Structured Mixture Models

by

Jason Hou-Liu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Statistics

Waterloo, Ontario, Canada, 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Steven Xiaogang Wang Professor, Department of Mathematics and Statistics, York University
Supervisor(s):	Ryan P. Browne Associate Professor, Department of Statistics and Actuarial Science, University of Waterloo
Internal Member:	Paul Marriott Professor, Department of Statistics and Actuarial Science, University of Waterloo
Internal Member:	Martin Lysy Associate Professor, Department of Statistics and Actuarial Science, University of Waterloo
Internal-External Member:	Yaoliang Yu Associate Professor, David R. Cheriton School of Computer Science, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The chapters of this thesis are the culmination of Jason Hou-Liu's work as supervised by Prof. Ryan P. Browne. This thesis consists of five manuscripts written for publication; all of which are currently published or have been submitted for publication at the corresponding journals.

- Chapter 3: Jason Hou-Liu and Ryan P. Browne. Chimeral clustering. *Journal of Classification*, 39(1):171–190, Mar 2022a. ISSN 1432-1343. doi: 10.1007/s00357-021-09396-3

  This manuscript is based on work done during Jason Hou-Liu's MMath degree at the University of Waterloo. Contributions during the PhD degree are the identifiability section and some refinements to the estimation procedure.

- Chapter 4: Jason Hou-Liu and Ryan P. Browne. Factor and hybrid components for model-based clustering. *Advances in Data Analysis and Classification*, 16(2):373–398, Jun 2022b. ISSN 1862-5355. doi: 10.1007/s11634-021-00483-2

- Chapter 5: Jason Hou-Liu and Ryan P. Browne. Model-based clustering with nested Gaussian clusters. *Revision submitted to Journal of Classification*, Jun 2023a

- Chapter 6: Jason Hou-Liu and Ryan P. Browne. Extrapolating conditional expectations to accelerate EM procedures. *Submitted to Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Mar 2023b

- Chapter 7: Jason Hou-Liu and Ryan P. Browne. Generalized linear models for massive data via doubly-sketching. *Statistics and Computing*, 33(5):105, Jul 2023c. ISSN 1573-1375. doi: 10.1007/s11222-023-10274-8

R software packages have been written by Jason Hou-Liu for these manuscripts, which will be available at `https://github.com/jhouliu`.

Abstract

Finite mixture models are a staple of model-based clustering approaches for distinguishing subgroups. A common mixture model is the finite Gaussian mixture model, whose degrees of freedom scales quadratically with increasing data dimension. Methods in the literature often tackle the degrees of freedom of the Gaussian mixture model by sharing parameters between the eigendecomposition of covariance matrices across all mixture components. We posit finite Gaussian mixture models with alternate forms of parameter sharing by imposing additional structure on the parameters, such as sharing parameters with other components as a convex combination of the corresponding parent components or by imposing a sequence of hierarchical clustering structure in orthogonal subspaces with common parameters across levels. Estimation procedures using the Expectation-Maximization (EM) algorithm are derived throughout, with application to simulated and real-world datasets. As well, the proposed model structures have an interpretable meaning that can shed light on clustering analyses performed by practitioners in the context of their data.

The EM algorithm is a popular estimation method for tackling issues of latent data, such as in finite mixture models where component memberships are often latent. One aspect of the EM algorithm that hampers estimation is a slow rate of convergence, which affects the estimation of finite Gaussian mixture models. To explore avenues of improvement, we explore the extrapolation of the sequence of conditional expectations admitting general EM procedures, with minimal modifications for many common models. With the same mindset of accelerating iterative algorithms, we also examine the use of approximate sketching methods in estimating generalized linear models via iteratively re-weighted least squares, with emphasis on practical data infrastructure constraints. We propose a sketching method that controls for both data transfer and computation costs, the former of which is often overlooked in asymptotic complexity analyses, and are able to achieve an approximate result in much faster wall-clock time compared to the exact solution on real-world hardware, and can estimate standard errors in addition to point estimates.

Acknowledgements

I am deeply grateful to my supervisor, Prof. Ryan P. Browne, for the amazing amount of care, attention, and support shown throughout my graduate studies. Your flexibility allowed my research endeavours to take on my own style, and your unwavering guidance and wisdom has given the same the substance needed to shine in the academic sea of knowledge. I have learned much from you and am honoured to have had the opportunity to be your student.

To my parents, I thank you for building the discipline and foundation of knowledge I stand on every day. You taught me to do the best I can do, and to make the most of every day. I appreciate all your wisdom and experience in all matters of life, and will be forever grateful for everything you have done.

To my special one, you stood by me through thick and thin, and I appreciate your warmth and exuberance. Thank you for your love and support, and reminding me of the variety in life.

To my fellow colleagues in both academia and industry, I would also like to express my sincere gratitude for your camaraderie; I enjoyed our escapades wherever they took us.

For everyone else whose paths I crossed, I would also like to express my thanks for helping shape who I am today.

Dedication

To my parents, for showing me the world and all its wonders.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis comprises five works, with the first three describing finite Gaussian mixture models with structured parameters and the latter two describing accelerations of existing algorithms. In the first three works, the theme of adding structure on top of finite Gaussian mixture models can be seen from two perspectives; we may either specify a model with a more parsimonious representation or share parameters between mixture model components. By positing a specific relationship between components of the mixture model, we may be able to find a more efficient representation of datasets whose structure is not captured by conventional specifications. Moreover, these structures can represent relationships in the intercluster sense that can be interpreted in the context of the data, providing additional value to analyses performed by practitioners.

We work with the finite Gaussian mixture model framework, often used in the model-based clustering of multivariate data, where traditional methods of parameter sharing and representations of intercluster structure exploit geometric redundancies such as volume, shape, and orientation in the covariance matrix (Celeux and Govaert, 1995; McNicholas and Murphy, 2008). Sharing parameters between multivariate normal distributions in other manners can be found in mixed and partial membership models (Airoldi et al., 2014). This family of models assigns observations to multiple clusters, possibly in varying degrees, and can implicitly define hybrid cluster components that are dependent on other cluster

parameters. We draw upon a motivating example from biology whereby hybrid species tend to exhibit a mixture of parent characteristics. As well, we extend this example in that species are often members of a larger overarching genus containing many other species. In this context, there are shared attributes that define the genus but idiosyncratic attributes distinguish species. When a dataset spans multiple genera, we posit in the a finite Gaussian mixture model framework that a subspace of the data captures features of a genus, and another orthogonal subspace captures within-genus variation of the constitutent species.

A popular method for fitting finite Gaussian mixture models to data where the multivariate normal distribution parameters of the components are unknown is the Expectation-Maximization (EM) algorithm (Dempster et al., 1977). This procedure is often used to estimate the unknown parameters of a finite Gaussian mixture model. When objective functions such as the Gaussian mixture model log-likelihood is intractable and not amenable to direct maximization, then the introduction of latent variables representing unobserved cluster memberships allows the application of the EM algorithm. As the EM algorithm is an iterative procedure that alternates between computing the conditional expectation of the latent data and the maximization/majorization in the model parameters, the estimation can be slow to converge which hampers estimation.

Other causes of slow estimation include very large dataset sizes, which can render algorithms typically considered fast such as Newton-Raphson too slow for practical usage. As an exploratory foray into alternate methods of accelerating algorithms, we also explore finding approximate solutions for generalized linear models (GLM) with massive datasets by applying randomized sketching (Ahfock et al., 2022) in the context of real-world data infrastructure and computer systems. Here, the backbone of GLM parameter estimation is the Iteratively Re-weighted Least Squares algorithm, to which quadratic convergence is often ascribed.

In Chapter 2, we provide a brief overview of a collection of common concepts used throughout this thesis; more specific concepts pertinent to each chapter are introduced in the corresponding sections within the chapter. In Chapter 3, we introduce Chimeral Clustering; a model which captures cluster hybridization in the form of chimeral components

whose distributions parameters are convex combinations of prototype distribution parameters. In Chapter 4, we introduce Factor-Hybrid Clustering, an altenate take on the ideas of Chimeral Clustering by using the more practitioner-friendly moment parameterization instead of the mathematically convenient canonical parameterization. In Chapter 5, we introduce Nested Gaussian Clusters to capture a hierarchy of nested class labels. In Chapter 6, we introduce a method of accelerating the EM algorithm by extrapolating in the conditional expectation of the missing data, with application to finite Gaussian mixture models. Finally, in Chapter 7 we introduce a randomized sketching method to estimate GLM parameters with massive datasets on a range of computational infrastructures.

Chapter 2

Background

We describe finite mixture models in Section 2.1 to set the stage for the literature. We narrow our focus to exemplar models using multivariate normal distributions for components, a family of models that flexibly models multivariate data. We describe this distribution in two common parameterisations in Section 2.2. We then consider some mixture models, and discuss the number of parameters and the curse of dimensionality in with mitigation strategies in Section 2.2.1. The central estimation method is the Expectation-Maximization algorithm (Dempster et al., 1977), overviewed in Section 2.3. Generalized linear models and the Iteratively Re-weighted Least Squares algorithm are described in Section 2.5.

2.1 Finite Mixture Models

Consider a population distribution consisting of multiple sub-populations, each of which exhibits different behaviours in some common set of variables. For example, among a population of light-bulbs with three sub-populations of incandescent, fluorescent, and LED, we might expect their power consumption to be typically 60 W, 12 W, and 4 W, respectively. From a distributional perspective, we have a population distribution that is hierarchical. Firstly, a categorical variable represents the sub-population to which a unit belongs. Sec-

ondly, conditional on being in a sub-population, the unit's variables follows a distribution specific to that sub-population.

Notationally, let there be a population with $K \geq 1$ such sub-populations. For a unit in sub-population $k = 1, 2, ..., K$ with probability $\pi_k > 0$ such that $\sum_{k=1}^{K} \pi_k = 1$, let the variates follow a distribution with density f_k and parameters $\boldsymbol{\theta}_k$. Then, the distribution of the whole population is

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}; \boldsymbol{\theta}_k). \tag{2.1}$$

Different choices of component density f_k yield different families of mixture models.

2.2 Multivariate Normal Distribution

A common choice for the sub-population component density is the multivariate normal distribution. The d-multivariate normal distribution generalizes the univariate normal distribution to d dimensions and can be parameterised by central moment parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ or natural parameters $(\boldsymbol{\eta}, \boldsymbol{\Lambda})$, for $\boldsymbol{\Sigma}$ or $\boldsymbol{\Lambda}$ positive-definite, with one-to-one correspondance $\boldsymbol{\eta} = -\frac{1}{2}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ and $\boldsymbol{\Lambda} = -\frac{1}{2}\boldsymbol{\Sigma}^{-1}$. Without loss of generality, we omit the constant factor by referring to $\boldsymbol{\eta} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ and $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ equivocally as the natural parameterization. The multivariate normal log-density function with moment parameters can be written as

$$\log \phi(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{d}{2} \log 2\pi - \frac{1}{2} \log \det \boldsymbol{\Sigma} - \frac{1}{2} \operatorname{Tr}\left[\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^{\top}\right],$$

and in canonical parameters as

$$\log \phi(\boldsymbol{x}; \boldsymbol{\eta}, \boldsymbol{\Lambda}) = -\frac{d}{2} \log 2\pi + \frac{1}{2} \log \det \boldsymbol{\Lambda} - \frac{1}{2}\boldsymbol{\eta}^{\top}\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta} + \boldsymbol{\eta}^{\top}\boldsymbol{x} - \frac{1}{2} \operatorname{Tr}(\boldsymbol{\Lambda}\boldsymbol{x}\boldsymbol{x}^{\top}),$$

where $\operatorname{Tr}(\mathbf{X})$ is the trace of a square matrix $\mathbf{X}$; i.e., the sum of the diagonal elements of $\mathbf{X}$. In canonical form, the log density is linear in parameters $\boldsymbol{\eta}$ and $\boldsymbol{\Lambda}$ and has a convex log-partition function.

Mixture models with multivariate normal component distributions are also referred to as finite Gaussian mixture models (Banfield and Raftery, 1993; Symons, 1981; Wolfe, 1963), and are useful for their flexibility in representing multivariate data and their mathematical properties. A theoretical guarantee for finite Gaussian mixtures is that they are identifiable (Teicher, 1961; Yakowitz and Spragins, 1968; Holzmann et al., 2006).

2.2.1 Curse of Dimensionality

In both the $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $(\boldsymbol{\eta}, \boldsymbol{\Lambda})$ parameterisations of the multivariate normal there are $d + d(d+1)/2$ free parameters. This increases quadratically with dimension d, leading to estimation difficulties and a lack of parsimony. Without further restrictions, a K-component mixture model with multivariate normal components has K times the number of parameters, compounding upon the issue.

One way of reducing the effective dimensionality of finite Gaussian mixtures is exploiting redundancies in geometry across multiple covariance matrices $\boldsymbol{\Sigma}$, such as a common shape, orientation, and/or size (Celeux and Govaert, 1995; Fraley and Raftery, 2002; McNicholas and Murphy, 2008; Browne and McNicholas, 2014).

2.3 Expectation-Maximization Algorithm

The EM algorithm (Dempster et al., 1977) is a method for performing optimization on an objective function, often a log-likelihood, with observed data $\mathbf{X}$ and missing data $\mathbf{Z}$. In such a case, taking a conditional expectation of the complete-data log-likelihood $\ell_{\text{com}}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})$ with respect to missing data $\mathbf{Z}$ can yield a tractably maximizable/majorizable surrogate function. Theoretical results indicate that an improvement in the value of the surrogate function yield guarantee an increase in the original log-likelihood of interest, which permits maximum likelihood estimation of complicated objective functions such as (2.1). Variations on the EM algorithm exist (Dempster et al., 1977; Celeux and Diebolt, 1986) that yield different behaviours such as faster estimation or escaping local modes.

As a general procedure, for some complete-data log-likelihood, the surrogate function as of iteration t can be written as

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) = \mathrm{E}\left[\ell_{\mathrm{com}}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) \mid \mathbf{X}; \boldsymbol{\theta}^{(t)}\right]$$

whose maximizer (or majorizer) $\boldsymbol{\theta}^*$ becomes the parameter $\boldsymbol{\theta}^{(t+1)}$ for the subsequent iteration. The construction of the surrogate function is considered the expectation step due to the conditional expectation in latent data $\mathbf{Z}$, and finding the maximizer is considered the maximization step. By iteratively alternating between these two steps, the EM procedure is able to attain monotonic ascent in the objective function.

2.4 Model Metrics

In the context of estimating finite mixture model parameters, a common problem is the determination of the number of clustering components K applicable to the data. In the absence of a priori information on K, a practical approach is to perform model selection using a penalized goodness-of-fit metric such as the Bayesian Information Criterion (BIC) (Schwarz, 1978). This metric is defined as

$$\mathrm{BIC} = k \log N - 2\ell(\boldsymbol{\theta}; \mathbf{X}),$$

where k is the number of free parameters of the model, N is the number of observations, and ℓ is the observed log-likelihood for model parameters $\boldsymbol{\theta}$. This parameterization of BIC implies lower values are considered better as excessive parameters yield an increase in this quantity while improved observed log-likelihood yields a decrease.

In addition to quantifying the goodness-of-fit, we may also measure the degree of concordance between the fitted cluster indices and a set of available class labels by evaluating the adjusted Rand index (ARI) metric of Hubert and Arabie (1985). This quantity is defined as

$$\mathrm{ARI} = \frac{\text{Rand Index} - \text{Expected Rand Index}}{\text{Maximum Rand Index} - \text{Expected Rand Index}}$$

which can be expanded into the form

$$\text{ARI} = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \sum_i \binom{n_{i\cdot}}{2}\sum_j \binom{n_{\cdot j}}{2}/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_{i\cdot}}{2} + \sum_j \binom{n_{\cdot j}}{2}\right] - \sum_i \binom{n_{i\cdot}}{2}\sum_j \binom{n_{\cdot j}}{2}/\binom{n}{2}}$$

for n_{ij} being the cross-tabulation between class i of the first classification and class j of the second classification, $n_{i\cdot}$ and $n_{\cdot j}$ are the corresponding marginal counts, and n is the overall total count.

2.5 Generalized Linear Models

Generalized linear models are a popular regression model for non-normally distributed response variables that are often found in practice, such as binary outcomes or count data (McCullagh and Nelder, 1989). These models typically specify the distribution of the observed response $Y_i = y_i$ for $i = 1, 2, ..., n$ as having a regular exponential family distribution with probability mass/density function of the form

$$f(y_i; \theta_i, \phi) = \exp\left[\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y; \phi)\right],$$

where a, b, and c are known functions, θ_i is the canonical parameter, and ϕ are any nuisance parameters assumed known. The canonical is a function of the linear predictor $\eta_i = \boldsymbol{x}_i^\top \boldsymbol{\beta}$ for covariate vector $\boldsymbol{x}_i \in \mathbb{R}^d$. GLMs often do not admit a closed-form solution for the regression coefficients $\boldsymbol{\beta}$ nor their standard errors; a desirable value for hypothesis testing. A common iterative method is that of Iteratively Reweighted Least Squares (IRLS), which relies on both first- and second-order derivative information of the objective log-likelihood to find a fixed-point. The IRLS update has asymptotic complexity $O(nd^2)$. In massive data settings, not only does the complexity of a dataset with high-dimension d have an impact on wall-clock estimation times, the number of observations n can be large enough to make IRLS too slow for practical usage.

Chapter 3

Chimeral Clustering

3.1 Introduction

In Greek mythology, chimeras are beings with the head of a lion, the body of a goat, and the tail of a snake. In biological studies, we may find genotypes and phenotypes of hybrid species are expressed in a similarly intermediate way. A well-recognized example of this is the *iris* dataset (Anderson, 1936; Fisher, 1936) whereby *Iris setosa* and *Iris virginica* hybridize to form *Iris versicolor* with intermediate sepal and petal dimensions. In a clustering context, we may find some clusters to exhibit parameters that are a mixture of those of other clusters. Two extant methods capture this notion of chimerality; Heller et al. (2008) describes a Bayesian method of assigning partial memberships to multiple clusters and Zhang (2013) describes an epistatic clustering method whereby observations may be assigned to more than one cluster. We consider these two methods to be on opposing sides of the spectrum in terms of flexibility; the former grants a large quantity of freedom, whereas the latter provides a rigid structure in which data must fit. To advance a middle ground, we introduce the chimeral clustering model as an extension of finite Gaussian mixture models by parameterizing chimeral clusters using convex combinations of non-chimeral (prototype) cluster parameters.

The present work postulates the chimeral clustering model and provides a theoretical

treatment of identifiability of chimeral clustering. We prove a sufficient condition for identifiability and extend the reasoning to show that epistatic clusters are not identifiable to the extent specified by Zhang (2013). We provide an estimation procedure based on the expectation-maximization algorithm (Dempster et al., 1977) and demonstrate its efficacy on multiple datasets with comparison to existing parsimonious Gaussian mixture models. Our evaluation datasets are the well-known *iris* dataset (Anderson, 1936; Fisher, 1936), a morphometric dataset describing species of hawks(Cannon et al., 2019), and a morphometric dataset describing water striders from the *limnoperus* genus (Klingenberg and Spence, 1993). Additionally, we craft a dataset which demonstrates the difference in style of redundancy not captured by parsimonious covariance matrices in finite Gaussian mixtures. Finally, we assess goodness of fit and parsimony using the Bayesian information criterion (Schwarz, 1978) and the adjusted Rand index (Rand, 1971; Hubert and Arabie, 1985).

3.1.1 Intercluster Structure

The idea of an individual being a hybrid of multiple sub-populations can be found in biological literature. In other words, these hybrids can be described as admixtures of other sub-populations. An example by Anderson (1936) demonstrates the hybridisation in the genus *iris* by interpolating between geometric dimensions of the flowers. Battle et al. (2005) and Pritchard et al. (2000) describe this kind in the distributions of genotype data. These approaches using genotypes consider individuals as potentially being an admixture of many groups instead simply a single group.

Moving from domain descriptions to model specifications that capture this sort of intercluster structure, we consider the mixed membership or partial membership models (Airoldi et al., 2014). The distinction between mixed and partial membership models can be resolved with the appropriate introduction of latent variables (Erosheva et al., 2007). These models permit individual observations to belong to multiple clusters simultaneously either by being assigned to an intermediate cluster with weighted combinations of parameters or assigned fractionally to multiple clusters.

Models falling under this umbrella include the pioneering Grade of Membership (Wood-

bury et al., 1978), Bayesian partial membership (Heller et al., 2008), and epistatic clustering (Zhang, 2013). When observed variables are categorical in nature, a viable option for modelling individual observations as a mixture of a collection of pure types is the Grade of Membership (GOM) model. This model is based on a clinical need to model patients as having degrees of multiple illnesses, and attributes them fractionally to these characteristic subpopulations using a question relevance factor. The GOM model behaves in a manner similar to fuzzy sets and permits membership labels to take values in $[0, 1]$. In GOM, each observation n is granted an intergrade representation as non-negative weights $g_{n1}, g_{n2}, ..., g_{nK}$ with unit sum. These weights determine the probability $p_{ij(l)}$ of manifesting an outcome associated with a pure type $1, 2, ..., K$ in each categorical variable. Pure types are characterised by a set of unconditional probabilities λ_{kjl} of manifesting outcomes in each variable. There are also extensions of the GOM model to rank data in place of categorical data (Gormley and Murphy, 2009).

A distant relationship to this concept is that of Latent Dirichlet Allocation Blei et al. (2003) with documents being a mixture of topics. This model is appropriate for count data or frequency data such as the number of occurrences of a word or event. Observations of text documents are the tabulation of word occurrences, which in turn are considered manifestations of a mixture of topics. In the LDA model, a topic distribution θ is drawn from a prior Dirichlet(α). From this distribution, a specific topic z_n is drawn according to Multinomial(θ). Finally, a word is drawn given that it is from topic z_n according to yet another multinomial probability distribution. In this sense, for a fixed dictionary of words, the observations are also categorical; however, this dictionary can be chosen to encompass all observed words over the entire document corpus.

Another distantly related model applicable to networks and their interactions is the Mixed Membership Stochastic Blockmodel (MMSB) (Airoldi et al., 2008), which describes a network over time as a mixture of roles; an individual in the network may act in multiple distinct capacities.

From the model-based clustering methods that apply to continuous variates as in finite Gaussian mixtures, we draw inspiration from two specific models: epistatic clustering

Zhang (2013) and Bayesian partial membership Heller et al. (2008). These are discussed in more detail in Section 3.1.1 and Section 3.1.1, respectively.

Epistatic Clustering

Epistatic clustering supposes a finite Gaussian mixture model to be decomposed into two types of clusters; a set of parent clusters and a set of epistatic clusters. Each parent component p follows the multivariate normal distribution, specified either in moment parameters as $\mathrm{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ or canonical form as $\mathrm{N}(\boldsymbol{\eta}_p, \boldsymbol{\Lambda}_p)$. Epistatic components are also multivariate normal; however, they parameters that are the average of parent component parameters.

In moment parameters, an epistatic cluster's mean and variances are equal to the arithmetic average of a subset of matrix-weighted parent means and weighted matrix-harmonic parent covariances. In canonical form, it is the simple arithmetic average of parent parameters. Specifically, if an epistatic cluster has parents denoted by index set $\mathcal{P}$, then

$$\begin{aligned} \boldsymbol{\mu}_{\text{epistatic}} &= \boldsymbol{\Sigma}_{\text{epistatic}} \textstyle\sum_{p\in\mathcal{P}} \boldsymbol{\Sigma}_p^{-1}\boldsymbol{\mu}_p \\ \boldsymbol{\Sigma}_{\text{epistatic}} &= \left(\tfrac{1}{|\mathcal{P}|}\textstyle\sum_{p\in\mathcal{P}} \boldsymbol{\Sigma}_p^{-1}\right)^{-1} \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \boldsymbol{\eta}_{\text{epistatic}} &= \tfrac{1}{|\mathcal{P}|}\textstyle\sum_{p\in\mathcal{P}} \boldsymbol{\eta}_p \\ \boldsymbol{\Lambda}_{\text{epistatic}} &= \tfrac{1}{|\mathcal{P}|}\textstyle\sum_{p\in\mathcal{P}} \boldsymbol{\Lambda}_p \end{aligned}$$

Otherwise, non-epistatic clusters have the same density and parameters as a multivariate normal distribution. This means that epistatic clusters no longer have any effectively free parameters; the model includes all combinations of epistatic clusters up to a pre-specified degree. For example, degree 2 represents all two-parent epistatic clusters, degree 3 all three-parent, and so forth. This results in a reduction in the number of parameters in the model, in addition to potentially using parsimonious parent covariance matrices as in Fraley and Raftery (2002).

Bayesian Partial Membership

Heller et al. (2008) gives a Bayesian Partial Membership framework where each observation has membership weights drawn from a Dirichlet distribution. Here, like in the GOM, LDA, and MMSB models, the Bayesian Partial Membership (BPM) model describes each

observation as a weighted parameter combination of a set of exponential family distribution clusters, effectively allowing an observation to be part of a hybrid distribution.

Like the GOM, LDA, and MMSB models, the BPM model permits each individual observation to have idiosyncratic weightings.

When applied in the context of multivariate normal distributions, the canonical parameterisation requirement of the BPM model is in terms of a precision matrix and a matrix-weighted mean.

3.1.2 Motivation

We motivate our model using the the well-known *iris* dataset of Anderson (1936) and Fisher (1936), which describes three related species of flower from the genus *iris*. Inspecting the plots in Figure 3.1 reveals a distinct intercluster structure; the intermediate cluster of *iris versicolor* is positioned in between the clusters for *iris setosa* and *iris virginica*. The shapes of each species' data points is also approximately multivariate normal, and the shape of the putative hybrid *iris versicolor* could also be described as an interpolation of the shapes of the parent species. Indeed, Anderson (1936) provides an argument for a 2:1 ratio hybridisation based on their morphological dimensions and chromosomal analysis. As such, we posit a hierarchical Gaussian mixture model whereby some clusters have means (locations) and covariances (shapes) derived from other clusters.

In relation to the existing EC and BPM models, we consider them to be two opposing extremes in treating hybridization. The former imposes epistatic cluster parameters as a fixed average of interacting primary clusters; there is no degree to which the clusters interact, only that they do. By contrast, the latter's full uncountable continuum of possible weights between prototypes renders mixing coefficients difficult to interpret meaningfully.

We distinguish this model from both mixed and partial membership models by not assigning each observation its own weightings but rather shifting the mixing into the cluster level. We pose the model under the mixture model framework by allocating observations to a single cluster; however, the cluster may be a prototype or chimeral cluster. The key

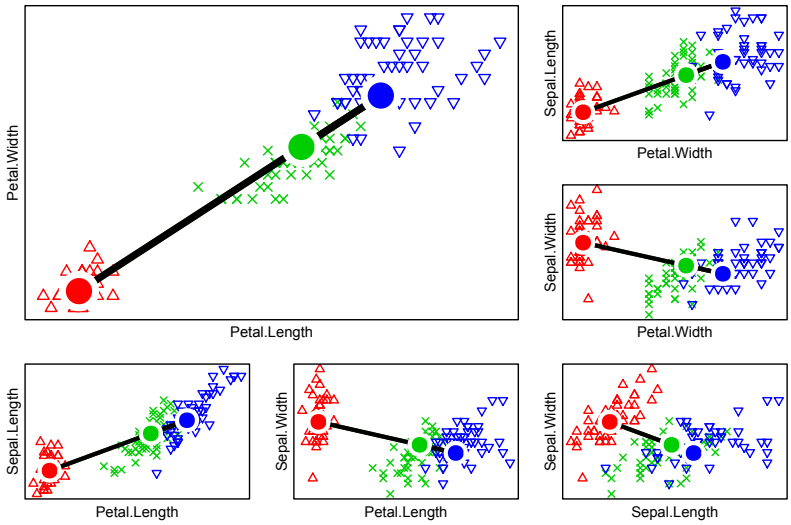

Figure 3.1: A collection of pair-wise scatterplots of the *iris* dataset. Labels and cluster means are from the factor/hybrid model in Chapter 4. Note the intermediacy of the central green cluster, primarily corresponding to *iris versicolor*.

difference is that a prototype cluster has the full flexibility of the multivariate normal distribution while a chimeral cluster is parameterized solely by a set of mixing proportions relative to the prototypes. A detailed comparison against the proposed model is made in Section 3.2.

The key distinction here is that the weighting occurs at the component-level instead of the observation-level. Like the BPM model, the EC model acts in the canonical parameterisation of the multivariate normal. Unlike the BPM model, the mixing weights for epistatic components are restricted to being averages; for example, weights of $\frac{1}{2}$ for a two-parent epistatic component, $\frac{1}{3}$ for a three-parent, and so forth. The BPM and EC models could be considered extensions to Gaussian mixture models (Wolfe, 1963; Symons, 1981; Banfield and Raftery, 1993) that use mixing parameters to describe an hybrid/epistatic cluster instead of a complete moment $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ or canonical $\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}\right)$ parameterisation.

A different approach to reducing the number of model parameters in finite Gaussian mixtures are to use parameter sharing via parsimonious covariance matrices that exploit

geometric shape, orientation, and volume redundancies (Celeux and Govaert, 1993; Fraley, 1998; McNicholas and Murphy, 2008; Browne and McNicholas, 2014). Compared to BPM and EC, these methods do not account for parameters shared by the means of the components.

3.2 Model Specification

With chimeral clustering, we use the canonical parameters of the multivariate normal distribution as described in Section 2.2. Herein, we denote the multivariate normal distribution as $\mathrm{N}(\boldsymbol{\eta}, \boldsymbol{\Lambda})$. The chimeral clustering model takes a middle ground with respect to the aforementioned epistatic clustering (Zhang, 2013) and Bayesian partial membership (Heller et al., 2008) works by allowing for a finite number of chimeral clusters with parameters defined by a convex combination of prototype clusters.

We propose a K-component finite Gaussian mixture with an alternate parameterisation for some components. That is, we partition the K components into a prototype set $\mathcal{P}$ and a chimeral set $\mathcal{C}$ with sizes K_P and K_C, respectively. The model density is given by

$$\begin{aligned} f(\boldsymbol{x} \mid \boldsymbol{\theta}) &= \sum_{k=1}^{K} \pi_k \phi_d(\boldsymbol{x} \mid \boldsymbol{\eta}_k, \boldsymbol{\Lambda}_k) \\ &= \sum_{p \in \mathcal{P}} \pi_p \phi_d(\boldsymbol{x} \mid \boldsymbol{\eta}_p, \boldsymbol{\Lambda}_p) + \sum_{c \in \mathcal{C}} \pi_c \phi_d(\boldsymbol{x} \mid \boldsymbol{\eta}_c, \boldsymbol{\Lambda}_c) \\ &= \sum_{p \in \mathcal{P}} \pi_p \phi_d(\boldsymbol{x} \mid \boldsymbol{\eta}_p, \boldsymbol{\Lambda}_p) + \sum_{c \in \mathcal{C}} \pi_c \phi_d \left(\boldsymbol{x} \middle| \boldsymbol{\eta}_c = \sum_{p \in \mathcal{P}} \alpha_{cp} \boldsymbol{\eta}_p, \boldsymbol{\Lambda}_c = \sum_{p \in \mathcal{P}} \alpha_{cp} \boldsymbol{\Lambda}_p \right) \end{aligned} \tag{3.1}$$

where a prototype cluster p follows a $\mathrm{N}(\boldsymbol{\eta}_p, \boldsymbol{\Lambda}_p)$ distribution with natural parameters $\boldsymbol{\eta}_p$ and $\boldsymbol{\Lambda}_p$ and a chimeral cluster c is instead parameterized by a vector of mixing coefficients $\boldsymbol{\alpha}_c$ in the standard K_P-simplex, implying a $\mathrm{N}(\boldsymbol{\eta}_c, \boldsymbol{\Lambda}_c)$ distribution with $\boldsymbol{\eta}_c = \sum_{p \in \mathcal{P}} \alpha_{cp} \boldsymbol{\eta}_p$ and $\boldsymbol{\Lambda}_c = \sum_{p \in \mathcal{P}} \alpha_{cp} \boldsymbol{\Lambda}_p$. As positive-definite matrices are closed under convex combinations, $\boldsymbol{\Lambda}_c$ is positive-definite. Figure 3.2 illustrates the mixing procedure for two prototype clusters and a single chimeral cluster in the natural parameter space. Furthermore, we require that

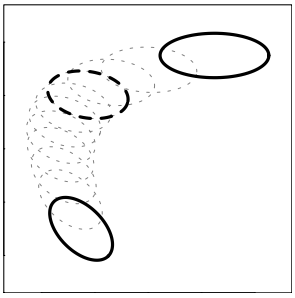

Figure 3.2: Two prototypes (solid) mixing in 10% increments (grey, dotted) and a particular 30%/70% mix (dashed). This figure resembles that of Heller et al. (2008); however, the proposed model contains an explicit and finite number of realizations of this continuum.

a cluster must be chimeral if it can be chimeral; this prevents chimeral clusters from being formed from convex combinations of other chimeral clusters.

By comparison, in epistatic clustering an observation can be assigned to multiple clusters, whose multivariate normal distribution is parameterized according to a weighted sum of natural parameters (Zhang, 2013, Eq. 2.2). Under the above presented framework, we may understand epistatic clustering as restricting $\boldsymbol{\alpha}_c$ to a fixed set of possibilities that average a subset of parameters such as $\langle \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \rangle$ or $\langle \frac{1}{2}, \frac{1}{2}, 0 \rangle$.

In Bayesian partial membership, Heller et al. (2008) propose a Dirichlet distribution for the weights $\boldsymbol{\pi}_{nk}$ over the K-simplex. As a similarity, this model can be understood by replacing the Dirichlet distribution for the weights $\boldsymbol{\pi}_n$ with a draw from a finite set of possible vectors $\{\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_{K_C}\}$ contained within the K-simplex. Indeed, Figure 3.2 parallels a related concept, though we only permit a finite number of such interpolated distributions and grant them an explicit realization to which observations may be assigned. As a difference, the Dirichlet distribution remains fixed but the set of vectors is dynamic in the chimeral clustering estimation procedure; this set itself will vary over the K_P-simplex.

As well, the distribution of a chimeral component coincides with the weighted Kullback-Leibler average (KLA) of the prototype component distributions (Battle et al., 2005). Indeed, the KLA for a collection $\mathcal{P}$ of multivariate normal distributions weighted by $\boldsymbol{\alpha}_c$ yields the same convex combination $\boldsymbol{\eta}_c$ and $\boldsymbol{\Lambda}_c$ as for a chimeral component with the same

hybridization weights.

3.2.1 Identifiability

A finite Gaussian mixture has been shown to be identifiable (Teicher, 1961; Yakowitz and Spragins, 1968; Holzmann et al., 2006) up to a permutation of indices. Suppose a K-component finite Gaussian mixture is parameterized by $\{\boldsymbol{\pi}, \boldsymbol{\eta}_1, ..., \boldsymbol{\eta}_K, \boldsymbol{\Lambda}_1, ..., \boldsymbol{\Lambda}_K\}$. We show that there exists a unique chimeral clustering reparameterisation of the model. We first assign indices $\mathcal{P} = \{P_1, ..., P_{K_P}\}$ and $\mathcal{C} = \{C_1, ..., C_{K_C}\}$ such that $\mathcal{P} \cup \mathcal{C} = \{1, 2, ..., K\}$. We also make a one-to-one representation of each cluster's parameters using $\boldsymbol{v}_k = \langle \boldsymbol{\eta}_k, \operatorname{vech} \boldsymbol{\Lambda}_k \rangle$. We use the terms parameters and vertices interchangeably using this relation. Then, we require some definitions and results from convex geometry.

Definition 1 *Let $\mathbb{V}$ be a vector space. A set $S \subset \mathbb{V}$ is convex if $x, y \in S$ implies $\lambda x + (1 - \lambda) y \in S$ for all $\lambda \in [0, 1]$.*

Definition 2 *For a set $V \subseteq \mathbb{V}$, the convex hull $S = \operatorname{conv}(V)$ is the smallest convex set containing V. If V is a finite set of vertices, then S is a convex polytope.*

Definition 3 *For a convex polytope $S \subseteq \mathbb{V}$, the extremal vertices $\operatorname{ext}(S)$ are points of S which do not lie in any open line segment joining two distinct points in S. The complement of $\operatorname{ext}(S)$ within S is the relative interior $\operatorname{relint}(S)$.*

There are two standard and equivalent (Grünbaum, 2003) ways to represent a convex polytope S. The $\mathcal{V}$-representation defines S as a convex combination of $\operatorname{ext}(S)$. The $\mathcal{H}$-representation defines S as the finite intersection of half-spaces forming facets of the polytope S.

Lemma 1 *The convex hull $\operatorname{conv}(V)$ of a finite set of vertices $V \subset \mathbb{R}^d$ (i) forms a convex polytope, (ii) is unique, and (iii) can be characterized by a set of extremal vertices $\operatorname{ext}(S)$ which are a unique subset of V.*

Proof 1 *(i) From Section 3.1 and 3.6 of Grünbaum (2003), the convex hull* $\text{conv}(V)$ *is a polytope whose* $\mathcal{V}$*-representation is a set of vertices. A similar result can be found in Section 14.1 of Gruber (2007).*

(ii) Uniqueness of $\text{conv}(V)$ *is given by the fact that intersection of convex sets is convex. If* A *and* B *are two different smallest convex hulls of* V *then* $A \cap B$ *contains* V *and so is a smaller convex hull; a contradiction.*

(iii) Finally, Section 14.1 of Gruber (2007) provides us with the result that the extreme points of $\text{conv}(V)$ *are a subset of the vertices in* V*.*

Theorem 1 *A chimeral clustering model given by (3.1) with parameters* $\{\boldsymbol{\pi}, \boldsymbol{\eta}_{P_1}, ..., \boldsymbol{\eta}_{P_{K_P}}, \boldsymbol{\Lambda}_{P_1}, ..., \boldsymbol{\Lambda}_{P_{K_P}}, \boldsymbol{\alpha}_{C_1}, ..., \boldsymbol{\alpha}_{C_{K_C}}\}$ *is identifiable provided there is a sufficient condition for identifiability of each* $\boldsymbol{\alpha}_c$*, such as minimizing* $\boldsymbol{\alpha}_c^\top \boldsymbol{\alpha}_c$*.*

Proof

As a finite Gaussian mixture, a chimeral clustering model is identifiable as normal distribution parameters $(\boldsymbol{\eta}_k, \boldsymbol{\Lambda}_k)$ for $k = 1, 2, ..., K$ and marginal probabilities $\boldsymbol{\pi}$.

Let $\boldsymbol{v}_k = \langle \boldsymbol{\eta}_k, \text{vech}\, \boldsymbol{\Lambda}_k \rangle$ represent the distribution parameters of cluster k as vertices in $\mathbb{R}^{d+d(d+1)/2}$, and let $\mathcal{V}$ be the set of these vertices over $k = 1, 2, ..., K$. Lemma 1 provides a unique partition of $\mathcal{V}$ into extremal vertices $\text{ext}(\mathcal{V})$ and relative interior vertices $\text{relint}(\mathcal{V})$, indexed by $\mathcal{P}$ and $\mathcal{C}$ of sizes K_P and K_C respectively.

If all clusters are prototypes, the model coincides with the finite Gaussian mixture model and is thus identifiable. Suppose there is at least one chimeral cluster. Given prototype parameters $\boldsymbol{v}_p$ for $p \in \mathcal{P}$, we show that each $\boldsymbol{v}_c$ has a unique representation as $\boldsymbol{\alpha}_c$ in the K_P-simplex for each $c \in \mathcal{C}$. By definition, each $\boldsymbol{v}_c$ for $c \in \mathcal{C}$ can be written as a convex combination of $\boldsymbol{v}_{P_1}, ..., \boldsymbol{v}_{P_{K_P}}$. This implies there exists an $\boldsymbol{\alpha}_c = \langle \alpha_{cP_1}, ..., \alpha_{cP_{K_P}} \rangle$ in the K_P-simplex that satisfies the linear system

$$\begin{bmatrix} | & & | \\ \boldsymbol{v}_{P_1} & \cdots & \boldsymbol{v}_{P_{K_P}} \\ | & & | \end{bmatrix} \boldsymbol{\alpha}_c = \mathbf{V}\boldsymbol{\alpha}_c = \boldsymbol{v}_c. \tag{3.2}$$

Any chimeral cluster or relative interior point $\boldsymbol{v}_c \in \text{relint}(S)$ has this characterization. By definition of relint(S), the system (3.2) is always consistent regardless of the shape of $\mathbf{V}$ and has a solution even if overdetermined. However, (3.2) may also have infinite solutions.

As such, it remains to determine a condition under which $\boldsymbol{\alpha}_c$ is uniquely identifiable. A natural constraint is to minimize the ℓ^2-norm of $\boldsymbol{\alpha}_c$ constrained to the standard K_P-simplex. This can be interpreted as minimizing the deviation from equal weighting or spreading out the weights over prototypes. Imposing this condition, we find $\boldsymbol{\alpha}_c$ to be that which satisfies the quadratic program

$$\begin{aligned} \underset{\boldsymbol{\alpha}_c}{\text{minimize}} \quad & \boldsymbol{\alpha}_c^\top \boldsymbol{\alpha}_c \\ \text{subject to} \quad & \boldsymbol{\alpha}_c \geq \mathbf{0}, \quad \mathbf{1}^\top \boldsymbol{\alpha}_c = 1, \quad \mathbf{V}\boldsymbol{\alpha}_c = \boldsymbol{v}_c. \end{aligned} \tag{3.3}$$

It remains to show uniqueness. We note that the standard K_P-simplex and the solution space to (3.2) are convex; hence, the constraint set formed by their intersection is convex. Additionally, the objective function is strictly convex as the quadratic form is positive-definite. Since the solution to a convex program (3.3) with a strictly convex objective is unique, this procedure yields a unique representation $\boldsymbol{\alpha}_c$ for each chimeral cluster $c \in \mathcal{C}$. Hence, the given condition for identifiability of chimeral clustering is sufficient. QED

Remark 1 *Epistatic clustering (Zhang, 2013) may be seen as a special case of chimeral clustering whereby $\boldsymbol{\alpha}_c$ is restricted to specific values averaging parameters over a subset of prototypes. Since epistatic clustering does not require a condition on the equivalent representation of $\boldsymbol{\alpha}_c$, it is not identifiable in the given formulation. It is susceptible to the*

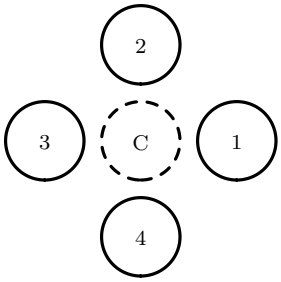

Figure 3.3: A chimeral cluster C with more than one $\boldsymbol{\alpha}_C$ representation in terms of two or four prototype clusters. For simplicity, all covariances are the identity matrix. This system can be characterized by (3.4).

example posed below in Figure 3.3 with the corresponding linear system given by

$$\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_{c1} \\ \alpha_{c2} \\ \alpha_{c3} \\ \alpha_{c4} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}. \tag{3.4}$$

An infinite number of solutions to (3.4) exist with three possible solutions being $\langle \frac{1}{2}, 0, \frac{1}{2}, 0 \rangle$, $\langle 0, \frac{1}{2}, 0, \frac{1}{2} \rangle$, *and* $\langle \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4} \rangle$.

Remark 2 *Impositions of an implicit condition to obtain identifiability can be found in factor analysis (Shapiro, 1985) and in finite Gaussian mixtures with parsimonious covariance matrices (Celeux and Govaert, 1995). In the former case, one may consider the varimax (Kaiser, 1958) or the oblimin (Clarkson and Jennrich, 1988) rotations, among others. In the latter case, one may restrict the determinant of the shape matrix* $\mathbf{A}_k$ *to unity.*

3.2.2 Number of Parameters

In the proposed model, the prototypes $\boldsymbol{\eta}_p$ and $\boldsymbol{\Lambda}_p$ contribute d and $d(d+1)/2$ respectively and the π_k values contribute $K_P + K_C - 1$ parameters. The remaining parameters in $\boldsymbol{\alpha}_c$ require some deeper consideration. In particular, they are represented by K_P values with an equality constraint $\mathbf{1}_{K_P}^\top \boldsymbol{\alpha}_c = 1$. At first glance, this seems to yield $K_P - 1$ free parameters. However, considering the characteristic system presented in (3.2) for K_P larger than $d + d(d+1)/2 + 1$, there are redundancies in $\boldsymbol{\alpha}_c$. Hence, when enumerating the number of parameters, we consider each $\boldsymbol{\alpha}_c$ to contribute $\min\{K_P - 1, d + \frac{d(d+1)}{2}\}$ parameters.

3.3 Parameter Estimation

The central process applies an expectation-maximization (EM) type algorithm by reformulating the problem into one of missing data; the cluster assignments are unobserved. As in Gaussian mixture models, the EM algorithm turns an intractable maximum likelihood problem into a tractable iterative solution of alternating expectation and maximization steps. In chimeral clustering, we further decompose the maximization step using expectation-conditional-maximization (Meng and Rubin, 1993) with multi-cycle updates. We opt to improve the objective function rather than maximize in some steps using the minorization-maximization algorithm (Ortega and Rheinboldt, 2000; De Leeuw and Heiser, 1977; Hunter and Lange, 2004), which qualifies the algorithm as generalized EM (Dempster et al., 1977). Of particular note is an application of the solution to the continuous-time algebraic Riccati equation of control theory (Laub, 1979).

3.3.1 Model Likelihood

We proceed as in finite Gaussian mixture models, defining the observations as $\boldsymbol{x}_n$ for $n = 1, 2, ..., N$ with corresponding cluster assignment indicator z_{nk} for $\boldsymbol{x}_n$ belonging to cluster k. As each chimeral cluster $c \in \mathcal{C}$ follows a $\mathrm{N}(\boldsymbol{\eta}_c, \boldsymbol{\Lambda}_c)$ distribution with a convex combination

of prototype multivariate normal parameters: $\boldsymbol{\eta}_c = \sum_{p\in\mathcal{P}} \alpha_{cp}\boldsymbol{\eta}_p$, $\boldsymbol{\Lambda}_c = \sum_{p\in\mathcal{P}} \alpha_{cp}\boldsymbol{\Lambda}_p$. The complete data log-likelihood is given by

$$\ell_c(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} \left[\sum_{p\in\mathcal{P}} z_{np} \log \phi(\boldsymbol{x}_i \mid \boldsymbol{\eta}_p, \boldsymbol{\Lambda}_p) + \sum_{c\in\mathcal{C}} z_{nc} \log \phi(\boldsymbol{x}_i \mid \boldsymbol{\eta}_c, \boldsymbol{\Lambda}_c) \right]$$

and the incomplete data log-likelihood is given by

$$\ell(\boldsymbol{\theta}; \mathbf{X}) = \sum_{n=1}^{N} \log \left[\sum_{p\in\mathcal{P}} \pi_p \phi(\boldsymbol{x}_i \mid \boldsymbol{\eta}_p, \boldsymbol{\Lambda}_p) + \sum_{c\in\mathcal{C}} \pi_c \phi(\boldsymbol{x}_i \mid \boldsymbol{\eta}_c, \boldsymbol{\Lambda}_c) \right];$$

we have expanded the sum into prototype terms and chimeral terms to emphasize the distinction.

Here, the set of parameters $\boldsymbol{\theta}$ is $\{\boldsymbol{\pi}, \{\boldsymbol{\eta}_p\}_{p\in\mathcal{P}}, \{\boldsymbol{\Lambda}_p\}_{p\in\mathcal{P}}, \{\boldsymbol{\alpha}_c\}_{c\in\mathcal{C}}\}$. We perform maximum likelihood estimation by maximizing with respect to the prototype $\boldsymbol{\eta}_p$ and $\boldsymbol{\Lambda}_p$ multivariate normal parameters, the chimeral cluster mixing coefficients α_{cp}, and the marginal probabilities π_k for $k \in \mathcal{P} \cup \mathcal{C}$.

3.3.2 Initialization

We present a heuristic for initializing the estimation procedure for pre-specified values of K_P and K_C by approximating a chimeral clustering model using an *mclust* model or a k-means model. For a given parsimonious covariance matrix specification, we perform a $K = K_P + K_C$ cluster model fit. Consider each of the K multivariate normal parameters $(\hat{\boldsymbol{\eta}}_k, \hat{\boldsymbol{\Lambda}}_k)$. Let $\bar{\boldsymbol{\eta}} = \frac{1}{K}\sum_{k=1}^{K} \hat{\boldsymbol{\eta}}_k$ and $\text{dist}_k = \|\hat{\boldsymbol{\eta}}_k - \bar{\boldsymbol{\eta}}\|$. Let $\mathcal{P}$ index the K_P largest dist_k and $\mathcal{C}$ the remainder.

For each $c \in \mathcal{C}$, find the best approximation to $\hat{\boldsymbol{\eta}}_c$ using convex combinations of $\{\hat{\boldsymbol{\eta}}_p\}_{p\in\mathcal{P}}$. That is, find $\boldsymbol{\alpha}_c$ satisfying

$$\begin{aligned} \underset{\boldsymbol{\alpha}_c}{\text{minimize}} \quad & \left\| \hat{\boldsymbol{\eta}}_c - \sum_{p\in\mathcal{P}} \alpha_{cp} \hat{\boldsymbol{\eta}}_p \right\|_2 \\ \text{subject to} \quad & \boldsymbol{\alpha}_c \succeq \mathbf{0}, \quad \mathbf{1}^\top \boldsymbol{\alpha}_c = 1. \end{aligned} \tag{3.5}$$

In addition to the fourteen choices of parsimonious covariance matrices, we allow an additional k-means initialization with a common weighted average of cluster covariances as $\boldsymbol{\Sigma}_g = \boldsymbol{\Sigma}$ and the $K_P + K_C$ centroids taking the role of $\boldsymbol{\mu}$. Finally, in the *mclust* case, $\boldsymbol{\pi}$ is initialized by permuting the corresponding $\boldsymbol{\pi}$ values from the *mclust* model fit into $\mathcal{P}$ and $\mathcal{C}$ indices. In the k-means case, the proportion of observations assigned to each cluster $g \in \mathcal{C} \cup \mathcal{P}$ is used to initialize π_k.

This results in up to fifteen starter models yielding initial parameter estimates $\boldsymbol{\eta}_p^{(0)} = \hat{\boldsymbol{\eta}}_p$, $\boldsymbol{\Lambda}_p^{(0)} = \hat{\boldsymbol{\Lambda}}_p$, $\boldsymbol{\alpha}_c^{(0)}$ and $\boldsymbol{\pi}^{(0)}$. We then populate the conditional probability estimates $\hat{z}_{nk}$ using these parameters and the expression in Section 3.3.4. For some number of initial iterations, we hold the initially estimated $\hat{z}_{nk}$ fixed. We then perform mini-EM by running a small number of iterations of the EM algorithm for each, and considering the highest log-likelihood among them to be the initialization (Biernacki et al., 2003). To exclude some degenerate cases, we may disqualify starter models that have a very small $\sum_{n=1}^{N} \hat{z}_{nk}$, or those that have a component with a covariance having a very small eigenvalue.

3.3.3 Alternate Initialization

We present a rough heuristic for a random initialization of the procedure and pre-specified values of K_P and K_C. We sample half of the data and fit a fully-varying covariance Gaussian mixture with $K = K_P + K_C$ components. We extract the K fitted centroids $\hat{\boldsymbol{\mu}}_1, ..., \hat{\boldsymbol{\mu}}_K$ and the overall centroid $\hat{\boldsymbol{\mu}}_\cdot$ and compute $\text{dist}_k = \|\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}}_\cdot\|_2$. We classify the K_P clusters with largest dist_k as prototype clusters and the remaining K_C as chimeral clusters. We convert the fitted moment parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ to $(\boldsymbol{\eta}_k, \boldsymbol{\Lambda}_k)$ for all $k \in \mathcal{P} \cup \mathcal{C}$; however, this does not guarantee compatible mixing proportions $\boldsymbol{\alpha}_c$ for any chimeral cluster $c \in \mathcal{C}$. We loosely approximate a valid $\boldsymbol{\alpha}_c$ for initialization by solving (3.6), though we re-initialize if any $\alpha_{cp} \approx 1$ to prevent a chimeral cluster from coinciding with a prototype at the outset.

$$\begin{aligned} \underset{\boldsymbol{\alpha}_c}{\text{minimize}} \quad & \left\| \begin{bmatrix} \hat{\boldsymbol{\eta}}_c \\ \operatorname{vech} \hat{\boldsymbol{\Lambda}}_c \end{bmatrix} - \sum_{p \in \mathcal{P}} \alpha_{cp} \begin{bmatrix} \hat{\boldsymbol{\eta}}_p \\ \operatorname{vech} \hat{\boldsymbol{\Lambda}}_p \end{bmatrix} \right\|_2 \\ \text{subject to} \quad & \sum_{p \in \mathcal{P}} \alpha_{cp} = 1, \\ & \alpha_{cp} \geq 0 \quad \text{and} \quad p \in \mathcal{P}. \end{aligned} \tag{3.6}$$

After estimating a valid $\boldsymbol{\alpha}_c$ for each $c \in \mathcal{C}$, we update $\boldsymbol{\eta}_c$ and $\boldsymbol{\Lambda}_c$ using the definition thereof to produce a valid initial set of parameters $\boldsymbol{\eta}_p$, $\boldsymbol{\Lambda}_p$, and $\boldsymbol{\alpha}_c$. We approximately initialize $\boldsymbol{\pi}$ using the proportion of observations assigned to each cluster by the Gaussian mixture model.

3.3.4 Expectation Step

As in finite Gaussian mixtures, we obtain a formula at iteration t for $\hat{z}_{nk}^{(t)}$ of the form

$$\hat{z}_{nk}^{(t)} = \frac{\pi_k^{(t)} \, \phi\left(\boldsymbol{x}_n \mid \boldsymbol{\eta}_k^{(t)}, \boldsymbol{\Lambda}_k^{(t)}\right)}{\sum_{j \in \mathcal{P} \cup \mathcal{C}} \pi_j^{(t)} \, \phi\left(\boldsymbol{x}_n \mid \boldsymbol{\eta}_j^{(t)}, \boldsymbol{\Lambda}_j^{(t)}\right)}. \tag{3.7}$$

The surrogate objective function from the expectation step of the EM algorithm is

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) = \sum_{n=1}^{N} \sum_{k \in \mathcal{P} \cup \mathcal{C}} \hat{z}_{nk}^{(t)} \left[\log \pi_k + \log \phi(\boldsymbol{x}_n \mid \boldsymbol{\eta}_k, \boldsymbol{\Lambda}_k)\right]. \tag{3.8}$$

3.3.5 Maximization Step

For simplicity, we first define $N_k^{(t)} = \sum_{n=1}^{N} \hat{z}_{nk}^{(t)}$, $\boldsymbol{y}_k^{(t)} = \sum_{n=1}^{N} \hat{z}_{nk}^{(t)} \boldsymbol{x}_n$ and $\mathbf{W}_k^{(t)} = \sum_{n=1}^{N} \hat{z}_{nk}^{(t)} \boldsymbol{x}_n \boldsymbol{x}_n^\top$. We decompose the maximization step into multiple conditional maximization steps, running the expectation step as necessary after each.

Maximization in π_k is as in finite Gaussian mixtures; we obtain the maximizer $\hat{\pi}_{k,\mathrm{ML}} = N_k^{(t)}/N$ for both types of clusters. This maximizer is independent of the other estimable parameters and is easy to compute, so we recompute this immediately after the expectation step during the other conditional maximization steps as well.

Next, we maximize in all $\boldsymbol{\eta}_p$ for $p \in \mathcal{P}$ simultaneously by solving the following linear system

$$\left[\mathbf{R}^{(t)} + \text{diag}\left(N_1 \boldsymbol{\Lambda}_1^{(t)-1}, ..., N_{K_P} \boldsymbol{\Lambda}_{K_P}^{(t)-1}\right)\right] \begin{bmatrix} \boldsymbol{\eta}_1 \\ \vdots \\ \boldsymbol{\eta}_{K_P} \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}_1^{(t)} + \sum_{c \in \mathcal{C}} \alpha_{c1}^{(t)} \boldsymbol{y}_c^{(t)} \\ \vdots \\ \boldsymbol{y}_{K_P}^{(t)} + \sum_{c \in \mathcal{C}} \alpha_{cK_P}^{(t)} \boldsymbol{y}_c^{(t)} \end{bmatrix}$$

$$\text{where } \mathbf{R}^{(t)} = \sum_{c \in \mathcal{C}} N_c^{(t)} (\boldsymbol{\alpha}_c^{(t)} \boldsymbol{\alpha}_c^{(t)\top} \otimes \mathbf{J}_{d \times d}) \odot (\mathbf{J}_{K_P \times K_P} \otimes \boldsymbol{\Lambda}_c^{(t)-1}).$$

Subsequently, we maximize in $\boldsymbol{\Lambda}_p$ for each $p \in \mathcal{P}$. We note the intractability of solving the matrix derivatives for the simultaneous maximizer of all $\boldsymbol{\Lambda}_p$, and resort to conditionally maximizing in each $\boldsymbol{\Lambda}_p$. Additionally, we employ minorization-maximization (Hunter and Lange, 2004; Hansen and Pedersen, 2003) to obtain a feasibly solvable equation for a minorizer of $\boldsymbol{\Lambda}_p$

$$-N_p^{(t)} \boldsymbol{\Lambda}_p^{-1} - \boldsymbol{\Lambda}_p^{-1} \mathbf{M}_p^{(t)} \boldsymbol{\Lambda}_p^{-1} + \left(\mathbf{W}_p^{(t)} + \sum_{c \in \mathcal{C}} \alpha_{cp} \mathbf{W}_c^{(t)}\right) = \mathbf{0}$$

where $\mathbf{M}_p^{(t)} = N_p^{(t)} \boldsymbol{\eta}_p^{(t)} \boldsymbol{\eta}_p^{(t)\top} + \sum_{c \in \mathcal{C}} \alpha_{cp} N_c^{(t)} \boldsymbol{\Lambda}_p^{(t)} \left(\boldsymbol{\Lambda}_c^{-(t)} \boldsymbol{\eta}_c^{(t)} \boldsymbol{\eta}_c^{(t)\top} \boldsymbol{\Lambda}_c^{-(t)} + \boldsymbol{\Lambda}_c^{-(t)}\right) \boldsymbol{\Lambda}_p^{(t)}$. This equation is the form of the continuous-time algebraic Riccati equation, for which a symmetric positive-definite solution of $\boldsymbol{\Lambda}_p^{-1}$ can be found in Laub (1979).

Finally, we maximize in each $\boldsymbol{\alpha}_c$ for each $c \in \mathcal{C}$. Due to the complexity of the objective function, we perform the soft-max reparameterization to obtain an unconstrained problem, and perform a single Newton-Raphson step each iteration starting from the previous value. We perform up to 100 Newton-Raphson iterations for each $\boldsymbol{\alpha}_c$ maximization step, and assess convergence by the ℓ^1-norm of successive iterations in the soft-max transformed space being below 10^{-12}. If the updated value of $\boldsymbol{\alpha}_c$ yields a lower value of the surrogate objective function, we revert the update.

3.4 Expectation Maximization Procedure

3.4.1 Expectation Step

As in finite Gaussian mixtures, we obtain a formula at iteration t of z_{nk} of the form 3.9.

$$\hat{z}_{nk}^{(t)} = \frac{\pi_k^{(t)} \, \phi\left(\boldsymbol{x}_n \mid \boldsymbol{\eta}_k^{(t)}, \boldsymbol{\Lambda}_k^{(t)}\right)}{\sum_{j \in \mathcal{P} \cup \mathcal{C}} \pi_j^{(t)} \, \phi\left(\boldsymbol{x}_n \mid \boldsymbol{\eta}_j^{(t)}, \boldsymbol{\Lambda}_j^{(t)}\right)} \tag{3.9}$$

The surrogate objective function from the expectation step of the EM algorithm follows:

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) = \sum_{n=1}^{N} \sum_{k \in \mathcal{P} \cup \mathcal{C}} \hat{z}_{nk}^{(t)} \left[\log \pi_k + \log \phi(\boldsymbol{x}_n \mid \boldsymbol{\eta}_k, \boldsymbol{\Lambda}_k)\right] \tag{3.10}$$

3.4.2 Sampling Step

We mitigate the presence of local extrema using an initial regime of stochastic EM (Celeux and Diebolt, 1986) for a preset number of iterations, followed by standard EM until convergence. If Stochastic EM (Celeux and Diebolt, 1986) applies to the current iteration, then after each Expectation step we generate a pseudosample based on the $\hat{z}_{nk}^{(t)}$ values. We perform a hard assignment of each observation n to a cluster $k \in \mathcal{P} \cup \mathcal{C}$ according to the Categorical$(\hat{z}_{n1}^{(t)}, ..., \hat{z}_{nK}^{(t)})$ distribution.

If Stochastic EM (Celeux and Diebolt, 1986) applies to the current iteration, we perform sampling based on the $\hat{z}_{nk}^{(t)}$ calculated in the previous step. To generate the pseudosample $\mathbf{Z}^{(t)}$, we draw an assignment for each observation n to a cluster using the categorical distribution $\tilde{z}_n^{(t)} \sim$ Categorical$(\hat{z}_{n1}^{(t)}, ..., \hat{z}_{nK}^{(t)})$. We incorporate this pseudosample into the maximization step by altering $\hat{z}_{nk}^{(t)} = \mathbb{1}(\tilde{z}_n^{(t)} = k)$ and retaining the same notation and steps.

As part of the investigation into initializing the EM algorithm, we applied the random start initialization procedure described below. The slight random perturbations in initial parameters afforded by subsampling the data before fitting a Gaussian mixture model lead to considerably different local optima.

An additional Stochastic EM sampling step (Celeux and Diebolt, 1986) was added to the EM algorithm to escape local optima with varying degrees of success. This step lead to more of the random starts reaching a better optima for some datasets. More concerningly however, it would also often miss viable optima by falling into a degenerate state with very small covariance eigenvalues or no responsibility allocated to a cluster.

While this procedure occasionally produced slightly better BIC values compared to the current deterministic estimation procedure, it required a very large number of random starts to achieve this.

3.4.3 Maximization Step

We decompose Equation (3.10) into prototype and chimeral sums at this point and substitute in the multivariate normal density to form (3.11). To reduce notational load, define $N_k^{(t)} = \sum_{n=1}^N \hat{z}_{nk}^{(t)}$, $\boldsymbol{y}_k^{(t)} = \sum_{n=1}^N \hat{z}_{nk}^{(t)} \boldsymbol{x}_n$ and $\mathbf{W}_k^{(t)} = \sum_{n=1}^N \hat{z}_{nk}^{(t)} \boldsymbol{x}_n \boldsymbol{x}_n^\top$ to represent the effective number of observations, weighted centroid of assigned observations, and weighted non-central scatter matrix for cluster $k \in \mathcal{P} \cup \mathcal{C}$, respectively.

$$
\begin{aligned}
Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) = &-\frac{d}{2} N \log 2\pi \\
&+ \sum_{p \in \mathcal{P}} \left[N_p \left(\log \pi_p + \frac{1}{2} \log \det \boldsymbol{\Lambda}_p - \frac{1}{2} \boldsymbol{\eta}_p^\top \boldsymbol{\Lambda}_p^{-1} \boldsymbol{\eta}_p \right) + \boldsymbol{\eta}_p^\top \boldsymbol{y}_p - \frac{1}{2} \operatorname{Tr}\left(\boldsymbol{\Lambda}_p \mathbf{W}_p \right) \right] \\
&+ \sum_{c \in \mathcal{C}} \left[N_c \left(\log \pi_c + \frac{1}{2} \log \det \boldsymbol{\Lambda}_c - \frac{1}{2} \boldsymbol{\eta}_c^\top \boldsymbol{\Lambda}_c^{-1} \boldsymbol{\eta}_c \right) + \boldsymbol{\eta}_c^\top \boldsymbol{y}_c - \frac{1}{2} \operatorname{Tr}\left(\boldsymbol{\Lambda}_c \mathbf{W}_c \right) \right]
\end{aligned}
\tag{3.11}
$$

Maximizing $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)})$ over $\boldsymbol{\theta} = \{\boldsymbol{\eta}_p, \boldsymbol{\Lambda}_p, \boldsymbol{\alpha}_c, \boldsymbol{\pi}\}$ is highly intractable due to the mixing parameters $\boldsymbol{\alpha}_c$ being nearly omnipresent; hence, we switch to multi-cycle EM and perform conditional maximizations over each set of parameters $\{\boldsymbol{\eta}_p, \boldsymbol{\Lambda}_p, \boldsymbol{\alpha}_c, \boldsymbol{\pi}\}$ separately.

Maximizing in $\boldsymbol{\pi}$

As in standard Gaussian mixture models, we note in Equation (3.11) the π_k term is separable and that there is a constraint $\sum_{k\in\mathcal{P}\cup\mathcal{C}}\pi_k = 1$. As such, we formulate problem using Lagrange multipliers as $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) - \lambda\left(\sum_{k\in\mathcal{P}\cup\mathcal{C}}\pi_k - 1\right)$ for $\lambda \in \mathbb{R}$. We solve to find the standard Gaussian mixture model solution $\hat{\pi}_{k,\mathrm{ML}} = N_k^{(t)}/N$.

Maximizing in $\boldsymbol{\eta}_g$

We differentiate (3.11) in $\boldsymbol{\eta}_g$ for $g \in \mathcal{P}$ and obtain Equation (3.12); by equating each $\partial Q/\partial\boldsymbol{\eta}_g$ to zero, we obtain a blockwise defined linear system solvable for all $\boldsymbol{\eta}_p$ simultaneously or a sequence of systems solvable for each $\boldsymbol{\eta}_p$ using multi-cycle updates.

$$\frac{\partial Q}{\partial \boldsymbol{\eta}_g} = \boldsymbol{y}_g^{(t)} - N_g^{(t)}\boldsymbol{\Lambda}_g^{-1}\boldsymbol{\eta}_g + \sum_{c\in\mathcal{C}}\alpha_{cg}\boldsymbol{y}_c^{(t)} - \sum_{c\in\mathcal{C}}\alpha_{cg}N_c^{(t)}\boldsymbol{\Lambda}_c^{-1}\boldsymbol{\eta}_c \tag{3.12}$$

$$\sum_{c\in\mathcal{C}}\begin{bmatrix} N_1\boldsymbol{\Lambda}_1^{-1} + N_c\alpha_{c1}\alpha_{c1}\boldsymbol{\Lambda}_c^{-1} & \cdots & N_c\alpha_{c1}\alpha_{cK_p}\boldsymbol{\Lambda}_c^{-1} \\ \vdots & \ddots & \vdots \\ N_c\alpha_{cK_P}\alpha_{c1}\boldsymbol{\Lambda}_c^{-1} & \cdots & N_{K_P}\boldsymbol{\Lambda}_{K_P}^{-1} + N_c\alpha_{cK_P}\alpha_{cK_P}\boldsymbol{\Lambda}_c^{-1} \end{bmatrix}$$

$$\sum_{c\in\mathcal{C}} N_c(\boldsymbol{\alpha}_c\boldsymbol{\alpha}_c^\top \otimes \mathbf{J}_{d\times d}) \odot (\mathbf{J}_{K_P\times K_P} \otimes \boldsymbol{\Lambda}_c^{-1}) + \operatorname{diag}\left(N_1\boldsymbol{\Lambda}_1^{-1}, ..., N_{K_P}\boldsymbol{\Lambda}_{K_P}^{-1}\right)$$

Maximizing in $\boldsymbol{\Lambda}_p$

To maximize in each prototype $\boldsymbol{\Lambda}_g$, we would like to compute the derivative of Equation (3.11) with respect to $\boldsymbol{\Lambda}_g$. While the differentiation itself is tractable, attempting to solve for the roots is not; thus, we resort to the minorization-maximization algorithm to produce an increase in Q rather than a maxima. Effectively, this renders the estimation process into a generalized EM algorithm.

We derive here an appropriate minorization for $\boldsymbol{\eta}_c^\top\boldsymbol{\Lambda}_c^{-1}\boldsymbol{\eta}_c$; to do so, we apply the operator Jensen inequality (Hansen and Pedersen, 2003, Theorem 2.1). The statement for an

operator convex function f, elements $x_1, x_2, ..., x_n$, and operators $\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_n$ satisfying $\sum_{i=1}^{n} \mathbf{A}_i^* \mathbf{A}_i = \mathbf{1}$ is as follows:

$$f\left(\sum_{i=1}^{n} \mathbf{A}_i^* x_i \mathbf{A}_i\right) \leq \sum_{i=1}^{n} \mathbf{A}_i^* f(x_i) \mathbf{A}_i \tag{3.13}$$

In particular, the matrix inverse is an operator convex function; let $f(\mathbf{X}) = \mathbf{X}^{-1}$. We now derive an appropriate minorization for the inverse of $\mathbf{\Lambda}_c$ by multiplying with an appropriate expansion of identity matrices.

$$\begin{aligned}\left(\sum_{p\in\mathcal{P}} \alpha_{cp}\mathbf{\Lambda}_p\right)^{-1} &= \mathbf{\Lambda}_c^{-\frac{(t)}{2}} \left(\sum_{p\in\mathcal{P}} \alpha_{cp}\mathbf{\Lambda}_c^{-\frac{(t)}{2}}\mathbf{\Lambda}_p\mathbf{\Lambda}_c^{-\frac{(t)}{2}}\right)^{-1} \mathbf{\Lambda}_c^{-\frac{(t)}{2}} \\ &= \mathbf{\Lambda}_c^{-\frac{(t)}{2}} \left(\sum_{p\in\mathcal{P}} \sqrt{\alpha_{cp}}\mathbf{\Lambda}_c^{-\frac{(t)}{2}}\mathbf{\Lambda}_p^{\frac{(t)}{2}}\mathbf{\Lambda}_p^{-\frac{(t)}{2}}\mathbf{\Lambda}_p\mathbf{\Lambda}_p^{-\frac{(t)}{2}}\mathbf{\Lambda}_p^{\frac{(t)}{2}}\mathbf{\Lambda}_c^{-\frac{(t)}{2}}\sqrt{\alpha_{cp}}\right)^{-1} \mathbf{\Lambda}_c^{-\frac{(t)}{2}}\end{aligned}$$

Define $\mathbf{A}_p = \sqrt{\alpha_{cp}}\mathbf{\Lambda}_p^{\frac{(t)}{2}}\mathbf{\Lambda}_c^{-\frac{(t)}{2}}$ so that $\mathbf{A}_p^*\mathbf{A}_p = \mathbf{A}_p^\top\mathbf{A}_p = \mathbf{\Lambda}_c^{-\frac{(t)}{2}}\left(\sum_{p\in P}\alpha_{cp}\mathbf{\Lambda}_p^{(t)}\right)\mathbf{\Lambda}_c^{-\frac{(t)}{2}} = \mathbf{I}$. Let $\mathbf{X}_p = \mathbf{\Lambda}_p^{-\frac{(t)}{2}}\mathbf{\Lambda}_p\mathbf{\Lambda}_p^{-\frac{(t)}{2}}$ for convenience.

$$\begin{aligned}\left(\sum_{p\in\mathcal{P}} \alpha_{cp}\mathbf{\Lambda}_p\right)^{-1} &= \mathbf{\Lambda}_c^{-\frac{(t)}{2}} \left(\sum_{p\in\mathcal{P}} \mathbf{A}_p^\top\mathbf{X}_p\mathbf{A}_p\right)^{-1} \mathbf{\Lambda}_c^{-\frac{(t)}{2}} \\ &= \mathbf{\Lambda}_c^{-\frac{(t)}{2}} f\left(\sum_{p\in\mathcal{P}} \mathbf{A}_p^\top\mathbf{X}_p\mathbf{A}_p\right) \mathbf{\Lambda}_c^{-\frac{(t)}{2}}\end{aligned}$$

We now apply Equation (3.13) to f.

$$\begin{aligned}\left(\sum_{p\in\mathcal{P}} \alpha_{cp}\mathbf{\Lambda}_p\right)^{-1} &\leq \mathbf{\Lambda}_c^{-\frac{(t)}{2}} \left(\sum_{p\in\mathcal{P}} \mathbf{A}_p^\top f(\mathbf{X}_p)\mathbf{A}_p\right) \mathbf{\Lambda}_c^{-\frac{(t)}{2}} \\ &= \mathbf{\Lambda}_c^{-\frac{(t)}{2}} \left(\sum_{p\in\mathcal{P}} \mathbf{A}_p^\top f(\mathbf{X}_p)\mathbf{A}_p\right) \mathbf{\Lambda}_c^{-\frac{(t)}{2}} \\ &= \sum_{p\in\mathcal{P}} \alpha_{cp}\mathbf{\Lambda}_c^{-(t)}\mathbf{\Lambda}_p^{(t)}\mathbf{\Lambda}_p^{-1}\mathbf{\Lambda}_p^{(t)}\mathbf{\Lambda}_c^{-(t)}\end{aligned} \tag{3.14}$$

Indeed, at $\mathbf{\Lambda}_p = \mathbf{\Lambda}_p^{(t)}$ for all $p \in \mathcal{P}$ we have equality as required of a minorizing surrogate. From the convexity of the log-determinant and the operator Jensen inequality, we arrive at the minorizations (3.15) and (3.16).

$$\log\det\mathbf{\Lambda}_c \geq \log\det\mathbf{\Lambda}_c^{(t)} + \operatorname{Tr}\left[\mathbf{\Lambda}_c^{-1}\left(\mathbf{\Lambda}_c - \mathbf{\Lambda}_c^{(t)}\right)\right] \tag{3.15}$$

$$-\boldsymbol{\eta}_c^\top\mathbf{\Lambda}_c^{-1}\boldsymbol{\eta}_c \geq -\sum_{p\in\mathcal{P}}\alpha_{cp}\boldsymbol{\eta}_c^\top\mathbf{\Lambda}_c^{-(t)}\mathbf{\Lambda}_p^{(t)}\mathbf{\Lambda}_p^{-1}\mathbf{\Lambda}_p^{(t)}\mathbf{\Lambda}_c^{-(t)}\boldsymbol{\eta}_c \tag{3.16}$$

We may further apply (3.16) to (3.15) to obtain a more useful minorizer (3.17).

$$\begin{aligned}\log\det\mathbf{\Lambda}_c^{(t)} + \operatorname{Tr}\left[\mathbf{\Lambda}_c^{-1}\left(\mathbf{\Lambda}_c - \mathbf{\Lambda}_c^{(t)}\right)\right] &= \log\det\mathbf{\Lambda}_c^{(t)} + d - \operatorname{Tr}\left(\mathbf{\Lambda}_c^{\frac{(t)}{2}}\mathbf{\Lambda}_c^{-1}\mathbf{\Lambda}_c^{\frac{(t)}{2}}\right)\\ &\geq \log\det\mathbf{\Lambda}_c^{(t)} + d - \operatorname{Tr}\left(\sum_{p\in\mathcal{P}}\alpha_{cp}\mathbf{\Lambda}_c^{\frac{(t)}{2}}\mathbf{\Lambda}_p^{(t)}\mathbf{\Lambda}_p^{-1}\mathbf{\Lambda}_p^{(t)}\mathbf{\Lambda}_c^{\frac{(t)}{2}}\right)\end{aligned} \tag{3.17}$$

Applying minorizations (3.16) and (3.17) to Equation (3.11), we obtain the minorized surrogate (3.18) and corresponding derivative (3.19).

$$\begin{aligned}Q_m = &\frac{N_g}{2}\left(\log\det\mathbf{\Lambda}_g - \boldsymbol{\eta}_g^\top\mathbf{\Lambda}_g^{-1}\boldsymbol{\eta}_g + \operatorname{Tr}\left(\mathbf{\Lambda}_g\mathbf{W}_g\right)\right)\\ &- \sum_{c\in\mathcal{C}}\left[\frac{N_c}{2}\operatorname{Tr}\left(\sum_{p\in\mathcal{P}}\alpha_{cp}\boldsymbol{\eta}_c^\top\mathbf{\Lambda}_c^{-\frac{(t)}{2}}\mathbf{\Lambda}_p^{(t)}\mathbf{\Lambda}_p^{-1}\mathbf{\Lambda}_p^{(t)}\mathbf{\Lambda}_c^{-\frac{(t)}{2}}\boldsymbol{\eta}_c\right)\right]\\ &- \sum_{c\in\mathcal{C}}\left(\frac{N_c}{2}\sum_{p\in\mathcal{P}}\alpha_{cp}\boldsymbol{\eta}_c^\top\mathbf{\Lambda}_c^{-(t)}\mathbf{\Lambda}_p^{(t)}\mathbf{\Lambda}_p^{-1}\mathbf{\Lambda}_p^{(t)}\mathbf{\Lambda}_c^{-(t)}\boldsymbol{\eta}_c\right)\\ &- \frac{1}{2}\sum_{c\in\mathcal{C}}\operatorname{Tr}\left(\mathbf{\Lambda}_c\mathbf{W}_c\right) + \text{constant}\end{aligned} \tag{3.18}$$

$$\begin{aligned}\frac{\partial Q_m}{\partial\mathbf{\Lambda}_g} = &\frac{N_g}{2}\left(\mathbf{\Lambda}_g^{-1} - \mathbf{\Lambda}_g^{-1}\boldsymbol{\eta}_g\boldsymbol{\eta}_g^\top\mathbf{\Lambda}_g^{-1}\right) - \frac{1}{2}\mathbf{W}_c\\ &+ \mathbf{\Lambda}_g^{-1}\left[\sum_{c\in\mathcal{C}}\alpha_{cg}\frac{N_c}{2}\underbrace{\mathbf{\Lambda}_g^{(t)}\left(\mathbf{\Lambda}_c^{-(t)}\boldsymbol{\eta}_c\boldsymbol{\eta}_c^\top\mathbf{\Lambda}_c^{-(t)} + \mathbf{\Lambda}_c^{-(t)}\right)\mathbf{\Lambda}_g^{(t)}}_{\mathbf{M}_{cg}^{(t)}}\right]\mathbf{\Lambda}_g^{-1} - \frac{1}{2}\sum_{c\in\mathcal{C}}\alpha_{cg}\mathbf{W}_c\end{aligned} \tag{3.19}$$

Setting (3.19) to zero, we can re-arrange the equation to obtain the Continuous-time Algebraic Riccati Equation (CARE):

$$-N_g\boldsymbol{\Lambda}_g^{-1} - \boldsymbol{\Lambda}_g^{-1}\left(N_g\boldsymbol{\eta}_g\boldsymbol{\eta}_g^\top + \sum_{c\in\mathcal{C}}\alpha_{cg}N_c\mathbf{M}_{cg}^{(t)}\right)\boldsymbol{\Lambda}_g^{-1} + \left(\mathbf{W}_g + \sum_{c\in\mathcal{C}}\alpha_{cg}\mathbf{W}_c\right) = \mathbf{0} \tag{3.20}$$

While the specifics are beyond the scope of this work, we state here the form and solutions available in the literature. In optimal control theory, the Continuous-time Algebraic Riccati Equation (CARE) characterizes the stabilizing solution of a infinite-time horizon linear-quadratic regulator (3.21) as an unknown symmetric matrix $\mathbf{X}$ with constant real-valued matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{Q}$, and $\mathbf{R}$. A method of solving (3.21) for $\mathbf{X}$ is given by Laub (1979).

$$\mathbf{A}^\top\mathbf{X} + \mathbf{X}\mathbf{A} - \mathbf{X}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top\mathbf{X} + \mathbf{Q} = \mathbf{0} \tag{3.21}$$

We obtain an updated estimate $\hat{\boldsymbol{\Lambda}}_g^{(t+1)}$ by solving (3.20) for $\boldsymbol{\Lambda}_g^{-1}$ and inverting. While this step can be iterated until convergence to reach a maxima before continuing to the next CM step, we perform only one iteration.

Maximizing in $\boldsymbol{\alpha}_c$

To maximize over mixing parameters $\boldsymbol{\alpha}_c$ for each chimeral cluster $c \in \mathcal{C}$, we would like set the gradient to zero. However, we are bound by the convex constraint $\sum_{p\in\mathcal{P}}\alpha_{cp} = 1$. We perform the soft-max reparameterization in order to obtain an unconstrained problem. Define β_{cp} for $c \in \mathcal{C}$ and $p \in \mathcal{P}$ such that:

$$\alpha_{cp} = \frac{\exp\beta_{cp}}{\sum_{p\in\mathcal{P}}\exp\beta_{cp}}$$

Hence, we may choose $\boldsymbol{\beta}_c \in \mathbb{R}^p$ up to an additive constant while satisfying the convexity constraint on $\boldsymbol{\alpha}_c$:

$$\alpha_{cp} = \frac{\exp\beta_{cp}}{\sum_{p\in\mathcal{P}}\exp\beta_{cp}} = \frac{\exp\left(\beta_{cp}+\lambda\right)}{\sum_{p\in\mathcal{P}}\exp\left(\beta_{cp}+\lambda\right)} \quad \forall\lambda\in\mathbb{R}$$

This maximization is an intractable problem as posed; thus, we resort to the Newton-Raphson numerical optimization algorithm to maximize with respect to α_{cg} by working in the soft-max transformed space $\boldsymbol{\beta}_c$. The derivative of Equation (3.11) with respect to β_{cg} is:

$$\begin{aligned}\frac{\partial Q}{\partial \beta_{cg}} = \alpha_{cg}\Bigg\{&\boldsymbol{\eta}_g^\top \boldsymbol{y}_c - \boldsymbol{\eta}_c^\top \boldsymbol{y}_c + \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{\Lambda}_c \mathbf{W}_c\right) - \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{\Lambda}_g \mathbf{W}_c\right) + \frac{N_c}{2}[\operatorname{Tr}\left(\boldsymbol{\Lambda}_c^{-1}\boldsymbol{\Lambda}_g\right) - d] \\ &- \frac{N_c}{2}\left(2\boldsymbol{\eta}_c^\top \boldsymbol{\Lambda}_c^{-1}\boldsymbol{\eta}_g - \boldsymbol{\eta}_c^\top \boldsymbol{\Lambda}_c^{-1}\boldsymbol{\eta}_c - \boldsymbol{\eta}_c^\top \boldsymbol{\Lambda}_c^{-1}\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_c^{-1}\boldsymbol{\eta}_c\right)\Bigg\}\end{aligned} \tag{3.22}$$

The system of equations given by Equation (3.22) for $g \in \mathcal{P}$ is intractably difficult to solve for $\boldsymbol{\beta}_c$ in a closed form. Thus, we switch to the numeric Newton-Raphson method for root finding on the derivative. To do so, we require the second derivative as given below:

$$\begin{aligned}\frac{\partial^2 Q}{\partial \beta_{cg}\,\partial \beta_{ch}} = \mathbb{1}(g=h)\frac{\partial Q}{\partial \beta_{cg}} + \alpha_{cg}\alpha_{ch}\Bigg\{&\boldsymbol{\eta}_g^\top \boldsymbol{y}_c - \boldsymbol{\eta}_c^\top \boldsymbol{y}_c \\ &+ \frac{1}{2}\left[\operatorname{Tr}\left(\boldsymbol{\Lambda}_c \mathbf{W}_c\right) - \operatorname{Tr}\left(\boldsymbol{\Lambda}_g \mathbf{W}_c\right)\right] \\ &+ \frac{1}{2}\left[\operatorname{Tr}\left(\boldsymbol{\Lambda}_c^{-1}\boldsymbol{\Lambda}_g\right) - \operatorname{Tr}\left(\boldsymbol{\Lambda}_c^{-1}\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_c^{-1}\boldsymbol{\Lambda}_h\right)\right] \\ &- \boldsymbol{\eta}_g^\top \boldsymbol{\Lambda}_c^{-1}\boldsymbol{\eta}_h + \boldsymbol{\eta}_g^\top \boldsymbol{\Lambda}_c^{-1}\boldsymbol{\Lambda}_h\boldsymbol{\Lambda}_c^{-1}\boldsymbol{\eta}_c + \boldsymbol{\eta}_c^\top \boldsymbol{\Lambda}_c^{-1}\boldsymbol{\eta}_h \\ &- \frac{1}{2}\boldsymbol{\eta}_c^\top \boldsymbol{\Lambda}_c^{-1}\boldsymbol{\eta}_c - \frac{1}{2}\boldsymbol{\eta}_c^\top \boldsymbol{\Lambda}_c^{-1}\boldsymbol{\Lambda}_h\boldsymbol{\Lambda}_c^{-1}\boldsymbol{\eta}_c \\ &+ \boldsymbol{\eta}_h^\top \boldsymbol{\Lambda}_c^{-1}\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_c^{-1}\boldsymbol{\eta}_c \\ &- \boldsymbol{\eta}_c^\top \boldsymbol{\Lambda}_c^{-1}\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_c^{-1}\boldsymbol{\Lambda}_h\boldsymbol{\Lambda}_c^{-1}\boldsymbol{\eta}_c\Bigg\}\end{aligned} \tag{3.23}$$

We define the matrix $\mathbf{H}$ and matrix $\mathbf{L}$ row-wise as follows:

$$\mathbf{H} = \begin{bmatrix} — & \boldsymbol{\eta}_1 & — \\ & \vdots & \\ — & \boldsymbol{\eta}_{K_P} & — \end{bmatrix} \qquad \mathbf{L} = \begin{bmatrix} — & \operatorname{vec}\boldsymbol{\Lambda}_1 & — \\ & \vdots & \\ — & \operatorname{vec}\boldsymbol{\Lambda}_{K_P} & — \end{bmatrix}$$

Hence, the matrix equivalent of Equation (3.22) is the following:

$$\begin{aligned}\nabla_{\boldsymbol{\beta}_c} Q = \boldsymbol{\alpha}_c \odot \Big\{ & \mathbf{H}(\boldsymbol{y}_c - N_c \boldsymbol{\Lambda}_c^{-1} \boldsymbol{\eta}_c) \\ & + \frac{1}{2} \mathbf{L} \operatorname{vec}\left(N_c \boldsymbol{\Lambda}_c^{-1} - \mathbf{W}_c + N_c \boldsymbol{\Lambda}_c^{-1} \boldsymbol{\eta}_c \boldsymbol{\eta}_c^\top \boldsymbol{\Lambda}_c^{-1}\right) \\ & + \left[\frac{1}{2} \operatorname{Tr}\left(\boldsymbol{\Lambda}_c \mathbf{W}_c\right) - \boldsymbol{\eta}_c^\top \boldsymbol{y}_c + \frac{N_c}{2} \boldsymbol{\eta}_c^\top \boldsymbol{\Lambda}_c^{-1} \boldsymbol{\eta}_c - \frac{N_c}{2} d\right] \mathbf{J}_{K_P \times 1} \Big\}\end{aligned} \tag{3.24}$$

For convenience, define the symmetric part of a matrix $\mathbf{A}$ as $\mathbf{A}_{\text{symm}} = \frac{\mathbf{A}+\mathbf{A}^\top}{2}$. The matrix equivalent of Equation (3.23) is the following:

$$\begin{aligned}\nabla\nabla^\top{}_{\boldsymbol{\beta}_c} Q = \boldsymbol{\alpha}_c \boldsymbol{\alpha}_c^\top \odot \Big\{ & \left[2\boldsymbol{\eta}_c^\top \boldsymbol{y}_c - \operatorname{Tr}\left(\boldsymbol{\Lambda}_c \mathbf{W}_c\right) - N_c \boldsymbol{\eta}_c^\top \boldsymbol{\Lambda}_c^{-1} \boldsymbol{\eta}_c + \frac{N_c}{2} d\right] \mathbf{J}_{K_P \times K_P} \\ & + 2\left[\mathbf{H}\left(N_c \boldsymbol{\Lambda}_c^{-1} \boldsymbol{\eta}_c - \boldsymbol{y}_c\right) \mathbf{J}_{1 \times K_P}\right]_{\text{symm}} \\ & + \left[\mathbf{L} \operatorname{vec}\left(\mathbf{W}_c - N_c \boldsymbol{\Lambda}_c^{-1} \boldsymbol{\eta}_c \boldsymbol{\eta}_c^\top \boldsymbol{\Lambda}_c^{-1}\right) \mathbf{J}_{1 \times K_P}\right]_{\text{symm}} \\ & - N_c \mathbf{L}\left(\boldsymbol{\Lambda}_c^{-1} \otimes \boldsymbol{\Lambda}_c^{-1} \boldsymbol{\eta}_c \boldsymbol{\eta}_c^\top \boldsymbol{\Lambda}_c^{-1}\right) \mathbf{L}^\top \\ & - \frac{N_c}{2} \mathbf{L}\left(\boldsymbol{\Lambda}_c^{-1} \otimes \boldsymbol{\Lambda}_c^{-1}\right) \mathbf{L}^\top \\ & - N_c \mathbf{H} \boldsymbol{\Lambda}_c^{-1} \mathbf{H}^\top \\ & + 2N_c \left[\mathbf{H}\left(\boldsymbol{\Lambda}_c^{-1} \otimes \boldsymbol{\eta}_c^\top \boldsymbol{\Lambda}_c^{-1}\right) \mathbf{L}^\top\right]_{\text{symm}} \Big\} \\ & + \operatorname{diag} \nabla_{\boldsymbol{\beta}_c} Q\end{aligned} \tag{3.25}$$

Thus, using Equation (3.24) and Equation (3.25), we compute a Newton-Raphson update for $\boldsymbol{\alpha}_c$. We note that the Hessian matrix is guaranteed to have zero eigenvalue in the direction $\mathbf{1}_{K_P} = \langle 1, 1, ..., 1 \rangle$ since the objective function in $\boldsymbol{\beta}_c$ is invariant to an additive constant $\lambda \mathbf{1}_{K_P}$. This makes a strictly standard Newton-Raphson update impossible due to a singular Hessian; however, we may project to the subspace formed by the orthogonal complement of $\mathbf{1}_{K_P}$ where the Hessian has full rank. After updating in this space, we return to the full $\boldsymbol{\beta}_c$ space by setting the $\mathbf{1}_{K_P}$ component to have zero magnitude. Since the objective is invariant in this direction, choosing zero is optimal as it also improves numerical stability.

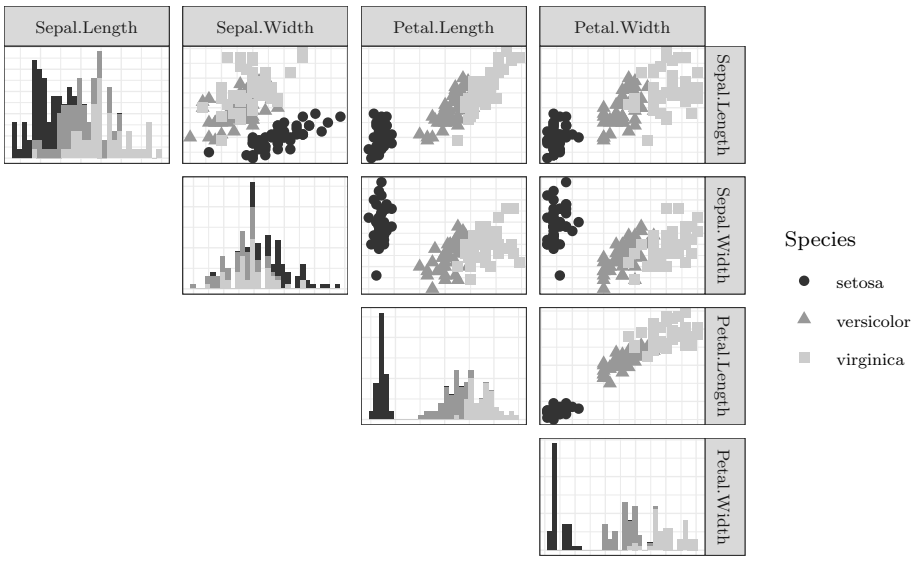

Figure 3.4: Pair-wise scatterplot matrix for the *iris* dataset

3.5 Applications

3.5.1 Iris dataset

The ubiquitous *Iris* dataset (Anderson, 1936; Fisher, 1936) describes three species of *iris* flowers. Figure 3.4 depicts the dimensions of *Iris versicolor* as intermediate to *Iris setosa* and *Iris virginica*. This data is a suitable candidate for chimeral clustering; in fact, Anderson (1936) offers a genotypical and phenotypical argument for hybridization.

To evaluate the performance of the proposed method, we evaluate each combination of $K_P + K_C \leq 6$. Using the initialization procedure in Section 4.4.1, we run all 15 starter models in mini-EM for 1000 iterations holding $\hat{z}_{nk}$ fixed for 500 of them. We discard starter models that have $N_k \leq 10^{-8}$ or any covariance with an eigenvalue $\leq 10^{-4}$. Subsequently, we run EM for up to 10,000 iterations on the best starter model. The BIC for each combination of K_P and K_C is presented in Figure 3.5 with the best BIC at $(K_P, K_C) = (2, 1)$.

Table 3.1 lists the performance metrics of the fitted chimeral clustering model and the

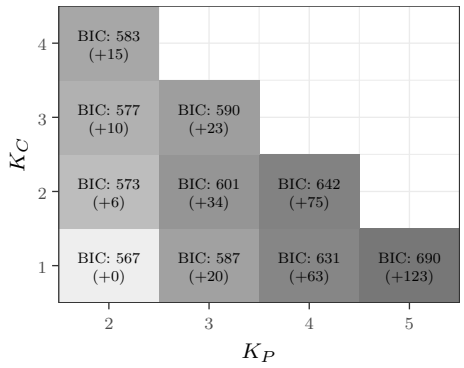

Figure 3.5: Bayesian information criterion for multiple choices of prototype clusters K_P and chimeral clusters K_C for the *iris* dataset.

best finite Gaussian mixture model of *mclust* with and without parsimonious covariance matrices. We note that *mclust* will choose a two cluster model for the *iris* dataset; for comparison purposes, three to six component models were evaluated for finite Gaussian mixtures. The chimeral model has the fewest parameters at 31; however, we see a corresponding trade-off in BIC that is partially mitigated by the reduction in complexity. BIC also indicates an improvement over the VVV model, but falls somewhat short of the best parsimonious covariance matrix model. Furthermore, the ARI with chimeral clustering is slightly better than both Gaussian mixture models and is reflected by the confusion matrix in Table 3.2.

Finally, the fitted $\boldsymbol{\alpha}_c$ for the single chimeral cluster of the best BIC run above is $\langle 0.0783, 0.9217 \rangle$. This suggests that *Iris versicolor* is much closer to *Iris virginica* than *Iris setosa*. This conclusion is consistent with the scatterplot in Figure 3.4, but is at odds with the ratio of (Anderson, 1936). We attribute this difference to the warping of the interpolated chimeral cluster caused by the differences in covariance shapes. In particular, Figure 3.2 exhibits this phenomenon as well; as α_c varies, the chimeral cluster appears more often near the lower prototype.

Table 3.1: Fitted model metrics for *iris* dataset using chimeral clustering, finite Gaussian mixtures, and finite Gaussian mixtures with parsimonious covariance matrices. Best values in bold.

		mclust	
	Chimeral Clustering	VVV	VEV
Number of Clusters	3 $(K_P = 2)$	3	3
Number of Parameters	**31**	44	38
Log-Likelihood	−206.0511	**−180.1858**	−186.0740
BIC	567.4320	580.8396	**562.5522**
ARI	**0.9410**	0.9039	0.9039

Table 3.2: Confusion matrix for *iris* dataset comparing chimeral clustering and the *mclust* VVV model with three clusters.

	Chimeral Clustering			mclust VVV		
	Prototype 1	Chimeral	Prototype 2	Cluster 1	Cluster 2	Cluster 3
setosa	50	0	0	50	0	0
versicolor	0	47	3	0	45	5
virginica	0	0	50	0	0	50

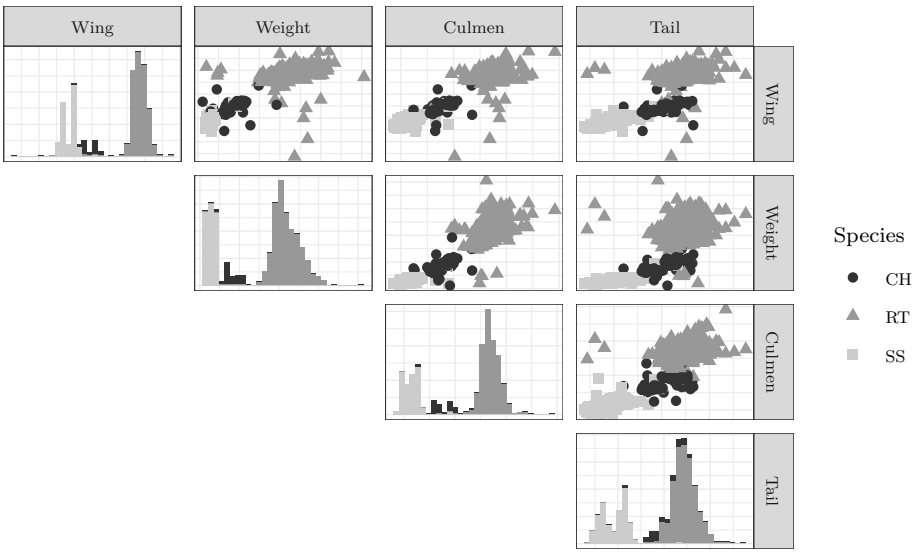

Figure 3.6: Pair-wise scatterplot matrix for the Hawks dataset.

3.5.2 Hawks dataset

The Hawks dataset (Cannon et al., 2019) describes some measurements for three different species of hawks. The dataset contains 908 observations with a multitude of variables; we consider only the four mostly-complete variables Wing, Weight, Culmen, and Tail. We further omit 16 incomplete cases. From the data scatterplot in Figure 3.6, we observe the morphometric measurements for three different species of hawks: Cooper's Hawks (CH), Red-Tailed (RT), and Sharp-Shinned (SS).

We search through parameters $K_P + K_C \leq 9$, with 1000 iterations of mini-EM. We restrict the minimum covariance eigenvalue for starter models to be $\geq 10^{-4}$ and require $N_k \geq 10^{-8}$. We run EM on the best starter for another 50,000 iterations afterwards. The results for each combination of K_P and K_C are presented in Figure 3.7. The best chimeral clustering and *mclust* model values are presented in Table 3.3, with associated confusion matrix in Table 3.4. Note that *mclust* selected the fully-varying VVV model over a parsimonious covariance structure. Finally, the chimeral cluster has hybridization

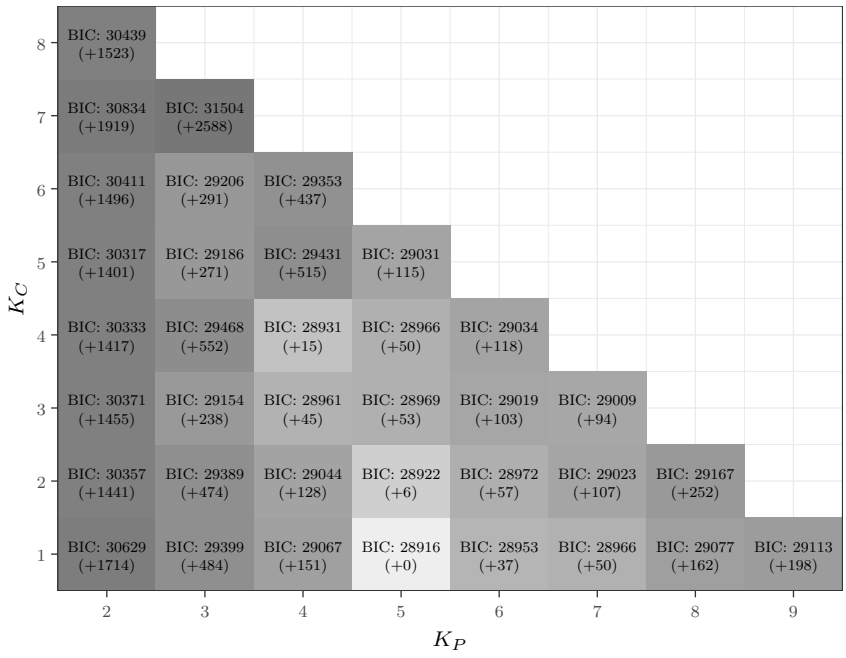

Figure 3.7: Bayesian information criterion for multiple choices of prototype clusters K_P and chimeral clusters K_C for the hawks dataset.

weights $\langle 0.051, 0.017, 0.928, 0.003, 0.000 \rangle$ corresponding to the five prototypes.

3.5.3 Limnoperus dataset

The water strider dataset (Klingenberg and Spence, 1993) measures the dimensions of genus *Limnoperus* over six different species. The measured variables for each specimen are eight morphometric dimensions comprising four antennae segment lengths and four different leg lengths.

We search through parameters $K_P \leq 10$ and $K_C \leq 10$, with 5000 iterations of mini-EM, holding $\hat{z}_{nk}$ fixed for 1000 of them. We then run EM on the best starter for another 50,000 iterations afterwards. The results for each combination of K_P and K_C are presented in Figure 3.8. The best chimeral clustering and *mclust* model values are presented in

Table 3.3: Fitted model metrics for hawks dataset using chimeral clustering, finite Gaussian mixtures, and finite Gaussian mixtures with parsimonious covariance matrices. Best values in bold. Note that *mclust* selects the VVV model over a parsimonious covariance model.

	Chimeral Clustering	mclust VVV
Number of Clusters	6 $(K_P = 5)$	6
Number of Parameters	**79**	89
Log-Likelihood	−14 189.43	**−14 172.78**
BIC	**28 915.54**	28 950.18
ARI	**0.7620**	0.4589

Table 3.4: Confusion matrix for hawks dataset comparing chimeral clustering and the *mclust* VVV model with six clusters.

	Chimeral Clustering					
	Prototype 1	Prototype 2	Prototype 3	Prototype 4	Prototype 5	Chimeral
CH	2	0	10	2	53	2
RT	0	0	46	520	2	0
SS	125	110	1	1	3	15
	mclust VVV					
	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
CH	0	53	2	1	13	0
RT	193	2	0	335	38	0
SS	1	7	130	0	6	111

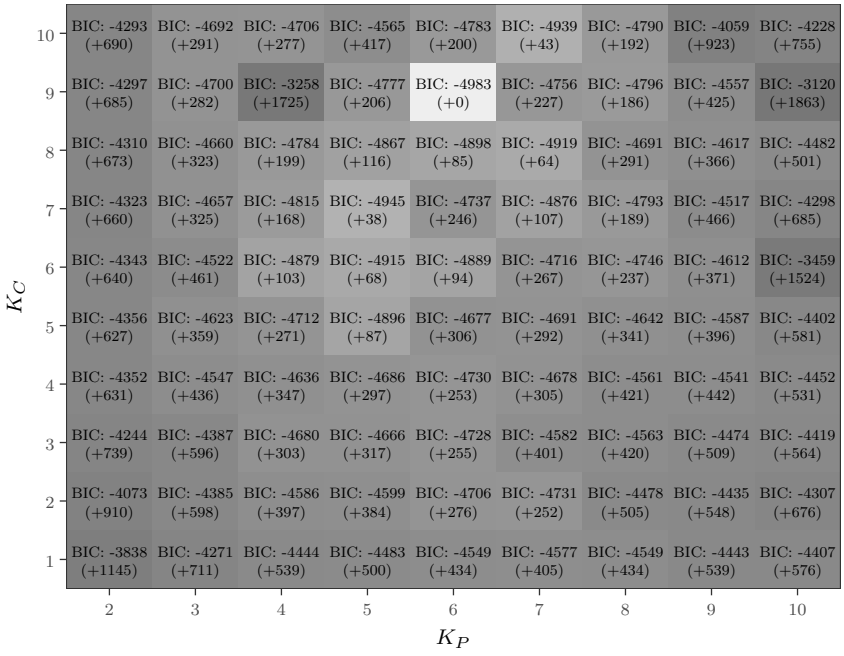

Figure 3.8: Bayesian information criterion for multiple choices of prototype clusters K_P and chimeral clusters K_C for the water strider dataset.

Table 3.5. For the chimeral clustering model with the best BIC, the hybridization weights for the chimeral clusters are visualized in Figure 3.9.

3.5.4 Yeast dataset

The yeast stress dataset (Gasch et al., 2000) is used as a real-world dataset in the prior epistatic clustering work by Zhang (2013). It describes the changes in gene expression of the yeast *Saccharomyces cerevisiae* in response to changes in the environmental conditions experienced by the cells. We attempt to replicate the same dataset by following the data pre-processing step described therein (Zhang, 2013, Section 5.1).

We begin with the dataset of Gasch et al. (2000) containing 6152 observations representing genes and 173 variables representing environmental conditions. We use the same

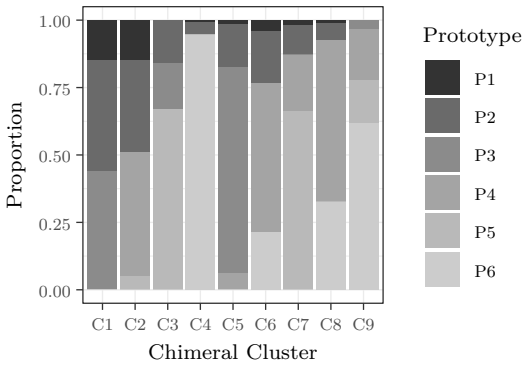

Figure 3.9: Chimeral mixing proportions $\boldsymbol{\alpha}_c$ for each of the nine chimeral clusters over the six prototype clusters for the water striders dataset.

Table 3.5: Fitted model metrics for the water strider dataset using chimeral clustering, finite Gaussian mixtures, and finite Gaussian mixtures with parsimonious covariance matrices via *mclust*. Best values in bold.

		mclust	
	Chimeral Clustering	VVV	VEE
Number of Clusters	15 $(K_P = 6)$	6	17
Number of Parameters	323	269	**204**
Log-Likelihood	**3507.46**	3074.73	3334.70
BIC	−4982.75	−4457.02	**−5385.92**
ARI	**0.1391**	0.0264	0.1304

15 variables as described by Zhang (2013), titled with the prefix "Heat Shock" and suffixed "hs-1" or "hs-2" as found in columns 4 through 19, inclusive. There are missing values in this subset of dataset and it is unclear how Zhang (2013) treats these cases; we leave incomplete observations in the dataset at this point. Subsequently, we remove noisy observations as done so by Zhang (2013), calculating a sample variance over each row. Due to data missingness, we compute the sample variance $\hat{\sigma}_i$ over the non-missing columns for each gene i. There is a single observation (YDL208W) with 14 missing values, leading to an undefined sample variance; we remove this observation. Let $S_1 = \{\hat{\sigma}_i \mid i = 1, 2, ..., 6151\}$ denote the set of sample variances. We select the subset of S_1 "within three-folds of the minimum sample variance" (Zhang, 2013) to form $S_2 = \{\hat{\sigma} \mid \hat{\sigma} \in S_1, \hat{\sigma} \leq 3 \times \min_{s \in S_1} s\}$ with $|S_2| = 169$. We define $\hat{\sigma}_0$ as the sample average over S_2, and construct the sample variability index v_i over the 6151 genes as $v_i = \hat{\sigma}_i / \hat{\sigma}_0$ for $i = 1, 2, ..., 6151$. Finally, we choose all genes i such that $v_i > 9$; this leaves 2294 genes. Since there are still missing values for these genes, we retain only complete cases for a final number of 1364 genes. By contrast, Zhang (2013) claim to have 496 genes at the end of the procedure. If we remove incomplete cases after selecting the 15 desired variables, we obtain 1361 genes at the end. If we remove incomplete cases from the entire dataset of 173 variables, we obtain 258 genes at the end. We approximately verify that our choice of 15 variables is correct by noting similar characteristics in the pairwise scatterplot (Zhang, 2013, Figure 1). Overlooking this discrepancy, we proceed with the application of chimeral clustering using the dataset of 1364 genes.

We evaluate for $K_P + K_C \leq 13$, running mini-EM for 5000 iterations and holding $\hat{z}_{nk}$ fixed for 1000 of them. We exclude models that have covariances with eigenvalues $\leq 10^{-8}$ and $N_g \leq 10^{-8}$ for any $g \in \mathcal{C} \cup \mathcal{P}$. We then run an additional 5000 iterations on the best starter model. The resultant BICs for each combination of K_P sand K_C are presented in Figure A.6.

Zhang (2013) obtains an epistatic clustering result with four primary clusters, three epistatic clusters, and a miscellaneous cluster. By contrast, we obtain three prototypes and eight chimeral clusters. The fitted model metrics are given in Table A.1 without ARI due to a lack of class labels. Again, we compare against the best fully varying covariance matrix Gaussian mixture and the best parsimonious covariance matrix Gaussian mixture fitted by

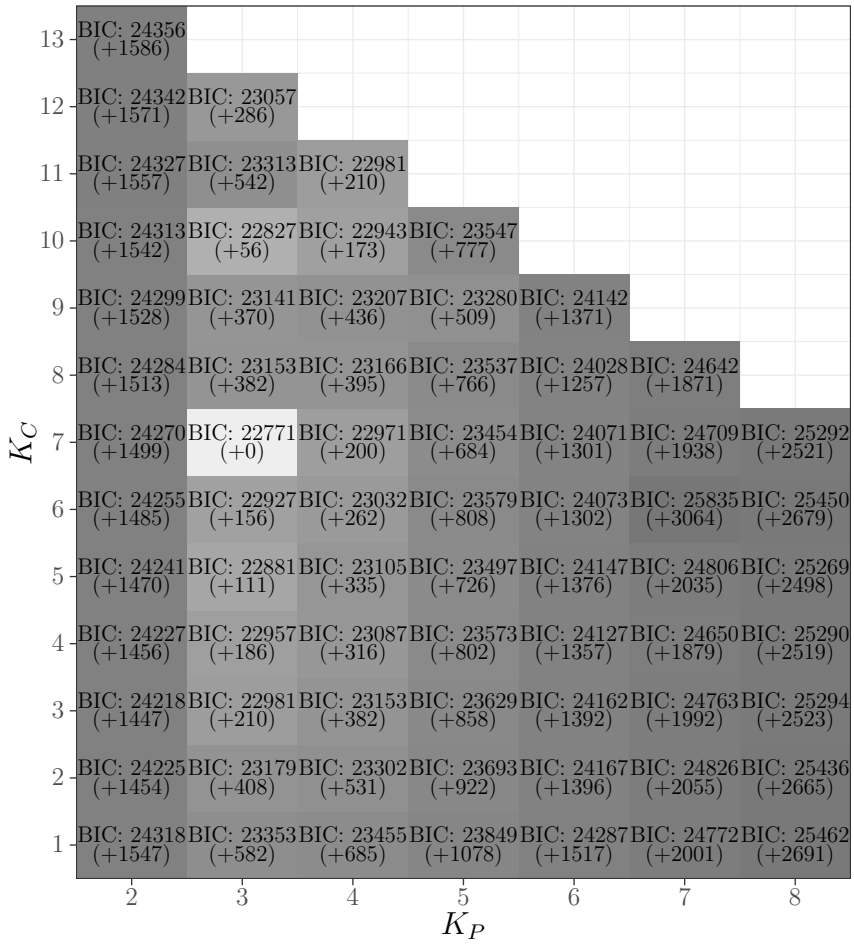

Figure 3.10: Minimum Bayesian Information Criterion for multiple choices of prototype clusters K_P and chimeral clusters K_C over 100 runs each of the *Saccharomyces cerevisiae* dataset. Graph truncated to $K_P \leq 8$ for presentation.

Table 3.6: Fitted model metrics for yeast dataset with up to 13 clusters, best value in bold.

		mclust	
	Chimeral Clustering	VVV	VEE
Number of Clusters	10 ($K_P = 3$)	4	10
Number of Parameters	428	543	**288**
Log-Likelihood	−9840.69	**−9855.30**	−10 489.53
BIC	**22 770.76**	23 630.07	23 057.90

mclust. Figure A.7 shows the $\boldsymbol{\alpha}_c$ quantities for each chimeral cluster. We can see the ability of chimeral clustering to better adapt to varying number of parents and unbalanced mixing proportions compared to the pre-specified values in epistatic clustering. With this dataset, we note that the chimeral clustering BIC outperforms both the parsimonious Gaussian mixture model and covariance VVV model.

3.5.5 Clams dataset

We demonstrate the efficacy of chimeral clustering beyond the simplest two prototype, one chimeral configuration using the data of Kitada et al. (2013b,a) which describes a collection of Manila clams (*Ruditapes philippinarum*) obtained from seas in Japan and China using the variables shell height (SH), shell width (SW), shell length (SL), and number of radial ribs found on the shell. As in the *Iris* dataset, the available variables are morphometric and thus could be expected to hybridize in the parameter space. Figure 3.12 depicts some examples of *Ruditapes philippinarum*. Kitada et al. (2013b) describes the historical import and cultivation practices which has led to present-day hybridized species taking root in different ecological niches found in different parts of the ocean. The data spans eleven geographical locations and three species, and is presented in Figure 3.13.

We evaluate for number of prototypes and chimeral clusters $K_P + K_C \leq 9$. For each combination of K_P and K_C, we run mini-EM for 1000 iterations, select the best starter model, and then run the EM algorithm for an additional 10,000 iterations. The best BIC

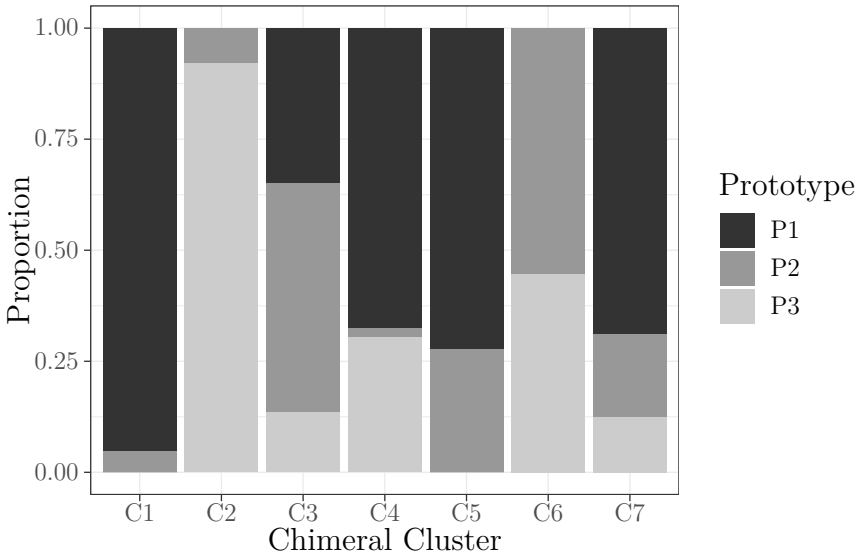

Figure 3.11: Mixing proportions $\boldsymbol{\alpha}_c$ for each of the eight chimeral clusters over the three prototype clusters for the *Saccharomyces cerevisiae* dataset. Both two and three parent clusters are visible.

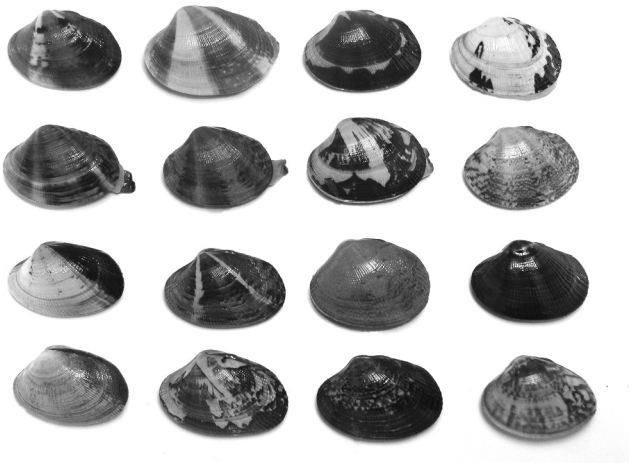

Figure 3.12: Examples of *Ruditapes philippinarum* (Takahashi, 2006)

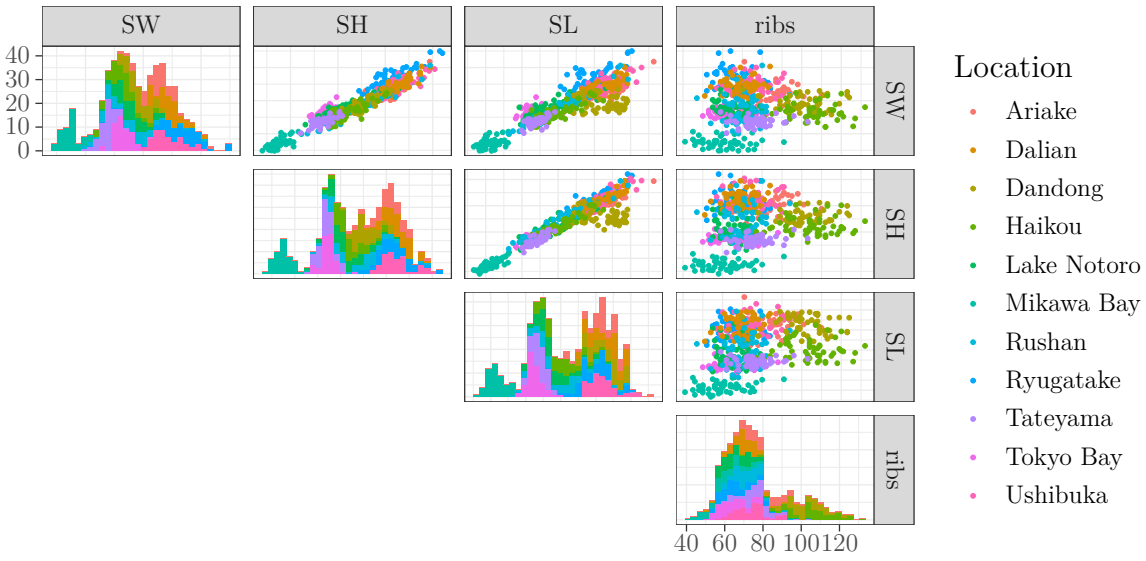

Figure 3.13: Pair-wise scatterplot matrix for the Manila clams dataset (Kitada et al., 2013a)

within each parameter combination is given in Figure 3.14. We find the best chimeral clustering model occurs at $K_P = 3$ and $K_C = 3$; the corresponding metrics are given in Table 3.7. Comparative finite Gaussian mixture models using *mclust* are also provided with and without parsimonious covariance matrices, and the model with the best BIC up to 9 clusters is displayed for both.

3.6 Simulation Study

3.6.1 Simulation Study

We describe a family of artificial datasets based on the proposed model to demonstrate the degree of parameter recovery of the weights $\boldsymbol{\alpha}_c$. We term members of this family the "d-radioactive" dataset, where the parameter d determines the dimension of the data. An example in two dimensions is presented in Figure 3.16; the data resembles a triangular

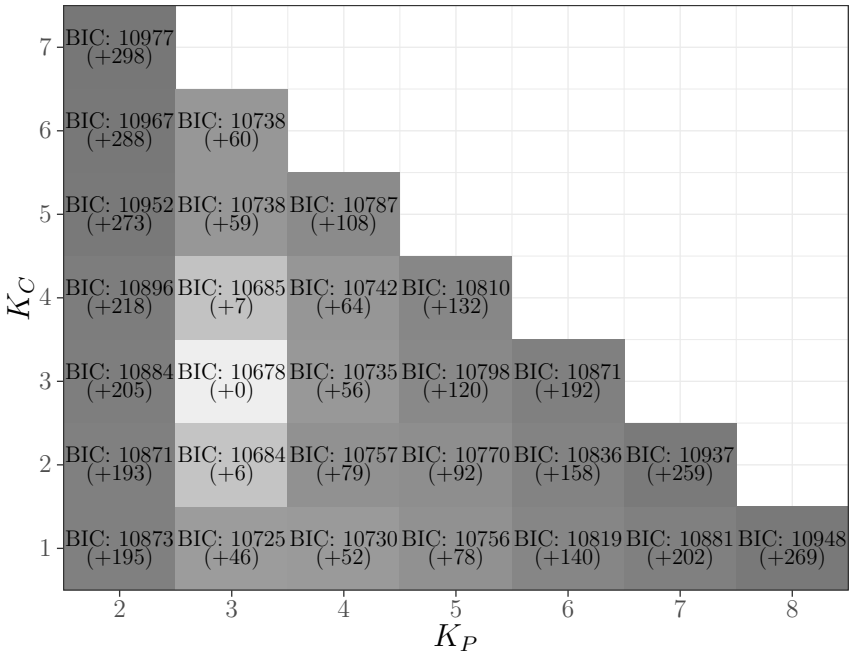

Figure 3.14: Bayesian Information Criterion for multiple choices of prototype clusters K_P and chimeral clusters K_C for the Manila clams dataset.

Table 3.7: Fitted model metrics for Manila clams dataset (Kitada et al., 2013a) with best models selected over 3 to 9 cluster models.

		mclust	
	Chimeral Clustering	VVV	VEE
Number of Clusters	6 ($K_P = 3$)	4	7
Number of Parameters	53	59	**50**
Log-Likelihood	**−5172.01**	−5196.38	−5182.32
BIC	**10 678.44**	10 765.05	10 680.13
ARI (Location)	0.2722	0.2767	**0.3472**
ARI (Species)	0.2895	**0.3313**	0.2140

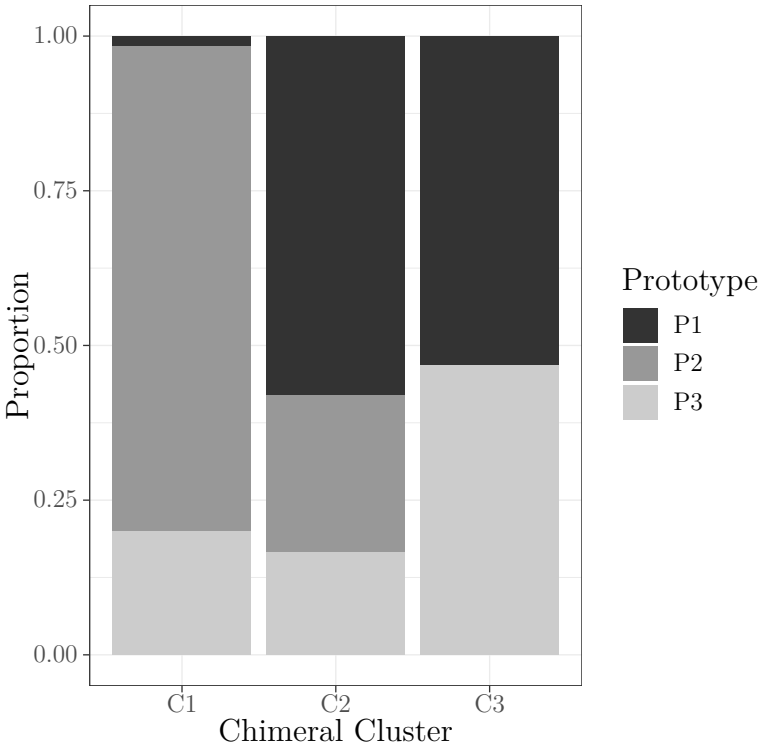

Figure 3.15: Chimeral mixing proportions $\boldsymbol{\alpha}_c$ for each of the three chimeral clusters over the three prototype clusters for the Manila clams dataset.

radioactive sign, after which the family of datasets is named. For a general $d \geq 2$, the prototype means $\boldsymbol{\mu}_p$ are placed on the regular d-simplex centred at the origin with radius $\sqrt{200d}$ in $\mathbb{R}^d$. This process yields the means of $d+1$ prototype clusters. The covariance matrix $\boldsymbol{\Sigma}_p$ is constructed such that it has $0 < r \ll 100$ eigenvalue in the direction of μ_p and 100 for the remaining eigenvalues. A formula for $\boldsymbol{\Sigma}_p$ is $100\mathbf{I}_d - (100-r)\boldsymbol{\mu}_p\boldsymbol{\mu}_p^\top/\|\boldsymbol{\mu}_p\|_2$. The parameter r controls the sphericity of the prototype clusters; smaller values r represent more flattened multivariate normal distributions.

Each prototype P_i has a corresponding chimeral cluster C_i with weight $\alpha_{C_iC_i} = \frac{3}{d+3}$, and weight $\alpha_{C_iq} = \frac{1}{d+3}$ on all other prototypes q. Finally, there is an additional central chimeral cluster C_{d+2} with equal weights on all prototypes. This yields $d+2$ chimeral clusters. From each prototype/chimeral cluster, we draw n observations to compose the dataset. Further details on the dataset construction are given in Appendix A.2.

We perform a simulation study using the described family of datasets varying three different parameters d, n, and r. We vary the data dimension d from 2 to 10, the number of observations per cluster n from 20 to 100 in steps of 20, and the eigenvalue r being 1, 5, or 10. For each of these parameter combinations, we perform ten replications. We perform mini-EM initialization with 1000 iterations, of which 500 hold $\hat{z}_{nk}$ constant, followed by 1000 further EM iterations on the best starter model. To reduce computation time, we specify K_P and K_C to be their true values to avoid a combinatoric search. A finite Gaussian mixture with parsimonious covariance matrices is fitted using *mclust* for each replication as comparison.

We measure the degree of parameter recovery for $\boldsymbol{\alpha}_c$ using cosine similarity. By expressing the hybridization weights in a vectorized form $\langle \boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_{d+2} \rangle$, we may compute the cosine of the angle between the true and estimated values. This metric provides a consistent comparison across different numbers and dimensions of $\boldsymbol{\alpha}_c$ as it varies with data dimension d. Since the estimation procedure arranges the indices haphazardly, we apply a brute-force algorithm to match the true indices by finding the permutation that maximizes the metric.

For brevity, a table summarizing some selected parameter values is presented in Ta-

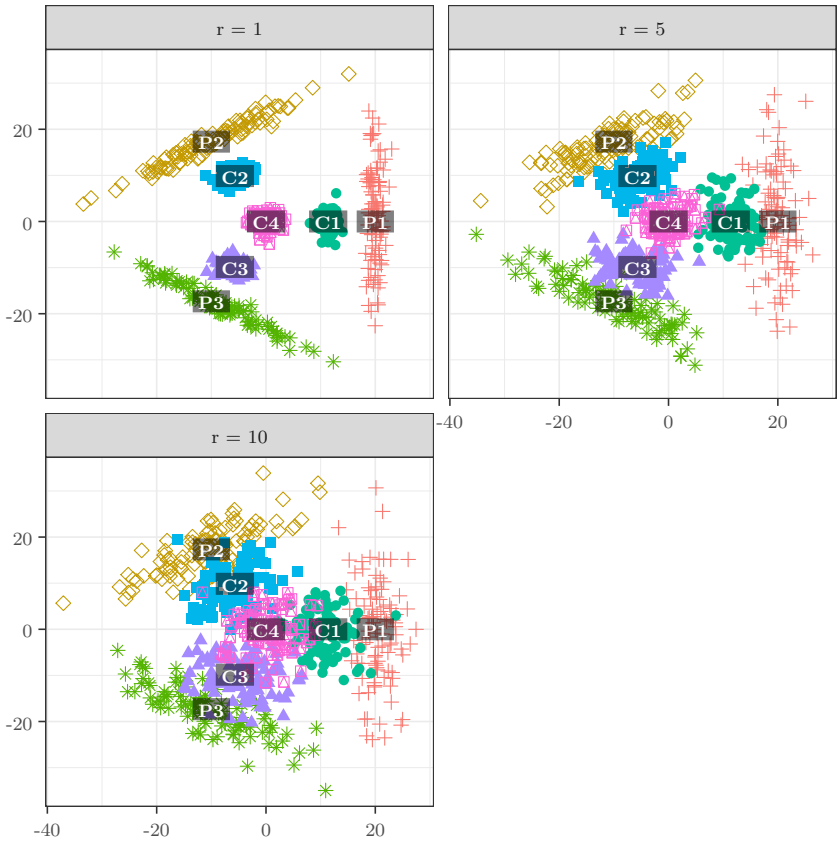

Figure 3.16: Scatterplot for one sample of the two-dimensional 2-radioactive artificial dataset. Three prototypes and four chimeral clusters are present. Three values of r are shown demonstrating the difference in cluster shapes.

ble 3.8 with extended results available in Appendix A.2.1. We observe that as r increases and the clusters begin to overlap, *mclust* models suffer a greater decrease in ARI than chimeral clustering. Conversely, we see that BIC's relative performance when parsimonious covariance matrices improves as r increases; indeed, Figure 3.16 shows the clusters' shapes becoming more spherical. In turn, this makes parameter sharing feasible and demonstrates the two different styles of parsimony found by chimeral clustering and parsimonious covariances. For both types of mixture models, ARI improves with larger n and/or d. Finally, we see that chimeral clustering has good recovery of $\boldsymbol{\alpha}_c$ except for high d and low n; the data sparsity; the lack of observations has a deleterious effect on the estimated $\boldsymbol{\alpha}_c$.

3.6.2 d-Radioactive Dataset

We describe here the method for generating the d-Radioactive dataset used in the simulation study. Here, d represents the dimension of the data. In two dimensions, we find that the data distribution resembles the radioactivity sign and is so named. In higher dimensions, the sketch of the mixture density is a regular d-simplex (triangle, tetrahedron, and so forth) with the prototypes densities roughly forming the $d-1$ dimension facets. Chimeral clusters are formed by taking three parts of one prototype and one part of all other prototypes for each prototype, with an extra cluster being equal parts of all prototypes. A constructive description follows.

For $d \geq 2$, define $d+1$ prototype clusters with their means being vertices of a regular d-dimensional simplex centered at the origin in $\mathbb{R}^d$ with radius $\sqrt{200d}$ (distance of each vertex to the origin, equivalently radius of the circumscribed d-sphere). Define $\boldsymbol{\mu}_i = \langle \mu_{i,1}, \mu_{i,2}, ..., \mu_{i,d} \rangle$ and construct successive vertices as follows.

- Set $\mu_{1,1} = 1$ and $\mu_{i,1} = -\frac{1}{d}$ for $i = 2, 3, ..., d+1$.
- For i in $2, ..., d+1$:
  - Set $\mu_{i,i} = \sqrt{1 - \sum_{j=1}^{i-1} \mu_{i,j}^2}$.
  - Set $\mu_{i+1,i}, ..., \mu_{i,d} = -\frac{1}{\mu_{i,i}} \left(\frac{1}{d} + \sum_{j=1}^{i-1} \mu_{i,j} \right)$.

Table 3.8: Average BIC, ARI, and cosine similarity values for selected simulation parameters in the d-radioactive dataset over fifty replications. Chimeral clustering and finite Gaussian mixtures with parsimonious covariances via *mclust* are compared by BIC and ARI. Cosine similarity values are provided for chimeral clustering to measure recovery of $\boldsymbol{\alpha}_c$. Sample standard deviations in given in the brackets.

Parameters			BIC		ARI		Cosine Sim.
r	d	n	Chimeral	*mclust*	Chimeral	*mclust*	Chimeral
1	2	20	1869 (52)	1913 (32)	0.986 (0.069)	0.990 (0.043)	0.974 (0.040)
1	2	100	8840 (56)	8905 (56)	0.999 (0.002)	0.999 (0.002)	0.997 (0.004)
1	10	20	34807 (218)	32748 (81)	0.993 (0.024)	0.997 (0.004)	0.613 (0.143)
1	10	100	158289 (197)	158020 (201)	1.000 (0.000)	1.000 (0.001)	0.988 (0.009)
5	2	20	2177 (25)	2185 (19)	0.803 (0.097)	0.561 (0.095)	0.915 (0.085)
5	2	100	10412 (48)	10484 (46)	0.880 (0.039)	0.799 (0.090)	0.983 (0.046)
5	10	20	38413 (128)	35818 (101)	0.924 (0.049)	0.893 (0.027)	0.576 (0.096)
5	10	100	175513 (236)	173727 (239)	0.983 (0.003)	0.926 (0.009)	0.988 (0.005)
10	2	20	2237 (20)	2227 (21)	0.562 (0.083)	0.454 (0.039)	0.907 (0.067)
10	2	100	10779 (39)	10808 (38)	0.676 (0.036)	0.503 (0.051)	0.991 (0.009)
10	10	20	39289 (113)	36606 (100)	0.725 (0.057)	0.706 (0.039)	0.573 (0.066)
10	10	100	180546 (211)	177897 (223)	0.880 (0.010)	0.803 (0.012)	0.976 (0.015)

For each prototype, define the covariance matrix $\mathbf{\Sigma}_i = 100\mathbf{I} - (100 - r)\mu_i\mu_i^\top$ for some parameter $0 < r < 100$. In order to preserve the separation of the clusters, r should be considerably lower than 100. Let the natural parameterization of the prototype distributions be $(\boldsymbol{\eta}_i, \mathbf{\Lambda}_i)$. Define $d + 2$ chimeral clusters, with the first $j = 1, 2, ..., d + 1$ being parameterized by $\boldsymbol{\alpha}_j = \frac{1}{d+3}\mathbf{1}_{d+1} + \frac{2}{d+3}\boldsymbol{e}_j$ for standard basis vectors $\boldsymbol{e}_j$. The last chimeral cluster is parameterized by $\boldsymbol{\alpha}_j = \frac{1}{d+1}\mathbf{1}_{d+1}$.

For $d = 2$, the parameters in numerical form are:

$$
\begin{aligned}
\boldsymbol{\eta}_{P_1} &= \langle 20, 0 \rangle \\
\boldsymbol{\eta}_{P_2} &= \langle -10, 17.3205 \rangle \\
\boldsymbol{\eta}_{P_3} &= \langle -10, -17.3205 \rangle \\
\mathbf{\Lambda}_{P_1} &= \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix} \\
\mathbf{\Lambda}_{P_2} &= \begin{bmatrix} 0.2575 & -0.4287 \\ -0.4287 & 0.7525 \end{bmatrix} \\
\mathbf{\Lambda}_{P_3} &= \begin{bmatrix} 0.2575 & 0.4287 \\ 0.4287 & 0.7525 \end{bmatrix} \\
\boldsymbol{\alpha}_{C_1} &= \langle 0.6, 0.2, 0.2 \rangle \\
\boldsymbol{\alpha}_{C_2} &= \langle 0.2, 0.6, 0.2 \rangle \\
\boldsymbol{\alpha}_{C_3} &= \langle 0.2, 0.2, 0.6 \rangle \\
\boldsymbol{\alpha}_{C_4} &= \langle 0.\bar{3}, 0.\bar{3}, 0.\bar{3} \rangle
\end{aligned}
$$

A plot of 1000 observations drawn from each cluster is given in Figure A.1. A three-dimensional version is visualized in Figure A.2 with 200 observations per cluster. In both figures, $r = 1$.

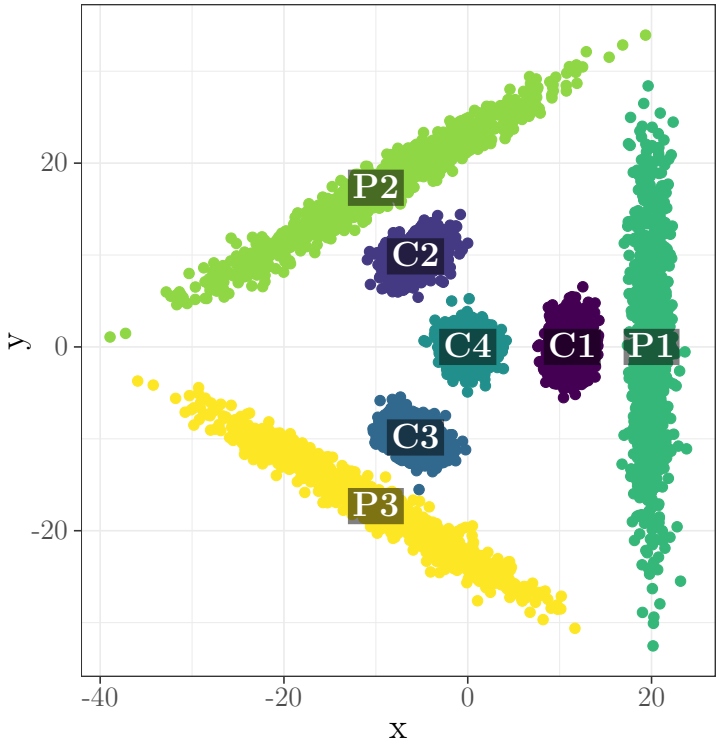

Figure 3.17: 2-dimensional radioactive dataset, 1000 observations per cluster. $r = 1$.

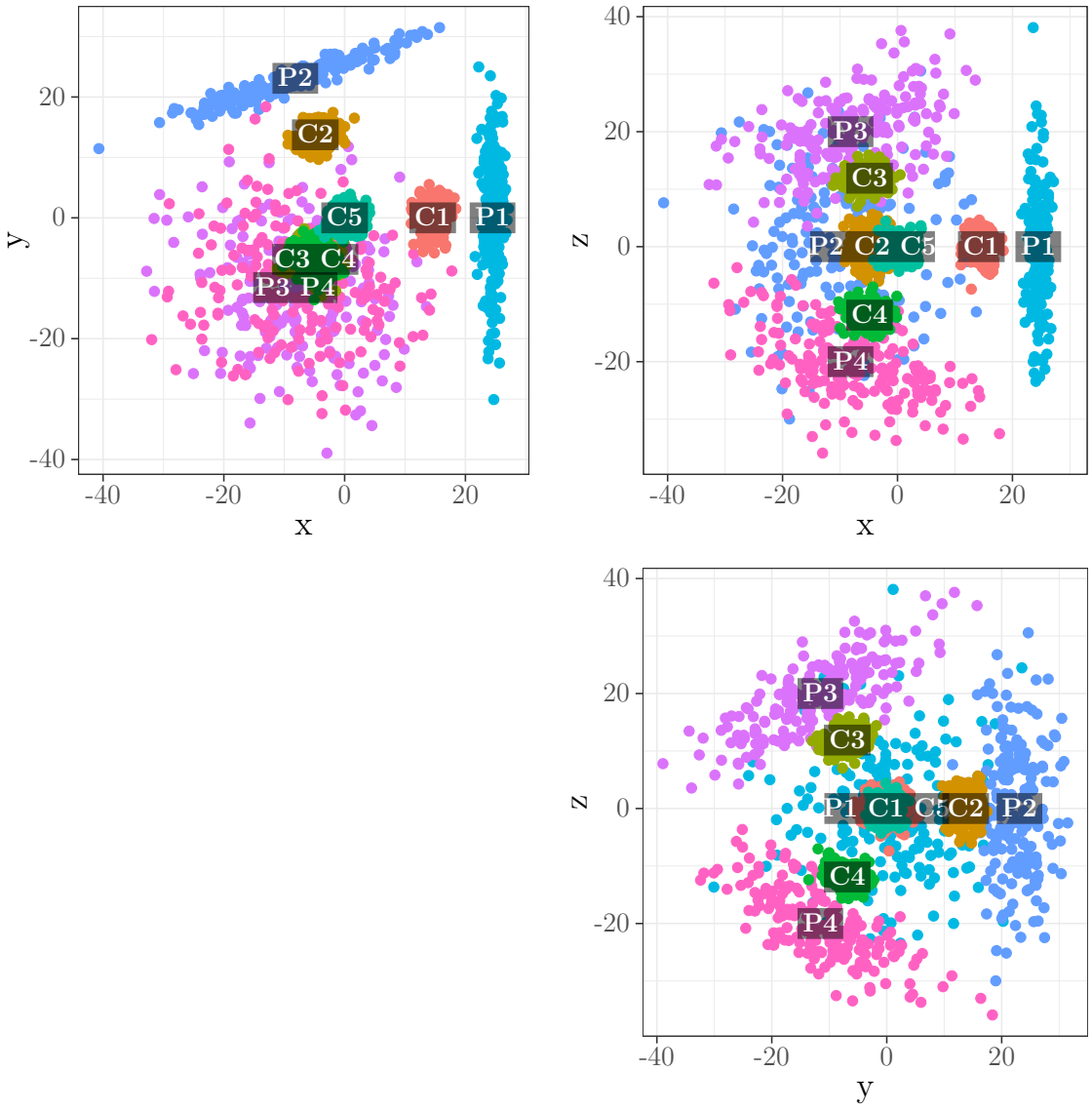

Figure 3.18: 3-dimensional radioactive dataset, 200 observations per cluster. $r = 1$.

3.6.3 Extended Simulation Results

A selection of these results are given in tabular form in the main work. We present here the full simulation results for $d = 2, 3, ..., 10$, $n = 20, 40, ..., 100$, and $r = 1, 5, 10$.

Cosine Similarity

The cosine similarities are defined for chimeral clustering as follows. Let $\boldsymbol{\alpha}_c$ be the true hybridization weights for cluster $c \in \mathcal{C}$ and let $\hat{\boldsymbol{\alpha}}_c$ be the estimated hybridization weights from the estimation procedure. Then, the cosine similarity of the entire fitted model could be computed as

$$\frac{\begin{bmatrix} \boldsymbol{\alpha}_{C_1} \\ \vdots \\ \boldsymbol{\alpha}_{C_{K_C}} \end{bmatrix}^\top \begin{bmatrix} \hat{\boldsymbol{\alpha}}_{C_1} \\ \vdots \\ \hat{\boldsymbol{\alpha}}_{C_{K_C}} \end{bmatrix}}{\left\| \begin{bmatrix} \boldsymbol{\alpha}_{C_1} \\ \vdots \\ \boldsymbol{\alpha}_{C_{K_C}} \end{bmatrix} \right\|_2 \left\| \begin{bmatrix} \hat{\boldsymbol{\alpha}}_{C_1} \\ \vdots \\ \hat{\boldsymbol{\alpha}}_{C_{K_C}} \end{bmatrix} \right\|_2}.$$

If the two sets of weights $\{\boldsymbol{\alpha}_c\}_{c\in\mathcal{C}}$ and $\{\hat{\boldsymbol{\alpha}}_c\}_{c\in\mathcal{C}}$ coincide, then the angle formed in between them is zero and so their cosine similarity is one. As the estimated vector deviates, the similarity metric decreases towards zero. However, this process requires the estimated indices $C_1, ..., C_{K_C}$ to match the true indices, something not guaranteed by the estimation procedure. Thus, we permute the estimated weights to maximize this quantity.

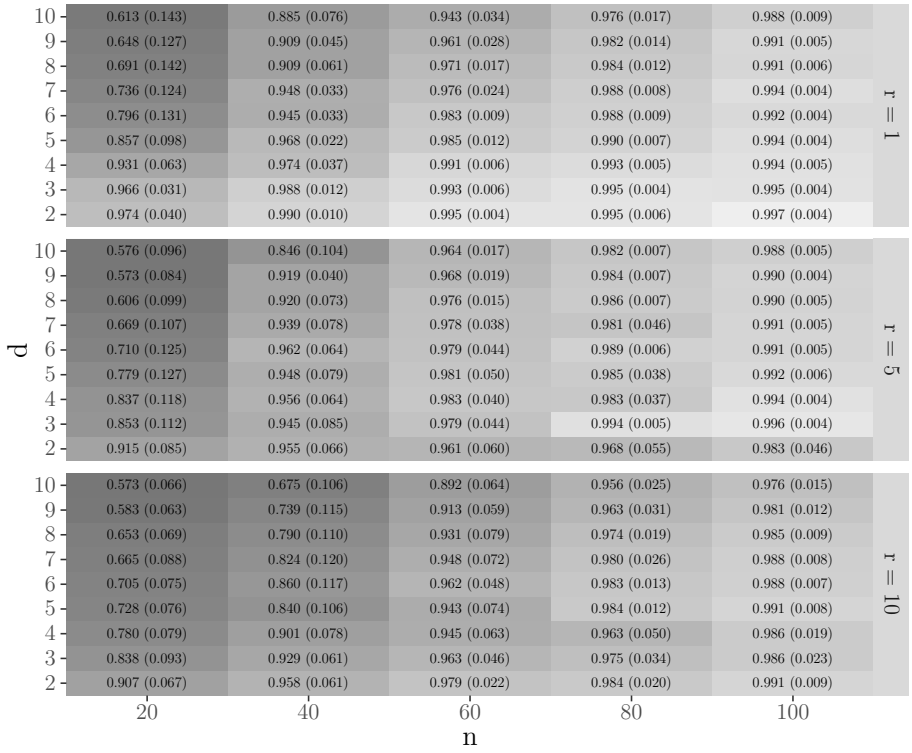

Figure 3.19: Cosine similarity values measuring the $\boldsymbol{\alpha}_c$ parameter recovery in the d-radioactive dataset over a range of data dimensions d, number of observations n, and prototype sphericity r. Higher values are better. Standard deviations over fifty replications in brackets.

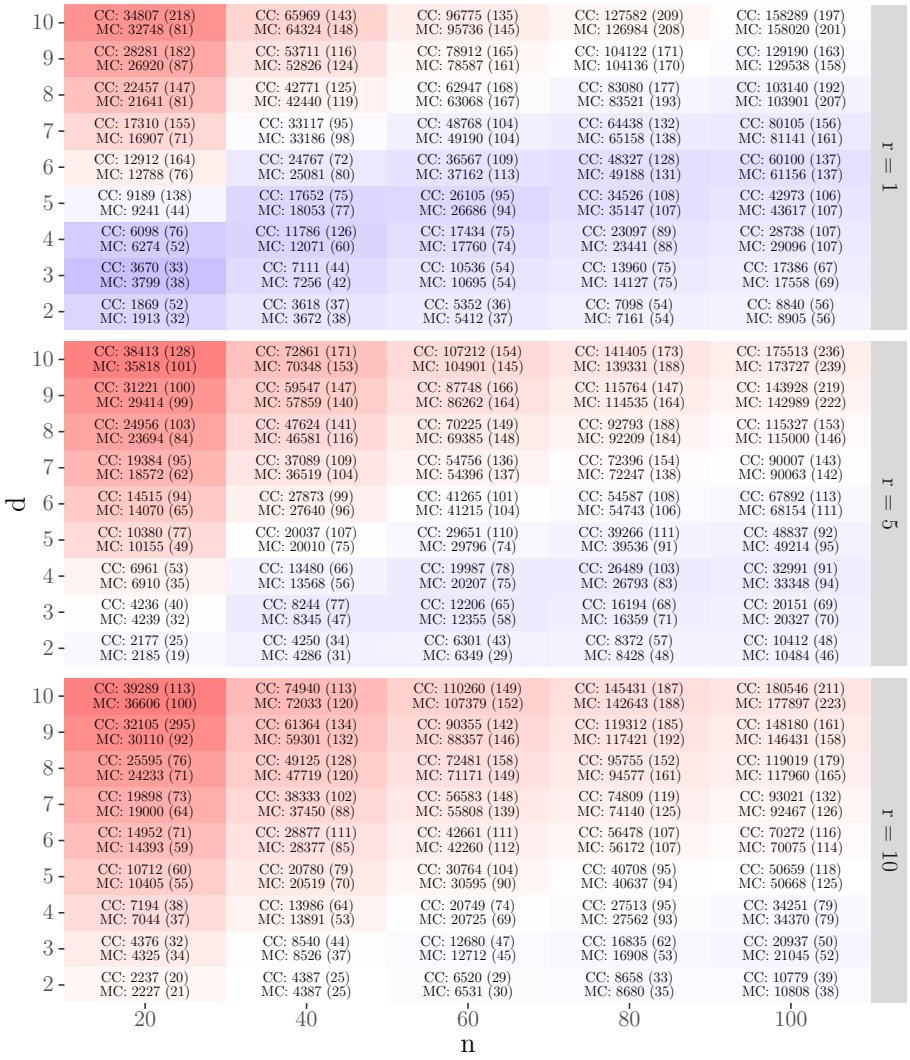

Figure 3.20: Bayesian Information Criterion values for the d-radioactive dataset over multiple parameter combinations. Both chimeral clustering (CC) and finite Gaussian mixtures with parsimonious covariances via *mclust* are presented; lower values are better. Standard deviations over fifty replications in brackets.

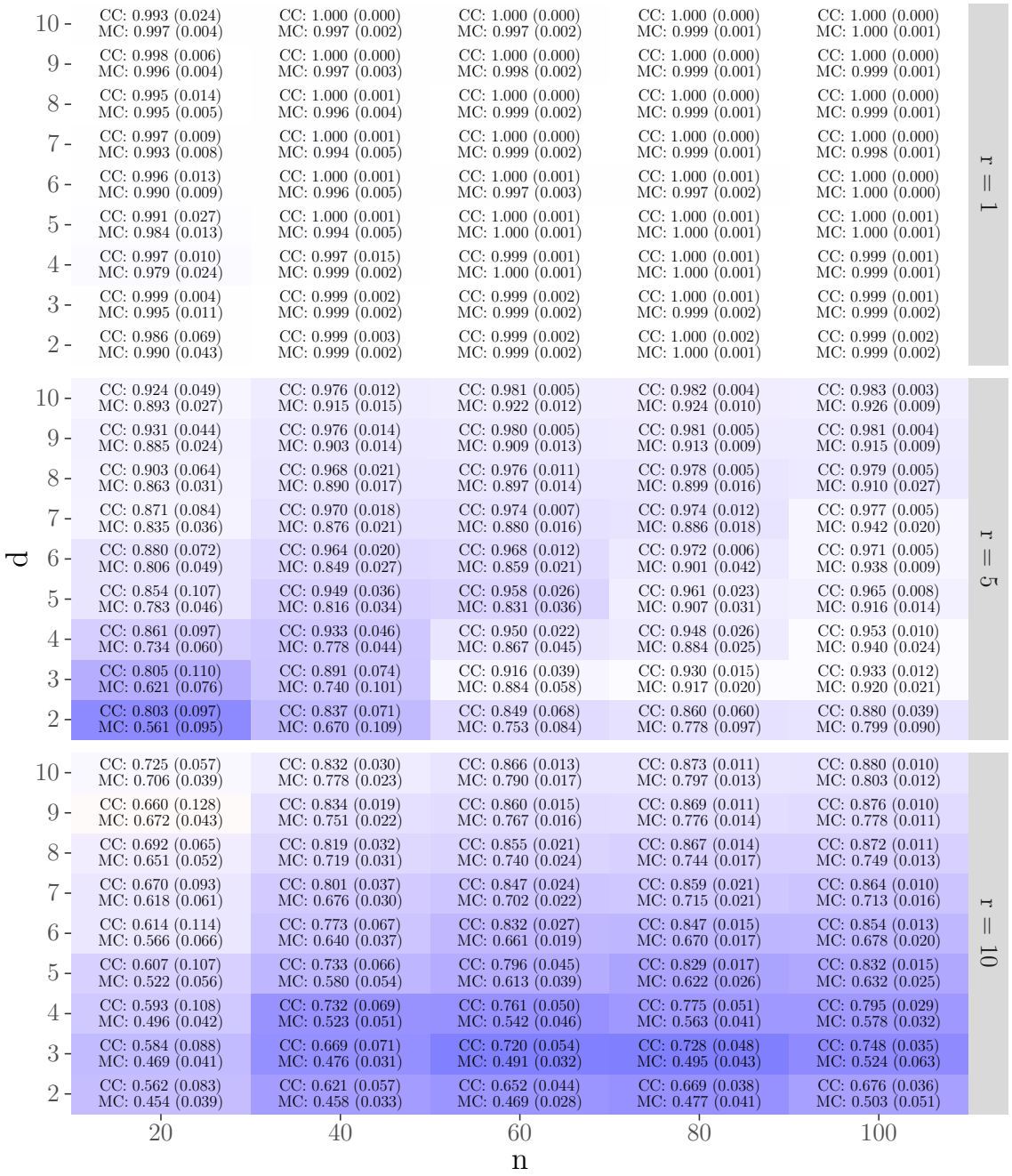

Figure 3.21: Adjusted Rand index values for the d-radioactive dataset over multiple parameter combinations. Both chimeral clustering (CC) and finite Gaussian mixtures with parsimonious covariances via *mclust* are presented; higher values are better. Standard deviations over fifty replications in brackets.

Chapter 4

Factor and Hybrid Components for Model-Based Clustering

4.1 Introduction

Our proposition in this model is an improvement over Chimeral Clustering in area of parameterization; this model uses the more interpretable moment parameters of the multivariate normal. This can be considered more useful for analyses where the hybridisation in the mean and covariances have a more palatable interpretation.

4.2 Relation to Existing Models

The above proposed model resembles the Epistatic Clustering model of Zhang (2013). Of the models described in the literature, this is the only model to hybridize at the component level. The remaining models produce a distinct set of hybridisation weights for each observation. Between these two paradigms, the former claims that there is a cluster of hybrids while the latter claims that individuals are hybrids of clusters.

The EC model describes epistatic clusters as multivariate normal distributions with parameters being an average of other parent cluster parameters. However, the EC model also imposes constraints on the weights. In the factor-hybrid notation, if an epistatic cluster h has $p \leq F$ parents, the p values of $\boldsymbol{\alpha}_h$ corresponding to parent cluster indices take value $1/p$ with remaining weights being zero. This restriction is lifted in the proposed model.

Moreover, both the BPM and EC models treat their distributions in the canonical parameters of the multivariate normal distribution instead of the moment parameters. While both parameterisations fully describe a multivariate normal distribution, moment parameters are more easily interpreted as measures of central tendency and spread.

The factor-hybrid stochastic relationship (4.1) is reminiscient of factor analysis models. In factor analysis, each observation is described in terms of loadings on some latent factors and an error term. The proposed model applies this concept to component distributions; a hybrid component realisation Y_h has convex loading weights $\boldsymbol{\alpha}_h$ on some factor components Y_f for $f \in \mathcal{F}$. Indeed, the estimation procedure bears resemblance to that of mixtures of factor analyzers in Ghahramani et al. (1996).

It is known that finite mixtures of multivariate normal distributions are identifiable (Yakowitz and Spragins, 1968; Teicher, 1961; Holzmann et al., 2006) up to a permutation of component indices. The mixture density is identifiable in terms of parameters $\left\{\boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f\right\}$ for $f \in \mathcal{F}$ and $\{\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h\}$ for $h \in \mathcal{H}$ so long as they are unique. As $\boldsymbol{\mu}_h$ is identifiable, $\boldsymbol{\alpha}_h$ is identifiable with the condition that $\boldsymbol{\alpha}_h^\top \boldsymbol{\alpha}_h$ is minimized. Identifiability of $\boldsymbol{\Psi}_h$ follows from the implied $\boldsymbol{\Sigma}_h$.

4.3 Model Specification

The model proposed in the present work is an evolution upon the idea of hybridisation of pure types or parent clusters from Section 3.1.1.

Suppose we have a real-valued dataset with N observations $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$, each a d-length vector. Let z_{ng} be the indicator for the membership of observation $n = 1, 2, ..., N$ to

component $g = 1, 2, ..., G$, with each observation being a member of exactly one component. Finally, let $\pi_1, ..., \pi_G > 0$ with $\sum_{g=1}^{G} \pi_g = 1$ be the mixing proportions of the components. Instead of all G components being treated symmetrically with one another as in finite Gaussian mixtures, we partition the set $\mathcal{G} = \{1, 2, ..., G\}$ into two index sets $\mathcal{F}$ and $\mathcal{H}$ denoting factors and hybrids, respectively. We let the number of factor components be $F = |\mathcal{F}|$ and the number of hybrid components be $H = |\mathcal{H}|$.

A factor component, indexed by $f \in \mathcal{F}$, is identical to a finite Gaussian mixture component; it is parameterised by a mean $\boldsymbol{\mu}_f$ and a covariance $\boldsymbol{\Sigma}_f$. Thus, $z_{nf} = 1$ for $f \in \mathcal{F}$ means $\boldsymbol{x}_n \sim \mathrm{N}(\boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f)$.

However, hybrid components indexed by $h \in \mathcal{H}$ are substantially different. When an observation $\boldsymbol{x}_n$ belongs to component $h \in \mathcal{H}$; that is, $z_{nh} = 1$, we assert that it is drawn from the stochastic relation

$$Y_h = \sum_{f \in \mathcal{F}} \alpha_{hf} Y_f + E_h, \tag{4.1}$$

for some latent factor representations $Y_f \sim \mathrm{N}(\boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f)$ and noise term $E_h \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Psi}_h)$ with diagonal covariance $\boldsymbol{\Psi}_h$. Thus, a hybrid component $h \in \mathcal{H}$ is said to be parameterised by the factor loadings $\boldsymbol{\alpha}_h = \langle \alpha_{h1}, ..., \alpha_{hF} \rangle \succeq 0$ with $\mathbf{1}_F^\top \boldsymbol{\alpha}_h = 1$ and a diagonal covariance matrix $\boldsymbol{\Psi}_h$. From the stochastic relation (4.1), we obtain

$$\boldsymbol{x}_n \mid z_{nh} = 1 \sim N\Bigg(\underbrace{\sum_{f \in \mathcal{F}} \alpha_{hf} \boldsymbol{\mu}_f}_{\boldsymbol{\mu}_h}, \underbrace{\sum_{f \in \mathcal{F}} \alpha_{hf}^2 \boldsymbol{\Sigma}_f + \boldsymbol{\Psi}_h}_{\boldsymbol{\Sigma}_h} \Bigg), \tag{4.2}$$

which paves the way to an efficient estimation procedure in the style of the mixture of factor analyzers model of Ghahramani et al. (1996). The mixture density is

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{f \in \mathcal{F}} \pi_f \phi_d \left(\boldsymbol{x}; \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f \right) + \sum_{h \in \mathcal{H}} \pi_h \phi_d \left(\boldsymbol{x}; \sum_{f \in \mathcal{F}} \alpha_{hf} \boldsymbol{\mu}_f, \sum_{f \in \mathcal{F}} \alpha_{hf}^2 \boldsymbol{\Sigma}_f + \boldsymbol{\Psi}_h \right). \tag{4.3}$$

The hybrid clustering model described above contains the parameters $\boldsymbol{\theta} = \left\{ \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f, \boldsymbol{\Psi}_h, \alpha_{hf}, \boldsymbol{\pi} \right\}$ for $f \in \mathcal{F}$ and $h \in \mathcal{H}$.

4.3.1 Parsimonious Noise Distribution Specifications

The model presented above specifies a general positive diagonal matrix $\mathbf{\Psi}_h$ distinct for each $h \in \mathcal{H}$ to allow for idiosyncratic error distributions for each hybrid component. However, this flexibility adds d parameters per hybrid cluster. Hence, we consider some alternate specifications that share parameters across hybrid components.

- EV: $\mathbf{\Psi}_h = \mathbf{\Psi}_h$; idiosyncratic diagonal error covariance.
- EE: $\mathbf{\Psi}_h = \mathbf{\Psi}$; shared diagonal error covariance.
- IV: $\mathbf{\Psi}_h = \psi_h \mathbf{I}_d$; idiosyncratic spherical error covariance.
- IE: $\mathbf{\Psi}_h = \psi \mathbf{I}_d$; shared spherical error covariance.
- C: $\mathbf{\Psi}_h = \varepsilon \mathbf{I}_d$; constant error covariance for some fixed $\varepsilon > 0$.

We may also interpret these error covariance types; when the hybrid components are expected to have a common error term, EE or IE errors are appropriate. When the errors have similar magnitude such as the measurement units being consistent across variables, then IV or IE errors are appropriate. To approximate the effect of $\mathbf{\Psi}_h = \mathbf{0}$, we may choose $\mathbf{\Psi}_h = \varepsilon \mathbf{I}_d$ with a very small ε such as 10^{-10}.

In the case of a single hybrid component, IV and EV are redundant and equivalent to IE and EE, respectively. To avoid confusion, we exclude IV and EV from the potential noise covariance specifications when $H = 1$.

4.3.2 Parsimonious Covariance Specifications

In the same manner of parsimony achieved by Celeux and Govaert (1995), we may share parameters between the factor component covariance matrices $\mathbf{\Sigma}_f$ for $f \in \mathcal{F}$. In these methods, geometric redundancies in shape, size, and/or orientation are exploited to represent these covariance matrices with fewer parameters. We consider the 14 covariance

Table 4.1: List of Parsimonious Covariance Matrices Types

Model	**Specification of $\boldsymbol{\Sigma}_g$**	**Factor Covariance Parameters**
EII	$\lambda\mathbf{I}$	1
VII	$\lambda_g\mathbf{I}$	F
EEI	$\lambda\mathbf{A}$	d
VEI	$\lambda_g\mathbf{A}$	$d+F-1$
EVI	$\lambda\mathbf{A}_g$	$dF-F+1$
VVI	$\lambda_g\mathbf{A}_g$	dF
EEE	$\lambda\mathbf{D}\mathbf{A}\mathbf{D}^\top$	$d\left(d+1\right)/2$
EEV	$\lambda\mathbf{D}_g\mathbf{A}\mathbf{D}_g^\top$	$Fd\left(d+1\right)/2-d\left(F-1\right)$
VEV	$\lambda_g\mathbf{D}_g\mathbf{A}\mathbf{D}_g^\top$	$Fd\left(d+1\right)/2-\left(d-1\right)\left(F-1\right)$
VVV	$\lambda_g\mathbf{D}_g\mathbf{A}_g\mathbf{D}_g^\top$	$Fd\left(d+1\right)/2$
EVE	$\lambda\mathbf{D}\mathbf{A}_g\mathbf{D}^\top$	$d\left(d+1\right)/2+\left(d-1\right)\left(F-1\right)$
VVE	$\lambda_g\mathbf{D}\mathbf{A}_g\mathbf{D}^\top$	$d\left(d+1\right)/2+d\left(F-1\right)$
VEE	$\lambda_g\mathbf{D}\mathbf{A}\mathbf{D}^\top$	$d\left(d+1\right)/2+\left(F-1\right)$
EVV	$\lambda\mathbf{D}_g\mathbf{A}_g\mathbf{D}_g^\top$	$Fd\left(d+1\right)/2-\left(F-1\right)$

decompositions (Celeux and Govaert, 1995) in Table 4.1. Of note is that performing parameter sharing in the factor covariances does not guarantee that hybrid covariances can be decomposed in the same way; parameter sharing for $\boldsymbol{\Sigma}_h$ is done through the interpolation coefficients $\boldsymbol{\alpha}_h$. This combines the parameter reducing effect of hybrid clusters with the same effect of parsimonious covariance matrices.

We may continue to interpret the eigen-decomposition and parameter sharing for factor covariances in the same way as Celeux and Govaert (1995). Moreover, with some choices of parsimonious factor covariance specifications, these parameter sharing traits extend to the hybrid components arizing therefrom, if we ignore the contribution of the error term. For example, selecting VEE for factors will induce the same $\mathbf{D}\mathbf{A}\mathbf{D}^\top$ in hybrids, though the size λ_h will differ. By choosing a noise term of type C and $\varepsilon \approx 0$, this holds approximately.

4.4 Estimation

We define here an Expectation-Maximization algorithm for the factor-hybrid clustering model. As alluded to in Section 4.3, the estimation procedure bears resemblance to that of Ghahramani et al. (1996). The estimable parameters are comprised of factor component parameters $\left(\boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f\right)$, interpolation coefficients α_{hf}, error distribution covariances $\boldsymbol{\Psi}_h$ and the mixing coefficients $\boldsymbol{\pi}$. The complete data likelihood function for the mixture density (4.3) is thus

$$\mathcal{L}(\boldsymbol{\Theta}; \mathbf{X}, \mathbf{Z}) = \prod_{n=1}^{N} \left\{ \prod_{f \in \mathcal{F}} \left[\pi_f \phi_d \left(\boldsymbol{x}_n; \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f \right) \right]^{z_{nf}} \right\} \left\{ \prod_{h \in \mathcal{H}} \left[\pi_h \phi_d \left(\boldsymbol{x}_n; \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h \right) \right]^{z_{nh}} \right\}. \tag{4.4}$$

To ease the estimation procedure, we decompose the hybrid components into their latent representation in terms of factor components. Specifically, we treat these latent representations as missing data in addition to missing membership labels z_{ng}, and assume independence of factor random variables. We may then replace the hybrid density $\phi_d \left(\boldsymbol{x}_n; \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h \right)$ for observation n with the product of the conditional density of the latent values $\{\mathbf{y}_{nf}\}_{f \in \mathcal{F}}$ and the corresponding marginal densities $\phi_d \left(\mathbf{y}_{nf}; \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f \right)$. As a shorthand, we abbreviate the condition of being given $\{\mathbf{y}_{nf}\}_{f \in \mathcal{F}}$ as being simply given $\mathbf{y}_n$. Thus, we have that

$$\phi_d \left(\boldsymbol{x}_n \mid z_{nh} = 1; \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h \right) = \phi_d \left(\boldsymbol{x}_n \mid \mathbf{y}_n, z_{nh} = 1; \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h \right) \prod_{f \in \mathcal{F}} \phi_d \left(\mathbf{y}_{nf} \mid Y_h = \boldsymbol{x}_n, z_{nh} = 1; \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f \right).$$

The conditional distribution of $\boldsymbol{x}_n \mid \mathbf{y}_n, z_{nh} = 1$ is $N\left(\sum_{f \in \mathcal{F}} \alpha_{hf} \mathbf{y}_{nf}, \boldsymbol{\Psi}_h \right)$, and the marginal density in factors decomposes into the product due to the assumed independence of factors. We may then re-write the complete likelihood (4.4) with this substitution to obtain

$$\mathcal{L}(\boldsymbol{\Theta}; \mathbf{X}, \mathbf{Z}) = \prod_{n=1}^{N} \prod_{f \in \mathcal{F}} \left[\pi_f \phi_d \left(\boldsymbol{x}_n; \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f \right) \right]^{z_{nf}} \prod_{h \in \mathcal{H}} \left[\pi_h \phi_d \left(\boldsymbol{x}_n \mid \mathbf{y}_n; \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h \right) \prod_{f \in \mathcal{F}} \phi_d \left(\mathbf{y}_{nf}; \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f \right) \right]^{z_{nh}}. \tag{4.5}$$

We are now ready to perform the EM algorithm of Dempster et al. (1977). For brevity, an abridged version of the corresponding expectation and maximization steps is provided herein. A full derivation is available in Appendix C.

4.4.1 Initialization

This section outlines a heuristic for initializing the factor-hybrid clustering model parameters. To generate randomised starting values, we start with a random subset of half of the data and fit a finite Gaussian mixture with $G = F + H$ components and the specified covariance type using the *mclust* package. When *mclust* with the chosen covariance type fails, we resort to searching over all 14 covariance types and select the best BIC among them. We treat the estimated mean vectors $\left\{ \boldsymbol{\mu}_g \right\}_{g=1}^{G}$ as a set of vertices in $\mathbb{R}^d$, and find the convex hull of these points using the $\mathcal{V}$-representation (Gruber, 2007; Grünbaum, 2003). If the number of extremal points of the convex hull is less than F, increment G by one and refit the Gaussian mixture. Repeat these steps until there are at least F vertices comprising the convex hull of mean vectors. Let the set of vertices defining the convex hull be $\hat{\mathcal{C}}$. If there are more than F vertices in the convex hull, search through all subsets of these extremal points of size F, and determine:

$$\underset{\mathcal{F} \subseteq \hat{\mathcal{C}}, |\mathcal{F}| = F}{\arg\min} \sum_{h \in \hat{\mathcal{C}}, h \notin \mathcal{F}} \left(\min_{\boldsymbol{\alpha}_h} \left\| \boldsymbol{\mu}_h - \sum_{f \in \mathcal{F}} \alpha_{hf} \boldsymbol{\mu}_f \right\|^2 \text{ subject to } \boldsymbol{\alpha}_h \succeq 0 \text{ and } \mathbf{1}^\top \boldsymbol{\alpha}_h = 1 \right).$$

Effectively, this searches for the lowest sum square error for approximating extremal points in $\hat{\mathcal{C}} \setminus \mathcal{F}$ as a convex combination of extremal points in $\mathcal{F}$. The subset $\mathcal{F}$ that minimizes this error will be used to index the factor components. Among the $G - F$ remaining

components (which may be more than H due to the incrementing process above), select the H components with the largest membership π_h. These components $\hat{\mathcal{H}}$ will represent the hybrid components. We initialize the factor means $\boldsymbol{\mu}_f$ and covariances $\boldsymbol{\Sigma}_f$ from the parameters of the fitted Gaussian mixture model. Armed with the factor component parameters, we initialize $\boldsymbol{\alpha}_h$ for each $h \in \hat{\mathcal{H}}$ by solving $\arg\min_{\boldsymbol{\alpha}_h} \left\| \boldsymbol{\mu}_h - \sum_{f \in \mathcal{F}} \alpha_{hf} \boldsymbol{\mu}_f \right\|^2$ subject to $\boldsymbol{\alpha}_h \succeq 0$ and $\mathbf{1}^\top \boldsymbol{\alpha}_h = 1$. We then initialize $\boldsymbol{\Psi}_h$ as a diagonal matrix $10^{-10}\mathbf{I}_d$. Finally, $\boldsymbol{\pi}$ is initialised by computing $\hat{z}_{ng}$ and using the maximizer in the following expectation and maximization steps, respectively.

We augment the above initialization procedure with mini-EM as described in Biernacki et al. (2003) with multiple random starts from the sampled halves of the data. We then run the following EM algorithm on each of these random starts for a pre-defined number of iterations and pick the start that has the best ensuing log-likelihood. The EM algorithm continues on this initialised model.

4.4.2 Expectation Step

Given the current iteration's parameter values of $\boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f, \boldsymbol{\Psi}_h, \alpha_{hf}, \boldsymbol{\pi}$, we apply Bayes theorem to obtain the conditional distribution of membership probabilities z_{ng}. We let $\hat{z}_{ng}$ denote the conditional expected value of the probability that observation n is assigned to a factor or hybrid cluster $g \in \mathcal{F} \cup \mathcal{H}$ at the current iteration, so that

$$\hat{z}_{ng} := \frac{\pi_g \phi_d \left(\boldsymbol{x}_n ; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g \right)}{\sum_{g \in \mathcal{H} \cup \mathcal{F}} \pi_g \phi_d \left(\boldsymbol{x}_n ; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g \right)}.$$

When $g \in \mathcal{F}$, $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ are the fully-parameterised factor distribution parameters. When $g \in \mathcal{H}$, $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ are the mean and covariance implied by the loadings $\boldsymbol{\alpha}_g$ and noise covariance $\boldsymbol{\Psi}_g$ as in (4.2).

Taking a logarithm and the expectation of (4.5) with respect to the latent variables $\mathbf{y}_{nf}$ and z_{ng}, we obtain the expected incomplete data log-likelihood. Substituting in the

appropriate multivariate normal densities, we find this surrogate function

$$
\begin{aligned}
Q\left(\theta\right) = & \sum_{n=1}^{N}\left(\sum_{f\in\mathcal{F}}\hat{z}_{nf}\log\pi_{f}+\sum_{h\in\mathcal{H}}\hat{z}_{nh}\log\pi_{h}\right)\\
& +\frac{1}{2}\sum_{n=1}^{N}\left[\sum_{f\in\mathcal{F}}\hat{z}_{nf}\left\{\log\left|\boldsymbol{\Sigma}_{f}^{-1}\right|-\mathrm{Tr}\left[\boldsymbol{\Sigma}_{f}^{-1}\left(\boldsymbol{x}_{n}-\boldsymbol{\mu}_{f}\right)\left(\boldsymbol{x}_{n}-\boldsymbol{\mu}_{f}\right)^{\top}\right]\right\}\right.\\
& +\sum_{h\in\mathcal{H}}\hat{z}_{nh}\left\{\log\left|\boldsymbol{\Psi}_{h}^{-1}\right|-\mathrm{Tr}\left[\boldsymbol{\Psi}_{h}^{-1}\left(\boldsymbol{x}_{n}-\sum_{f\in\mathcal{F}}\alpha_{hf}\bar{\mathbf{y}}_{nfh}\right)\left(\boldsymbol{x}_{n}-\sum_{f\in\mathcal{F}}\alpha_{hf}\bar{\mathbf{y}}_{nfh}\right)^{\top}\right.\right.\\
& \left.+\boldsymbol{\Psi}_{h}^{-1}\sum_{f\in\mathcal{F}}\sum_{q\in\mathcal{F}}\alpha_{hf}\alpha_{hq}\mathbf{S}_{hfq}\right]\\
& \left.\left.+\sum_{f\in\mathcal{F}}\left(\log\left|\boldsymbol{\Sigma}_{f}^{-1}\right|-\mathrm{Tr}\left[\boldsymbol{\Sigma}_{f}^{-1}\left(\bar{\mathbf{y}}_{nfh}-\boldsymbol{\mu}_{f}\right)\left(\bar{\mathbf{y}}_{nfh}-\boldsymbol{\mu}_{f}\right)^{\top}+\boldsymbol{\Sigma}_{f}^{-1}\mathbf{S}_{hff}\right]\right)\right\}\right].
\end{aligned}
$$

Here, we have defined for notational convenience the expressions for $h\in\mathcal{H}$, $f,q\in\mathcal{F}$, and $n=1,2,...,N$.

4.4.3 Maximisation Step

We now maximize the surrogate function Q in the model parameters $\left\{\boldsymbol{\pi},\boldsymbol{\mu}_{f},\boldsymbol{\Sigma}_{f},\boldsymbol{\Psi}_{h},\boldsymbol{\alpha}_{h}\right\}$. The corresponding maximizers for $\boldsymbol{\pi},\boldsymbol{\mu}_{f},\boldsymbol{\Sigma}_{f}$ are

$$
\begin{aligned}
\hat{\pi}_{g} &= \frac{1}{N}\sum_{n=1}^{N}\hat{z}_{ng},\\
\hat{\boldsymbol{\mu}}_{f} &= \frac{\sum_{n=1}^{N}\left(\hat{z}_{nf}\boldsymbol{x}_{n}+\sum_{h\in\mathcal{H}}\hat{z}_{nh}\bar{\mathbf{y}}_{nfh}\right)}{\sum_{n=1}^{N}\left(\hat{z}_{nf}+\sum_{h\in\mathcal{H}}\hat{z}_{nh}\right)},\\
\hat{\boldsymbol{\Sigma}}_{f} &= \frac{\sum_{n=1}^{N}\left\{\hat{z}_{nf}\left(\boldsymbol{x}_{n}-\boldsymbol{\mu}_{f}\right)\left(\boldsymbol{x}_{n}-\boldsymbol{\mu}_{f}\right)^{\top}+\sum_{h\in\mathcal{H}}\hat{z}_{nh}\left[\left(\bar{\mathbf{y}}_{nfh}-\boldsymbol{\mu}_{f}\right)\left(\bar{\mathbf{y}}_{nfh}-\boldsymbol{\mu}_{f}\right)^{\top}+\mathbf{S}_{hff}\right]\right\}}{\sum_{n=1}^{N}\left(\hat{z}_{nf}+\sum_{h\in\mathcal{H}}\hat{z}_{nh}\right)}.
\end{aligned}
$$

With respect to $\hat{\boldsymbol{\Sigma}}_{f}$, we apply the updates found in Celeux and Govaert (1995) and the improved updates in Browne and McNicholas (2014) when a covariance type other than VVV is specified.

Table 4.2: Maximizers for the Factor/Hybrid error distribution $\boldsymbol{\Psi}_h$.

Name	$\boldsymbol{\Psi}_h$	**Maximizer**
EV	$\boldsymbol{\Psi}_h$	$\hat{\boldsymbol{\Psi}}_h = \operatorname{diag} \frac{\sum_{n=1}^{N} \hat{z}_{nh} \left[\left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\mathbf{y}}_{nfh} \right) \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\mathbf{y}}_{nfh} \right)^\top + \sum_{f \in \mathcal{F}} \sum_{q \in \mathcal{F}} \alpha_{hf} \alpha_{hq} \mathbf{S}_{hfq} \right]}{\sum_{n=1}^{N} \hat{z}_{nh}}$
EE	$\boldsymbol{\Psi}$	$\hat{\boldsymbol{\Psi}} = \frac{\sum_{n=1}^{N} \sum_{h \in \mathcal{H}} \hat{z}_{nh} \left[\left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\mathbf{y}}_{nfh} \right) \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\mathbf{y}}_{nfh} \right)^\top + \sum_{f \in \mathcal{F}} \sum_{q \in \mathcal{F}} \alpha_{hf} \alpha_{hq} \mathbf{S}_{hfq} \right]}{\sum_{n=1}^{N} \sum_{h \in \mathcal{H}} \hat{z}_{nh}}$
IV	$\psi_h \mathbf{I}_d$	$\hat{\psi}_h = \frac{\sum_{n=1}^{N} \hat{z}_{nh} \left\{ \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\mathbf{y}}_{nfh} \right)^\top \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\mathbf{y}}_{nfh} \right) + \sum_{f \in \mathcal{F}} \sum_{q \in \mathcal{F}} \alpha_{hf} \alpha_{hq} \operatorname{Tr}\left[\mathbf{S}_{hfq} \right] \right\}}{\sum_{n=1}^{N} \hat{z}_{nh}}$
IE	$\psi \mathbf{I}_d$	$\hat{\psi} = \frac{\sum_{n=1}^{N} \sum_{h \in \mathcal{H}} \hat{z}_{nh} \left[\left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\mathbf{y}}_{nfh} \right)^\top \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\mathbf{y}}_{nfh} \right) + \sum_{f \in \mathcal{F}} \sum_{q \in \mathcal{F}} \alpha_{hf} \alpha_{hq} \operatorname{Tr}\left[\mathbf{S}_{hfq} \right] \right]}{\sum_{n=1}^{N} \sum_{h \in \mathcal{H}} \hat{z}_{nh}}$
C	$\varepsilon \mathbf{I}_d$	$\varepsilon > 0$ specified and held constant.

Depending on the specified error distribution covariance, the maximizers in $\boldsymbol{\Psi}_h$ are listed in Table 4.2. Finally, the maximizer in each $\boldsymbol{\alpha}_h$ can be obtained by using the solution of Goldfarb and Idnani (1983) to the constrained quadratic programming problem

$$\min_{\boldsymbol{\alpha}_h} \frac{1}{2} \boldsymbol{\alpha}_h^\top \left(\sum_{n=1}^{N} \hat{z}_{nh} \mathbf{A}_{nh}^\top \boldsymbol{\Psi}_h^{-1} \mathbf{A}_{nh} + \mathbf{B}_h \sum_{n=1}^{N} \hat{z}_{nh} \right) \boldsymbol{\alpha}_h + \left(\sum_{n=1}^{N} \hat{z}_{nh} \mathbf{A}_{nh}^\top \boldsymbol{\Psi}_h^{-1} \boldsymbol{x}_n \right) \boldsymbol{\alpha}_h$$
$$\text{subject to } \boldsymbol{\alpha}_h \succeq 0 \text{ and } \mathbf{1}^\top \boldsymbol{\alpha}_h = 1.$$

Again, we define for simplicity the expressions

$$\mathbf{A}_{nh} = \begin{bmatrix} \bar{\mathbf{y}}_{n1h} & \cdots & \bar{\mathbf{y}}_{nFh} \end{bmatrix},$$
$$\mathbf{B}_h = \begin{bmatrix} \operatorname{Tr}\left[\boldsymbol{\Psi}_h^{-1} \mathbf{S}_{h11} \right] & \cdots & \operatorname{Tr}\left[\boldsymbol{\Psi}_h^{-1} \mathbf{S}_{h1F} \right] \\ \vdots & \ddots & \vdots \\ \operatorname{Tr}\left[\boldsymbol{\Psi}_h^{-1} \mathbf{S}_{hF1} \right] & \cdots & \operatorname{Tr}\left[\boldsymbol{\Psi}_h^{-1} \mathbf{S}_{hFF} \right] \end{bmatrix}.$$

We note that the optimisation in $\boldsymbol{\Psi}_h$ depends on $\boldsymbol{\alpha}_h$ and vice-versa, with no tractable way of finding the simultaneous solution to the system. Hence, we resort to the multi-cycle EM algorithm of Meng and Rubin (1993) and maximize in $\left\{ \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f, \boldsymbol{\Psi}_h, \boldsymbol{\pi} \right\}$ and $\boldsymbol{\alpha}_h$ in alternate iterations.

4.4.4 Convergence

The convergence of the EM algorithm is assessed using Aitken acceleration (McNicholas et al., 2010). Let t denote the iteration count of the EM algorithm, and let $\ell^{(t)}$ be the log-likelihood for that iteration. For a sequence of log-likelihoods $\ell^{(1)}, \ell^{(2)}, ...$, we define the Aitken acceleration at iteration t as

$$a^{(t)} = \frac{\ell^{(t+1)} - \ell^{(t)}}{\ell^{(t)} - \ell^{(t-1)}}$$

and an estimate of the asymptotic log-likelihood ℓ_∞ (Böhning et al., 1994) as

$$\ell_\infty^{(t+1)} = \ell^{(t)} + \frac{1}{1 - a^{(t)}} \left(\ell^{(t+1)} - \ell^{(t)} \right).$$

The convergence criteria for stopping the EM algorithm is

$$0 \leq \ell_\infty^{(t+1)} - \ell^{(t)} \leq \varepsilon$$

for some $\varepsilon > 0$ (McNicholas et al., 2010). We have chosen $\varepsilon = 10^{-8}$ throughout.

We only apply this convergence criterion in the main EM algorithm; during the mini-EM procedure described in Section 4.4.1, we run the full number of specified mini-EM iterations for each random start.

4.5 Applications

4.5.1 Iris dataset

The *iris* dataset is described in Fisher (1936) with 150 observations of flowers from *iris* classified into three species: setosa, versicolor, and virginica. There are four variables denoting the sepal width/length and petal width/length of each sampled flower, and a class label for the species with a fifty observations of each. We evaluate up to a total of five components for both models, and present the best parameter combination in Table 4.3. Of note is that *mclust* selects a two-component model using BIC; we have presented

Table 4.3: Results for the *iris* dataset. Factor/Hybrid clustering and mclust are evaluated up to 6-component models. Epistatic Clustering is evaluated up to a total of 7-components, one of which is a miscellaneous cluster. Note that Epistatic Clustering uses a different nomenclature for covariance types. Best values in bold.

	Factor/Hybrid		**mclust**		**Epistatic Clustering**	
	Best BIC	Best ARI	Best BIC	Best ARI	Best BIC	Best ARI
# Factor	2	2	2	3	3	4
# Hybrid	1	1	-	-	1	3
Cov. Type	VEV	EEV	VEV	EEE	EEE	EVE
Err. Dist. Type	C (10^{-10})	C (10^{-10})	-	-	-	-
Log-Likelihood	**-206.71**	-238.57	-215.73	-256.35	-224.18	-180.19
Free Parameters	28	31	26	**24**	35	62
BIC	**553.72**	632.46	561.73	632.96	604.08	671.03
ARI	0.9222	**0.9410**	0.5681	**0.9410**	0.5681	0.9039

a three-component model as well for complete exploration of the *iris* dataset. The ARI is computed against the species variable in the dataset.

From Table 4.3, we can see that factor-hybrid clustering results in the best BIC and better ARI compared to *mclust*. Epistatic Clustering does not perform well since the hybrid cluster is not located at the midpoint between factors. Indeed, by examining the fitted $\boldsymbol{\alpha}_h$ values for the solitary hybrid cluster, we find that the hybrid component representing 46 *iris versicolor* instances is composed of 73.7% factor component 1 (50 *iris virginica* and 4 mis-classified *iris versicolor*) and 26.3% factor component 2 (50 *iris setosa*). By comparison to Plate 23 of Anderson (1936), which states that the hybrid *iris versicolor* is composed of two parts *iris virginica* and one part *iris setosa*, the purported interpolation of multivariate normal parameters is three-to-one here. Figure 3.1 demonstrates the fitted model and the parameter hybridisation visually.

4.5.2 Penguin dataset

The penguins dataset (Horst et al., 2020) is similar to the iris dataset, describing three species of penguin, Adélie ($n = 146$), Chinstrap ($n = 68$), and Gentoo ($n = 119$), in terms of bill length/depth, flipper length, and body mass. The data also includes variables for the island on which each observation was taken. We evaluate up to a total of six components for factor-hybrid and *mclust* models and up to seven components for EC, and present the best parameter combination in Table 4.4. An ARI is computed against the species variable, as well as the island variable and their concatenation.

In this dataset, we find that *mclust* outperforms on ARI (species) and slightly on BIC. Indeed, inspection of the scatterplots of the penguin data shows interpolation behaviour between species less clearly than in the *iris* dataset. However, factor/hybrid clustering shows a marginal improvement in ARI when the island variable is added to the mix. Epistatic Clustering also produces the best species ARI when selecting for the best BIC.

Of note is that both the Factor-Hybrid and *mclust* models tend to select for equal orientation. However, the Factor-Hybrid model improves species ARI by simultaneously

Table 4.4: Results for the *penguin* dataset. Factor/Hybrid clustering and mclust are evaluated up to 6-component models. Epistatic Clustering is evaluated up to a total of 7-components, one of which is a miscellaneous cluster. Note that Epistatic Clustering uses a different nomenclature for covariance types. Best values in bold. Here, the best ARI is with respect to the species variable.

	Factor/Hybrid		**mclust**		**Epistatic Clustering**
	Best BIC	Best ARI	Best BIC	Best ARI	Best BIC & ARI
# Factor	4	3	4	3	4
# Hybrid	1	1	-	-	3
Cov. Type	EEE	VVE	VEE	EEE	EVE
Err. Dist. Type	IE	C	-	-	-
Log-Likelihood	-5029.68	-5033.12	-5025.09	-5055.16	**-5016.59**
Free Parameters	34	35	32	**24**	62
BIC	10256.83	10269.53	**10236.04**	10249.71	10393.27
ARI (species)	0.7591	**0.9623**	0.8217	0.9591	0.9590
ARI (island)	**0.4030**	0.3689	0.2105	0.3683	0.3708
ARI (species $\times$ island)	**0.6996**	0.6739	0.4736	0.6645	0.6717

Table 4.5: Results for the olive oil dataset. Factor/Hybrid clustering and mclust are evaluated up to 14-component models. Epistatic Clustering is evaluated up to a total of 16-components, one of which is a miscellaneous cluster. Note that Epistatic Clustering uses a different nomenclature for covariance types. Best values in bold.

	Factor/Hybrid		**mclust**		**Epistatic Clustering**	
	Best BIC	Best ARI	Best BIC	Best ARI	Best BIC	Best ARI
# Factor	4	3	10	3	6	4
# Hybrid	2	1	-	-	10	3
Cov. Type	VVV	VVV	VVE	EEE	EVE	EVE
Err. Dist. Type	EE	EE	-	-	-	-
Log-Likelihood	**-20235.42**	-20793.94	-20448.03	-21997.31	-20391.97	-21022.98
Free Parameters	195	145	197	**62**	279	182
BIC	**41708.92**	42508.51	42146.84	44348.27	42555.35	43201.51
ARI	0.4594	0.9640	0.3145	0.9163	0.5798	**0.9976**

removing a factor component and constraints on the factor covariances, suggesting that species information is captured by the volume and shape of the factor components. The *mclust* model exhibits a similar behaviour in volume only.

4.5.3 Olive oil dataset

This dataset (Forina et al., 1983) describes 572 observations of olive oil using eight measured fatty acid levels. Specifically, the levels of palmitic, palitoleic, stearic, oleic, linleic, linolenic, arachidic, and eicosenoic acid were measured for olive oils taken from nine different regions of Italy. We evaluate up to a total of 14 components for factor-hybrid and *mclust* and 16 components for EC, and present the best parameter combination in Table 4.5. An ARI is computed against the region variable.

Here we can see considerable gains from the factor/hybrid model; both BIC and ARI

Table 4.6: Results for the wine dataset. Factor/Hybrid clustering and mclust are evaluated up to 6-component models. Epistatic Clustering is evaluated up to a total of 7-components, one of which is a miscellaneous cluster. Note that Epistatic Clustering uses a different nomenclature for covariance types. Best values in bold.

	Factor/Hybrid		**mclust**		**Epistatic Clustering**	
	Best BIC	Best ARI	Best BIC	Best ARI	Best BIC	Best ARI
# Factor	3	3	3	3	3	4
# Hybrid	1	1	-	-	1	3
Cov. Type	VVI	VEI	EVI	EEE	EVI	EVD
Err. Dist. Type	EE	C (10^{-10})	-	-	-	-
Log-Likelihood	-11458.92	-11764.99	-11557.21	**-10884.00**	-11078.86	-11892.70
Free Parameters	194	**115**	162	461	464	521
BIC	**23923.10**	24125.89	23953.87	24159.80	24562.06	26485.12
ARI	0.8343	0.9404	0.8301	**1.0000**	0.0000	0.9976

perform better than only parsimonious covariance matrices despite the apparent gain in free parameters.

Examining the covariance structures, we see that both Factor-Hybrid and Epistatic Clustering selects for the most flexibility in factor (parent) cluster shapes using both BIC and ARI. Moreover, the error distribution here suggests hybrid components are better represented with an enlarged covariance.

4.5.4 Wine dataset

The Italian wines dataset (von Weinen, 1986) describes 178 observations of wines using 27 different physical and chemical properties. We have used the 27-variable dataset in lieu of the more common 13-variable dataset. There is an associated categorical variable for the type of wine for each observation. Table 4.6 displays the results for this dataset.

We can see here that the factor/hybrid model slightly outperforms mclust on BIC, but does not perform as well for ARI. It may be the case that the three types of wine are mostly factors; the estimated hybrid component contains only fifteen observations, which may suggest a small intermediate cluster of mixed nature. By inspecting $\boldsymbol{\alpha}_h$ from the best BIC fit, we see that the hybridisation proportions are 55% factor 1 (all 57 are Grignolino) and 44% factor 3 (all 56 are Barolo). The remaining 1% belongs to factor 2, (48 Barbera, 2 Grignolino). Moreover, both *mclust* and EC can recover class labels well with an optimal parameter choice, but EC's best BIC model completely fails to recover class labels.

In this dataset, we observe a wide selection of covariance types. This is also the widest dataset at 27 variables, potentially producing a large number of free parameters in the covariance matrices. Indeed, most of the models seem to prefer the much more economical diagonal covariance. This likely is due to sparsity in the 27-dimensional space.

4.6 Simulation Study

4.6.1 Factor-Hybrid Data

To assess the parameter recovery and model performance, we perform a simulation study with a family of generated datasets from the proposed model. These datasets are drawn from the proposed model distribution with cluster memberships known a priori. With respect to the number of free parameters in the model, we opt to test the worst case of VVV covariance and EV error distribution. We term the family of datasets used here in the d-hypercube dataset as it can be scaled to arbitrary dimension $d \geq 2$.

We consider a factor-hybrid data generating model with three parameters; the data dimension d, the number of observations n, and a tunable degree of component overlap λ. For $d \geq 2$, we place 2^d factor components centred at the vertices of the d-dimensional hypercube $\{-1, +1\}^d$ and denote the means $\boldsymbol{\mu}_{f_1}, \boldsymbol{\mu}_{f_2}, ...\boldsymbol{\mu}_{f_{2^d}}$. To each of these $\boldsymbol{\mu}_f$, we associate a parity value p_f equal to the sign of the product of $\boldsymbol{\mu}_f$ elements. The covariance

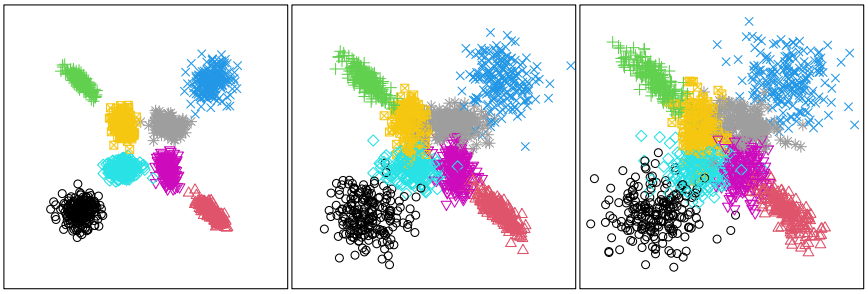

Figure 4.1: Scatterplots of the hypercube dataset in 2D with 200 observations per cluster. λ is set to 1 (left), 3 (middle), and 5 (right) to demonstrate increasing overlap.

of each factor f is given by

$$\boldsymbol{\Sigma}_f = \begin{cases} 0.025\lambda \mathbf{I}_d & \text{if } p_f = +1 \\ \frac{0.018\lambda}{\boldsymbol{\mu}_f^\top \boldsymbol{\mu}_f} \boldsymbol{\mu}_f \boldsymbol{\mu}_f^\top + 0.002\lambda \mathbf{I}_d & \text{if } p_f = -1 \end{cases}$$

and can be scaled up/down by λ to represent greater/lesser overlap.

With each factor component f_i for $i \in 1, 2, 3, ..., 2^d$ having mean $\boldsymbol{\mu}_{f_i}$ we associate a hybrid component h_i such that $\boldsymbol{\mu}_{h_i} = \frac{1}{3}\boldsymbol{\mu}_{f_i}$ also with the same parity $p_{h_i} = p_{f_i}$. For each $\boldsymbol{\mu}_h$, we back out a corresponding $\boldsymbol{\alpha}_h$ by solving the quadratic program

$$\min_{\boldsymbol{\alpha}_h} \boldsymbol{\alpha}_h^\top \boldsymbol{\alpha}_h \quad \text{subject to } \mathbf{1}^\top \boldsymbol{\alpha}_h = 1, \boldsymbol{\alpha}_h \succeq 0, \sum_{f \in \mathcal{F}} \alpha_{hf} \boldsymbol{\mu}_f = \boldsymbol{\mu}_h.$$

Finally, we define the noise covariance for each hybrid component by

$$\boldsymbol{\Psi}_h = \begin{cases} 0.001\lambda \, \text{diag}(10, 1, 10, 1, ...) & \text{if } p_h = +1 \\ 0.001\lambda \, \text{diag}(1, 10, 1, 10, ...) & \text{if } p_h = -1 \end{cases}$$

again scaled by λ.

This completes the specification of 2^d factor and 2^d hybrid component densities. Figure 4.1 provides a visualisation of the dataset in two-dimensions.

We consider all combinations of dataset parameters with $d = 2, 3, 5$, $n = 20, 200$, and $\lambda = 1, 5$, and perform 10 replications at each parameter combination. We use the

estimation procedure defined in Section 4.4, with the assumption that parameters F, H, and both covariance types are correctly-specified for computational simplicity. Moreover, we skip mini-EM and initialise with the full dataset and proceed directly to a maximum of 10000 EM iterations, with convergence criterion $0 \leq \ell_\infty^{(t+1)} - \ell^{(t)} \leq 10^{-8}$.

Similarly, we also run epistatic clustering with the correct specification of EVE parent clusters corresponding to VVV factor covariances. We restrict epistatic clustering to two-parent clusters only; otherwise, the number of potential epistatic clusters becomes computationally difficult. Lastly, we run parsimonious Gaussian mixture models using *mclust* with the correct number of iterations, selecting covariance type using BIC.

We summarise these parameter combinations in Tables 4.7, 4.8 and 4.9 in terms of ARI and BIC for all three models, and $\boldsymbol{\alpha}$ parameter recovery for the factor-hybrid model. We note that the number of possible two-parent clusters exceeds the number of actual hybrid clusters, which poses a problem for epistatic clustering as this inflates the parameter count and BIC value. Additionally, epistatic clustering fails computationally for this family of datasets when $d = 5$ as there are $2^5 = 32$ parents with 496 potential two-parent clusters.

To assess parameter recovery of $\boldsymbol{\alpha}$ for hybrid components, we need to permute the component indices to best match the original indices. Firstly, we solve the assignment problem to minimise the sum of L_2 norms between true and fitted factor $\boldsymbol{\mu}_f$ to permute factor indices. Next, we use that $\boldsymbol{\mu}_{h_i}$ is closest to $\boldsymbol{\mu}_{f_i}$ by design and so permute hybrid indices such that $\alpha_{h_i f_i}$ has maximum weight within $\boldsymbol{\alpha}_{h_i}$. Finally, we may calculate a cosine similarity measure as the cosine of the angle formed between each true and fitted $\boldsymbol{\alpha}_h$ and average over the h hybrid components.

4.6.2 Epistatic Data

We now consider the case of model misspecification; the data actually comes from the epistatic clustering model instead of the factor-hybrid clustering model. We modify the d-hypercube dataset slightly to reflect the epistatic clustering model by instead placing parent clusters at the vertices $\{-2, +2\}^d$ of a d-dimensional hypercube, and using parent

Table 4.7: Simulation results for the d-hypercube dataset over multiple parameter combinations fitted using the proposed factor-hybrid model. Data is generated from the factor-hybrid model. Average results with standard deviations in brackets over ten replications.

d	λ	n	BIC (sd)	ARI (sd)	$\boldsymbol{\alpha}$ Cos. Sim. (sd)
2	1	20	344.53 (19.90)	0.9971 (0.0061)	0.9009 (0.0661)
2	1	200	1647.49 (77.62)	0.9987 (0.0011)	0.9720 (0.0279)
2	5	20	760.96 (19.18)	0.6673 (0.0400)	0.8437 (0.0596)
2	5	200	5892.69 (59.04)	0.7729 (0.0171)	0.8737 (0.0811)
3	1	20	771.75 (53.28)	0.9851 (0.0296)	0.6840 (0.0345)
3	1	200	431.61 (74.72)	0.9999 (0.0003)	0.7181 (0.0281)
3	5	20	2223.50 (82.92)	0.6861 (0.0582)	0.6503 (0.0342)
3	5	200	14613.45 (110.92)	0.8622 (0.0074)	0.7248 (0.0357)
5	1	20	8348.50 (105.35)	1.0000 (0.0000)	0.3986 (0.0298)
5	1	200	-20577.11 (185.00)	1.0000 (0.0001)	0.4274 (0.0205)
5	5	20	18416.16 (120.77)	0.8419 (0.0196)	0.3859 (0.0190)
5	5	200	79646.51 (279.13)	0.9198 (0.0119)	0.4138 (0.0283)

Table 4.8: Simulation results for the d-hypercube dataset over multiple parameter combinations fitted using epistatic clustering with two-parent clusters. Data is generated from the factor-hybrid model. Average results with standard deviations in brackets over ten replications. For $d = 5$, epistatic clustering encountered computational troubles.

d	λ	n	BIC (sd)	ARI (sd)
2	1	20	414.98 (34.40)	0.8162 (0.0627)
2	1	200	2677.67 (323.37)	0.8580 (0.1140)
2	5	20	709.17 (17.76)	0.6104 (0.0870)
2	5	200	5914.48 (61.72)	0.5925 (0.0366)
3	1	20	925.32 (116.10)	0.7227 (0.0875)
3	1	200	3357.58 (581.95)	0.8070 (0.0457)
3	5	20	1995.86 (38.11)	0.6101 (0.1020)
3	5	200	15160.05 (195.06)	0.6719 (0.0461)
5	1	20	-	-
5	1	200	-	-
5	5	20	-	-
5	5	200	-	-

Table 4.9: Simulation results for the d-hypercube dataset over multiple parameter combinations fitted using *mclust*. Data is generated from the factor-hybrid model. Average results with standard deviations in brackets over ten replications.

d	λ	n	BIC (sd)	ARI (sd)
2	1	20	310.79 (19.72)	0.9971 (0.0061)
2	1	200	1658.27 (57.83)	0.9984 (0.0016)
2	5	20	746.14 (20.14)	0.6625 (0.0545)
2	5	200	5917.24 (70.10)	0.7627 (0.0145)
3	1	20	677.37 (40.59)	0.9993 (0.0022)
3	1	200	252.07 (155.09)	0.9999 (0.0003)
3	5	20	2076.26 (37.89)	0.7634 (0.0386)
3	5	200	14657.37 (105.02)	0.8617 (0.0070)
5	1	20	4503.35 (121.14)	0.9990 (0.0011)
5	1	200	-25663.34 (351.50)	1.0000 (0.0000)
5	5	20	14415.35 (153.75)	0.7865 (0.0231)
5	5	200	74564.66 (338.26)	0.9111 (0.0093)

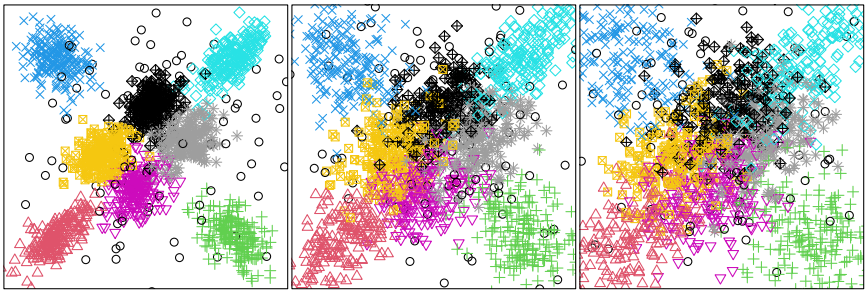

Figure 4.2: Scatterplots of the modified epistatic hypercube dataset in 2D with 200 observations per parent/epistatic cluster and 100 observations in a miscellaneous cluster. λ is set to 1 (left), 3 (middle), and 5 (right) to demonstrate increasing overlap.

covariances (denoted $\boldsymbol{\Sigma}_f$ for consistency) given by

$$\boldsymbol{\Sigma}_f = \begin{cases} \frac{0.2\lambda}{\boldsymbol{\mu}_f^\top \boldsymbol{\mu}_f} \boldsymbol{\mu}_f \boldsymbol{\mu}_f^\top + 0.05\lambda \mathbf{I}_d & \text{if } p_f = +1 \\ \frac{0.1\lambda}{\boldsymbol{\mu}_f^\top \boldsymbol{\mu}_f} \boldsymbol{\mu}_f \boldsymbol{\mu}_f^\top + 0.1\lambda \mathbf{I}_d & \text{if } p_f = -1 \end{cases}$$

with λ and p_f playing the same roles as in the unmodified d-hypercube dataset. This covariance specification is VVV in factor-hybrid and *mclust*, and EVE in epistatic clustering.

An epistatic cluster is now constructed for every edge between two vertices of the hypercube; the two clusters at these vertices are considered the two parents of the epistatic cluster. This produces $d \times 2^{d-1}$ epistatic clusters. Epistatic cluster means and covariances are computed as in Zhang (2013); specifically, they are effectively averaged as canonical parameters rather than moment parameters. Finally, a miscellaneous cluster centred at the origin with covariance $3\mathbf{I}_d$ is added with $\lfloor \frac{n}{2} \rfloor$ observations. Figure 4.2 shows the modified epistatic d-hypercube dataset in two-dimensions for multiple values of λ.

We fit the proposed factor-hybrid model, epistatic clustering, and *mclust* to the dataset for $d = 2, 3$, $n = 20, 200$, and $\lambda = 1, 5$, and perform 10 replications at each parameter combination. As in the unmodified d-hypercube dataset, we avoid fitting $d = 5$ for epistatic clustering. Additionally, there are no parameters in epistatic clustering corresponding to $\boldsymbol{\alpha}_h$ or $\boldsymbol{\Psi}_h$ in the factor-hybrid model; thus, we exclude parameter recovery of $\boldsymbol{\alpha}_h$ from consideration and set the hybrid noise covariance to type EV to accommodate epistatic

Table 4.10: Simulation results for the modified epistatic d-hypercube dataset over multiple parameter combinations fitted using the proposed factor-hybrid model. Data is generated from the epistatic clustering model. Average results with standard deviations in brackets over ten replications.

d	λ	n	BIC (sd)	ARI (sd)
2	1	20	1217.32 (23.28)	0.8667 (0.0348)
2	1	200	10104.01 (92.77)	0.8867 (0.0088)
2	5	20	1495.08 (26.35)	0.3670 (0.0564)
2	5	200	12870.39 (61.81)	0.4407 (0.0146)
3	1	20	4356.53 (51.76)	0.9110 (0.0244)
3	1	200	33155.30 (127.30)	0.9461 (0.0036)
3	5	20	5631.82 (73.04)	0.3726 (0.0707)
3	5	200	45355.78 (223.21)	0.4145 (0.0523)

clusters being hybridisations of canonical parameters instead of moment parameters. ARI and BIC results are presented in Tables 4.10, 4.11, and 4.12.

We note that despite the data being drawn from the EC model, the EC procedure fails to yield as high an ARI as one may expect. The EC model does not allow selecting an exact number of epistatic clusters, only the number of parents each epistatic cluster may have. Thus, by specifying that we want two-parent clusters (for the d-hypercube, there are $2^d(2^{d-1})/2$ possible such epistatic clusters), we have specified more than are actually present ($d \times 2^{d-1}$). As a result, a better log-likelihood is attained by splitting or redistributing observations between epistatic clusters, leading to a loss of ARI. Indeed, we see Epistatic Clustering obtains worse ARI as the dimension increases; the number of parents and possible epistatic clusters grows as well.

Table 4.11: Simulation results for the modified epistatic d-hypercube dataset over multiple parameter combinations fitted using epistatic clustering with two-parent clusters. Data is generated from the epistatic clustering model. Average results with standard deviations in brackets over ten replications.

d	λ	n	BIC (sd)	ARI (sd)
2	1	20	1142.07 (21.58)	0.8263 (0.0325)
2	1	200	10082.56 (80.93)	0.8272 (0.0111)
2	5	20	1394.43 (25.48)	0.3191 (0.0501)
2	5	200	12735.80 (88.10)	0.3959 (0.0410)
3	1	20	3985.36 (43.76)	0.7106 (0.0530)
3	1	200	34510.37 (721.04)	0.7904 (0.0574)
3	5	20	4915.93 (49.93)	0.3039 (0.0250)
3	5	200	44423.89 (140.06)	0.4219 (0.0222)

Table 4.12: Simulation results for the modified epistatic d-hypercube dataset over multiple parameter combinations fitted using *mclust*. Data is generated from the epistatic clustering model. Average results with standard deviations in brackets over ten replications.

d	λ	n	BIC (sd)	ARI (sd)
2	1	20	1194.43 (21.88)	0.8441 (0.0651)
2	1	200	10023.56 (61.81)	0.8866 (0.0051)
2	5	20	1436.84 (34.02)	0.3345 (0.0865)
2	5	200	12852.55 (102.04)	0.3922 (0.0203)
3	1	20	4174.58 (37.34)	0.8741 (0.0377)
3	1	200	32825.72 (97.52)	0.9465 (0.0049)
3	5	20	5291.78 (25.32)	0.3111 (0.0215)
3	5	200	45088.35 (179.43)	0.3624 (0.0167)

Chapter 5

Model-Based Clustering with Nested Gaussian Clusters

5.1 Introduction

For some datasets, class labels can exhibit multiple levels of hierarchy. As an example, a dataset with observational units of cities across the globe can have as class labels the nation and province/state to which a city belongs. Some of the variation in the data can be attributed to national behaviours, and others to provincial/state behaviours. Continuing the analogy in a mixture modelling context, we consider the case of finite Gaussian mixture models. Assuming each cluster characterises a province, we can posit that these are nested within the nation to which they belong. To this end, we extend the finite Gaussian mixture model to capture this type of nested behaviour.

In the literature, Vermunt (2003) provides a framework for tackling multiple levels of latent classes with a nested structure. Galimberti and Soffritti (2007) propose a related model for multiple clusterings in disjoint subsets of the manifest variables. Here, these subsets are treated independently by fitting two distinct Gaussian mixture models. In Galimberti and Soffritti (2010), the model of Galimberti and Soffritti (2007) is extended

to cover a more complex layering of simultaneous and independent clusterings. A followup extends this by adding a regression framework whereby the second clustering is modelled as a mixture of regressions using the first set of clustering variables as covariates (Galimberti et al., 2018). The remaining variables are considered noise, which are also related to the clustering subspaces through a regression model. This approach is reminiscient of Bouveyron and Brunet (2012), which fits a Gaussian mixture model in a subspace of the observed data with unrelated and independent noise in the orthogonal complement. Moreover, Marbac and Vandewalle (2019) proposes a multiple partitions model (MPM) with a similar two-clustering structure. In both cases, the collection of class labels are simultaneous while the realizations in the observed variables may exhibit a dependence structure. Both Marbac and Vandewalle (2019) and Galimberti et al. (2018) include a variable selection step to partition the data into their respective clusterings.

Building upon the literature, we propose a nested model that allows successive tiers of clusterings that are nested within parent clusterings. The proposed model-based clustering method models a tree-like class structure and captures class labels with dependency, whose manifestation occurs in different directions and subspaces for different levels of the hierarchy. In lieu of a variable selection or partitioning the variable set, we rotate the data using an orthogonal linear transformation. This allows variables to fractionally participate in each level of clustering by being rotated to partially project into the corresponding intrinsic subspace. As a special case, the proposed model framework can perform variable selection or partitioning as in the existing literature (Galimberti and Soffritti, 2007; Galimberti et al., 2018; Marbac and Vandewalle, 2019) by restricting the rotation to be a permutation matrix. Moreover, the nesting specification allows the total effective number of clusters to no longer be product of two positive integers. We provide the model specification for the nested Gaussians model with estimation via the Expectation-Maximization algorithm (Dempster et al., 1977) and some applications and simulation results.

5.2 Nested Gaussian Mixture Clusters Model

We propose each observation $n = 1, 2, ..., N$ to have a latent intrinsic representation $\boldsymbol{r}_n \in \mathbb{R}^p$ which we observe in the manifest variables $\boldsymbol{v}_n \in \mathbb{R}^p$ through the orthogonal rotation matrix $\boldsymbol{\Gamma}$ such that $\boldsymbol{v}_n = \boldsymbol{\Gamma}^\top \boldsymbol{r}_n$. We specify $\boldsymbol{r}_n$ to have a block structure $\langle \boldsymbol{x}_n, \boldsymbol{y}_n, \boldsymbol{u}_n \rangle$ representing sequentially nested levels of clustering occuring in the primary, secondary, and noise intrinsic subspaces having dimensions $p_x \geq 1$, $p_y \geq 1$, and $p_u \geq 0$, respectively, totalling $p_x + p_y + p_u = p$.

An observation n in primary cluster $g = 1, 2, ..., G$ can have secondary subclustering $g{:}h = 1, 2, ..., H_g$ conditional on the primary clustering g. The primary clustering occurs in the intrinsic variables $\boldsymbol{x}_n \sim \mathrm{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$, and the secondary clustering occurs in the intrinsic variables $\boldsymbol{y}_n \sim \mathrm{N}(\boldsymbol{\eta}_{g:h} + \mathbf{B}_{g:h}\boldsymbol{x}_n, \boldsymbol{\Lambda}_{g:h})$. We use the notation $g{:}h$ to denote the secondary cluster h of primary cluster g in a sequential structure akin to a tree. For example, cluster 3:4 is the fourth secondary subcluster of the third primary cluster. An extension to tertiary subcluster i could be written as $g{:}h{:}i$. Any remaining manifest variable dimensions $p_u = p - p_x - p_y$ are considered noise; as in Bouveyron and Brunet (2012), we assume the noise variables $\boldsymbol{u}_n$ follow $\mathrm{N}(\boldsymbol{\xi}, \boldsymbol{\Psi})$ for diagonal $\boldsymbol{\Psi}$, irrespective of cluster membership and uncorrelated to other intrinsic variables. Without loss of generality, we order the intrinsic dimensions as $\langle \boldsymbol{x}_n, \boldsymbol{y}_n, \boldsymbol{u}_n \rangle$ by permuting the columns of $\boldsymbol{\Gamma}$. This rotation matrix $\boldsymbol{\Gamma}$ serves the same purpose as the rotation matrix in a principal components analysis (PCA); consequently, we may interpret the intrinsic subspace as the collection of principal components in which a clustering takes place. The same caveat from PCA applies; the linear combination of manifest variables may not have a contextual meaning, but could be useful for purposes such as data visualization or input into other statistical models as extracted features. The resulting model density is

$$f(\boldsymbol{\Gamma}\boldsymbol{v}; \boldsymbol{\theta}) = \sum_{g=1}^{G} \sum_{h=1}^{H_g} \pi_g \tau_{g:h} \phi_{p_x}(\boldsymbol{x}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \phi_{p_y}(\boldsymbol{y}; \boldsymbol{\eta}_{g:h} + \mathbf{B}_{g:h}\boldsymbol{x}, \boldsymbol{\Lambda}_{g:h}) \phi_{p_u}(\boldsymbol{u}; \boldsymbol{\xi}, \boldsymbol{\Psi}), \tag{5.1}$$

with constraints $\sum_{g=1}^{G} \pi_g = 1$, $\sum_{h=1}^{H_g} \tau_{g:h} = 1$ and $\pi_g, \tau_{g:h} > 0$. The parameter set $\boldsymbol{\theta}$ is comprised of $\left\{\pi_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \tau_{g:h}, \boldsymbol{\eta}_{g:h}, \mathbf{B}_{g:h}, \boldsymbol{\Lambda}_{g:h}, \boldsymbol{\xi}, \boldsymbol{\Psi}, \boldsymbol{\Gamma}\right\}$ for valid combinations of g and $g{:}h$.

In terms of the intrinsic data $\boldsymbol{r} = \boldsymbol{\Gamma}\boldsymbol{v}$, the model density can be expressed in terms of a finite Gaussian mixture model density with structured parameters of the form

$$f(\boldsymbol{\Gamma}\boldsymbol{v};\boldsymbol{\theta}) = \sum_{g=1}^{G}\sum_{h=1}^{H_g} \pi_g \tau_{g:h} \phi_p \left(\boldsymbol{\Gamma}\boldsymbol{v}; \underbrace{\begin{bmatrix} \boldsymbol{\mu}_g \\ \boldsymbol{\eta}_{g:h} + \mathbf{B}_{g:h}\boldsymbol{\mu}_g \\ \boldsymbol{\xi} \end{bmatrix}}_{\bar{\boldsymbol{\mu}}_{g:h}}, \underbrace{\begin{bmatrix} \boldsymbol{\Sigma}_g & \boldsymbol{\Sigma}_g\mathbf{B}_{g:h}^\top & \mathbf{0} \\ \mathbf{B}_{g:h}\boldsymbol{\Sigma}_g & \mathbf{B}_{g:h}\boldsymbol{\Sigma}_g\mathbf{B}_{g:h}^\top + \boldsymbol{\Lambda}_{g:h} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Psi} \end{bmatrix}}_{\bar{\boldsymbol{\Sigma}}_{g:h}} \right). \tag{5.2}$$

We may also express the model density in terms of the manifest variables $\boldsymbol{v} = \boldsymbol{\Gamma}^\top\boldsymbol{r} = \langle \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u} \rangle$ as

$$f(\boldsymbol{v};\boldsymbol{\theta}) = \sum_{g=1}^{G}\sum_{h=1}^{H_g} \pi_g \tau_{g:h} \phi_p \left(\boldsymbol{v}; \boldsymbol{\Gamma}^\top\bar{\boldsymbol{\mu}}_{g:h}, \boldsymbol{\Gamma}^\top\bar{\boldsymbol{\Sigma}}_{g:h}\boldsymbol{\Gamma} \right). \tag{5.3}$$

5.2.1 Model Variations

A multitude of variations on the model presented in Section 5.2 are presented in this section with a brief description of their behaviour. A graphical representation of the model's dependency structure and the variations listed in this section are presented in Figure 5.1.

Conditionally Independent

As another alternate variation, if we assume the secondary clustering does not depend on the membership in the primary clustering, then we replace the index for secondary clusters $g{:}h$ by h alone for $h = 1, 2, ..., H$ secondary clusters. This leads to the model density

$$f(\boldsymbol{\Gamma}\boldsymbol{v};\boldsymbol{\theta}) = \sum_{g=1}^{G}\sum_{h=1}^{H} \pi_g \tau_h \phi_{p_x}(\boldsymbol{x};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\phi_{p_y}(\boldsymbol{y};\boldsymbol{\eta}_h + \mathbf{B}_h\boldsymbol{x},\boldsymbol{\Lambda}_h)\phi_{p_u}(\boldsymbol{u};\boldsymbol{\xi},\boldsymbol{\Psi}). \tag{5.4}$$

This coincides with the relation found in the two independent clusterings formulation of Galimberti et al. (2018), with the difference that the intrinsic noise variables are independent. Practically speaking, this means that the two sets of class labels occur independently

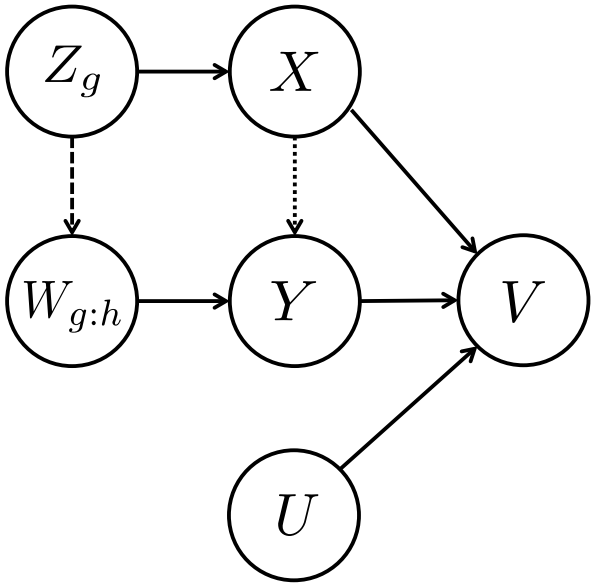

Figure 5.1: Plate diagram for the proposed model. The dashed segment represents the optional conditional dependence. Similarly, the dotted segment represents the regression dependence; if $\mathbf{B}_{g:h}$ is constrained to be zero, it is also not present.

rather than in a nested relationship with one set being primary and the other secondary. However, a hierarchy within the intrinsic subspaces is still present with $\boldsymbol{y}_n$ dependent on $\boldsymbol{x}_n$ through the regression terms $\mathbf{B}_{g:h}$.

Intrinsically Independent

As another alternative specification, we may assume $\boldsymbol{y}_n$ to be independent of $\boldsymbol{x}_n$ by constraining $\mathbf{B}_{g:h}$ to be zero, effectively removing this dependence from the model. This means the intrinsic variables themselves are uncorrelated and removes a sizable quantity of free parameters from the model. In this case, the model density is

$$f(\boldsymbol{\Gamma v};\boldsymbol{\theta}) = \sum_{g=1}^{G}\sum_{h=1}^{H_g} \pi_g \tau_{g:h} \phi_{p_x}(\boldsymbol{x};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\phi_{p_y}(\boldsymbol{y};\boldsymbol{\eta}_{g:h},\boldsymbol{\Lambda}_{g:h})\phi_{p_u}(\boldsymbol{u};\boldsymbol{\xi},\boldsymbol{\Psi}). \tag{5.5}$$

Most notably, in this variation the covariance matrices across all rotated intrinsic variables forms a block-diagonal structure $\text{diag}(\boldsymbol{\Sigma}_g, \boldsymbol{\Lambda}_{g:h}, \boldsymbol{\Psi})$.

Conditionally and Intrinsically Independent

We may also simultaneously specify both variations, in which case we have two independent finite Gaussian mixtures in two different orthogonal subspaces of the rotated data space. Composing the two modifications, the model density is

$$f(\boldsymbol{\Gamma v};\boldsymbol{\theta}) = \sum_{g=1}^{G}\sum_{h=1}^{H} \pi_g \tau_h \phi_{p_x}(\boldsymbol{x};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\phi_{p_y}(\boldsymbol{y};\boldsymbol{\eta}_h,\boldsymbol{\Lambda}_h)\phi_{p_u}(\boldsymbol{u};\boldsymbol{\xi},\boldsymbol{\Psi}). \tag{5.6}$$

This variation resembles the earlier Galimberti and Soffritti (2007) with two independent clusterings occurring in variable subsets and is a considerably simpler model. In this case, the total independence within intrinsic variables $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ in addition to class labels $z_{n,g}$ and $w_{n,h}$ effectively mean there is no nesting; however, for convenience we will continue to use the primary and secondary nomenclature simply to address the two different sets of clusterings.

Permutation Matrix as Rotation

The matrix $\boldsymbol{\Gamma}$ is specified as a general orthogonal matrix, representing a rotation in the space. If we were to restrict $\boldsymbol{\Gamma}$ to be a permutation matrix, we get as a special case the variable selection behaviour found in Galimberti and Soffritti (2007) and Galimberti et al. (2018). For the purposes of this work, we omit this case due to the difficulty in searching over the set of permutation matrices.

5.2.2 Identifiability

By inspecting Equation (5.3), the proposed model is a special case of a finite Gaussian mixture model with additional structure imposed on the mean and covariance parameters. This means the mean and covariance parameters are identifiable as their $\boldsymbol{\Gamma}^\top \bar{\boldsymbol{\mu}}_{g:h}$ and $\boldsymbol{\Gamma}^\top \bar{\boldsymbol{\Sigma}}_{g:h} \boldsymbol{\Gamma}$ expressions (Yakowitz and Spragins, 1968; Teicher, 1961; Holzmann et al., 2006). We show in this section sufficient conditions for the identifiability of the decomposed parameters $\boldsymbol{\Gamma}, \boldsymbol{\mu}_g, \boldsymbol{\eta}_{g:h}, \boldsymbol{\Sigma}_g, \boldsymbol{\Lambda}_{g:h}, \boldsymbol{\Psi}$. We use without proof that the finite Gaussian mixture model with density

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{g=1}^{G} \pi_g \phi_p(\boldsymbol{x}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \tag{5.7}$$

is identifiable in parameters $\pi_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g$ (Yakowitz and Spragins, 1968; Teicher, 1961; Holzmann et al., 2006).

Lemma 2 *A finite Gaussian mixture model with model density*

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{g=1}^{G} \pi_g \phi_p(\boldsymbol{x}; \boldsymbol{\Gamma}^\top \boldsymbol{\mu}_g, \boldsymbol{\Gamma}^\top \boldsymbol{\Sigma}_g \boldsymbol{\Gamma}) \tag{5.8}$$

where $\boldsymbol{\Gamma}$ is an orthogonal rotation matrix is identifiable up to a permutation of variables if one covariance matrix is diagonal with all distinct (eigen)values. Without loss of generality, we index the cluster with such a covariance matrix by index 1.

Proof 2 *Define* $\boldsymbol{\mu}'_g = \boldsymbol{\Gamma}^\top \boldsymbol{\mu}_g$ *and* $\boldsymbol{\Sigma}'_g = \boldsymbol{\Gamma}^\top \boldsymbol{\Sigma}_g \boldsymbol{\Gamma}$*, so that we may re-write the model density in the form of a finite Gaussian mixture as in (5.7)*

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{g=1}^{G} \pi_g \phi_p(\boldsymbol{x}; \boldsymbol{\mu}'_g, \boldsymbol{\Sigma}'_g)$$

for general mean vectors $\boldsymbol{\mu}'_g \in \mathbb{R}^p$ *and general symmetric positive-definite matrices* $\boldsymbol{\Sigma}'_g$*. Since* $\boldsymbol{\Gamma}$ *and* $\boldsymbol{\Gamma}^\top$ *are orthogonal matrices,* $\boldsymbol{\Sigma}'_g$ *has the same eigenvalues as* $\boldsymbol{\Sigma}_g$*. In particular, this means that* $\boldsymbol{\Sigma}'_1$ *has a unique eigendecomposition into* $\mathbf{PDP}^\top$ *for an (invertible) orthogonal matrix* $\mathbf{P}$ *and diagonal matrix* $\mathbf{D}$ *is unique up to an re-ordering of the columns. Thus, from* $\boldsymbol{\Sigma}'_1$ *we may define* $\boldsymbol{\Gamma} = \mathbf{P}$ *and recover* $\boldsymbol{\Sigma}_1 = \mathbf{D}$*. From the identifiability of finite Gaussian mixture density parameters in this form, we can invert the transformations to uniquely recover the* $\boldsymbol{\Gamma}, \boldsymbol{\mu}_g$ *and* $\boldsymbol{\Sigma}_g$ *parameters up to a permutation of variables. As well, we observe that the model in (5.8) has* $p(p-1)/2, Gp, p, (G-1)p(p+1)/2,$ *and* $G-1$ *free parameters in* $\boldsymbol{\Gamma}, \{\boldsymbol{\mu}_g\}_{g=1}^G, \boldsymbol{\Sigma}_1, \{\boldsymbol{\Sigma}_g\}_{g=2}^G,$ *and* $\{\pi_g\}_{g=1}^G$ *respectively, which totals* $G(p+1)(p+2)/2 - 1$*, matching the free parameter count of (5.7).*

Corollary 1 *The finite Gaussian mixture model with model densities as in Equations* (5.1) *to* (5.3) *is identifiable if* $\bar{\boldsymbol{\Sigma}}_{g:h}$ *is a diagonal matrix with distinct values for at least one* $g{:}h$*, up to a permutation of variables.*

Proof 3 *From Lemma 2, we have that the composed component parameters* $\bar{\boldsymbol{\mu}}_{g:h}$ *and* $\bar{\boldsymbol{\Sigma}}_{g:h}$ *are identifiable, in addition to mixing parameters* π_g *and* $\tau_{g:h}$*. By definition,* $\bar{\boldsymbol{\mu}}_{g:h}$ *and* $\bar{\boldsymbol{\Sigma}}_{g:h}$ *is a bijective transformation using their constituent* $\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\eta}_{g:h}, \boldsymbol{\Lambda}_{g:h}, \mathbf{B}_{g:h}, \boldsymbol{\xi}$ *and* $\boldsymbol{\Psi}$ *parameters.*

More specifically, $\boldsymbol{\xi}$ *and* $\boldsymbol{\Psi}$ *are identified by marginalizing the density in the corresponding intrinsic noise subspace (Kent, 1983).* $\boldsymbol{\mu}_g$ *and* $\boldsymbol{\Sigma}_g$ *are obtained in the same manner. By non-singularity of* $\boldsymbol{\Sigma}_g$*, solving the off-diagonal blocks* $\boldsymbol{\Sigma}_g \mathbf{B}_{g:h}^\top$ *or* $\mathbf{B}_{g:h} \boldsymbol{\Sigma}_g$ *uniquely yields* $\mathbf{B}_{g:h}$*. Finally, by re-arranging the corresponding blocks of* $\bar{\boldsymbol{\mu}}_{g:h}$ *and* $\bar{\boldsymbol{\Sigma}}_{g:h}$*, we also uniquely obtain* $\boldsymbol{\eta}_{g:h}$ *and* $\boldsymbol{\Lambda}_{g:h}$ *as required.*

For identifiability of the model's intrinsic subspace dimensions $\langle p_x, p_y, p_u \rangle$, we require further uniqueness constraints on the parameters. This prevents multiple possible choices of intrinsic subspace dimensions $\langle p_x, p_y, p_u \rangle$ yield identical model densities by minimizing the dimensions of p_x and p_y. As an illustrating example, we assume a intrinsically independent but conditionally dependent model with $p = 4$, $G = 2$, and $H_g = \langle 2, 2 \rangle$ while ignoring covariance matrices such that some $\boldsymbol{\Gamma}$ yields the component parameters

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \boldsymbol{\mu}_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \text{ and}$$

$$\boldsymbol{\eta}_{1:1} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \boldsymbol{\eta}_{1:2} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \boldsymbol{\eta}_{2:1} = \begin{bmatrix} -2 \\ 3 \end{bmatrix}, \boldsymbol{\eta}_{2:2} = \begin{bmatrix} -2 \\ 4 \end{bmatrix},$$

then we disallow the choice of subspace $\langle p_x, p_y, p_u \rangle = \langle 2, 2, 0 \rangle$ in favour of $\langle p_x, p_y, p_u \rangle = \langle 1, 3, 0 \rangle$ instead, with parameters

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}, \boldsymbol{\mu}_2 = \begin{bmatrix} -1 \\ -1 \\ -2 \end{bmatrix}, \begin{bmatrix} \eta_{1:1} \\ \eta_{1:2} \\ \eta_{2:1} \\ \eta_{2:2} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}.$$

Similarly, if primary subspaces have common parameters in $\boldsymbol{\mu}_g$ across all $g = 1, 2, ..., G$, then the common parameter would instead be associated with the noise subspace $\boldsymbol{\xi}$.

Remark 3 *When an eigenvalue λ appears more than once in a matrix $\boldsymbol{\Sigma}$, then it's diagonalization as $\mathbf{PDP}^\top$ is no longer unique as the eigenspace corresponding to λ has an infinite number of basis. As an example, if the first eigenvalue λ_1 has multiplicity 2, then for an arbitrary 2×2 rotation matrix $\mathbf{R}$ and a block-diagonal matrix*

$$\mathbf{Q} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

we obtain another valid diagonalization $\boldsymbol{\Sigma} = (\mathbf{PQ})\mathbf{D}(\mathbf{PQ})^\top$. This is akin to the identifiability problem in factor analysis whereby introducing an arbitrary rotation matrix does not change the model density.

Remark 4 *The identifiability results in this section assumes that the number of primary and secondary clusters G and H_g are known a priori. As well, we assume that the dimensions of the intrinsic subspaces $\langle p_x, p_y, p_u \rangle$ are known a priori.*

5.2.3 Model Parameters

Due to the complexity of the parameter space and the nested hierarchy of clusters, we enumerate the model parameters explicitly in this section. Firstly, the set of π_g parameters contribute $G-1$ free parameters due to the unit sum constraint; similarly, each set of $\tau_{g:h}$ for each g contributes $H_g - 1$ free parameters. Each primary clustering g contributes $p_x + p_x(p_x+1)/2$ parameters from $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$. Each secondary clustering $g{:}h$ contributes $p_y + p_y(p_y+1)/2$ from $\boldsymbol{\eta}_{g:h}$ and $\boldsymbol{\Lambda}_{g:h}$, with an additional $p_x p_y$ if a regression $\mathbf{B}_{g:h}$ is present. In the conditionally independent variation, each secondary cluster h contributes the aforementioned quantities once overall instead of once for each g. Finally, the intrinsic noise variables $\boldsymbol{\xi}$ and $\boldsymbol{\Phi}$ contribute $2p_u$ total.

In a general $p \times p$ rotation matrix, there are $p(p-1)/2$ free parameters. In the case of $\boldsymbol{\Gamma}$, the first p_x columns span the intrinsic subspace of the primary clustering. If we redefine this subspace in terms of a new set of p_x orthonormal basis vectors, we may re-write all $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ in this new basis to re-obtain the same density function by cancelling out the rotation. This parameterization ambiguity means there are $p_x(p_x-1)/2$ redundant parameters in $\boldsymbol{\Gamma}$. By the same argument, there are $p_y(p_y-1)/2$ redundant parameters for the secondary clustering subspace as well. This results in a net total of $p(p-1)/2 - p_x(p_x-1)/2 - p_y(p_y-1)/2$ free parameters in $\boldsymbol{\Gamma}$.

As the number of parameters changes with the choice of model variation, a tabular listing of the variations and their associated number of free parameters is provided in Table 5.1.

Table 5.1: List of model variations and the number of free parameters for nested Gaussians. As a shorthand, we use $n_x = p_x + p_x(p_x+1)/2$, $n_y = p_y + p_y(p_y+1)/2$, and $n_{\boldsymbol{\Gamma}} = p_x p_y + p_x p_u + p_y p_u - p_u/2 + p_u^2/2$ to denote the number of free parameters in primary cluster component parameters $\{\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\}$, secondary cluster non-regression parameters $\{\boldsymbol{\eta}_{g:h}, \boldsymbol{\Lambda}_{g:h}\}$, and rotation matrix $\boldsymbol{\Gamma}$, respectively.

Intr. Indep.	Cond. Indep.	Free Parameters
No	No	$G(n_x) + \sum_{g=1}^{G} H_g(n_y + p_x p_y + 1) + 2p_u - 1 + n_{\boldsymbol{\Gamma}}$
No	Yes	$G(n_x + 1) + H(n_y + p_x p_y + 1) + 2p_u - 2 + n_{\boldsymbol{\Gamma}}$
Yes	No	$G(n_x) + \sum_{g=1}^{G} H_g(n_y + 1) + 2p_u - 1 + n_{\boldsymbol{\Gamma}}$
Yes	Yes	$G(n_x + 1) + H(n_y + 1) + 2p_u - 2 + n_{\boldsymbol{\Gamma}}$

5.3 Estimation

We provide an Expectation-Maximization EM algorithm (Dempster et al., 1977) for estimation of the proposed model's parameters including the aforementioned variations. Let class membership indicators be denoted by $z_{n,g}$ and $w_{n,g:h}$ for primary and secondary clusterings, respectively. In a clustering context, we consider these values to be unobserved and latent. For observed manifest data variables $\boldsymbol{v}_n$ composed row-wise to form a data matrix $\mathbf{V}$, we have the expected complete data log-likelihood

$$
\begin{aligned}
Q(\mathbf{V}; \boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{g=1}^{G_\cdot} \sum_{h=1}^{H_g} \hat{z}_{n,g} \hat{w}_{n,g:h} \Big[& \log \pi_g + \log \tau_{g:h} + \log \phi_{p_x}\left(\boldsymbol{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\right) \\
& + \log \phi_{p_y}\left(\boldsymbol{y}_n; \boldsymbol{\eta}_{g:h} + \mathbf{B}_{g:h}\boldsymbol{x}_n, \boldsymbol{\Lambda}_{g:h}\right) + \log \phi_{p_u}\left(\boldsymbol{u}_n; \boldsymbol{\xi}, \boldsymbol{\Psi}\right) \Big].
\end{aligned}
$$

Alternatively, we may write this explicitly in terms of the manifest variables $\boldsymbol{v}_n = \boldsymbol{\Gamma}^\top \boldsymbol{r}_n$ by expressing $\boldsymbol{\Gamma}$ as a block-matrix with row-blocks

$$
\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\Gamma}_x \\ \boldsymbol{\Gamma}_y \\ \boldsymbol{\Gamma}_u \end{bmatrix}
$$

such that $\boldsymbol{\Gamma}_x, \boldsymbol{\Gamma}_y, \boldsymbol{\Gamma}_u$ are of sizes $p_x \times p, p_y \times p, p_u \times p$ respectively. Thus, we may write

$$Q(\mathbf{V}; \boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{g=1}^{G:} \sum_{h=1}^{H_g} \hat{z}_{n,g} \hat{w}_{n,g:h} \Big[\log \pi_g + \log \tau_{g:h} + \log \phi_{p_x} \left(\boldsymbol{\Gamma}_x \boldsymbol{v}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g \right) + \log \phi_{p_y} \left(\boldsymbol{\Gamma}_y \boldsymbol{v}_n; \boldsymbol{\eta}_{g:h} + \mathbf{B}_{g:h} \boldsymbol{x}_n, \boldsymbol{\Lambda}_{g:h} \right) + \log \phi_{p_u} \left(\boldsymbol{\Gamma}_u \boldsymbol{v}_n; \boldsymbol{\xi}, \boldsymbol{\Psi} \right) \Big].$$

We use $\hat{z}_{n,g}$ here to denote the probability of observation n being in primary cluster g given the parameters $\boldsymbol{\theta}$. Similarly, we use $\hat{w}_{n,g:h}$ for the same observation's probability of being in secondary cluster $g{:}h$. We also note the estimate for the joint membership $\mathrm{P}(z_{n;g,h} = 1 \mid \boldsymbol{v}_n)$ is the product of the marginal $\hat{z}_{n,g}$ and conditional $\hat{w}_{n,g:h}$. In this estimation procedure, we treat the parameter set $\boldsymbol{\theta}$ as specified in Section 5.2, and take the intrinsic subspace dimensions p_x, p_y, p_u, number of clusters G, H_g, and the choice of intrinsic and conditional independence as fixed. We search over these parameters outside of the EM algorithm as part of the model selection procedure.

5.3.1 Initialization

We employ different strategies to obtain starting values for conditionally dependent and conditionally independent models. In all cases, we initialize $\boldsymbol{\Gamma}$ using the right-singular vectors from a singular value decomposition (SVD) of the scaled observed manifest data. The implied intrinsic subspace ordering provides the directions of greatest variation to the primary clustering. This effect is consistent with an intuition that primary clustering would exhibit greater variation than the nested secondary clustering; for example, variation over nations could be expected to be larger than variation in provinces/states.

To generate starting values for a conditionally dependent fit, we first fit a finite Gaussian mixture model using the *mclust* package (Scrucca et al., 2016) on the data rotated by the initial $\boldsymbol{\Gamma}$ with $\sum_{g=1}^{G} H_g$ clusters to obtain cluster probabilities, which we assign to the joint membership probability $\hat{z}_{n,g}\hat{w}_{n,g:h}$ in an arbitrary order. As a second starting value, we also search through all permutations of the first p_x dimensions of the *mclust* fitted means to distinguish G primary clusters by minimizing the total within-cluster sum square error; in effect, this is akin to k-means fit except the cluster sizes are pre-specified by H_g.

To generate starting values for a conditionally independent fit, we use the intrinsic variables $\boldsymbol{x}_n$ given by the above-initialized $\boldsymbol{\Gamma}$ to fit a finite Gaussian mixture model with G clusters, and likewise for $\boldsymbol{y}_n$ with H clusters. The second starting value is again to fit a $G \times H$ *mclust* model, and find not only the best permutation of G groups on the leading p_x dimensions of the fitted means to generate a primary clustering but also the best permutation of H groups on the subsequent p_y dimensions to generate a secondary clustering.

To combat a multimodal and divergent likelihood landscape, we generate multiple random starting values to perform Mini-EM (Biernacki et al., 2003), where we repeatedly generate starting values using a randomly subsampled fraction of the data and select the best log-likelihood after performing a modest number of EM updates on each such initialization. Unless otherwise specified, we perform 100 random starts using this Mini-EM procedure each receiving 100 EM updates. The corresponding best parameters are considered the actual initial parameters for the remainder of the EM procedure. When Mini-EM is not desired for computational complexity reasons, we initialize without subsampling with both methods and use the best log-likelihood after 100 EM updates.

Altogether, the initialization provides the primary and secondary cluster probabilities $\hat{z}_{n,g}$ and $\hat{w}_{n,g:h}$ (or $\hat{w}_{n,h}$ for the conditionally independent model). In turn, these initial probabilities are sufficient for the maximization step defined in Section 5.3.2 to produce initial parameter estimates for the primary and secondary clusterings. If some observations have known class labels, we use the corresponding semi-supervised *mclust* model with the available class information in generating starting values.

5.3.2 Expectation-Maximization Algorithm

Expectation Step

Using the parameters at the current iteration, we update the expectations for memberships using

$$\hat{z}_{n,g} = \frac{\pi_g \phi(\boldsymbol{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)}{\sum_{l=1}^{G} \pi_l \phi(\boldsymbol{x}_n; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}, \quad \hat{w}_{n,g:h} = \frac{\tau_{g:h} \phi(\boldsymbol{y}_n; \boldsymbol{\eta}_{g:h} + \mathbf{B}_{g:h}\boldsymbol{x}_n, \boldsymbol{\Lambda}_{g:h})}{\sum_{l=1}^{H_g} \tau_{g:l} \phi(\boldsymbol{y}_n; \boldsymbol{\eta}_{g:l} + \mathbf{B}_{g:l}\boldsymbol{x}_n, \boldsymbol{\Lambda}_{g:l})}.$$

To extend the proposed estimation procedure to cover semi-supervised and fully supervised contexts, we may adjust the expectation step to reflect known cluster assignments. For example, if the cluster assignment for an observation n is known to be in primary cluster g and secondary cluster h, then we take $z_{n,g} = 1$ and $w_{n,g:h} = 1$ with all other z and w values zero. Then the above expectation step update is applied for all other observations for which the assignment is not known. In a fully supervised context, the expectation step can be skipped in its' entirety due to all z and w values being known.

Additionally, it is possible to perform semi-supervised clustering when only the primary labelling is known. This can be useful if primary class labels are available but a secondary clustering is assumed or suspected. In this case, we take $z_{n,g} = 1$ as fixed for the observation with known primary cluster g and only allow $w_{n,g:h}$ to be estimated by the above update. The reverse scenario where a secondary clustering is known but a primary clustering is suspected is more convoluted due to the need to assign the a priori secondary clusters to unknown primary clusters. We omit the consideration of this latter case in this work.

Maximization Step

Given the current iteration's $\boldsymbol{\Gamma}$, the updates for the remaining parameters can be computed. The marginal and conditional cluster probabilities are computed using

$$\pi_g = \frac{1}{N} \sum_{n=1}^{N} \hat{z}_{n,g}, \quad \tau_{g:h} = \frac{1}{N} \sum_{n=1}^{N} \hat{w}_{n,g:h}.$$

Within the primary clustering, the component distributional parameters updates are updated with

$$\boldsymbol{\mu}_g = \frac{\sum_{n=1}^{N}\sum_{h=1}^{H_g}\hat{z}_{n,g}\hat{w}_{n,g:h}\boldsymbol{x}_n}{\sum_{n=1}^{N}\sum_{l=1}^{H_g}\hat{z}_{n,g}\hat{w}_{n,g:l}},$$
$$\boldsymbol{\Sigma}_g = \frac{\sum_{n=1}^{N}\sum_{h=1}^{H_g}\hat{z}_{n,g}\hat{w}_{n,g:h}(\boldsymbol{x}_n-\boldsymbol{\mu}_g)(\boldsymbol{x}_n-\boldsymbol{\mu}_g)^\top}{\sum_{n=1}^{N}\sum_{l=1}^{H_g}\hat{z}_{n,g}\hat{w}_{n,g:l}}.$$

For the secondary clustering parameters in the intrinsically dependent case, we treat $\boldsymbol{\eta}_{g:h}$ and $\mathbf{B}_{g:h}$ together as intercepts and slopes, respectively. This leads to the simultaneous updates

$$\begin{bmatrix}\boldsymbol{\eta}_{g:h} & \mathbf{B}_{g:h}\end{bmatrix} = (\tilde{\mathbf{X}}^\top \operatorname{diag}(\boldsymbol{z}_{g:h})\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^\top \operatorname{diag}(\boldsymbol{z}_{g:h})\mathbf{Y},$$
$$\boldsymbol{\Lambda}_{g:h} = \frac{\sum_{n=1}^{N}\hat{w}_{n,g:h}(\boldsymbol{y}_n-(\boldsymbol{\eta}_{g:h}+\mathbf{B}_{g:h}\boldsymbol{x}_n))(\boldsymbol{y}_n-(\boldsymbol{\eta}_{g:h}+\mathbf{B}_{g:h}\boldsymbol{x}_n))^\top}{\sum_{n=1}^{N}\hat{w}_{n,g:h}},$$

where $\begin{bmatrix}\mathbf{X} & \mathbf{Y} & \mathbf{U}\end{bmatrix} = \mathbf{V}\boldsymbol{\Gamma}^\top$, $\tilde{\mathbf{X}} = \begin{bmatrix}\mathbf{1} & \mathbf{X}\end{bmatrix}$, and $\boldsymbol{z}_{g:h}$ is a vector of $\hat{z}_{n,g}\hat{w}_{n,g:h}$ estimates. In the non-regression case, where $\mathbf{B}_{g:h}$ is constrained to be zero, $\boldsymbol{\eta}_{g:h}$ and $\boldsymbol{\Lambda}_{g:h}$ updates become the weighted mean and covariance using intrinsic variables $\mathbf{Y}$, respectively. In other words,

$$\boldsymbol{\eta}_{g:h} = \frac{\sum_{n=1}^{N}\hat{w}_{n,g:h}\boldsymbol{y}_n}{\sum_{n=1}^{N}\hat{w}_{n,g:h}}, \quad \boldsymbol{\Lambda}_{g:h} = \frac{\sum_{n=1}^{N}\hat{w}_{n,g:h}(\boldsymbol{y}_n-\boldsymbol{\eta}_{g:h})(\boldsymbol{y}_n-\boldsymbol{\eta}_{g:h})^\top}{\sum_{n=1}^{N}\hat{w}_{n,g:h}}.$$

If diagonal covariance matrices are assumed in the model specification, we set off-diagonal values of $\boldsymbol{\Lambda}_{g:h}$ to zero.

Finally, in the intrinsic noise variables, we use the updates

$$\boldsymbol{\xi} = \frac{\sum_{n=1}^{N}\boldsymbol{u}_n}{N}, \quad \boldsymbol{\Psi} = \operatorname{diag}\frac{\sum_{n=1}^{N}(\boldsymbol{u}_n-\boldsymbol{\xi})(\boldsymbol{u}_n-\boldsymbol{\xi})^\top}{N},$$

where the diag operator sets all values off the main diagonal to zero. In practice, we pre-center the data so that $\boldsymbol{\xi} \equiv \mathbf{0}$ during the estimation procedure, which simplifies the process.

After updating the above parameters, we update $\boldsymbol{\Gamma}$ under the orthogonality constraint $\boldsymbol{\Gamma}^\top\boldsymbol{\Gamma} = \mathbf{I}$ using the Majorization-Minimization (MM) procedure in Kiers (2002). The

surrogate function of the EM algorithm in terms of $\mathbf{\Gamma}$ only is

$$Q(\mathbf{\Gamma}; \mathbf{V}) = \text{constant} + 2 \operatorname{Tr}\left[\sum_{g=1}^{G}\sum_{h=1}^{H_g}\left(\sum_{n=1}^{N}\hat{z}_{n,g}\hat{w}_{n,g:h}\boldsymbol{v}_n\right)\bar{\boldsymbol{\mu}}_{g:h}^{\top}\bar{\mathbf{\Sigma}}_{g:h}^{-1}\mathbf{\Gamma}\right] - \sum_{g=1}^{G}\sum_{h=1}^{H_g}\operatorname{Tr}\left[\bar{\mathbf{\Sigma}}_{g:h}\mathbf{\Gamma}\mathbf{W}_{g:h}\mathbf{\Gamma}^{\top}\right], \tag{5.9}$$

where $\mathbf{W}_{g:h} = \sum_{n=1}^{N}\hat{z}_{n,g}\hat{w}_{n,g:h}\boldsymbol{v}_n\boldsymbol{v}_n^{\top}$. The expression (5.9) here follows the form of expression (1) in Kiers (2002) with $\mathbf{\Gamma}$ being the parameter in which to optimize. Moreover, since the desired constraint is orthonormality $\mathbf{\Gamma}^{\top}\mathbf{\Gamma} = \mathbf{I}$ and the matrices $\bar{\mathbf{\Sigma}}_{g:h}$ and $\mathbf{W}_{g:h}$ are positive semi-definite we may take the majorizing function on page 164 of Kiers (2002) as our surrogate objective function with the associated $\mathbf{F}$ matrix identified from the above (5.9) as

$$\mathbf{F} = 2\sum_{g=1}^{G}\sum_{h=1}^{H_g}\left(\sum_{n=1}^{N}\hat{z}_{n,g}\hat{w}_{n,g:h}\boldsymbol{v}_n\right)\bar{\boldsymbol{\mu}}_{g:h}^{\top}\bar{\mathbf{\Sigma}}_{g:h}^{-1} - 2\sum_{g=1}^{G}\sum_{h=1}^{H_g}\left[\mathbf{W}_{g:h}\mathbf{\Gamma}_{\text{old}}^{\top}\left(\bar{\mathbf{\Sigma}}_{g:h}^{-1} + \lambda_{g:h}\mathbf{I}\right)\right],$$

where $\lambda_{g:h}$ is the maximum eigenvalue of $\mathbf{W}_{g:h}$. The update for $\mathbf{\Gamma}$ is $\mathbf{\Gamma}_{\text{new}} = \mathbf{Q}\mathbf{P}^{\top}$ with $\mathbf{P}$ and $\mathbf{Q}$ from the singular value decomposition $\operatorname{svd}(-\mathbf{F}) = \mathbf{P}\mathbf{D}\mathbf{Q}^{\top}$.

We observe we may also rearrange Equation (5.9) to be a function of $\mathbf{\Gamma}^{\top}$ with the same form of objective function of Kiers (2002) by transposing the arguments of the traces and applying cyclic permutation, neither of which affect the value of the trace operator. This yields a slightly different MM algorithm for $\mathbf{\Gamma}$. We apply these two MM algorithm updates in pairs up to 100 times or until the average element-wise change in $\mathbf{\Gamma}$ is less than 10^{-8}. A complete derivation of this parameter update is given in Appendix C.1.

5.3.3 Computational Considerations

Convergence

To assess convergence of the model during the estimation procedure, we compute the log-likelihood for the parameters at each EM iteration. If the improvement is below the

defined threshold of 10^{-4} or if the number of iterations exceeds 100,000, we terminate the estimation procedure.

In order to aid identification of the primary and secondary clustering subspaces, we first run the estimation procedure on a scaled version of the initial data. This prevents the initialization for $\mathbf{\Gamma}$ from assigning the variables with the largest range and capturing differences in measurement scale. Using the parameters from this first estimation as starting values, we run the next estimation procedure on the unscaled data. We hold the cluster probabilities $\hat{z}_{n,g}$ and $\hat{w}_{n,g:h}$ fixed at the previous values until convergence, at which point we allow updates to them until final convergence.

Model Metrics

Over the set of pre-specified models as specified by p_x, p_y, G, and $\{H_g\}_{g=1}^{G}$, we may run the estimation procedure for each combination thereof. Moreover, we may also select over the three different types of model variations: conditional independence, intrinsic independence, and isotropic covariance matrices. This yields a further factor of eight possible models. Among these, we select the best model using the Bayesian Information Criterion (BIC) (Schwarz, 1978) defined as

$$\text{BIC} = k \log N - 2\ell(\boldsymbol{\theta}; \mathbf{V}),$$

where k is the number of free parameters of the model from Section 5.2.3 and ℓ is the log-likelihood for the parameters $\boldsymbol{\theta}$. In this parameterization of BIC lower values are considered better.

In addition, we apply the adjusted Rand index (ARI) of Hubert and Arabie (1985) to examine the degree of similarity with observed class labels. In the proposed model, we have restricted ourselves to only a primary and a secondary clustering. When the data allows, we will compare both levels of nested labels against available class labels and present the ARI for each comparison. For completeness, we note that the recovered labels may not necessarily align with a single extant observable class label.

5.4 Simulation Study

In this section, we perform a simulation study to examine the performance of the model under a variety of situations. To that end, we define the honeycomb family of datasets as examples of the proposed model. We investigate different values of intrinsic subspace dimension, the well-separatedness of the clusters, and the number of observations.

5.4.1 Synthetic Honeycomb Dataset

In this section, we describe the synthetic honeycomb dataset generated from the proposed model. In each dataset, we prescribe $G = 3$ primary clusters with $H_g = \langle 2, 3, 2 \rangle$ secondary clusters. We place the three primary clusters in the corresponding intrinsic subspace at $\{-10, 0, 10\} \times \mathbf{1}_{p_x}$ with covariance $\mathbf{\Sigma}_g$ defined element-wise as $[\mathbf{\Sigma}_g]_{ij} = \lambda 2^{-|i-j|}$ for some variance scaling parameter $\lambda > 0$. In the secondary clustering, we specify the means in the secondary intrinsic subspace as $\{-5, 5\} \times \mathbf{1}_{p_y}$, $\{-10, 0, 10\} \times \mathbf{1}_{p_y}$, and $\{-5, 5\} \times \mathbf{1}_{p_y}$ for the three primary clusters, respectively. The secondary covariances $\mathbf{\Lambda}_{g:h}$ are defined in the same manner as $\mathbf{\Sigma}_g$. The regression coefficients are given between $[\mathbf{B}_{g:h}]_{ij} = (-1)^{i+j} \times 2^{-|g-h|}$. In the noise subspace, we specify an isotropic noise subspace covariance $\mathbf{\Psi} = \lambda \mathbf{I}_{p_u \times p_u}$ for simplicity, whereas the model posits a diagonal $\mathbf{\Psi}$ throughout. The rotation $\mathbf{\Gamma}$ is a randomly generated orthogonal matrix, and rotates the resulting data to form the dataset. A visualization of this dataset in its observed rotated and original unrotated forms is given in Figure 5.2.

5.4.2 Intrinsic Subspace and Class Label Recovery

We explore the effect of data dimensionality by varying the intrinsic subspace dimensions p_x, p_y, p_u simultaneously from one to three. Additionally, we examine the effect of the number of observations N by sampling 100, 1000, and 2000 observations from each cluster. Lastly, we vary the degree of separation and overlap of the clusters by varying the variance scaling parameter λ over the values 1, 2, and 3.

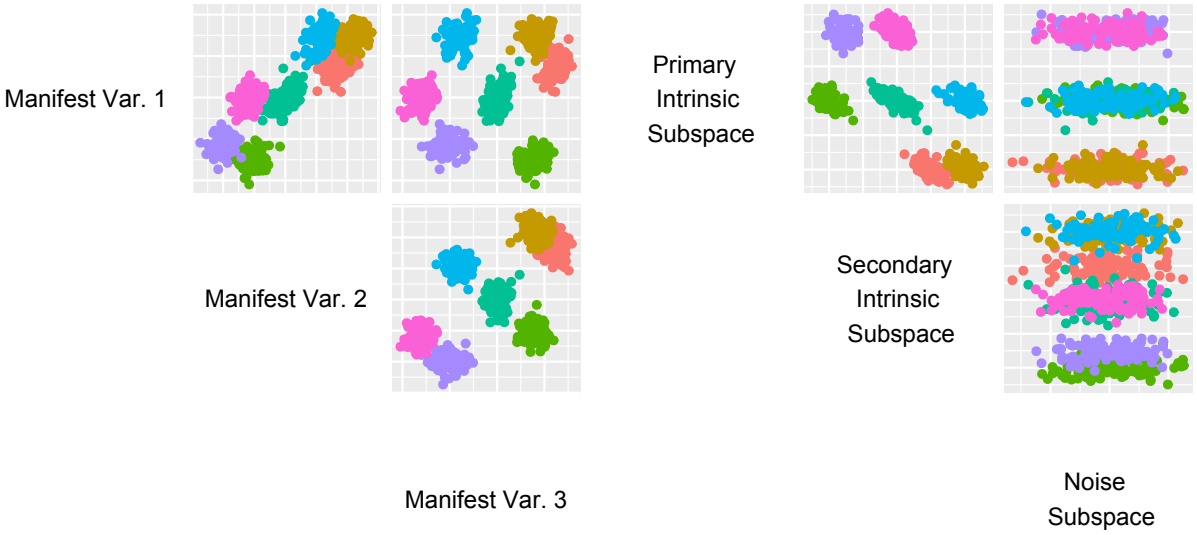

Figure 5.2: An example of the simulation dataset presented as a scatterplot, with dataset parameters $p_x = p_y = p_u = 1$, 100 observations, and $\lambda = 1$. The observed manifest variables (left) are a rotation of the intrinsic variables (right).

The behavior of interest is the recovery of the intrinsic subspaces of $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{U}$ given by the fitted $\boldsymbol{\Gamma}$ matrix. This is accomplished by using the Grassmann distance between the columns of the fitted and true $\boldsymbol{\Gamma}$ matrices corresponding to the bases of the respective intrinsic subspaces. As the decomposition of the rotation matrix $\boldsymbol{\Gamma}$ yields three different subspaces; one for each of $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{U}$, we may wish to compare the similarity of these subspaces. Noting that a distance must be invariant to a change of basis within a subspace, we turn to the Grassmann distance for this invariance. This distance arises over a metric space over finite dimensional linear subspaces (Lee, 2012). For orthogonal matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ of dimension $d \times k$ where columns represent an orthonormal basis of a k-dimensional subspace of $\mathbb{R}^d$, the Grassmann distance can be calculated as $\sum_{i=1}^{k} \arccos \sigma_i$ for σ_i being a singular value of $\mathbf{M}_1^\top \mathbf{M}_2$. Consequently, we obtain three Grassmann distances d_X, d_Y, d_U, one for each of the intrinsic subspaces. We combine these into a unified distance $d = (d_X^2 + d_Y^2 + d_U^2)^{1/2}$ and use this distance throughout the remainder of this section. For each simulated dataset, we fit the conditionally and intrinsically dependent model with $G = 3$ and $H_g = \langle 2, 3, 2 \rangle$. The simulation results in terms of ARI and the combined Grassmann distance are presented in Table 5.2.

A graphical representation for each of the generating model specifications as in Table 5.2 is provided in Figure 5.3, marginalizing over the effects of the other parameters.

From this simulation study, we find that the clustering becomes more inconsistent in both the ARI and combined Grassmann distance as the number of dimensions p_x, p_y, p_u increases. The effect of overlap controlled by parameter λ seems to play a minimal role in the classification and rotation parameter $\boldsymbol{\Gamma}$ recovery. We remark that the initialization process is only a heuristic; the initial rotation $\boldsymbol{\Gamma}^{(0)}$ is obtained via an SVD decomposition and allocates variation in decreasing order to the primary, secondary, and noise subspaces. In this synthetic dataset, as seen in Figure 5.2, the dataset does not conform to this assumption.

Table 5.2: Model metrics for simulated dataset over multiple parameter combinations. Each value is the average over 100 replications, with standard deviation in parentheses.

p_x, p_y, p_u	N	λ	ARI	Grassmann distance
1	100	1	0.5254 (0.0376)	0.5870 (0.7245)
1	100	2	0.4713 (0.0107)	0.6219 (0.7197)
1	100	3	0.4649 (0.0135)	0.8557 (0.7708)
1	1000	1	0.4158 (0.0463)	1.2477 (0.4106)
1	1000	2	0.4718 (0.0124)	1.5000 (0.6684)
1	1000	3	0.4738 (0.0006)	0.6710 (0.9548)
1	2000	1	0.4513 (0.0605)	1.3371 (0.4801)
1	2000	2	0.4599 (0.0181)	1.4375 (0.6943)
1	2000	3	0.4654 (0.0109)	0.9663 (0.8832)
2	100	1	0.5136 (0.0465)	1.5135 (0.5054)
2	100	2	0.4731 (0.0434)	1.6990 (0.5206)
2	100	3	0.4499 (0.0304)	1.9428 (0.4552)
2	1000	1	0.4466 (0.0291)	2.2800 (0.3384)
2	1000	2	0.4431 (0.0264)	2.2707 (0.2994)
2	1000	3	0.4407 (0.0300)	2.2778 (0.2665)
2	2000	1	0.4232 (0.0712)	2.2698 (0.3564)
2	2000	2	0.4223 (0.0411)	2.2845 (0.3385)
2	2000	3	0.4120 (0.0320)	2.2278 (0.3111)
3	100	1	0.5613 (0.1684)	2.7724 (0.3053)
3	100	2	0.4700 (0.0859)	2.8348 (0.2758)
3	100	3	0.4368 (0.0824)	2.8280 (0.2722)
3	1000	1	0.5381 (0.1509)	2.6910 (0.3412)
3	1000	2	0.4979 (0.1236)	2.7572 (0.3554)
3	1000	3	0.4327 (0.1147)	2.6519 (0.3636)
3	2000	1	0.5580 (0.1813)	2.4970 (0.3408)
3	2000	2	0.4484 (0.2001)	2.5284 (0.4079)
3	2000	3	0.4112 (0.1117)	2.6363 (0.3776)

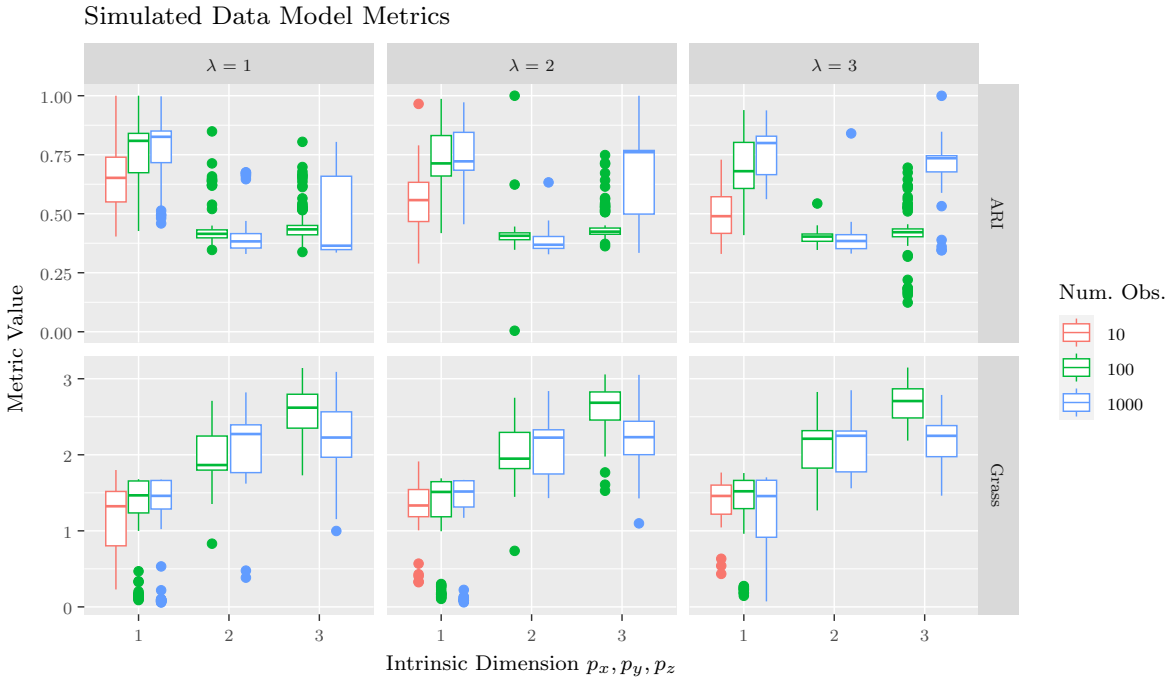

Figure 5.3: ARI, and $\mathbf{\Gamma}$ Grassmann distances for the simulated dataset across multiple parameter configurations.

5.4.3 Intrinsic Subspace Dimension Recovery

We define a synthetic dataset to investigate the recovery of the intrinsic subspace dimensions with $G = 3$ and $H_g = \langle 2, 3, 2 \rangle$ as in the synthetic honeycomb data, and fixed $p_x = p_y = p_u = 2$, $\lambda = 1$, and $N = 1000$. We fix dataset component parameters such that

$$\begin{aligned}
\boldsymbol{\mu}_1 &= \begin{bmatrix} 3 \\ 5 \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{\mu}_3 = \begin{bmatrix} -5 \\ 3 \end{bmatrix}, \\
\boldsymbol{\eta}_{1:1} &= \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \quad \boldsymbol{\eta}_{1:2} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \\
\boldsymbol{\eta}_{2:1} &= \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \quad \boldsymbol{\eta}_{2:2} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{\eta}_{2:3} = \begin{bmatrix} 4 \\ -3 \end{bmatrix}, \\
\boldsymbol{\eta}_{3:1} &= \begin{bmatrix} -2 \\ -1 \end{bmatrix}, \quad \boldsymbol{\eta}_{3:2} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \\
\boldsymbol{\Sigma}_g &= \boldsymbol{\Lambda}_{g:h} = \boldsymbol{\Psi} = \mathbf{I}_2, \quad \mathbf{B}_{g:h} = \mathbf{0},
\end{aligned}$$

and draw a random $\boldsymbol{\Gamma}$ uniformly from the space of orthogonal matrices. This dataset guarantees that the three primary components utilize the full $p_x = 2$ dimensions by not being collinear, with a similar imposition on the secondary cluster components. We evaluate a model space over all combinations of $1 \leq p_x, p_y \leq 3$ for each of 100 such dataset to determine the model selection effectiveness of BIC in an exhaustive search. The remaining model space is restricted to be conditionally dependent but intrinsically independent, with $G = 3$ and $H_g = \langle 2, 3, 2 \rangle$. Table 5.3 shows the average BIC across 100 replications at each tried p_x, p_y, p_u combination as well as the proportion of replications that selected the corresponding intrinsic subspace dimension. Indeed, we see that the class labels are best recovered under the correct model, which in turn is also selected by BIC in 94 of 100 dataset replications.

Table 5.3: Model selection results under the simulated dataset for $p_x = p_y = p_u = 2$. BIC averages are computed average over 100 replications, with ΔBIC being the difference in averages against the best value. The frequency of replicated datasets selecting a particular model and the average ARI is also given. All values in parentheses are standard deviations.

p_x	p_y	Freq.	BIC	ΔBIC	ARI
1	1	0	25294 (734)	+1896	0.426 (0.047)
1	2	0	24436 (478)	+1039	0.657 (0.057)
1	3	2	23479 (545)	+82	0.961 (0.064)
2	1	0	24362 (102)	+964	0.662 (0.026)
2	2	94	23398 (114)	0	0.970 (0.008)
2	3	3	23448 (117)	+51	0.965 (0.026)
3	1	0	24061 (863)	+664	0.700 (0.085)
3	2	0	23672 (306)	+274	0.859 (0.129)
3	3	1	23488 (145)	+90	0.958 (0.046)

5.5 Real-World Datasets

In this section, we apply the proposed clustering model and its variations to real-world datasets. Specifically, we examine the *Leptograpsus* crabs dataset (Campbell and Mahon, 1974) and the olive oil dataset (Forina et al., 1983) in both unsupervised and semi-supervised clustering contexts, and also examine the `Cars93` dataset (Lock, 1993) and handwritten digits dataset (van Breukelen et al., 1998; van Breukelen and Duin, 1998; Jain et al., 2000) in an unsupervised manner. For comparison, we compare against a baseline finite Gaussian mixture model with parsimonious covariance matrices as fitted by *mclust*. To include parsimony in this case, we allow *mclust* to estimate and select over all 14 covariance matrix specifications (Celeux and Govaert, 1995). For understandability and consistency, we use the covariance type abbreviations as specified in *mclust*. We also only consider comparable models with multivariate normal distributions to maintain consistency in cluster densities; alternatives such as multivariate-t or the skew-normal may exhibit better

performance on the datasets. When results or code are available from related works in the literature, the fitted model summaries are presented as well. To simplify estimation, we pre-process the manifest variables by centering the data, which yields $\boldsymbol{\xi} \equiv \mathbf{0}$; simplifying estimation slightly.

5.5.1 Crabs Dataset

This dataset describes 200 observations of *Leptograpsus* crabs spanning two sexes and two species (Campbell and Mahon, 1974). The observed manifest variables are five morphometric measurements of the crab itself. As an initial observation, some careful rotation of the dataset reveals the two clusterings essentially occur on independent subsets of the data.

We evaluate the proposed model for all combinations of parameters of $p_x \leq 4$, $p_y \leq 4$, $G \leq 3$, $H_g \leq 3$ in conjunction with the variations of regression and conditional dependence. Among these, we select the best model parameters using BIC. We also discard fitted models with a covariance eigenvalue below 10^{-3} to exclude models with degenerate solutions with near-singular covariance parameters and a divergent log-likelihood. As such, BIC selects $p_x = 2$, $p_y = 2$, $G = H = 2$; including regression but without conditional dependence. For simplicity, we have only presented this model from the four possible independence configurations; the remaining models produce particularly unreasonable results featuring empty clusters or near-singular covariance matrices, and do not have optimal BIC. As well, we allow for primary and secondary covariances $\boldsymbol{\Sigma}_g$ and $\boldsymbol{\Lambda}_{g:h}$ to be either fully-varying or diagonal, akin to the *mclust* VVV and VVI models, respectively. Model metrics are presented in Table 5.4 with comparisons to some models from the literature. We find a better BIC with the proposed model on the *Leptographsus* crabs data primarily due to the reduction in the number of parameters compared to Model 5 of (Galimberti et al., 2018). The ARI is slightly decreased but demonstrates similar behaviour in that the species variable is fully recovered. Accordingly, we can visualize the data in the resultant rotated directions in Figure 5.4. In this figure, we find and depict the intrinsic variables in which the primary clustering appears, which may be used to enhance further graphical analyses or aid understanding within the *Leptographsus* context. From the selected model of Table 5.4,

Table 5.4: Fitted model metrics for *Leptograpsus* dataset using the proposed model, finite Gaussian mixtures using *mclust*, and the model of Galimberti et al. (2018). The best model is selected from each family by selecting the best BIC, where lower is better.

	Nested Gaussians	*mclust*	Galimberti et al. (2018), M_5
Number of Clusters	$G = 2$, $H = 2$	4	$K_1 = 2$, $K_2 = 2$
Dimensions	$p_x = 2$, $p_y = 2$	5	3, 2
Number of Parameters	40	68	68
Intrinsic Independence	No	–	No
Conditional Independence	Yes	–	Yes
Covariance Type	Full	EEV	–
BIC	2771.066	2842.30	2808.3
ARI (overall)	0.8143	0.7939	0.873
ARI (sex)		0.3504	
Primary	0.7211		0.8091
Secondary	-0.0040		-0.0051
ARI (species)		0.5004	
Primary	-0.0051		-0.0044
Secondary	1.0000		1.0000

we obtain the rotation matrix which rotates the data into the form seen in Figure 5.4

$$\boldsymbol{\Gamma} = \begin{bmatrix} 0.2858 & -0.1803 & 0.8214 & -0.4530 & -0.0767 \\ -0.3896 & -0.8769 & 0.0836 & 0.2314 & 0.1369 \\ -0.1346 & 0.3234 & 0.4165 & 0.4185 & 0.7271 \\ -0.7795 & 0.3058 & 0.3002 & 0.0080 & -0.4568 \\ 0.3752 & -0.0220 & 0.2340 & 0.7523 & -0.4878 \end{bmatrix}.$$

In a semi-supervised context, we assume the class label for the first observation in each of the four groups is known. As the nesting structure of species and sex is not known a priori, both possibilities are considered in this case. The results are presented in Table 5.5.

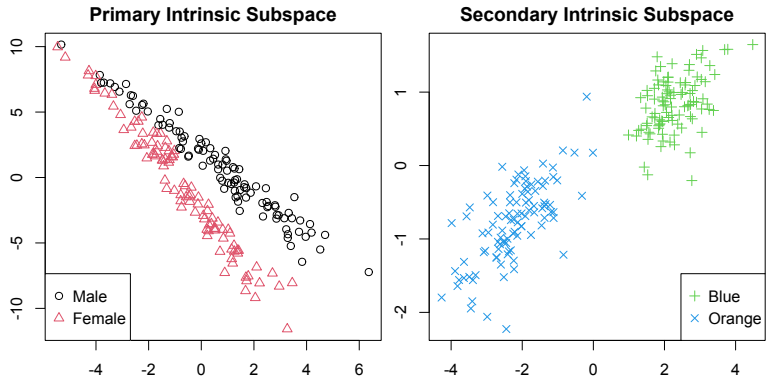

Figure 5.4: A scatterplot of the *Leptograpsus* crabs dataset in the primary intrinsic subspace (left) and secondary intrinsic subspace (right) based on the fitted model in Table 5.4. Points are labelled by the true class labels. In the secondary intrinsic subspace, the points are adjusted by subtracting $\mathbf{B}_h^\top \boldsymbol{x}_n$, where h is the secondary clustering component to which the observation is assigned.

In the semi-supervised situation, we find similar behaviour as in Table 5.4, albeit with slightly better recovery of the sex class label but slightly worse recovery of the species class label. The BIC metric continues to outperform the finite Gaussian mixture model with parsimonious covariance matrices as estimated by the *mclust* package primarily due to the reduction in the number of parameters.

5.5.2 Olive Oil Dataset

The olive dataset from Forina et al. (1983) describes the chemical composition of 572 different olive oils for three areas of Italy with multiple constituent regions. There are eight variables representing different fatty acids found in olive oil. As class labels, we have as the first area of Northern Italy the three regions Umbria, East Liguria, and West Liguria. The second area is Sardinia divided into two regions: Inland Sardinia and Coastal Sardinia. Lastly, the Southern Italy area has four regions: North Apulia, Calabria, South Apulia, and Sicily.

Table 5.5: Fitted model metrics for *Leptograpsus* dataset using the proposed model and finite Gaussian mixtures using *mclust* in a semi-supervised setting. The best model is selected from each family by selecting the best BIC, where lower is better.

	Nested Gaussians	*mclust*
Number of Clusters	$G = 2$, $H = 2$	4
Dimensions	$p_x = 2$, $p_y = 2$	5
Number of Parameters	40	68
Intrinsic Independence	No	–
Conditional Independence	Yes	–
Covariance Type	Full	EEV
BIC	2775.97	2903.31
ARI (overall)	0.8230	0.8032
ARI (sex)		0.3701
Primary	0.7733	
Secondary	-0.0046	
ARI (species)		0.4672
Primary	-0.0046	
Secondary	0.9602	

Table 5.6: Fitted model metrics for the Italian olive oil dataset using the proposed model and finite Gaussian mixtures using *mclust*. The best model is selected from each family by selecting the best BIC, where lower is better.

	Nested Gaussians	*mclust*
Number of Clusters	$G = 4$, $H = 3$	10
Dimensions	$p_x = 6$, $p_y = 2$	8
Number of Parameters	176	197
Intrinsic Independence	No	–
Conditional Independence	Yes	–
Covariance Type	Full	VVE
BIC	41855.23	42146.84
ARI (area)		0.3701
Primary	0.9273	
Secondary	0.5164	
ARI (region)		0.8032
Primary	0.0444	
Secondary	0.0555	

For this dataset, we search over models having $p_x, p_y \leq 7$ with $G, H_g \leq 4$. We skip the Mini-EM process from Section 5.3.1 due to the high computational burden for a dataset of this size and the number of potential model variations. Again, we exclude models with a covariance eigenvalue below 10^{-3}. Using BIC to perform model selection, the best model is presented in Table 5.6 along with the best *mclust* clustering model. We find better BIC using the proposed nested Gaussian model, with good recovery of the primary clustering of area but not the secondary clustering of region. As well, the conditionally dependent model expected given the context of the dataset is not selected by BIC.

We also consider the best BIC among models with G and H_g correctly specified as $G = 3$ and $\langle H_1, H_2, H_3 \rangle = \langle 4, 2, 3 \rangle$, respectively, in Table 5.7. A comparison is *mclust* with correctly specified number of clusters $G = 9$. In this case, the area class labels are

Table 5.7: Fitted model metrics for the Italian olive oil dataset using the proposed model and finite Gaussian mixtures using *mclust*, both with the correct number of clusters specified. Lower BIC is better.

	Nested Gaussians	*mclust*
Number of Clusters	$G = 3$, $\langle H_1, H_2, H_3 \rangle = \langle 4, 2, 3 \rangle$	9
Dimensions	$p_x = 4$, $p_y = 4$	8
Number of Parameters	336	180
Intrinsic Independence	No	–
Conditional Independence	No	–
Covariance Type	Full	VVE
BIC	43184.24	42195.75
ARI (area)		0.3481
Primary	1.0000	
Secondary	0.4776	
ARI (region)		0.6490
Primary	0.3984	
Secondary	0.5065	

well-recovered by the proposed model at the detriment of region class label recovery. It is possible that region behaviours do not fit neatly within the same dimensions for each area. A plot of the primary clustering subspace is provided in Figure 5.5; while it may seem that some observations ought to be misclassified in this view, they are also partially informed by the secondary clustering $g{:}h$ whose subspace is not seen.

Semi-Supervised Setting

In addition to the fully unsupervised approach, we also examine the olive oil dataset in a semi-supervised setting. Specifically, we assume the primary and secondary class labels of the first observation in each region are known. Effectively, this also restricts the model

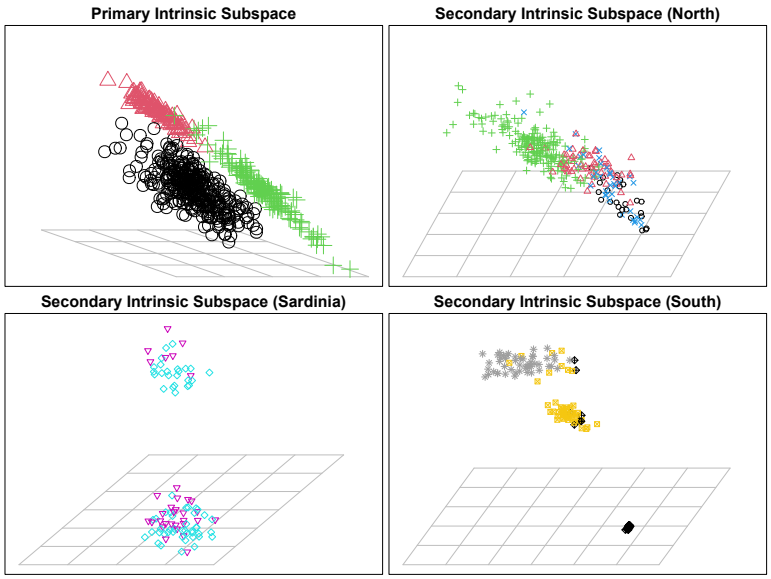

Figure 5.5: A scatterplot of the Italian olive oil dataset in a projection of the primary intrinsic subspace (top-left) and secondary intrinsic subspaces (top-right, bottom) based on the fitted model in Table 5.7 with the correct number of specified clusters. Points are labelled by the true class labels. In the secondary intrinsic subspace, the points are adjusted by subtracting $\mathbf{B}_h \boldsymbol{x}_n$, where h is the secondary cluster to which the observation is assigned.

Table 5.8: Fitted model metrics for the Italian olive oil dataset using the proposed model and finite Gaussian mixtures using *mclust* in a semi-supervised context. The first observation in each of the nine regions has known class labels. Lower BIC is better.

	Nested Gaussians	*mclust*
Number of Clusters	$G = 3$, $\langle H_1, H_2, H_3 \rangle = \langle 4, 2, 3 \rangle$	9
Dimensions	$p_x = 2$, $p_y = 3$	8
Number of Parameters	188	180
Intrinsic Independence	No	–
Conditional Independence	No	–
Covariance Type	–	VVV
BIC	43820.68	42471.20
ARI (area)		0.3416
Primary	1.0000	
Secondary	0.4776	
ARI (region)		0.4581
Primary	0.4089	
Secondary	0.4154	

selection procedure to the true number of primary and secondary clusters, leaving only the choice of p_x, p_y, and model variations to be searched over. The results of the semi-supervised procedure are given in Table 5.8. While the BIC is higher than the semi-supervised *mclust* model, the proposed model is able to fully recover the area class labels. Interestingly, introducing a small number of observed class labels observations causes the BIC-selected model to use much fewer dimensions for the primary and secondary intrinsic subspaces.

Additionally, since there is a primary and secondary hierarchy in the class labels, we also investigate the application when the primary labelling is known but the secondary clustering is completely unknown. For the first observation in each of the three areas, we assume their area membership $z_{n,g}$ to be known. Aside from fixing the primary cluster count $G = 3$, we use the same estimation procedure as previously. This yields the results

Table 5.9: Fitted model metrics for the Italian olive oil dataset using the proposed model and finite Gaussian mixtures using *mclust* in a semi-supervised context with a portion of primary clustering labels known. The first observation in each of the nine regions has known area class labels. Lower BIC is better.

	Nested Gaussians
Number of Clusters	$G = 3$, $H = 2$
Dimensions	$p_x = 3$, $p_y = 5$
Intrinsic Independence	Yes
Conditional Independence	No
Number of Parameters	99
Covariance Type	Diagonal
BIC	42857.46
ARI (area)	
Primary	0.5406
Secondary	0.4288
ARI (region)	
Primary	0.0657
Secondary	0.1528

in Table 5.9. As there is no equivalent concept in *mclust*, we have opted to present the proposed model alone. Here, we see that both the primary and secondary clusterings tend to recover the area variable.

5.5.3 93 cars Dataset

In this dataset, we examine the `Cars93` dataset from the `MASS` R package (Venables and Ripley, 2002). This dataset covers 27 parameters of 93 cars from the year 1993, covering a range of qualitative and quantitative variables. For the present analysis, we take as manifest variables Price, MPG.city, MPG.highway, EngineSize, Horsepower, RPM, Rev.per.mile, Fuel.tank.capacity, Length, Wheelbase, Width, Turn.circle, and Weight. The

numerical variables Min.Price and Max.Price are omitted to reduce redundancy with the Price variable, and the Cylinders variable was treated as categorical. The two variables Rear.seat.room and Luggage.room were dropped due to missingness. As potential class labels, we have Manufacturer, Type, AirBags, DriveTrain, Cylinders, Man.trans.avail, and Origin. For further description of these variables, refer to the dataset description in the `MASS` package or the underlying work (Lock, 1993). Due to the contextual complexity of the data, we allow for more complicated relationships between the fitted class memberships and observed class labels. Specifically, there is a variety of available observed class labels, namely Manufacturer, Model, Type, AirBags, DriveTrain, Cylinders, Man.trans.avail, and Origin. There is no strict hierarchical relationship nor an unambiguous selection of primary/secondary clusterings. As well, we do not expect each categorical level to be a distinct fitted cluster. Thus, we opt to cross-tabulate the fitted class labels against more than one set for inspection as the adjusted Rand index is not very insightful.

As a pre-processing step, we have performed PCA on the 13 variables and taken the first five principal components (PCs). The first three PCs account for 99.92% of total variation; we consider the fourth and fifth PCs we have included under the potential assumption that they are noise. In total these five PCs comprise 99.992% of total variation. For model selection, we search over model specifications having $p_x, p_y \leq 5$ with $G, H_g \leq 4$. We again skip the Mini-EM process for computational reasons due to the expansive model space. After estimation, we continue to exclude fitted models with a covariance eigenvalue below 10^{-3} and select the best model via BIC. As there are no definitive corresponding class labels, we perform a more detailed analysis of the clustering labels within the context of vehicles by presenting a cross-tabulation of clustering labels against selected observed labels. The summary table of the fitted model and a baseline *mclust* model is provided in Table 5.10. The *mclust* comparison spans all 14 covariance types and is also run on the pre-processed data.

The cross-tabulations of the primary clustering labels against a select subset of observed class variables in the dataset is given in Table 5.11. We cross-tabulate the fitted primary clustering classes against the Cylinder, Type, and AirBags variables. n the first primary cluster $g = 1$, we see a mix of 4- and 6-cylinder vehicles and a range of larger vehicle types.

Table 5.10: Fitted model metrics for 93 cars dataset using the proposed model and finite Gaussian mixtures using *mclust* on the first 5 principal components of the data. The best model is selected from each family by selecting the best BIC, where lower is better.

	Nested Gaussians	*mclust*
Number of Clusters	$G = 3$, $H = 2$	2
Dimensions	$p_x = 2$, $p_y = 2$	5
Number of Parameters	43	17
Intrinsic Independence	No	–
Conditional Independence	Yes	–
Covariance Type	Diagonal	VEI
BIC	5776.77	5796.84

In $g = 2$, the dominant vehicle type are small vehicles with almost exclusively 3- and 4-cylinders with front-wheel drive. Finally, $g = 3$ is also comprised of 8-cylinder vehicles sporty or midsize vehicles with mostly driver-only airbags; interestingly, the sole rotary engine car fell into this cluster. Turning our attention to secondary clustering labels in Table 5.12, we have two clusters that are evaluated against the Type and Cylinder variables. We observe that the secondary cluster $h = 2$ captures almost exclusively 6-cylinder vans. A large table representing the joint cross-tabulation against all three classes is given in Appendix C.3.

5.5.4 Handwritten Digits Dataset

In this section, we apply the proposed method to a dataset representing handwritten digits 0 through 9 in various features (van Breukelen et al., 1998; van Breukelen and Duin, 1998; Jain et al., 2000) as retrieved from the UCI datasets repository (Dua and Graff, 2017). Across 2000 observations, the dataset contains 200 of each digit with known class labels. Among the variety of available features, we select the three continuous morphological features as the data of interest.

Table 5.11: Primary clustering labels for 93 cars dataset using the proposed model from Table 5.10. A selected subset of available class information from the dataset is presented here; particularly, the number of cylinders, type of vehicle, and type of airbag configuration.

Cylinder	$g=1$	$g=2$	$g=3$	Type	$g=1$	$g=2$	$g=3$
3-cyl	0	3	0	Compact	11	5	0
4-cyl	27	22	0	Large	11	0	0
5-cyl	1	1	0	Midsize	15	5	2
6-cyl	29	1	1	Small	9	12	0
8-cyl	4	0	3	Sporty	7	4	3
rotary	0	0	1	Van	8	1	0
AirBags	$g=1$	$g=2$	$g=3$				
Driver & Passenger	15	0	1				
Driver only	30	9	4				
None	16	18	0				

Table 5.12: Secondary clustering labels for 93 cars dataset using the proposed model from Table 5.10. A selected subset of available class information from the dataset is presented here.

Type	$h=1$	$h=2$	Cylinder	$h=1$	$h=2$
Compact	16	0	3-cyl	3	0
Large	10	1	4-cyl	48	1
Midsize	22	0	5-cyl	1	1
Small	21	0	6-cyl	23	8
Sporty	11	3	8-cyl	6	1
Van	1	8	rotary	0	1

Table 5.13: Fitted model metrics for the handwritten digits dataset using the proposed model and finite Gaussian mixtures using *mclust*. The best model is selected from each family by selecting the best BIC, where lower is better.

	Nested Gaussians	*mclust*
Number of Clusters	$G = 5$, $H = 3$	7
Dimensions	$p_x = 2$, $p_y = 1$	3
Number of Parameters	45	69
Intrinsic Independence	No	–
Conditional Independence	Yes	–
Covariance Type	–	VVV
BIC	45530.76	44889.69
ARI (digit)		0.3452
Primary	0.3007	
Secondary	0.0629	

We explore the model space for $p_x, p_y \leq 2$ with $G, H_g \leq 5$, and perform a single initialization without Mini-EM. Again, we discard fitted models with a covariance eigenvalue below 10^{-3}. A comparison against *mclust* models up to 15 clusters is made. In both cases, we select the best model by BIC. The summary table of values is given in Table 5.13.

Indeed, while neither the BIC nor ARI outperform the *mclust* model, an interesting phenomenon appears in the actual nested clustering results. In Table 5.14, the cross-tabulations between the true digit class label and the primary/secondary clustering labels is presented. In the primary clustering, the first cluster $g = 1$ captures zeros and eights; two symmetric digits with loops. The second cluster $g = 2$ captures a mix of ones, sixes and nines; the latter two being very similar digits with a single loop. The third cluster $g = 3$ seems to capture mostly fours and sevens, rather angular digits. The fourth cluster $g = 4$ seems to be a small miscellaneous cluster. The fifth cluster $g = 5$ mostly captures the digits two, three, and five; generally curved digits without a loop. We note that both one and seven is seemingly out of place in the primary cluster to which they tend to be

Table 5.14: Clustering labels for the digits dataset for the proposed model in both the primary and secondary clusterings.

Digit	$g=1$	$g=2$	$g=3$	$g=4$	$g=5$	Digit	$h=1$	$h=2$	$h=3$
0	195	2	0	1	2	0	20	152	28
1	13	125	62	0	0	1	10	69	121
2	0	0	23	0	177	2	52	117	31
3	0	1	48	0	151	3	51	115	34
4	0	3	174	4	19	4	55	130	15
5	0	0	8	0	192	5	20	46	134
6	0	142	49	0	9	6	61	104	35
7	0	3	161	1	35	7	170	30	0
8	176	8	12	3	1	8	14	148	38
9	0	146	45	1	8	9	48	118	34

assigned. To resolve this, we turn to the secondary clustering, in which the first cluster $h = 1$ has a much higher proportion of sevens. Similarly, ones are somewhat distinguished by the third secondary cluster $h = 3$. In this manner, we see that the primary clustering tends to separate digits based on whether or not they have loops and their symmetry (e.g., 0 and 8 versus 6 and 9), and the secondary clustering mostly distinguishes the one and seven digits.

For this dataset, the best proposed model fit yielded a rotation matrix estimate

$$\mathbf{\Gamma} = \begin{bmatrix} 0.0063 & -0.9526 & -0.3041 \\ 0.0294 & 0.3041 & -0.9522 \\ 0.9995 & -0.0029 & 0.0300 \end{bmatrix}.$$

Indeed, from this rotation on the data and the observed cross-tabulation table Table 5.13, we can observe the separating out of different groups in the intrinsic subspaces in Figure 5.6. In particular, the scatterplot of the two-dimensional primary intrinsic subspace shows separation of the 0/8, 1/6/9, 4/7, and 2/3/5 clusters and the kernel density estimate

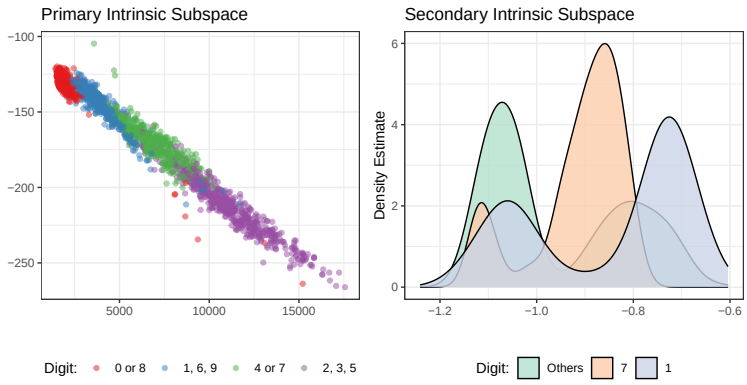

Figure 5.6: A scatterplot and kernel density estimate of the handwritten dataset in the primary intrinsic subspace (left) and secondary intrinsic subspace (right) based on the fitted model in Table 5.13. Points are labelled by the true class labels. In the secondary intrinsic subspace, the points are adjusted by subtracting $\mathbf{B}_h^\top \boldsymbol{x}_n$, where h is the secondary clustering component to which the observation is assigned.

of the one-dimensional secondary intrinsic subspace shows the separation of the 1 and 7 clusters.

5.6 Discussion

The proposed model demonstrates in the *Leptographsus* crabs dataset an ability to isolate relevant subspaces in which clusterings appear. In the Italian olive oil dataset, the primary subspace recovers the area class labels when the correct number of clusters is specified. With comparison to the family of finite Gaussian mixtures fitted by *mclust*, the BIC metric sees some improvement despite using fully-varying covariance matrices for both the primary and secondary clusterings.

As in Galimberti et al. (2018), the proposed method identifies the appropriate subspace variables in which the species variable separates for the *Leptographsus* crabs dataset. In the intrinsic subspace separating the sex class labels, the effect is seen to a lesser extent as there

remains a degree of overlap. A similar occurrence is seen with the Italian olive oil dataset under the correct cluster specification, whereby the relevant subspace for separating the area classes is identified, with moderate separation in the secondary region subspace.

5.6.1 Future Work

Regularization could be applied to the rotation matrix $\mathbf{\Gamma}$ to obtain a permutation matrix. This would produce as a special case the behaviour of variable selection in lieu of estimating intrinsic subspaces.

Moreover, the need to select both the number of primary and secondary cluster greatly enlarges the model space. In the conditionally dependent case, H_g is specified by G different integers; a combinatorial expansion in the number of possibilities. This is compounded by the selection of intrinsic subspace dimensions. Potential remedies include genetic algorithms as in Galimberti et al. (2018) or automatic model selection methods.

Primary and secondary cluster covariance matrices could be specified as parsimonious covariance matrices (Celeux and Govaert, 1995; Browne and McNicholas, 2014; McNicholas and Murphy, 2008) to improve model flexibility and allow for more parsimony. However, this further increases the model space and compounds upon the model selection challenges described above.

Finally, we can consider the cases beyond two stages of clustering, whereby the data exhibits tertiary clusters $g{:}h{:}k$ for secondary cluster $g{:}h$. The intrinsic subspaces may also have different dimensionalities p_y for secondary clusters, leading to additional parsimony and allowing clusters to separate into their informative subspaces at different depths.

Chapter 6

Extrapolating Conditional Expectations to Accelerate EM Procedures

6.1 Introduction

The Expectation-Maximization (EM) procedure (Dempster et al., 1977) is a popular method for maximizing intractable objective functions such as in maximum likelihood estimation. However, the algorithm suffers from slow convergence in many cases; particularly, it often experiences a linear rate of convergence in the vicinity of a local optima. While switching to second-order methods that make use of the local curvature information can be much faster, determining an expression for or numerically computing the Hessian may be difficult. As a result, methods for accelerating the EM procedure have been investigated (Meng and Rubin, 1993; Liu and Rubin, 1994; Liu et al., 1998; Varadhan and Roland, 2008; He and Liu, 2012) to help amend some of the drawbacks with EM itself.

The EM procedure, so named for its iterative and alternating use of an expectation step (E-step) and maximization step (M-step), is often understood from one of two perspectives

(Berlinet and Roland, 2007). The first treats the procedure as a sequence of fixed-point iterations. Approaches to accelerating the EM procedure under this paradigm often acts within the parameter space of the problem, whereby the parameters $\boldsymbol{\theta}$ are extrapolated as a vector-valued sequence in a Euclidean vector-space $\mathbb{R}^p$. The second emphasizes the alternating nature of E-steps and M-steps, and understand the output of the E-step as a surrogate function to be maximized in the M-step. This second mode makes stronger use of the missing data's distribution, explicitly identifying and computing the required expected values within each EM iteration.

The EM algorithm has enjoyed broad application in a variety of subjects and has itself been the subject of much study. We provide here a brief technical recap of the framework of EM with associated discussion and leave other details to the original work of Dempster et al. (1977) and texts such as McLachlan and Krishnan (2008). For problems with missing data, or problems where missing data could be intentionally introduced, the direct maximization of the log-likelihood function can be intractable. In some of these cases, we may turn to the EM algorithm to decompose the central problem into a sequence of simpler, tractable problems. In particular, we define the complete-data log-likelihood $\ell_{\text{com}}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})$ to be a function of both the observed data $\mathbf{X}$ and latent data $\mathbf{Z}$. Since the latent data is by definition unobservable, we use the conditional distribution of $\mathbf{Z}$ given observable $\mathbf{X}$ with parameters $\boldsymbol{\theta}^{(t)}$ as of iteration t obtain the expected complete-data log-likelihood $\mathrm{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) = \mathrm{E}_{\mathbf{Z}}^{(t)} \left[\ell_{\text{com}}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})\right]$.

Many extensions to the EM algorithm have been implemented to help improve upon its convergence speed; a selection of examples are given here. In the Expectation-Conditional-Maximization (ECM) algorithm (Meng and Rubin, 1993), the M-step is decomposed into multiple conditional maximization steps (CM-steps) whereby the parameter set is partitioned, with each subset maximized conditional on the previous subsets until all parameters are updated, at which point the procedure returns to the E-step. In the Expectation-Conditional-Maximization-Either (ECME) (Liu and Rubin, 1994) extension to ECM, a CM-step may also optimize the observed log-likelihood directly instead of the surrogate. In SQUAREM (Varadhan and Roland, 2008), vector extrapolation methods are used every few EM iterations to skip ahead in the parameter space; this method can be used to

accelerate EM, ECM, and ECME as it considers a single parameter update as including both the E-step and the M-step. In Parameter-Expanded EM (PX-EM) Liu et al. (1998) augment the complete-data parameter space to achieve faster convergence. In DECME, He and Liu (2012) extend the parameter update to searching over the subspaces spanned by the past one (DECME-1) and two (DECME-2) parameter update differences. Other acceleration schema include Polyak's heavy-ball method (Polyak, 1964) and Nesterov's accelerated gradient descent (Nesterov, 1983).

We propose an acceleration scheme for the EM algorithm that acts upon the sequence of surrogate functions as the object of extrapolation from iteration to iteration. Suppose the problem at hand is the maximization of a log-likelihood function $\ell(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})$ with observed data $\mathbf{X}$ and latent/missing data $\mathbf{Z}$. In standard EM at iteration t, the M-step acts upon the E-step surrogate $\mathrm{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$. In the proposed method, the M-step acts upon the α-accelerated surrogate $(1-\alpha)\,\mathrm{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) - \alpha\,\mathrm{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)})$. For many problems, this has the benefit of requiring minimal changes to the M-step optimization and is independent of the parameterization of the problem. In the present work, we define the acceleration schema with a general optimization in the acceleration factor α, with a practical choice of taking α to be the Aitken's acceleration factor on the observed log-likelihood. In effect, this method performs leapfrogging in the surrogate function for faster parameter estimation as evaluated by wall-clock elapsed time.

6.2 Methodology

Suppose we intend to perform maximum likelihood estimation where direct maximization is intractable but an EM procedure is feasible. Let $\mathbf{X}$ denote the observed data, $\mathbf{Z}$ the latent data, and $\boldsymbol{\theta}$ the model parameters to be estimated. We denote the observed log-likelihood by $\ell_{\mathrm{obs}}(\boldsymbol{\theta}; \mathbf{X})$ and the complete-data log-likelihood by $\ell_{\mathrm{com}}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})$. Let a superscript of (t) denote the value of the variable at iteration t of the estimation procedure, and let $\mathrm{E}^{(t)}$ be the expectation under model parameters $\boldsymbol{\theta}^{(t)}$.

As part of the E-step of the EM procedure at iteration t, we take the expectation of the

complete-data log-likelihood to yield the surrogate function $\mathrm{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$ which is fed-forward into the M-step. This surrogate invariably contains expectations of functions of the latent data $\mathbf{Z}$, which are in turn distributions parameterized by $\boldsymbol{\theta}^{(t)}$. We insert an accelerated LM-step here maximizing an alternative function with an acceleration factor α acting in the span of the current and one-iteration-before surrogate functions. In other words, define

$$\mathrm{R}(\boldsymbol{\theta}, \alpha; \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)}) = (1+\alpha)\,\mathrm{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) + (-\alpha)\,\mathrm{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)}) \tag{6.1}$$

for $\alpha \geq 0$. Equivalently, we may re-write this in the form

$$\mathrm{R}(\boldsymbol{\theta}, \alpha; \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)}) = \mathrm{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) + \alpha[\mathrm{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) - \mathrm{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)})];$$

i.e., we perform a line-search in the function space spanned by $\mathrm{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) - \mathrm{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t-1)})$. We propose a leapfrog maximization step (LM-step) based on this accelerated-surrogate R that replaces the M-step. In full generality, the LM-step solves the maximization

$$\underset{\alpha \in \mathbb{R}^{\geq 0}, \boldsymbol{\theta} \in \boldsymbol{\Theta}}{\arg\max}\ \mathrm{R}(\boldsymbol{\theta}, \alpha; \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)}) \tag{6.2}$$

for both the acceleration factor α and the parameter update $\boldsymbol{\theta}^{(t+1)}$ simultaneously. As a special case, when $\alpha = 0$, R reduces to the surrogate Q, and so the LM-step becomes an M-step. As this joint maximizer is invariably difficult, we define

$$\boldsymbol{\theta}^{(t+1)}(\alpha) = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\arg\max}\, \mathrm{R}(\boldsymbol{\theta}, \alpha; \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)})$$

as the conditional maximizer in $\boldsymbol{\theta}$ for a given α at iteration t. In many practical applications, the expected complete-data log-likelihood can be re-written in the form

$$\mathrm{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) = \sum_i \mathrm{E}^{(t)}[f_i(\mathbf{Z}, \mathbf{X}) \mid \mathbf{X}] g_i(\mathbf{X}; \boldsymbol{\theta})$$

for some set of functions f and g, and $\mathrm{E}^{(t)}$ being the expectation with parameters $\boldsymbol{\theta}^{(t)}$. Thus, accelerated surrogate function can be recast into leapfrogging in the expectations of latent data $\mathbf{Z}$

$$\mathrm{R}(\boldsymbol{\theta}, \alpha; \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)}) = \sum_i \{(1+\alpha)\,\mathrm{E}^{(t)}[f_i(\mathbf{Z}, \mathbf{X}) \mid \mathbf{X}] - \alpha\,\mathrm{E}^{(t-1)}[f_i(\mathbf{Z}, \mathbf{X}) \mid \mathbf{X}]\} g_i(\mathbf{X}; \boldsymbol{\theta}).$$

Examples of such are provided in Section 6.3. In effect, the LM-step acts upon the extrapolated expected complete-data likelihood Equation (6.1), which leapfrogs over the need to wait for multiple E- and M-steps. A stylized representation is given in Figure 6.1, with the general form of the proposed procedure is given in Algorithm 1.

An important observation is that the leapfrog-accelerated parameters $\boldsymbol{\theta}^{(t+1)}$ from (6.2) may result in a decrease in the observed log-likelihood, whereupon we perform a standard M-step with $\mathrm{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$. An additional remark is that the maximizer $\boldsymbol{\theta}$ of R may be undefined for certain values of α, though for fixed $\alpha = 0$ the LM-step coincides with the M-step, where it invariably exists. Indeed, for cases such as finite mixture models, an excessively large value of α can yield accelerated maximizers for the covariance matrices that are no longer positive-definite by rendering all of the corresponding cluster probabilities negative.

Algorithm 1 Leapfrog-Expectation Acceleration Procedure

initialize $\boldsymbol{\theta}^{(0)}$, $\boldsymbol{\theta}^{(1)}$, $t \leftarrow 2$
while $t \leq t_{\max}$ **do**
  perform E-step to determine $Q^{(t)}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$
  perform LM-step to determine

$$(\alpha^{(t+1)}, \boldsymbol{\theta}^{(t+1)}) \leftarrow \underset{\alpha \in \mathbb{R}^{\geq 0}, \boldsymbol{\theta} \in \boldsymbol{\Theta}}{\arg\max}\ \mathrm{R}^{(t)}(\boldsymbol{\theta}, \alpha; \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)})$$

  if $\ell_{\text{obs}}(\boldsymbol{\theta}^{(t+1)}; \mathbf{X}) \leq \ell_{\text{obs}}(\boldsymbol{\theta}^{(t)}; \mathbf{X})$ or $\boldsymbol{\theta}^{(t+1)}$ invalid **then**
    perform M-step to determine

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\arg\max}\, Q^{(t)}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$$

  end if
  $t \leftarrow t + 1$
  if convergence **then**
    break
  end if
end while

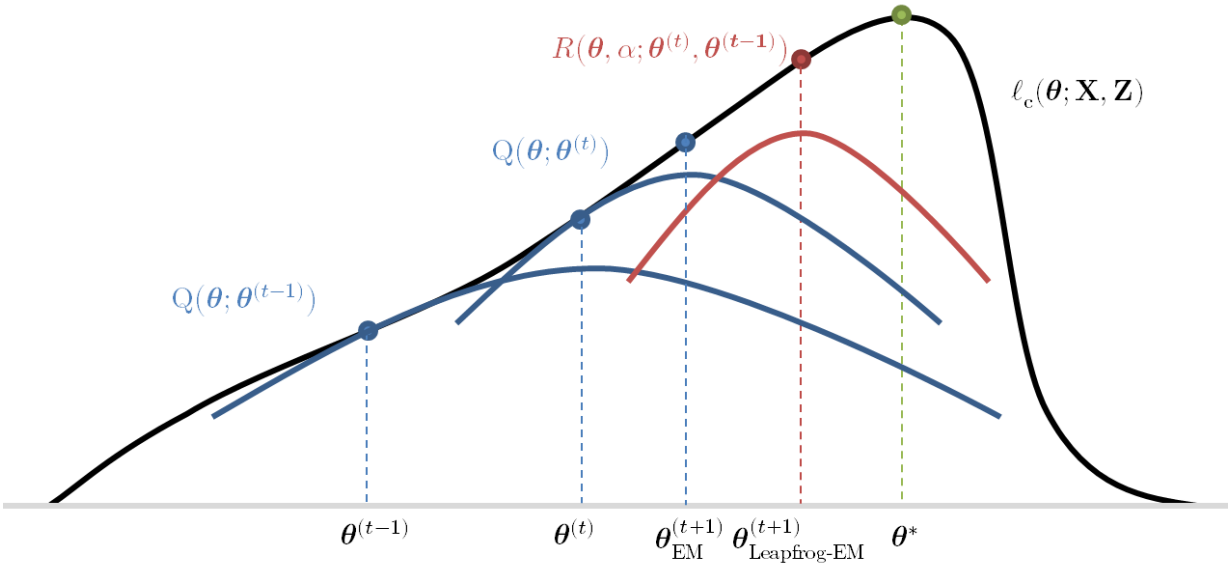

Figure 6.1: A stylized representation of the leapfrog-expectation acceleration procedure. The two (blue) surrogate functions Q based on parameters $\boldsymbol{\theta}^{(t-1)}$ and $\boldsymbol{\theta}^{(t)}$ respectively are accelerated to form the (red) accelerated surrogate function R, whose parameter update $\boldsymbol{\theta}^{(t+1)}_{\text{Leapfrog-EM}}$ is accelerated compared to $\boldsymbol{\theta}^{(t+1)}_{\text{EM}}$.

6.2.1 Aitken's acceleration guided backtracking line search

The argmax problem (6.2) in both α and $\boldsymbol{\theta}$ simultaneously is difficult in both theory and practice. In the former case, any potential closed-form maximizers in $\boldsymbol{\theta}$ alone as used in the regular M-step are often unusable. In the latter case, numerical optimization requires evaluating the objective function, its gradient, and potentially its Hessian repeatedly which is also computationally expensive. We propose in this section a heuristic for performing conditional majorization in LM-step using Aitken's acceleration factor on the observed log-likelihoods $\ell_{\text{obs}}(\boldsymbol{\theta}; \mathbf{X})$ as a stand-in for α.

Specifically, at iteration $t \geq 3$ define the Aitken's acceleration factor as in McNicholas and Murphy (2008) and Böhning et al. (1994) by

$$a_{\text{aitken}} = \frac{\ell_{\text{obs}}(\boldsymbol{\theta}^{(t)}; \mathbf{X}) - \ell_{\text{obs}}(\boldsymbol{\theta}^{(t-1)}; \mathbf{X})}{\ell_{\text{obs}}(\boldsymbol{\theta}^{(t-1)}; \mathbf{X}) - \ell_{\text{obs}}(\boldsymbol{\theta}^{(t-2)}; \mathbf{X})}$$

and define the putative leapfrog acceleration factor $\alpha_{\text{aitken}} = 1/(1 - a_{\text{aitken}})$. As α_{aitken} may be negative at certain iterations, we start the line search with the previous starting $\alpha_{\text{aitken}}^{(t-1)}$.

We perform a line-search with backtracking in α guided by the quantity α_{aitken} in the span of the current and one-iteration-behind surrogate functions. We impose an upper-bound $\alpha_{\max}$ on α, the backtracking fraction $b \in (0, 1)$, and the number of backtracking attempts $n_b \geq 0$. We initialize the line-search by fixing $\alpha = \min\{\alpha_{\text{aitken}}, \alpha_{\max}\}$ so the maximizer $\boldsymbol{\theta}^{(t+1)}(\alpha)$ of (6.2) can be found in isolation. The observed log-likelihood $\ell_{\text{obs}}(\boldsymbol{\theta}^{(t+1)}(\alpha); \mathbf{X})$ is checked against the current $\ell_{\text{obs}}(\boldsymbol{\theta}^{(t)}; \mathbf{X})$ to verify an improvement. If ℓ_{obs} has decreased, shrink α by a factor of b and retry again up to n_b times, at which point we fall-back to performing a regular M-step. By design, the reversion of invalid or bad updates and falling back to EM/ECME allows the proposed procedure to inherit the monotonic increase property of the underlying EM/ECME procedure (Dempster et al., 1977). This particular implementation of the leapfrog procedure is given in full detail in Algorithm 2.

The α search parameters can be tuned to improve the efficiency and reliability of the procedure. The $\alpha_{\max}$ upper-bound limits the maximum amount of extrapolation that can

Algorithm 2 Leapfrog-Expectation Acceleration Procedure with Aitken's acceleration guided backtracking line search

initialize $\boldsymbol{\theta}^{(0)}$, $\boldsymbol{\theta}^{(1)}$, $t \leftarrow 2$
specify $\alpha_{\max} > 0$, $b \in (0, 1)$, $n_b \geq 0$
while $t \leq t_{\max}$ **do**
    perform E-step to determine $Q^{(t)}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$
    compute $\alpha = \min\{\alpha^{(t)}_{\text{aitken}}, \alpha_{\max}\}$
    for $\alpha^{(t)} = \alpha, b\alpha, b^2\alpha, ..., b^{n_b}\alpha, 0$ **do**
        perform LM-step to determine

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \arg\max_{\boldsymbol{\theta}} \mathrm{R}^{(t)}(\boldsymbol{\theta}, \alpha^{(t)}; \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)})$$

        if $\ell_{\text{obs}}(\boldsymbol{\theta}^{(t+1)}; \mathbf{X}) \geq \ell_{\text{obs}}(\boldsymbol{\theta}^{(t)}; \mathbf{X})$ and $\boldsymbol{\theta}^{(t+1)}$ valid **then**
            break
        end if
    end for
    $t \leftarrow t + 1$
    if convergence **then**
        break
    end if
end while

occur, which can otherwise be so large as to dramatically overshoot the valid parameter region. The backtracking fraction b and the retry count n_b can help mitigate this by increasing the search granularity at the cost of wall-clock time cost of evaluating $\ell_{\text{obs}}(\boldsymbol{\theta}; \mathbf{X})$ and the maximizer $\boldsymbol{\theta}^{(t+1)}$ more often per iteration. When the compute cost of these are relatively smaller than the compute cost of the E-step, then the line-search can be made more granular. We may also warm-up the process by performing the underlying the EM (or ECME) procedure for a number of iterations; i.e., by constraining $\alpha = 0$ during this warm-up phase. The examples in Section 6.3 use $\alpha_{\max} = 100$ and $n_b = 0$ with 100 iterations of warm-up throughout.

6.2.2 Theoretical Results

In this section, we show under some assumptions about the log-likelihood and its surrogate function that the LM-step update yields a leapfrog-acceleration factor $\alpha > 0$; that is, the leapfrog-expectation acceleration yields an improvement over EM at the same iteration. Particularly, we show this near the vicinity of the MLE where the EM procedure often underperforms second-order methods such as Newton-Raphson and slows down to it's oft-quoted linear convergence rate.

Lemma 3 *Let* $\boldsymbol{\theta}^{(t+1)}(\alpha) = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathrm{R}(\boldsymbol{\theta}, \alpha; \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)})$ *be the maximizer of* R *for the given* $\alpha \geq 0$*, and assume mild regularity conditions on* Q*. Then, at an LM-step, the derivative of* ℓ_{obs} *with respect to the leapfrog-acceleration factor* α *evaluated at* $\alpha = 0$ *is given by*

$$\left[\frac{\mathrm{d}\ell_{obs}(\boldsymbol{\theta}^{(t+1)}(\alpha))}{\mathrm{d}\alpha}\right]_{\alpha=0} = \mathrm{S}(\boldsymbol{\theta}_{EM}^{(t+1)})^\top \, \mathrm{H}_{\mathrm{Q}}^{(t)}(\boldsymbol{\theta}_{EM}^{(t+1)})^{-1} \, \mathrm{S}_{\mathrm{Q}}^{(t-1)}(\boldsymbol{\theta}_{EM}^{(t+1)}),$$

where S *and* H *are the score function and Hessian matrix corresponding to the observed log-likelihood* ℓ*, respectively, and* S_{Q} *and* H_{Q} *those corresponding to the surrogate function* Q*, and* $\boldsymbol{\theta}_{EM}^{(t+1)} = \boldsymbol{\theta}^{(t+1)}(0)$ *is the unaccelerated EM parameter update.*

Proof 4 *In this proof, we follow the general framework of Samuel and Tappen (2009) in differentiating through an argmax function. To reduce notational burden, define* $\mathrm{Q}^{(t)}(\boldsymbol{\theta}) =$

$\mathrm{Q}(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})$ *and* $\mathrm{R}^{(t)}(\boldsymbol{\theta},\alpha) = \mathrm{R}(\boldsymbol{\theta},\alpha;\boldsymbol{\theta}^{(t)},\boldsymbol{\theta}^{(t-1)})$. *For* Q *with sufficient regularity, then we have that* $\left[\frac{\partial \mathrm{R}^{(t)}(\boldsymbol{\theta},\alpha)}{\partial \boldsymbol{\theta}}\right]_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(t+1)}(\alpha)} = 0$ *for all* $\alpha \geq 0$ *by definition of* $m^{(t)}$ *as the maximizer of* R. *Differentiating both sides by* α *yields*

$$\left[\frac{\partial^2 \mathrm{R}^{(t)}(\boldsymbol{\theta},\alpha)}{\partial \boldsymbol{\theta}\, \partial\alpha}\right]_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(t+1)}(\alpha)} + \left[\frac{\partial^2 \mathrm{R}^{(t)}(\boldsymbol{\theta},\alpha)}{\partial \boldsymbol{\theta}\, \partial\boldsymbol{\theta}^\top}\right]_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(t+1)}(\alpha)} \frac{\partial \boldsymbol{\theta}^{(t+1)}(\alpha)}{\partial\alpha} = 0$$

whereupon substituting in the corresponding derivatives, we get

$$\left[\mathrm{S}_{\mathrm{Q}}^{(t)}(\boldsymbol{\theta}^{(t+1)}(\alpha)) - \mathrm{S}_{\mathrm{Q}}^{(t-1)}(\boldsymbol{\theta}^{(t+1)}(\alpha))\right] + \left[(1+\alpha)\,\mathrm{H}_{\mathrm{Q}}^{(t)}(\boldsymbol{\theta}^{(t+1)}(\alpha)) - \alpha\,\mathrm{H}_{\mathrm{Q}}^{(t-1)}(\boldsymbol{\theta}^{(t+1)}(\alpha))\right] \frac{\mathrm{d}\boldsymbol{\theta}^{(t+1)}(\alpha)}{\mathrm{d}\alpha} = 0.$$

Re-arranging, we obtain

$$\frac{\mathrm{d}\boldsymbol{\theta}^{(t+1)}(\alpha)}{\mathrm{d}\alpha} = \left[(1+\alpha)\,\mathrm{H}_{\mathrm{Q}}^{(t)}(\boldsymbol{\theta}^{(t+1)}(\alpha)) - \alpha\,\mathrm{H}_{\mathrm{Q}}^{(t-1)}(\boldsymbol{\theta}^{(t+1)}(\alpha))\right]^{-1} \left[\mathrm{S}_{\mathrm{Q}}^{(t-1)}(\boldsymbol{\theta}^{(t+1)}(\alpha)) - \mathrm{S}_{\mathrm{Q}}^{(t)}(\boldsymbol{\theta}^{(t+1)}(\alpha))\right].$$

This expression is found in the derivative of the observed log-likelihood ℓ_{obs} *with respect to leapfrog-acceleration factor. Specifically,*

$$\frac{\mathrm{d}\ell_{obs}(\boldsymbol{\theta}^{(t+1)}(\alpha))}{\mathrm{d}\alpha} = \mathrm{S}(\boldsymbol{\theta}^{(t+1)}(\alpha))^\top \frac{\partial \boldsymbol{\theta}^{(t+1)}(\alpha)}{\partial\alpha}.$$

We note that at $\alpha = 0$, $\mathrm{R}^{(t)}$ *reduces to the standard EM surrogate* $\mathrm{Q}^{(t)}$ *whose maximizer we denote by* $\boldsymbol{\theta}_{EM}^{(t+1)}$ *as it would be the parameter update for the next iteration of EM. By the regularity of the surrogate* Q*, the gradient at it's maximum is zero; that is,* $\mathrm{S}_{\mathrm{Q}}^{(t)}(\boldsymbol{\theta}_{EM}^{(t+1)}) = 0$. *Thus, we have that*

$$\left[\frac{\mathrm{d}\ell_{obs}(\boldsymbol{\theta}^{(t+1)}(\alpha))}{\mathrm{d}\alpha}\right]_{\alpha=0} = \mathrm{S}(\boldsymbol{\theta}_{EM}^{(t+1)})^\top\, \mathrm{H}_{\mathrm{Q}}^{(t)}(\boldsymbol{\theta}_{EM}^{(t+1)})^{-1}\, \mathrm{S}_{\mathrm{Q}}^{(t-1)}(\boldsymbol{\theta}_{EM}^{(t+1)}).$$

Theorem 2 *For parameters* $\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)}$ *obtained in a small neighbourhood around the maximum likelihood estimate, and whose observed log-likelihood* ℓ_{obs} *and surrogate functions* Q *can be approximated by a quadratic form characterised by positive-definite matrices* $\mathbf{I}_{obs}$ *and* $\mathbf{I}_{com}$*, then the optimum leapfrog-acceleration factor* α *is positive.*

Proof 5 *From Lemma 3, we use the expression*

$$\left[\frac{\mathrm{d}\ell_{obs}(\boldsymbol{\theta}^{(t+1)}(\alpha))}{\mathrm{d}\alpha}\right]_{\alpha=0} = \mathrm{S}(\boldsymbol{\theta}_{EM}^{(t+1)})^{\top}\,\mathrm{H}_{\mathrm{Q}}^{(t)}(\boldsymbol{\theta}_{EM}^{(t+1)})^{-1}\,\mathrm{S}_{\mathrm{Q}}^{(t-1)}(\boldsymbol{\theta}_{EM}^{(t+1)}).$$

and the assumption of $\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)}$ *being near the maximum likelihood estimate* $\boldsymbol{\theta}^{*}$ *to apply the approximations*

$$\ell_{obs}(\boldsymbol{\theta}; \mathbf{X}) = -\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^{*})^{\top}\mathbf{I}_{obs}(\boldsymbol{\theta} - \boldsymbol{\theta}^{*}) \text{ and}$$
$$\mathrm{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) = -\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^{*})^{\top}\mathbf{I}_{com}(\boldsymbol{\theta} - \boldsymbol{\theta}^{*}).$$

We substitute in the associated derivatives into the expression

$$\begin{aligned}\left[\frac{\mathrm{d}\ell_{obs}(\boldsymbol{\theta}^{(t+1)}(\alpha))}{\mathrm{d}\alpha}\right]_{\alpha=0} &= (\boldsymbol{\theta}_{EM}^{(t+1)} - \boldsymbol{\theta}^{*})^{\top}\mathbf{I}_{obs}\mathbf{I}_{com}^{-1}\mathbf{I}_{com}(\boldsymbol{\theta}_{EM}^{(t+1)} - \boldsymbol{\theta}^{*}) \\ &= (\boldsymbol{\theta}_{EM}^{(t+1)} - \boldsymbol{\theta}^{*})^{\top}\mathbf{I}_{obs}(\boldsymbol{\theta}_{EM}^{(t+1)} - \boldsymbol{\theta}^{*})\end{aligned}$$

whereupon the assumption of $\mathbf{I}_{obs}$ *and* $\mathbf{I}_{com}$ *being positive-definite implies the above is a positive quantity. Hence, increasing* α *above zero and performing an LM-step yields a better observed log-likelihood than* $\alpha = 0$*, which corresponds to an M-step.*

6.3 Examples and Simulations

In this section, we apply the proposed method to three families of models often estimated by EM-type procedures: the variance components in linear mixed-effects model (Laird and Ware, 1982), the factor analysis model (Rubin and Thayer, 1982; Jöreskog, 1967), and the finite Gaussian mixture model (Banfield and Raftery, 1993). Throughout this section, we reference the parameterization and EM/ECME steps given in McLachlan and Krishnan (2008). When ECME procedures are available, we include them and their leapfrog-acceleration as well. Across all procedures, we assess convergence for the unaccelerated and leapfrog-accelerated procedures by checking the difference in observed log-likelihood $\ell_{\mathrm{obs}}(\boldsymbol{\theta}; \mathbf{X})$ from iteration-to-iteration, and stopping if $\ell_{\mathrm{obs}}(\boldsymbol{\theta}^{(t)}; \mathbf{X}) - \ell_{\mathrm{obs}}(\boldsymbol{\theta}^{(t-1)}; \mathbf{X}) \leq 10^{-10}$.

Throughout, we compare the leapfrog-accelerated procedures against the corresponding SQUAREM acceleration procedure (Varadhan and Roland, 2008), an acceleration scheme that admits a general fixed-point optimization procedure such as EM or ECME. We distinguish the leapfrog-accelerated and SQUAREM-accelerated methods from unaccelerated procedures by the prefixes Leapfrog- and SQUAREM-, respectively. As the SQUAREM acceleration acts on the parameters $\boldsymbol{\theta}$, which is often subject to constraints, we benchmark using the usual constrained parameterization flattened into a vector $\boldsymbol{\theta}_{\text{cons}}$ as well as using a transformation into an unconstrained version $\boldsymbol{\theta}_{\text{unc}}$. For the latter parameterization, we denote procedure with a suffix -T. For SQUAREM-accelerated procedures, we use the default convergence criteria with tolerance 10^{-7} and require monotone convergence (Du and Varadhan, 2020). We assess the resulting parameter estimates by evaluating the observed log-likelihood at the final iteration, determined by either the aftorementioned convergence criteria or reaching the maximum alloted number of iterations $t_{\max} = 10000$. We avoid comparing parameter estimates due to identifiability issues in the examples, such as cluster index permutation in the finite Gaussian mixture model and rotational invariance in the factor analysis model.

As we generate multiple datasets from the assumed model in the simulation study, each dataset may have a different maximum observed log-likelihood. Thus, we compare the deviations $\Delta_{\max}$ from the best observed log-likelihood across all procedures within each dataset, and classify procedures into four different possible groups. We consider procedures which have converged with $\Delta_{\max} \in [0, 10^{-8}), [10^{-8}, 10^{-4}), [10^{-4}, \infty)$ to be near-optimal, sub-optimal, and very sub-optimal, respectively. We also datasets where all tested procedures converge to highlight the relative performance when procedures produce significantly differing results. Finally, when a procedure does not converge within the alloted iterations, we consider it as a distinct class regardless of its $\Delta_{\max}$.

6.3.1 Variance Components Model

The variance components model is a linear mixed effects model useful for analyzing repeated measurements across multiple groups. The observed data are $\boldsymbol{y}_j \in \mathbb{R}^{n_j}$ for observational

unit $j = 1, 2, ..., m$ with the assumption that

$$\boldsymbol{y}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\boldsymbol{b}_j + \boldsymbol{e}_j$$

for fixed effect covariate matrix $\mathbf{X}_j \in \mathbb{R}^{n_j \times p}$, random effect covariate matrix $\mathbf{Z}_j \in \mathbb{R}^{n_j \times q}$, fixed effects $\boldsymbol{\beta} \in \mathbb{R}^p$, latent random effects $\boldsymbol{b}_j \sim \mathrm{N}(\mathbf{0}, \mathbf{D})$, and noise component $\boldsymbol{e}_j \overset{\text{i.i.d}}{\sim} \mathrm{N}(\mathbf{0}, \sigma^2\mathbf{R}_j)$. Here, $\mathbf{R}_j$ are known constants and $\boldsymbol{\beta}$, $\mathbf{D}$, and σ^2 are parameters to be estimated. The observed log-likelihood for the parameters is given by

$$\ell_{\text{obs}}(\boldsymbol{\theta}, \mathbf{Y}) = \sum_{j=1}^{m} \log \phi(\boldsymbol{y}_j; \mathbf{X}_j\boldsymbol{\beta}, \mathbf{Z}_j\mathbf{D}\mathbf{Z}_j^\top + \sigma^2\mathbf{R}_j),$$

where ϕ is the multivariate normal density function. The complete-data log-likelihood for observations $(\boldsymbol{y}_j, \boldsymbol{b}_j)$ is then

$$\ell_{\text{com}}(\boldsymbol{\theta}; \mathbf{Y}, \mathbf{B}) = \sum_{j=1}^{m} \log \phi \left(\begin{bmatrix} \boldsymbol{y}_j \\ \boldsymbol{b}_j \end{bmatrix}; \begin{bmatrix} \mathbf{X}_j\boldsymbol{\beta} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{Z}_j\mathbf{D}\mathbf{Z}_j^\top + \sigma^2\mathbf{R}_j & \mathbf{Z}_j\mathbf{D} \\ \mathbf{D}\mathbf{Z}_j^\top & \mathbf{D} \end{bmatrix} \right).$$

Taking expectations, we obtain the surrogate function

$$\begin{aligned} \mathrm{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) = -\frac{1}{2} \sum_{j=1}^{m} \sum_{i=1}^{n_j} \Big[& (p+q)\log 2\pi + \log\det \sigma^2\mathbf{R}_j + \log\det \mathbf{D} \\ & + (\boldsymbol{y}_j - \mathbf{X}_j\boldsymbol{\beta})^\top \mathbf{D}^{-1} (\boldsymbol{y}_j - \mathbf{X}_j\boldsymbol{\beta}) \\ & + 2(\boldsymbol{y}_j - \mathbf{X}_j\boldsymbol{\beta})^\top \mathbf{D}^{-1} \mathbf{Z}_j \, \mathrm{E}^{(t)}[\boldsymbol{b}_j] \\ & + \mathrm{Tr}(\mathrm{E}^{(t)}[\boldsymbol{b}_j\boldsymbol{b}_j^\top](\mathbf{Z}_j^\top\mathbf{D}^{-1}\mathbf{Z}_j + \sigma^{-2}\mathbf{R}_j^{-1})) \Big], \end{aligned}$$

where $\mathrm{Tr}(\cdot)$ is the matrix trace operator. By linearity of $\mathrm{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$ in the expectations of $\boldsymbol{b}_j$ and $\boldsymbol{b}_j\boldsymbol{b}_j^\top$, the leapfrog-accelerated surrogate $\mathrm{R}(\boldsymbol{\theta}, \alpha; \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)})$ is equivalent to extrapolating in these expected values.

Simulation Study

In this section, we investigate the estimation procedure on the variance components model with a full-factorial design over the number of observational units $m = 2, 3, 4$, number of

measured responses $n_j \equiv 100$, number of fixed effects $p = 2$, number of random effects $q = 2$, and noise magnitude $\sigma^2 = 0.5, 1, 2$. We sample the fixed effect $\mathbf{X}_j$ and random effect $\mathbf{Z}_j$ matrix elements from i.i.d. Unif$(-1, 1)$ distributions, $\mathbf{D} \sim \text{Wishart}(\mathbf{I}, q)$, and $\boldsymbol{\beta} \sim \text{N}(\mathbf{0}, \mathbf{I})$. For simplicity, we assume $\mathbf{R}_j = \mathbf{I}$ throughout and initialize the estimation procedure with parameters ${\sigma^2}^{(0)} = 1$, $\mathbf{D}^{(0)} = \mathbf{I}$, and $\boldsymbol{\beta}^{(0)} = \mathbf{0}$. We perform 100 replications at each parameter combination and estimate parameters using the EM procedure and two ECME procedures, henceforth ECME1 and ECME2 named as per Section 5.9.4 of McLachlan and Krishnan (2008), along with their leapfrog-accelerated and SQUAREM-accelerated procedures. For CM-step 3 in ECME2, we perform Newton-Raphson with numerical derivatives on the observed log-likelihood to obtain the parameter update for σ^2. As the parameters σ^2 and $\mathbf{D}$ must be positive(-definite), we consider the vectorized parameter representations $\boldsymbol{\theta}_{\text{cons}} = \langle \boldsymbol{\beta}, \operatorname{diag} \mathbf{D}, \sigma^2 \rangle$ and $\boldsymbol{\theta}_{\text{unc}} = \langle \boldsymbol{\beta}, \log \operatorname{diag} \mathbf{D}, \log \sigma^2 \rangle$, where $\operatorname{diag}(\cdot)$ extracts the main diagonal and $\log(\cdot)$ acts element-wise, as their constrained and unconstrained forms for SQUAREM.

In Table 6.1, we observe the degree of convergence to the maximum observed log-likelihood for each of the twelve procedures. For this data model, we see that the leapfrog-accelerated EM/ECME1/ECME2 procedures out-perform both the unaccelerated procedures and SQUAREM-accelerated procedures in both frequency of convergence within $t_{\max}$ iterations and arriving in the neighbourhood of the best maximum log-likelihood. Indeed, Figure 6.2 shows the rates of non-convergence being consistently lower across all leapfrog-accelerated procedures. Simultaneously, we observe that the SQUAREM acceleration is susceptible to the choice of parameterization across all three underlying EM-type procedures as the proportion of very sub-optimal log-likelihoods is dramatically higher when using the constrained parameters $\boldsymbol{\theta}_{\text{cons}}$. In terms of computation time, we see in Figure 6.3 that the leapfrog-accelerated ECME1 and EM procedures have the lowest average compute times compared to their unaccelerated variants.

Table 6.1: Variance Components example, convergence frequency across 10800 simulated dataset replications. For simplicity, we present the table marginally without distinguishing the model parameters m, n_j, p, q, σ^2. Figure 6.2 provides a more detailed graphical breakdown by these parameters. Leapfrog and SQUAREM are abbreviated to LF and SQ for brevity.

	EM	LF-EM	SQ-EM	SQ-EM-T
1) All Near-Optima	419	419	419	419
2) Near-Optima	315	365	105	117
3) Sub-Optimal	6	114	16	116
4) Very Sub-Optimal	0	0	153	38
5) Did Not Converge	160	2	207	210
	ECME1	LF-ECME1	SQ-ECME1	SQ-ECME1-T
1) All Near-Optima	419	419	419	419
2) Near-Optima	315	371	99	116
3) Sub-Optimal	6	106	16	119
4) Very Sub-Optimal	0	0	152	36
5) Did Not Converge	160	4	214	210
	ECME2	LF-ECME2	SQ-ECME2	SQ-ECME2-T
1) All Near-Optima	419	419	419	419
2) Near-Optima	314	376	272	291
3) Sub-Optimal	6	103	19	134
4) Very Sub-Optimal	0	0	150	24
5) Did Not Converge	161	2	40	32

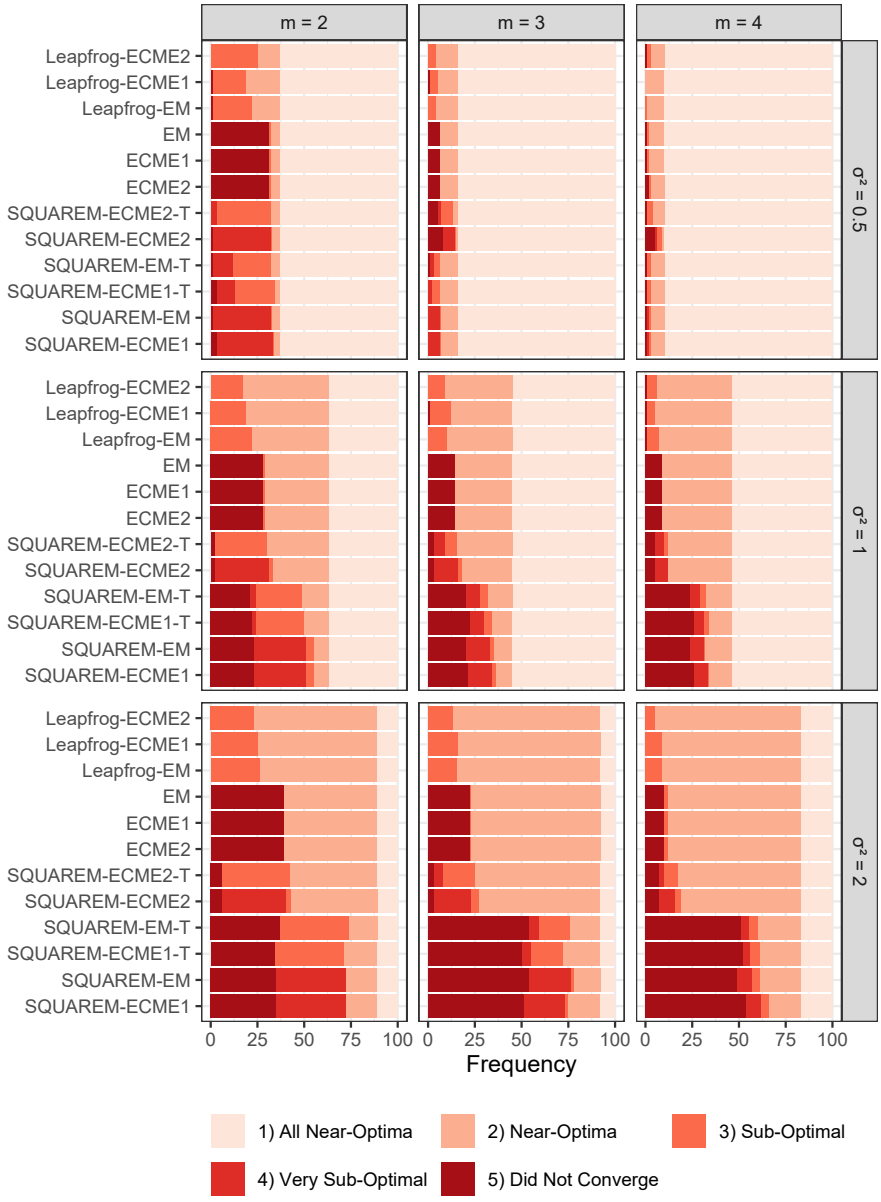

Figure 6.2: Variance Components example, stacked barplot of convergence frequency for each parameter combination of m and σ^2 with $n_j \equiv 100$ and $p = q = 2$. Estimation procedures are sorted from decreasing frequency of near-optima across the simulation study.

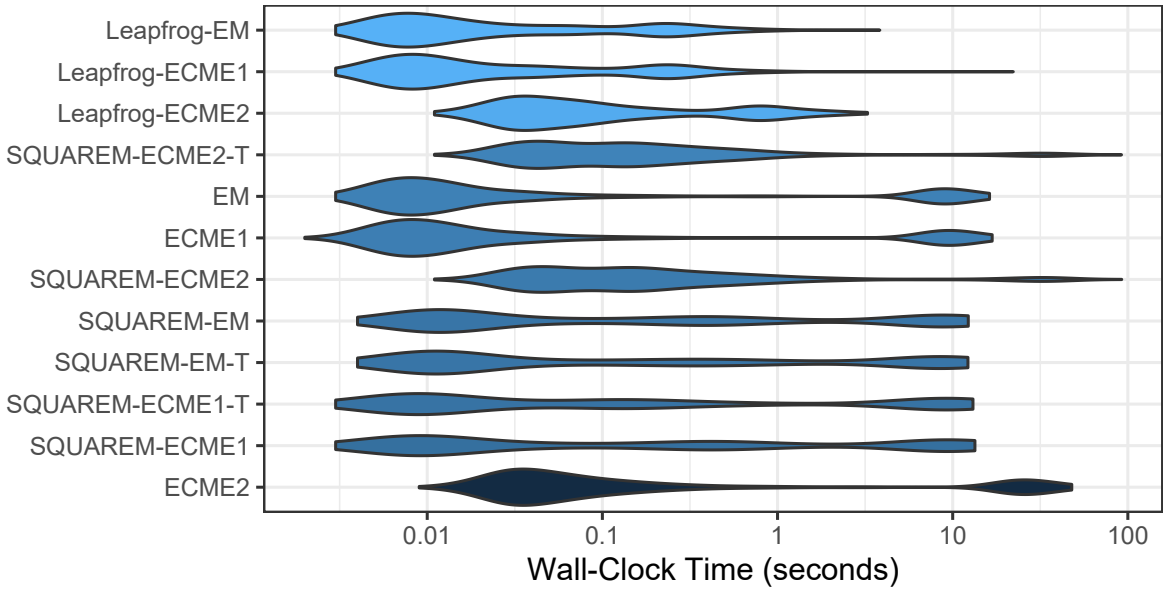

Figure 6.3: Variance Components example, violin plot of the compute time taken across twelve estimation procedures. Results are marginal across all model parameters. Procedures are sorted from top-down and shaded by increasing average wall-clock time.

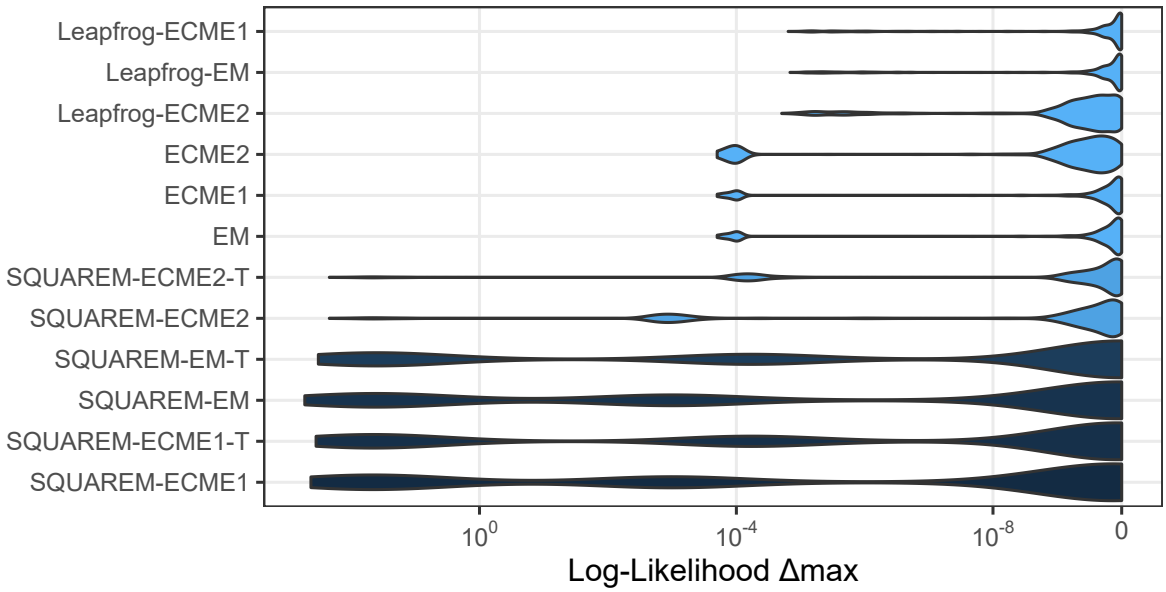

Figure 6.4: Variance Components example, violin plot of the difference in maximum log-likelihood achieved across twelve estimation procedures. Results are marginal across all model parameters. Procedures are sorted from top-down and shaded by increasing average $\Delta_{\max}$.

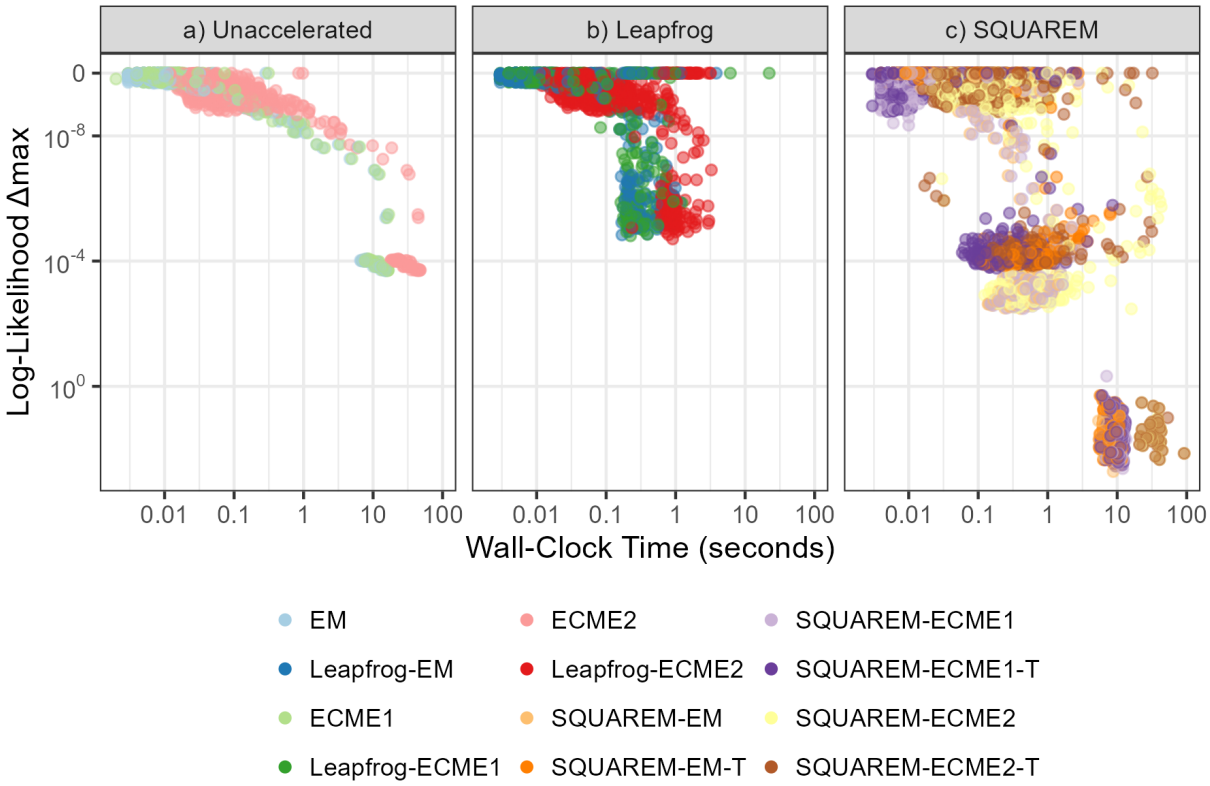

Figure 6.5: Variance Components example, scatterplot of all simulated datasets convergence results by $\Delta_{\max}$ from the best log-likelihood within each dataset and the total wall-clock compute time. ECME2 methods are omitted for simplicity, and points towards the top-left are better.

6.3.2 Factor Analysis

The factor analysis model is a dimension-reduction method whereby the observed data is assumed to be a noisy manifestation of variation within a lower-dimensional subspace. Specifically, if we observe $\boldsymbol{x}_n \in \mathbb{R}^p$ for $n = 1, 2, ..., N$, and suppose that each $\boldsymbol{x}_n$ is determined by $q < p$ latent factors $\boldsymbol{f}_n$, we let

$$\boldsymbol{x}_n = \boldsymbol{\mu} + \mathbf{L}\boldsymbol{f}_n + \varepsilon_n$$

for some mean vector $\boldsymbol{\mu}$, loading matrix $\mathbf{L} \in \mathbb{R}^{p \times q}$ to be estimated, and independent noise component ε_n. We suppose $\boldsymbol{f}_n \overset{\text{i.i.d.}}{\sim} \mathrm{N}(\mathbf{0}, \mathbf{I}_q)$ and $\varepsilon_n \overset{\text{i.i.d.}}{\sim} \mathrm{N}(\mathbf{0}, \mathbf{D})$ for some unknown diagonal matrix $\mathbf{D}$. Here, the observed log-likelihood is

$$\ell_{\text{obs}}(\boldsymbol{\theta}; \mathbf{X}) = \sum_{n=1}^{N} \log \phi(\boldsymbol{x}_n; \boldsymbol{\mu}, \mathbf{L}\mathbf{L}^\top + \mathbf{D}),$$

and the complete-data log-likelihood for latent data $\mathbf{F}$ is

$$\ell_{\text{com}}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{F}) = \sum_{n=1}^{N} \left[\log \phi(\boldsymbol{f}_n; \mathbf{0}, \mathbf{I}_q) + \log \phi(\boldsymbol{x}_n; \boldsymbol{\mu} + \mathbf{L}\boldsymbol{f}_n, \mathbf{D})\right],$$

where ϕ is the multivariate normal density function. Without loss of generality, we may assume the observed data to be centred and $\boldsymbol{\mu} = \mathbf{0}$. This factor analysis model lends itself directly to applying EM-type procedures (Rubin and Thayer, 1982). Conveniently, the complete-data log-likelihood can also be re-written in terms of the sufficient statistics (McLachlan and Krishnan, 2008)

$$\mathbf{C}_{xx} = \mathbf{X}^\top \mathbf{X}, \mathbf{C}_{xf} = \mathbf{X}^\top \mathbf{F}, \mathbf{C}_{ff} = \mathbf{F}^\top \mathbf{F},$$

the latter two of which are linear in latent data $\mathbf{F}$. As such, the leapfrog-acceleration surrogate can be re-arranged to perform acceleration on the expected values of $\mathbf{C}_{xf}$ and $\mathbf{C}_{ff}$ from iteration to iteration. However, for large values of acceleration factor α, the leapfrog-accelerated expectation of $\mathbf{C}_{ff}$ may not necessarily be positive-definite. As well, the parameter $\mathbf{D}$ must be a (diagonal) positive-definite matrix. When either requirement is violated, we consider the parameters invalid and reject the LM-step update in favour of

a backtracking step or an M-step update. Furthermore, we use the constrained parameters $\boldsymbol{\theta}_{\text{cons}} = \langle \text{vec}\, \mathbf{L}, \text{diag}\, \mathbf{D} \rangle$ and its unconstrained transformation $\boldsymbol{\theta}_{\text{unc}} = \langle \text{vec}\, \mathbf{L}, \log \text{diag}\, \mathbf{D} \rangle$ for SQUAREM acceleration.

Simulation Study

To examine the behaviour of the proposed procedure, we generate simulated data from the specified factor analysis and perform parameter estimation. We use a full-factorial design over the number of observations $n = 500, 1000, 2000$, observed data dimension $p = 8, 10$, latent factor dimension $q = 4, 6$, and noise diagonal matrix $\mathbf{D} = \sigma^2 \mathbf{I}_p = 0.5\mathbf{I}_p, 1\mathbf{I}_p, 2\mathbf{I}_p$. Both the loading matrix and latent factor elements are drawn from i.i.d. standard normal distributions $\text{N}(0, 1)$ with the true mean $\boldsymbol{\mu}$ chosen to be $\mathbf{0}$. We run 100 replications at each parameter combination, with an iteration limit of $t_{\max} = 10000$. We evaluate the EM and ECME procedure of Liu and Rubin (1994) with their leapfrog and SQUAREM accelerated versions. Initial parameter values are determined by a principal components analysis.

In Table 6.2, we see that among the accelerated procedures the leapfrog-accelerated ECME procedure yields the greatest proportion of near-optimal solutions, while the leapfrog-accelerated EM has the lowest proportion. Interestingly, for SQUAREM acceleration the proportion of near-optima is reversed compared to leapfrog-acceleration. In terms of Heywood cases in Section 6.3.2, we see that leapfrog-accelerated EM produces a disproportionately high number thereof whilst leapfrog-accelerated ECME only has a slightly higher number compared to the SQUAREM procedures. Moreover, the number of non-convergent solutions is also lowest for leapfrog-accelerated ECME, suggesting that it is the most reliable among the eight methods. Conversely, using Figures 6.7 to 6.9 we observe that leapfrog-accelerated EM has the fastest average compute time, and tends to be faster consistently at the cost of a greater deviation from the maximum observed log-likelihood. SQUAREM acceleration exhibits a more extreme bimodal distribution on the convergence times than the other methods compared to leapfrog acceleration. From Figure 6.6, we see that leapfrog acceleration's performance remains more consistent with increased noise variance σ^2 than SQUAREM.

Table 6.2: Factor Analysis example, convergence frequency across all simulated dataset replications. For simplicity, we present the table marginally without distinguishing the factor analysis model parameters n, p, q. Figure 6.6 provides a graphical breakdown by these parameters. SQUAREM is abbreviated to SQ for brevity.

	EM	Leapfrog-EM	SQ-EM	SQ-EM-T
1) All Near-Optima	1783	1783	1783	1783
2) Near-Optima	34	205	867	779
3) Sub-Optimal	698	1082	499	4
4) Very Sub-Optimal	14	21	30	1
5) Did Not Converge	1071	509	421	1033
	ECME	Leapfrog-ECME	SQ-ECME	SQ-ECME-T
1) All Near-Optima	1783	1783	1783	1783
2) Near-Optima	213	914	528	592
3) Sub-Optimal	1340	794	885	798
4) Very Sub-Optimal	67	96	77	66
5) Did Not Converge	197	13	327	361

Finally, we note that the SQUAREM-acceleration of EM with the unconstrained parameterization either performs well or fails to converge with little in between, which is dramatically different from the same method with the untransformed constrained parameters. The same effect is much less apparent with ECME and SQUAREM acceleration.

6.3.3 Finite Gaussian Mixture Models

Finite Gaussian mixture models are a common method for clustering and classification tasks with continuous data. The model purports the data to follow a finite mixture of multivariate normal distributions, whereby each observation $\boldsymbol{x}_n$ for $n = 1, 2, ..., N$ comes from class $g = 1, 2, ..., G$ with probability π_g and observed data distribution $\boldsymbol{x}_n \sim \mathrm{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$.

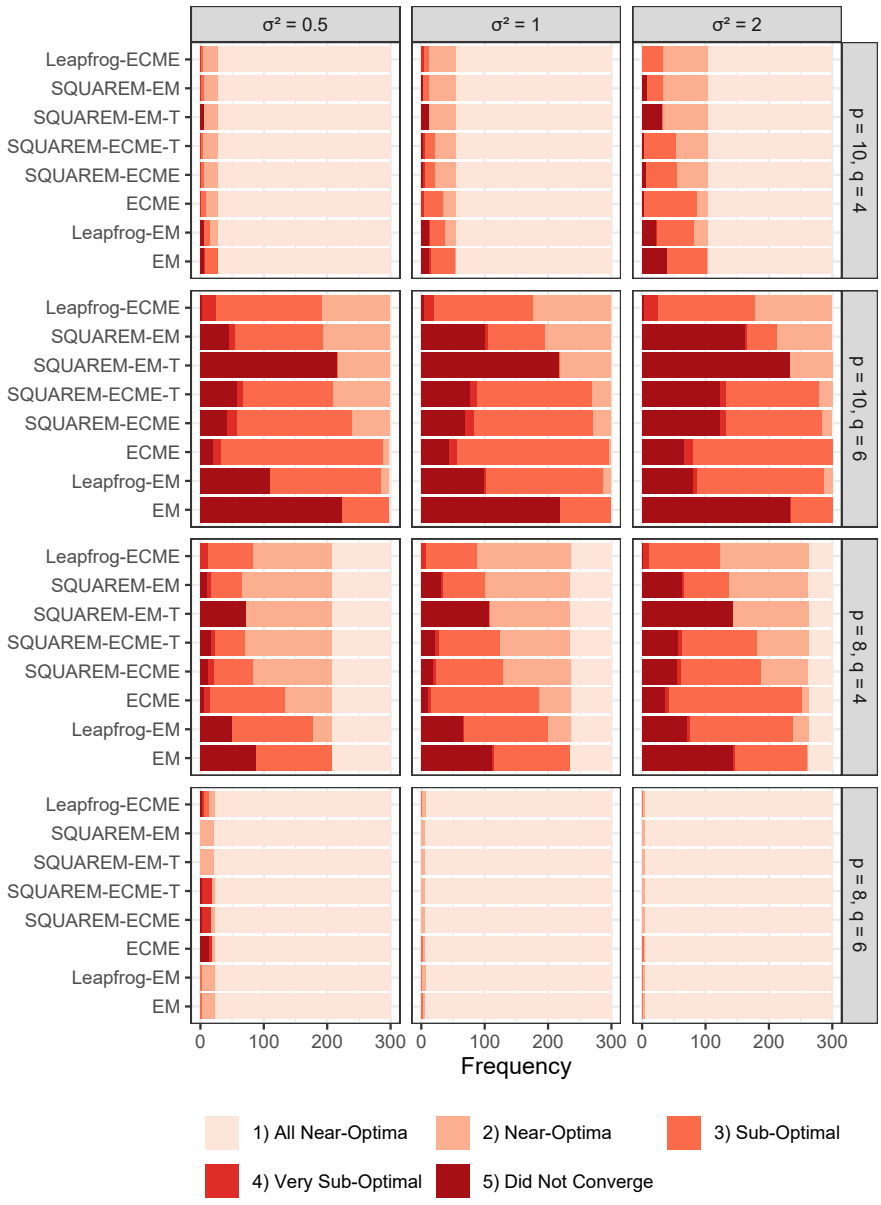

Figure 6.6: Factor Analysis example, stacked barplot of convergence frequencies marginalized across p and q for each estimation procedure. Near-optimality is as defined in Section 6.3. Estimation procedures are organized from top-down by decreasing total frequency of near-optima.

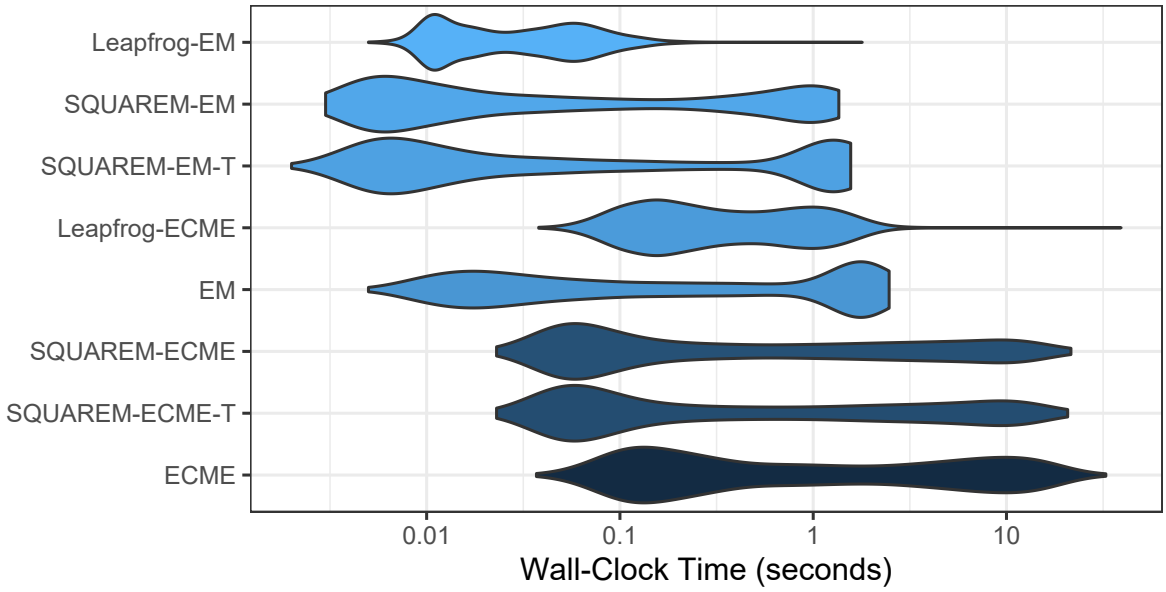

Figure 6.7: Factor Analysis example, violin plot of the compute time taken across eight estimation procedures. Results are marginal across all model parameters. Procedures are sorted from top-down and shaded by increasing average wall-clock time.

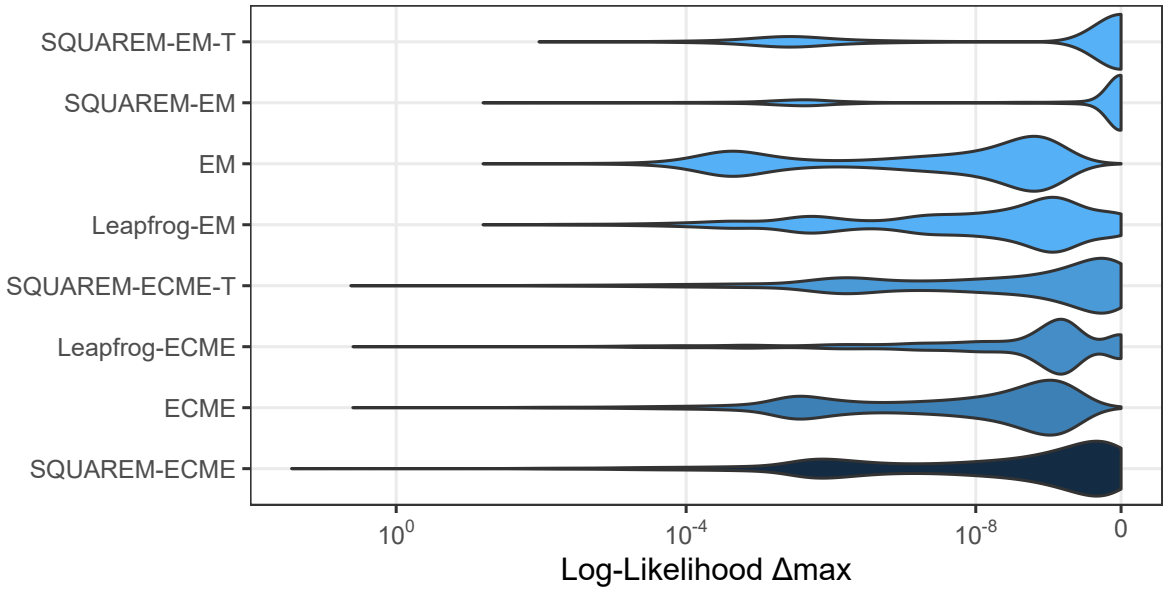

Figure 6.8: Factor Analysis example, violin plot of the difference in maximum log-likelihood achieved across twelve estimation procedures. Results are marginal across all model parameters. Procedures are sorted from top-down and shaded by increasing average $\Delta_{\max}$.

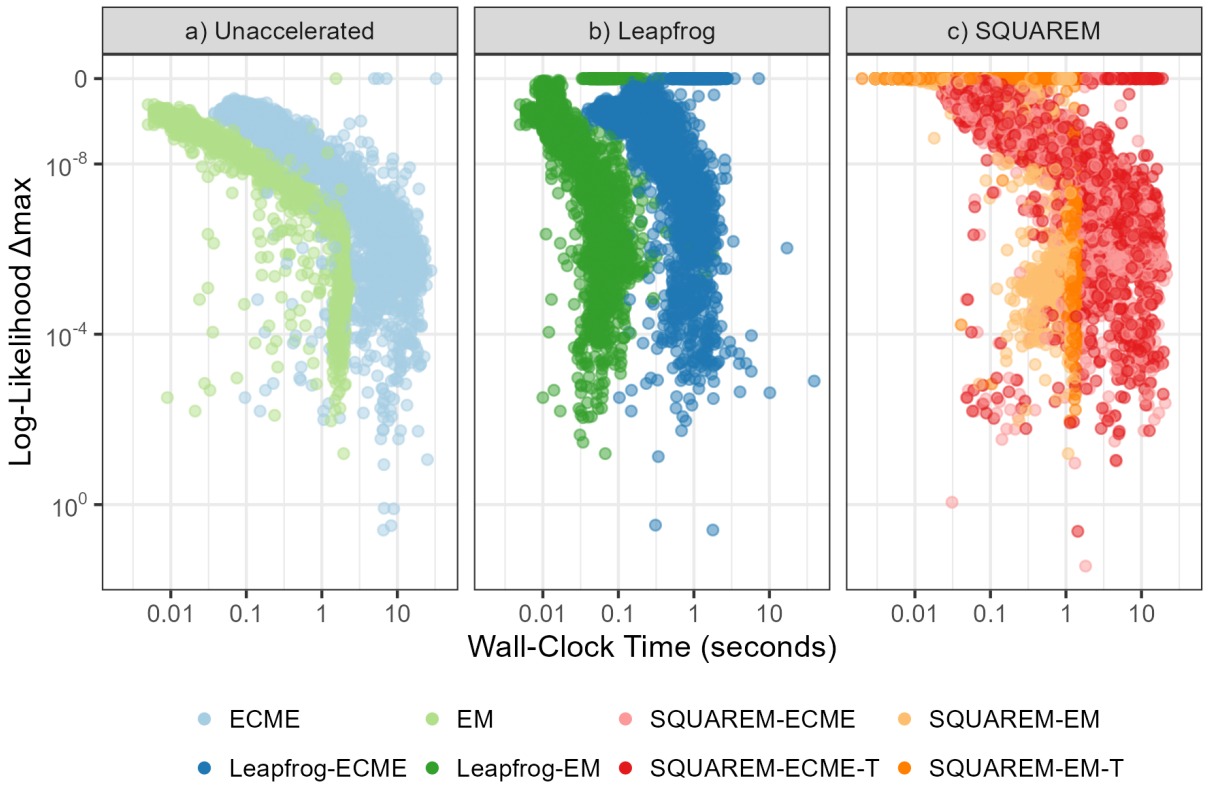

Figure 6.9: Factor Analysis, scatterplot of all simulated datasets convergence results by $\Delta_{\max}$ from the best log-likelihood within each dataset and the total wall-clock compute time.

Table 6.3: Factor Analysis example, Heywood case frequency across all simulated dataset replications. Heywood cases are defined as those with a fitted variance below 10^{-8}, including negative values.

	Non-Heywood Case	Heywood Case
EM	3600	0
Leapfrog-EM	2753	847
SQUAREM-EM	3596	4
SQUAREM-EM-T	3600	0
ECME	3596	4
Leapfrog-ECME	3588	12
SQUAREM-ECME	3593	7
SQUAREM-ECME-T	3597	3

The problem can be formulated as a maximum likelihood problem directly on the observed log-likelihood

$$\ell_{\text{obs}}(\boldsymbol{\theta}; \mathbf{X}) = \sum_{n=1}^{N} \log \left[\sum_{g=1}^{G} \pi_g \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \right],$$

where ϕ is the multivariate normal density function. However, it is more tractable to introduce missing data $\mathbf{Z}$ where $z_{ng} = 1$ if observation n belongs to cluster g. This allows use of the EM procedure on the complete-data log-likelihood

$$\ell_{\text{com}}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} z_{ng} \left[\log \pi_g + \log \phi(\boldsymbol{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \right].$$

As indicated in Section 6.2, we note that $\ell_{\text{com}}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})$ is linear in the latent data z_{ng}, and so the leapfrog-expectation accelerated surrogate can be written as

$$\mathrm{R}(\boldsymbol{\theta}, \alpha; \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)}) = \sum_{n=1}^{N} \left[(1+\alpha)\, \mathrm{E}^{(t)}[z_{ng}] - \alpha\, \mathrm{E}^{(t-1)}[z_{ng}] \right] \left[\log \pi_g + \log \phi(\boldsymbol{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \right]$$

so as to apparently permit the same M-step as in EM. However, the leapfrog-accelerated expected value $(1+\alpha)\, \mathrm{E}^{(t)}[z_{ng}] - \alpha\, \mathrm{E}^{(t-1)}[z_{ng}]$ may be negative for one or many n at some

α. While this cannot cause an invalid parameter estimate for $\boldsymbol{\mu}_g \in \mathbb{R}^d$, it can no longer guarantee a positive-definite estimate for $\boldsymbol{\Sigma}_g \in \mathbb{R}^{d \times d}$. When a $\boldsymbol{\Sigma}_g$ is not positive-definite, we consider the parameters to be invalid and reject the LM-step update in favour of an M-step update. Similarly, we consider $\pi_g \leq 0$ as invalid parameters for any g.

Here, for SQUAREM we consider the alternate parameterization by transforming the typical constrained parameterization

$$\boldsymbol{\theta}_{\text{cons}} = \langle \boldsymbol{\pi}, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_g, \text{vech}\, \boldsymbol{\Sigma}_1, ..., \text{vech}\, \boldsymbol{\Sigma}_g \rangle$$

into the unconstrained parameterization

$$\boldsymbol{\theta}_{\text{unc}} = \langle \log \pi_1 - \log \pi_g, ..., \log \pi_{g-1} - \log \pi_g, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_g, \text{vech chol}\, \boldsymbol{\Sigma}_1, ..., \text{vech chol}\, \boldsymbol{\Sigma}_g \rangle,$$

where chol$(\cdot)$ is the Cholesky decomposition operator and vech$(\cdot)$ is the half-vectorization operator. The inverse transformation is to apply the soft-argmax function to recover $\boldsymbol{\pi}$ and multiplying out the Cholesky factorization to recover $\boldsymbol{\Sigma}_g$.

Simulation Study

To examine the behaviour of the proposed procedure, we generate simulated data from the finite Gaussian mixture model and perform parameter estimation. The simulation study is a full-factorial design across number of observations $n = 1000, 2000, 3000$, data dimension $d = 2, 3, 4$, and component covariance matrices $\boldsymbol{\Sigma}_g = \sigma^2 \mathbf{I} \in \{0.5\mathbf{I}, \mathbf{I}, 2\mathbf{I}, 4\mathbf{I}\}$. The cluster centres $\boldsymbol{\mu}_g$ are placed at the corners of the d-hypercube $\{-1, +1\}^p$ so as to yield 2^d clusters. The cluster probabilities π_g are chosen to be equal at 2^{-d}. 100 replications are performed at each parameter combination, with datasets drawn from the true model. Both the EM and its leapfrog accelerated procedures are fitted on each dataset replication with an upper-bound on the maximum number of iterations $t_{\max} = 10000$ for both procedures. Initial parameter values $\boldsymbol{\theta}^{(0)}$ are determined by a k-means start with the correct number of clusters.

For the SQUAREM accelerated procedure, we consider an alternative parameterization that frees the constraints on $\boldsymbol{\pi}$ and $\boldsymbol{\Sigma}_g$. We define the constrained parameterization $\boldsymbol{\theta}_{\text{cons}} =$

$\langle \boldsymbol{\pi}, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_g, \text{vech}\, \boldsymbol{\Sigma}_1, ..., \text{vech}\, \boldsymbol{\Sigma}_g \rangle$ and the unconstrained transformation thereof

$$\boldsymbol{\theta}_{\text{unc}} = \langle \log \pi_1 - \log \pi_g, ..., \log \pi_{g-1} - \log \pi_g, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_g, \text{vech chol}\, \boldsymbol{\Sigma}_1, ..., \text{vech chol}\, \boldsymbol{\Sigma}_g \rangle$$

, where chol$(\cdot)$ is the Cholesky decomposition operator and vech$(\cdot)$ is the half-vectorization operator.

In Table 6.4, we examine the different convergence behaviours of the four tested procedures over all model parameter combinations. The leapfrog-accelerated EM procedure seems to converge to a near-optimal solution more often than either of the SQUAREM variations. Moreover, we also see that the SQUAREM-accelerated EM procedure can fail to converge slightly more often, and the transformed parameterization of the same fails to converge much more often.

In Table 6.4, we observe the convergence frequency of all four tested procedures. We see that SQUAREM- and leapfrog-accelerated EM both show a significant improvement over unaccelerated EM, which exhibits very slow time to convergence. In this case, SQUAREM is more reliable in terms of approaching the best observed log-likelihood, but has a slightly higher frequency of non-convergence. Similar behaviour is seen in Figures 6.11 to 6.13, where an improvement over unaccelerated EM is seen. In Figure 6.10, we see that all methods experience increasing difficulty for greater σ^2, signifying greater cluster overlap, and larger d, meaning more dimensions and exponentially more mixture components.

6.4 Discussion

The leapfrog-expectation acceleration is a novel approach towards accelerating EM-type algorithms by moving along the implicit landscape defined by the surrogate function, yielding notable speed increases compared to unaccelerated methods without being constrained by the need to choose a specific problem parameterization. We provide the leapfrog-acceleration as a general framework, and a specific applied procedure with Aitken's acceleration guidance for backtracking line-search. In simulation studies over a selection of popular statistical models often estimated slowly with the EM (and ECME) procedures,

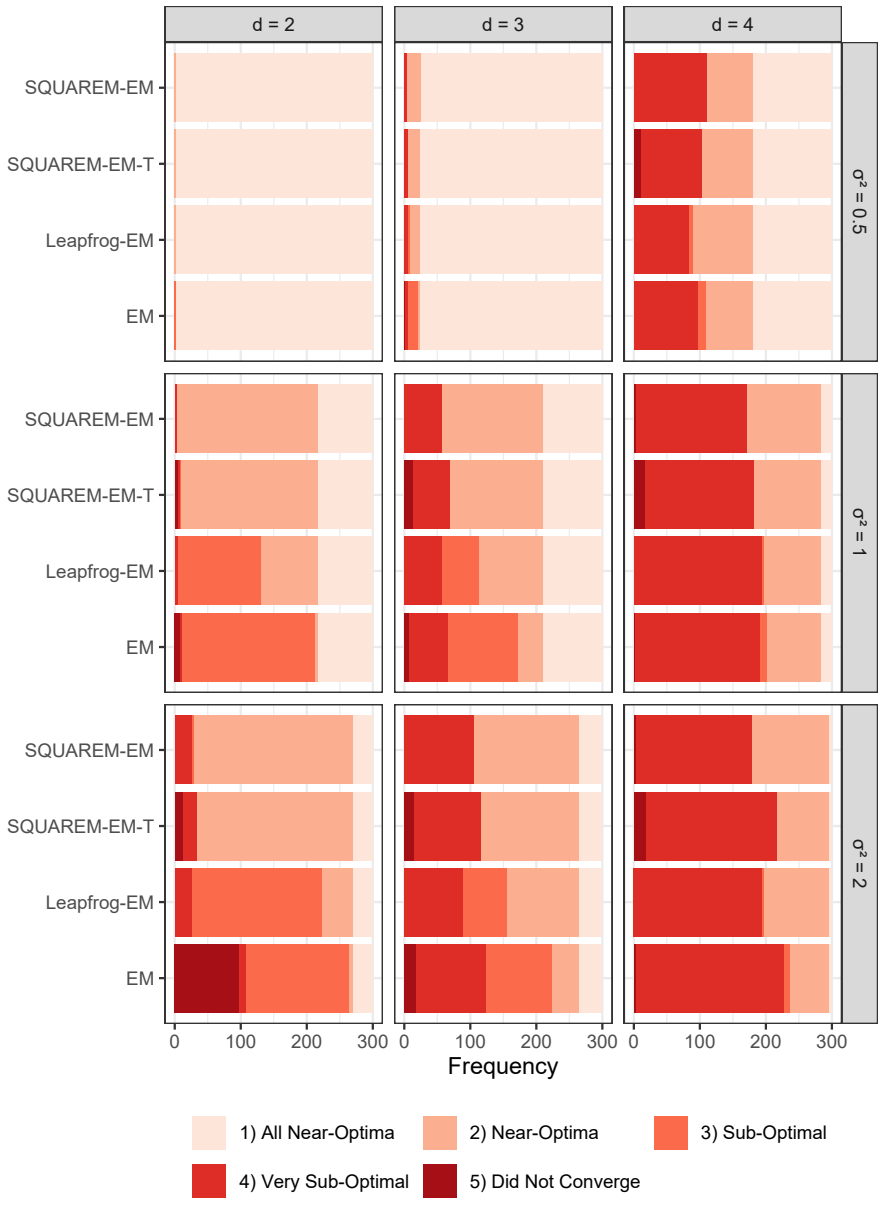

Figure 6.10: Finite Gaussian Mixture Model example, stacked barplot of convergence frequencies marginalized across p and q for each estimation procedure. Near-optimality is as defined in Section 6.3. Estimation procedures are organized from left-to-right by decreasing total frequency of near-optima.

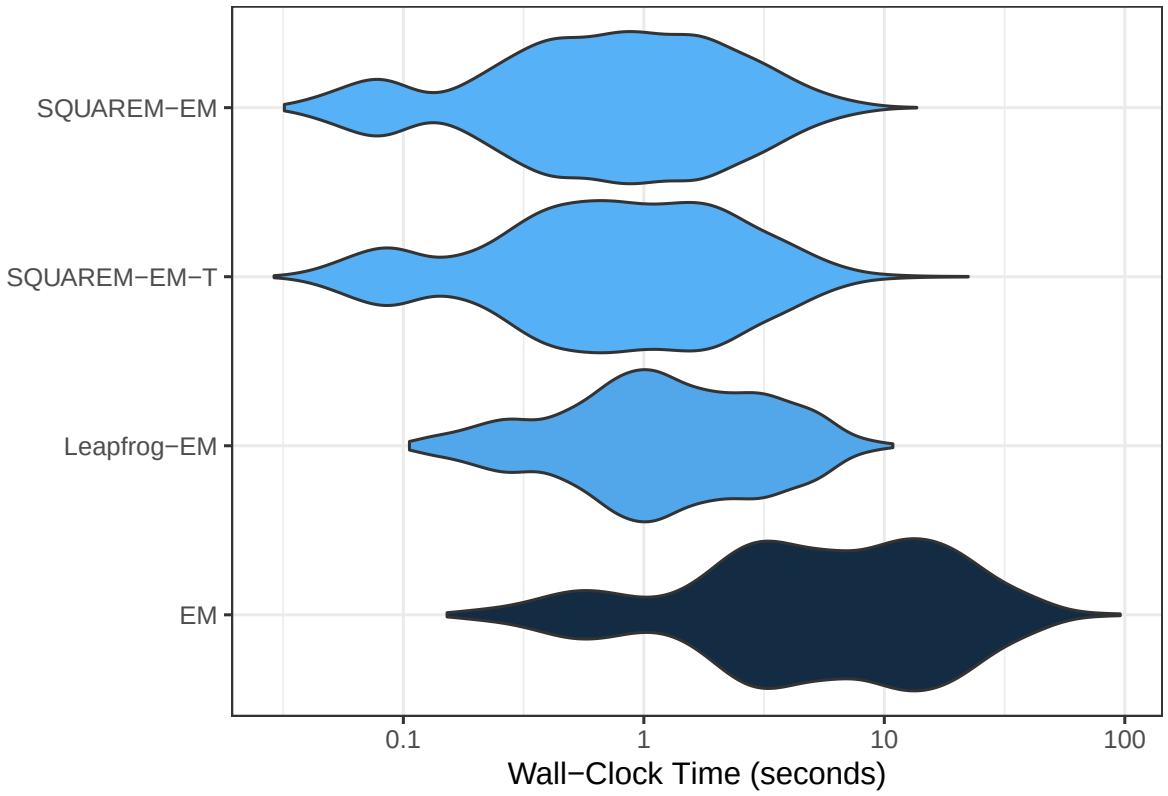

Figure 6.11: Finite Gaussian Mixture Model example, violin plot of the compute time taken across eight estimation procedures. Results are marginal across all model parameters. Procedures are sorted from top-down and shaded by increasing average wall-clock time.

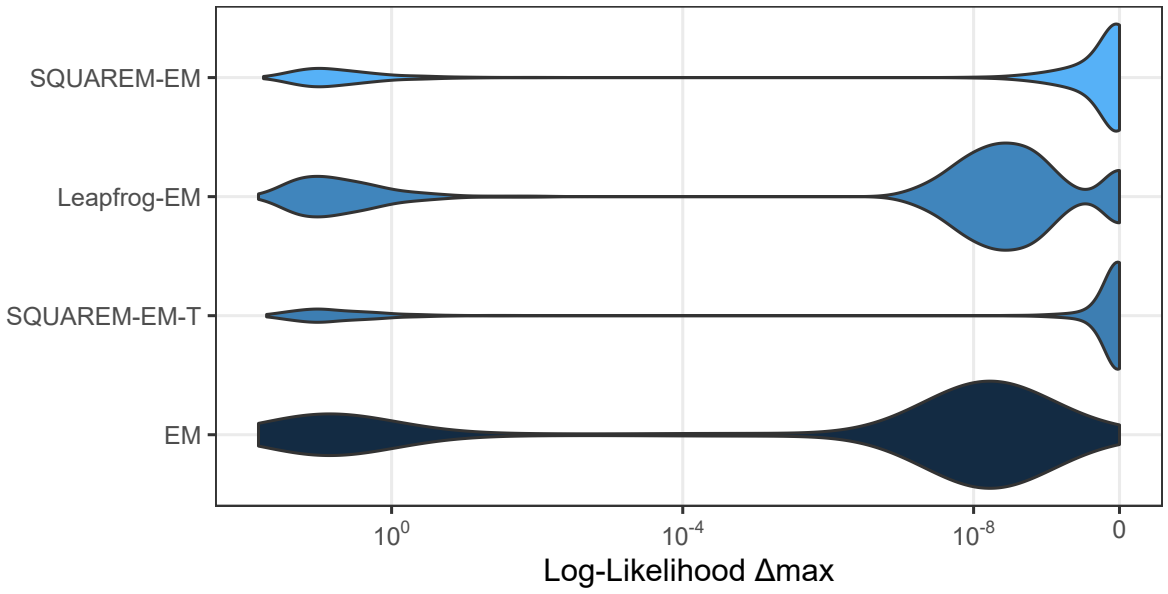

Figure 6.12: Finite Gaussian Mixture Model example, violin plot of the difference in maximum log-likelihood achieved across twelve estimation procedures. Results are marginal across all model parameters. Procedures are sorted from top-down and shaded by increasing average $\Delta_{\max}$.

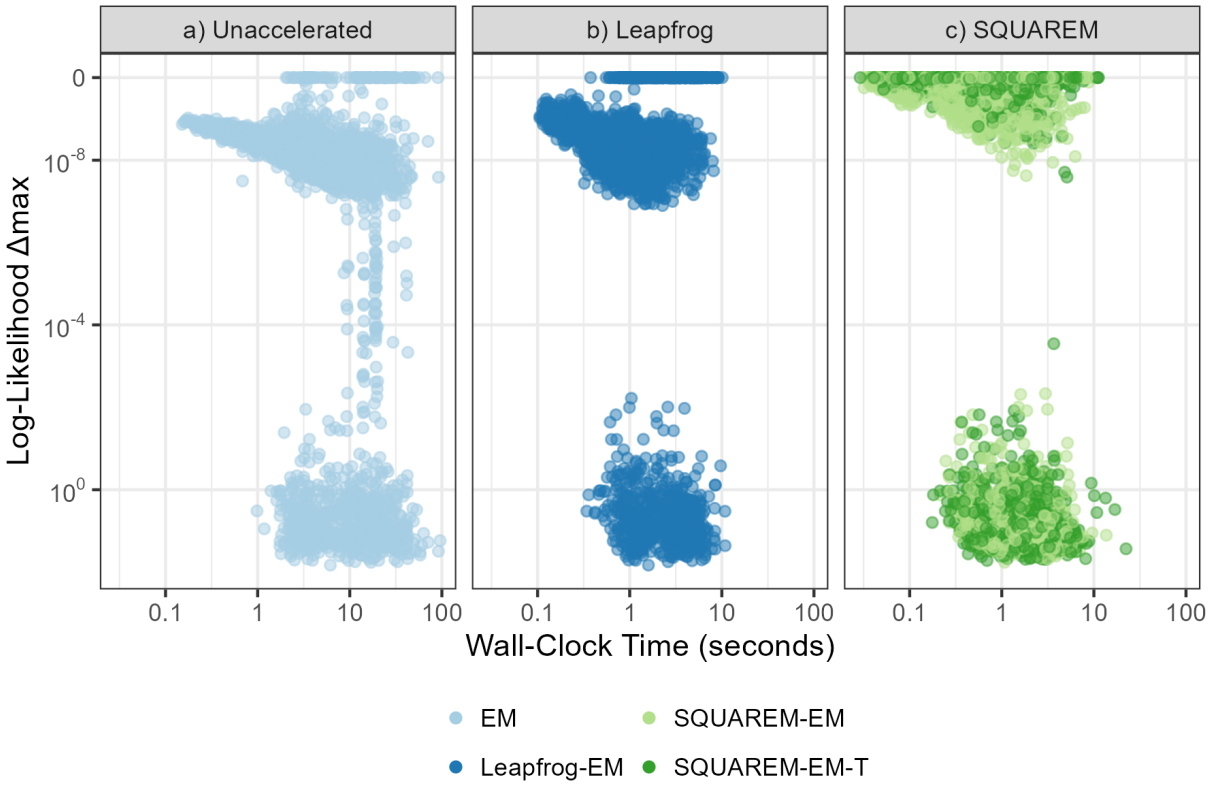

Figure 6.13: Finite Gaussian Mixture Model, scatterplot of all simulated datasets convergence results by $\Delta_{\max}$ from the best log-likelihood within each dataset and the total wall-clock compute time.

Table 6.4: Convergence frequencies for the Finite Gaussian Mixture model, marginally over all model parameters n, d, σ^2. Near-optimality is defined as converging to a solution within 10^{-8} of the best log-likelihood across all procedures for that simulation replication. Sub-optimal is defined similarly between 10^{-4} and 10^{-8}, and very sub-optimal contains all remaining convergent solutions. For brevity, SQUAREM is abbreviated to SQ.

	EM	Leapfrog-EM	SQ-EM	SQ-EM-T
1) All Near-Optima	1045	1045	1045	1045
2) Near-Optima	435	885	1553	1472
3) Sub-Optimal	805	686	6	1
4) Very Sub-Optimal	1018	982	983	907
5) Did Not Converge	297	2	13	175

and in comparison to the SQUAREM acceleration framework (Varadhan and Roland, 2008), we see that the leapfrog-acceleration produces comparable estimation speeds with greater reliability. In particular, for the Variance Components model in Section 6.3.1 and the Factor Analysis model in Section 6.3.2 we see that the performance of SQUAREM can change dramatically based on the chosen parameterization; performing a change of variables into an unconstrained space can yield either major improvements or major detriments to the estimation procedure.

Chapter 7

Generalized Linear Models for Massive Data via Doubly-Sketching

7.1 Introduction

In the contemporary age, large-scale data gathering can produce datasets with millions or billions of observations. Generalized linear models (GLMs) (McCullagh and Nelder, 1989) form the mainstay of many analyses, providing a framework capable of handling more response variable distributions compared to ordinary least squares regression. As datasets grow in size, previously tractable methods can consume infeasible amounts of time and computational resources. Estimating GLM parameters often requires the Iteratively Reweighted Least Squares (IRLS) procedure that can run aground on such datasets. In particular, infrastructure constraints such as machine memory, storage medium, and disk/network transfer speeds can all pose bottlenecks for effective analysis of larger datasets. Stochastic methods offer a solution by trading off accuracy in the answer for reduced computational burden.

In broad strokes, existing methodologies tackling computational tractability on large datasets can be considered data engineering approaches or stochastic approximation approaches, or a combination of the two. In the former case, parallelization techniques from

numerical linear algebra can be used on a multi-threaded CPU (Dagum and Menon, 1998; Blackford et al., 1997; Chapman et al., 2007) or across multiple computers (Leskovec et al., 2020) in a split-apply-combine approach. Yu et al. (2022a) extends this paradigm to computing maximum quasi-likelihood estimates in a distributed fashion. Other techniques include leveraging GPU computational capabilities (Suchard et al., 2013; Kylasa et al., 2019). We focus our efforts on the latter mode of achieving computational speed. In this approach, stochastic approximation relies on some probabilistic data reduction method to bring the problem into a tractable regime. In Newton-Raphson, subsampling the data to form the Hessian/information matrix and/or gradient/score vector for a (quasi-)Newton's method approach are possible (Pilanci and Wainwright, 2017; Byrd et al., 2011; Bollapragada et al., 2018; Xu et al., 2016; Roosta-Khorasani and Mahoney, 2019; Lacotte et al., 2020). As a special case, sketching the gradient/score only can be considered a variation on stochastic gradient descent. For ordinary least squares regression, there are a variety of approximate linear algebra methods (Sarlos, 2006), (Dhillon et al., 2013; Drineas et al., 2011; Kleiner et al., 2014), leverage scores (Wang et al., 2019; Ma and Sun, 2015), and stochastic approximations (Cormode, 2011; Mahoney, 2011; Ahfock et al., 2020). In the context of logistic regression, Munteanu et al. (2021) provides a method for logistic regression with probabilistic guarantees on the sketched estimate. An alternative method to reduce the computational burden is by subsampling the dataset in an informed way, such as the OSMAC method of (Wang et al., 2018), which is compared against in Section 7.4.1. A variation of IRLS using leverage scores across the whole dataset is found in Dahiya et al. (2018).

We propose a stochastic approximation to the IRLS procedure using two sequential sketching steps to control both data transfer and computation cost, respectively. We focus on the computational tractability problem under the wall-clock time paradigm as opposed to a computational complexity analysis of long-run performance and show the value of the proposed method under practical infrastructure constraints, particularly when data transfer time is dominant and local memory is constrained. The central proposition is the generation of a smaller and more tractable surrogate dataset to be used in lieu of the full dataset at each iteration of the IRLS algorithm. Moreover, we avoid evaluating the

objective log-likelihood, gradient, or Hessian at any iteration, which improves performance for massive datasets stored off-site whose retrieval is bottlenecked by data transfer speeds. By comparison, data-aware methods that use leverage scores for every observation in the dataset such as Wang et al. (2018, 2019); Ma and Sun (2015); Yu et al. (2022b); Drineas et al. (2012) require passing through the entire dataset at least once. To put this comparison into perspective, to retrieve 100 gigabytes of data across a pedestrial gigabit network connection requires approximately 13 minutes under ideal conditions. As well, the proposed method is independent of the model specification and can fit multiple models simultaneously, greatly aiding model selection; statistical leverage score methods produce sampling weights which apply to a single model specification and so the weighted subsamples are only valid for that model.

In Section 7.2, the proposed model is described and theoretical properties are shown. To evaluate the probabilistic and computational properties, we perform a simulation study in Section 7.3 over a variety of dataset magnitudes and infrastructures. In Section 7.4.1, we use the SUSY dataset comprising 5 million observations in comparison to the OSMAC Wang et al. (2018), Optimal Distributed Sampling (ODS) (Yu et al., 2022b), Fast A-Optimal Subsampling Probability Approximation (FASA), stochastic gradient descent (SGD), and single subsamples. In Section 7.4.3, we investigate the performance at a scale of 1.7 billion observations under adverse dataset conditioning and obtain approximate GLM parameter estimates and standard error estimates in 25 minutes compared to a full IRLS approach taking over 20 hours.

7.2 Generalized Linear Models via Doubly-Sketching

In the present work, we propose a method of approximately fitting generalized linear model regression models with sketching to mitigate two distinct sources of computational burden. We apply sketching at each iteration of the IRLS procedure to update the estimates of regression coefficients and standard errors for large datasets with computational infrastructure constraints. A central consideration is that at no iteration is a complete traversal of

the dataset used to calculate either the log-likelihood, gradient, or Hessian; this tackles the challenges of large-scale data transfer simultaneously with the usual computational complexity. Specifically, larger datasets impose storage constraints and infrastructure limits; storing the entirety of the data within random-access memory (RAM) yields the best performance but is often infeasible due to unavailability. More spacious storage can be found in non-volatile local storage such as solid-state disks (SSD) and mechanical hard drives (HDD) at substantial performance detriment (Foong and Hady, 2016). Datasets can also be stored across the network in a remote database or data warehouse which can exacerbate slowness especially when accessed simultaneously by multiple users or under poor network conditions. We intend to control the cost of both data transfer and computational complexity through the surrogate dataset sizes in a two-stage sketch design.

7.2.1 Preliminaries

Generalized linear models are an extension of ordinary least squares regression models whereby the response variable Y can assume a variety of distributions. The particular subclass of GLMs that we are interested in are those whose response variables Y_i follow a regular exponential family (REF) distribution; that is, the probability mass/density function has the form

$$f(y_i; \theta_i, \phi) = \exp\left[\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y; \phi)\right],$$

where a, b, and c are known functions that characterize the response distribution, θ_i is the canonical parameter, and ϕ are any nuisance parameters assumed to be known. The covariates $\boldsymbol{x}_i$ corresponding to observation i enter the model by way of a linear predictor $\eta_i = \boldsymbol{x}_i^\top \boldsymbol{\beta}$, which in turn is associated with the expected value μ_i of y_i via a link function g such that $g(\mu_i) = \eta_i$. When g is the canonical link function, η_i and θ_i coincide. While a variety of possible response distributions and link functions are possible, we select three common examples; the binomial with logit link, the binomial with complementary log-log (cloglog) link, and the Poisson with log link.

We are often interested in the parameter vector $\boldsymbol{\beta}$, representing the effects of each covariate, which is commonly estimated by maximum likelihood methods. For a dataset with n observations the parameter of interest is

$$\hat{\boldsymbol{\beta}}_{\text{MLE}} = \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \log f(y_i; \theta(\boldsymbol{x}_i^\top \boldsymbol{\beta}), \phi). \tag{7.1}$$

As the maximization (7.1) is often intractable, $\hat{\boldsymbol{\beta}}_{\text{MLE}}$ can be estimated iteratively using the IRLS algorithm (McCullagh and Nelder, 1989). Let $\mathbf{X}$ be the $n \times d$ matrix of covariates and $\boldsymbol{y}$ be the $n \times 1$ vector of responses. The corresponding IRLS update, sometimes known as Fisher Scoring, can be expressed as

$$\hat{\boldsymbol{\beta}}^{(t+1)} = \hat{\boldsymbol{\beta}}^{(t)} + (\mathbf{X}^\top \mathbf{W}^{(t)} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}^{(t)} \boldsymbol{z}^{(t)} \tag{7.2}$$

where $\mathbf{W}^{(t)}$ is a diagonal matrix with elements $w_i = \frac{1}{a(\phi)} \frac{\partial \theta_i}{\partial \mu_i} \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2$ and $z_i = (y_i - \mu_i) \frac{\partial \eta_i}{\partial \mu_i}$ is known as the adjusted response variable. The superscripts on $\mathbf{W}^{(t)}$ and $\boldsymbol{z}^{(t)}$ here emphasize that they are functions of $\hat{\boldsymbol{\beta}}^{(t)}$; for reduced notational burden we will usually omit these in the remainder of the work. As an iterative algorithm, this update is repeated until some convergence criterion is met. Asymptotically for $n > d$, the update (7.2) has time-complexity $O(nd^2)$.

For sufficiently large datasets, previously viable approaches may be inadequate under practical constraints such as wall-clock time or hardware budget. The $O(nd^2)$ complexity of IRLS can become intractable for both large n and large d. Sketching (Cormode, 2011; Mahoney, 2011; Ahfock et al., 2020) is a method for tackling the former situation by using a surrogate dataset with a reduced number of observations to increase computational tractability. In data engineering disciplines, this practice falls under the umbrella of approximate query processing (Cormode, 2011).

A linear sketch can be characterized by a stochastic matrix $\mathbf{S} \in \mathbb{R}^{k \times n}$ that projects the data from n observations down to $k < n$, and can be data-aware or data-oblivious (Ahfock et al., 2020). The former is a function of the data, such as weighting by leverage scores (Ma and Sun, 2015). The latter draws $\mathbf{S}$ independently of the data. Four notable data-oblivious linear sketches (Pilanci and Wainwright, 2017; Ahfock et al., 2022) are

the Uniform, Gaussian, Hadamard (Ailon and Chazelle, 2009), and Clarkson-Woodruff (Clarkson and Woodruff, 2017) sketches.

In a regression context with covariates $\mathbf{X}$ and response $\boldsymbol{y}$, we may draw a sketched surrogate dataset $(\tilde{\mathbf{X}}, \tilde{\boldsymbol{y}}) = (\mathbf{S}\mathbf{X}, \mathbf{S}\boldsymbol{y})$. Normally, the fitted regression coefficients $\hat{\boldsymbol{\beta}}$ have closed-form solution $(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{y}$. Applying sketching to approximate this solution (Sarlos, 2006; Drineas et al., 2006) yields two approaches: partial sketching where only the Gram matrix $\mathbf{X}^\top\mathbf{X}$ is sketched, and complete sketching where $\mathbf{X}^\top\boldsymbol{y}$ is also sketched (Drineas et al., 2006; Pilanci and Wainwright, 2016; Dhillon et al., 2013). In particular, the complete sketching estimate is written as

$$\hat{\boldsymbol{\beta}}_{\text{Sketch}} = (\mathbf{X}^\top\mathbf{S}^\top\mathbf{S}\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{S}^\top\mathbf{S}\boldsymbol{y} \tag{7.3}$$

for a random sketch matrix $\mathbf{S}$. As $\mathbf{S}$ is a random variable, this induces a distribution on $\hat{\boldsymbol{\beta}}_{\text{Sketch}}$, whose asymptotic distributional properties are discussed in Ahfock et al. (2020) for select sketches.

7.2.2 Methodology

Herein, we propose a doubly-sketched approximate method for finding the maximum likelihood estimate of a GLM problem of the form (7.1). We define our sketch to be the composition of a uniform sketch with a Clarkson-Woodruff sketch, with each sketch tackling a different facet of the computational tractability problem (Figure 7.1). For datasets in great excess of available system memory, we use the uniform sketch size to control the data transfer cost from the storage medium on which the data resides. For situations with restricted compute power or parameter update speed deadlines, the Clarkson-Woodruff sketch size can be used to control the local compute cost after retrieving the uniformly sketched data.

We define a doubly-sketching random matrix by $\mathbf{S} = \mathbf{S}_{\text{CW}}\mathbf{S}_{\text{Uniform}}$ where $\mathbf{S}_{\text{Uniform}}$ is a uniform sketch matrix and $\mathbf{S}_{\text{CW}}$ is a Clarkson-Woodruff sketch matrix. Indeed, the matrix product is also a sketch matrix; the doubly-sketching nature arises due to two distinct

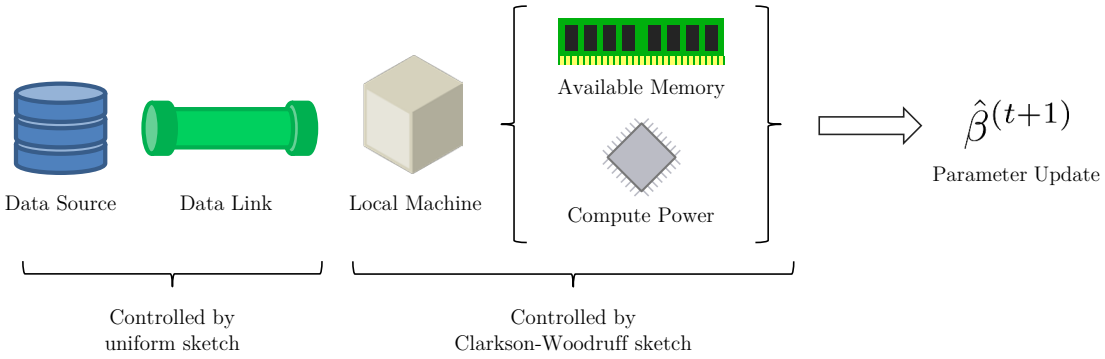

Figure 7.1: Stylized diagram depicting the flow of data from source to update for the IRLS procedure and where the respective sketches control computational speed. Data sources can be local to the machine on various storage mediums or on a remote machine; in either case, the data link represents the limitations on data transfer speed.

surrogate sketch size parameters. Let m be the Uniform sketch size and k be the Clarkson-Woodruff sketch size such that $k < m < n$. We draw rows of $\mathbf{S}_{\text{Uniform}} \in \{0, \sqrt{n/m}\}^{m \times n}$ by sampling with replacement from rows of $\sqrt{n/m}\mathbf{I}_{n \times n}$ where $\mathbf{I}$ is the identity matrix. Independently, we draw columns of $\mathbf{S}_{\text{CW}} \in \{-1, 0, 1\}^{k \times m}$ by sampling with replacement from columns of the columns of $-\mathbf{I}_{k \times k}$ and $\mathbf{I}_{k \times k}$. This has the effect of sequentially applying a Uniform sketch followed by a Clarkson-Woodruff sketch. In practice, the initial Uniform sketch can be realized by drawing and retrieving a random sample from the data storage medium. In sketches such as Clarkson-Woodruff or a randomized Hadamard transform, there is a need to scan over the entire dataset. The secondary Clarkson-Woodruff sketch is used for its simplicity of execution; it can be re-interpreted as adding/subtracting each observation to/from a random accumulator representing a sketched observation, which can be of benefit to computationally constrained devices. In situations where only the data retrieval or transfer speed is the bottleneck, we consider $m = k$ as a special case where the proposed method omits the second Clarkson-Woodruff sketch and reduces to Uniform sketching only.

Taking the Fisher Scoring update (7.2) and re-writing it with the substitutions $\mathbf{X}_W^{(t)} = \sqrt{\mathbf{W}^{(t)}}\mathbf{X}$ and $\boldsymbol{z}_W^{(t)} = \sqrt{\mathbf{W}^{(t)}}\boldsymbol{z}$, we obtain

$$\begin{aligned}\hat{\boldsymbol{\beta}}^{(t+1)} &= (\mathbf{X}^\top \mathbf{W}^{(t)} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}^{(t)} \boldsymbol{z}^{(t)} \\ &= \left(\mathbf{X}_W^{(t)\top} \mathbf{X}_W^{(t)} \right)^{-1} \mathbf{X}_W^{(t)\top} \boldsymbol{z}_W^{(t)} \end{aligned} \tag{7.4}$$

for which we propose an iterative stochastic approximation of the form

$$\hat{\boldsymbol{\beta}}^{(t+1)} = \hat{\boldsymbol{\beta}}^{(t)} + a_t \frac{\sum_{i=1}^t \tilde{h}^{(i)} \tilde{\mathbf{H}}^{(i)^{-1}}}{\sum_{i=1}^t \tilde{h}^{(i)}} \tilde{\boldsymbol{g}}^{(t)} \tag{7.5}$$

with approximate sketched Hessian matrices

$$\tilde{\mathbf{H}}^{(t)} = \mathbf{X}_W^{(t)\top} \mathbf{S}^{(t)\top} \mathbf{S}^{(t)} \mathbf{X}_W^{(t)} + \frac{\hat{z}}{\sqrt{t}} \mathbf{I}_{d \times d}$$

having determinant $\tilde{h}^{(t)}$, sketched gradient vector $\tilde{\boldsymbol{g}}^{(t)} = \mathbf{X}_W^{(t)\top} \mathbf{S}^{(t)\top} \mathbf{S}^{(t)} \boldsymbol{z}_W^{(t)}$, a damping factor $a_t = t^{-1}$, and a regularization constant $\hat{z} > 0$. We apply the determinantal averaging (Derezinski and Mahoney, 2019) estimate of the inverse Hessian here to mitigate both the inversion bias for matrix-valued random variates as well as the effect of a single poorly-conditioned sketched Hessian estimate $\tilde{\mathbf{H}}^{(t)}$. This yields an approximate iterative procedure for estimating $\hat{\boldsymbol{\beta}}_{\text{MLE}}$.

Intuitively, since the IRLS procedure can be viewed as alternating between updating the weights $\mathbf{W}^{(t)}$ and the regression coefficients $\boldsymbol{\beta}^{(t)}$, performing a partially informative update on $\boldsymbol{\beta}$ using a portion of the data can be less wasteful as further iterations are required regardless. Moreover, the sparse structure of the uniform sketch avoids computing w_i, μ_i, and η_i for observations not selected by the sketch. Finally, the Clarkson-Woodruff sketch controls the computational complexity of the update after retrieving the data by reducing the surrogate dataset further to a size $k < m$. This step can also be used to compensate for datasets where the uniform sketch size m must be large enough to capture rare events in categorical covariates.

From a practitioner's point of view, the choice of m and k tunes the performance characteristics of the doubly-sketched procedure based on the specific use-case and computational

infrastructure bottlenecks as illustrated in Figure 7.1. When accessing the data is very slow but matrix operations are relatively fast, we may take $k < m \ll n$; for example, a fairly fast local computer can be bottlenecked by a slow Internet connection to a remote database. Similarly, on many commodity personal computers, system memory is insufficient to retain the entire dataset, and so at any given time only a small fraction of data is visible to the system before more data must be retrieved from slower storage media, meriting a $m \ll n$ situation. Conversely, if accessing the data is fast but matrix operations are very slow, we make take $k \ll m < n$; a use-case may be a low-power system attached to fast storage. In cases where the cost of forming the Gram matrix estimate ${\mathbf{X}_W^{(t)}}^\top {\mathbf{S}^{(t)}}^\top \mathbf{S}^{(t)} \mathbf{X}_W^{(t)}$ with complexity $O(md^2)$ is acceptable, then the Clarkson-Woodruff sketch can be omitted, leaving only $\mathbf{S}_{\text{Uniform}}$ as a special case.

7.2.3 Theoretical Properties

In this section, we discuss some theoretical properties of the proposed methodology. We show in Section 7.2.3 some basic properties at the maximum likelihood estimate. In Section 7.2.3 we derive the moments of $\mathbf{S}^\top \mathbf{S}$, and show convergence behaviour in Section 7.2.3.

Asymptotic Properties

In this section, we discuss the behaviour of the proposed method near the maximum likelihood estimate as the number of iterations $t \to \infty$. We assume throughout that a canonical link function is used so that the log-likelihood objective (7.1) is strictly concave in $\boldsymbol{\beta}$ and a unique maximizer exists (Wedderburn, 1976; Haberman, 1977). While a non-canonical link function does not enjoy these properties in general, we include an example in simulation studies for completeness.

Let the maximum likelihood estimate (MLE) for a GLM with data $(\mathbf{X}, \boldsymbol{y})$ be denoted $\boldsymbol{\beta}_{\text{MLE}}$. At this MLE, we have that the score/gradient vector is zero, and so the update terms in Equation (7.2) are trivially zero. Thus, we are finding the root to the equation $(\mathbf{X}_W^\top \mathbf{X}_W)^{-1} \mathbf{X}_W^\top \boldsymbol{z}_W = \mathbf{0}$. Considering that the approximate update (7.5) is stochastic due to

the randomness of $\mathbf{S}$, we apply an algorithm reminiscent of Robbins-Monro; a convergence analysis is presented in Section 7.2.3. Robbins and Monro (1951) is a root-finding method for systems of the form $M(\theta) = \alpha$ where the M can only be observed stochastically as some random variable $N(\theta)$ such that $\mathrm{E}[N(\theta)] = M(\theta)$ and α is a constant. Blum (1954) generalizes the method to multivariate parameters $\boldsymbol{\theta}$. The iterative updates take the form

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - a_t(N(\boldsymbol{\theta}) - \alpha)$$

for a damping sequence $\{a_t\}_{t=1}^{\infty} \subset \mathbb{R}$ such that $\sum_{t=1}^{\infty} a_t = \infty$ and $\sum_{t=1}^{\infty} a_t^2 < \infty$. A common example of such a sequence is $a_t = t^{-1}$, which is assumed throughout the present work.

We first show that the update term in (7.5) converges in distribution for a fixed $\boldsymbol{\beta}$ to $\mathbf{H}^{-1}\tilde{\boldsymbol{g}}^{(t)}$ as $t \to \infty$. Let $\mathbf{X}_W$ and $\boldsymbol{z}_W$ be fixed by $\mathbf{W}$ computed using $\boldsymbol{\beta}$. By Theorem 1 of Derezinski and Mahoney (2019) and noting that $\frac{\hat{z}}{\sqrt{t}}\mathbf{I}_{d\times d} \to \mathbf{0}$ in $\tilde{\mathbf{H}}^{(t)}$ as suggested in Section 1.2 of the same work, the expression

$$\frac{\sum_{i=1}^{t} \tilde{h}^{(i)} \tilde{\mathbf{H}}^{(i)^{-1}}}{\sum_{i=1}^{t} \tilde{h}^{(i)}} \xrightarrow[t\to\infty]{\text{a.s.}} \mathbf{H}^{-1} = \left(\mathbf{X}_W^{\top}\mathbf{X}_W\right)^{-1}.$$

Trivially, $\tilde{\boldsymbol{g}}^{(t)} \to \tilde{\boldsymbol{g}}^{(t)}$ in distribution and so by Slutsky's theorem we have that their product demonstrates convergence in distribution of the form

$$\frac{\sum_{i=1}^{t} \tilde{h}^{(i)} \tilde{\mathbf{H}}^{(i)^{-1}}}{\sum_{i=1}^{t} \tilde{h}^{(i)}} \tilde{\boldsymbol{g}}^{(t)} \xrightarrow[t\to\infty]{d} \mathbf{H}^{-1}\tilde{\boldsymbol{g}}^{(t)} = (\mathbf{X}_W^{\top}\mathbf{X}_W)^{-1}\mathbf{X}_W^{\top}\mathbf{S}^{\top}\mathbf{S}\boldsymbol{z}_W.$$

Expected Values and Covariances

Since $\mathbf{S}$ is a matrix-valued random variate, we are interested in the moments of the derived expression $\mathbf{S}^{\top}\mathbf{S}$ due to its appearance in (7.5). We compute in the following lemmas the moments of elements of the matrix product $\mathbf{S}^{\top}\mathbf{S}$ for double-sketch matrix $\mathbf{S}$ as defined in Section 7.2 for later use in the convergence analysis.

Lemma 4 *Let* $\mathbf{R} = \mathbf{S}_{\mathrm{CW}}^{\top}\mathbf{S}_{\mathrm{CW}}$ *where* $\mathbf{S}_{CW}$ *is a Clarkson-Woodruff sketch matrix of dimension* $k \times m$*. For elements* r_{ab}, r_{cd} *of* $\mathbf{R}$ *such that* $a, b, c, d \in \{1, 2, ..., m\}$*, we have that*

$$\mathrm{E}[r_{ab}] = \mathbb{1}_{a=b}, \quad \mathrm{E}[r_{ab}r_{cd}] = \mathbb{1}_{a=b}\mathbb{1}_{c=d} + \frac{1}{k}\mathbb{1}_{a\neq b}(\mathbb{1}_{a=c}\mathbb{1}_{b=d} + \mathbb{1}_{a=d}\mathbb{1}_{b=c}).$$

Alternatively, the expectation of the products are non-zero for indices $a \neq b$ *taking values*

$$\mathrm{E}[r_{aa}r_{aa}] = 1, \quad \mathrm{E}[r_{aa}r_{bb}] = 1, \quad \mathrm{E}[r_{ab}r_{ab}] = \mathrm{E}[r_{ab}r_{ba}] = \frac{1}{k}.$$

Proof 6 *Element* r_{ab} *can be expressed as* $\boldsymbol{s}_a^{\top}\boldsymbol{s}_b$*, where* $\boldsymbol{s}$ *are the rows of* $\mathbf{S}_{CW}$*. Elements of* $\mathbf{S}_{CW}$ *are* $s_{km} = D_m\mathbb{1}_{m\to k}$ *where* D_m *are IID Rademacher random variables and* $\mathbb{1}_{m\to k}$ *is the assignment random variable for surrogate observation* k *sampling observation* m *such that exactly one of* $\mathbb{1}_{m\to k} = 1$ *for each* m*. We use throughout that* $D_m^2 = +1$ *and* $\mathbb{1}_{m\to k}^2 = \mathbb{1}_{m\to k}$ *and* D_m *is independent of* $\mathbb{1}_{m\to k}$*. Hence, we may expand the expectation*

$$\mathrm{E}[r_{ab}] = \mathrm{E}\left[D_aD_b\sum_k \mathbb{1}_{a\to k}\mathbb{1}_{b\to k}\right] = \mathrm{E}[D_aD_b]\,\mathrm{E}\left[\sum_k \mathbb{1}_{a\to k}\mathbb{1}_{b\to k}\right].$$

Suppose first that $a = b$ *so that*

$$\mathrm{E}[r_{aa}] = \mathrm{E}\left[D_a^2\sum_k \mathbb{1}_{a\to k}^2\right] = \mathrm{E}\left[\sum_k \mathbb{1}_{a\to k}\right] = 1.$$

Conversely, suppose $a \neq b$ *so that* D_a *is independent of* D_b*, yielding*

$$\mathrm{E}[r_{ab}] = \mathrm{E}[D_a]\,\mathrm{E}[D_b]\,\mathrm{E}\left[\sum_k \mathbb{1}_{a\to k}\mathbb{1}_{b\to k}\right] = 0 \times 0 \times \mathrm{E}\left[\sum_k \mathbb{1}_{a\to k}\mathbb{1}_{b\to k}\right] = 0.$$

For the expectation of the product, the expanded expression is

$$\mathrm{E}[r_{ab}r_{cd}] = \mathrm{E}\left[D_aD_bD_cD_d\sum_k \mathbb{1}_{a\to k}\mathbb{1}_{b\to k}\sum_k \mathbb{1}_{c\to k}\mathbb{1}_{d\to k}\right].$$

We partition the a, b, c, d *indices into three cases and consider them separately. First, suppose that* $a = b$ *and* $c = d$ *so that*

$$\mathrm{E}[r_{aa}r_{cc}] = \mathrm{E}\left[D_a^2D_c^2\sum_k \mathbb{1}_{a\to k}^2\sum_k \mathbb{1}_{c\to k}^2\right] = 1.$$

Secondly, suppose that either $(a,b)=(c,d)$ *or* $(a,b)=(d,c)$ *but* $a \neq b$ *so that*

$$\mathrm{E}[r_{ab}r_{ab}] = \mathrm{E}\left[D_a^2 D_b^2 \left(\sum_k \mathbb{1}_{a\to k}\mathbb{1}_{b\to k}\right)^2\right] = \mathrm{E}[r_{ab}r_{ba}].$$

Observe that the sum can only take values in $\{0,1\}$ *so that it is equal to its own square. By linearity of the expectation and independence of the assignments of observations* a *and* b*, we have*

$$\mathrm{E}[r_{ab}r_{ab}] = \sum_k \mathrm{E}[\mathbb{1}_{a\to k}]\,\mathrm{E}[\mathbb{1}_{b\to k}] = \sum_k \frac{1}{k^2} = \frac{1}{k}.$$

Finally, in all remaining cases, there is at least one index appearing exactly once among a,b,c,d*; without loss of generality, let this index be* a*. By independence, we have that*

$$\mathrm{E}[r_{ab}r_{cd}] = \mathrm{E}[D_a]\,\mathrm{E}\left[D_b D_c D_d \sum_k \mathbb{1}_{a\to k}\mathbb{1}_{b\to k}\sum_k \mathbb{1}_{c\to k}\mathbb{1}_{d\to k}\right] = 0.$$

Lemma 5 *Let* $\mathbf{Q}=\mathbf{S}^\top\mathbf{S}$ *for* $\mathbf{S}=\mathbf{S}_{\mathrm{CW}}\mathbf{S}_{\mathrm{Uniform}}$ *with uniform sketch size* m *and Clarkson-Woodruff sketch size* k*, and let* n *be the original number of observations. For elements* q_{st}, q_{uv} *of* $\mathbf{Q}$ *such that* $s,t,u,v \in \{1,2,...,n\}$*, we have that*

$$\begin{aligned}\mathrm{E}[q_{st}] &= \mathbb{1}_{s=t},\\ \mathrm{Cov}[q_{st},q_{uv}] &= \frac{m-1}{km}(\mathbb{1}_{s=u}\mathbb{1}_{t=v}+\mathbb{1}_{s=v}\mathbb{1}_{t=u}) + \frac{n}{m}\mathbb{1}_{s=t=u=v} - \frac{1}{m}\mathbb{1}_{s=t}\mathbb{1}_{u=v}.\end{aligned}$$

Equivalently, the expectation is the identity matrix and the non-zero indices of the covariance with $s \neq t$ *take the values*

$$\begin{aligned}\mathrm{Cov}[q_{ss},q_{ss}] &= \frac{k(n-1)+2(m-1)}{km},\\ \mathrm{Cov}[q_{ss},q_{tt}] &= -\frac{1}{m},\\ \mathrm{Cov}[q_{st},q_{st}] = \mathrm{Cov}[q_{st},q_{ts}] &= \frac{m-1}{km}.\end{aligned}$$

Proof 7 *We first re-write* $\mathbf{Q} = \mathbf{S}_{Uniform}^\top\mathbf{S}_{CW}^\top\mathbf{S}_{CW}\mathbf{S}_{Uniform} = \mathbf{S}_{Uniform}^\top\mathbf{R}\mathbf{S}_{Uniform}$ *and use Lemma 4. Let* $s_{ij} = \sqrt{\frac{n}{m}}\mathbb{1}_{j\to i}$ *denote the* $(i,j)^{th}$ *element of* $\mathbf{S}_{Uniform}$*, where* $\mathbb{1}_{ij}$ *is the indicator random variable for sampled observation* i *being observation* j*. Let indices* $a,b,c,d \in$

$\{1, 2, ..., m\}$ index elements of $\mathbf{R}$, and indices $s, t, u, v \in \{1, 2, ..., n\}$ index elements of $\mathbf{Q}$. We express elements of $\mathbf{Q}$ as $q_{st} = \sum_{a,b} s_{as} r_{ab} s_{bt}$ to compute expectations and covariances.

Consider the expectation

$$\begin{aligned}\mathrm{E}[q_{st}] &= \sum_{a,b} \mathrm{E}[s_{as} r_{ab} s_{bt}] \overset{s \perp r}{=} \sum_{a,b} \mathrm{E}[r_{ab}]\,\mathrm{E}[s_{as} s_{bt}] \overset{lem.}{=} \sum_{a,b} \mathbb{1}_{a=b}\,\mathrm{E}[s_{as} s_{bt}] \\ &= \sum_{a} \mathrm{E}[s_{as} s_{at}] = \frac{n}{m} \sum_{a} \mathrm{E}\left[\mathbb{1}_{as} \mathbb{1}_{at}\right]\end{aligned}$$

for cases $s = t$ and $s \neq t$. When $s = t$, the expectation simplifies to

$$\mathrm{E}[q_{ss}] = \frac{n}{m} \sum_{a} \mathrm{E}[\mathbb{1}_{as}^2] = \frac{n}{m} \sum_{a} \frac{1}{n} = 1.$$

Otherwise, when $s \neq t$ the product $\mathbb{1}_{as} \mathbb{1}_{at}$ is zero as observation a cannot be assigned to both sketched observation s and t simultaneously under uniform sketching, and so $\mathrm{E}[q_{st}] = 0$. Therefore, $\mathrm{E}[q_{st}] = \mathbb{1}_{s=t}$ for all s, t.

The covariance between two entries q_{st} and q_{uv} can be written as

$$\begin{aligned}\mathrm{Cov}[q_{st}, q_{uv}] &= \mathrm{Cov}\left[\sum_{a,b} s_{as} r_{ab} s_{bt}, \sum_{c,d} s_{cu} r_{cd} s_{dv}\right] \\ &= \sum_{a,b,c,d} \mathrm{Cov}\left(s_{as} r_{ab} s_{bt}, s_{cu} r_{cd} s_{dv}\right) \\ &= \sum_{a,b,c,d} \left(\mathrm{E}[s_{as} r_{ab} s_{bt} s_{cu} r_{cd} s_{dv}] - \mathrm{E}[s_{as} r_{ab} s_{bt}]\,\mathrm{E}[s_{cu} r_{cd} s_{dv}]\right) \\ &\overset{s \perp r}{=} \sum_{a,b,c,d} \left(\mathrm{E}[r_{ab} r_{cd}]\,\mathrm{E}[s_{as} s_{bt} s_{cu} s_{dv}] - \mathrm{E}[r_{ab}]\,\mathrm{E}[s_{as} s_{bt}]\,\mathrm{E}[r_{cd}]\,\mathrm{E}[s_{cu} s_{dv}]\right) \\ &\overset{lem.}{=} \sum_{a,b,c,d} \left(\mathrm{E}[r_{ab} r_{cd}]\,\mathrm{E}[s_{as} s_{bt} s_{cu} s_{dv}] - \mathbb{1}_{a=b}\,\mathrm{E}[s_{as} s_{bt}] \mathbb{1}_{c=d}\,\mathrm{E}[s_{cu} s_{dv}]\right).\end{aligned}$$

Observe that $\mathbb{1}_{a=b}\,\mathrm{E}[s_{as} s_{bt}] = \frac{n}{m} \mathbb{1}_{a=b}\,\mathrm{E}[\mathbb{1}_{as} \mathbb{1}_{at}]$ and that exactly one of $\mathbb{1}_{as}$ is one for any given a; hence, $\mathbb{1}_{as} \mathbb{1}_{at} = 1$ only if $s = t$, which occurs with probability $\frac{1}{n}$. Hence, $\mathbb{1}_{a=b}\,\mathrm{E}[s_{as} s_{bt}] = \frac{1}{m} \mathbb{1}_{a=b} \mathbb{1}_{s=t}$ with an equivalent result for $\mathbb{1}_{c=d}\,\mathrm{E}[s_{cu} s_{dv}]$ by symmetry. Thus,

$$\begin{aligned}\mathrm{Cov}[q_{st}, q_{uv}] &= \sum_{a,b,c,d} \left(\mathrm{E}[r_{ab} r_{cd}]\,\mathrm{E}[s_{as} s_{bt} s_{cu} s_{dv}] - \frac{1}{m^2} \mathbb{1}_{a=b} \mathbb{1}_{c=d} \mathbb{1}_{s=t} \mathbb{1}_{u=v}\right) \\ &= \sum_{a,b,c,d} \mathrm{E}[r_{ab} r_{cd}]\,\mathrm{E}[s_{as} s_{bt} s_{cu} s_{dv}] - \mathbb{1}_{s=t} \mathbb{1}_{u=v}.\end{aligned}$$

To simplify the summation over a, b, c, d, we treat four distinct cases as in the proof of Lemma 4. First, when $a = b = c = d$ we have that $\mathrm{E}[r_{aa}r_{aa}] \overset{lem.}{=} 1$, $\mathrm{E}[s_{as}s_{at}s_{au}s_{av}] = \frac{n}{m^2}\mathbb{1}_{s=t=u=v}$, *and*

$$\sum_{\substack{a,b,c,d \\ a=b=c=d}} \mathrm{E}[r_{ab}r_{cd}]\,\mathrm{E}[s_{as}s_{bt}s_{cu}s_{dv}] = \frac{n}{m^2}\mathbb{1}_{s=t=u=v}\sum_{a} 1 = \frac{n}{m}\mathbb{1}_{s=t=u=v}.$$

Second, when $(a, b) = (c, d)$ but $a \neq b$ we have that $\mathrm{E}[r_{ab}r_{ab}] \overset{lem.}{=} \frac{1}{k}$ *and* $\mathrm{E}[s_{as}s_{bt}s_{au}s_{bv}] \overset{ind.}{=} \mathrm{E}[s_{as}s_{au}]\,\mathrm{E}[s_{bt}s_{bv}] = \frac{1}{m^2}\mathbb{1}_{s=u}\mathbb{1}_{t=v}$ *so that*

$$\sum_{\substack{a,b,c,d \\ a=c,b=d,a\neq b}} \mathrm{E}[r_{ab}r_{cd}]\,\mathrm{E}[s_{as}s_{bt}s_{cu}s_{dv}] = \frac{1}{km^2}\mathbb{1}_{s=u}\mathbb{1}_{t=v}\sum_{\substack{a,b \\ a\neq b}} 1 = \frac{m-1}{km}\mathbb{1}_{s=u}\mathbb{1}_{t=v}.$$

By symmetry, when $(a, b) = (d, c)$ but $a \neq b$ we obtain $\frac{m-1}{km}\mathbb{1}_{s=v}\mathbb{1}_{t=u}$. Third, we consider $a = b$ and $c = d$ but $a \neq c$ so that $\mathrm{E}[r_{aa}r_{cc}] \overset{lem.}{=} 1$, $\mathrm{E}[s_{as}s_{at}s_{cu}s_{cv}] \overset{ind.}{=} \mathrm{E}[s_{as}s_{at}]\,\mathrm{E}[s_{cu}s_{cv}] = \frac{1}{m^2}\mathbb{1}_{s=t}\mathbb{1}_{u=v}$ *and*

$$\sum_{\substack{a,b,c,d \\ a=b,c=d,a\neq c}} \mathrm{E}[r_{ab}r_{cd}]\,\mathrm{E}[s_{as}s_{bt}s_{cu}s_{dv}] = \frac{1}{m^2}\mathbb{1}_{s=t}\mathbb{1}_{u=v}\sum_{\substack{a,c \\ a\neq c}} 1 = \frac{m-1}{m}\mathbb{1}_{s=t}\mathbb{1}_{u=v}.$$

Finally, the remaining cases features at least one index appearing exactly once among a, b, c, d; thus, $\mathrm{E}[r_{ab}r_{cd}] \overset{lem.}{=} 0$ *and so the entire sum is zero. We reconstitute the sum with the simplified expressions to obtain*

$$\begin{aligned}
\mathrm{Cov}[q_{st}, q_{uv}] &= \frac{m-1}{km}\mathbb{1}_{s=u}\mathbb{1}_{t=v} + \frac{m-1}{km}\mathbb{1}_{s=v}\mathbb{1}_{t=u} \\
&\quad + \frac{m-1}{m}\mathbb{1}_{s=t}\mathbb{1}_{u=v} + \frac{n}{m}\mathbb{1}_{s=t=u=v} - \mathbb{1}_{s=t}\mathbb{1}_{u=v} \\
&= \frac{m-1}{km}(\mathbb{1}_{s=u}\mathbb{1}_{t=v} + \mathbb{1}_{s=v}\mathbb{1}_{t=u}) + \frac{n}{m}\mathbb{1}_{s=t=u=v} - \frac{1}{m}\mathbb{1}_{s=t}\mathbb{1}_{u=v}.
\end{aligned}$$

Cases $\mathrm{Cov}[q_{ss}, q_{ss}]$, $\mathrm{Cov}[q_{st}, q_{st}]$, $\mathrm{Cov}[q_{st}, q_{ts}]$ and $\mathrm{Cov}[q_{ss}, q_{tt}]$ for indices $s \neq t$ can be obtained by evaluating the indicator functions at the desired values.

Proposition 1 *Let $\boldsymbol{\beta} \in \mathbb{R}^p$ be a parameter vector and let $\mathbf{W}$ be the corresponding diagonal weight matrix. Let $\mathbf{X}_W = \sqrt{\mathbf{W}}\mathbf{X}$ and $\boldsymbol{z}_W = \sqrt{\mathbf{W}}\boldsymbol{z}$, and so that the gradient of the log-likelihood is $\boldsymbol{g}(\boldsymbol{\beta}) = \mathbf{X}_W^\top \boldsymbol{z}_W$. If the doubly-sketched gradient is $\tilde{\boldsymbol{g}}(\boldsymbol{\beta}) = \mathbf{X}_W^\top \mathbf{S}_{Uniform}^\top \mathbf{S}_{CW}^\top \mathbf{S}_{CW} \mathbf{S}_{Uniform} \boldsymbol{z}_W$ with uniform sketch size m and Clarkson-Woodruff sketch size k, then*

$$\mathrm{E}\left[\|\boldsymbol{g}(\boldsymbol{\beta}) - \tilde{\boldsymbol{g}}(\boldsymbol{\beta})\|_2^2\right] = \frac{n}{m}\|\mathbf{X}^\top \operatorname{diag} \boldsymbol{z}\|_F^2 + \frac{m-k-1}{km}\|\mathbf{X}^\top \boldsymbol{z}\|_2^2 + \frac{m-1}{km}\|\mathbf{X}\|_F^2 \|\boldsymbol{z}\|_2^2$$

where $\|\cdot\|_F$ is the Frobenius norm, and $\operatorname{diag}(\cdot)$ composes a diagonal matrix with the vector-valued argument as the main diagonal.

Proof 8 *To reduce notational burden, we drop the argument $\boldsymbol{\beta}$ and the W subscript on $\mathbf{X}$ and $\boldsymbol{z}$ in this proof and define $\mathbf{Q} = \mathbf{S}_{Uniform}^\top \mathbf{S}_{CW}^\top \mathbf{S}_{CW} \mathbf{S}_{Uniform}$ to apply Lemma 5. Let $s, t, u, v \in \{1, 2, ..., n\}$ and $\boldsymbol{x}_n$ be the n^{th} row of $\mathbf{X}$ so that*

$$\begin{aligned}
\mathrm{E}\left[\|\boldsymbol{g} - \tilde{\boldsymbol{g}}\|_2^2\right] &= \sum_{s,t,u,v} \mathrm{Cov}[q_{st}, q_{uv}] z_t z_v \boldsymbol{x}_s \boldsymbol{x}_u^\top \\
&= \sum_s \mathrm{Cov}[q_{ss}, q_{ss}] z_s^2 \boldsymbol{x}_s \boldsymbol{x}_s^\top + \sum_{\substack{s,t \\ s \neq t}} \mathrm{Cov}[q_{ss}, q_{tt}] z_s z_t \boldsymbol{x}_s \boldsymbol{x}_t^\top \\
&\quad + \sum_{\substack{s,t \\ s \neq t}} \mathrm{Cov}[q_{st}, q_{st}] z_s z_t \boldsymbol{x}_s \boldsymbol{x}_t^\top + \sum_{\substack{s,t \\ s \neq t}} \mathrm{Cov}[q_{st}, q_{ts}] z_s^2 \boldsymbol{x}_t \boldsymbol{x}_t^\top
\end{aligned}$$

We observe that the covariances are constant within each summation's indices and can be taken out of the summation to obtain

$$\begin{aligned}
\mathrm{E}\left[\|\boldsymbol{g} - \tilde{\boldsymbol{g}}\|_2^2\right] &= \mathrm{Cov}[q_{11}, q_{11}] \sum_s z_s^2 \boldsymbol{x}_s \boldsymbol{x}_s^\top + \mathrm{Cov}[q_{11}, q_{22}] \sum_{\substack{s,t \\ s \neq t}} z_s z_t \boldsymbol{x}_s \boldsymbol{x}_t^\top \\
&\quad + \mathrm{Cov}[q_{12}, q_{12}] \sum_{\substack{s,t \\ s \neq t}} z_s z_t \boldsymbol{x}_s \boldsymbol{x}_t^\top + \mathrm{Cov}[q_{12}, q_{21}] \sum_{\substack{s,t \\ s \neq t}} z_s^2 \boldsymbol{x}_t \boldsymbol{x}_t^\top
\end{aligned}$$

whereupon we perform the substitutions

$$\begin{aligned}\sum_s z_s^2 \boldsymbol{x}_s \boldsymbol{x}_s^\top &= \|\mathbf{X}^\top \operatorname{diag} \boldsymbol{z}\|_F^2 \\ \sum_{\substack{s,t \\ s \neq t}} z_s z_t \boldsymbol{x}_s \boldsymbol{x}_t^\top &= \|\mathbf{X}^\top \boldsymbol{z}\|_2^2 - \|\mathbf{X}^\top \operatorname{diag} \boldsymbol{z}\|_F^2 \\ \sum_{\substack{s,t \\ s \neq t}} z_s^2 \boldsymbol{x}_t \boldsymbol{x}_t^\top &= \|\mathbf{X}\|_F^2 \|\boldsymbol{z}\|_2^2 - \|\mathbf{X}^\top \operatorname{diag} \boldsymbol{z}\|_F^2\end{aligned}$$

to obtain the simplification

$$\begin{aligned}\mathrm{E}\left[\|\boldsymbol{g} - \tilde{\boldsymbol{g}}\|_2^2\right] &= (\operatorname{Cov}[q_{11}, q_{11}] - \operatorname{Cov}[q_{12}, q_{12}] - \operatorname{Cov}[q_{12}, q_{21}] - \operatorname{Cov}[q_{11}, q_{22}])\|\mathbf{X}^\top \operatorname{diag} \boldsymbol{z}\|_F^2 \\ &\quad + (\operatorname{Cov}[q_{12}, q_{12}] + \operatorname{Cov}[q_{11}, q_{22}])\|\mathbf{X}^\top \boldsymbol{z}\|_2^2 + \operatorname{Cov}[q_{12}, q_{21}]\|\mathbf{X}\|_F^2 \|\boldsymbol{z}\|_2^2 \\ &\overset{lem.}{=} \frac{n}{m}\|\mathbf{X}^\top \operatorname{diag} \boldsymbol{z}\|_F^2 + \frac{m-k-1}{km}\|\mathbf{X}^\top \boldsymbol{z}\|_2^2 + \frac{m-1}{km}\|\mathbf{X}\|_F^2 \|\boldsymbol{z}\|_2^2.\end{aligned}$$

Convergence

We now show convergence of the algorithm for a GLM with canonical link to the global optimum as the number of iterations t goes to infinity. Let $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \cdots$ be a filtration on the probability space, where the randomness at each iteration t is in the sketch matrix $\mathbf{S}^{(t)}$. In this section, we parameterize the problem as one of minimization to be consistent with the optimization literature; i.e., we define the objective function $f(\boldsymbol{\beta}) = -\ell(\boldsymbol{\beta})$ and seek its minima.

For convenience, we let $\mathbf{A}^{(t)} = \frac{\sum_{i=1}^t \tilde{h}^{(i)} \tilde{\mathbf{H}}^{(i)^{-1}}}{\sum_{i=1}^t \tilde{h}^{(i)}}$ and observe that it is a constant symmetric positive-definite matrix given $\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_t$. Thus, the update (7.5) can be written as

$$\hat{\boldsymbol{\beta}}^{(t+1)} = \hat{\boldsymbol{\beta}}^{(t)} - a_t(\mathbf{A}^{(t-1)} \boldsymbol{g}^{(t)} + \boldsymbol{\varepsilon}^{(t+1)})$$

where $\boldsymbol{\varepsilon}^{(t)} = \mathbf{A}^{(t-1)}({\mathbf{X}_W^{(t-1)}}^\top {\mathbf{S}^{(t)}}^\top \mathbf{S}^{(t)} \boldsymbol{z}_W^{(t-1)} - \boldsymbol{g}^{(t-1)})$ is a $\mathcal{F}_t$-measurable random variable with $\mathrm{E}[\boldsymbol{\varepsilon}^{(t)} \mid \mathcal{F}_{t-1}] = \mathbf{0}$ and $\mathrm{E}[\|\boldsymbol{\varepsilon}^{(t)}\|^2 \mid \mathcal{F}_{t-1}] = \sigma^{2(t)} < \infty$.

We define for $\boldsymbol{h}^{(t)}(\boldsymbol{\beta}) = \mathbf{A}^{(t-1)} \boldsymbol{g}^{(t)}$ an associated Lyapunov function $V(\boldsymbol{\beta}) = f(\boldsymbol{\beta}) - \inf_\beta f(\boldsymbol{\beta})$. We proceed to examine some properties of V. By definition, V is non-negative

and is strictly convex as the associated GLM log-likelihood is strictly concave, giving V a unique minima $\boldsymbol{\beta}_{\mathrm{ML}}$. We assume the trajectories $\{\boldsymbol{\beta}^{(t)}\}_{t=1}^{\infty}$ lie within some domain $B = \{\delta \leq \|\boldsymbol{\beta} - \boldsymbol{\beta}_{\mathrm{ML}}\| \leq \delta^{-1}\} \subset \mathbb{R}^p$ for some $\delta \in (0, 1)$. Under this assumption, we have that a variety of common GLMs satisfy V being continuously differentiable and having L-Lipschitz-continuous gradients on B. For GLMs with canonical link functions, this latter condition can be simplified to

$$\begin{aligned}
\|V(\boldsymbol{\beta}) - V(\boldsymbol{\alpha})\| &\overset{\text{def.}}{=} \left\| -\frac{1}{a(\phi)} \sum_{i=1}^{n} (b'(\boldsymbol{x}_i^\top \boldsymbol{\beta}) - b'(\boldsymbol{x}_i^\top \boldsymbol{\alpha}))\boldsymbol{x}_i \right\| \\
&\leq \text{Constant} \times \max_i \|\boldsymbol{x}_i\| \sum_{i=1}^{n} |b'(\boldsymbol{x}_i^\top \boldsymbol{\beta}) - b'(\boldsymbol{x}_i^\top \boldsymbol{\alpha})| \\
&= \text{Constant} \times \sum_{i=1}^{n} |b'(\boldsymbol{x}_i^\top \boldsymbol{\beta}) - b'(\boldsymbol{x}_i^\top \boldsymbol{\alpha})| \\
&\leq L\|\boldsymbol{\beta} - \boldsymbol{\alpha}\|.
\end{aligned}$$

Hence, it suffices to verify that $b'(\boldsymbol{x}^\top \boldsymbol{\beta})$ is L-Lipschitz for $\boldsymbol{\beta} \in B$. For Binomial-Logit models, we have that $b''(\theta) \leq 0.25$. For Poisson-Log models, we have that $b''(\boldsymbol{x}^\top \boldsymbol{\beta}) = \exp(\theta)$ is bounded for $\boldsymbol{\beta} \in B$.

In terms of the search direction, we have that there exists $\lambda^{(t)} > 0$ such that $\boldsymbol{h}^{(t)}(\boldsymbol{\beta})^\top V'(\boldsymbol{\beta}) \geq \lambda^{(t)} \|V'(\boldsymbol{\beta})\|^2$. As $\mathbf{A}^{(t-1)}$ is symmetric positive-definite, we have that

$$\frac{\boldsymbol{h}^{(t)}(\boldsymbol{\beta})^\top V'(\boldsymbol{\beta})}{\|V'(\boldsymbol{\beta})\|^2} = \frac{\boldsymbol{g}(\boldsymbol{\beta})^\top \mathbf{A}^{(t-1)} \boldsymbol{g}(\boldsymbol{\beta})}{\boldsymbol{g}(\boldsymbol{\beta})^\top \boldsymbol{g}(\boldsymbol{\beta})} \geq \lambda_{\min}(\mathbf{A}^{(t-1)}) \overset{\text{pos.def.}}{>} 0.$$

Finally, we examine the control of $\boldsymbol{h}^{(t)}$ by checking $\|\boldsymbol{h}^{(t)}(\boldsymbol{\beta})\|^2 \leq C^{(t)}(1 + V(\boldsymbol{\beta}))$ for some

$C^{(t)} > 0$ and for all $\boldsymbol{\beta} \in B$. We rearrange

$$\begin{aligned}
\|\boldsymbol{h}^{(t)}(\boldsymbol{\beta})\|^2 &= \|\mathbf{A}^{(t-1)}\boldsymbol{g}(\boldsymbol{\beta})\|^2 \\
&\leq \|\mathbf{A}^{(t-1)}\|^2 \|\boldsymbol{g}(\boldsymbol{\beta})\|^2 \\
&= \text{Constant} \times \left\| \frac{1}{a(\phi)} \sum_{i=1}^{n} (y_i - b'(\boldsymbol{x}_i^\top \boldsymbol{\beta}))\boldsymbol{x}_i \right\|^2 \\
&\leq \text{Constant} \times \max_i \|\boldsymbol{x}_i\|^2 \sum_{i=1}^{n} |y_i - b'(\boldsymbol{x}_i^\top \boldsymbol{\beta})|^2 \\
&= \text{Constant} \times \sum_{i=1}^{n} (y_i - b'(\boldsymbol{x}_i^\top \boldsymbol{\beta}))^2 \\
&\leq C^{(t)}(1 + V(\boldsymbol{\beta})).
\end{aligned}$$

For the Binomial-Logit model, we have that $|y_i - b'(\boldsymbol{x}_i^\top \boldsymbol{\beta})|^2 \leq 1$ for all i so that it suffices to take $C^{(t)} \propto n$. For Poisson-Log, we inspect the magnitudes of the gradients of the left and right-hand sides of the inequality with the fact that $a(\phi) = 1$ and $b = b' = b'' = \exp$ to obtain

$$\text{Constant} \times \left\| 2 \times \sum_{i=1}^{n} (y_i - \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta}))\boldsymbol{x}_i \right\| \leq C^{(t)} \times \left\| 2 \times \sum_{i=1}^{n} (y_i - \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta}))\boldsymbol{x}_i \right\|$$

and notice that $C^{(t)} = 2/\text{Constant}$ suffices.

Applying the definition and properties of V on the domain B, we obtain

$$\begin{aligned}
V(\boldsymbol{\beta}^{(t+1)}) &\leq V(\boldsymbol{\beta}^{(t)}) + V'(\boldsymbol{\beta}^{(t)})^\top (\boldsymbol{\beta}^{(t+1)} - \boldsymbol{\beta}^{(t)}) + \frac{L}{2} \|\boldsymbol{\beta}^{(t+1)} - \boldsymbol{\beta}^{(t)}\|^2 \\
&= V(\boldsymbol{\beta}^{(t)}) - a^{(t)} {\boldsymbol{g}^{(t)}}^\top \mathbf{A}^{(t-1)} {\mathbf{X}_W^{(t)}}^\top {\mathbf{S}^{(t+1)}}^\top \mathbf{S}^{(t+1)} \boldsymbol{z}_W^{(t)} \\
&\quad + \frac{L}{2} {a^{(t)}}^2 \|{\mathbf{X}_W^{(t)}}^\top {\mathbf{S}^{(t+1)}}^\top \mathbf{S}^{(t+1)} \boldsymbol{z}_W^{(t)}\|^2.
\end{aligned}$$

Taking expectations conditional on $\mathcal{F}_t$ we have that

$$\mathrm{E}[V(\boldsymbol{\beta}^{(t+1)}) \mid \mathcal{F}_t] \leq V(\boldsymbol{\beta}^{(t)}) - a^{(t)} {\boldsymbol{g}^{(t)}}^\top \mathbf{A}^{(t-1)} \boldsymbol{g}^{(t)} + \frac{L}{2} {a^{(t)}}^2 {\sigma^2}^{(t)}$$

is a positive super-martingale for which we apply the theorem of Robbins and Siegmund (1971) to obtain convergence. The theorem requires that $\sum_{t=1}^{\infty} \frac{L}{2} {a^{(t)}}^2 {\sigma^2}^{(t)} < \infty$ almost

surely. Indeed, Proposition 1 gives us that $\sigma^{2(t)} \leq \|\mathbf{A}^{(t-1)}\|_F^2 \, \mathrm{E}\left[\|\boldsymbol{g}(\boldsymbol{\beta}) - \tilde{\boldsymbol{g}}(\boldsymbol{\beta})\|_2^2\right]$. For $\boldsymbol{\beta} \in B$, the weight matrix $\mathbf{W}$ is bounded and hence so are the norms found in Proposition 1. Moreover, $\mathbf{A}^{(t-1)}$ is symmetric positive-definite with bounded norm on B by construction.

Thus, $\lim_{t\to\infty} V(\boldsymbol{\beta}^{(t)})$ exists and is finite and that $\sum_{t=1}^{\infty} a^{(t)} \boldsymbol{g}^{(t)\top} \mathbf{A}^{(t-1)} \boldsymbol{g}^{(t)} < \infty$ almost surely. Since $\sum_{t=1}^{\infty} a^{(t)} = \infty$ by assumption, we have that $\boldsymbol{g}^{(t)\top} \mathbf{A}^{(t-1)} \boldsymbol{g}^{(t)} \xrightarrow[t\uparrow\infty]{\text{a.s.}} 0$ with positive-definiteness of $\mathbf{A}$ implying $\boldsymbol{g}^{(t)} \xrightarrow[t\uparrow\infty]{\text{a.s.}} 0$. The strict convexity of f for a GLM with canonical link gives that $\boldsymbol{\beta}^{(t)} \xrightarrow[t\uparrow\infty]{\text{a.s.}} \boldsymbol{\beta}_{\text{ML}}$, as required.

7.2.4 Standard Errors

Considering the update (7.5), we note that initially the sketched Hessians represent the curvature of the log-likelihood away from $\boldsymbol{\beta}_{\text{MLE}}$ and so are detrimental for the estimate in practice. This can slow convergence at early iterations by contaminating the inverse Hessian estimate with convexity from distant prior parameters. We can expedite convergence by periodically resetting the memory of the summations in the determinantal averaging step; i.e., we would substitute the sums at iteration (t)

$$\sum_{i=1}^{t} \tilde{h}^{(i)} \tilde{\mathbf{H}}^{(i)^{-1}} \to \sum_{i=b_t}^{t} \tilde{h}^{(i)} \tilde{\mathbf{H}}^{(i)^{-1}} \text{ and}$$
$$\sum_{i=1}^{t} \tilde{h}^{(i)} \to \sum_{i=b_t}^{t} \tilde{h}^{(i)}$$

for some integer sequence $b_t \leq t$. An equally asymptotic choice would be $b_t = \max_{j\in\mathbb{N}}(\sum_{i=1}^{j} i \mid \sum_{i=1}^{j} i < t)$; in other words, the interval between resets increases by one after each successive reset. A cruder but simpler choice could be $b'_t = 100 \times \lfloor t/100 \rfloor$; a reset every 100 iterations. Going forward, we opt to use the former choice as it provides much needed flexibility at early iterations.

We also note that

$$\widehat{\mathbf{H}^{-1}} = \frac{\sum_{i=b_t}^{t} \tilde{h}^{(i)} \tilde{\mathbf{H}}^{(i)^{-1}}}{\sum_{i=b_t}^{t} \tilde{h}^{(i)}}$$

is an asymptotically consistent estimator for the inverse Hessian when $\hat{\boldsymbol{\beta}}$ is $\boldsymbol{\beta}_{\text{MLE}}$, which is also an appropriate estimator for the covariance matrix $\text{Var}[\boldsymbol{\beta}_{\text{MLE}}]$. This simultaneously yields an estimate for the GLM standard errors. Given the issue of resetting at iteration b_t, to obtain a reasonable approximation of the standard errors, the stopping iteration should immediately precede a reset b_t to maximize the accuracy of determinantal averaging. Explicitly, b_t takes values in the sequence of triangular numbers, whose values near the first few powers of ten are

$$\{b_t\}_{t=1}^{\infty} \subset \{1, 3, 6, 10, ..., 990, 1035, ..., 9870, 10011, ..., 99681, 100128, ...\}. \tag{7.6}$$

For GLM families with a dispersion parameter, such as the Gamma family, we introduce an additional dispersion estimation sub-step using the Uniform sketch intermediary at each iteration. In particular, we use the dispersion estimate

$$d = \sum_{i=1}^{n} w_i \left(\frac{y_i - \mu_i}{\partial \mu_i / \partial \eta_i} \right)^2$$

to be consistent with the `glm()` implementation in R (R Core Team, 2019). By defining the residuals vector $\boldsymbol{r} = \boldsymbol{z}$ with elements $r_i = \frac{y_i - \mu_i}{\partial \mu_i / \partial \eta_i} = z_i$ and the weight matrix $\mathbf{W}$, we may re-write this dispersion expression in matricial form as

$$d = \boldsymbol{r}^\top \mathbf{W} \mathbf{r} = (\sqrt{\mathbf{W}}\boldsymbol{r})^\top (\sqrt{\mathbf{W}}\boldsymbol{r}) = \boldsymbol{z}_W^\top \boldsymbol{z}_W.$$

This expression immediately admits the use of sketching as well. Hence, we compute at each iteration

$$d^{(t+1)} = d^{(t)} + a_t(\boldsymbol{z}_W^\top \mathbf{S}^\top \mathbf{S} \boldsymbol{z}_W - d^{(t)})$$

to produce an estimate of the dispersion, which will be used to estimate the standard errors $\text{se}(\boldsymbol{\beta}_{\text{MLE}})$ for GLM families that require it.

7.2.5 Initialization and Convergence

While the crux of the proposed method is in the estimation procedure, it is book-ended by a need to initialize $\hat{\boldsymbol{\beta}}^{(0)}$ and a need to assess convergence as the iteration t increases. For

initialization, a straight-forward method would be to draw a small pilot sample of size m and use the coefficients from a fitted GLM model using IRLS as the starting value. To avoid conflating the estimation behaviour with the initialization scheme, we opt to initialize $\hat{\boldsymbol{\beta}}^{(0)} = \mathbf{0}$ throughout the remainder of the present work.

When assessing convergence of the proposed method, we may either assess convergence of the log-likelihood objective (7.1) or convergence of the parameter estimates $\hat{\boldsymbol{\beta}}^{(t)}$. We note that calculating the log-likelihood over all n observations requires transferring the entire dataset; defeating the efficiency gained from the proposed method. Hence, we opt to assess convergence in the parameter estimates. As such, we may specify a tolerance ε_{tol} where we may consider the procedure converged if $\|\hat{\boldsymbol{\beta}}^{(t+1)}_{\text{Sketch}} - \hat{\boldsymbol{\beta}}^{(t)}_{\text{Sketch}}\|_1 < \varepsilon_{\text{tol}}$, though we may opt to continue if t is near the previous reset b_t to ensure a good estimate for the standard error as per Section 7.2.4. By the stochastic nature of the iterative update, it is possible that the convergence condition is spuriously satisfied; hence, an alternate convergence condition is impose a threshold on the average change in parameter values over some number of iterations.

In the following simulation and real-world dataset sections, we take the rudimentary approach of specifying an exact number of iterations $t_{\max}$ as is common for stochastic methods such as simulated annealing. This value will generally be a element of the sequence $\{b_t\}_{t=1}^{\infty}$. A crude rule-of-thumb guide for choosing this can be a value of $t_{\max}$ that transfers approximately n observations; $t_{\max} \approx \lceil n/m \rceil$, representing the intuition that the entire dataset has a high probability of participating in the procedure. Moreover, by retaining the determinantal averaging values $\sum_{i=b_t}^{t} \tilde{h}^{(i)}$ and $\sum_{i=b_t}^{t} \tilde{h}^{(i)} \tilde{\mathbf{H}}^{(i)^{-1}}$, we may resume the doubly-sketching procedure should a desired convergence criteria at originally chosen $t_{\max}$ not be met by simply drawing additional sketches and continuing the updates.

7.3 Simulation Study

In this section, we study the behaviour of the proposed method *in silico* across a variety of simulated datasets. We investigate both the parameter recovery aspect as well as the

practical computational efficiency in two separate simulation studies.

7.3.1 Comparison to IRLS

We test the parameter recovery of the proposed doubly-sketched method in a full-factorial simulation design over the following configurations:

- Sample size: $n \in \{1 \times 10^5, 4 \times 10^5, 7 \times 10^5, 1 \times 10^6\}$
- Uniform sketch size: $m \in \{1000, 10000\}$
- Clarkson-Woodruff sketch size: $k \in \{100, 500, 1000, 5000\}$
- Covariate Dimension: $d \in 5, 10, 50, 100$
- Covariate Distribution: $\boldsymbol{x}_i \sim N(0, \frac{1}{d}), \frac{1}{d}t_{10}, \text{Uniform}(-\frac{1}{d}, \frac{1}{d})$
- Response-Link: Binomial-Logit, Poisson-Log, Binomial-cloglog
- True Parameters: $\beta_i = (-1)^{i+1}$

We exclude configurations where $k \geq m$ as it is non-sensical to take a larger sketch after a smaller sketch. To evaluate the asymptotic behaviour of $\hat{\boldsymbol{\beta}}_{\text{Sketch}}$, we run a large number of iterations $t_{\max} = 10011$ chosen as per (7.6). We perform 10 replications at each configuration, generating a new dataset each time, and also run the IRLS algorithm with convergence condition $\|\hat{\boldsymbol{\beta}}_{\text{IRLS}}^{(t+1)} - \hat{\boldsymbol{\beta}}_{\text{IRLS}}^{(t)}\|_1 \leq 10^{-10}$. Finally, we compare the estimated $\hat{\boldsymbol{\beta}}_{\text{Sketch}}$ and $\hat{\boldsymbol{\beta}}_{\text{IRLS}}$ against each other in addition to the true $\boldsymbol{\beta}$ in terms of mean square error. For vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$, we denote their mean square error to be $\text{MSE}(\boldsymbol{a}, \boldsymbol{b}) = \sum_{i=1}^{d}(a_i - b_i)^2$.

Figure 7.2 depicts the ratio of mean square errors $\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{Sketch}}, \boldsymbol{\beta})/\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{IRLS}}, \boldsymbol{\beta})$ over a selected subset of simulation configurations with $d = 100$, $m = 10000$, and binomial response with logit link. We recognize that the situation $d = k = 100$ represents a boundary case where the Hessian $\mathbf{H}^{(t)}$ is likely rank-deficient at each iteration if the ridge regularization term $\frac{\hat{z}}{\sqrt{t}}\mathbf{I}_{d\times d}$ is omitted. Indeed, Figure 7.2 shows poor finite iteration behaviour at

$k = 100$ even though convergence is provided by Section 7.2.3. As coverage of the dataset by a surrogate sketch decreases from $m/n = 0.1$ to 0.01, we find $\hat{\boldsymbol{\beta}}_{\text{Sketch}}$ deviates from the true parameter values more so than $\hat{\boldsymbol{\beta}}_{\text{IRLS}}$. As k increases beyond 100, the behaviour significantly improves and the parameter recovery capabilities of $\hat{\boldsymbol{\beta}}_{\text{Sketch}}$ increasingly coincides with that of $\hat{\boldsymbol{\beta}}_{\text{IRLS}}$. The covariate distribution appears to have a small impact when $k > 100$; when $k = 100$, the doubly-sketched estimator attains the best median MSE ratio for uniformly distributed $\mathbf{X}$.

The error between the doubly-sketched and IRLS estimates are explored in Figure 7.3. We again see poor performance at $d = k = 100$, though the effect of dataset size n diminishes greatly. This suggests that the choice of m and k is not dramatically affected by the size of the entire dataset. The effect of the covariate distribution appears to be reversed when $\hat{\boldsymbol{\beta}}_{\text{Sketch}}$ is compared against $\hat{\boldsymbol{\beta}}_{\text{IRLS}}$, with the uniformly distributed $\mathbf{X}$ performing the worst among the three.

7.3.2 Wall-Clock Time and Storage Medium

In this section we examine the performance of the proposed method and IRLS in terms of wall-clock time. A single simulated dataset comprising 10^8 observations with 10^2 covariates is used throughout; this dataset stored as a CSV takes approximately 10 GiB of space. For simplicity, the covariates are $N(0, 1)$ and the response variable is binomial with logistic link with true regression coefficients $\beta_i = (-1)^{i+1}$. To emulate real-world scenarios, we run the proposed doubly-sketched algorithm and IRLS on four different data infrastructures described briefly below.

1. Fully in-memory (RAM): The entire dataset is stored in memory, representing the ideal case where the practitioner has sufficient memory for the analysis.

2. Solid-State Drive (SSD): The dataset is stored in a SQLite database on a solid-state drive, selected rows are loaded as needed. This represents a dataset too large for system memory.

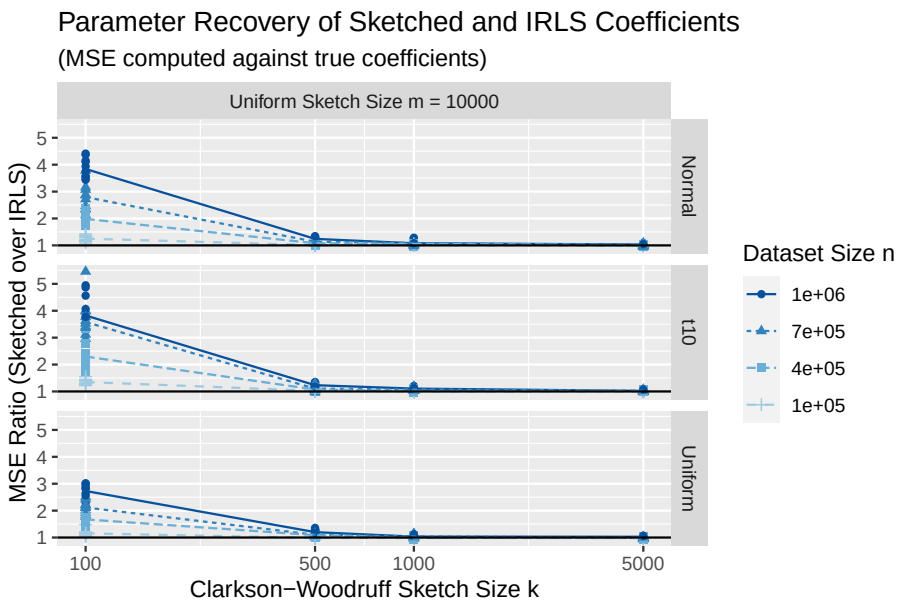

Figure 7.2: Ratio of coefficient parameter MSEs of the doubly-sketched and IRLS estimation procedure against the true parameters, with line segments joining group-wise medians. The doubly-sketched estimate approaches the quality of the IRLS estimate as k increases and uniformly distributed $\mathbf{X}$ behaving better than normally and t_{10} distributed $\mathbf{X}$.

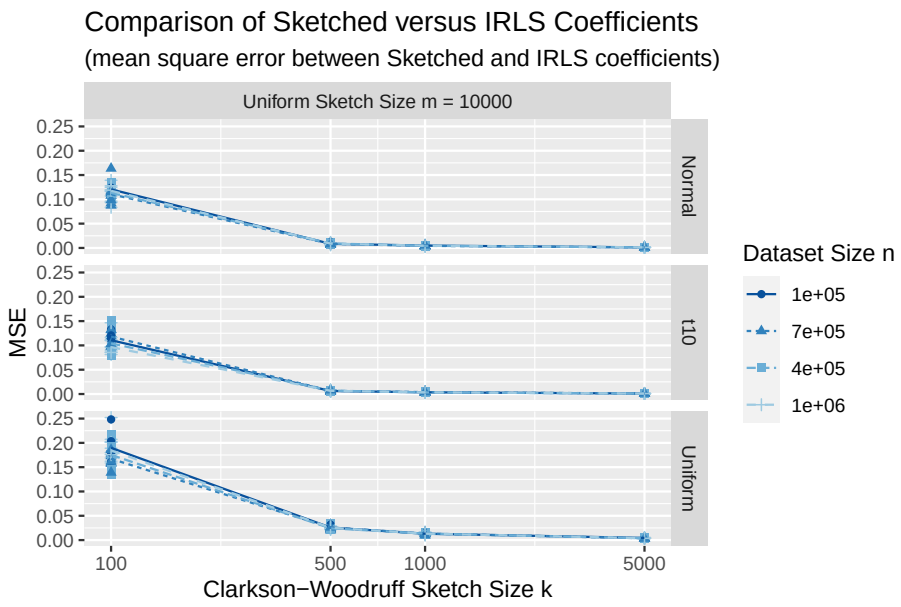

Figure 7.3: Coefficient parameter MSEs of the doubly-sketched coefficients versus the IRLS coefficients, with line segments joining group-wise medians. The doubly-sketched estimate's recovery of the IRLS coefficients appear robust to changes in dataset sizes n for a given m and k.

3. Hard Drive (HDD): The dataset is stored in a SQLite database on a mechanical hard drive as in the SSD case. This represents a dataset too large for system memory, but also restricted to slower and lower-cost storage media.

4. Network Storage (NET): The dataset is stored in a PostgreSQL database on a server across a local network. This represents a case where the data is not available locally but stored on another machine, possibly shared among multiple practitioners.

We assume here that $m = 1000$ and $k = 500$; for further examination of the sketch sizes, see Section 7.4.1. Due to the sparse nature of the Uniform sketch, we only request the rows sampled by $\mathbf{S}_{\text{Uniform}}$ at each iteration of the doubly-sketched procedure from the storage device; this is where we expect to see wall-clock time savings due to the cost of data transfer.

Figure 7.4 demonstrates the performance of the proposed doubly-sketched method compared to IRLS on these four storage infrastructures from iteration to iteration. For the faster MEM and SSD storage media we choose $t_{\max} = 10011$, for the slower HDD and NET we choose a smaller $t_{\max} = 1035$. We observe that as the iteration t increases, the sketched estimate $\hat{\boldsymbol{\beta}}^{(t)}_{\text{Sketch}}$ shows convergence behaviour as expected from Section 7.2.3. Moreover, the curves show potential wall-clock time savings by trading-off some accuracy for speed. We also note the effect of sketching on NET is more pronounced than that of sketching on HDD. From a technical standpoint, we explicitly flush the operating system's cache in the SSD and HDD cases as there was sufficient system memory to cache the entire SQLite database in-memory; defeating the measurement. This was not done in the NET case as not enough memory was allocated to the database for the same phenomenon to occur. Further on a technical level, we note that transferring the data in a contiguous manner is more efficient than retrieving fragmented observations. This effect becomes more pronounced going from memory to solid state disks to mechanical hard drives.

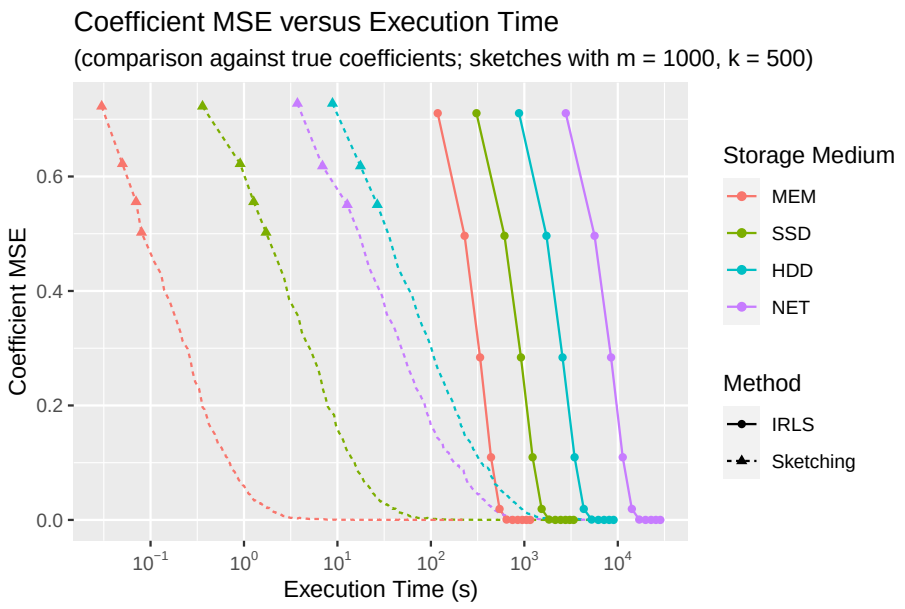

Figure 7.4: Coefficient MSE of the doubly-sketched and IRLS estimates against the simulated dataset's true $\boldsymbol{\beta}$, compared at each iteration against wall-clock execution time. Sketching plot glyphs partially suppressed for clarity. The doubly-sketched procedure converges with additional iterations; an approximate estimate can be obtained with reduced wall-clock time.

7.4 Real-world Datasets

In this section, we evaluate the proposed method against other methods in the literature, particularly OSMAC (Wang et al., 2019), FASA (Lee et al., 2021), and Optimal Distributed Subsampling (ODS) (Yu et al., 2022b) in addition to stochastic gradient descent (SGD) and drawing a single subsample. We treat SGD as a special case of the proposed doubly-sketching method without using any Hessian information, and as in doubly-sketching, we ignore cases where $m < k$ and treat cases where $m = k$ as uniform sketching only which corresponds to conventional SGD. To assure convergence, we continue to use the prefactor $a_t = t^{-1}$ for SGD (Robbins and Monro, 1951; Blum, 1954). Additionally, while the examples provided in this work assume a single pre-defined model specification, an added benefit of the method's data-oblivious nature is that each sketched surrogate dataset is valid for any model specification, allowing multiple proposed GLMs to be estimated in one set of iterations and aiding model selection. For ODS, which is a distributed method assuming multiple workers, we run five workers on the same local device each with access to one-fifth of the dataset. A single subsample consists of a simple random sample without replacement from the data; for simplicity, we ignore any local memory constraints for this method and allow subsamples with size far exceeding the uniform sketch size m.

7.4.1 Supersymmetric Particles Dataset

We evaluate the proposed method in comparison to the OSMAC method of Wang et al. (2018) and basic subsampling. The data of interest is a real-world dataset; the supersymmetric benchmark dataset (SUSY) (Baldi et al., 2014) available from the UCI Machine Learning Repository (Dua and Graff, 2017). This dataset comprises $n = 5 \times 10^6$ observations with $d = 18$ real-valued covariates and a single binary response variable, which we will model using a binomial GLM with logit link as in Wang et al. (2018).

We evaluate the proposed method on this dataset in comparison to multiple competing methods. We perform doubly-sketching with $m, k \in \{1000, 2000, 5000\}$ with $t_{\max} \in \{1035, 2080, 3004\}$ against 1) the full data IRLS, 2) a single subsample of size $m \in$

$\{5 \times 10^4, 10^5, 5 \times 10^5, 10^6\}$ from the data and fitting with IRLS, 3) the OSMAC method (Wang et al., 2018) with $r_0, r \in \{10^4, 5 \times 10^4, 10^5, 5 \times 10^5\}$ in both the mVc and mMSE modes, 4) the FASA method (Lee et al., 2021) for pilot sample and subsample sizes $\in \{10000, 50000, 100000\}$ and $r_2 \in \{10, 20, 30\}$, 5) the Optimal Distributed Subsampling (ODS) method (Yu et al., 2022b) with $r_0, r \in \{100000, 500000\}$ and $\varrho \in \{0.25, 0.5, 0.75\}$ in the uniform, mVc, and mMSE modes, and 6) stochastic gradient descent. For FASA, we use the logistic regression (Binomial-Logit) method for datasets with non-power-of-2 number of observations.

In the absence of true coefficients $\boldsymbol{\beta}$, the evaluated methods are compared against IRLS values with MSE as a metric. The performance in terms of both coefficient parameters $\hat{\boldsymbol{\beta}}$ and the standard errors are shown as a function of execution time in Figure 7.5. Table 7.1 and Table 7.2 provide the fitted coefficients and standard errors obtained across a selection of tested methods and configurations, chosen to have approximately the same wall-clock execution time in memory on the efficient frontier outlined in Figure 7.5.

We observe in Figure 7.5 that the doubly-sketching produces results comparable to most methods in terms of coefficient MSE, though the uniform-only sketch produces the most wall-clock time efficient frontier across all tested methods. FASA and OSMAC yield similar performance in their best configurations, though in both methods the mvc subsamples outperform the mmse subsamples. Interestingly, the ODS method with five parallel workers does not seem to yield a dramatic benefit in this comparative study, though we ascribe this to the unrealistic implementation of all five 'distributed' workers being simulated on the same computer and using the same pool of memory to store their respective dataset partitions. The stochastic gradient descent method is less efficient overall due to the lack of curvature information given by the Hessian and suffers slow convergence; indeed, it appears the diminuitive size of each sketch means that the relatively fixed overhead of each iteration dominates the incremental cost of computing the Hessian. In the lower half of Figure 7.5 and interpreting the MSE on the standard errors $\widehat{\text{se}}(\boldsymbol{\beta}_{\text{MLE}})$, the single subsample does stand out in terms of efficiency though both double and uniform sketching yield competitive MSEs at a slower wall-clock time.

In the fitted coefficients of Table 7.1, we see that for the chosen settings, doubly-sketching and uniform-only sketching both retrieve a total of 2.07 million rows of the data. For ODS, while a total of 600 thousand rows are retrieved for use in the actual estimation, all 5 million observations are also retrieved temporarily in order to calculate the weights. A similar situation occurs for OSMAC and FASA. In the single subsample we disregarded any constraints on how much data can be retained in order to compute IRLS on the subsample; moreover, a subsample of size 10^6 already retrieves and stores one-fifth of the full dataset. In the standard errors of Table 7.2, we note that these values are of interest during GLM model fitting as they allow for hypothesis testing on the point estimates of the GLM coefficients, which in turn yield oft-desired p-values.

The design of the OSMAC, FASA, and ODS methods all draw an initial pilot subsample from the full dataset, upon which a set of GLM coefficients are estimated. This pilot set of coefficients is used to compute $\mu_i = g^{-1}(\eta_i)$ which is in turn used to calculate the sampling probabilities $\{\pi_i\}_{i=1}^n$ across the full dataset; a very computationally expensive proposition under the assumption of slow data transfer. Indeed, for larger choices of the second-stage subsample size as weighted by $\{\pi_i\}_{i=1}^n$, the large fixed cost of determining these weights can be better amortized. Indeed, as this weighted subsample size increases, we can see that the performance of these methods becomes more competitive in Figure 7.5.

Figure 7.5: Coefficient and standard error MSE of multiple methods compared against the SUSY dataset's IRLS-fitted values, versus wall-clock execution time. 10 replications were taken at each choice of parameters, represented by a point for the median time and MSE with error bars omitted for clarity. Line segments form a frontier towards the origin representing the optimal set of parameter combinations for each method; envelopes toward the bottom-left are more efficient. The horizontal axis is square-root transformed and the vertical axis is log-transformed to increase visual separation; the reference IRLS with zero MSE is shown as a vertical line. Methods that do not provide estimates of the standard error $\widehat{\text{se}}(\boldsymbol{\beta}_{\text{MLE}})$ are not included in the lower figure.

Table 7.1: Fitted coefficients for the SUSY dataset for multiple methods with a selected set of parameters that have similar execution time and are on the efficient coefficient frontier of Figure 7.5. Values are averaged over 10 replications with standard deviations in brackets.

Covariate	IRLS	Doubly-Sketching	Uniform Only	ODS mvc	OSMAC mvc	FASA	Single Subsample
(Intercept)	−1.659	−1.650 (0.033)	−1.660 (0.016)	−1.659 (0.041)	−1.644 (0.060)	−1.635 (0.162)	−1.665 (0.020)
X1	2.326	2.308 (0.032)	2.323 (0.009)	2.306 (0.047)	2.312 (0.066)	2.363 (0.119)	2.335 (0.031)
X2	0.317	0.316 (0.019)	0.315 (0.004)	0.306 (0.023)	0.313 (0.030)	0.324 (0.075)	0.319 (0.011)
X3	0.204	0.156 (0.161)	0.204 (0.047)	0.327 (0.215)	0.215 (0.250)	0.132 (0.536)	0.213 (0.100)
X4	−1.601	−1.572 (0.068)	−1.597 (0.026)	−1.589 (0.100)	−1.590 (0.080)	−1.692 (0.258)	−1.602 (0.050)
X5	−1.714	−1.715 (0.025)	−1.710 (0.015)	−1.708 (0.032)	−1.730 (0.073)	−1.750 (0.141)	−1.719 (0.019)
X6	0.098	0.088 (0.017)	0.097 (0.011)	0.086 (0.020)	0.092 (0.038)	0.124 (0.073)	0.102 (0.027)
X7	−2.038	−1.990 (0.151)	−2.034 (0.042)	−2.145 (0.198)	−2.047 (0.197)	−1.961 (0.511)	−2.047 (0.095)
X8	0.533	0.542 (0.031)	0.533 (0.017)	0.538 (0.026)	0.532 (0.078)	0.510 (0.090)	0.533 (0.040)
X9	−0.623	−0.623 (0.013)	−0.623 (0.006)	−0.627 (0.027)	−0.619 (0.023)	−0.620 (0.090)	−0.617 (0.013)
X10	1.106	1.122 (0.043)	1.109 (0.013)	1.077 (0.032)	1.126 (0.089)	1.154 (0.202)	1.119 (0.042)
X11	0.000	0.001 (0.003)	−0.001 (0.002)	0.000 (0.009)	−0.001 (0.004)	0.001 (0.022)	0.001 (0.003)
X12	−0.002	−0.003 (0.005)	−0.001 (0.002)	−0.003 (0.008)	−0.002 (0.007)	−0.001 (0.018)	−0.003 (0.004)
X13	0.470	0.474 (0.012)	0.471 (0.006)	0.466 (0.015)	0.466 (0.045)	0.446 (0.066)	0.466 (0.014)
X14	0.001	0.001 (0.005)	0.001 (0.002)	0.001 (0.007)	0.001 (0.009)	0.012 (0.016)	0.000 (0.005)
X15	0.000	0.001 (0.002)	0.000 (0.001)	−0.001 (0.008)	−0.002 (0.006)	−0.004 (0.013)	0.000 (0.004)
X16	4.683	4.650 (0.078)	4.675 (0.019)	4.692 (0.062)	4.672 (0.057)	4.751 (0.179)	4.680 (0.025)
X17	0.004	0.005 (0.004)	0.004 (0.003)	0.005 (0.008)	0.003 (0.008)	0.000 (0.021)	0.002 (0.003)
X18	−0.410	−0.405 (0.026)	−0.412 (0.007)	−0.410 (0.020)	−0.400 (0.024)	−0.398 (0.049)	−0.412 (0.013)
Settings		$m = 2000$ $k = 1000$ $t_{\max} = 1035$	$m = 1000$ $t_{\max} = 2016$	$r_0 = 10^5$ $r = 5 \times 10^5$ $\varrho = 0.50$	$r_0 = 10^5$ $r = 10^4$	$r_0 = 10^4$ $r_1 = 10^4$ $r_2 = 10$	$m = 10^6$
Time (s)	23.53	2.468 (0.357)	2.226 (0.326)	2.756 (0.072)	2.618 (0.088)	2.796 (0.555)	3.067 (0.021)

7.4.2 Airline Delays Dataset

In this section, we investigate the airline delays dataset (Bureau of Transportation Statistics, 2008) using a Gamma family generalized linear model. The full dataset features 118,914,458 observations and 29 variables. We restrict the dataset to flights that have an arrival delay greater than one minute and an absolute difference between the departure and arrival delays no greater than twelve hours. This is done to remove apparent outliers such as departures being delayed 23 hours but arriving 23 hours early. We construct as the response variable y the natural logarithm of the arrival delay in minutes with covariates x_1 being the standardized distance of the flight, x_2 an indicator variable for whether the flight departed between 0700 and 1800 in the local time zone, x_3 an indicator for weekends, and x_4 an indicator for the departure delay exceeding 15 minutes. On these five variables, we retain only complete cases, leaving 52,691,955 observations for the analysis which in CSV form consumes 2.17 GiB of disk space. To this data, we fit a Gamma GLM with the canonical inverse link function with the above four covariates and an intercept term. Here, in lieu of the starting value $\boldsymbol{\beta}^{(0)} = \mathbf{0}$ which is invalid for Gamma GLMs, we initialize with the intercept-only estimate $\boldsymbol{\beta}^{(0)} = \langle 1/\bar{y}, 0, ..., 0 \rangle$ across the entire dataset.

Here, we compare doubly-sketching against a single subsample as well as the ODS method (Yu et al., 2022b), again using the IRLS fit as a baseline. In order for doubly-sketching to produce standard errors comparable to the `glm()` function of the R language (R Core Team, 2019), we also estimate the dispersion using the additional sub-step described in Section 7.2.4. The doubly-sketching method is the same as in Section 7.4.1. The ODS tested ODS setup is also as in Section 7.4.1 with five workers on the same local computer, each with one-fifth of the full dataset. Due to the larger size of this dataset, we have increased the tested range of the pilot sample size r_0 and second-stage subsample size r to $\{10^5, 10^6, 5 \times 10^6, 10^7\}$. Again, the uniform, mVc, and mMSE modalities are tested with $\varrho \in \{0.25, 0.5, 0.75\}$.

Figure 7.6 shows the mean square error of the coefficients and standard errors with the efficient frontier highlighted. We see that sketching with $m = k$ yields the best coefficient estimates on average. ODS with uniform sampling weights provides competitive performance

Table 7.2: Fitted standard errors (x100) for the SUSY dataset for the methods and parameters found in Table 7.1. Values are averaged over 10 replications with standard deviations in brackets.

Covariate	IRLS	Doubly-Sketching	Uniform Only	Single Subsample
(Intercept)	0.763	0.777 (0.006)	0.771 (0.005)	0.764 (0.002)
X1	0.856	0.868 (0.007)	0.862 (0.003)	0.857 (0.001)
X2	0.457	0.463 (0.003)	0.460 (0.002)	0.457 (0.001)
X3	3.647	3.684 (0.025)	3.675 (0.031)	3.645 (0.009)
X4	1.396	1.414 (0.012)	1.405 (0.004)	1.396 (0.004)
X5	0.781	0.793 (0.007)	0.787 (0.006)	0.782 (0.002)
X6	0.500	0.505 (0.004)	0.504 (0.003)	0.501 (0.001)
X7	3.385	3.418 (0.025)	3.410 (0.035)	3.382 (0.010)
X8	0.909	0.920 (0.007)	0.916 (0.003)	0.910 (0.002)
X9	0.465	0.472 (0.003)	0.469 (0.002)	0.465 (0.001)
X10	1.156	1.172 (0.010)	1.161 (0.007)	1.157 (0.002)
X11	0.126	0.127 (0.000)	0.127 (0.000)	0.126 (0.000)
X12	0.123	0.124 (0.000)	0.124 (0.000)	0.123 (0.000)
X13	0.463	0.470 (0.003)	0.467 (0.003)	0.463 (0.001)
X14	0.126	0.128 (0.001)	0.127 (0.001)	0.126 (0.000)
X15	0.120	0.121 (0.000)	0.121 (0.000)	0.120 (0.000)
X16	1.000	1.010 (0.008)	1.008 (0.005)	0.999 (0.002)
X17	0.119	0.120 (0.001)	0.119 (0.000)	0.119 (0.000)
X18	0.415	0.420 (0.005)	0.419 (0.003)	0.416 (0.001)
Settings		$m = 2000$ $k = 1000$ $t_{\max} = 1035$	$m = 1000$ $t_{\max} = 2016$	$m = 10^6$
Time (s)	23.53	2.468 (0.357)	2.226 (0.326)	3.067 (0.021)

using five distributed workers with taking a single subsample without replacement, whereas ODS with non-uniform sampling weights suffers a large up-front performance penalty due to the need to compute said sampling weights across the entire dataset's $n = 52,691,955$ observations. Doubly-sketching performs comparably to ODS in this regard. In terms of the standard errors, a single subsample performs best, followed by uniform-only sketching, and doubly-sketching, though all three methods are numerically comparable.

7.4.3 New York Yellow Taxicab Dataset

In this section we apply the proposed method to a much larger real-world dataset describing taxicab trips in New York (NYC Taxi and Limousine Commission, 2022). We utilise the entirety of the Yellow Taxi Trip Records dataset available as of writing, spanning from January 2009 to January 2023 inclusive. After pre-processing and importing into a PostgreSQL database with an additional index column, the table consumed 160 GiB of disk space holding a total of 1,669,852,068 observations. A complete technical description of the data cleaning and import process is given in Appendix D.3.

This dataset is of particular interest for multiple reasons. First, as a real-world dataset with less than ideal cleanliness, it contains heterogeneous data with major outliers caused by data entry errors; we have chosen to keep these in the dataset to challenge the proposed method. For example, the trip record with distance recorded as 134,619,063 miles and a trip time of 49 seconds, with roughly 14 times the speed of light, participates in the estimation procedure with probability $\left(1 - \frac{m}{n}\right)^{t_{\max}}$. Second, the data contains many categorical covariates, adding to the difficulty induced by uniform sketching, especially as some events are relatively rare. The use of categorical variables also introduces the caveat that factor levels must be known a priori to form the regressor matrix $\mathbf{X}$. Third, the monetary amounts exhibit high multi-collinearity, compounding upon the second point by further reducing the conditioning of the Hessian matrix.

We construct a generalized linear model of the passenger count as a Poisson response with log-link with the following covariates derived from the available set of variables. An

Figure 7.6: Coefficient and standard error MSE of multiple methods compared against the airline delay dataset's IRLS-fitted values, versus wall-clock execution time. 10 replications were taken at each choice of parameters and are summarized by their median. Line segments form a frontier towards the origin representing the optimal set of parameter combinations for each method; envelopes toward the bottom-left are more efficient. The horizontal axis is square-root transformed and the vertical axis is log-transformed to increase visual separation; the reference IRLS with zero MSE is shown as a vertical line.

intercept is included and no interaction terms are considered. For simplicity, the first element in each categorical variable is taken to be the reference level.

- Vendor ID: Factor with three levels: 1, 2, 3.
- Rate Code: Factor with six levels: 1 through 6.
- Payment Type: Factor with five levels: 1 through 5.
- Day of Week: Factor with seven levels: Monday, Tuesday, ..., Sunday.
- Time of Day: Factor level with four levels: Twilight [00:00, 06:00), Morning [06:00,12:00), Afternoon [12:00,18:00), Night [18:00, 00:00).
- Duration: Time elapsed between pickup and drop-off in minutes.
- Distance, Fare Amount, Tip Amount, Tolls Amount, Total Amount: As specified in the data dictionary.

Due to the categorical nature of many of these covariates and the possibility of omitting a factor level entirely during the Uniform sketch, we can mitigate this by selecting larger sketch sizes; here, we choose $m = 10000$ with corresponding Clarkson-Woodruff sketch size $k = 5000$. We perform $t_{\max} = 1035$ iterations of the proposed procedure with ten replications, noting that sometimes a particularly pathological sketch can cause numerical issues. Possible measures to detect pathological sketches include checking the (reciprocal) condition number of the sketched Hessian $\tilde{\mathbf{H}}^{(t)}$ or controlling the regularization by increasing the constant $\hat{z}$. For this dataset, we apply the following heuristic to revert bad updates. Let $\Delta^{(t)} = \|\hat{\boldsymbol{\beta}}^{(t)} - \hat{\boldsymbol{\beta}}^{(t-1)}\|_2$ be the length of the update from iteration $t-1$ to t. If the proposed update is such that $\Delta^{(t)} \geq 10 \times \frac{1}{t-1} \sum_{i=1}^{t-1} \Delta^{(i)}$, then revert the update on $\hat{\boldsymbol{\beta}}^{(t)}$ for this iteration. Comparison is made against sketching with uniform-only sketches of size $m = 10000$ as well as using a single subsample from the data at sizes $m \in \{10^4, 10^5, 10^6, 10^7\}$ The largest subsample size 10^7 approximates the total data transfer quantity of the doubly-

and uniform-only sketching procedures at $1035 \times 10000 = 10350000$. As in the supersymmetric and airline delays datasets, we ignore any local system memory constraints that may prevent storage of a large subsample.

With reference to the technical remark in Section 7.3.2, and considering the scale of the dataset, we take advantage of an approximate sampling method available in PostgreSQL in place of the Uniform sketch. This query method is called `TABLESAMPLE` in the SQL standard and produces an approximation to simple random sampling without replacement by performing one-stage cluster sampling over blocks of rows for a considerable speedup in data retrieval. We use this in-place of a simple random sample in all tested methods for performance reasons. A brief technical description of this sampling method is given in Appendix D.3.1.

The GLM model coefficients and standard errors fitted via sketching with ten replications is presented in Table 7.3 and Table 7.4 with comparison to a model fitted using IRLS with the same convergence condition as in Section 7.3.1. Figure 7.7 depicts the MSE values of each of the ten replications against execution time. Due to the lack of a true $\boldsymbol{\beta}$, we compare against the fitted IRLS estimate $\hat{\boldsymbol{\beta}}_{\text{IRLS}}$ to obtain an MSE value.

We draw attention to the almost 21 hour long procedure needed to fit the IRLS conventionally at 2.5 hours per iteration, whereas sketching only required 25 minutes; a fifty-fold improvement. While the choice of $t_{\max} = 1035$ was considerably conservative with a total data transfer comprising approximately 0.6% of the complete dataset, fairly accurate results were still obtained despite the multi-collinearity and categorical nature of the covariates. Indeed, ten replications of sketching at 25 minutes apiece run in serial sequence would require approximately four hours of wall-clock time, running all ten replications simultaneously yielded a total execution time of approximately one hour, suggesting potential for parallelism in the tested computational infrastructure.

Table 7.3: Fitted regression coefficients for the NYC Yellow Taxicab data using a PostgreSQL database across the network. Averages and standard deviations in round brackets over ten replications are shown with reference to the IRLS fitted coefficients, for which a single replication was performed.

Covariate	IRLS	Doubly-Sketching	Uniform Only	Sample 10^4	Sample 10^5	Sample 10^6	Sample 10^7
(Intercept)	0.319	0.325 (0.007)	0.320 (0.018)	0.319 (0.039)	0.333 (0.009)	0.328 (0.004)	0.318 (0.022)
vendor_id2	−0.044	−0.041 (0.001)	−0.042 (0.001)	−0.026 (0.056)	−0.047 (0.020)	−0.042 (0.005)	−0.044 (0.004)
vendor_id3	−0.048	−0.045 (0.003)	−0.045 (0.002)	−0.038 (0.038)	−0.048 (0.021)	−0.045 (0.005)	−0.048 (0.005)
payment_type2	−0.047	−0.044 (0.002)	−0.044 (0.002)	−0.056 (0.032)	−0.049 (0.019)	−0.045 (0.003)	−0.047 (0.005)
payment_type3	−0.044	−0.041 (0.003)	−0.040 (0.001)	−0.064 (0.034)	−0.047 (0.018)	−0.043 (0.004)	−0.044 (0.005)
payment_type4	−0.027	−0.024 (0.003)	−0.024 (0.001)	−0.019 (0.047)	−0.033 (0.017)	−0.023 (0.005)	−0.026 (0.005)
payment_type5	0.007	0.008 (0.002)	0.008 (0.000)	0.010 (0.049)	0.006 (0.018)	0.009 (0.005)	0.008 (0.002)
duration	0.000	0.002 (0.006)	0.004 (0.004)	0.005 (0.006)	0.003 (0.001)	0.001 (0.002)	0.000 (0.000)
timeofday[6,12)	0.000	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
timeofday[12,18)	0.039	0.049 (0.003)	0.049 (0.002)	0.053 (0.015)	0.050 (0.004)	0.049 (0.001)	0.040 (0.021)
timeofday[18,24]	0.046	0.036 (0.002)	0.034 (0.009)	0.035 (0.027)	0.036 (0.007)	0.039 (0.002)	0.042 (0.006)
dayofweek2	−0.049	−0.059 (0.006)	−0.057 (0.004)	−0.056 (0.152)	−0.071 (0.035)	−0.061 (0.013)	−0.053 (0.008)
dayofweek3	−0.159	−0.165 (0.020)	−0.167 (0.012)	−0.217 (0.300)	−0.225 (0.040)	−0.178 (0.013)	−0.164 (0.026)
dayofweek4	0.060	0.050 (0.018)	0.061 (0.015)	−0.071 (0.443)	−0.025 (0.072)	0.065 (0.045)	0.060 (0.023)
dayofweek5	−0.035	−0.036 (0.004)	−0.033 (0.005)	−0.028 (0.043)	−0.030 (0.015)	−0.039 (0.004)	−0.035 (0.004)
dayofweek6	−0.001	0.000 (0.002)	0.003 (0.005)	0.010 (0.035)	0.003 (0.020)	0.000 (0.004)	−0.001 (0.002)
dayofweek7	−0.080	−0.083 (0.003)	−0.081 (0.005)	−0.079 (0.054)	−0.083 (0.015)	−0.084 (0.003)	−0.080 (0.010)
distance	0.039	0.046 (0.001)	0.044 (0.006)	0.050 (0.013)	0.047 (0.005)	0.047 (0.001)	0.038 (0.021)
fare_amount	0.039	0.051 (0.001)	0.048 (0.009)	0.056 (0.018)	0.054 (0.004)	0.052 (0.002)	0.042 (0.020)
tip_amount	−0.039	−0.049 (0.001)	−0.048 (0.001)	−0.053 (0.014)	−0.050 (0.004)	−0.049 (0.001)	−0.039 (0.021)
tolls_amount	0.453	0.455 (0.001)	0.455 (0.001)	0.453 (0.014)	0.450 (0.010)	0.453 (0.002)	0.453 (0.003)
total_amount	0.031	0.026 (0.005)	0.026 (0.004)	0.024 (0.099)	0.031 (0.010)	0.028 (0.007)	0.032 (0.008)
Settings		$m = 10000$ $k = 5000$ $t_{\max} = 1035$	$m = 10000$ $t_{\max} = 1035$	$m = 10^4$	$m = 10^5$	$m = 10^6$	$m = 10^7$
Time	20h35m40s	25m08s (1m50s)	22m38s (2m18s)	5s (3s)	24s (3s)	3m35s (12s)	21m43s (30s)

Table 7.4: Standard errors (x10000) for the NYC Yellow Taxicab data using a PostgreSQL database across the network. Estimates for doubly-sketching with $(m, k) = (10000, 5000)$ and uniform-only sketching with $m = 10000$ for $t_{\max} = 1035$ iterations and single sub-sample with sizes $m = 10^4, 10^5, 10^6, 10^7$. Averages and standard deviations in round brackets over ten replications are shown with reference to the IRLS fitted coefficients, for which a single replication was performed.

Covariate	IRLS	Doubly-Sketching	Uniform Only	Sample 10^4	Sample 10^5	Sample 10^6	Sample 10^7
(Intercept)	0.785	0.872 (0.038)	0.853 (0.045)	0.862 (0.069)	0.850 (0.019)	0.838 (0.012)	0.823 (0.017)
vendor_id2	0.749	0.775 (0.035)	0.784 (0.031)	0.768 (0.060)	0.751 (0.023)	0.747 (0.008)	0.750 (0.001)
vendor_id3	0.734	0.761 (0.026)	0.755 (0.037)	0.771 (0.066)	0.743 (0.017)	0.731 (0.009)	0.734 (0.001)
payment_type2	0.727	0.745 (0.039)	0.745 (0.041)	0.748 (0.060)	0.732 (0.012)	0.727 (0.007)	0.727 (0.002)
payment_type3	0.720	0.749 (0.015)	0.750 (0.043)	0.768 (0.079)	0.717 (0.014)	0.717 (0.009)	0.720 (0.002)
payment_type4	0.711	0.732 (0.027)	0.747 (0.039)	0.735 (0.070)	0.714 (0.016)	0.710 (0.006)	0.712 (0.002)
payment_type5	0.701	0.713 (0.040)	0.717 (0.054)	0.739 (0.061)	0.703 (0.017)	0.701 (0.008)	0.702 (0.002)
duration	0.000	0.077 (0.037)	0.093 (0.032)	0.117 (0.017)	0.107 (0.018)	0.040 (0.051)	0.000 (0.000)
timeofday[6,12)	0.000	0.005 (0.001)	0.005 (0.000)	0.005 (0.002)	0.005 (0.000)	0.003 (0.002)	0.001 (0.002)
timeofday[12,18)	0.213	0.231 (0.026)	0.219 (0.010)	0.230 (0.017)	0.223 (0.007)	0.218 (0.003)	0.173 (0.089)
timeofday[18,24]	0.387	0.491 (0.011)	0.490 (0.007)	0.506 (0.027)	0.489 (0.013)	0.486 (0.006)	0.473 (0.007)
dayofweek2	3.992	4.049 (0.232)	3.849 (0.183)	3.962 (0.333)	3.987 (0.191)	4.031 (0.050)	4.003 (0.013)
dayofweek3	6.479	6.742 (1.049)	6.253 (0.539)	7.126 (1.909)	6.514 (0.475)	6.440 (0.143)	6.488 (0.030)
dayofweek4	7.983	7.639 (0.987)	8.213 (1.214)	11.467 (7.293)	8.211 (0.722)	7.818 (0.324)	7.980 (0.135)
dayofweek5	0.649	0.672 (0.034)	0.663 (0.029)	0.692 (0.053)	0.657 (0.021)	0.648 (0.004)	0.650 (0.002)
dayofweek6	0.638	0.650 (0.037)	0.651 (0.024)	0.672 (0.050)	0.644 (0.021)	0.634 (0.004)	0.639 (0.002)
dayofweek7	0.688	0.710 (0.033)	0.708 (0.019)	0.720 (0.061)	0.694 (0.021)	0.687 (0.005)	0.688 (0.004)
distance	0.213	0.269 (0.023)	0.256 (0.008)	0.270 (0.020)	0.259 (0.007)	0.256 (0.003)	0.220 (0.061)
fare_amount	0.213	0.283 (0.030)	0.280 (0.015)	0.293 (0.015)	0.283 (0.018)	0.265 (0.020)	0.207 (0.064)
tip_amount	0.213	0.228 (0.026)	0.215 (0.008)	0.224 (0.016)	0.219 (0.007)	0.216 (0.002)	0.173 (0.090)
tolls_amount	0.397	0.400 (0.004)	0.401 (0.002)	0.405 (0.013)	0.398 (0.006)	0.397 (0.002)	0.397 (0.001)
total_amount	2.253	2.348 (0.311)	2.258 (0.330)	2.722 (0.981)	2.349 (0.176)	2.273 (0.052)	2.252 (0.011)
Settings		$m = 10000$ $k = 5000$ $t_{\max} = 1035$	$m = 10000$ $t_{\max} = 1035$	$m = 10^4$	$m = 10^5$	$m = 10^6$	$m = 10^7$
Time	20h35m40s	25m08s (1m50s)	22m38s (2m18s)	5s (3s)	24s (3s)	3m35s (12s)	21m43s (30s)

7.5 Discussion

In the above sections, we propose and evaluate an approximate iterative solution to the problem of generalized linear models using a sequence of surrogate datasets generated by sketching. We find that the method demonstrates computational tractability especially in regards to massive datasets and across a variety of commodity computational infrastructure that are readily accessible to most statistics practitioners.

For both simulation studies and real-world datasets, we find desirable numerical properties that support the theoretical properties that support the proposed model's usage in practical situations where estimation of GLM coefficient parameters and standard errors are desired. Potential future work includes finding a heuristic for determining appropriate sketch sizes m and k for different contexts, while balancing runtime and accuracy using a priori information on dataset peculiarities such as categorical covariates, rare event distributions, and degree of multi-collinearity. A dynamic approach to selecting m and k during the estimation procedure could also increase efficiency of the algorithm by avoiding unnecessary data transfer when not needed at a particular iteration. Conversely, when a pathological sketch is obtained, such as in the New York Yellow Taxicab dataset, the Hessian and gradient estimates could be retained instead of discarded and amalgamated with the subsequent sketch to avoid wasting the data transfer cost. Finally, strategies for deciding whether to take a large number of iterations $t_{\max}$ or perform multiple shorter replications and combining the results may yield interesting extensions of the proposed method.

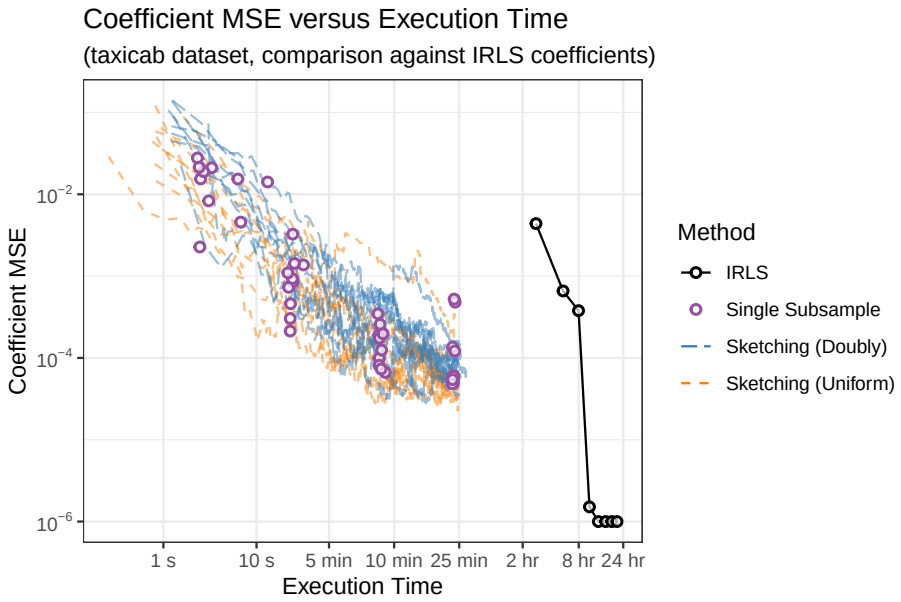

Figure 7.7: Coefficient MSE of the sketched coefficients for the New York Yellow Taxicab dataset against $\hat{\boldsymbol{\beta}}_{\text{IRLS}}$, compared against wall-clock execution time. Doubly-sketching performed with $(m, k) = (10000, 5000)$, uniform-only sketching performed with $m = 10000$. each sketching and IRLS replication is indicated by a MSE trace. Points omitted for sketched traces for clarity, and IRLS MSE is truncated to 10^{-6} to improve axis scaling. Single subsamples count the total query time and assumes the entire retrieved sample can fit into working memory.

Chapter 8

Conclusion

This thesis explored a range of structured finite Gaussian mixture models whose parameters correspond to a variety of real-world data relationships, with application to datasets both simulated and real. Methods were proposed to accelerate the EM procedure and GLM model fitting via IRLS. Chapters 3 to 5 provide an investigation into parameter hybridization and hierarchical parameter sharing, reducing the impact of dimensionality and adding an additional aspect of interpretability to the mixture model. Chapter 6 provided a means to accelerate the EM procedure by extrapolating the conditional expectations, with attention to finite Gaussian mixture models. Chapter 7 details a faster but approximate method of estimating GLM coefficients and standard errors, with emphasis on practical data infrastructures. The conclusion of this thesis is by no means an end to the potential avenues of future exploration; food for future thought has been left for future generations of scholars in the corresponding chapter.

References

Daniel Ahfock, William J. Astle, and Sylvia Richardson. On randomized sketching algorithms and the Tracy-Widom law, 2022. URL https://arxiv.org/abs/2201.00450.

Daniel C Ahfock, William J Astle, and Sylvia Richardson. Statistical properties of sketching algorithms. *Biometrika*, 108(2):283–297, 07 2020. ISSN 0006-3444. doi: 10.1093/biomet/asaa062. URL https://doi.org/10.1093/biomet/asaa062.

Nir Ailon and Bernard Chazelle. The fast Johnson–Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009. doi: 10.1137/060673096. URL https://doi.org/10.1137/060673096.

Edoardo M Airoldi, David Blei, Elena A Erosheva, and Stephen E Fienberg. *Handbook of mixed membership models and their applications*. CRC press, 2014.

Edoardo Maria Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of machine learning research*, 8(4):597, 2008. ISSN 2227-7390. doi: 10.3390/math8040597. URL http://dx.doi.org/10.3390/math8040597.

Edgar Anderson. The species problem in Iris. *Annals of the Missouri Botanical Garden*, 23(3):457–509, 1936. ISSN 00266493. doi: https://doi.org/10.2307/2394164. URL http://www.jstor.org/stable/2394164.

Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-

energy physics with deep learning. *Nature Communications*, 5(1):4308, Jul 2014. ISSN 2041-1723. doi: 10.1038/ncomms5308. URL https://doi.org/10.1038/ncomms5308.

Jeffrey D. Banfield and Adrian E. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49(3):803–821, 1993. ISSN 0006341X, 15410420. doi: https://doi.org/10.2307/2532201. URL http://www.jstor.org/stable/2532201.

Alexis Battle, Eran Segal, and Daphne Koller. Probabilistic discovery of overlapping cellular processes and their regulation. *Journal of Computational Biology*, 12(7):909–927, 2005. doi: 10.1089/cmb.2005.12.909. URL https://doi.org/10.1089/cmb.2005.12.909. PMID: 16201912.

A. Berlinet and Ch. Roland. Acceleration schemes with application to the EM algorithm. *Computational Statistics & Data Analysis*, 51(8):3689–3702, 2007. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2006.12.013. URL https://www.sciencedirect.com/science/article/pii/S0167947306004919.

Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3):561–575, 2003. ISSN 0167-9473. doi: https://doi.org/10.1016/S0167-9473(02)00163-9. URL https://www.sciencedirect.com/science/article/pii/S0167947302001639. Recent Developments in Mixture Model.

L. S. Blackford, J. Choi, A. Cleary, E. D'Azeuedo, J. Demmel, I. Dhillon, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, R. C. Whaley, and Jack J. Dongarra. *ScaLAPACK User's Guide*. Society for Industrial and Applied Mathematics, USA, 1997. ISBN 0898713978.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, March 2003. ISSN 1532-4435.

Julius R. Blum. Multidimensional stochastic approximation methods. *The Annals of*

Mathematical Statistics, 25(4):737–744, 1954. ISSN 00034851. URL http://www.jstor.org/stable/2236657.

Dankmar Böhning, Ekkehart Dietz, Rainer Schaub, Peter Schlattmann, and Bruce G. Lindsay. The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family. *Annals of the Institute of Statistical Mathematics*, 46(2):373–388, Jun 1994. ISSN 1572-9052. doi: 10.1007/BF01720593. URL https://doi.org/10.1007/BF01720593.

Raghu Bollapragada, Richard H Byrd, and Jorge Nocedal. Exact and inexact subsampled Newton methods for optimization. *IMA Journal of Numerical Analysis*, 39(2):545–578, 04 2018. ISSN 0272-4979. doi: 10.1093/imanum/dry009. URL https://doi.org/10.1093/imanum/dry009.

Charles Bouveyron and Camille Brunet. Simultaneous model-based clustering and visualization in the fisher discriminative subspace. *Statistics and Computing*, 22(1):301–324, Jan 2012. ISSN 1573-1375. doi: 10.1007/s11222-011-9249-9. URL https://doi.org/10.1007/s11222-011-9249-9.

Ryan P. Browne and Paul D. McNicholas. Estimating common principal components in high dimensions. *Advances in Data Analysis and Classification*, 8(2):217–226, June 2014. ISSN 1862-5355. doi: 10.1007/s11634-013-0139-1. URL https://doi.org/10.1007/s11634-013-0139-1.

Bureau of Transportation Statistics. Data Expo 2009: Airline on time data, 2008. URL https://doi.org/10.7910/DVN/HG7NV7.

Richard H. Byrd, Gillian M. Chin, Will Neveitt, and Jorge Nocedal. On the use of stochastic Hessian information in optimization methods for machine learning. *SIAM Journal on Optimization*, 21(3):977–995, 2011. doi: 10.1137/10079923X. URL https://doi.org/10.1137/10079923X.

N. A. Campbell and R. J. Mahon. A multivariate study of variation in two species of rock

crab of the genus leptograpsus. *Australian Journal of Zoology*, 22(3):417–425, 1974. doi: 10.1071/ZO9740417. URL https://doi.org/10.1071/ZO9740417.

Ann Cannon, George Cobb, Bradley Hartlaub, Julie Legler, Robin Lock, Thomas Moore, Allan Rossman, and Jeffrey Witmer. *Stat2Data: Datasets for Stat2*, 2019. R package version 2.0.0.

Gilles Celeux and Jean Diebolt. L'algorithme SEM : un algorithme d'apprentissage probabiliste pour la reconnaissance de mélange de densités. *Revue de Statistique Appliquée*, 34(2):35–52, 1986. URL www.numdam.org/item/RSA_1986__34_2_35_0/.

Gilles Celeux and Gérard Govaert. Comparison of the mixture and the classification maximum likelihood in cluster analysis. *Journal of Statistical Computation and Simulation*, 47(3-4):127–146, 1993. doi: 10.1080/00949659308811525. URL https://doi.org/10.1080/00949659308811525.

Gilles Celeux and Gérard Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793, 1995. ISSN 0031-3203. doi: https://doi.org/10.1016/0031-3203(94)00125-6. URL http://www.sciencedirect.com/science/article/pii/0031320394001256.

Barbara Chapman, Gabriele Jost, and Ruud Van Der Pas. *Using OpenMP: portable shared memory parallel programming*. MIT press, 2007.

Douglas B. Clarkson and Robert I. Jennrich. Quartic rotation criteria and algorithms. *Psychometrika*, 53(2):251–259, Jun 1988. ISSN 1860-0980.

Kenneth L. Clarkson and David P. Woodruff. Low-rank approximation and regression in input sparsity time. *J. ACM*, 63(6), jan 2017. ISSN 0004-5411. doi: 10.1145/3019134. URL https://doi.org/10.1145/3019134.

Graham Cormode. Sketch techniques for approximate query processing. *Foundations and Trends in Databases. NOW publishers*, 2011.

Leonardo Dagum and Ramesh Menon. OpenMP: an industry standard API for shared-memory programming. *IEEE Computational Science and Engineering*, 5(1):46–55, 1998. doi: 10.1109/99.660313.

Yogesh Dahiya, Dimitris Konomis, and David P. Woodruff. An empirical evaluation of sketching for numerical linear algebra. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 1292–1300, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220098. URL https://doi.org/10.1145/3219819.3220098.

Jan De Leeuw and Willem J Heiser. Convergence of correction matrix algorithms for multidimensional scaling. *Geometric representations of relational data*, pages 735–752, 1977.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977. doi: https://doi.org/10.1111/j.2517-6161.1977.tb01600.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x.

Michal Derezinski and Michael W Mahoney. Distributed estimation of the inverse Hessian by determinantal averaging. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/b1b20d09041289e6c3fbb81850c5da54-Paper.pdf.

Paramveer Dhillon, Yichao Lu, Dean P Foster, and Lyle Ungar. New subsampling algorithms for fast least squares regression. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/3cec07e9ba5f5bb252d13f5f431e4bbb-Paper.pdf.

Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Sampling algorithms for l2 regression and applications. In *Proceedings of the Seventeenth Annual ACM-SIAM*

Symposium on Discrete Algorithm, SODA '06, page 1127–1136, USA, 2006. Society for Industrial and Applied Mathematics. ISBN 0898716055.

Petros Drineas, Michael W. Mahoney, S. Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *Numerische Mathematik*, 117(2):219–249, Feb 2011. ISSN 0945-3245. doi: 10.1007/s00211-010-0331-6. URL https://doi.org/10.1007/s00211-010-0331-6.

Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13(111):3475–3506, 2012. URL http://jmlr.org/papers/v13/drineas12a.html.

Yu Du and Ravi Varadhan. SQUAREM: An R package for off-the-shelf acceleration of EM, MM and other EM-like monotone algorithms. *Journal of Statistical Software*, 92 (7):1–41, 2020. doi: 10.18637/jss.v092.i07.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Elena A. Erosheva, Stephen E. Fienberg, and Cyrille Joutard. Describing disability through individual-level mixture models for multivariate binary data. *The annals of applied statistics*, 1(2):346–384, 2007. ISSN 1932-6157. doi: 10.1214/07-aoas126. URL https://pubmed.ncbi.nlm.nih.gov/21687832. 21687832[pmid].

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936. doi: https://doi.org/10.1111/j.1469-1809.1936.tb02137.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x.

Annie Foong and Frank Hady. Storage as fast as rest of the system. In *2016 IEEE 8th International Memory Workshop (IMW)*, pages 1–4, 2016. doi: 10.1109/IMW.2016.7495289.

Michele Forina, C Armanino, Sergio Lanteri, and E Tiscornia. Classification of olive oils from their fatty acid composition. In *Food research and data analysis: proceedings from the IUFoST Symposium, September 20-23, 1982, Oslo, Norway/edited by H. Martens and H. Russwurm, Jr.* London: Applied Science Publishers, 1983., 1983.

Chris Fraley. Algorithms for model-based gaussian hierarchical clustering. *SIAM Journal on Scientific Computing*, 20(1):270–281, January 1998. doi: 10.1137/s1064827596311451. URL https://doi.org/10.1137/s1064827596311451.

Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458): 611–631, 2002. doi: 10.1198/016214502760047131. URL https://doi.org/10.1198/016214502760047131.

Giuliano Galimberti and Gabriele Soffritti. Model-based methods to identify multiple cluster structures in a data set. *Computational Statistics & Data Analysis*, 52(1):520–536, 2007. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2007.02.019. URL https://www.sciencedirect.com/science/article/pii/S0167947307000758.

Giuliano Galimberti and Gabriele Soffritti. Finite mixture models for clustering multilevel data with multiple cluster structures. *Statistical Modelling*, 10(3):265–290, 2010. doi: 10.1177/1471082X0801000302. URL https://doi.org/10.1177/1471082X0801000302.

Giuliano Galimberti, Annamaria Manisi, and Gabriele Soffritti. Modelling the role of variables in model-based cluster analysis. *Statistics and Computing*, 28(1):145–169, Jan 2018. ISSN 1573-1375. doi: 10.1007/s11222-017-9723-0. URL https://doi.org/10.1007/s11222-017-9723-0.

Audrey P. Gasch, Paul T. Spellman, Camilla M. Kao, Orna Carmel-Harel, Michael B. Eisen, Gisela Storz, David Botstein, and Patrick O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, 11 (12):4241–4257, 2000. doi: 10.1091/mbc.11.12.4241. URL https://doi.org/10.1091/mbc.11.12.4241. PMID: 11102521.

Zoubin Ghahramani, Geoffrey E Hinton, et al. The EM algorithm for mixtures of factor analyzers. Technical report, Technical Report CRG-TR-96-1, University of Toronto, 1996.

Donald Goldfarb and Ashok Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical programming*, 27(1):1–33, 1983.

Isobel Claire Gormley and Thomas Brendan Murphy. A grade of membership model for rank data. *Bayesian Analysis*, 4(2):265 – 295, 2009. doi: 10.1214/09-BA410. URL https://doi.org/10.1214/09-BA410.

Peter M Gruber. *Convex and discrete geometry*, volume 336. Springer Science & Business Media, 2007. doi: 10.1007/978-3-540-71133-9. URL https://doi.org/10.1007/978-3-540-71133-9.

Branko Grünbaum. *Convex Polytopes*. Springer New York, 2003. doi: 10.1007/978-1-4613-0019-9. URL https://doi.org/10.1007/978-1-4613-0019-9.

Shelby J. Haberman. Maximum Likelihood Estimates in Exponential Response Models. *The Annals of Statistics*, 5(5):815 – 841, 1977. doi: 10.1214/aos/1176343941. URL https://doi.org/10.1214/aos/1176343941.

Frank Hansen and Gert K. Pedersen. Jensen's operator inequality. *Bulletin of the London Mathematical Society*, 35(4):553–564, 2003. doi: https://doi.org/10.1112/S0024609303002200. URL https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/S0024609303002200.

Yunxiao He and Chuanhai Liu. The dynamic 'expectation–conditional maximization either' algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74 (2):313–336, 2012. doi: https://doi.org/10.1111/j.1467-9868.2011.01013.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2011.01013.x.

Katherine A. Heller, Sinead Williamson, and Zoubin Ghahramani. Statistical models for partial membership. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 392–399, New York, NY, USA, 2008. Association for

Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390206. URL https://doi.org/10.1145/1390156.1390206.

Hajo Holzmann, Axel Munk, and Tilmann Gneiting. Identifiability of finite mixtures of elliptical distributions. *Scandinavian Journal of Statistics*, 33(4):753–763, 2006. doi: https://doi.org/10.1111/j.1467-9469.2006.00505.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9469.2006.00505.x.

Allison Marie Horst, Alison Presmanes Hill, and Kristen B Gorman. *palmerpenguins: Palmer Archipelago (Antarctica) penguin data*, 2020. URL https://allisonhorst.github.io/palmerpenguins/. R package version 0.1.0.

Jason Hou-Liu and Ryan P. Browne. Chimeral clustering. *Journal of Classification*, 39(1): 171–190, Mar 2022a. ISSN 1432-1343. doi: 10.1007/s00357-021-09396-3.

Jason Hou-Liu and Ryan P. Browne. Factor and hybrid components for model-based clustering. *Advances in Data Analysis and Classification*, 16(2):373–398, Jun 2022b. ISSN 1862-5355. doi: 10.1007/s11634-021-00483-2.

Jason Hou-Liu and Ryan P. Browne. Model-based clustering with nested Gaussian clusters. *Revision submitted to Journal of Classification*, Jun 2023a.

Jason Hou-Liu and Ryan P. Browne. Extrapolating conditional expectations to accelerate EM procedures. *Submitted to Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Mar 2023b.

Jason Hou-Liu and Ryan P. Browne. Generalized linear models for massive data via doubly-sketching. *Statistics and Computing*, 33(5):105, Jul 2023c. ISSN 1573-1375. doi: 10.1007/s11222-023-10274-8.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2 (1):193–218, December 1985. ISSN 1432-1343. doi: 10.1007/BF01908075. URL https://doi.org/10.1007/BF01908075.

David R Hunter and Kenneth Lange. A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37, 2004. doi: 10.1198/0003130042836. URL https://doi.org/10.1198/0003130042836.

Anil K. Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000. doi: 10.1109/34.824819.

Karl G. Jöreskog. Some contributions to maximum likelihood factor analysis. *Psychometrika*, 32(4):443–482, Dec 1967. ISSN 1860-0980. doi: 10.1007/BF02289658. URL https://doi.org/10.1007/BF02289658.

Henry F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200, September 1958. ISSN 1860-0980. doi: 10.1007/BF02289233. URL https://doi.org/10.1007/BF02289233.

John T. Kent. Identifiability of finite mixtures for directional data. *The Annals of Statistics*, 11(3):984–988, 1983. ISSN 00905364. URL http://www.jstor.org/stable/2240660.

Henk A. L. Kiers and Jos M. F. ten Berge. Minimization of a class of matrix trace functions by means of refined majorization. *Psychometrika*, 57(3):371–382, Sep 1992. ISSN 1860-0980. doi: 10.1007/BF02295425. URL https://doi.org/10.1007/BF02295425.

Henk A.L. Kiers. Setting up alternating least squares and iterative majorization algorithms for solving various matrix optimization problems. *Computational Statistics & Data Analysis*, 41(1):157–170, 2002. ISSN 0167-9473. doi: https://doi.org/10.1016/S0167-9473(02)00142-1. URL https://www.sciencedirect.com/science/article/pii/S0167947302001421. Matrix Computations and Statistics.

Shuichi Kitada, Chie Fujikake, Yoshiho Asakura, Hitomi Yuki, Kaori Nakajima, Kelley M. Vargas, Shiori Kawashima, Katsuyuki Hamasaki, and Hirohisa Kishino. Data from: Molecular and morphological evidence of hybridization between native Ruditapes philippinarum and the introduced Ruditapes form in Japan, 2013a. URL https://doi.org/10.5061/dryad.9c31c.

Shuichi Kitada, Chie Fujikake, Yoshiho Asakura, Hitomi Yuki, Kaori Nakajima, Kelley M. Vargas, Shiori Kawashima, Katsuyuki Hamasaki, and Hirohisa Kishino. Molecular and morphological evidence of hybridization between native Ruditapes philippinarum and the introduced ruditapes form in Japan. *Conservation Genetics*, 14(3):717–733, June 2013b. ISSN 1572-9737. doi: 10.1007/s10592-013-0467-x. URL https://doi.org/10.1007/s10592-013-0467-x.

Ariel Kleiner, Ameet Talwalkar, Purnamrita Sarkar, and Michael I. Jordan. A scalable bootstrap for massive data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4):795–816, 2014. doi: https://doi.org/10.1111/rssb.12050. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/rssb.12050.

Christian Peter Klingenberg and John R. Spence. Heterochrony and allometry: Lessons from the water strider genus limnoporus. *Evolution*, 47(6):1834–1853, 1993.

Sudhir Kylasa, Fred (Farbod) Roosta, Michael W. Mahoney, and Ananth Grama. *GPU Accelerated Sub-Sampled Newton's Method for Convex Classification Problems*, pages 702–710. 2019. doi: 10.1137/1.9781611975673.79. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611975673.79.

Jonathan Lacotte, Sifan Liu, Edgar Dobriban, and Mert Pilanci. Optimal iterative sketching methods with the subsampled randomized hadamard transform. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9725–9735. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/6e69ebbfad976d4637bb4b39de261bf7-Paper.pdf.

Nan M. Laird and James H. Ware. Random-effects models for longitudinal data. *Biometrics*, 38(4):963–974, 1982. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/2529876.

A. Laub. A Schur method for solving algebraic Riccati equations. *IEEE Transactions on Automatic Control*, 24(6):913–921, 1979. doi: 10.1109/TAC.1979.1102178.

John M. Lee. *Smooth Manifolds*, pages 1–31. Springer New York, New York, NY, 2012. ISBN 978-1-4419-9982-5. doi: 10.1007/978-1-4419-9982-5_1. URL https://doi.org/10.1007/978-1-4419-9982-5_1.

JooChul Lee, Elizabeth D. Schifano, and HaiYing Wang. Fast optimal subsampling probability approximation for generalized linear models. *Econometrics and Statistics*, 2021. ISSN 2452-3062. doi: https://doi.org/10.1016/j.ecosta.2021.02.007. URL https://www.sciencedirect.com/science/article/pii/S2452306221000290.

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive data sets*. Cambridge university press, 2020.

Chuanhai Liu and Donald B. Rubin. The ECME algorithm: A simple extension of EM and ECM with faster monotone convergence. *Biometrika*, 81(4):633–648, 12 1994. ISSN 0006-3444. doi: 10.1093/biomet/81.4.633. URL https://doi.org/10.1093/biomet/81.4.633.

Chuanhai Liu, Donald B. Rubin, and Ying Nian Wu. Parameter expansion to accelerate EM: The PX-EM algorithm. *Biometrika*, 85(4):755–770, 12 1998. ISSN 0006-3444. doi: 10.1093/biomet/85.4.755. URL https://doi.org/10.1093/biomet/85.4.755.

Robin H. Lock. 1993 new car data. *Journal of Statistics Education*, 1(1):null, 1993. doi: 10.1080/10691898.1993.11910459. URL https://doi.org/10.1080/10691898.1993.11910459.

Ping Ma and Xiaoxiao Sun. Leveraging for big data regression. *WIREs Computational Statistics*, 7(1):70–76, 2015. doi: https://doi.org/10.1002/wics.1324. URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.1324.

Michael W. Mahoney. Randomized algorithms for matrices and data. *Found. Trends Mach. Learn.*, 3(2):123–224, feb 2011. ISSN 1935-8237. doi: 10.1561/2200000035. URL https://doi.org/10.1561/2200000035.

Matthieu Marbac and Vincent Vandewalle. A tractable multi-partitions clustering. *Computational Statistics & Data Analysis*, 132:167–179, 2019. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2018.06.013. URL https://www.sciencedirect.com/science/article/pii/S0167947318301592. Special Issue on Biostatistics.

Peter McCullagh and John A. Nelder. *Generalized linear models (Second edition)*. London: Chapman & Hall, 1989.

Geoffrey J. McLachlan and Thriyambakam Krishnan. John Wiley & Sons, Ltd, 2008. ISBN 9780470191613. doi: https://doi.org/10.1002/9780470191613.

Paul David McNicholas and Thomas Brendan Murphy. Parsimonious gaussian mixture models. *Statistics and Computing*, 18(3):285–296, September 2008. ISSN 1573-1375. doi: 10.1007/s11222-008-9056-0. URL https://doi.org/10.1007/s11222-008-9056-0.

P.D. McNicholas, T.B. Murphy, A.F. McDaid, and D. Frost. Serial and parallel implementations of model-based clustering via parsimonious gaussian mixture models. *Computational Statistics & Data Analysis*, 54(3):711–723, 2010. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2009.02.011. URL https://www.sciencedirect.com/science/article/pii/S0167947309000632. Second Special Issue on Statistical Algorithms and Software.

Xiao-Li Meng and Donald B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, 06 1993. ISSN 0006-3444. doi: 10.1093/biomet/80.2.267. URL https://doi.org/10.1093/biomet/80.2.267.

Alexander Munteanu, Simon Omlor, and David Woodruff. Oblivious sketching for logistic regression. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7861–7871. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/munteanu21a.html.

Yurii Nesterov. A method for solving the convex programming problem with convergence

rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983. URL https://cir.nii.ac.jp/crid/1571980074954583424.

NYC Taxi and Limousine Commission. TLC trip record data, 2022. URL https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Society for Industrial and Applied Mathematics, January 2000. doi: 10.1137/1.9780898719468. URL https://doi.org/10.1137/1.9780898719468.

Mert Pilanci and Martin J. Wainwright. Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares. *Journal of Machine Learning Research*, 17(53):1–38, 2016. URL http://jmlr.org/papers/v17/14-460.html.

Mert Pilanci and Martin J. Wainwright. Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. 27(1):205–245, January 2017. doi: 10.1137/15m1021106. URL https://doi.org/10.1137/15m1021106.

Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. ISSN 0041-5553. doi: https://doi.org/10.1016/0041-5553(64)90137-5. URL https://www.sciencedirect.com/science/article/pii/0041555364901375.

Jonathan K. Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945, June 2000. URL http://www.genetics.org/content/155/2/945.abstract.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. URL https://www.R-project.org/.

William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971. doi: 10.1080/01621459.1971.10482356. URL https://amstat.tandfonline.com/doi/abs/10.1080/01621459.1971.10482356.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951. ISSN 00034851. URL http://www.jstor.org/stable/2236626.

Herbert Robbins and David Siegmund. A convergence theorem for non negative almost supermartingales and some applications. In Jagdish S. Rustagi, editor, *Optimizing Methods in Statistics*, pages 233–257. Academic Press, 1971. ISBN 978-0-12-604550-5. doi: https://doi.org/10.1016/B978-0-12-604550-5.50015-8. URL https://www.sciencedirect.com/science/article/pii/B9780126045505500158.

Farbod Roosta-Khorasani and Michael W. Mahoney. Sub-sampled Newton methods. *Mathematical Programming*, 174(1):293–326, Mar 2019. ISSN 1436-4646. doi: 10.1007/s10107-018-1346-5. URL https://doi.org/10.1007/s10107-018-1346-5.

Donald B. Rubin and Dorothy T. Thayer. EM algorithms for ML factor analysis. *Psychometrika*, 47(1):69–76, Mar 1982. ISSN 1860-0980. doi: 10.1007/BF02293851. URL https://doi.org/10.1007/BF02293851.

Kegan G. G. Samuel and Marshall F. Tappen. Learning optimized map estimates in continuously-valued mrf models. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 477–484, 2009. doi: 10.1109/CVPR.2009.5206774.

Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 143–152, 2006. doi: 10.1109/FOCS.2006.37.

Gideon Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 03 1978. doi: 10.1214/aos/1176344136. URL https://doi.org/10.1214/aos/1176344136.

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):289–317, 2016. doi: 10.32614/RJ-2016-021. URL https://doi.org/10.32614/RJ-2016-021.

Alexander Shapiro. Identifiability of factor analysis: some results and open problems. *Linear Algebra and its Applications*, 70:1–7, 1985. ISSN 0024-3795.

Marc A. Suchard, Shawn E. Simpson, Ivan Zorych, Patrick Ryan, and David Madigan. Massive parallelization of serial inference algorithms for a complex generalized linear model. *ACM Trans. Model. Comput. Simul.*, 23(1), jan 2013. ISSN 1049-3301. doi: 10.1145/2414416.2414791. URL https://doi.org/10.1145/2414416.2414791.

M. J. Symons. Clustering criteria and multivariate normal mixtures. *Biometrics*, 37(1): 35–43, 1981. ISSN 0006341X, 15410420. doi: https://doi.org/10.2307/2530520. URL http://www.jstor.org/stable/2530520.

Takahashi. Ruditapes philippinarum, 2006. URL https://en.wikipedia.org/wiki/File:Ruditapes_philippinarum.jpg. [Online; accessed August 1, 2019].

Henry Teicher. Maximum likelihood characterization of distributions. *Ann. Math. Statist.*, 32(4):1214–1222, 12 1961. doi: 10.1214/aoms/1177704861. URL https://doi.org/10.1214/aoms/1177704861.

Martijn van Breukelen and Robert P. W. Duin. Neural network initialization by combined classifiers. In *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*, volume 1, pages 215–218 vol.1, 1998. doi: 10.1109/ICPR.1998.711119.

Martijn van Breukelen, Robert P. W. Duin, David M. J. Tax, and J. E. Den Hartog. Handwritten digit recognition by combined classifiers. *Kybernetika*, 34(4):381–386, 1998.

Ravi Varadhan and Christophe Roland. Simple and globally convergent methods for accelerating the convergence of any em algorithm. *Scandinavian Journal of Statistics*, 35(2):335–353, 2008. doi: https://doi.org/10.1111/j.1467-9469.2007.00585.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9469.2007.00585.x.

William N. Venables and Brian D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. URL https://www.stats.ox.ac.uk/pub/MASS4/. ISBN 0-387-95457-0.

Jeroen K. Vermunt. Multilevel latent class models. *Sociological Methodology*, 33(1):213–239, 2003. doi: https://doi.org/10.1111/j.0081-1750.2003.t01-1-00131.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.0081-1750.2003.t01-1-00131.x.

Multivariable Datenanalyse zur Sortenklassifizierung von Weinen. Multivariate data analysis as a discriminating method of the origin of wines. *Vitis*, 25:189–201, 1986.

HaiYing Wang, Rong Zhu, and Ping Ma. Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association*, 113(522):829–844, 2018. doi: 10.1080/01621459.2017.1292914. URL https://doi.org/10.1080/01621459.2017.1292914. PMID: 30078922.

HaiYing Wang, Min Yang, and John Stufken. Information-based optimal subdata selection for big data linear regression. *Journal of the American Statistical Association*, 114(525): 393–405, 2019. doi: 10.1080/01621459.2017.1408468. URL https://doi.org/10.1080/01621459.2017.1408468.

Robert W. M. Wedderburn. On the existence and uniqueness of the maximum likelihood estimates for certain generalized linear models. *Biometrika*, 63(1):27–32, 04 1976. ISSN 0006-3444. doi: 10.1093/biomet/63.1.27. URL https://doi.org/10.1093/biomet/63.1.27.

John Harmon Wolfe. *Object cluster analysis of social areas*. PhD thesis, University of California, 1963.

Max A. Woodbury, Jonathan Clive, and Arthur Garson. Mathematical typology: A grade of membership technique for obtaining disease definition. *Computers and Biomedical Research*, 11(3):277–298, 1978. ISSN 0010-4809. doi: https://doi.org/10.1016/0010-4809(78)90012-5. URL http://www.sciencedirect.com/science/article/pii/0010480978900125.

Peng Xu, Jiyan Yang, Fred Roosta, Christopher Ré, and Michael W Mahoney. Sub-sampled Newton methods with non-uniform sampling. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*,

volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/55c567fd4395ecef6d936cf77b8d5b2b-Paper.pdf.

Sidney J. Yakowitz and John D. Spragins. On the identifiability of finite mixtures. *Ann. Math. Statist.*, 39(1):209–214, 02 1968. doi: 10.1214/aoms/1177698520. URL https://doi.org/10.1214/aoms/1177698520.

Jun Yu, HaiYing Wang, Mingyao Ai, and Huiming Zhang. Optimal distributed subsampling for maximum quasi-likelihood estimators with massive data. *Journal of the American Statistical Association*, 117(537):265–276, 2022a. doi: 10.1080/01621459.2020.1773832. URL https://doi.org/10.1080/01621459.2020.1773832.

Jun Yu, HaiYing Wang, Mingyao Ai, and Huiming Zhang. Optimal distributed subsampling for maximum quasi-likelihood estimators with massive data. *Journal of the American Statistical Association*, 117(537):265–276, 2022b. doi: 10.1080/01621459.2020.1773832. URL https://doi.org/10.1080/01621459.2020.1773832.

Jian Zhang. Epistatic clustering: A model-based approach for identifying links between clusters. *Journal of the American Statistical Association*, 108(504):1366–1384, 2013. doi: 10.1080/01621459.2013.835661. URL https://doi.org/10.1080/01621459.2013.835661.

APPENDICES

Appendix A

Chimeral Clustering

A.1 Proof of Lemma 1

(i) From Section 3.1 and 3.6 of Grünbaum (2003), the convex hull $\text{conv}(V)$ is a polytope whose $\mathcal{V}$-representation is a set of vertices. A similar result can be found in Section 14.1 of Gruber (2007).

(ii) Uniqueness of $\text{conv}(V)$ is given by the fact that intersection of convex sets is convex. If A and B are two different smallest convex hulls of V then $A \cap B$ contains V and so is a smaller convex hull; a contradiction.

(iii) Finally, Section 14.1 of Gruber (2007) provides us with the result that the extreme points of $\text{conv}(V)$ are a subset of the vertices in V.

A.2 d-Radioactive Dataset

We describe here the method for generating the d-Radioactive dataset used in the simulation study. Here, d represents the dimension of the data. In two dimensions, we find that the data distribution resembles the radioactivity sign and is so named. In higher dimensions, the sketch of the mixture density is a regular d-simplex (triangle, tetrahedron, and so

forth) with the prototypes densities roughly forming the $d-1$ dimension facets. Chimeral clusters are formed by taking three parts of one prototype and one part of all other prototypes for each prototype, with an extra cluster being equal parts of all prototypes. A constructive description follows.

For $d \geq 2$, define $d+1$ prototype clusters with their means being vertices of a regular d-dimensional simplex centered at the origin in $\mathbb{R}^d$ with radius $\sqrt{200d}$ (distance of each vertex to the origin, equivalently radius of the circumscribed d-sphere). Define $\boldsymbol{\mu}_i = \langle \mu_{i,1}, \mu_{i,2}, ..., \mu_{i,d} \rangle$ and construct successive vertices as follows.

- Set $\mu_{1,1} = 1$ and $\mu_{i,1} = -\frac{1}{d}$ for $i = 2, 3, ..., d+1$.
- For i in $2, ..., d+1$:
  - Set $\mu_{i,i} = \sqrt{1 - \sum_{j=1}^{i-1} \mu_{i,j}^2}$.
  - Set $\mu_{i+1,i}, ..., \mu_{i,d} = -\frac{1}{\mu_{i,i}} \left(\frac{1}{d} + \sum_{j=1}^{i-1} \mu_{i,j} \right)$.

For each prototype, define the covariance matrix $\boldsymbol{\Sigma}_i = 100\mathbf{I} - (100 - r)\boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top$ for some parameter $0 < r < 100$. In order to preserve the separation of the clusters, r should be considerably lower than 100. Let the natural parameterization of the prototype distributions be $(\boldsymbol{\eta}_i, \boldsymbol{\Lambda}_i)$. Define $d+2$ chimeral clusters, with the first $j = 1, 2, ..., d+1$ being parameterized by $\boldsymbol{\alpha}_j = \frac{1}{d+3}\mathbf{1}_{d+1} + \frac{2}{d+3}\boldsymbol{e}_j$ for standard basis vectors $\boldsymbol{e}_j$. The last chimeral cluster is parameterised by $\boldsymbol{\alpha}_j = \frac{1}{d+1}\mathbf{1}_{d+1}$.

For $d = 2$, the parameters in numerical form are:

$$
\begin{aligned}
\boldsymbol{\eta}_{P_1} &= \langle 20, 0 \rangle \\
\boldsymbol{\eta}_{P_2} &= \langle -10, 17.3205 \rangle \\
\boldsymbol{\eta}_{P_3} &= \langle -10, -17.3205 \rangle \\
\boldsymbol{\Lambda}_{P_1} &= \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix} \\
\boldsymbol{\Lambda}_{P_2} &= \begin{bmatrix} 0.2575 & -0.4287 \\ -0.4287 & 0.7525 \end{bmatrix} \\
\boldsymbol{\Lambda}_{P_3} &= \begin{bmatrix} 0.2575 & 0.4287 \\ 0.4287 & 0.7525 \end{bmatrix} \\
\boldsymbol{\alpha}_{C_1} &= \langle 0.6, 0.2, 0.2 \rangle \\
\boldsymbol{\alpha}_{C_2} &= \langle 0.2, 0.6, 0.2 \rangle \\
\boldsymbol{\alpha}_{C_3} &= \langle 0.2, 0.2, 0.6 \rangle \\
\boldsymbol{\alpha}_{C_4} &= \langle 0.\bar{3}, 0.\bar{3}, 0.\bar{3} \rangle
\end{aligned}
$$

A plot of 1000 observations drawn from each cluster is given in Figure A.1. A three-dimensional version is visualized in Figure A.2 with 200 observations per cluster. In both figures, $r = 1$.

A.2.1 Extended Simulation Results

A selection of these results are given in tabular form in the main work. We present here the full simulation results for $d = 2, 3, ..., 10$, $n = 20, 40, ..., 100$, and $r = 1, 5, 10$.

Cosine Similarity

The cosine similarities are defined for chimeral clustering as follows. Let $\boldsymbol{\alpha}_c$ be the true hybridization weights for cluster $c \in \mathcal{C}$ and let $\hat{\boldsymbol{\alpha}}_c$ be the estimated hybridization weights

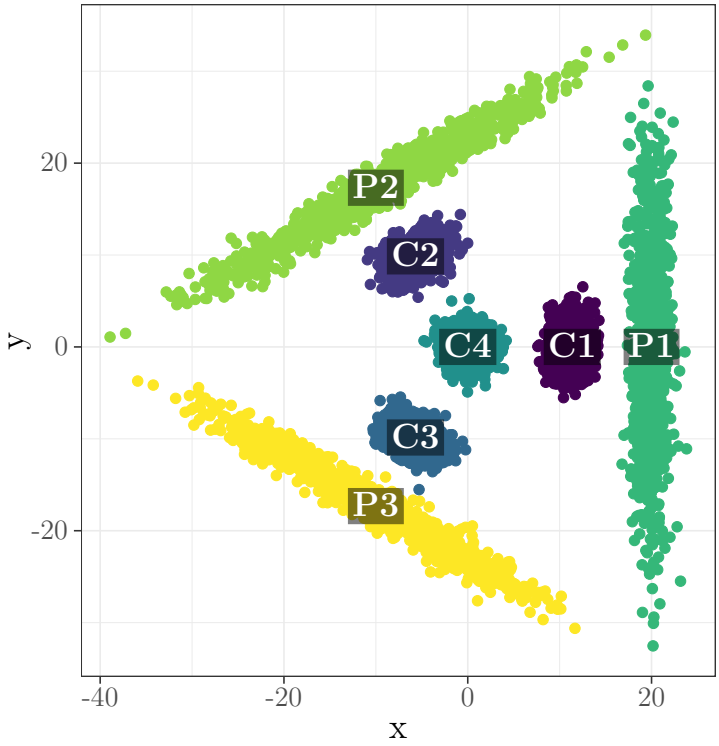

Figure A.1: 2-dimensional radioactive dataset, 1000 observations per cluster. $r = 1$.

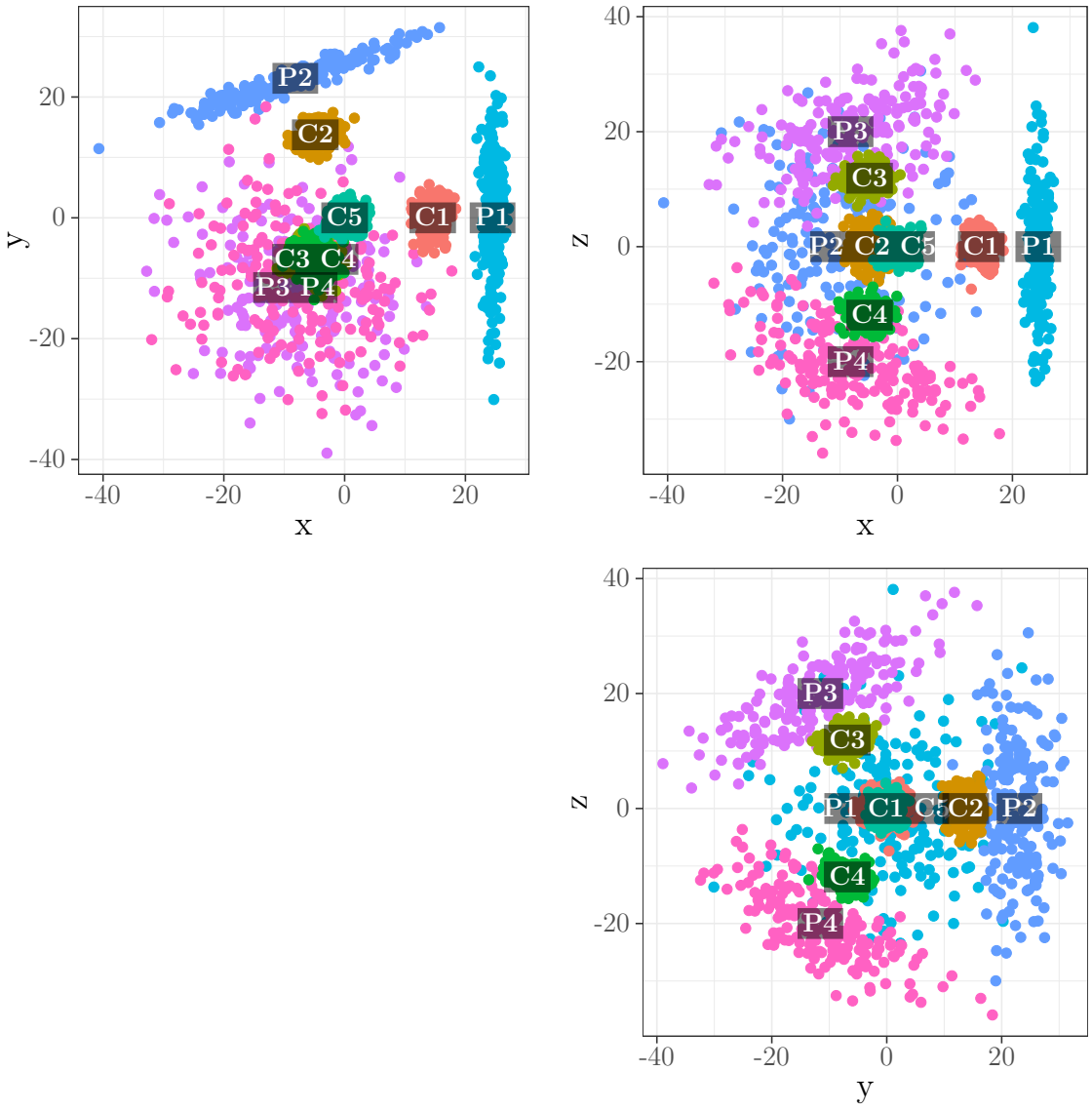

Figure A.2: 3-dimensional radioactive dataset, 200 observations per cluster. $r = 1$.

from the estimation procedure. Then, the cosine similarity of the entire fitted model could be computed as

$$\frac{\begin{bmatrix} \boldsymbol{\alpha}_{C_1} \\ \vdots \\ \boldsymbol{\alpha}_{C_{K_C}} \end{bmatrix}^\top \begin{bmatrix} \hat{\boldsymbol{\alpha}}_{C_1} \\ \vdots \\ \hat{\boldsymbol{\alpha}}_{C_{K_C}} \end{bmatrix}}{\left\| \begin{bmatrix} \boldsymbol{\alpha}_{C_1} \\ \vdots \\ \boldsymbol{\alpha}_{C_{K_C}} \end{bmatrix} \right\|_2 \left\| \begin{bmatrix} \hat{\boldsymbol{\alpha}}_{C_1} \\ \vdots \\ \hat{\boldsymbol{\alpha}}_{C_{K_C}} \end{bmatrix} \right\|_2}.$$

If the two sets of weights $\{\boldsymbol{\alpha}_c\}_{c\in\mathcal{C}}$ and $\{\hat{\boldsymbol{\alpha}}_c\}_{c\in\mathcal{C}}$ coincide, then the angle formed in between them is zero and so their cosine similarity is one. As the estimated vector deviates, the similarity metric decreases towards zero. However, this process requires the estimated indices $C_1, ..., C_{K_C}$ to match the true indices, something not guaranteed by the estimation procedure. Thus, we permute the estimated weights to maximize this quantity.

A.3 Extra Datasets

In this section, we include some datasets investigated as part of the process. They are provided as information only.

A.3.1 Yeast dataset

The yeast stress dataset (Gasch et al., 2000) is used as a real-world dataset in the prior epistatic clustering work by Zhang (2013). It describes the changes in gene expression of the yeast *Saccharomyces cerevisiae* in response to changes in the environmental conditions experienced by the cells. We attempt to replicate the same dataset by following the data pre-processing step described therein (Zhang, 2013, Section 5.1).

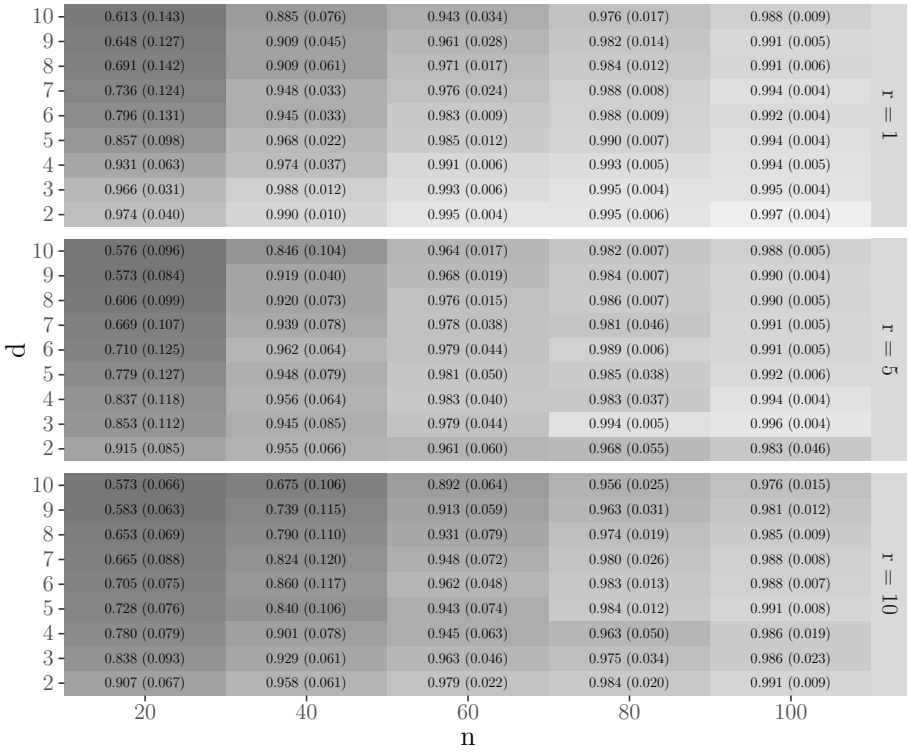

Figure A.3: Cosine similarity values measuring the $\boldsymbol{\alpha}_c$ parameter recovery in the d-radioactive dataset over a range of data dimensions d, number of observations n, and prototype sphericity r. Higher values are better. Standard deviations over fifty replications in brackets.

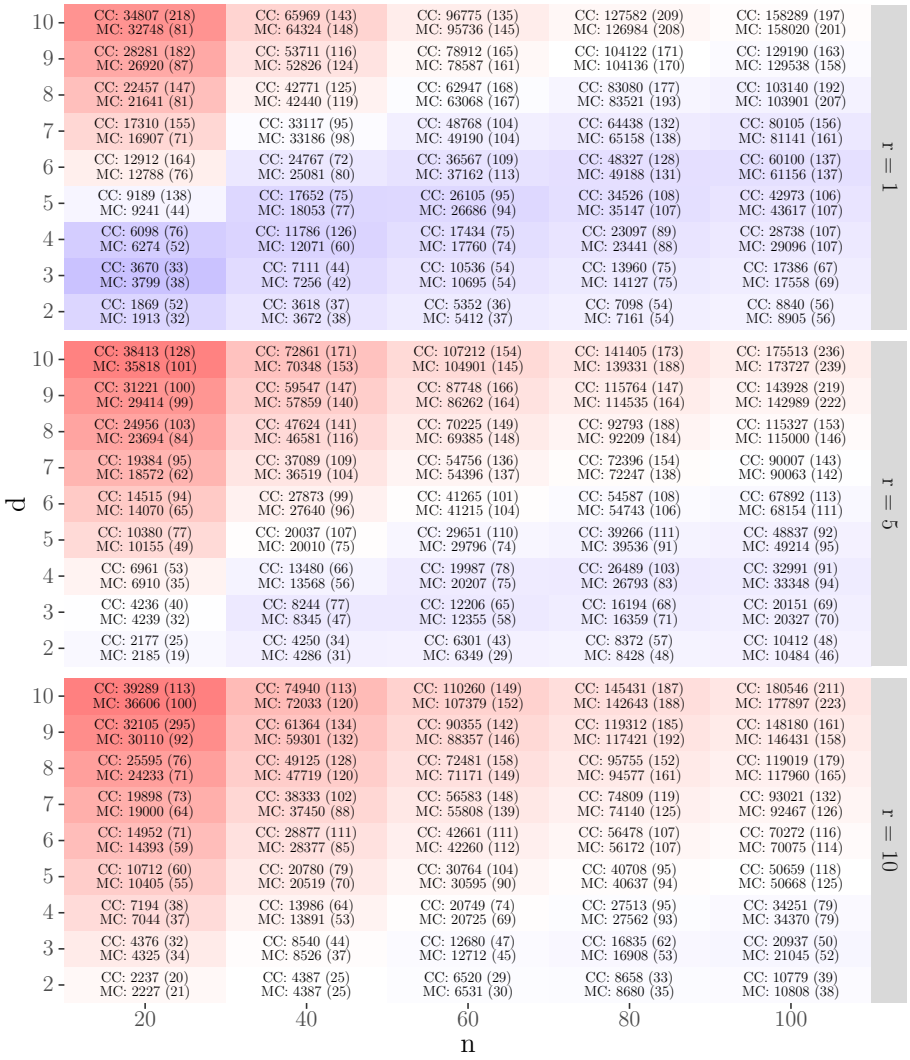

Figure A.4: Bayesian Information Criterion values for the d-radioactive dataset over multiple parameter combinations; lower values are better. Standard deviations over fifty replications in brackets. Red indicates better *mclust*, blue indicates better chimeral clustering performance.

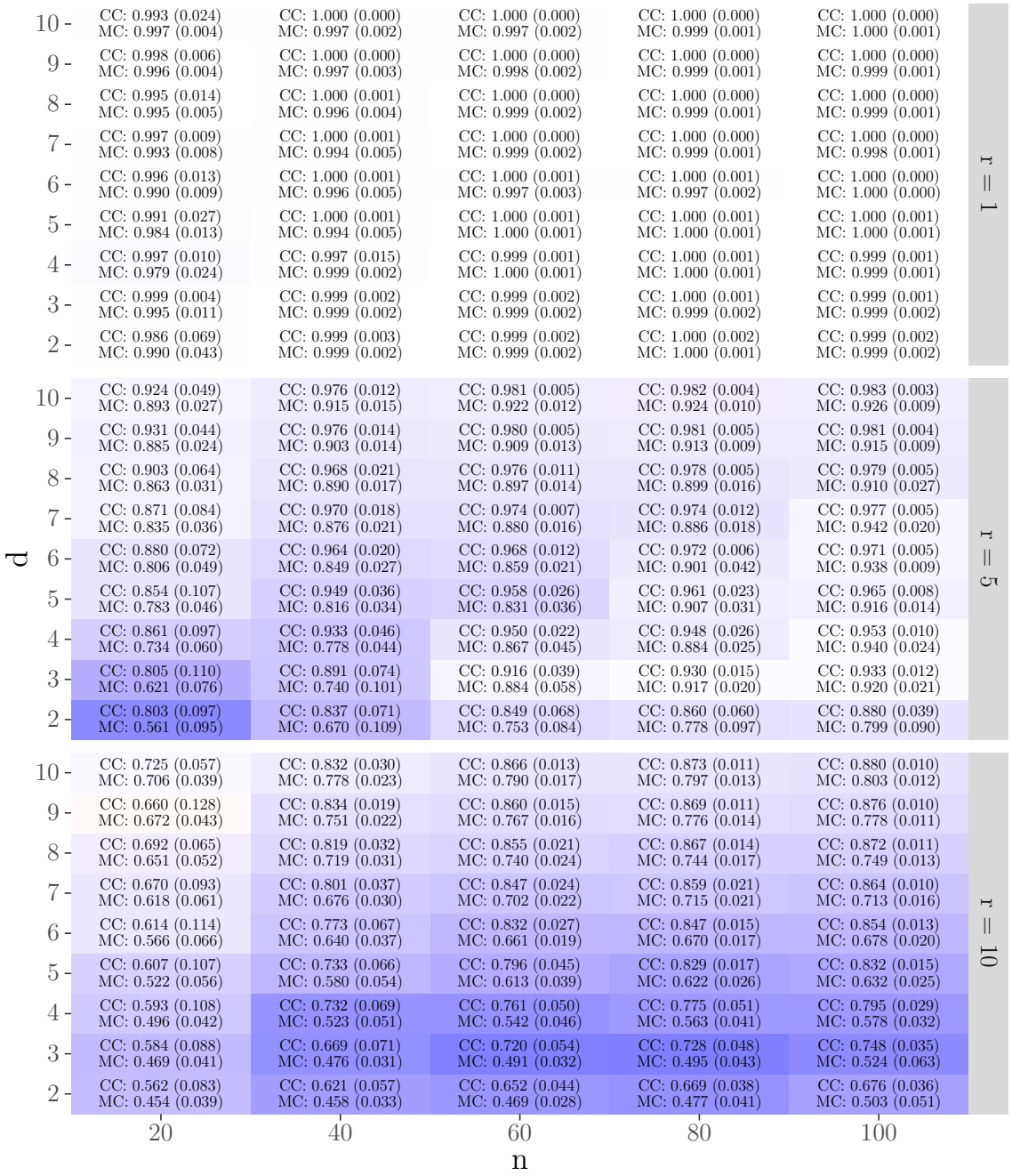

Figure A.5: Adjusted Rand index values for the d-radioactive dataset over multiple parameter combinations; higher values are better. Standard deviations over fifty replications in brackets. Red indicates better *mclust*, blue indicates better chimeral clustering performance.

We begin with the dataset of Gasch et al. (2000) containing 6152 observations representing genes and 173 variables representing environmental conditions. We use the same 15 variables as described by Zhang (2013), titled with the prefix "Heat Shock" and suffixed "hs-1" or "hs-2" as found in columns 4 through 19, inclusive. There are missing values in this subset of dataset and it is unclear how Zhang (2013) treats these cases; we leave incomplete observations in the dataset at this point. Subsequently, we remove noisy observations as done so by Zhang (2013), calculating a sample variance over each row. Due to data missingness, we compute the sample variance $\hat{\sigma}_i$ over the non-missing columns for each gene i. There is a single observation (YDL208W) with 14 missing values, leading to an undefined sample variance; we remove this observation. Let $S_1 = \{\hat{\sigma}_i \mid i = 1, 2, ..., 6151\}$ denote the set of sample variances. We select the subset of S_1 "within three-folds of the minimum sample variance" (Zhang, 2013) to form $S_2 = \{\hat{\sigma} \mid \hat{\sigma} \in S_1, \hat{\sigma} \leq 3 \times \min_{s \in S_1} s\}$ with $|S_2| = 169$. We define $\hat{\sigma}_0$ as the sample average over S_2, and construct the sample variability index v_i over the 6151 genes as $v_i = \hat{\sigma}_i / \hat{\sigma}_0$ for $i = 1, 2, ..., 6151$. Finally, we choose all genes i such that $v_i > 9$; this leaves 2294 genes. Since there are still missing values for these genes, we retain only complete cases for a final number of 1364 genes. By contrast, Zhang (2013) claim to have 496 genes at the end of the procedure. If we remove incomplete cases after selecting the 15 desired variables, we obtain 1361 genes at the end. If we remove incomplete cases from the entire dataset of 173 variables, we obtain 258 genes at the end. We approximately verify that our choice of 15 variables is correct by noting similar characteristics in the pairwise scatterplot (Zhang, 2013, Figure 1). Overlooking this discrepancy, we proceed with the application of chimeral clustering using the dataset of 1364 genes.

We evaluate for $K_P + K_C \leq 13$, running mini-EM for 5000 iterations and holding $\hat{z}_{nk}$ fixed for 1000 of them. We exclude models that have covariances with eigenvalues $\leq 10^{-8}$ and $N_k \leq 10^{-8}$ for any $g \in \mathcal{C} \cup \mathcal{P}$. We then run an additional 5000 iterations on the best starter model. The resultant BICs for each combination of K_P sand K_C are presented in Figure A.6.

Zhang (2013) obtains an epistatic clustering result with four primary clusters, three epistatic clusters, and a miscellaneous cluster. By contrast, we obtain three prototypes and eight chimeral clusters. The fitted model metrics are given in Table A.1 without ARI due

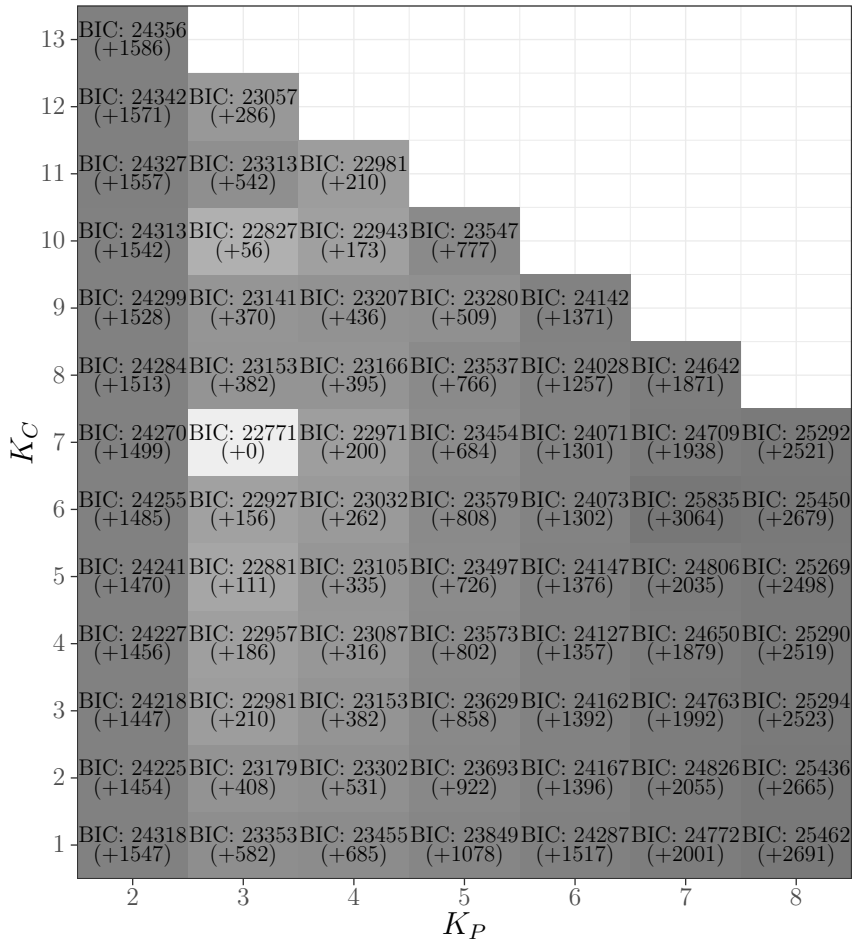

Figure A.6: Minimum Bayesian Information Criterion for multiple choices of prototype clusters K_P and chimeral clusters K_C over 100 runs each of the *Saccharomyces cerevisiae* dataset. Graph truncated to $K_P \leq 8$ for presentation.

		mclust	
	Chimeral Clustering	VVV	VEE
Number of Clusters	10 $(K_P = 3)$	4	10
Number of Parameters	428	543	**288**
Log-Likelihood	−9840.69	**−9855.30**	−10 489.53
BIC	**22 770.76**	23 630.07	23 057.90

Table A.1: Fitted model metrics for yeast dataset with up to 13 clusters, best value in bold.

to a lack of class labels. Again, we compare against the best fully varying covariance matrix Gaussian mixture and the best parsimonious covariance matrix Gaussian mixture fitted by mclust. Figure A.7 shows the $\boldsymbol{\alpha}_c$ quantities for each chimeral cluster. We can see the ability of chimeral clustering to better adapt to varying number of parents and unbalanced mixing proportions compared to the pre-specified values in epistatic clustering. With this dataset, we note that the chimeral clustering BIC outperforms both the parsimonious Gaussian mixture model and covariance VVV model.

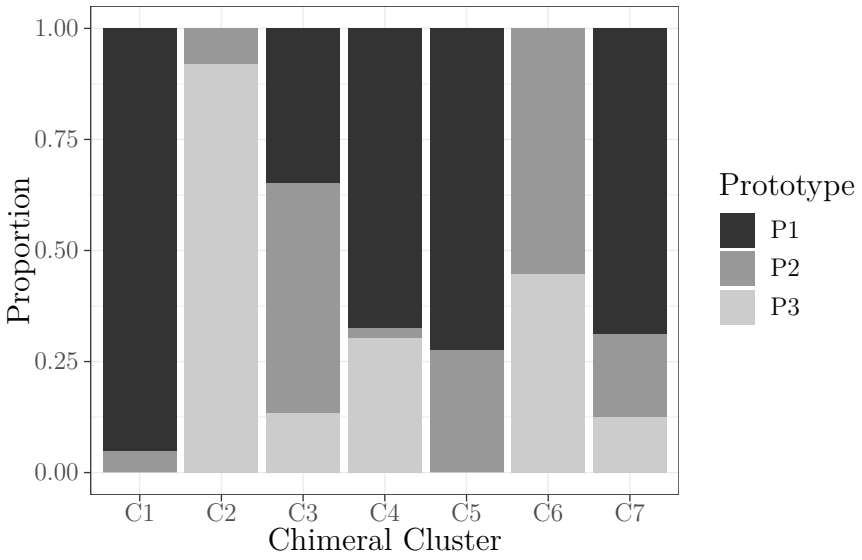

Figure A.7: Mixing proportions $\boldsymbol{\alpha}_c$ for each of the eight chimeral clusters over the three prototype clusters for the *Saccharomyces cerevisiae* dataset. Both two and three parent clusters are visible.

Appendix B

Factor and Hybrid Components for Model-Based Clustering

B.1 Expectation

We obtain the expected incomplete data log-likelihood (B.1).

$$\begin{aligned}
\mathrm{E}\left[\ell\left(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}\right)\right] &= \sum_{n=1}^{N} \Bigg[\sum_{f \in \mathcal{F}} \hat{z}_{nf} \left[\log \pi_f + \log \phi_f\left(\boldsymbol{x}_n\right)\right] \\
&\quad + \sum_{h \in \mathcal{H}} \hat{z}_{nh} \left\{\log \pi_h + \mathrm{E}\left[\log \phi_h\left(\boldsymbol{x}_n\right) \mid \boldsymbol{x}_n, z_{nh} = 1\right] \right. \\
&\quad \left. + \mathrm{E}\left[\sum_{f \in \mathcal{F}} \log \phi_f(\boldsymbol{y}_{nf}) \mid Y_h = \boldsymbol{x}_n, z_{nh} = 1\right] \right\} \Bigg]
\end{aligned}$$

Substituting in the appropriate multivariate normal densities produces the necessary surrogate function Q. We omit expressing the condition to reduce notational load. While Q appears unpalatable, it fortunately results in a tractable subsequent maximization step.

$$
\begin{aligned}
Q\left(\boldsymbol{\theta}\right) \;=\; & \text{constant} + \sum_{n=1}^{N}\left[\sum_{f\in\mathcal{F}}\hat{z}_{nf}\left\{\log\pi_f - \frac{1}{2}\log\left|\boldsymbol{\Sigma}_f\right| - \frac{1}{2}\operatorname{Tr}\left[\boldsymbol{\Sigma}_f^{-1}\left(\boldsymbol{x}_n-\boldsymbol{\mu}_f\right)\left(\boldsymbol{x}_n-\boldsymbol{\mu}_f\right)^{\top}\right]\right\}\right. \\
& + \sum_{h\in\mathcal{H}}\hat{z}_{nh}\left\{\log\pi_h - \frac{1}{2}\log\left|\boldsymbol{\Psi}_h\right|\right. \\
& - \frac{1}{2}\operatorname{Tr}\left[\boldsymbol{\Psi}_h^{-1}\,\mathrm{E}\left[\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)^{\top}\right]\right] \\
& \left.\left. + \sum_{f\in\mathcal{F}}\left(-\frac{1}{2}\log\left|\boldsymbol{\Sigma}_f\right| - \frac{1}{2}\operatorname{Tr}\left[\boldsymbol{\Sigma}_f^{-1}\,\mathrm{E}\left[\left(\boldsymbol{y}_{nf}-\boldsymbol{\mu}_f\right)\left(\boldsymbol{y}_{nf}-\boldsymbol{\mu}_f\right)^{\top}\right]\right]\right)\right\}\right]
\end{aligned}
$$

It remains to evaluate the two expectations on latent $\boldsymbol{y}_{nf}$'s, namely

$$
\mathrm{E}\left[\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)^{\top}\right] \text{ and } \mathrm{E}\left[\left(\boldsymbol{y}_{nf}-\boldsymbol{\mu}_f\right)\left(\boldsymbol{y}_{nf}-\boldsymbol{\mu}_f\right)^{\top}\right],
$$

which can be expanded into an expression involving expectations on $\boldsymbol{y}_{nf}$ and $\boldsymbol{y}_{nf}\boldsymbol{y}_{nq}^{\top}$ for $f, q \in \mathcal{F}$. Hence, conditional on assignment $z_{nh} = 1$, we first obtain the joint distribution (B.1) of each $\boldsymbol{y}_{nf}$ and the hybrid Y_h.

$$
\begin{bmatrix} Y_h \\ Y_{f_1} \\ \vdots \\ Y_{f_F} \end{bmatrix} \mid z_{nh} = 1 \sim N\left(\begin{bmatrix} \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{\mu}_f \\ \boldsymbol{\mu}_{f_1} \\ \vdots \\ \boldsymbol{\mu}_{f_F} \end{bmatrix}, \begin{bmatrix} \sum_{f\in\mathcal{F}}\alpha_{hf}^2\boldsymbol{\Sigma}_f + \boldsymbol{\Psi}_h & \alpha_{hf_1}\boldsymbol{\Sigma}_{f_1} & \cdots & \alpha_{hf_F}\boldsymbol{\Sigma}_{f_F} \\ \alpha_{hf_1}\boldsymbol{\Sigma}_{f_1} & \boldsymbol{\Sigma}_{f_1} & & 0 \\ \vdots & & \ddots & \\ \alpha_{hf_F}\boldsymbol{\Sigma}_{f_F} & 0 & & \boldsymbol{\Sigma}_{f_F} \end{bmatrix}\right) \tag{B.1}
$$

Given that we observe $Y_h = \boldsymbol{x}_n$, we use standard properties of the multivariate normal distribution to obtain the conditional distribution of the latent prototype parameters $\boldsymbol{y}_{nf} \mid Y_h = \boldsymbol{x}_n, z_{nh} = 1$:

$$
\begin{aligned}
&\mathrm{E}\left[\begin{bmatrix} Y_{f_1} \\ \vdots \\ Y_{f_F} \end{bmatrix} \mid Y_h = \boldsymbol{x}_n, z_{nh} = 1\right] \\
&= \begin{bmatrix} \boldsymbol{\mu}_{f_1} \\ \vdots \\ \boldsymbol{\mu}_{f_F} \end{bmatrix} + \begin{bmatrix} \alpha_{hf_1}\boldsymbol{\Sigma}_{f_1} \\ \vdots \\ \alpha_{hf_F}\boldsymbol{\Sigma}_{f_F} \end{bmatrix} \left(\sum_{f\in\mathcal{F}} \alpha_{hf}^2 \boldsymbol{\Sigma}_f + \boldsymbol{\Psi}_h\right)^{-1} \left(\boldsymbol{x}_n - \sum_{g\in\mathcal{F}} \alpha_{hg}\boldsymbol{\mu}_g\right) \\
&\mathrm{Var}\left[\begin{bmatrix} Y_{f_1} \\ \vdots \\ Y_{f_F} \end{bmatrix} \mid Y_h = \boldsymbol{x}_n, z_{nh} = 1\right] \\
&= \begin{bmatrix} \boldsymbol{\Sigma}_{f_1} & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{\Sigma}_{f_F} \end{bmatrix} - \begin{bmatrix} \alpha_{hf_1}\boldsymbol{\Sigma}_{f_1} \\ \vdots \\ \alpha_{hf_F}\boldsymbol{\Sigma}_{f_F} \end{bmatrix} \left(\sum_{f\in\mathcal{F}} \alpha_{hf}^2 \boldsymbol{\Sigma}_f + \boldsymbol{\Psi}_h\right)^{-1} \begin{bmatrix} \alpha_{hf_1}\boldsymbol{\Sigma}_{f_1} & \cdots & \alpha_{hf_F}\boldsymbol{\Sigma}_{f_F} \end{bmatrix}
\end{aligned}
$$

We note that the expectation breaks apart nicely to give for each $f \in \mathcal{F}$, which we assign the shorthand $\bar{\boldsymbol{y}}_{nfh}$ for convenience:

$$
\mathrm{E}\left[Y_f \mid Y_h = \boldsymbol{x}_n, z_{nh} = 1\right] = \boldsymbol{\mu}_f + \alpha_{hf}\boldsymbol{\Sigma}_f \left(\sum_{g\in\mathcal{F}} \alpha_{hg}^2 \boldsymbol{\Sigma}_g + \boldsymbol{\Psi}_h\right)^{-1} \left(\boldsymbol{x}_n - \sum_{g\in\mathcal{F}} \alpha_{hg}\boldsymbol{\mu}_g\right)
$$

However, the variance does not decompose into a block-diagonal matrix, and so we must handle the expectation of the cross-term $\mathrm{E}\left[Y_f Y_q^\top \mid Y_h = \boldsymbol{x}_n\right]$ for $f, q \in \mathcal{F}$ in a more tedious way:

$$
\mathrm{E}\left[Y_f Y_q^\top \mid Y_h = \boldsymbol{x}_n, z_{nh} = 1\right] = \underbrace{\mathbb{1}_{(f=q)}\boldsymbol{\Sigma}_f - \alpha_{hf}\alpha_{hq}\boldsymbol{\Sigma}_f \left(\sum_{g\in\mathcal{F}} \alpha_{hg}^2 \boldsymbol{\Sigma}_g + \boldsymbol{\Psi}_h\right)^{-1} \boldsymbol{\Sigma}_q}_{:=\mathbf{S}_{hfq}} + \bar{\boldsymbol{y}}_{nfh}\bar{\boldsymbol{y}}_{nqh}^\top
$$

Hence, we expand may re-write the two expectations in question in the following form:

$$
\begin{aligned}
\mathrm{E}\left[\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}} \alpha_{hf}\boldsymbol{y}_{nf}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}} \alpha_{hf}\boldsymbol{y}_{nf}\right)^\top\right] &= \left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}} \alpha_{hf}\bar{\boldsymbol{y}}_{nfh}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}} \alpha_{hf}\bar{\boldsymbol{y}}_{nfh}\right)^\top \\
&\quad + \sum_{f\in\mathcal{F}}\sum_{q\in\mathcal{F}} \alpha_{hf}\alpha_{hq}\mathbf{S}_{hfq} \\
\mathrm{E}\left[\left(\boldsymbol{y}_{nf} - \boldsymbol{\mu}_f\right)\left(\boldsymbol{y}_{nf} - \boldsymbol{\mu}_f\right)^\top\right] &= \left(\bar{\boldsymbol{y}}_{nfh} - \boldsymbol{\mu}_f\right)\left(\bar{\boldsymbol{y}}_{nfh} - \boldsymbol{\mu}_f\right)^\top + \mathbf{S}_{hff}
\end{aligned}
$$

We thus obtain a final expression suitable for the maximization step of the EM algorithm after making the above substitution, simplifying the expression somewhat, and dropping additive constants:

$$
\begin{aligned}
Q\left(\boldsymbol{\theta}\right) &= \sum_{n=1}^{N}\left(\sum_{f\in\mathcal{F}} \hat{z}_{nf}\log\pi_f + \sum_{h\in\mathcal{H}} \hat{z}_{nh}\log\pi_h\right) \\
&\quad + \frac{1}{2}\sum_{n=1}^{N}\left[\sum_{f\in\mathcal{F}} \hat{z}_{nf}\left\{\log\left|\boldsymbol{\Sigma}_f^{-1}\right| - \mathrm{Tr}\left[\boldsymbol{\Sigma}_f^{-1}\left(\boldsymbol{x}_n - \boldsymbol{\mu}_f\right)\left(\boldsymbol{x}_n - \boldsymbol{\mu}_f\right)^\top\right]\right\}\right. \\
&\quad + \sum_{h\in\mathcal{H}} \hat{z}_{nh}\left\{\log\left|\boldsymbol{\Psi}_h^{-1}\right| - \mathrm{Tr}\left[\boldsymbol{\Psi}_h^{-1}\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}} \alpha_{hf}\bar{\boldsymbol{y}}_{nfh}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}} \alpha_{hf}\bar{\boldsymbol{y}}_{nfh}\right)^\top\right.\right. \\
&\quad \left.\left. + \boldsymbol{\Psi}_h^{-1}\sum_{f\in\mathcal{F}}\sum_{q\in\mathcal{F}} \alpha_{hf}\alpha_{hq}\mathbf{S}_{hfq}\right]\right. \\
&\quad \left. + \sum_{f\in\mathcal{F}}\left(\log\left|\boldsymbol{\Sigma}_f^{-1}\right| - \mathrm{Tr}\left[\boldsymbol{\Sigma}_f^{-1}\left(\bar{\boldsymbol{y}}_{nfh} - \boldsymbol{\mu}_f\right)\left(\bar{\boldsymbol{y}}_{nfh} - \boldsymbol{\mu}_f\right)^\top + \boldsymbol{\Sigma}_f^{-1}\mathbf{S}_{hff}\right]\right)\right\}\right]
\end{aligned}
$$

B.2 Maximization

B.2.1 Maximizing in Factor Means

$$
\begin{aligned}
\frac{\partial Q}{\partial \boldsymbol{\mu}_f} &= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\mu}_f}\sum_{n=1}^{N}\left[\hat{z}_{nf}\left\{-\operatorname{Tr}\left[\boldsymbol{\Sigma}_f^{-1}\left(\boldsymbol{x}_n-\boldsymbol{\mu}_f\right)\left(\boldsymbol{x}_n-\boldsymbol{\mu}_f\right)^{\top}\right]\right\}\right. \\
&\quad \left.+\sum_{h\in\mathcal{H}}\hat{z}_{nh}\left(-\operatorname{Tr}\left[\boldsymbol{\Sigma}_f^{-1}\operatorname{E}\left[\left(\boldsymbol{y}_{nf}-\boldsymbol{\mu}_f\right)\left(\boldsymbol{y}_{nf}-\boldsymbol{\mu}_f\right)^{\top}\right]\right]\right)\right] \\
&= \frac{1}{2}\sum_{n=1}^{N}\left[\hat{z}_{nf}\left\{2\boldsymbol{\Sigma}_f^{-1}\left(\boldsymbol{x}_n-\boldsymbol{\mu}_f\right)\right\}+\sum_{h\in\mathcal{H}}\hat{z}_{nh}\left\{2\boldsymbol{\Sigma}_f^{-1}\left(\bar{\boldsymbol{y}}_{nfh}-\boldsymbol{\mu}_f\right)\right\}\right] \\
&= \sum_{n=1}^{N}\left[\hat{z}_{nf}\boldsymbol{\Sigma}_f^{-1}\left(\boldsymbol{x}_n-\boldsymbol{\mu}_f\right)+\sum_{h\in\mathcal{H}}\hat{z}_{nh}\boldsymbol{\Sigma}_f^{-1}\left(\bar{\boldsymbol{y}}_{nfh}-\boldsymbol{\mu}_f\right)\right] \\
\Rightarrow \mathbf{0} &= \sum_{n=1}^{N}\left[\hat{z}_{nf}\boldsymbol{\Sigma}_f^{-1}\left(\boldsymbol{x}_n-\boldsymbol{\mu}_f\right)+\sum_{h\in\mathcal{H}}\hat{z}_{nh}\boldsymbol{\Sigma}_f^{-1}\left(\bar{\boldsymbol{y}}_{nfh}-\boldsymbol{\mu}_f\right)\right] \\
\mathbf{0} &= \sum_{n=1}^{N}\left[\hat{z}_{nf}\left(\boldsymbol{x}_n-\boldsymbol{\mu}_f\right)+\sum_{h\in\mathcal{H}}\hat{z}_{nh}\left(\bar{\boldsymbol{y}}_{nfh}-\boldsymbol{\mu}_f\right)\right] \\
\boldsymbol{\mu}_f^{*} &= \frac{\sum_{n=1}^{N}\left(\hat{z}_{nf}\boldsymbol{x}_n+\sum_{h\in\mathcal{H}}\hat{z}_{nh}\bar{\boldsymbol{y}}_{nfh}\right)}{\sum_{n=1}^{N}\left(\hat{z}_{nf}+\sum_{h\in\mathcal{H}}\hat{z}_{nh}\right)}
\end{aligned}
$$

B.2.2 Maximizing in Factor Covariances

$$
\begin{aligned}
\frac{\partial Q}{\partial \boldsymbol{\Sigma}_f^{-1}} &= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\Sigma}_f^{-1}}\sum_{n=1}^{N}\left[\hat{z}_{nf}\left\{\log\left|\boldsymbol{\Sigma}_f^{-1}\right| - \mathrm{Tr}\left[\boldsymbol{\Sigma}_f^{-1}\left(\boldsymbol{x}_n - \boldsymbol{\mu}_f\right)\left(\boldsymbol{x}_n - \boldsymbol{\mu}_f\right)^\top\right]\right\}\right. \\
&\quad \left. + \sum_{h\in\mathcal{H}}\hat{z}_{nh}\left(\log\left|\boldsymbol{\Sigma}_f^{-1}\right| - \mathrm{Tr}\left[\boldsymbol{\Sigma}_f^{-1}\,\mathrm{E}\left[\left(\boldsymbol{y}_{nf} - \boldsymbol{\mu}_f\right)\left(\boldsymbol{y}_{nf} - \boldsymbol{\mu}_f\right)^\top\right]\right]\right)\right] \\
&= \frac{1}{2}\sum_{n=1}^{N}\left[\hat{z}_{nf}\left\{\boldsymbol{\Sigma}_f - \left(\boldsymbol{x}_n - \boldsymbol{\mu}_f\right)\left(\boldsymbol{x}_n - \boldsymbol{\mu}_f\right)^\top\right\}\right. \\
&\quad \left. + \sum_{h\in\mathcal{H}}\hat{z}_{nh}\left(\boldsymbol{\Sigma}_f - \mathrm{E}\left[\left(\boldsymbol{y}_{nf} - \boldsymbol{\mu}_f\right)\left(\boldsymbol{y}_{nf} - \boldsymbol{\mu}_f\right)^\top\right]\right)\right] \\
&= \frac{1}{2}\sum_{n=1}^{N}\left[\hat{z}_{nf}\left\{\boldsymbol{\Sigma}_f - \left(\boldsymbol{x}_n - \boldsymbol{\mu}_f\right)\left(\boldsymbol{x}_n - \boldsymbol{\mu}_f\right)^\top\right\}\right. \\
&\quad \left. + \sum_{h\in\mathcal{H}}\hat{z}_{nh}\left(\boldsymbol{\Sigma}_f - \left[\mathbf{S}_{hff} + \left(\bar{\boldsymbol{y}}_{nfh} - \boldsymbol{\mu}_f\right)\left(\bar{\boldsymbol{y}}_{nfh} - \boldsymbol{\mu}_f\right)^\top\right]\right)\right] \\
\Rightarrow \mathbf{0} &= \sum_{n=1}^{N}\left[\hat{z}_{nf}\left\{\boldsymbol{\Sigma}_f - \left(\boldsymbol{x}_n - \boldsymbol{\mu}_f\right)\left(\boldsymbol{x}_n - \boldsymbol{\mu}_f\right)^\top\right\}\right. \\
&\quad \left. + \sum_{h\in\mathcal{H}}\hat{z}_{nh}\left(\boldsymbol{\Sigma}_f - \left[\mathbf{S}_{hff} + \left(\bar{\boldsymbol{y}}_{nfh} - \boldsymbol{\mu}_f\right)\left(\bar{\boldsymbol{y}}_{nfh} - \boldsymbol{\mu}_f\right)^\top\right]\right)\right] \\
\boldsymbol{\Sigma}_f^* &= \frac{\sum_{n=1}^{N}\left\{\hat{z}_{nf}\left(\boldsymbol{x}_n - \boldsymbol{\mu}_f\right)\left(\boldsymbol{x}_n - \boldsymbol{\mu}_f\right)^\top + \sum_{h\in\mathcal{H}}\hat{z}_{nh}\left[\mathbf{S}_{hff} + \left(\bar{\boldsymbol{y}}_{nfh} - \boldsymbol{\mu}_f\right)\left(\bar{\boldsymbol{y}}_{nfh} - \boldsymbol{\mu}_f\right)^\top\right]\right\}}{\sum_{n=1}^{N}\left(\hat{z}_{nf} + \sum_{h\in\mathcal{H}}\hat{z}_{nh}\right)}
\end{aligned}
$$

B.2.3 Maximizing in Hybrid Error

Diagonal, Varying Error Distributions

In the most general case where $\boldsymbol{\Psi}_h$, we obtain the following maximization for a specific $h \in \mathcal{H}$:

$$
\begin{aligned}
\frac{\partial Q}{\partial \boldsymbol{\Psi}_h^{-1}} &= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\Psi}_h^{-1}}\sum_{n=1}^{N}\hat{z}_{nh}\left\{\log\left|\boldsymbol{\Psi}_h^{-1}\right| - \mathrm{Tr}\left[\boldsymbol{\Psi}_h^{-1}\,\mathrm{E}\left[\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)^\top\right]\right]\right\} \\
&= \frac{1}{2}\sum_{n=1}^{N}\hat{z}_{nh}\left\{\boldsymbol{\Psi}_h - \mathrm{E}\left[\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)^\top\right]\right\} \\
&= \frac{1}{2}\sum_{n=1}^{N}\hat{z}_{nh}\left\{\boldsymbol{\Psi}_h - \left[\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\bar{\boldsymbol{y}}_{nfh}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\bar{\boldsymbol{y}}_{nfh}\right)^\top + \sum_{f\in\mathcal{F}}\sum_{q\in\mathcal{F}}\alpha_{hf}\alpha_{hq}\mathbf{S}_{hfq}\right]\right\} \\
\Rightarrow \mathbf{0} &= \sum_{n=1}^{N}\hat{z}_{nh}\left\{\boldsymbol{\Psi}_h - \left[\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\bar{\boldsymbol{y}}_{nfh}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\bar{\boldsymbol{y}}_{nfh}\right)^\top + \sum_{f\in\mathcal{F}}\sum_{q\in\mathcal{F}}\alpha_{hf}\alpha_{hq}\mathbf{S}_{hfq}\right]\right\} \\
\boldsymbol{\Psi}_h^* &= \mathrm{diag}\,\frac{\sum_{n=1}^{N}\hat{z}_{nh}\left[\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\bar{\boldsymbol{y}}_{nfh}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\bar{\boldsymbol{y}}_{nfh}\right)^\top + \sum_{f\in\mathcal{F}}\sum_{q\in\mathcal{F}}\alpha_{hf}\alpha_{hq}\mathbf{S}_{hfq}\right]}{\sum_{n=1}^{N}\hat{z}_{nh}}
\end{aligned}
$$

Spherical, Varying Error Distributions

In the case where $\boldsymbol{\Psi}_h = \psi_h\mathbf{I}_d$ we obtain the following maximization in ψ_h for a specific $h \in \mathcal{H}$:

$$
\begin{aligned}
Q =& \sum_{n=1}^{N} \hat{z}_{nh} \left\{ \log \left| \psi_h^{-1} \mathbf{I}_d \right| - \operatorname{Tr} \left[\psi_h^{-1} \mathbf{I}_d \operatorname{E} \left[\left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \boldsymbol{y}_{nf} \right) \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \boldsymbol{y}_{nf} \right)^\top \right] \right] \right\} \\
=& \sum_{n=1}^{N} \hat{z}_{nh} \left\{ d \log \psi_h^{-1} - \psi_h^{-1} \operatorname{Tr} \left[\left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\boldsymbol{y}}_{nfh} \right) \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\boldsymbol{y}}_{nfh} \right)^\top \right. \right. \\
& \left. \left. + \sum_{f \in \mathcal{F}} \sum_{q \in \mathcal{F}} \alpha_{hf} \alpha_{hq} \mathbf{S}_{hfq} \right] \right\} \\
=& d \log \psi_h^{-1} \left(\sum_{n=1}^{N} \hat{z}_{nh} \right) - \psi_h^{-1} \sum_{n=1}^{N} \hat{z}_{nh} \left\{ \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\boldsymbol{y}}_{nfh} \right)^\top \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\boldsymbol{y}}_{nfh} \right) \right. \\
& \left. + \sum_{f \in \mathcal{F}} \sum_{q \in \mathcal{F}} \alpha_{hf} \alpha_{hq} \mathbf{S}_{hfq} \right\}
\end{aligned}
$$

Differentiating in ψ_h^{-1}:

$$
\begin{aligned}
\frac{\partial Q}{\partial \psi_h^{-1}} =& \psi_h d \left(\sum_{n=1}^{N} \hat{z}_{nh} \right) \\
& - \sum_{n=1}^{N} \hat{z}_{nh} \left\{ \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\boldsymbol{y}}_{nfh} \right)^\top \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\boldsymbol{y}}_{nfh} \right) + \sum_{f \in \mathcal{F}} \sum_{q \in \mathcal{F}} \alpha_{hf} \alpha_{hq} \mathbf{S}_{hfq} \right\}
\end{aligned}
$$

Hence, the maximizer is:

$$
\psi_h^* = \frac{\sum_{n=1}^{N} \hat{z}_{nh} \left\{ \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\boldsymbol{y}}_{nfh} \right)^\top \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\boldsymbol{y}}_{nfh} \right) + \sum_{f \in \mathcal{F}} \sum_{q \in \mathcal{F}} \alpha_{hf} \alpha_{hq} \mathbf{S}_{hfq} \right\}}{d \left(\sum_{n=1}^{N} \hat{z}_{nh} \right)}
$$

Diagonal, Equal Error Distributions

In the case where $\boldsymbol{\Psi}_h = \boldsymbol{\Psi}$ we obtain the following maximization in $\boldsymbol{\Psi}$:

$$
\begin{aligned}
Q &= \sum_{n=1}^{N}\sum_{h\in\mathcal{H}} \hat{z}_{nh}\left\{\log\left|\boldsymbol{\Psi}^{-1}\right| - \operatorname{Tr}\left[\boldsymbol{\Psi}^{-1}\,\mathrm{E}\left[\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)^{\top}\right]\right]\right\} \\
&= \log\left|\boldsymbol{\Psi}^{-1}\right|\left(\sum_{n=1}^{N}\sum_{h\in\mathcal{H}}\hat{z}_{nh}\right) - \sum_{n=1}^{N}\sum_{h\in\mathcal{H}}\hat{z}_{nh}\operatorname{Tr}\left[\boldsymbol{\Psi}^{-1}\,\mathrm{E}\left[\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)^{\top}\right]\right]
\end{aligned}
$$

Taking derivative in $\boldsymbol{\Psi}^{-1}$:

$$
\frac{\partial Q}{\partial \boldsymbol{\Psi}^{-1}} = \boldsymbol{\Psi}\left(\sum_{n=1}^{N}\sum_{h\in\mathcal{H}}\hat{z}_{nh}\right) - \sum_{n=1}^{N}\sum_{h\in\mathcal{H}}\hat{z}_{nh}\,\mathrm{E}\left[\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)^{\top}\right]
$$

Hence the maximizer is:

$$
\boldsymbol{\Psi}^{*} = \frac{\sum_{n=1}^{N}\sum_{h\in\mathcal{H}}\hat{z}_{nh}\left[\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\bar{\boldsymbol{y}}_{nfh}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\bar{\boldsymbol{y}}_{nfh}\right)^{\top} + \sum_{f\in\mathcal{F}}\sum_{q\in\mathcal{F}}\alpha_{hf}\alpha_{hq}\mathbf{S}_{hfq}\right]}{\sum_{n=1}^{N}\sum_{h\in\mathcal{H}}\hat{z}_{nh}}
$$

Spherical, Equal Error Distributions

In the case where $\boldsymbol{\Psi}_h = \psi\mathbf{I}_d$ we obtain the following maximization in ψ:

$$
\begin{aligned}
Q &= \sum_{n=1}^{N}\sum_{h\in\mathcal{H}} \hat{z}_{nh}\left\{\log\left|\psi^{-1}\mathbf{I}_d\right| - \operatorname{Tr}\left[\psi^{-1}\mathbf{I}_d\,\mathrm{E}\left[\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)^{\top}\right]\right]\right\} \\
&= \log\left|\psi^{-1}\mathbf{I}_d\right|\left(\sum_{n=1}^{N}\sum_{h\in\mathcal{H}}\hat{z}_{nh}\right) \\
&\quad -\psi^{-1}\sum_{n=1}^{N}\sum_{h\in\mathcal{H}}\hat{z}_{nh}\operatorname{Tr}\left[\mathrm{E}\left[\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)\left(\boldsymbol{x}_n - \sum_{f\in\mathcal{F}}\alpha_{hf}\boldsymbol{y}_{nf}\right)^{\top}\right]\right]
\end{aligned}
$$

Taking derivative in ψ^{-1}:

$$\begin{aligned}\frac{\partial Q}{\partial \psi^{-1}} =& \psi d \left(\sum_{n=1}^{N} \sum_{h \in \mathcal{H}} \hat{z}_{nh} \right) - \sum_{n=1}^{N} \sum_{h \in \mathcal{H}} \hat{z}_{nh} \left[\left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\boldsymbol{y}}_{nfh} \right)^\top \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\boldsymbol{y}}_{nfh} \right) \right. \\ & \left. + \sum_{f \in \mathcal{F}} \sum_{q \in \mathcal{F}} \alpha_{hf} \alpha_{hq} \mathbf{S}_{hfq} \right]\end{aligned}$$

Hence the maximizer is:

$$\psi^* = \frac{\sum_{n=1}^{N} \sum_{h \in \mathcal{H}} \hat{z}_{nh} \left[\left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\boldsymbol{y}}_{nfh} \right)^\top \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\boldsymbol{y}}_{nfh} \right) + \sum_{f \in \mathcal{F}} \sum_{q \in \mathcal{F}} \alpha_{hf} \alpha_{hq} \mathbf{S}_{hfq} \right]}{d \sum_{n=1}^{N} \sum_{h \in \mathcal{H}} \hat{z}_{nh}}$$

B.2.4 Quadratic Programming Derivation

We must optimize Q with respect to $\boldsymbol{\alpha}_h$ for $h \in \mathcal{H}$ and subject to the constraints that $\boldsymbol{\alpha}_h \geq 0$ and $\sum_{f \in \mathcal{F}} \alpha_{hf} = \|\boldsymbol{\alpha}_h\|_1 = \mathbf{1}^\top \boldsymbol{\alpha}_h = 1$. We first show that Q is in fact quadratic in $\boldsymbol{\alpha}_h$; consider that, with respect to $\boldsymbol{\alpha}_h$ for a specific $h \in \mathcal{H}$ and holding all others constant, we have the following expression:

$$\begin{aligned} Q =& \text{constant} - \frac{1}{2} \sum_{n=1}^{N} \hat{z}_{nh} \operatorname{Tr} \left[\boldsymbol{\Psi}_h^{-1} \operatorname{E} \left[\left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \boldsymbol{y}_{nf} \right) \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \boldsymbol{y}_{nf} \right)^\top \right] \right] \\ =& \text{constant} - \frac{1}{2} \sum_{n=1}^{N} \hat{z}_{nh} \operatorname{Tr} \left[\boldsymbol{\Psi}_h^{-1} \left[\left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\boldsymbol{y}}_{nfh} \right) \left(\boldsymbol{x}_n - \sum_{f \in \mathcal{F}} \alpha_{hf} \bar{\boldsymbol{y}}_{nfh} \right)^\top \right. \right. \\ & \left. \left. + \sum_{f \in \mathcal{F}} \sum_{q \in \mathcal{F}} \alpha_{hf} \alpha_{hq} \mathbf{S}_{hfq} \right] \right] \end{aligned}$$

In order to recast these summations over $\mathcal{F}$ into an expression more suitable for matrix-oriented algebra, we define the following helper matrices:

$$
\begin{aligned}
\mathbf{A}_{nh} &= \begin{bmatrix} \bar{\boldsymbol{y}}_{n1h} & \cdots & \bar{\boldsymbol{y}}_{nFh} \end{bmatrix} \\
\mathbf{B}_h &= \begin{bmatrix} \operatorname{Tr}\left[\boldsymbol{\Psi}_h^{-1}\mathbf{S}_{h11}\right] & & \operatorname{Tr}\left[\boldsymbol{\Psi}_h^{-1}\mathbf{S}_{h11}\right] \\ & & \\ \operatorname{Tr}\left[\boldsymbol{\Psi}_h^{-1}\mathbf{S}_{h11}\right] & & \operatorname{Tr}\left[\boldsymbol{\Psi}_h^{-1}\mathbf{S}_{h11}\right] \end{bmatrix}
\end{aligned}
$$

We make the appropriate substitutions and re-arrange into a suitable quadratic form.

$$
\begin{aligned}
Q = \text{const} - \frac{1}{2}\sum_{n=1}^{N}\hat{z}_{nh}\operatorname{Tr}\Big[\boldsymbol{\Psi}_h^{-1}\Big[\boldsymbol{x}_n\boldsymbol{x}_n^\top - \boldsymbol{x}_n\left(\mathbf{A}_{nh}\boldsymbol{\alpha}_h\right)^\top - \left(\mathbf{A}_{nh}\boldsymbol{\alpha}_h\right)\boldsymbol{x}_n^\top + \mathbf{A}_{nh}\boldsymbol{\alpha}_h\boldsymbol{\alpha}_h^\top\mathbf{A}_{nh}^\top \\
+ \sum_{f\in\mathcal{F}}\sum_{q\in\mathcal{F}}\alpha_{hf}\alpha_{hq}\mathbf{S}_{hfq}\Big]\Big]
\end{aligned}
$$

We now re-arrange into a suitable quadratic form:

$$
\begin{aligned}
Q &= \text{const} - \frac{1}{2}\Bigg\{\sum_{n=1}^{N}\hat{z}_{nh}\left(\operatorname{Tr}\left[\boldsymbol{\Psi}_h^{-1}\mathbf{A}_{nh}\boldsymbol{\alpha}_h\boldsymbol{\alpha}_h^\top\mathbf{A}_{nh}^\top\right] + \sum_{f\in\mathcal{F}}\sum_{q\in\mathcal{F}}\alpha_{hf}\alpha_{hq}\operatorname{Tr}\left[\boldsymbol{\Psi}_h^{-1}\mathbf{S}_{hfq}\right]\right) \\
&\quad - 2\sum_{n=1}^{N}\hat{z}_{nh}\operatorname{Tr}\left[\boldsymbol{\Psi}_h^{-1}\mathbf{A}_{nh}\boldsymbol{\alpha}_h\boldsymbol{x}_n^\top\right] + \sum_{n=1}^{N}\hat{z}_{nh}\operatorname{Tr}\left[\boldsymbol{\Psi}_h^{-1}\boldsymbol{x}_n\boldsymbol{x}_n^\top\right]\Bigg\} \\
Q &= \text{const} - \frac{1}{2}\Bigg\{\sum_{n=1}^{N}\hat{z}_{nh}\left(\operatorname{Tr}\left[\boldsymbol{\alpha}_h^\top\mathbf{A}_{nh}^\top\boldsymbol{\Psi}_h^{-1}\mathbf{A}_{nh}\boldsymbol{\alpha}_h\right] + \boldsymbol{\alpha}_h^\top\mathbf{B}_h\boldsymbol{\alpha}_h\right) \\
&\quad - 2\sum_{n=1}^{N}\hat{z}_{nh}\mathbf{A}_{nh}^\top\boldsymbol{\Psi}_h^{-1}\boldsymbol{x}_n\boldsymbol{\alpha}_h + \text{const}\Bigg\}
\end{aligned}
$$

Dropping constants:

$$
Q = -\frac{1}{2}\left\{\boldsymbol{\alpha}_h^\top\left(\sum_{n=1}^{N}\hat{z}_{nh}\mathbf{A}_{nh}^\top\boldsymbol{\Psi}_h^{-1}\mathbf{A}_{nh} + \mathbf{B}_h\sum_{n=1}^{N}\hat{z}_{nh}\right)\boldsymbol{\alpha}_h + \left(-2\sum_{n=1}^{N}\hat{z}_{nh}\mathbf{A}_{nh}^\top\boldsymbol{\Psi}_h^{-1}\boldsymbol{x}_n\right)\boldsymbol{\alpha}_h\right\}
$$

Multiply through by -1 to convert the maximization problem into a minimization problem:

$$Q = \frac{1}{2}\boldsymbol{\alpha}_h^\top \underbrace{\left(\sum_{n=1}^{N} \hat{z}_{nh}\mathbf{A}_{nh}^\top \boldsymbol{\Psi}_h^{-1}\mathbf{A}_{nh} + \mathbf{B}_h \sum_{n=1}^{N} \hat{z}_{nh}\right)}_{\mathbf{F}_h}\boldsymbol{\alpha}_h + \underbrace{\left(\sum_{n=1}^{N} \hat{z}_{nh}\mathbf{A}_{nh}^\top \boldsymbol{\Psi}_h^{-1}\boldsymbol{x}_n\right)}_{\boldsymbol{q}_h^\top}\boldsymbol{\alpha}_h$$

Here, we see that Q was indeed a quadratic form, so we postulate the following quadratic programming problem:

$$\min_{\boldsymbol{\alpha}_h} \frac{1}{2}\boldsymbol{\alpha}_h^\top \mathbf{F}_h \boldsymbol{\alpha}_h + \boldsymbol{q}_h^\top \boldsymbol{\alpha}_h \text{ subject to } \boldsymbol{\alpha}_h \geq 0 \text{ and } \mathbf{1}^\top \boldsymbol{\alpha}_h = 1$$

Appendix C

Model-Based Clustering with Nested Gaussian Clusters

C.1 Majorization-Minimization update for rotation $\boldsymbol{\Gamma}$

The expected complete-data log-likelihood for the data under the proposed model in the conditional expression is given by

$$Q(\mathbf{V};\boldsymbol{\theta}) = \sum_{n=1}^{N}\sum_{g=1}^{G_\cdot}\sum_{h=1}^{H_g}\hat{z}_{n,g}\hat{w}_{n,g:h}\Big[\log\pi_g + \log\tau_{g:h} + \log\phi_{p_x}\left(\boldsymbol{x}_n;\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g\right) + \log\phi_{p_y}\left(\boldsymbol{y}_n;\boldsymbol{\eta}_{g:h}+\mathbf{B}_{g:h}\boldsymbol{x}_n,\boldsymbol{\Lambda}_{g:h}\right) + \log\phi_{p_u}\left(\boldsymbol{u}_n;\boldsymbol{\xi},\boldsymbol{\Psi}\right)\Big].$$

We can re-write the product of marginal and conditional densities as the joint density with structured parameters of the form

$$\begin{aligned}
&\phi_{p_x}\left(\boldsymbol{x}_n;\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g\right)\phi_{p_y}\left(\boldsymbol{y}_n;\boldsymbol{\eta}_{g:h}+\mathbf{B}_{g:h}\boldsymbol{x}_n,\boldsymbol{\Lambda}_{g:h}\right)\phi_{p_u}\left(\boldsymbol{u}_n;\boldsymbol{\xi},\boldsymbol{\Psi}\right)\\
&= \phi_p\left(\boldsymbol{v}_n;\boldsymbol{\Gamma}^\top\begin{bmatrix}\boldsymbol{\mu}_g\\ \boldsymbol{\eta}_{g:h}+\mathbf{B}\boldsymbol{\mu}_g\\ \boldsymbol{\xi}\end{bmatrix},\boldsymbol{\Gamma}^\top\begin{bmatrix}\boldsymbol{\Sigma}_g & \boldsymbol{\Sigma}_g\mathbf{B}_{g:h}^\top & \mathbf{0}\\ \mathbf{B}_{g:h}\boldsymbol{\Sigma}_g & \mathbf{B}_{g:h}\boldsymbol{\Sigma}_g\mathbf{B}_{g:h}^\top+\boldsymbol{\Lambda}_{g:h} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Psi}\end{bmatrix}\boldsymbol{\Gamma}\right)\\
&= \phi_p\left(\boldsymbol{v}_n;\boldsymbol{\Gamma}^\top\bar{\boldsymbol{\mu}}_{g:h},\boldsymbol{\Gamma}^\top\bar{\boldsymbol{\Sigma}}_{g:h}\boldsymbol{\Gamma}\right).
\end{aligned}$$

Thus, we can re-write the expected complete-data log-likelihood as

$$Q(\mathbf{V};\boldsymbol{\theta}) = \sum_{n=1}^{N}\sum_{g=1}^{G_\cdot}\sum_{h=1}^{H_g} \hat{z}_{n,g}\hat{w}_{n,g:h}\left[\log(\pi_g\tau_{g:h}) + \log\phi_p\left(\boldsymbol{v}_n; \boldsymbol{\Gamma}^\top\bar{\boldsymbol{\mu}}_{g:h}, \boldsymbol{\Gamma}^\top\bar{\boldsymbol{\Sigma}}_{g:h}\boldsymbol{\Gamma}\right)\right].$$

To update $\boldsymbol{\Gamma}$, we treat the other parameters as fixed. We can expand the log-density as

$$\begin{aligned}
&\log\phi_p\left(\boldsymbol{v}_n; \boldsymbol{\Gamma}^\top\bar{\boldsymbol{\mu}}_{g:h}, \boldsymbol{\Gamma}^\top\bar{\boldsymbol{\Sigma}}_{g:h}\boldsymbol{\Gamma}\right)\\
&= -\frac{p}{2}\log 2\pi - \frac{1}{2}\log\det\boldsymbol{\Gamma}^\top\bar{\boldsymbol{\Sigma}}_{g:h}\boldsymbol{\Gamma} - \frac{1}{2}\operatorname{Tr}\left[(\boldsymbol{\Gamma}^\top\bar{\boldsymbol{\Sigma}}_{g:h}\boldsymbol{\Gamma})^{-1}(\boldsymbol{v}_n - \boldsymbol{\Gamma}^\top\bar{\boldsymbol{\mu}}_{g:h})(\boldsymbol{v}_n - \boldsymbol{\Gamma}^\top\bar{\boldsymbol{\mu}}_{g:h})^\top\right]\\
&= -\frac{p}{2}\log 2\pi - \frac{1}{2}\log\det\bar{\boldsymbol{\Sigma}}_{g:h} - \frac{1}{2}\operatorname{Tr}\left[\boldsymbol{\Gamma}^{-1}\bar{\boldsymbol{\Sigma}}_{g:h}^{-1}\boldsymbol{\Gamma}^{\top-1}(\boldsymbol{v}_n\boldsymbol{v}_n^\top - \boldsymbol{\Gamma}^\top\bar{\boldsymbol{\mu}}_{g:h}\boldsymbol{v}_n^\top - \boldsymbol{v}_n\bar{\boldsymbol{\mu}}_{g:h}^\top\boldsymbol{\Gamma}\right.\\
&\quad \left. + \boldsymbol{\Gamma}^\top\bar{\boldsymbol{\mu}}_{g:h}\bar{\boldsymbol{\mu}}_{g:h}^\top\boldsymbol{\Gamma})\right]\\
&\propto \text{constant} - \operatorname{Tr}\left[\boldsymbol{\Gamma}^{-1}\bar{\boldsymbol{\Sigma}}_{g:h}^{-1}\boldsymbol{\Gamma}^{\top-1}\boldsymbol{v}_n\boldsymbol{v}_n^\top\right] + 2\operatorname{Tr}\left[\boldsymbol{\Gamma}^{-1}\bar{\boldsymbol{\Sigma}}_{g:h}^{-1}\boldsymbol{\Gamma}^{\top-1}\boldsymbol{\Gamma}^\top\bar{\boldsymbol{\mu}}_{g:h}\boldsymbol{v}_n^\top\right]\\
&\quad - \operatorname{Tr}\left[\boldsymbol{\Gamma}^{-1}\bar{\boldsymbol{\Sigma}}_{g:h}^{-1}\boldsymbol{\Gamma}^{\top-1}\boldsymbol{\Gamma}^\top\bar{\boldsymbol{\mu}}_{g:h}\bar{\boldsymbol{\mu}}_{g:h}^\top\boldsymbol{\Gamma}\right]\\
&= \text{constant} - \operatorname{Tr}\left[\boldsymbol{\Gamma}^\top\bar{\boldsymbol{\Sigma}}_{g:h}^{-1}\boldsymbol{\Gamma}\boldsymbol{v}_n\boldsymbol{v}_n^\top\right] + 2\operatorname{Tr}\left[\boldsymbol{\Gamma}^\top\bar{\boldsymbol{\Sigma}}_{g:h}^{-1}\bar{\boldsymbol{\mu}}_{g:h}\boldsymbol{v}_n^\top\right]\\
&= \text{constant} + 2\operatorname{Tr}\left[\boldsymbol{v}_n\bar{\boldsymbol{\mu}}_{g:h}^\top\bar{\boldsymbol{\Sigma}}_{g:h}^{-1}\boldsymbol{\Gamma}\right] - \operatorname{Tr}\left[\bar{\boldsymbol{\Sigma}}_{g:h}^{-1}\boldsymbol{\Gamma}\boldsymbol{v}_n\boldsymbol{v}_n^\top\boldsymbol{\Gamma}^\top\right].
\end{aligned}$$

Substituting this into the surrogate function and interchanging linear operators with finite summations, we obtain

$$\begin{aligned}
Q(\mathbf{V};\boldsymbol{\theta}) \propto\ &\text{constant}\\
&+ 2\operatorname{Tr}\left[\left(\sum_{g=1}^{G}\sum_{h=1}^{H_g}\left(\sum_{n=1}^{N}\hat{z}_{n,g}\hat{w}_{n,g:h}\boldsymbol{v}_n\right)\bar{\boldsymbol{\mu}}_{g:h}^\top\bar{\boldsymbol{\Sigma}}_{g:h}^{-1}\right)\boldsymbol{\Gamma}\right]\\
&- \sum_{g=1}^{G}\sum_{h=1}^{H_g}\operatorname{Tr}\left[\boldsymbol{\Gamma}^\top\bar{\boldsymbol{\Sigma}}_{g:h}^{-1}\boldsymbol{\Gamma}\left(\sum_{n=1}^{N}\boldsymbol{v}_n\boldsymbol{v}_n^\top\right)\right].
\end{aligned}$$

In the form of (1) in (Kiers, 2002), we re-write this as the matrix minimization problem

$$
\begin{aligned}
g(\boldsymbol{\Gamma}) &= \operatorname{Tr} \mathbf{A}\boldsymbol{\Gamma} + \sum_{g:h} \operatorname{Tr} \mathbf{B}_{g:h}\boldsymbol{\Gamma}\mathbf{C}_{g:h}\boldsymbol{\Gamma}^\top, \text{ where} \\
\mathbf{A} &= 2\sum_{g=1}^{G}\sum_{h=1}^{H_g}\left(\sum_{n=1}^{N}\hat{z}_{n,g}\hat{w}_{n,g:h}\boldsymbol{v}_n\right)\bar{\boldsymbol{\mu}}_{g:h}^\top\bar{\boldsymbol{\Sigma}}_{g:h}^{-1}, \\
\mathbf{B}_{g:h} &= -\bar{\boldsymbol{\Sigma}}_{g:h}^{-1}, \text{ and} \\
\mathbf{C}_{g:h} &= \sum_{n=1}^{N}\boldsymbol{v}_n\boldsymbol{v}_n^\top.
\end{aligned}
$$

By making $\mathbf{C}_{g:h}$ positive definite, we can make use of the better majorizing function (Kiers and ten Berge, 1992) given by

$$
m_{\mathrm{c}}(\boldsymbol{\Gamma}) = \text{constant} + \operatorname{Tr}\mathbf{F}\boldsymbol{\Gamma} + \sum_{g=1}^{G}\sum_{h=1}^{H_g}\lambda_{g:h}\operatorname{Tr}\mathbf{C}_{g:h}\boldsymbol{\Gamma}^\top\boldsymbol{\Gamma}
$$

where $\boldsymbol{\Gamma}^\top\boldsymbol{\Gamma} = \mathbf{I}$ by definition of $\boldsymbol{\Gamma}$, and the matrix

$$
\begin{aligned}
\mathbf{F} &= \mathbf{A} + \sum_{g=1}^{G}\sum_{h=1}^{H_g}(\mathbf{C}_{g:h}\boldsymbol{\Gamma}_{\text{old}}^\top(\mathbf{B}_{g:h} + \mathbf{B}_{g:h}^\top) - 2\lambda_{g:h}\mathbf{C}_{g:h}\boldsymbol{\Gamma}_{\text{old}}^\top) \\
&= 2\sum_{g=1}^{G}\sum_{h=1}^{H_g}\bar{\boldsymbol{\Sigma}}_{g:h}^{-1}\bar{\boldsymbol{\mu}}_{g:h} - 2\sum_{g=1}^{G}\sum_{h=1}^{H_g}\mathbf{C}_{g:h}\boldsymbol{\Gamma}_{\text{old}}^\top(\bar{\boldsymbol{\Sigma}}_{g:h}^{-1} + \lambda_{g:h}\mathbf{I}).
\end{aligned}
$$

From this, we perform the indicated update in Kiers (2002) by taking the singular value decomposition of $\mathbf{F} = \mathbf{P}\mathbf{D}\mathbf{Q}^\top$ and updating $\boldsymbol{\Gamma}_{\text{new}} = \mathbf{Q}\mathbf{P}^\top$. Since the factor of two in $\mathbf{F}$ does not change the matrices $\mathbf{P}$ and $\mathbf{Q}$, it can be dropped without changing the solution for $\boldsymbol{\Gamma}_{\text{new}}$.

As well, by the invariance of the trace under cyclic permutation of the matrix product and invariance under transposition, we can define $\boldsymbol{\Omega} = \boldsymbol{\Gamma}^\top$ with the orthogonality constraint

$\boldsymbol{\Omega}^\top \boldsymbol{\Omega} = \mathbf{I}$ and re-write the minimization objective g as

$$\begin{aligned} g'(\boldsymbol{\Omega}) &= \operatorname{Tr} \mathbf{A}^\top \boldsymbol{\Omega} + \sum_{g:h} \operatorname{Tr} \mathbf{C}_{g:h} \boldsymbol{\Omega} \mathbf{B}_{g:h} \boldsymbol{\Omega}^\top, \text{ where} \\ &= \operatorname{Tr} \mathbf{A}' \boldsymbol{\Gamma} + \sum_{g:h} \operatorname{Tr} \mathbf{B}'_{g:h} \boldsymbol{\Omega} \mathbf{C}'_{g:h} \boldsymbol{\Omega}^\top, \text{ where} \\ \mathbf{A}' &= \mathbf{A}^\top, \\ \mathbf{B}'_{g:h} &= -\mathbf{C}_{g:h}, \text{ and} \\ \mathbf{C}'_{g:h} &= -\mathbf{B}_{g:h}. \end{aligned}$$

so that $\mathbf{C}'_{g:h}$ is positive definite again, allowing the better majorizing function to be used. In this case,

$$\begin{aligned} \mathbf{F}' &= \mathbf{A} + \sum_{g=1}^{G} \sum_{h=1}^{H_g} (\mathbf{C}_{g:h} \boldsymbol{\Gamma}_{\text{old}}^\top (\mathbf{B}_{g:h} + \mathbf{B}_{g:h}^\top) - 2\lambda_{g:h} \mathbf{C}_{g:h} \boldsymbol{\Gamma}_{\text{old}}^\top) \\ &= 2 \sum_{g=1}^{G} \sum_{h=1}^{H_g} \bar{\boldsymbol{\Sigma}}_{g:h}^{-1} \bar{\boldsymbol{\mu}}_{g:h} - 2 \sum_{g=1}^{G} \sum_{h=1}^{H_g} \mathbf{C}_{g:h} \boldsymbol{\Gamma}_{\text{old}}^\top (\bar{\boldsymbol{\Sigma}}_{g:h}^{-1} + \lambda_{g:h} \mathbf{I}]). \end{aligned}$$

C.2 Real-World Dataset Estimated Parameters

In this section, we provide the parameters for all of the fitted models selected by BIC as shown in the main manuscript.

C.2.1 Crabs dataset

Unsupervised

Recalling that BIC selected a model with conditional independence but not intrinsic independence with $G = 2$ and $H = 2$, we have that

$$\boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix} = \begin{bmatrix} 0.4997 \\ 0.5003 \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} 0.5600 \\ 0.4400 \end{bmatrix}.$$

The primary and secondary clustering mean parameters are given by

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0.3297 \\ 1.1374 \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} -0.4196 \\ -1.4476 \end{bmatrix}, \quad \boldsymbol{\eta}_1 = \begin{bmatrix} 2.2603 \\ 0.8762 \end{bmatrix}, \quad \boldsymbol{\eta}_2 = \begin{bmatrix} -2.2007 \\ -0.8044 \end{bmatrix}.$$

The regression matrices are given by

$$\mathbf{B}_1 = \begin{bmatrix} -0.8599 & -0.2927 \\ 1.7079 & 0.9070 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} -0.7744 & -0.5208 \\ 1.9450 & 0.8867 \end{bmatrix}.$$

The covariance matrices are given by

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 6.3006 & -8.9754 \\ -8.9754 & 13.4708 \end{bmatrix}, \qquad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 3.5831 & -8.2122 \\ -8.2122 & 19.6559 \end{bmatrix},$$
$$\boldsymbol{\Lambda}_1 = \begin{bmatrix} 0.3962 & 0.1130 \\ 0.1130 & 0.1450 \end{bmatrix}, \qquad \boldsymbol{\Lambda}_2 = \begin{bmatrix} 0.7289 & 0.3304 \\ 0.3304 & 0.3046 \end{bmatrix}.$$

The rotation matrix is given by

$$\boldsymbol{\Gamma} = \begin{bmatrix} -0.2228 & 0.5468 & -0.2524 & -0.7624 & 0.0795 \\ 0.9044 & 0.3427 & -0.2109 & 0.0369 & -0.1373 \\ -0.1340 & -0.3007 & -0.5990 & -0.0541 & -0.7279 \\ 0.2870 & -0.6752 & -0.2414 & -0.4406 & 0.4575 \\ -0.1792 & 0.1929 & -0.6890 & 0.4694 & 0.4854 \end{bmatrix}.$$

Semi-Supervised

Recalling that BIC selected a model with conditional independence but not intrinsic independence with $G = 2$ and $H = 2$, we have that

$$\boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix} = \begin{bmatrix} 0.4999 \\ 0.5001 \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} 0.5370 \\ 0.4630 \end{bmatrix}.$$

The primary and secondary clustering mean parameters are given by

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0.4588 \\ -0.9254 \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} -0.5322 \\ 1.0734 \end{bmatrix}, \quad \boldsymbol{\eta}_1 = \begin{bmatrix} 2.3753 \\ -0.3878 \end{bmatrix}, \quad \boldsymbol{\eta}_2 = \begin{bmatrix} -2.4241 \\ 0.4469 \end{bmatrix}.$$

The regression matrices are given by

$$\mathbf{B}_1 = \begin{bmatrix} 0.8978 & -0.1965 \\ 2.2005 & -0.2458 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 1.0516 & 0.0212 \\ 1.8723 & -0.2951 \end{bmatrix}.$$

The covariance matrices are given by

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 6.3808 & 8.4419 \\ 8.4419 & 11.7865 \end{bmatrix}, \qquad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 4.2960 & 9.1759 \\ 9.1759 & 20.3848 \end{bmatrix},$$

$$\boldsymbol{\Lambda}_1 = \begin{bmatrix} 0.9039 & -0.1401 \\ -0.1401 & 0.1566 \end{bmatrix}, \qquad \boldsymbol{\Lambda}_2 = \begin{bmatrix} 0.6269 & -0.0264 \\ -0.0264 & 0.0916 \end{bmatrix}.$$

The rotation matrix is given by

$$\boldsymbol{\Gamma} = \begin{bmatrix} -0.1107 & -0.3514 & 0.3553 & -0.8555 & 0.0786 \\ 0.8699 & -0.4060 & 0.2081 & 0.1281 & -0.1369 \\ -0.1624 & 0.2441 & 0.6103 & 0.1073 & -0.7281 \\ 0.3385 & 0.7180 & 0.3742 & -0.1413 & 0.4581 \\ -0.3001 & -0.3694 & 0.5639 & 0.4693 & 0.4849 \end{bmatrix}.$$

C.2.2 Olive Oil Dataset

Unsupervised

Recalling that BIC selected a model with conditional independence but not intrinsic independence with $G = 4$ and $H = 3$, we have that

$$\boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{bmatrix} = \begin{bmatrix} 0.1543 \\ 0.5647 \\ 0.1731 \\ 0.1080 \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \begin{bmatrix} 0.3924 \\ 0.0726 \\ 0.5350 \end{bmatrix}.$$

The primary and secondary clustering mean parameters are given by

$$\boldsymbol{\mu}_1 = \begin{bmatrix} -28.9153 \\ 227.5089 \\ 221.6169 \\ -114.5314 \\ -208.0146 \\ -34.8238 \end{bmatrix}, \qquad \boldsymbol{\mu}_2 = \begin{bmatrix} 3.7179 \\ -86.1419 \\ -39.6550 \\ -4.4542 \\ 23.6603 \\ 14.6131 \end{bmatrix},$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 52.1938 \\ -10.1990 \\ -116.6432 \\ 121.7839 \\ 156.4276 \\ 5.3307 \end{bmatrix}, \qquad \boldsymbol{\mu}_4 = \begin{bmatrix} -61.7838 \\ 141.7932 \\ 77.7046 \\ -8.2628 \\ -77.2556 \\ -35.2118 \end{bmatrix},$$

$$\boldsymbol{\eta}_1 = \begin{bmatrix} 6.2399 \\ 7.4205 \end{bmatrix}, \qquad \boldsymbol{\eta}_2 = \begin{bmatrix} 44.4287 \\ -4.6681 \end{bmatrix},$$

$$\boldsymbol{\eta}_3 = \begin{bmatrix} -3.8371 \\ -2.5258 \end{bmatrix}.$$

The regression matrices are given by

$$\mathbf{B}_1 = \begin{bmatrix} -0.4539 & -0.3894 \\ 1.7909 & 0.3050 \\ 0.7359 & 0.5112 \\ 0.5724 & 0.5338 \\ 0.4043 & 0.5731 \\ -2.7105 & 0.4559 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} -1.0674 & 0.1422 \\ 2.0961 & -0.0090 \\ 0.7799 & 1.1501 \\ 0.4053 & 1.4176 \\ 0.3430 & 0.3098 \\ -0.2478 & 0.0990 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} -0.0631 & -0.5023 \\ 1.6452 & 0.2872 \\ 1.1412 & 0.5952 \\ 1.7234 & 0.7403 \\ -0.1424 & 0.5668 \\ -1.7054 & 0.3976 \end{bmatrix}.$$

The covariance matrices are given by

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 440.6502 & -837.6397 & -974.6520 & 470.6914 & 488.7108 & 238.3555 \\ -837.6397 & 2709.1462 & 3021.3604 & -1383.7775 & -1169.1515 & -684.6526 \\ -974.6520 & 3021.3604 & 3954.8088 & -1717.1222 & -1534.3128 & -781.5812 \\ 470.6914 & -1383.7775 & -1717.1222 & 972.7261 & 878.3539 & 299.7057 \\ 488.7108 & -1169.1515 & -1534.3128 & 878.3539 & 1107.5628 & 273.0765 \\ 238.3555 & -684.6526 & -781.5812 & 299.7057 & 273.0765 & 195.2260 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 496.6858 & -997.8638 & -1188.8008 & 1049.9881 & 1969.3574 & 155.4274 \\ -997.8638 & 22666.2063 & 13418.5349 & -7085.3169 & -16375.5283 & -5622.3652 \\ -1188.8008 & 13418.5349 & 10771.1775 & -6235.4205 & -12216.2856 & -3124.0482 \\ 1049.9881 & -7085.3169 & -6235.4205 & 4377.0516 & 8328.0910 & 1569.0545 \\ 1969.3574 & -16375.5283 & -12216.2856 & 8328.0910 & 17117.4840 & 3790.7315 \\ 155.4274 & -5622.3652 & -3124.0482 & 1569.0545 & 3790.7315 & 1454.2978 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_3 = \begin{bmatrix} 224.2354 & -404.5342 & -611.1761 & 427.0222 & 588.5578 & 77.4043 \\ -404.5342 & 2587.1254 & 3658.9929 & -2233.9834 & -3130.4829 & -432.4653 \\ -611.1761 & 3658.9929 & 5764.8167 & -3657.7987 & -5011.1844 & -541.0220 \\ 427.0222 & -2233.9834 & -3657.7987 & 2415.7610 & 3301.5483 & 303.6785 \\ 588.5578 & -3130.4829 & -5011.1844 & 3301.5483 & 4646.1970 & 439.2035 \\ 77.4043 & -432.4653 & -541.0220 & 303.6785 & 439.2035 & 86.6288 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_4 = \begin{bmatrix} 315.0019 & -264.4656 & -685.8318 & 747.6558 & 1283.3587 & -12.3680 \\ -264.4656 & 3813.9625 & 3670.6804 & -2014.9200 & -3696.1474 & -786.4141 \\ -685.8318 & 3670.6804 & 8609.8556 & -5565.7481 & -7754.9139 & -316.4647 \\ 747.6558 & -2014.9200 & -5565.7481 & 4213.5024 & 6198.9563 & 14.6824 \\ 1283.3587 & -3696.1474 & -7754.9139 & 6198.9563 & 9958.0470 & 210.5271 \\ -12.3680 & -786.4141 & -316.4647 & 14.6824 & 210.5271 & 217.1968 \end{bmatrix},$$

$$\boldsymbol{\Lambda}_1 = \begin{bmatrix} 3434.8112 & -218.7220 \\ -218.7220 & 14.3208 \end{bmatrix},$$

$$\boldsymbol{\Lambda}_2 = \begin{bmatrix} 4756.1141 & 607.1258 \\ 607.1258 & 898.7417 \end{bmatrix},$$

$$\boldsymbol{\Lambda}_3 = \begin{bmatrix} 2259.8418 & -92.1147 \\ -92.1147 & 64.7717 \end{bmatrix}.$$

The rotation matrix is given by

$$\mathbf{\Gamma} = \begin{bmatrix} -0.1351 & 0.0199 & 0.3437 & -0.2907 & -0.4602 & 0.0153 & -0.6364 & 0.4022 \\ -0.1977 & -0.8923 & -0.0581 & -0.1857 & -0.1267 & 0.2343 & 0.2198 & 0.0870 \\ -0.2452 & 0.1448 & -0.7487 & 0.1302 & -0.5816 & -0.0562 & 0.0069 & -0.0080 \\ -0.0522 & 0.2595 & 0.2418 & -0.0968 & -0.2388 & -0.0695 & 0.7094 & 0.5450 \\ 0.0801 & -0.1073 & -0.3414 & 0.3173 & 0.4387 & -0.0121 & -0.2076 & 0.7274 \\ 0.2432 & 0.1927 & -0.3661 & -0.7981 & 0.2093 & 0.2916 & 0.0154 & 0.0610 \\ 0.8892 & -0.1465 & -0.0057 & 0.1843 & -0.3740 & 0.1146 & 0.0101 & 0.0278 \\ 0.1538 & -0.2121 & -0.0943 & -0.2883 & 0.0277 & -0.9158 & 0.0002 & 0.0014 \end{bmatrix}.$$

Semi-Supervised

Recalling that BIC selected a model with conditional independence but not intrinsic independence with $G = 3$ and $\langle H_1, H_2, H_3 \rangle = \langle 4, 2, 3 \rangle$, we have that

$$\boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{bmatrix} = \begin{bmatrix} 0.1543 \\ 0.5647 \\ 0.1731 \\ 0.1080 \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \begin{bmatrix} 0.3924 \\ 0.0726 \\ 0.5350 \end{bmatrix}.$$

The primary and secondary clustering mean parameters are given by

$$\boldsymbol{\mu}_1 = \begin{bmatrix} -28.9153 \\ 227.5089 \\ 221.6169 \\ -114.5314 \\ -208.0146 \\ -34.8238 \end{bmatrix}, \qquad \boldsymbol{\mu}_2 = \begin{bmatrix} 3.7179 \\ -86.1419 \\ -39.6550 \\ -4.4542 \\ 23.6603 \\ 14.6131 \end{bmatrix},$$

$$\boldsymbol{\mu}_3 = \begin{bmatrix} 52.1938 \\ -10.1990 \\ -116.6432 \\ 121.7839 \\ 156.4276 \\ 5.3307 \end{bmatrix}, \qquad \boldsymbol{\mu}_4 = \begin{bmatrix} -61.7838 \\ 141.7932 \\ 77.7046 \\ -8.2628 \\ -77.2556 \\ -35.2118 \end{bmatrix},$$

$$\boldsymbol{\eta}_1 = \begin{bmatrix} 6.2399 \\ 7.4205 \end{bmatrix}, \qquad \boldsymbol{\eta}_2 = \begin{bmatrix} 44.4287 \\ -4.6681 \end{bmatrix},$$

$$\boldsymbol{\eta}_3 = \begin{bmatrix} -3.8371 \\ -2.5258 \end{bmatrix}.$$

The regression matrices are given by

$$\mathbf{B}_1 = \begin{bmatrix} -0.4539 & -0.3894 \\ 1.7909 & 0.3050 \\ 0.7359 & 0.5112 \\ 0.5724 & 0.5338 \\ 0.4043 & 0.5731 \\ -2.7105 & 0.4559 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} -1.0674 & 0.1422 \\ 2.0961 & -0.0090 \\ 0.7799 & 1.1501 \\ 0.4053 & 1.4176 \\ 0.3430 & 0.3098 \\ -0.2478 & 0.0990 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} -0.0631 & -0.5023 \\ 1.6452 & 0.2872 \\ 1.1412 & 0.5952 \\ 1.7234 & 0.7403 \\ -0.1424 & 0.5668 \\ -1.7054 & 0.3976 \end{bmatrix}.$$

The covariance matrices are given by

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 440.6502 & -837.6397 & -974.6520 & 470.6914 & 488.7108 & 238.3555 \\ -837.6397 & 2709.1462 & 3021.3604 & -1383.7775 & -1169.1515 & -684.6526 \\ -974.6520 & 3021.3604 & 3954.8088 & -1717.1222 & -1534.3128 & -781.5812 \\ 470.6914 & -1383.7775 & -1717.1222 & 972.7261 & 878.3539 & 299.7057 \\ 488.7108 & -1169.1515 & -1534.3128 & 878.3539 & 1107.5628 & 273.0765 \\ 238.3555 & -684.6526 & -781.5812 & 299.7057 & 273.0765 & 195.2260 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 496.6858 & -997.8638 & -1188.8008 & 1049.9881 & 1969.3574 & 155.4274 \\ -997.8638 & 22666.2063 & 13418.5349 & -7085.3169 & -16375.5283 & -5622.3652 \\ -1188.8008 & 13418.5349 & 10771.1775 & -6235.4205 & -12216.2856 & -3124.0482 \\ 1049.9881 & -7085.3169 & -6235.4205 & 4377.0516 & 8328.0910 & 1569.0545 \\ 1969.3574 & -16375.5283 & -12216.2856 & 8328.0910 & 17117.4840 & 3790.7315 \\ 155.4274 & -5622.3652 & -3124.0482 & 1569.0545 & 3790.7315 & 1454.2978 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_3 = \begin{bmatrix} 224.2354 & -404.5342 & -611.1761 & 427.0222 & 588.5578 & 77.4043 \\ -404.5342 & 2587.1254 & 3658.9929 & -2233.9834 & -3130.4829 & -432.4653 \\ -611.1761 & 3658.9929 & 5764.8167 & -3657.7987 & -5011.1844 & -541.0220 \\ 427.0222 & -2233.9834 & -3657.7987 & 2415.7610 & 3301.5483 & 303.6785 \\ 588.5578 & -3130.4829 & -5011.1844 & 3301.5483 & 4646.1970 & 439.2035 \\ 77.4043 & -432.4653 & -541.0220 & 303.6785 & 439.2035 & 86.6288 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_4 = \begin{bmatrix} 315.0019 & -264.4656 & -685.8318 & 747.6558 & 1283.3587 & -12.3680 \\ -264.4656 & 3813.9625 & 3670.6804 & -2014.9200 & -3696.1474 & -786.4141 \\ -685.8318 & 3670.6804 & 8609.8556 & -5565.7481 & -7754.9139 & -316.4647 \\ 747.6558 & -2014.9200 & -5565.7481 & 4213.5024 & 6198.9563 & 14.6824 \\ 1283.3587 & -3696.1474 & -7754.9139 & 6198.9563 & 9958.0470 & 210.5271 \\ -12.3680 & -786.4141 & -316.4647 & 14.6824 & 210.5271 & 217.1968 \end{bmatrix},$$

$$\boldsymbol{\Lambda}_1 = \begin{bmatrix} 3434.8112 & -218.7220 \\ -218.7220 & 14.3208 \end{bmatrix},$$

$$\boldsymbol{\Lambda}_2 = \begin{bmatrix} 4756.1141 & 607.1258 \\ 607.1258 & 898.7417 \end{bmatrix},$$

$$\boldsymbol{\Lambda}_3 = \begin{bmatrix} 2259.8418 & -92.1147 \\ -92.1147 & 64.7717 \end{bmatrix}.$$

The rotation matrix is given by

$$\boldsymbol{\Gamma} = \begin{bmatrix} -0.1351 & 0.0199 & 0.3437 & -0.2907 & -0.4602 & 0.0153 & -0.6364 & 0.4022 \\ -0.1977 & -0.8923 & -0.0581 & -0.1857 & -0.1267 & 0.2343 & 0.2198 & 0.0870 \\ -0.2452 & 0.1448 & -0.7487 & 0.1302 & -0.5816 & -0.0562 & 0.0069 & -0.0080 \\ -0.0522 & 0.2595 & 0.2418 & -0.0968 & -0.2388 & -0.0695 & 0.7094 & 0.5450 \\ 0.0801 & -0.1073 & -0.3414 & 0.3173 & 0.4387 & -0.0121 & -0.2076 & 0.7274 \\ 0.2432 & 0.1927 & -0.3661 & -0.7981 & 0.2093 & 0.2916 & 0.0154 & 0.0610 \\ 0.8892 & -0.1465 & -0.0057 & 0.1843 & -0.3740 & 0.1146 & 0.0101 & 0.0278 \\ 0.1538 & -0.2121 & -0.0943 & -0.2883 & 0.0277 & -0.9158 & 0.0002 & 0.0014 \end{bmatrix}.$$

C.2.3 93 Cars Dataset

Recalling that BIC selected a model with conditional independence but not intrinsic independence with $G = 3$ and $H = 2$, we have that

$$\boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix} = \begin{bmatrix} 0.8673 \\ 0.1327 \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 0.6489 \\ 0.2974 \\ 0.0538 \end{bmatrix}.$$

The primary and secondary clustering mean parameters are given by

$$\boldsymbol{\mu}_1 = \begin{bmatrix} -4.6984 \\ -125.7398 \end{bmatrix}, \qquad \boldsymbol{\mu}_2 = \begin{bmatrix} 31.1875 \\ 380.1100 \end{bmatrix}, \qquad \boldsymbol{\mu}_3 = \begin{bmatrix} -115.7834 \\ -584.6934 \end{bmatrix},$$
$$\boldsymbol{\eta}_1 = \begin{bmatrix} -3.1018 \\ 19.2254 \end{bmatrix}, \qquad \boldsymbol{\eta}_2 = \begin{bmatrix} 27.0032 \\ -358.8575 \end{bmatrix}.$$

The regression matrices are given by

$$\mathbf{B}_1 = \begin{bmatrix} 0.4083 & 8.5755 \\ -0.4933 & 0.3102 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 0.7568 & -0.8025 \\ -0.5013 & -0.3371 \end{bmatrix}.$$

The covariance matrices are given by

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 545.7705 & 0.0000 \\ 0.0000 & 30894.2732 \end{bmatrix}, \qquad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 37.2384 & 0.0000 \\ 0.0000 & 93570.2397 \end{bmatrix},$$
$$\boldsymbol{\Sigma}_3 = \begin{bmatrix} 142.3097 & 0.0000 \\ 0.0000 & 1962.7674 \end{bmatrix},$$
$$\boldsymbol{\Lambda}_1 = \begin{bmatrix} 53.7494 & 0.0000 \\ 0.0000 & 39641.8026 \end{bmatrix}, \qquad \boldsymbol{\Lambda}_2 = \begin{bmatrix} 24.0091 & 0.0000 \\ 0.0000 & 27184.6035 \end{bmatrix}.$$

The rotation matrix is given by

$$\boldsymbol{\Gamma} = \begin{bmatrix} -0.0114 & -0.1543 & 0.0752 & -0.2249 & 0.9591 \\ -0.0586 & -0.5706 & 0.2579 & -0.7243 & -0.2825 \\ 0.0071 & -0.6855 & 0.3280 & 0.6498 & 0.0165 \\ 0.8845 & 0.1696 & 0.4318 & -0.0485 & -0.0074 \\ -0.4627 & 0.3898 & 0.7961 & 0.0145 & -0.0018 \end{bmatrix}.$$

C.2.4 Handwritten Digits Dataset

Recalling that BIC selected a model with conditional independence but not intrinsic independence with $G = 5$ and $H = 3$, we have that

$$\boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \end{bmatrix} = \begin{bmatrix} 0.1876 \\ 0.2035 \\ 0.2905 \\ 0.0068 \\ 0.3115 \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 0.3235 \\ 0.4262 \\ 0.2503 \end{bmatrix}.$$

The primary and secondary clustering mean parameters are given by

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 1856.5336 \\ -130.3318 \end{bmatrix}, \qquad \boldsymbol{\mu}_2 = \begin{bmatrix} 3470.8475 \\ -138.8670 \end{bmatrix}, \qquad \boldsymbol{\mu}_3 = \begin{bmatrix} 5961.5899 \\ -163.0458 \end{bmatrix},$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 6775.9996 \\ -173.8813 \end{bmatrix}, \qquad \boldsymbol{\mu}_5 = \begin{bmatrix} 10658.3140 \\ -209.0189 \end{bmatrix}, \qquad \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} = \begin{bmatrix} -0.8262 \\ -1.0759 \\ -0.7641 \end{bmatrix}.$$

The regression matrices are given by

$$\mathbf{B}_1 = \begin{bmatrix} 0.0308 \\ 0.3229 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 0.0308 \\ 0.3207 \end{bmatrix}, \quad \mathbf{B}_3 = \begin{bmatrix} 0.0309 \\ 0.3233 \end{bmatrix}.$$

The covariance matrices are given by

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 55224.5560 & -687.2779 \\ -687.2779 & 18.8778 \end{bmatrix}, \qquad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 243268.4210 & -2483.0248 \\ -2483.0248 & 33.4820 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_3 = \begin{bmatrix} 3131740.9966 & -23299.6526 \\ -23299.6526 & 195.5277 \end{bmatrix}, \qquad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 2768020.7865 & -63193.7137 \\ -63193.7137 & 1452.2865 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_5 = \begin{bmatrix} 5934825.0438 & -50317.0775 \\ -50317.0775 & 453.0749 \end{bmatrix}, \qquad \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \\ \Lambda_3 \end{bmatrix} = \begin{bmatrix} 0.0044 \\ 0.0038 \\ 0.0028 \end{bmatrix}.$$

The rotation matrix is given by

$$\boldsymbol{\Gamma} = \begin{bmatrix} 0.0063 & -0.9526 & -0.3041 \\ 0.0294 & 0.3041 & -0.9522 \\ 0.9995 & -0.0029 & 0.0300 \end{bmatrix}.$$

C.3 Cross-Tabulations

C.3.1 93 Cars dataset

We provide in Table C.1 the cross-tabulation of the best fitted model against the joint class of the observable class labels Cylinder, Type, and AirBags. Each class label is expressed

as a tuple (Cylinder, Type, Airbags), and the fitted class labels are expressed as $g{:}h$.

C.4 Computation Time

Due to the scope of the model space of the proposed model and the use of exhaustive search with the Bayesian Information Criterion to perform model selection, we used a 120-core server to perform the estimation procedure in parallel. For example, in the crabs dataset, evaluating 2520 different models takes approximately 15 minutes, in the olive oil dataset 57792 models takes approximately 8 hours, in the cars dataset 41280 models takes approximately 1 hour, and in the handwritten digits dataset 73840 models takes approximately 3 hours. Each individual model specification can be estimated in a few minutes, though this depends on the complexity implied by p_x, p_y, p_u, G, H_g and the type of independence.

Table C.1: Clustering labels for 93 cars dataset using the proposed model tabulated against the Cylinder, Type, and AirBags simultaneously. Missing combinations of class labels are omitted.

(Cylinder, Type, AirBags)	1:1	1:2	2:1	2:2	3:1	3:2
(3, Small, None)	0	0	3	0	0	0
(4, Compact, Driver & Passenger)	2	0	0	0	0	0
(4, Compact, Driver only)	6	0	2	0	0	0
(4, Compact, None)	2	0	3	0	0	0
(4, Midsize, Driver only)	2	0	3	0	0	0
(4, Midsize, None)	1	0	1	0	0	0
(4, Small, Driver only)	3	0	2	0	0	0
(4, Small, None)	6	0	7	0	0	0
(4, Sporty, Driver & Passenger)	1	0	0	0	0	0
(4, Sporty, Driver only)	3	0	2	0	0	0
(4, Sporty, None)	0	0	2	0	0	0
(4, Van, Driver only)	0	1	0	0	0	0
(5, Midsize, Driver & Passenger)	1	0	0	0	0	0
(5, Van, None)	0	0	0	1	0	0
(6, Compact, Driver only)	1	0	0	0	0	0
(6, Large, Driver & Passenger)	2	1	0	0	0	0
(6, Large, Driver only)	4	0	0	0	0	0
(6, Midsize, Driver & Passenger)	5	0	0	0	0	0
(6, Midsize, Driver only)	5	0	0	0	0	0
(6, Midsize, None)	1	0	1	0	0	0
(6, Sporty, Driver & Passenger)	2	0	0	0	0	0
(6, Sporty, Driver only)	0	0	0	0	1	0
(6, Sporty, None)	0	1	0	0	0	0
(6, Van, Driver only)	0	2	0	0	0	0
(6, Van, None)	1	4	0	0	0	0
(8, Large, Driver & Passenger)	1	0	0	0	0	0
(8, Large, Driver only)	3	0	0	0	0	0
(8, Midsize, Driver & Passenger)	0	0	0	0	1	0
(8, Midsize, Driver only)	0	0	0	0	1	0
(8, Sporty, Driver only)	0	0	0	0	0	1
(rotary, Sporty, Driver only)	0	0	0	0	0	1

Appendix D

Generalized Linear Models for Massive Data via Doubly-Sketching

D.1 Comparison to IRLS

Within this simulation, a large variety of configurations were tested with the highlights of the results distilled into the main work. The less interesting and banal results are presented here. Figure D.1 shows the recovered parameter MSE grouped by all of the simulation parameter configurations and summarized as an average. Based on this figure, we can observe that the general shape of the trend is the same for many of the parameter choices. Moreover, the MSE values for both IRLS and the doubly-sketched method are of similar magnitude and so is visualized as a ratio in the main work.

Overall, we see reasonable recovery against the true $\boldsymbol{\beta}$ by both IRLS and the proposed sketching method except in the case of $k = 100$, where the sketched surrogate dataset size is very aggressively chosen. As noted previously, the case of $d = k = 100$ is a case where each sketched Hessian matrix is at most rank 100 and at risk of being poorly-conditioned. This is exacerbated by the choice of Clarkson-Woodruff sketch; it is possible $\mathbf{S}_{\text{CW}}$ does not allocate any first-sketch observations $1, 2, ..., m$ to a second-sketch surrogate observation, causing $\tilde{\mathbf{X}} \in \mathbb{R}^{100\times 100}$ to have a zero row and thus $\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}$ to be singular.

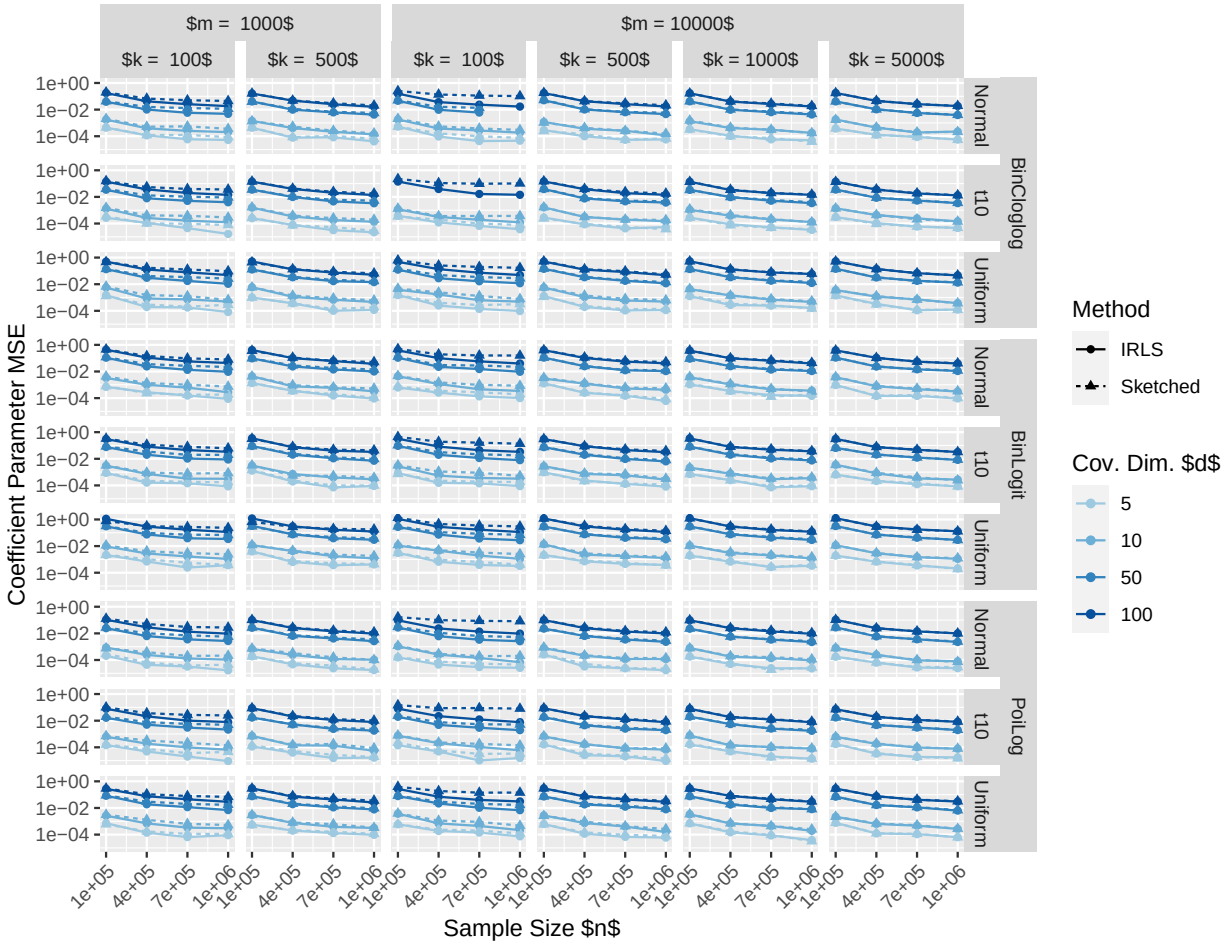

Figure D.1: Coefficient MSE of the doubly-sketched and IRLS estimation procedure against the true parameters. MSE values are indicated individually by points, with group-wise averages across replications joined by line segments.

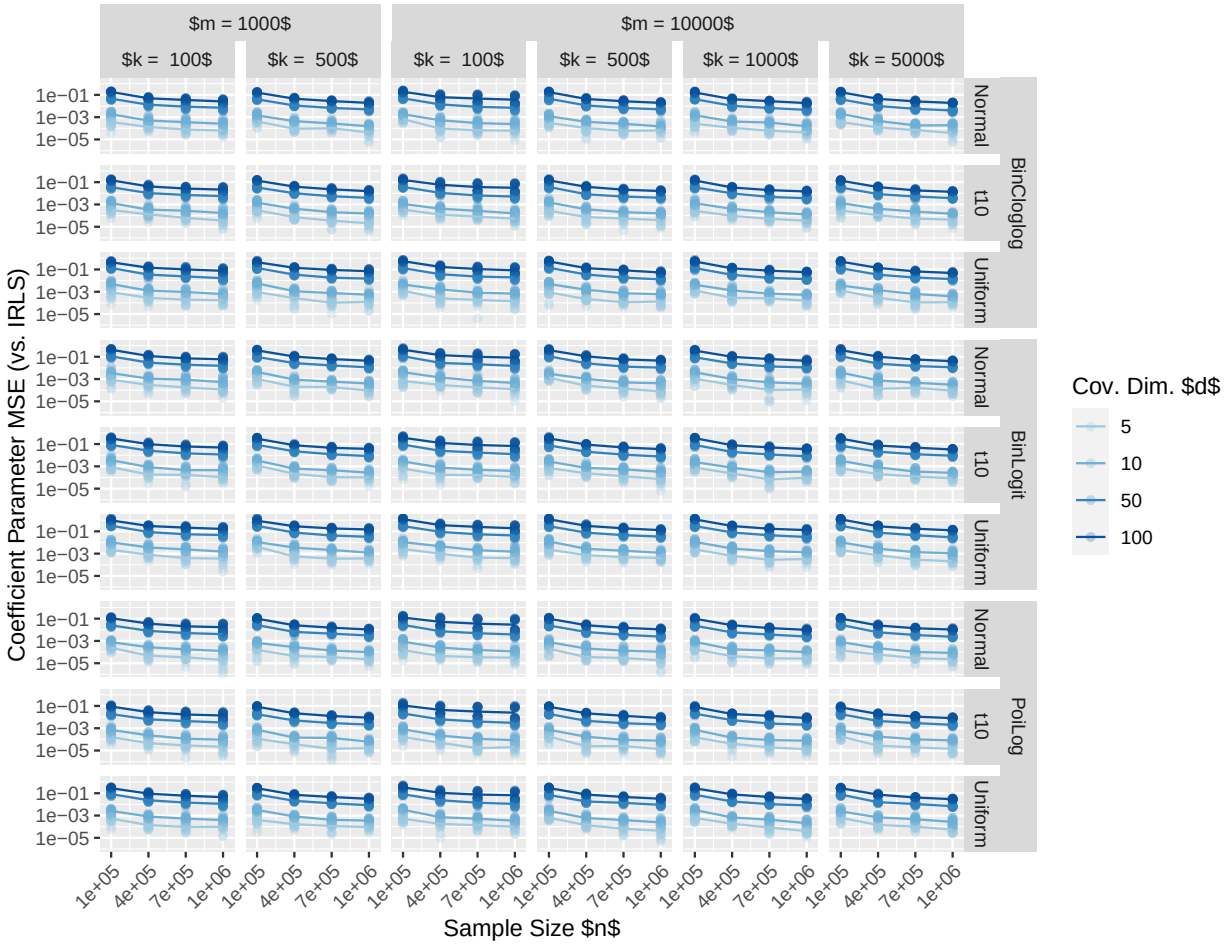

Figure D.2: Coefficient MSE of the doubly-sketched parameters against the IRLS fitted parameters. MSE values are indicated individually by points, with group-wise averages across replications joined by line segments.

In addition to the recovery of the true $\boldsymbol{\beta}$, we also examine the result of the doubly-sketched estimate against the IRLS estimate in Figure D.2. This is again measured using the MSE, and quantifies how well the proposed doubly-sketched method approximates the full data IRLS solution in the long-run. We find the effects of sketch sizes m and k to have the greatest effect on the distance between the estimates $\boldsymbol{\beta}_{\text{Sketch}}$ and $\boldsymbol{\beta}_{\text{IRLS}}$.

D.2 Testing Hardware

Computation time was calculated on a personal laptop computer with an Intel Core i7-3720QM, 32GB of RAM, a 500GB SSD and a 500GB 7200RPM hard drive attached via wired gigabit Ethernet to a network shared by the network server. All timed simulation runs were run sequentially; only the parameter recovery simulation was run in parallel and untimed.

When testing data stored across the network, we stored the data in a PostgreSQL 14.2 database hosted on a TrueNAS server on the same local network. The server is equipped with an AMD Athlon 3000G, 16GB of RAM and four 8TB 7200RPM hard drives in a ZFS pool with RAIDZ2 redundancy. No L2ARC is enabled, and the TrueNAS dataset holding the PostgreSQL server has a record size of 16 KiB.

D.3 New York Yellow Taxicab Dataset

The dataset was obtained from NYC Taxi and Limousine Commission (2022) for the dates from January 2009 to January 2023 inclusive. For months with more than one type of taxicab data, only the yellow taxicab data was used. Each month of data is provided in a single Parquet file. Unfortunately, the data columns are not consistent across months, nor are the contents consistent with the provided data dictionary. The following section describes how the dataset was pre-processed before usage. For each month, we load the CSV into R R Core Team (2019) and edit column names using the following patterns:

1. Convert column names to lowercase.
2. Remove leading `trip_` or `tpep_`.
3. Rename any column containing `vendor` to `vendor_id`.
4. Replace leading `start_` with `pickup_`.
5. Replace leading `end_` with `dropoff_`.
6. Replace substring `amt` with `amount`.

We adjust the column contents as follows:

1. Despite the data dictionary only listing two possible choices for Vendor ID, there are multiple possibilities listed in the data. We code factor levels 1 and 2 in accordance with the data dictionary, and use 3 for any other value including missing values.
2. The payment type in some months is provided as text with a variety of spellings and abbreviations of the intended payment type. The data dictionary encodes the intended values as numerical values. To be consistent with the latter, we interpreted and recoded the text as lowercase with the mapping:

   - `crd`, `cre`, and `credit` $\rightarrow$ 1
   - `cas`, `cash`, and `csh` $\rightarrow$ 2
   - `no`, `no charge`, and `noc` $\rightarrow$ 3
   - `dis`, `dispute`, and `credit` $\rightarrow$ 4
   - `unk` $\rightarrow$ 5

   All other values were recoded as missing. No instance of voided trips was found in months requiring recoding.

3. When payment type is provided as an integer as specified in the data dictionary, there are values beyond the valid levels of 1 through 6. These were also recoded as missing. This resulted in no occurrences of the level 6 and so only factor levels 1 through 5 were used in the final analysis.

We retain the covariate columns `vendor_id`, `payment_type`, `duration`, `pickup_datetime`, `dropoff_datetime`, `distance`, `fare_amount`, `tip_amount`, `tolls_amount`, `total_amount`, and the response column `passenger_count`. This is done due to the quantity of missing data in other columns, and inconsistent availability across months. The resulting month's complete cases are written out to a new CSV file. This procedure is repeated for all 180 months of data.

A table was created in PostgreSQL with the following schema:

```
CREATE TABLE yellowtaxi (
id serial8 not null,
passenger_count int2,
vendor_id int2,
payment_type int2,
pickup_datetime timestamp,
dropoff_datetime timestamp,
distance float8,
fare_amount float8,
tip_amount float8,
tolls_amount float8,
total_amount float8,
primary key (id)
)
WITH (fillfactor=100)
```

A `serial8` column was used to assign a unique identifier to each row for use during sketching; in principle, the first Uniform sketch will specify the rows to retrieve. This

simultaneously doubles as a primary key for indexing, which can speed data retrieval. However, due to the way PostgreSQL functions, among other SQL databases, retrieving a single row requires retrieving the entire page in which it is stored from disk. This page can contain a varying number of other rows, the majority of which are discarded as it is unlikely that the page contains another sampled index. As a result, this is incredibly inefficient by magnifying the necessary amount of reads, compounded upon by the fact that spinning mechanical hard drives perform poorly under random access.

D.3.1 Sampling via TABLESAMPLE

In lieu of using a true simple random sample with replacement, we leverage a function built into PostgreSQL to approximately sample from the database with greater performance. Again, data in a PostgreSQL table is stored in pages which can contain many different rows; moreover, pages may not hold a consistent number of rows. Reading one row from a page requires reading the entire page anyway, so one may simply opt to sample over the set of pages. From a technical perspective, reading in one page of 100 rows is much faster than reading in 100 rows across 100 pages. The `TABLESAMPLE` operation defined in PostgreSQL returns a percentage of pages in the table as an approximation to sampling rows within the table and offers a very large performance benefit in the context of Uniform sketching; however, there is a trade-off. In the case where rows are stored in some sort of sequence or with some degree of clustering, such that a single page will tend to store similar rows, then `TABLESAMPLE` may be less effective as the variation in the sampled data is lower than would be under true SRSWOR. From a sampling point of view, this is effectively Poisson sampling over the set of pages in the dataset, which we treat as an approximation to uniform sketching for large n. Due to the sequential manner in which data is loaded, pages are filled with sequential rows from the dataset.