 6$ and $p = 10$ setting in Figure 2.3; we note that the ARI performs similarly to the G-test for independence in all cases.

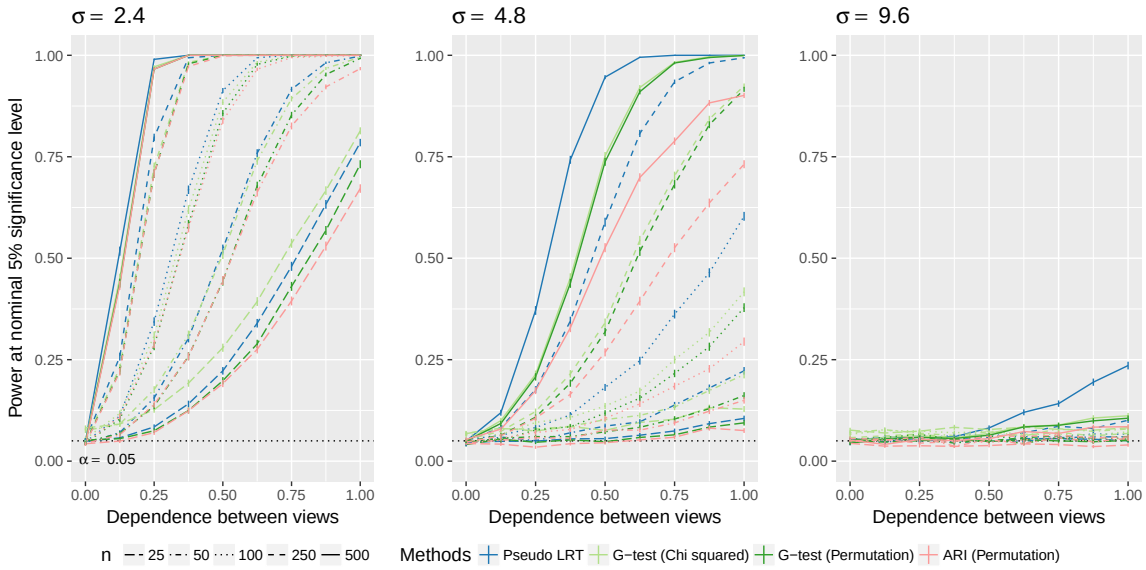

Figure A.6: For the simulation study described in Appendix A.6.1, power of the pseudo likelihood ratio test, the G-test for independence, and the adjusted Rand Index (ARI) for $p = 10$, $K = 3$ and $\sigma \in \{2.4, 4.8, 9.6\}$, with δ, defined in (A.14) on the x-axis and power on the y-axis.

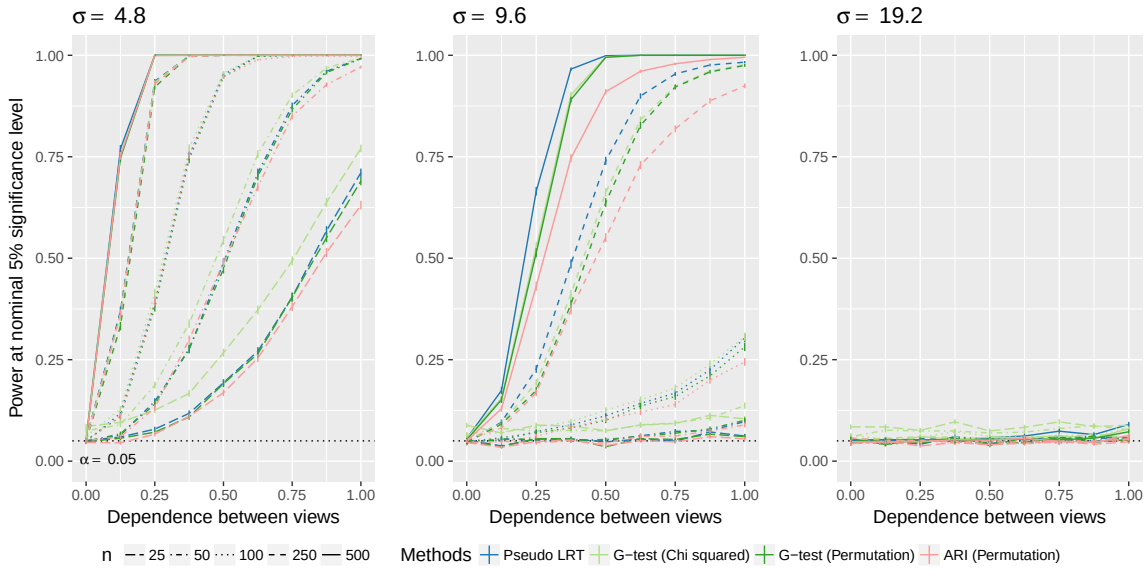

Figure A.7: For the simulation study described in Appendix A.6.1, power of the pseudo likelihood ratio test, the G-test for independence, and the adjusted Rand Index (ARI) for $p = 100$, $K = 3$ and $\sigma \in \{4.8, 9.6, 19.2\}$, with δ, defined in (A.14), on the x-axis, and power on the y-axis.

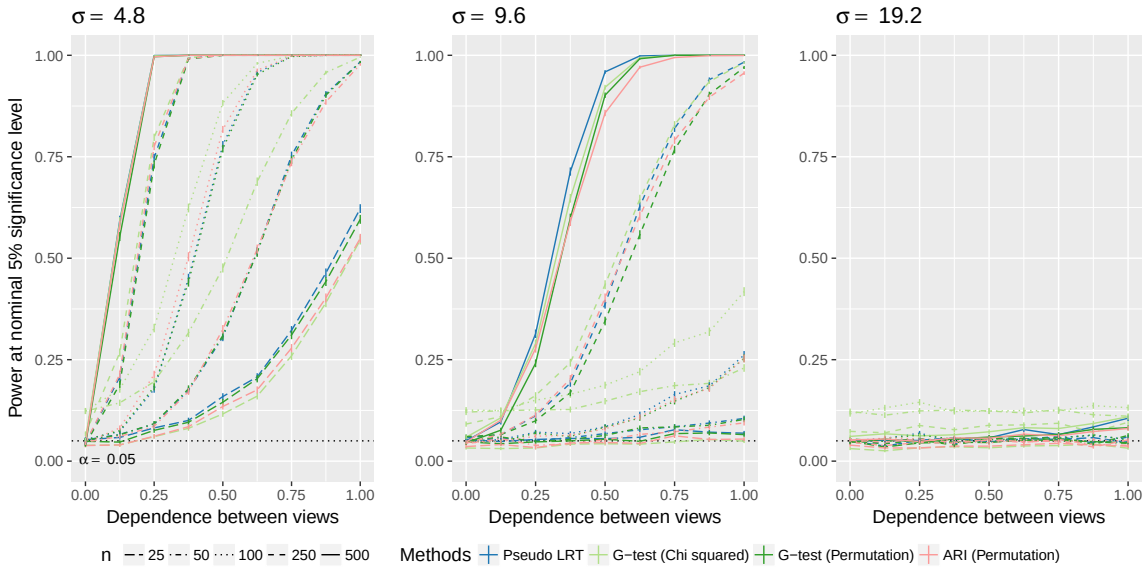

Figure A.8: For the simulation study described in Appendix A.6.1, power of the pseudo likelihood ratio test, the G-test for independence, and the adjusted Rand Index (ARI) for $p = 100$, $K = 6$ and $\sigma \in \{4.8, 9.6, 19.2\}$, with δ, defined in (A.14), on the x-axis, and power on the y-axis.

A.6.2 Additional values of $\Sigma^{(1)}$ *and* $\Sigma^{(2)}$

We return to the simulation set-up described in Section 2.4 with $K = 3$, $\mu^{(l)}$ given by

$$\mu^{(1)} = \begin{bmatrix} 0 & 0 & \sqrt{12} \\ 2 & -2 & 0 \end{bmatrix}, \quad \mu^{(2)} = \begin{bmatrix} -2 & 0 & 2 \\ 0 & \sqrt{12} & 0 \end{bmatrix}, \tag{A.17}$$

and for two choices of $\Sigma^{(l)}$:

$$\textit{Choice 1: } \Sigma^{(1)} = \Sigma^{(2)} = \begin{pmatrix} 2.25 & 0.5 \\ 0.5 & 2.25 \end{pmatrix},$$

$$\textit{Choice 2: } \Sigma^{(1)} = \begin{pmatrix} 2.25 & 0.5 \\ 0.5 & 2.25 \end{pmatrix} \text{ and } \Sigma^{(2)} = \text{diag}(2.25, 4).$$

To perform Step 1 of Algorithm 1, we use the package `mclust` [96] to fit Gaussian mixture models with a common dense covariance matrix (the "EEE" covariance structure in `mclust`) for $\Sigma^{(l)}$ given by choice 1 above, and to fit Gaussian mixture models with a common diagonal

covariance matrix (the "EEI" covariance structure in `mclust`) for $\Sigma^{(l)}$ given by choice 2 above. We compare the performance of the pseudo likelihood ratio test of $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$ at nominal significance level $\alpha = 0.05$ to the G-test for independence for testing the null hypothesis $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$. The results are in Figure A.9(i) and Figure A.9(ii). Results are similar to results from the $K = 6$ and $p = 10$ setting with $\sigma = 4.8$ in Figure 2.3.

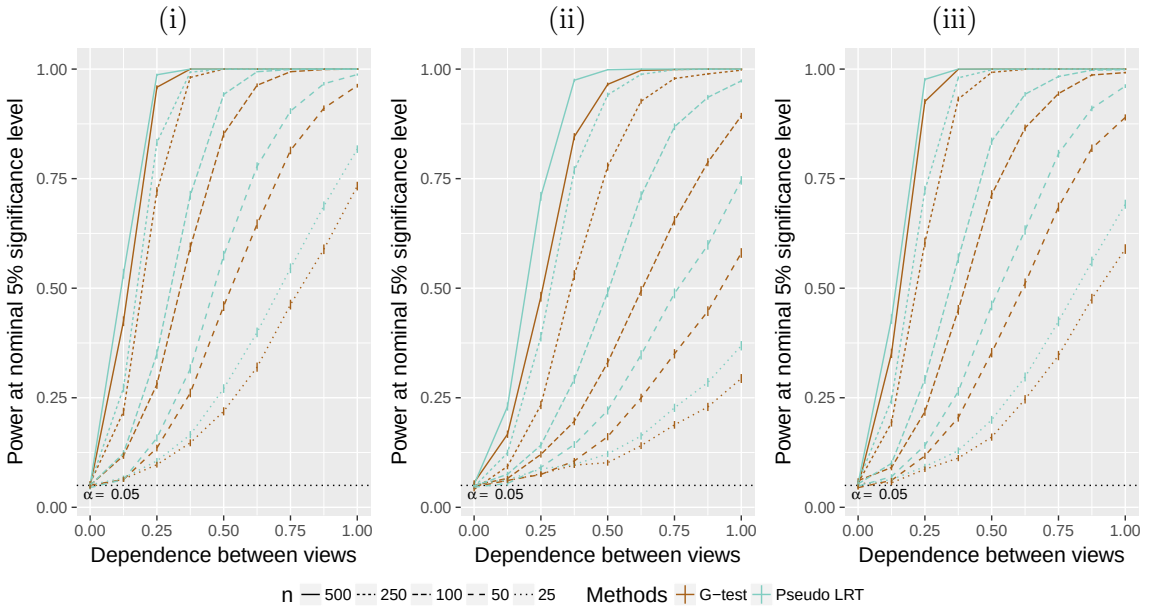

Figure A.9: The x-axis displays δ, defined in (A.14), and the y-axis displays power. Power of the pseudo likelihood ratio test and the G-test for independence for (i) Gaussian mixture components with $\Sigma^{(1)} = \Sigma^{(2)} = \Sigma$, where Σ has no non-zero elements, and (ii) Gaussian mixture components with $\Sigma^{(1)} = \Sigma^{(2)} = \Sigma$, where Σ is diagonal, and (iii) bivariate Student's t-distributions as mixture components. Details for (i) and (ii) are in Appendix A.6.2, and details for (iii) are in Appendix A.6.3.

A.6.3 Model misspecification

We consider a simulation set-up which compares the performance of the pseudo likelihood ratio test under model misspecification. We generate data from model (2.1) – (2.2), with $\Pi = \frac{1-\delta}{K^2} 1_K 1_K^T + \frac{\delta}{K} I_K$ for $K = 3$. In the lth data view, the observations are drawn from a finite mixture model for which the kth component in the mixture (corresponding to the kth cluster) is a bivariate Student's t-distribution with location parameter $\mu_k^{(l)}$ and scale

matrix $\Sigma = \begin{pmatrix} 2.25 & 0.5 \\ 0.5 & 2.25 \end{pmatrix}$, where the mean matrices for the two data views are of the form (A.17). We fit Gaussian mixture models with a common covariance matrix (the "EEE" covariance structure in `mclust`), and again use $B = 200$ permutation samples. The results are in Figure A.9(iii). We compare the performance of the pseudo likelihood ratio test of $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$ at nominal significance level $\alpha = 0.05$ to the G-test for independence for testing the null hypothesis $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$, with p-values obtained with a permutation approach. Results remain similar to results from the $K = 6$ and $p = 10$ setting with $\sigma = 4.8$ in Figure 2.3.

Appendix B

SUPPLEMENTARY MATERIALS FOR CHAPTER 3

B.1 A detailed review of [4]

Let $\hat{Z} \in \{1, \ldots, K\}^n$ be an initial estimate of the community memberships of the n nodes. Specifically, [4] proposed using a regularized spectral clustering procedure called spectral clustering with perturbations to obtain $\hat{Z}$. In what follows, the dependency of $\hat{Z}$ on X is ignored, and $\hat{Z}$ is treated as fixed. Let $\hat{b}$ be the $n \times K$ matrix defined by

$$\hat{b}_{im} = \sum_{j=1}^{n} X_{ij} \mathbb{1}\{\hat{Z}_j = m\}, \quad 1 \leq i \leq n, 1 \leq m \leq K. \tag{B.1}$$

Let $\hat{b}_i$ denote the ith row of $\hat{b}$. Let $d = X1_n$. In this section, we review the derivation of a pseudolikelihood function from [4] which is based on an approximation to the conditional density of $\hat{b}$ given d. We note that [4] also derived a pseudolikelihood function which is based on the unconditional density of $\hat{b}$. However, the estimators which maximize the former pseudolikelihood function are more robust against misspecification of the conditional distribution of X given Z in the stochastic block model (Section 3.2.1) than the estimators which maximize the latter pseudolikelihood function [4]. This is because the conditional distribution of X given Z in the stochastic block model provides a poor fit to networks with heterogeneous node degrees within communities, and conditioning on d (the node degrees) improves the goodness of fit.

It follows from the definition of the stochastic block model (Section 3.2.1) that:

- For $(i, j), (i', j') \in \{1, 2, \ldots, n\}^2$, conditional on Z, $X_{ij} \perp X_{i'j'}$, and

- For $(i, j, m), (i', j', m') \in \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\} \times \{1, 2, \ldots, K\}$, conditional on Z,

$$X_{ij} \mathbb{1}\{\hat{Z}_j = m\} \perp X_{i'j'} \mathbb{1}\{\hat{Z}_{j'} = m'\}. \tag{B.2}$$

Thus, conditional on Z, $\{(\hat{b}_i, d_i)\}_{i=1}^n$ are weakly dependent when n is large, and so

$$f(\{\hat{b}_i\}_{i=1}^n \mid Z, d) = \frac{f(\{\hat{b}_i\}_{i=1}^n, d \mid Z)}{f(d \mid Z)} \approx \frac{\prod_{i=1}^n f(\hat{b}_i, d_i \mid Z)}{\prod_{i=1}^n f(d_i \mid Z)} = \prod_{i=1}^n f(\hat{b}_i \mid Z, d_i). \tag{B.3}$$

Next, we derive approximations to $f(\hat{b}_i \mid Z, d_i)$. Recall from the definition of the stochastic block model (Section 3.2.1) that conditional on Z, X_{ij} are independent Bernoulli variables for $1 \leq i < j \leq n$. Thus, it follows from the definition of $\hat{b}_{im}$ in (B.1) that conditional on Z, $\hat{b}_{im}$ is the sum of independent Bernoulli random variables, and can be approximated by a Poisson distribution:

$$\hat{b}_{im} \mid Z \mathrel{\dot{\sim}} \text{Poisson}\left(\sum_{j=1}^n \mathbb{E}[X_{ij} \mathbb{1}\{\hat{Z}_j = m\} \mid Z]\right). \tag{B.4}$$

Ignoring the fact that $X_{ii} = 0$, and instead assuming that $X_{ii} \mid Z \sim \text{Bernoulli}(\theta_{Z_i Z_i})$ with $\{X_{ij}\}_{1 \leq j \leq i \leq n}$ conditionally independent given Z,

$$\mathbb{E}[\hat{b}_{im} \mid Z] \approx \sum_{j=1}^n \theta_{Z_i Z_j} \mathbb{1}\{\hat{Z}_j = m\} = \sum_{j=1}^n \sum_{m'=1}^K \theta_{Z_i m'} \mathbb{1}\{\hat{Z}_j = m, Z_j = m'\} = \sum_{m'=1}^K \theta_{Z_i m'} \hat{R}_{mm'}, \tag{B.5}$$

where $\hat{R}$ is the confusion matrix of $\hat{Z}$ defined by

$$\hat{R}_{mm'} = \sum_{j=1}^n \mathbb{1}\{\hat{Z}_j = m, Z_j = m'\}, \quad 1 \leq m \leq K, 1 \leq m' \leq K. \tag{B.6}$$

Combining (B.4) and (B.5),

$$\hat{b}_{im} \mid Z \mathrel{\dot{\sim}} \text{Poisson}\left(\sum_{m'=1}^K \theta_{Z_i m'} \hat{R}_{mm'}\right), \quad 1 \leq i \leq n, 1 \leq m \leq K. \tag{B.7}$$

Now, the joint distribution of independent Poisson random variables conditional on their sum is multinomial. It follows from (B.1) and (B.2) that $\{\hat{b}_{im}\}_{i=1}^n$ are conditionally independent given Z. Furthermore, from (B.7), conditional on Z, $\hat{b}_{im}$ are approximately Poisson. Thus,

$$\hat{b}_i \mid d_i, Z \mathrel{\dot{\sim}} \text{Multinomial}\left(d_i, \left(\frac{\sum_{m'=1}^K \theta_{Z_i m'} \hat{R}_{1m'}}{\sum_{m=1}^K \sum_{m'=1}^K \theta_{Z_i m'} \hat{R}_{mm'}}, \ldots, \frac{\sum_{m'=1}^K \theta_{Z_i m'} \hat{R}_{Km'}}{\sum_{m=1}^K \sum_{m'=1}^K \theta_{Z_i m'} \hat{R}_{mm'}},\right)\right), \quad 1 \leq i \leq n. \tag{B.8}$$

We use (B.8) to write

$$\hat{b}_i \mid d_i, Z \dot{\sim} g(\hat{b}_i; d_i, \eta_{Z_i}), \quad 1 \leq i \leq n, \tag{B.9}$$

where $g(\cdot; q)$ denotes the probability mass function of a Multinomial$(N, q_1, \ldots, q_K)$ random variable, and $\eta = \left(\text{diag}(\theta\hat{R}1_K)\right)^{-1}\theta\hat{R}$. Now, combining (B.3) and (B.9),

$$\hat{b} \mid Z, d \dot{\sim} \prod_{i=1}^{n} g(\hat{b}_i; d_i, \eta_{Z_i}). \tag{B.10}$$

Treating η as fixed, and marginalizing over Z in (B.10), ignoring any dependency of d on Z, yields

$$\hat{b} \mid d \dot{\sim} \prod_{i=1}^{n} \left(\sum_{k=1}^{K} \pi_k g(\hat{b}_i; d_i, \eta_k) \right). \tag{B.11}$$

Based on (B.11), [4] defined the log-pseudolikelihood function to be:

$$\ell_{PL}(\eta, \pi) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k g(\hat{b}_i; d_i, \eta_k) \right).$$

This is (3.4).

B.2 The DCSBM for two network data views with dependent popularities

In Section 3.7.2, we generated data from a DCSBM for two network data views, where $\delta^{(1)}$ (the popularities of the nodes in the first view) and $\delta^{(2)}$ (the popularities of the nodes in the second view) are independent. In this section, we will modify the DCSBM for two network data views to a case of maximal dependence between the node popularities of the two views: $\delta_i^{(1)} = \delta_i^{(2)}$ for all $i = 1, 2, \ldots, n$.

Type I error rate of the P^2LRT

We will generate each network view from the DCSBM. We generate n vectors $(Z_i^{(1)}, Z_i^{(2)}, \delta_i^{(1)}, \delta_i^{(2)})$ i.i.d. for $i = 1, 2, \ldots, n$, with $Z_i^{(1)}$ and $Z_i^{(2)}$ categorical with $K^{(1)}$ and $K^{(2)}$ levels, respectively, and $(Z_i^{(1)}, Z_i^{(2)}) \perp (\delta_i^{(1)}, \delta_i^{(2)})$. We let $\delta_i^{(1)} = \delta_i^{(2)}$ for $i = 1, 2, \ldots, n$, so that the node popularities in the two views are identical. We generate each view with

$$X^{(l)} \mid Z^{(l)}, \delta^{(l)} \sim \prod_{j=1}^{n} \prod_{i=1}^{j-1} \left(\delta_i^{(l)} \delta_j^{(l)} \theta^{(l)}_{Z_i^{(l)} Z_j^{(l)}} \right)^{X_{ij}^{(l)}} \left(1 - \delta_i^{(l)} \delta_j^{(l)} \theta^{(l)}_{Z_i^{(l)} Z_j^{(l)}} \right)^{1 - X_{ij}^{(l)}}, \quad l = 1, 2.$$

We set $n = 50$, $K^{(1)} = K^{(2)} = K = 2$, $\pi^{(1)} = \pi^{(2)} = 1_2/2$, $\theta^{(1)} = \theta^{(2)} = \begin{bmatrix} 0.5 & 0.25 \\ 0.25 & 1 \end{bmatrix}$, and $\delta_i^{(1)} \sim \text{Uniform}(0.14, 0.84)$. We let $C = 1_2 1_2^T$, so that $Z^{(1)}$ and $Z^{(2)}$ are independent. We simulate 200 data sets with $C = 1_2 1_2^T$.

We apply the P^2LRT of $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$ described in Section 3.4, using the same number of communities in each data view, and varying the number of communities used from 2 to $n = 50$. We also apply the P^2LRT using the value of $K^{(1)}$ and $K^{(2)}$ estimated by applying the method of [60] to $X^{(1)}$ and $X^{(2)}$, respectively. The results are shown in Figure B.1.

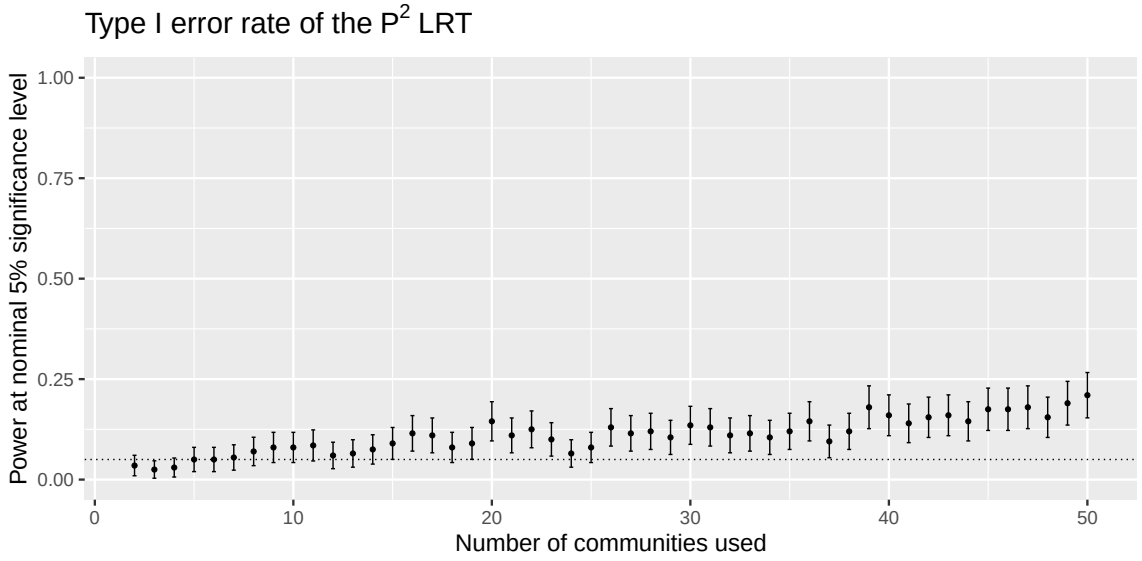

Figure B.1: For the simulation study described in Appendix B.2, we display the Type I error rate of the P^2LRT described in Section 3.4 for $n = 50$, $K = 2$, and $\delta_i^{(1)} = \delta_i^{(2)}$ for $i = 1, 2, \ldots, n$. The x-axis displays the number of communities used, and the y-axis displays Type I error rate. The Type I error rate of the P^2LRT with the value of $K^{(1)}$ and $K^{(2)}$ estimated by applying the method of [60] to $X^{(1)}$ and $X^{(2)}$, respectively, is 0.035 (95% confidence interval: $0.0095, 0.0605$).

We see that when we grossly overspecify the number of communities, the Type I error rate is inflated, and when we do not grossly overspecify the number of communities, the Type I error rate is controlled at the nominal $\alpha = 0.05$ level.

Number of communities used and Type I error rate

In this subsection, we will explain why the Type I error rate is inflated when $\delta^{(1)}$ and $\delta^{(2)}$ are dependent and we grossly overspecify the number of communities.

The P^2LRT statistic defined in (3.13) is closely related to the mutual information (a measure of dependence, [79]) between the estimated community memberships in each view; the derivation of this relationship is similar to the fifth section of Chapter 2. This suggests that if the community memberships in the two views are independent, but the *estimated* community memberships in the two views are dependent, then the Type I error rate will be inflated. Furthermore, if

1. the estimated community assignments in view 1 and $\delta^{(1)}$ are dependent,

2. the estimated community assignments in view 2 and $\delta^{(2)}$ are dependent, and

3. $\delta^{(1)}$ and $\delta^{(2)}$ are dependent,

then the estimated community assignments in the two views will likely be dependent.

In Appendix B.2, we generate data with $\delta^{(1)}$ and $\delta^{(2)}$ dependent. When we specify a very large number of communities, the estimation procedure tends to assign nodes with similar values of $\delta^{(l)}$ to the same community. Thus, Conditions 1–3 above are satisfied, leading to dependence between the estimated community memberships, and hence Type I error inflation.

When we do not grossly overspecify the number of communities, the estimated community assignments are not highly dependent on $\delta^{(l)}$, and thus the P^2LRT controls the Type I error rate. Estimating the number of communities using the method of [60] controls the Type I error rate, because the method of [60] does not grossly overspecify the number of communities.

B.2.1 DCSBM for a network view and a multivariate view

As in Section 3.7.3, we will evaluate the performance of four tests of $H_0 : C = 1_{K^{(1)}} 1_{K^{(2)}}^T$:

1. The P^2LRT proposed in Section 3.5.3, using the true values of $K^{(1)}$ and $K^{(2)}$,

2. The P^2LRT, using estimated values of $K^{(1)}$ and $K^{(2)}$,

3. The G-test applied to the estimated community assignments in the network view and the estimated cluster memberships in the multivariate view, using the true value of $K^{(1)}$ and $K^{(2)}$, and

4. The G-test, using the estimated values of $K^{(1)}$ and $K^{(2)}$,

where $K^{(1)}$ (the number of communities in the network view) is estimated by applying the method of [60] to X, and $K^{(2)}$ (the number of clusters in the multivariate view) is estimated using BIC. In all four tests, we approximate the null distribution with a permutation approach, as in Algorithm 4, using $M = 200$ permutation samples.

We generate the network data view from a DCSBM, and the multivariate data view from a Gaussian mixture model. We generate n vectors $(Z_i^{(1)}, Z_i^{(2)}, \delta_i)$ i.i.d. for $i = 1, 2, \ldots, n$, with $Z_i^{(1)}$ and $Z_i^{(2)}$ categorical with $K^{(1)}$ and $K^{(2)}$ levels, respectively, and $(Z_i^{(1)}, Z_i^{(2)}) \perp \delta_i$. We generate the network view with

$$X \mid Z^{(1)}, \delta \sim \prod_{j=1}^{n} \prod_{i=1}^{j-1} \left(\delta_i \delta_j \theta_{Z_i^{(1)} Z_j^{(1)}} \right)^{X_{ij}} \left(1 - \delta_i \delta_j \theta_{Z_i^{(1)} Z_j^{(1)}} \right)^{1 - X_{ij}},$$

and generate the multivariate data view with

$$Y \mid Z^{(2)} \sim \prod_{i=1}^{n} \phi(Y_i; \mu_k, \sigma^2 I_{10}),$$

where $\phi(\cdot; \mu, \Sigma)$ denotes the density of a $N_{10}(\mu, \Sigma)$ random variable. The mean matrix for the multivariate data view is given by $\mu = \begin{bmatrix} 0 \cdot 1_5 & 0 \cdot 1_5 & \sqrt{12} \cdot 1_5 \\ 2 \cdot 1_5 & -2 \cdot 1_5 & 0 \cdot 1_5 \end{bmatrix}$.

We set $n = 500$, and $K^{(1)} = K^{(2)} = K = 3$. Let $\pi^{(1)} = \pi^{(2)} = 1_K / K$, and let C be given by (3.20). Let θ be given by (3.21), so that the expected edge density $s = 0.015$. We simulate 2000 data sets for $n = 500$ and a range of values of Δ, r, and σ. Results are shown in Figure B.2, and are similar to the results in Section 3.7.3.

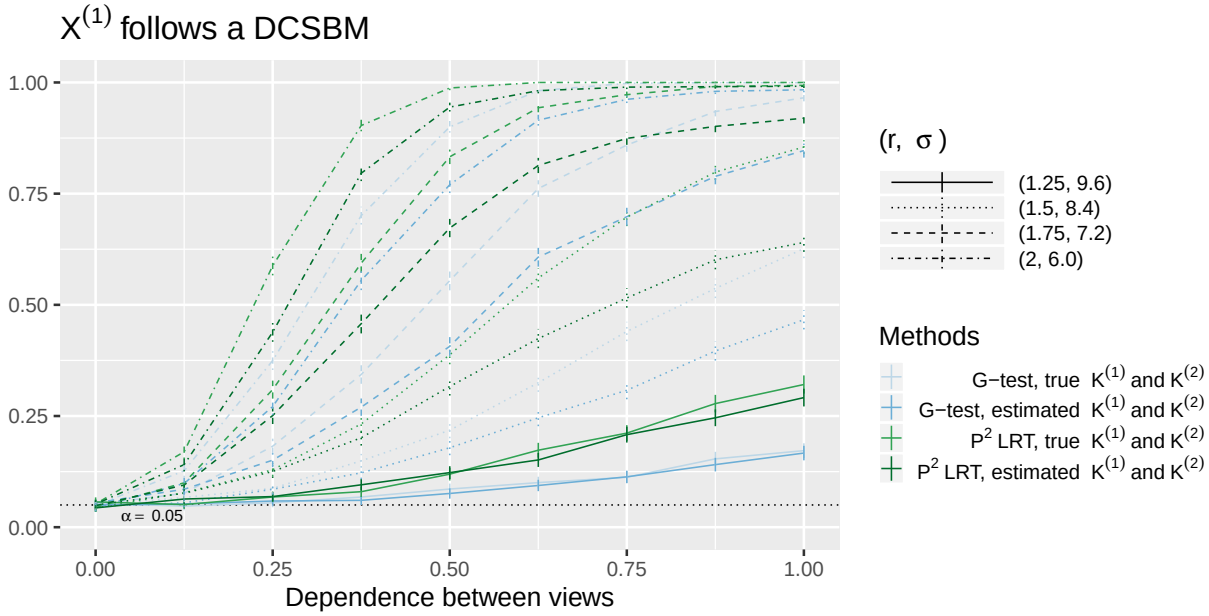

Figure B.2: Power of the P^2LRT and the G-test with the multivariate view drawn from a Gaussian mixture model and the network view drawn from a DCSBM, varying the dependence between views (Δ), the strength of the communities (r), the variance of the clusters (σ), and how the number of communities and the number of clusters are selected. The expected network density (s) is fixed at 0.015. Details are in Appendix B.2.1.

Appendix C

SUPPLEMENTARY MATERIALS FOR CHAPTER 4

C.1 Proof of Theorem 1

Recall that $\boldsymbol{\pi}_{\hat{\nu}}^{\perp}\mathbf{X} = \mathbf{I}_n - \frac{\hat{\nu}\hat{\nu}^T}{\|\hat{\nu}\|_2^2}$. It folllows from algebra that

$$\mathbf{X} = \boldsymbol{\pi}_{\hat{\nu}}^{\perp}\mathbf{X} + (\mathbf{I}_n - \boldsymbol{\pi}_{\hat{\nu}}^{\perp})\mathbf{X} = \boldsymbol{\pi}_{\hat{\nu}}^{\perp}\mathbf{X} + \left(\frac{\|\mathbf{X}^T\hat{\nu}\|_2}{\|\hat{\nu}\|_2^2}\right)\hat{\nu}\left(\text{dir}(\mathbf{X}^T\hat{\nu})\right)^T. \tag{C.1}$$

Substituting (C.1) into the definition of p given by (4.9) yields

$$p = \mathbb{P}_{H_0}\big(\|\mathbf{X}^T\hat{\nu}\|_2 \geq \|\mathbf{x}^T\hat{\nu}\|_2 \mid \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}\left(\boldsymbol{\pi}_{\hat{\nu}}^{\perp}\mathbf{x} + \left(\frac{\|\mathbf{X}^T\hat{\nu}\|_2}{\|\hat{\nu}\|_2^2}\right)\hat{\nu}\left(\text{dir}(\mathbf{x}^T\hat{\nu})\right)^T\right),$$
$$\boldsymbol{\pi}_{\hat{\nu}}^{\perp}\mathbf{X} = \boldsymbol{\pi}_{\hat{\nu}}^{\perp}\mathbf{x}, \text{dir}(\mathbf{X}^T\hat{\nu}) = \text{dir}(\mathbf{x}^T\hat{\nu})\big). \tag{C.2}$$

To simplify (C.2), we now show that

$$\|\mathbf{X}^T\hat{\nu}\|_2 \perp \boldsymbol{\pi}_{\hat{\nu}}^{\perp}\mathbf{X}, \tag{C.3}$$

and that under $H_0 : U^T\hat{\nu} = 0_q$,

$$\|\mathbf{X}^T\hat{\nu}\|_2 \perp \text{dir}(\mathbf{X}^T\hat{\nu}). \tag{C.4}$$

First, recall that $\boldsymbol{\pi}_{\hat{\nu}}^{\perp}$ is the orthogonal projection matrix onto the subspace orthogonal to span$(\hat{\nu})$. Thus, $\boldsymbol{\pi}_{\hat{\nu}}^{\perp}\hat{\nu} = 0_n$. It follows from properties of the matrix normal and multivariate normal distributions that $\boldsymbol{\pi}_{\hat{\nu}}^{\perp}\mathbf{X} \perp \mathbf{X}^T\hat{\nu}$. This implies (C.3). Secondly, it follows from (4.1) that $\mathbf{X}_i \overset{ind}{\sim} N_q(U_i, \sigma^2 I_q)$. Thus, under $H_0 : U^T\hat{\nu} = 0_q$, $\frac{\mathbf{X}^T\hat{\nu}}{\sigma\|\hat{\nu}\|_2} \sim N_q(0, I_q)$, and (C.4) follows from the independence of the length and direction of a standard multivariate normal random vector.

We now apply (C.3) and (C.4) to (C.2). This yields

$$p = \mathbb{P}_{H_0}\left(\|\mathbf{X}^T\hat{\nu}\|_2 \geq \|\mathbf{x}^T\hat{\nu}\|_2 \mid \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}\left(\boldsymbol{\pi}_{\hat{\nu}}^{\perp}\mathbf{x} + \left(\frac{\|\mathbf{X}^T\hat{\nu}\|_2}{\|\hat{\nu}\|_2^2}\right)\hat{\nu}\left(\text{dir}(\mathbf{x}^T\hat{\nu})\right)^T\right)\right). \tag{C.5}$$

Now, let $\phi = \|\mathbf{X}^T\hat{\nu}\|_2$. Recalling that $\mathbf{X}_i \overset{ind}{\sim} N_q(\mu_i, \sigma^2 I_q)$, observe that under $H_0 : \boldsymbol{\mu}^T\hat{\nu} = 0_q$, $\frac{\phi^2}{\sigma^2\|\hat{\nu}\|_2^2} \sim \chi_q^2$. Furthermore, it follows from algebra that

$$\boldsymbol{\pi}_{\hat{\nu}}^{\perp}\mathbf{x} + \left(\frac{\|\mathbf{X}^T\hat{\nu}\|_2}{\|\hat{\nu}\|_2^2}\right)\hat{\nu}\left(\mathrm{dir}(\mathbf{x}^T\hat{\nu})\right)^T = \mathbf{x} + \left(\frac{\phi - \|\mathbf{x}^T\hat{\nu}\|_2}{\|\hat{\nu}\|_2^2}\right)\hat{\nu}\left(\mathrm{dir}(\mathbf{x}^T\hat{\nu})\right)^T = x'(\phi).$$

Therefore,

$$p = \mathbb{P}\left(\phi \geq \|\mathbf{x}^T\hat{\nu}\|_2 \mid \hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}(x'(\phi))\right).$$

This is (4.10).

C.2 Proof of Lemma 2

Suppose that $\mathcal{C} = \mathcal{C}^{(n-K+1)}$. Now suppose that

$$\underset{\mathcal{H}_1,\mathcal{H}_2\in\mathcal{C}^{(l)}(\mathbf{x}),\ \mathcal{H}_1\neq\mathcal{H}_2}{\arg\min} d(\mathcal{H}_1, \mathcal{H}_2; \mathbf{x}'(\phi)) = (\mathcal{G}_1^{(l)}(\mathbf{x}), \mathcal{G}_2^{(l)}(\mathbf{x})), \quad \forall\ l = 1, 2, \ldots, n-K, \tag{C.6}$$

where $(\mathcal{G}_1^{(l)}(\mathbf{x}), \mathcal{G}_2^{(l)}(\mathbf{x}))$ is defined in Algorithm 5 to be the pair of clusters merged in $\mathbf{x}$ at the lth step of the agglomerative hierarchical clustering procedure. Then, $\mathcal{C}^{(n-K+1)}(\mathbf{x}'(\phi)) = \mathcal{C}^{(n-K+1)}(\mathbf{x})$, so since we defined $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ to be two estimated clusters in $\mathcal{C}(\mathbf{x}) = \mathcal{C}^{(n-K+1)}(\mathbf{x})$, we have that $\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}^{(n-K+1)}(\mathbf{x}'(\phi))$. This proves one direction of Lemma 2.

We will now prove the other direction of Lemma 2. Suppose that $\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}^{(n-K+1)}(\mathbf{x}'(\phi))$. We will prove that (C.6) holds by induction.

The first step is to prove the base case ($l = 1$). Suppose that

$$\underset{\mathcal{H}_1,\mathcal{H}_2\in\mathcal{C}^{(1)}(\mathbf{x}),\ \mathcal{H}_1\neq\mathcal{H}_2}{\arg\min} d(\mathcal{H}_1, \mathcal{H}_2; \mathbf{x}'(\phi)) \neq (\mathcal{G}_1^{(1)}, \mathcal{G}_2^{(1)}). \tag{C.7}$$

Since $\mathcal{C}^{(1)}(\mathbf{x}) = \mathcal{C}^{(1)}(\mathbf{x}'(\phi))$ by definition (Algorithm 5), and we assumed that there are no ties in the agglomerative hierarchical clustering procedure (Algorithm 5), there must exist $\mathcal{A}, \mathcal{B} \in \mathcal{C}^{(1)}(\mathbf{x}'(\phi))$ such that

$$d(\mathcal{A}, \mathcal{B}; \mathbf{x}'(\phi)) < d(\mathcal{G}_1^{(1)}(\mathbf{x}), \mathcal{G}_2^{(1)}(\mathbf{x}); \mathbf{x}'(\phi)). \tag{C.8}$$

Since the sequence of clusterings produced by $\{\mathcal{C}^{(l)}\}_{l=1}^n$ are nested, and $\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}^{(n-K+1)}(\mathbf{x})$ by definition, it must be the case that either $\mathcal{G}_1^{(1)}(\mathbf{x})$ and $\mathcal{G}_2^{(1)}(\mathbf{x})$ are both in $\hat{\mathcal{C}}_k$, are both

in $\hat{\mathcal{C}}_{k'}$, or are both in $\hat{\mathcal{C}}_k \cup \hat{\mathcal{C}}_{k'}$. Similarly, since $\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}^{(n-K+1}(\mathbf{x}'(\phi))$ by assumption, it must be the case that either $\mathcal{A}$ and $\mathcal{B}$ are both in $\hat{\mathcal{C}}_k$, are both in $\hat{\mathcal{C}}_{k'}$, or are both in $\hat{\mathcal{C}}_k \cup \hat{\mathcal{C}}_{k'}$. Thus, we can apply Lemma 1 to (C.8) to yield

$$d(\mathcal{A}, \mathcal{B}; \mathbf{x}) < d(\mathcal{G}_1^{(1)}(\mathbf{x}), \mathcal{G}_2^{(1)}(\mathbf{x}); \mathbf{x}),$$

which contradicts the definition of $(\mathcal{G}_1^{(1)}(\mathbf{x}), \mathcal{G}_2^{(1)}(\mathbf{x}))$ as the pair of clusters with the smallest dissimilarity (Algorithm 5). Thus, (C.7) cannot be the case, and it follows that

$$\underset{\mathcal{H}_1, \mathcal{H}_2 \in \mathcal{C}^{(1)}(\mathbf{x}),\ \mathcal{H}_1 \neq \mathcal{H}_2}{\arg\min} d(\mathcal{H}_1, \mathcal{H}_2; \mathbf{x}'(\phi)) = (\mathcal{G}_1^{(1)}(\mathbf{x}), \mathcal{G}_2^{(1)}(\mathbf{x})),$$

so we have proved the base case.

We now prove the inductive step. Suppose that for $m \in \{1, 2, \ldots, n-K-1\}$,

$$\underset{\mathcal{H}_1, \mathcal{H}_2 \in \mathcal{C}^{(l)}(\mathbf{x}),\ \mathcal{H}_1 \neq \mathcal{H}_2}{\arg\min} d(\mathcal{H}_1, \mathcal{H}_2; \mathbf{x}'(\phi)) = (\mathcal{G}_1^{(l)}(\mathbf{x}), \mathcal{G}_2^{(l)}(\mathbf{x})), \quad \forall\ l = 1, 2, \ldots, m. \tag{C.9}$$

Now, suppose that

$$\underset{\mathcal{H}_1, \mathcal{H}_2 \in \mathcal{C}^{(m+1)}(\mathbf{x}),\ \mathcal{H}_1 \neq \mathcal{H}_2}{\arg\min} d(\mathcal{H}_1, \mathcal{H}_2; \mathbf{x}'(\phi)) \neq (\mathcal{G}_1^{(m+1)}(\mathbf{x}), \mathcal{G}_2^{(m+1)}(\mathbf{x})). \tag{C.10}$$

Since (C.9) implies that $\mathcal{C}^{(m+1)}(\mathbf{x}) = \mathcal{C}^{(m+1)}(\mathbf{x}'(\phi))$, and we assumed that there are no ties in the agglomerative hierarchical clustering procedure, there must exist $\mathcal{A}, \mathcal{B} \in \mathcal{C}^{(m)}(\mathbf{x}'(\phi))$ such that

$$d(\mathcal{A}, \mathcal{B}; \mathbf{x}'(\phi)) < d(\mathcal{G}_1^{(m+1)}(\mathbf{x}), \mathcal{G}_2^{(m+1)}(\mathbf{x}); \mathbf{x}'(\phi)). \tag{C.11}$$

Since the sequence of clusterings produced by $\{\mathcal{C}^{(l)}\}_{l=1}^n$ are nested, and $\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}^{(n-K+1)}(\mathbf{x})$ by definition, it must be the case that either $\mathcal{G}_1^{(m+1)}(\mathbf{x})$ and $\mathcal{G}_2^{(m+1)}(\mathbf{x})$ are both in $\hat{\mathcal{C}}_k$, are both in $\hat{\mathcal{C}}_{k'}$, or are both in $\hat{\mathcal{C}}_k \cup \hat{\mathcal{C}}_{k'}$. Similarly, since $\hat{\mathcal{C}}_k, \hat{\mathcal{C}}_{k'} \in \mathcal{C}^{(n-K+1}(\mathbf{x}'(\phi))$ by assumption, it must be the case that either $\mathcal{A}$ and $\mathcal{B}$ are both in $\hat{\mathcal{C}}_k$, are both in $\hat{\mathcal{C}}_{k'}$, or are both in $\hat{\mathcal{C}}_k \cup \hat{\mathcal{C}}_{k'}$. Thus, we can apply Lemma 1 to (C.8) to yield

$$d(\mathcal{A}, \mathcal{B}; \mathbf{x}) < d(\mathcal{G}_1^{(m+1)}(\mathbf{x}), \mathcal{G}_2^{(m+1)}(\mathbf{x}); \mathbf{x}),$$

which contradicts the definition of $(\mathcal{G}_1^{(m+1)}(\mathbf{x}), \mathcal{G}_2^{(m+1)}(\mathbf{x}))$ as the pair of clusters with the smallest dissimilarity (Algorithm 5). Thus, (C.10) cannot be the case, and it follows that

$$\underset{\mathcal{H}_1, \mathcal{H}_2 \in \mathcal{C}^{(m+1)}(\mathbf{x}),\ \mathcal{H}_1 \neq \mathcal{H}_2}{\arg\min} d(\mathcal{H}_1, \mathcal{H}_2; \mathbf{x}'(\phi)) = (\mathcal{G}_1^{(m+1)}(\mathbf{x}), \mathcal{G}_2^{(m+1)}(\mathbf{x})), \tag{C.12}$$

so we have proved the inductive step.

C.3 Proof of Lemma 5

Recall from (4.24) that

$$\mathcal{S} = \bigcap_{l=1}^{n-K} \bigcap_{\substack{\mathcal{A},\mathcal{B}\in\mathcal{C}^{(l)}(\mathbf{x}),\mathcal{A}\neq\mathcal{B},\\ (\mathcal{A},\mathcal{B})\neq(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x}))}} \bigcap_{i\in\mathcal{A}} \bigcap_{j\in\mathcal{B}} \left\{\phi \geq 0 : d(\{i\},\{j\};\mathbf{x}'(\phi)) > d(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x});\mathbf{x})\right\}$$
$$\equiv \bigcap_{l=1}^{n-K} \mathcal{S}_{single}^{(l)},$$

where $\mathcal{C}^{(l)}(\mathbf{x})$ is the lth clustering in the sequence generated by applying agglomerative hierarchical clustering to $\mathbf{x}$, and $(\mathcal{G}_1^{(l)}(\mathbf{x}), \mathcal{G}_2^{(l)}(\mathbf{x}))$ is the pair of clusters merged in $\mathbf{x}$ at the lth step of the hierarchical clustering procedure (Algorithm 5. Recall Lemma 5 claims that $\mathcal{S} = \mathcal{S}_{single}^{(n-K)}$. To prove this claim, it suffices to show that $\mathcal{S}_{single}^{(n-K)} \subseteq \bigcap_{l=1}^{n-K} \mathcal{S}_{single}^{(l)}$, because $\bigcap_{l=1}^{n-K} \mathcal{S}_{single}^{(l)} \subseteq \mathcal{S}_{single}^{(n-K)}$.

Let $l \in \{1, 2, \ldots, n-K-1\}$. Suppose that $\phi \in \mathcal{S}_{single}^{(n-K)}$. Then, there exists $i, j \in \mathcal{C}^{(n-K)}(\mathbf{x})$ such that observations i and j are assigned to different clusters in $\mathcal{C}^{(n-K+1)}(\mathbf{x})$, and

$$d(\{i\},\{j\};\mathbf{x}'(\phi)) > d(\mathcal{G}_1^{(n-K)}(\mathbf{x}),\mathcal{G}_2^{(n-K)}(\mathbf{x});\mathbf{x}).$$

Since single-linkage cannot produce inversions, the heights at which clusters merge in the dendogram are ordered. Thus,

$$d(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x});\mathbf{x}) < d(\mathcal{G}_1^{(n-K)}(\mathbf{x}),\mathcal{G}_2^{(n-K)}(\mathbf{x});\mathbf{x}).$$

Thus,

$$d(\{i\},\{j\};\mathbf{x}'(\phi)) > d(\mathcal{G}_1^{(l)}(\mathbf{x}),\mathcal{G}_2^{(l)}(\mathbf{x});\mathbf{x}).$$

Furthermore, since $\{\mathcal{C}^{(l)}\}_{l=1}^{n}$ produces a nested sequence of clusterings, and $l < n-K$, the fact that i and j are assigned to different clusters in $\mathcal{C}^{(n-K+1)}(\mathbf{x})$ implies that i and j are assigned to different clusters in $\mathcal{C}^{(l+1)}(\mathbf{x})$. Thus, $\phi \in \mathcal{S}_{single}^{(l)}$ by definition.

Since we have shown that $\phi \in \mathcal{S}_{single}^{(n-K)}$ implies that $\phi \in \mathcal{S}_{single}^{(l)}$ for any $l \in \{1, 2, \ldots, n-K-1\}$, it follows that $\mathcal{S}_{single}^{(n-K)} \subseteq \bigcap_{l=1}^{n-K} \mathcal{S}_{single}^{(l)}$, which completes the proof.

C.4 Additional simulation results for Section 4.5.2

We return to the second simulation study described in Section 4.5.2, where we consider the power (i.e. the probability of rejecting $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$) of the test as a function of the effect size Δ, defined in (4.31). Recall that in order to smooth our estimates of power, we fit a regression spline using the `gam` function in the `R` package `mgcv`. We display the power of the test as a function of the effect size Δ when $\min\{|\hat{\mathcal{C}}_k|, |\hat{\mathcal{C}}_{k'}|\} < 10$ in Figure C.1.

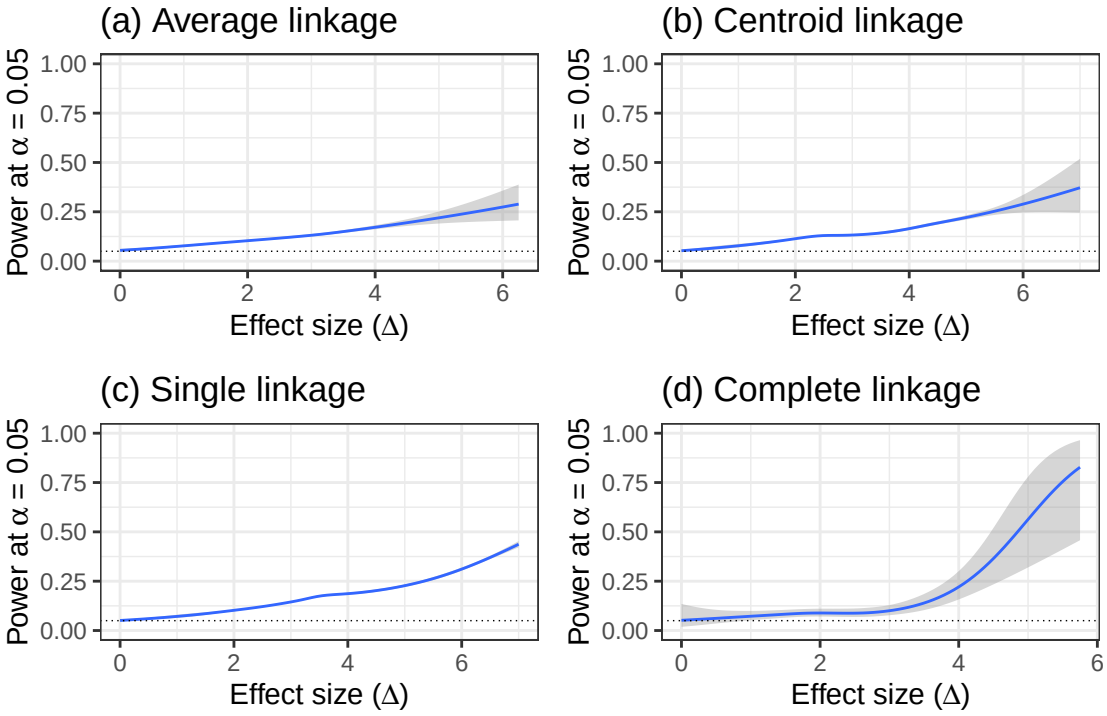

Figure C.1: For the simulated data sets such that $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ do not both have at least 10 observations in them in the simulation study described in Section 4.5.2, power as a function of effect size Δ, defined in (4.31) to be the true difference in means between $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$, scaled by the variance parameter σ, for the test proposed in Section 4.2.1, when $\mathcal{C}$ is the map that results from cutting the dendrogram resulting from (a) average-linkage, (b) centroid-linkage, (c) single-linkage, and (d) complete-linkage hierarchical clustering, to get three clusters.

As in Section 4.5.2, for all four linkages, the power to reject $H_0 : \bar{\mu}_{\hat{\mathcal{C}}_k} = \bar{\mu}_{\hat{\mathcal{C}}_{k'}}$ increases as the effect size Δ increases, where $\Delta = (\bar{\mu}_{\hat{\mathcal{C}}_k} - \bar{\mu}_{\hat{\mathcal{C}}_{k'}})/\sigma$. Again, this implies that our test

is more likely to detect a larger difference in means between estimated clusters $\hat{\mathcal{C}}_k$ and $\hat{\mathcal{C}}_{k'}$ than a smaller difference in means.

Furthermore, we find that the power of the test is quite similar for single, average, and centroid linkage. However, the power of the test for complete linkage is rather different. This is because single, average, and centroid linkage tend to produce extremely unbalanced clusters (e.g. $|\hat{\mathcal{C}}_k| = |\hat{\mathcal{C}}_{k'}| = 1$) when $\min\{|\hat{\mathcal{C}}_k|, |\hat{\mathcal{C}}_{k'}|\} < 10$, which leads to lower power, and complete linkage tends to produce more balanced clusters (e.g. $|\hat{\mathcal{C}}_k| = 9, |\hat{\mathcal{C}}_{k'}| = 8$) when $\min\{|\hat{\mathcal{C}}_k|, |\hat{\mathcal{C}}_{k'}|\} < 10$, which leads to higher power in Figure C.1.