Beyond the Standard Mixture Model: Novel Families of Parsimonious Model-Based Clustering Methods

Keefe Murphy
13204117

This thesis is submitted to University College Dublin in fulfillment of the requirements for the degree of

Doctor of Philosophy

School of Mathematics and Statistics
Head of School: Prof. Brendan Murphy
Supervisors: Assoc. Prof. Claire Gormley and Prof. Brendan Murphy

Doctoral Studies Panel Members:
Prof. Nial Friel and Dr. Derek Greene

September 2019

« *Fais apparaître ce qui sans toi ne serait peut-être jamais vu* »

– Robert Bresson

For my parents, Eleanor and Bryan,
and in memory of Molly, Theresa, and Sneasel.

Acknowledgements

It is my pleasure to thank all of those who have helped me bring this thesis to fruition. Firstly, I would like to thank my supervisors, Assoc. Prof. Claire Gormley and Prof. Brendan Murphy. Both have provided fantastic guidance throughout my time as a postgraduate student and been an absolute pleasure to collaborate with. Anyone would be fortunate to have one supervisor as helpful and inspiring as Brendan or Claire; I feel extremely privileged to have benefited from the tutelage of both. If I am ever given the opportunity to mentor students of my own, I hope to do justice to the examples set by these two brilliant statisticians.

I am sure Claire and Brendan would join me in saying '*molte grazie*' to our co-authors Prof. Cinzia Viroli and Assoc. Prof. Raffaella Piccarreta. Their thoughtful contributions elevated this research in many ways.

In addition, Prof. Pierre Alquier, Prof. William Watson, and Dr. James Sweeney have been instrumental in illuminating my path in UCD. I owe them a debt of gratitude for their supervision of my MSc. thesis, their supervision during my earlier time as a research assistant in biostatistics, and most of all for encouraging me to pursue a doctoral programme in the first place.

All staff members in the School of Mathematics and Statistics have aided my progress immeasurably. In particular, I would like to thank those in the school office, especially Dr. Nuria García Ordiales, Ms. Rhona Preston, and Ms. Genevieve Peterson, and the members of my Doctoral Studies panel, Prof. Nial Friel and Dr. Derek Greene. They have always been on hand to entertain my queries and ensure I had the necessary resources at my disposal. I am humbled that the School entrusted me with several invaluable lecturing opportunities. It would

be remiss of me not to acknowledge that my experiences teaching Statistical Machine Learning helped inform some of the work in this thesis.

This research was supported by the Science Foundation Ireland funded Insight Centre for Data Analytics. This grant allowed me to attend many interesting graduate courses, working groups, and conferences, while visiting parts of the world I had never been to before. The members of Prof. Adrian Raftery's Working Group on Model Based Clustering and the presenters at UCD's own Working Group on Statistical Learning seminar series provided many stimulating research ideas and engaged in constructive discussions.

Within the student circle, I have enjoyed the camaraderie and support of many colleagues. Two fellow graduate students have been uniquely helpful in this regard; Alan Benson, for teaching me some of the finer points of numerical stability, and Sen Hu for his cherished friendship and many inputs over innumerable coffee-breaks. I wish them both success in their future careers and hope our paths cross again.

Outside the statistical bubble, I would like to thank my extended family as well as my friends from Limerick and the NCH Gamelan Orchestra. Begrudgingly, I must also acknowledge Rollo and Ornette, without whom this thesis would have been completed much sooner.

Without the love and sustenance of my partner Danielle, on the other hand, this thesis would never have been completed at all. Danielle, you give the best advice, listen with the greatest curiosity, and provide the keenest assistance. Serendipitously, I met you one week before commencing my Master's degree in UCD. I couldn't have completed the journey from then to now without your patience, humour, encyclopedic knowledge of cinema, and superlative baking skills, and I certainly can't imagine the rest of life's journey without you either.

I give my final thank you to my wonderful parents, to whom this thesis is dedicated. Bryan, thank you for providing many welcome distractions. Eleanor, thank you for making sure I always did my homework. I hope I have made you both proud.

Statement of Original Authorship

I hereby certify that the submitted work is my own work, was completed while registered as a candidate for the degree stated on the Title Page, and I have not obtained a degree elsewhere on the basis of the research presented in this submitted work.

Keefe Murphy

3rd September 2019

Sponsor

This work was supported by the Science Foundation Ireland funded Insight Centre for Data Analytics in University College Dublin under grant number SFI/12/RC/2289_P2.

Contents

Abstract

This thesis takes as its starting point a well-known family of finite mixtures of parsimoniously parameterised multivariate Gaussian distributions. Three main limitations of this model family are identified. Firstly, covariates are not incorporated into the clustering process. Secondly, the assumption of multivariate normality for the underlying component distributions is inappropriate for categorical data. Thirdly, the models are generally not well suited to high-dimensional settings where the number of variables is comparable to or even greater than the number of observations.

Thus, the standard finite mixture model is extended in three separate streams, with a full family of methods introduced in each case. Parsimony and model selection are common themes throughout. Maximum likelihood estimation or Bayesian estimation are used as appropriate to the task at hand.

Firstly, the finite Gaussian mixture model is extended to accommodate covariates of mixed type, by combining a range of parsimonious constraints on the component covariance matrices with the special cases of the mixtures of experts framework.

Secondly, a parsimonious family of mixtures of exponential-distance models are developed for clustering categorical sequence trajectories, motivated by an application to social survey data on the monthly employment activities of a cohort of Northern Irish youths.

Finally, the infinite mixture of infinite factor analysers model is presented as a computationally efficient and somewhat choice-free approach for clustering high-dimensional data. It assumes factor-analytic

covariance structures under a fully Bayesian nonparametric framework which theoretically allows infinitely many mixture components and infinitely many latent factors within each component simultaneously. The model is inferred using an adaptive MCMC algorithm which facilitates automatic estimation of these quantities.

A number of applications in each chapter illustrate the performance of the proposed families of methods. All novel models in this thesis are implemented in distributed software packages.

Collaborations

Isobel Claire Gormley	As my principal supervisor, Assoc. Prof. Gormley collaborated on the work in Chapter 3 and Chapter 4.
Thomas Brendan Murphy	As my co-supervisor, Prof. Murphy collaborated on the work in Chapter 2 and Chapter 3.
Raffaella Piccarreta	Assoc. Prof. Piccarreta (Università Bocconi, Milan) led the organisation and writing of the introductory literature review and provided insights for the interpretation of the model results in Chapter 3.
Cinzia Viroli	Prof. Viroli (Università di Bologna) collaborated on the work in Chapter 4 during a visit by Assoc. Prof. Gormley and I to Bologna. In particular, Prof. Viroli contributed some R code for the slice sampling and conducted the comparisons to the DP-BP and MFMA methods throughout Section 3.
Lorraine Brennan	Prof. Brennan (UCD) kindly provided the spectral metabolomic data set analysed in Section 4.3.2 of Chapter 4.

Publications

The material in Chapter 2 has been published in the peer-reviewed journal *Advances in Data Analysis and Classification*; the article is available online in advance of appearing in a special issue on model-based clustering. The material in Chapter 3 has been submitted to the *Journal of the Royal Statistical Society: Series A (Statistics in Society)* and is currently under review. Finally, the material in Chapter 4 has appeared as an advance online publication in the peer-reviewed journal *Bayesian Analysis*.

Peer-reviewed journal papers:

- Murphy, K. and T. B. Murphy (2019). Gaussian parsimonious clustering models with covariates and a noise component. *Advances in Data Analysis and Classification*, advance publication, 1–33. URL https://doi.org/10.1007/s11634-019-00373-8.
- Murphy, K., C. Viroli, and I. C. Gormley (2019). Infinite mixtures of infinite factor analysers. *Bayesian Analysis*, advance publication, 1–27. URL https://projecteuclid.org/euclid.ba/1570586978

Submitted articles (under review):

- Murphy, K., T. B. Murphy, R. Piccarreta, and I. C. Gormley (2019). Clustering longitudinal life-course sequences using mixtures of exponential-distance models. Under review with the *Journal of the Royal Statistical Society: Series A (Statistics in Society)*. *arXiv* pre-print: https://arxiv.org/abs/1908.07963

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is presented in the form of three distinct, self-contained chapters, each with their own introduction, conclusion, bibliography, and appendices. This overall introduction chapter aims to outline the content contained within the chapters that follow and to draw parallels between their overlapping themes and purposes, where appropriate. Broadly speaking, this thesis describes some extensions of model-based clustering methods using novel families of parsimonious mixture models.

1.1 Motivation

Clustering methods, in a general sense, are used to uncover group structure in heterogeneous populations and identify patterns in a data set which may represent distinct subpopulations. While there is no universally applicable definition of what constitutes a cluster (Hennig, 2015), it is commonly assumed that clusters should be well separated from each other and cohesive in an ideal analysis (Everitt et al., 2011). Conversely, objects within a cluster should be more similar to each other in some sense, in such a way that an observation has a defined relationship with observations in the same cluster, but not with observations from other clusters.

Clustering methods can be broadly grouped into heuristic algorithms such as partitioning-around-medoids (PAM; Kaufman and Rousseeuw, 1990) and hierarchical clustering on one hand and model-based clustering (MBC) approaches on the other. The MBC paradigm assumes that data arise from a (usually finite) mixture of probability distributions (McLachlan and Peel, 2000; Bouveyron et al., 2019). Mix-

ture models can be thought of as a type of MBC method where each observation in a data set is assigned to a component probability distribution. Ideally, mixtures of distributions are supposed to provide a good model for the heterogeneity in a data set; that is, once an observation has been assigned to a component, it is assumed to be well-represented by the component distribution. Typically, a one-to-one correspondence is assumed between component distributions and clusters (Fraley and Raftery, 2002), though this is not always the case (Hennig, 2010). The inferred underlying parameters of the component distributions can be used to summarise the subpopulations and the indices of the component to which each observation belongs can be used to define the clustering partition.

Inference for finite mixture models became much more straightforward with the introduction of the expectation-maximisation (EM) algorithm (Dempster et al., 1977), and in particular with the introduction of the incomplete data formulation, in which each observation's component membership is treated as the "missing" latent variable which must be estimated. This formulation assumes that the data are *conditionally* independent and identically distributed, where the conditioning is with respect to a latent variable representation of the data (Blei et al., 2003) in which the latent variable indicates component — and hence cluster — membership. The EM algorithm for model-based clustering will be described more fully in Chapters 2 and 3, in which it is employed.

Several alternatives to the EM algorithm are available; under a Bayesian specification, one prominent example is the class of methods referred to as Markov chain Monte Carlo (MCMC; Gelfand and Smith, 1990), where the goal is to sample from a distribution of interest in an iterative manner, rather than to identify the values which maximise it. While such methods typically require increased computational cost, their used is necessitated in Chapter 4 by virtue of the flagship model therein being a fully Bayesian nonparametric *infinite* mixture.

Though Agresti (2002) highlights the connection between MBC and an earlier method, latent class analysis (Lazarsfeld and Henry, 1968), and McNicholas (2016) argues that the notion of defining a cluster as a component in a mixture model can be traced back even further to Tiedeman (1955) and Wolfe (1965), the term 'model-based clustering' was largely popularised with the introduction of Gaussian parsimonious clustering models (GPCMs) by Banfield and Raftery

(1993) and Celeux and Govaert (1995). Indeed, the term 'model-based clustering' is often synonymous for many with the notion of a finite Gaussian mixture model.

Finite Gaussian mixture models assume data arise from a mixture distribution of the following form

$$f(\mathbf{y}_i \,|\, \boldsymbol{\theta}) = \sum_{g=1}^{G} \tau_g \phi(\mathbf{y}_i \,|\, \boldsymbol{\theta}_g = \{\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\}) \,, \tag{1.1}$$

where G is the number of components, $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$ are the observed data of dimension p, and $\phi(\mathbf{y}_i \,|\, \cdot)$ denotes the density of a p-variate Gaussian evaluated at $\mathbf{y}_i$. The component mixing proportions, $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_G)$ are such that $\tau_g > 0 \; \forall \; g$ and $\sum_{g=1}^{G} \tau_g = 1$. The collection of parameters specific to each component distribution $\boldsymbol{\theta}_g$ comprises $\boldsymbol{\mu}_g$, the mean parameters of the g-th component, and $\boldsymbol{\Sigma}_g$, its $p \times p$ covariance matrix. That the covariance matrices are explicitly modelled further differentiates Gaussian mixture models from other heuristic clustering approaches; the well-known k-means clustering algorithm (MacQueen, 1967), for instance, focusses only on detecting differences in mean signals and does not account for the dependencies among variables.

Within the models comprising the GPCM family, the component distributions are assumed to be multivariate Gaussians with parsimoniously parameterised covariance matrices. The range of constraints on the covariance matrices are described in detail in Chapter 2, where they are extensively employed. The influence of GPCMs is clear on a huge volume of recent works giving consideration to parsimony in mixture models; one which is particularly relevant from the perspective of Chapter 4 of this thesis uses a range of constrained factor-analytic structures in the component covariance matrices (McNicholas and Murphy, 2008). Parsimony is also a common theme of the family of models introduced in Chapter 3.

Owing to their ubiquity, the term 'standard mixture model' in the title of this thesis is taken to refer to the family of parameterised finite Gaussian mixture models described in (1.1), beyond which the novel families of parsimonious model-based clustering methods proposed in Chapters 2 to 4 extend in various ways. In particular, this model is limited by not incorporating information contained in related covariates, being inappropriate for categorical data, and being intractable in cases where p is comparable to or even greater than the number of observations n unless the restriction that the component covariance matrices are diagonal is assumed.

1.2 Chapter Summaries

In each of the following subsections, an overview is provided of Chapters 2–4 and the MoEClust, MEDseq, and IMIFA model families introduced therein. The rationale for the proposed extension beyond the model in (1.1) is presented in each case.

As these Chapters all exist in the form of journal articles either published (Chapters 2 and 4) or under review (Chapter 3), some material will be repeated and the notation used will differ in some instances, despite the efforts made to maximise cohesiveness as much as possible. Note that while each chapter contains appendices, Appendices 2.D, 2.E, 3.D, and 4.E do not appear in the corresponding journal articles, nor do Appendices 2.F, 3.E, and 4.F (see Section 1.3).

1.2.1 Chapter 2: MoEClust

The MoEClust family of models introduced in Chapter 2 extend the standard mixture model in (1.1) to the mixtures of experts (MoE) setting (Jacobs et al., 1991; Gormley and Frühwirth-Schnatter, 2019), in which either or both the component mixing proportions and/or component means are allowed to depend on related fixed covariates $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of possibly mixed type. Hence the covariates are used to guide the construction of the clusters, thereby making them endogenous to the model. This allows richer insight to be gleaned into the type of observation which characterises each cluster. In contrast, many analyses using finite Gaussian mixture models cluster the outcome variables $\mathbf{y}_i$ only and do not make reference to $\mathbf{x}_i$ until the uncovered clustering structure is investigated. For instance, the implementation of GPCMs in the popular R package `mclust` (Scrucca et al., 2016) does not accommodate covariates.

MoEClust models rely on the parsimonious parameterisations of $\boldsymbol{\Sigma}_g$ achieved in GPCMs by imposing constraints on the components of an eigen-decomposition of the form

$$\boldsymbol{\Sigma}_g = \lambda_g \mathbf{D}_g \mathbf{A}_g \mathbf{D}_g^\top.$$

The constraints encompass models with as few as 1 and as many as $Gp(p+1)/2$ covariance parameters. The GPCM family also includes models for univariate data where the variances are assumed to be equal or unequal across components.

The novel MoEClust family of Gaussian parsimonious MoE models is thus developed by combining the GPCM constraints with all special cases of the MoE framework defined by Gormley and Murphy (2011), whereby *different* subsets of covariates are allowed to influence either, neither, or both parts of the model. The full range of GPCM constraints have hitherto not been incorporated into MoE models. A simple trick involving the residuals of the seemingly unrelated regressions (Zellner, 1962) in the component distributions simplifies estimation of the constrained $\boldsymbol{\Sigma}_g$ matrices under the EM approach employed for model fitting.

Giving consideration to constraining the mixing proportions across components when they do not depend on covariates, by fixing $\tau_g = {}^1/_G \; \forall \; g$ as per Celeux and Govaert (1995), expands the model family further. So too does the potential inclusion of a uniform noise component for capturing non-Gaussian outliers, as per Banfield and Raftery (1993). Two ways of treating covariates in the mixing proportions of a model with a noise component are proposed. The model family could be instead described as GPCMs with various ways of accommodating covariates. Notably, the model in (1.1) is naturally subsumed into the MoEClust family as a special case.

Compared to non-parsimonious Gaussian MoE models, the reduction in the number of covariance parameters is shown to offset the number of parameters introduced by covariates. However, the interpretation of the model fundamentally differs depending on where covariates enter. A novel stepwise selection procedure is employed for identifying the optimal subset of covariates and where to put them, the number of components, and the set of GPCM covariance parameterisations. A novel initialisation procedure is employed to improve convergence of the EM algorithm used for model fitting when covariates enter the component distributions.

While some special cases of MoEClust are effectively mixtures of regressions with or without concomitant variables, the name *MoEClust* comes from employing MoE models, albeit with GPCM covariance structures, chiefly for clustering purposes. MoEClust models demonstrate significant improvement over both GPCMs without covariates incorporated and non-parsimonious Gaussian MoE models. This is borne out in applications to univariate data on CO_2 emissions (Hurn et al., 2003), with reference to Gross National Product *per capita*, and a multivariate data set containing hematological response variables for a cohort of Australian Institute of Sport athletes and related biological measurements (Cook and Weisberg, 1994).

1.2.2 Chapter 3: MEDseq

Motivated by an application to a data set containing information on the career sequence trajectories of a cohort of Northern Irish youths from the Status Zero Survey (henceforth referred to as the 'MVAD' data; McVicar, 2000; McVicar and Anyadike-Danes, 2002), the MEDseq family of models is introduced in Chapter 3. As the data are represented by an ordered collection of monthly employment activity states, the sense in which these models depart from the standard mixture model is most apparent by the data themselves being both categorical (with v categories) and longitudinal (with T time points). The MVAD data also contain information on different characteristics — related to gender, community, geographic and social conditions, and personal abilities — in the form of baseline covariates, as well as observation-specific sampling weights used to correct for response bias in the original survey.

The MEDseq family relies on exponential-distance models of the form

$$f(\mathbf{s}_i \mid \boldsymbol{\theta}, \lambda, \mathrm{d}) = \Psi(\lambda, \boldsymbol{\theta} \mid T, v)^{-1} \, e^{-\lambda \mathrm{d}(\mathbf{s}_i, \boldsymbol{\theta})}, \tag{1.2}$$

where $\mathbf{s}_i$ denotes an observed categorical sequence, $\mathrm{d}(\mathbf{s}_i, \boldsymbol{\theta})$ is a generic distance function from a location parameter $\boldsymbol{\theta}$, λ is a non-negative precision parameter, and $\Psi(\lambda, \boldsymbol{\theta} \mid T, v)$ is a normalising constant such that $f(\mathbf{s}_i \mid \boldsymbol{\theta}, \lambda, \mathrm{d})$ is a valid probability mass function (PMF). Distance-based models have been used by several authors (e.g. Mallows, 1957), and mixtures thereof have also been used for clustering (e.g. Murphy and Martin, 2003). Interestingly, exponential-distance models share some of the properties of the Gaussian distribution; indeed, when $\mathbf{s}_i$ is continuous and $\mathrm{d}(\mathbf{s}_i, \boldsymbol{\theta})$ is the squared Euclidean distance from the mean, the PMF in (1.2) relates to the density of the Gaussian distribution. Moreover, mixtures of exponential-distance models are shown to correspond to a model-based equivalent of PAM.

MEDseq models are effectively mixtures of exponential-distance models for sequences, hence the name, which employ the Hamming distance metric (Hamming, 1950). Sampling weights are also accounted for by weighting the likelihood function appropriately. The sense in which MEDseq models constitute a family of methods is two-fold. Firstly, baseline covariates are allowed to affect or not affect the probability of component membership, similar to one of the special cases of the MoEClust model family introduced in Chapter 2. Secondly, a range of settings for the preci-

sion parameter λ — allowing the parameter to be constrained or unconstrained across components and/or time points — define useful weighted generalisations of the Hamming distance and introduce parsimony to the MEDseq model family. Furthermore, models with a uniform noise component arise naturally by restricting the parameter space. Thus, the aim is not actually to induce parsimony, but to move between the simplest model with only a single precision parameter to more heavily parameterised models which provide greater flexibility.

Sequence analysis (SA) is the umbrella term used for a collection of methods used for analysing such life-course data. The properties of the weighted variants of the Hamming distance metric employed are contrasted to other dissimilarity measures often used within the SA community, namely optimal matching (OM; Abbott and Forrest, 1986; Abbott and Hrycak, 1990) and the dynamic Hamming distance (DHD; Lesnard, 2010), in terms of their implicitly assumed substitution costs measuring the dissimilarities between pairs of states. Notably, however, the normalising constant of an exponential-distance model using OM or DHD is not available in closed form, as a sum over all possible sequences is required. This is infeasible for even moderately large sequence lengths or numbers of categories. On the other hand, an exact expression for the normalising constant exists for MEDseq models based on the Hamming distance, thereby greatly simplifying model fitting.

It is common among the SA community to apply heuristic or partitional clustering algorithms to a dissimilarity matrix. In McVicar and Anyadike-Danes (2002), the categorical variable indicating cluster membership — obtained by applying Ward's hierarchical clustering (Ward, 1963) to an OM dissimilarity matrix — is treated as the input to a weighted multinomial logistic regression in order to relate the observed sequences to the covariates. MEDseq models, by virtue of being both model-based and distance-based and including both the covariates and the weights only once, in a simultaneous fashion, allow new insights to be gleaned from the MVAD data.

1.2.3 Chapter 4: IMIFA

Chapter 4 returns to the assumption that the component distributions are Gaussian. However, the focus here is on settings where the data dimension p is com-

parable to or even greater than n. In such settings, performing clustering on a dimensionally-reduced data set is typically computationally cheap, though caution is advised (Chang, 1983). Regularisation to ease covariance matrix inversion (e.g. Fraley and Raftery, 2007), LASSO-like penalisation methods (e.g. Zhou et al., 2009), and co-clustering algorithms (Govaert and Nadif, 2013) are viable alternatives. So too, however, are parsimonious mixture models. That said, the GPCM framework in (1.1) is limited when $n \leq p$ in that only models with diagonal covariance structures are tractable. Thus, subspace clustering approaches via models of the following form are adopted

$$f(\mathbf{y}_i \,|\, \boldsymbol{\theta}) = \sum_{g=1}^{G} \tau_g \phi\Big(\mathbf{y}_i \,|\, \boldsymbol{\theta}_g = \Big\{\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g \boldsymbol{\Lambda}_g^\top + \boldsymbol{\Psi}_g\Big\}\Big) \,. \tag{1.3}$$

Subspace clustering methods model data in low-dimensional subspaces; the model in (1.3) specifically assumes a factor-analytic covariance structure, where $\boldsymbol{\Lambda}_g$ is the cluster-specific factor loadings matrix of dimensions $p \times q$, where $q \ll p$ is the dimension of the subspace. This model — a (Gaussian) mixture of factor analysers (MFA; Ghahramani and Hinton, 1996; McLachlan et al., 2003) — typically requires estimating fewer parameters than the model in (1.1) for large p, having at most $G(pq - q(q-1)/2) + Gp$ covariance parameters. The difficulty is that both the number of components and the dimension of the subspace must be chosen. Typically, a range of models with different specified values of G and q are fitted and the pair which optimise some model selection criterion are chosen.

However, by simultaneously allowing infinitely many components and infinitely many latent factors within each component, both quantities can be estimated in an automatic fashion with only a single run. This is achieved under the Bayesian paradigm using nonparametric shrinkage priors. As well as reducing the computational burden associated with the search of the model space, this has the added advantages of allowing cluster-specific numbers of latent factors q_g and facilitating quantification of the uncertainty in the estimates $\widehat{G}$ and $\widehat{q}_g$.

Hence, the infinite mixture of infinite factor analysers model is introduced as the flagship model at the head of the IMIFA family of factor-analytic mixtures, incrementally built from the basic factor analysis model (Knott and Bartholomew, 1999)

to the most flexible and computationally efficient IMIFA model. The model family includes among its special cases finite mixtures of finite factor analysers, given by (1.3), finite mixtures of infinite factor analysers (MIFA), and infinite mixtures in which the number of factors is finite (IMFA). Versions which overfit the number of components are also included (Rousseau and Mengersen, 2011), namely the overfitted mixture of finite factor analysers (Papastamoulis, 2018) and the novel overfitted mixture of infinite factor analysers.

The IMIFA model relies on the use of a nonparametric Pitman-Yor process (PYP) prior (Perman et al., 1992; Pitman and Yor, 1997), of which the well-known Dirichlet process (Ferguson, 1973) is a special case. The stick-breaking construction (Pitman, 1996) and an independent slice-efficient sampler (Kalli et al., 2011) are employed to facilitate this. Following Frühwirth-Schnatter and Malsiner-Walli (2019), the hyperpriors assumed on the PYP parameters for the IMIFA and IMFA models are matched to the corresponding prior on the mixing proportions in the overfitted setting to yield 'sparse' infinite mixtures which inhibit overestimation of the number of clusters.

Allowing infinitely many factors within each cluster for the infinite factor models in the IMIFA family is achieved by assuming multiplicative gamma process (MGP) shrinkage priors (Bhattacharya and Dunson, 2011; Durante, 2017) on the cluster-specific factor loadings matrices. Such a prior allows the degree of shrinkage of the factor loadings towards zero to increase as the subspace dimension tends towards infinity. In the mixture setting, the MGP prior reflects the belief that the degree of shrinkage is specific to each cluster. The modal values of the number of non-empty components and numbers of loadings columns with non-negligible entries are used to construct histogram approximations to the posterior distributions of G and q_g respectively, with the modal values used as the estimates of these quantities.

An MCMC sampling scheme is devised which, by virtue of the conditionally conjugate nature of the assumed priors, consists mostly of straightforward Gibbs updates. In particular, a computationally efficient adaptive Gibbs sampler algorithm is employed for dynamically truncating the loadings matrices for computational efficiency. The additional hyperpriors for the parameters of the PYP prior necessitate Metropolis-Hastings steps. Identifiability issues are addressed offline in order to yield interpretable posterior summaries.

The advantages of the IMIFA model over other models in the IMIFA family are demonstrated through application to well-known data on the composition of fatty acids in Italian olive oils (Forina and Tiscornia, 1982; Forina et al., 1983). In particular, the flexibility in allowing $q_g \neq q_{g'}$ is shown to lead to improved clustering performance. The IMIFA model is also fitted to spectral metabolomic data from an epilepsy study (Carmody and Brennan, 2010), for which $n \ll p$, and handwritten digit data from the United States Postal Service (Hastie et al., 2001), a setting under which fitting sub-models in the IMIFA family is computationally infeasible. Furthermore, a novel strategy for posterior predictive checking (Gelman et al., 2004) is introduced and the potential scale reduction factor of Brooks and Gelman (1998) is employed to demonstrate the IMIFA model's performance from the perspectives of assessing model fit and mixing, respectively. Comparisons are made throughout to other state-of-the-art clustering methods designed for high-dimensional data, including spectral clustering (Ng et al., 2001), mixtures of factor mixture analysers (Viroli, 2010), and finite mixtures of matrix-variate normal distributions (Viroli, 2011). Additional simulation studies are provided in Appendix 4.B.

Finally, Chapter 5 concludes the thesis with a discussion of issues common across the main chapters, in which general limitations are examined and potential further extensions beyond those identified in the individual chapters are suggested.

1.3 Software Contributions

Other contributions of this research are the distributed software packages for implementing the families of methods described in each chapter. With one exception[1], all analysis was performed using the statistical software platform R (R Core Team, 2019) and can be reproduced using the provided packages. Indeed, the `MoEClust` package (Murphy and Murphy, 2019), related to Chapter 2, the `MEDseq` package (Murphy et al., 2019), related to Chapter 3, and the `IMIFA` package (Murphy et al., 2019), related to Chapter 4, are all freely available from the Comprehensive R Archive Network (CRAN)[2]. With the exception of the spectral metabolomic data

[1] The DP-BP method considered as a comparator of the IMIFA model in Section 4.3.3 of Chapter 4 was fitted using MATLAB code provided by the first author of Chen et al. (2010).

[2] `https://cran.r-project.org/`

analysed in Chapter 4, all data sets used in this thesis are made available in the corresponding R packages also.

The final Appendix in each chapter reproduces the vignette document associated with the corresponding R package; `MoEClust` in Appendix 2.F, `MEDseq` in Appendix 3.E, and `IMIFA` in Appendix 4.F. Notably, the vignette in Appendix 3.E contains a summary of the results of a second application of the MEDseq model family to data on the yearly family life states from a retrospective survey carried out by the Swiss Household Panel in 2002 (Müller et al., 2007).

As of September 3rd 2019, the combined number of downloads of the three R packages exceeds 16,000. Of these, the country of origin of the download is reported by CRAN in approximately 15,000 cases. Figure 1.1 shows a map of the world indicating where this subset of downloads have come from, produced with the aid of the `cranlogs` (Csárdi, 2019) and `rworldmap` (South, 2011) R packages. The country with the highest number of downloads is the U.S.A. (6,480); the Republic of Ireland has 257.

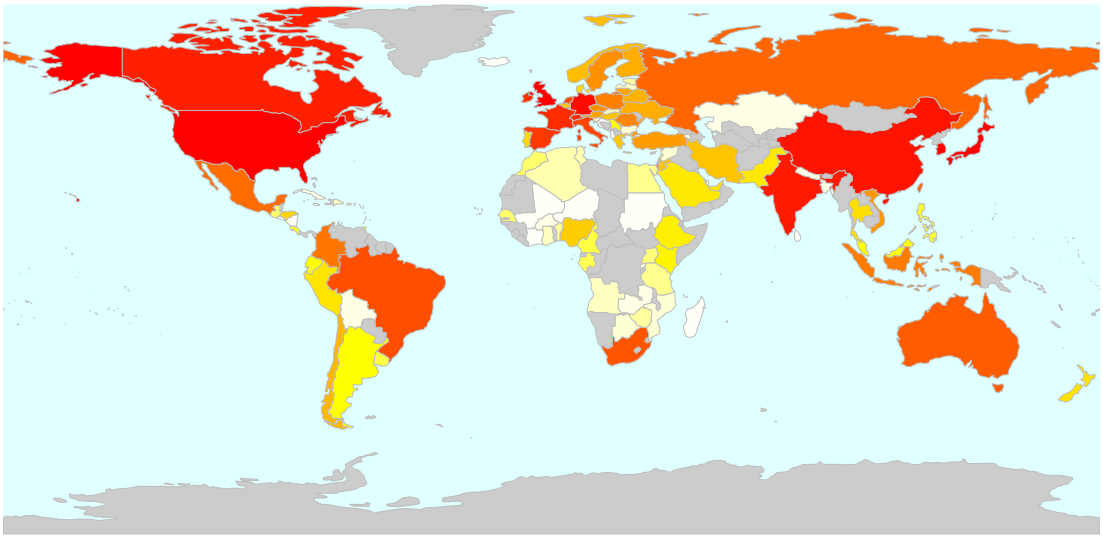


Figure 1.1: Heat map of the combined number of downloads from CRAN to date of the R packages `MoEClust`, `MEDseq`, and `IMIFA` on a country-by-country basis as of September 3rd 2019. Darker colours correspond to greater numbers of downloads and *vice versa*, while countries in grey have not recorded any downloads of these packages from CRAN. In total, 118 distinct countries are represented.

References

Abbott, A. and J. Forrest (1986). Optimal matching methods for historical sequences. *Journal of Interdisciplinary History 16*(3), 471–494. 7

Abbott, A. and A. Hrycak (1990). Measuring resemblance in sequence data: an optimal matching analysis of musician's careers. *American Journal of Sociology 96*(1), 145–185. 7

Agresti, A. (2002). *Categorical Data Analysis*. New York: John Wiley & Sons. 2

Banfield, J. and A. E. Raftery (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics 49*(3), 803–821. 2, 5

Bhattacharya, A. and D. B. Dunson (2011). Sparse Bayesian infinite factor models. *Biometrika 98*(2), 291–306. 9

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research 3*, 993–1022. 2

Bouveyron, C., G. Celeux, T. B. Murphy, and A. E. Raftery (2019). *Model-Based Clustering and Classification for Data Science: With Applications in R*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press. 1

Brooks, S. P. and A. Gelman (1998). Generative methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics 7*(4), 434–455. 10

Carmody, S. and L. Brennan (2010). Effects of pentylenetetrazole-induced seizures on metabolomic profiles of rat brain. *Neurochemistry International 56*(2), 340–344. 10

Celeux, G. and G. Govaert (1995). Gaussian parsimonious clustering models. *Pattern Recognition 28*(5), 781–793. 3, 5

Chang, W. C. (1983). On using principal components before separating a mixture of two multivariate normal distributions. *Applied Statistics 32*(3), 267–375. 8

Chen, M., J. Silva, J. Paisley, C. Wang, D. B. Dunson, and L. Carin (2010). Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: algorithm and performance bounds. *IEEE Transactions on Signal Processing 58*(12), 6140–6155. 10

Cook, R. D. and S. Weisberg (1994). *An Introduction to Regression Graphics*. New York: John Wiley & Sons. 5

Csárdi, G. (2019). `cranlogs`*: download logs from the 'RStudio' 'CRAN' mirror*. R package version 2.1.1. 11

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 39*(1), 1–38. 2

Durante, D. (2017). A note on the multiplicative gamma process. *Statistics & Probability Letters 122*, 198–204. 9

Everitt, B. S., S. Landau, M. Leese, and D. Stahl (2011). *Cluster Analysis* (Fifth ed.). Wiley Series in Probability and Statistics. New York: John Wiley & Sons. 1

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics 1*(2), 209–230. 9

Forina, M., C. Armanino, S. Lanteri, and E. Tiscornia (1983). Classification of olive oils from their fatty acid composition. In H. Martens and H. Russrum, Jr. (Eds.), *Food Research and Data Analysis*, pp. 189–214. Applied Science Publishers, London. 10

Forina, M. and E. Tiscornia (1982). Pattern recognition methods in the prediction of Italian olive oil by their fatty acid content. *Annali di Chimica 72*, 143–155. 10

Fraley, C. and A. E. Raftery (2002). Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association 97*(458), 611–631. 2

Fraley, C. and A. E. Raftery (2007). Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of Classification 24*(2), 155–181. 8

Frühwirth-Schnatter, S. and G. Malsiner-Walli (2019). From here to infinity: sparse finite versus Dirichlet process mixtures in model-based clustering. *Advances in Data Analysis and Classification 13*(1), 33–63. 9

Gelfand, A. E. and A. F. Smith (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association 85*(410), 398–409. 2

Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin (2004). *Bayesian Data Analysis* (Third ed.). Chapman and Hall/CRC Press. 10

Ghahramani, Z. and G. E. Hinton (1996). The EM algorithm for mixtures of factor analyzers. Technical report, Deparment of Computer Science, University of Toronto. 8

Gormley, I. C. and S. Frühwirth-Schnatter (2019). Mixtures of experts models. In S. Frühwirth-Schnatter, G. Celeux, and C. P. Robert (Eds.), *Handbook of Mixture Analysis*, Chapter 12, pp. 279–316. London: Chapman and Hall/CRC Press. 4

Gormley, I. C. and T. B. Murphy (2011). Mixture of experts modelling with social science applications. In K. Mengersen, C. Robert, and D. M. Titterington (Eds.), *Mixtures: Estimation and Applications*, Chapter 9, pp. 101–121. New York: John Wiley & Sons. 5

Govaert, G. and M. Nadif (2013). *Co-Clustering: models, algorithms and applications*. ISTE-Wiley. 8

Hamming, R. W. (1950). Error detecting and error correcting codes. *The Bell System Technical Journal 29*(2), 147–160. 6

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning* (Second ed.). Springer Series in Statistics. New York: Springer. 10

Hennig, C. (2010). Methods for merging Gaussian mixture components. *Advances in Data Analysis and Classification 4*(1), 3–34. 2

Hennig, C. (2015). What are the true clusters? *Pattern Recognition Letters 64*, 53–62. 1

Hurn, M., A. Justel, and C. P. Robert (2003). Estimating mixtures of regressions. *Journal of Computational and Graphical Statistics 12*(1), 55–79. 5

Jacobs, R. A., M. I. Jordan, S. J. Nowlan, and G. E. Hinton (1991). Adaptive mixtures of local experts. *Neural Computation 3*(1), 79–87. 4

Kalli, M., J. E. Griffin, and S. G. Walker (2011). Slice sampling mixture models. *Statistics and Computing 21*(1), 93–105. 9

Kaufman, L. and P. J. Rousseeuw (1990). Partitioning around medoids (program PAM). In *Finding Groups in Data: An Introduction to Cluster Analysis*, pp. 68–125. New York: John Wiley & Sons. 1

Knott, M. and D. J. Bartholomew (1999). *Latent Variable Models and Factor Analysis* (Second ed.). Number 7 in Kendall's library of statistics. London: Edward Arnold. 8

Lazarsfeld, P. F. and N. W. Henry (1968). *Latent Structure Analysis*. Boston: Houghton Mifflin. 2

Lesnard, L. (2010). Setting cost in optimal matching to uncover contemporaneous socio-temporal patterns. *Sociological Methods & Research 38*(3), 389–419. 7

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman (Eds.), *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Volume 1, pp. 281–297. University of California Press. 3

Mallows, C. L. (1957). Non-null ranking models. *Biometrika 44*(1/2), 114–130. 6

McLachlan, G. J. and D. Peel (2000). *Finite Mixture Models*. Wiley Series in Probability and Statistics. New York: John Wiley & Sons. 1

McLachlan, G. J., D. Peel, and R. W. Bean (2003). Modelling high-dimensional data by mixtures of factor analyzers. *Computatioal Statistics and Data Analysis 41*, 379–388. 8

McNicholas, P. D. (2016). Model-based clustering. *Journal of Classification 33*(3), 331–373. 2

McNicholas, P. D. and T. B. Murphy (2008). Parsimonious Gaussian mixture models. *Statistics and Computing 18*(3), 285–296. 3

McVicar, D. (2000). Status 0 four years on: young people and social exclusion in Northern Ireland. *Labour Market Bulletin 14*, 114–119. 6

McVicar, D. and M. Anyadike-Danes (2002). Predicting successful and unsuccessful transitions from school to work by using sequence methods. *Journal of the Royal Statistical Society: Series A (Statistics in Society) 165*(2), 317–334. 6, 7

Müller, N. S., M. Studer, and G. Ritschard (2007). Classification de parcours de vie à l'aide de l'optimal matching. In *XIVe Rencontre de la Société francophone de classification (SFC 2007), Paris, 5–7 septembre 2007*, pp. 157–160. 11

Murphy, K. and T. B. Murphy (2019). `MoEClust`*: Gaussian parsimonious clustering models with covariates and a noise component*. R package version 1.3.0. 10

Murphy, K., T. B. Murphy, R. Piccarreta, and I. C. Gormley (2019). `MEDseq`*: mixtures of exponential-distance models with covariates*. R package version 1.1.0. 10

Murphy, K., C. Viroli, and I. C. Gormley (2019). `IMIFA`*: infinite mixtures of infinite factor analysers and related models*. R package version 2.1.2. 10

Murphy, T. B. and D. Martin (2003). Mixtures of distance-based models for ranking data. *Computational Statistics and Data Analysis 41*(3–4), 645–655. 6

Ng, A. Y., M. I. Jordan, and Y. Weiss (2001). On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems*, Cambridge, MA, USA, pp. 849–856. MIT Press. 10

Papastamoulis, P. (2018). Overfitting Bayesian mixtures of factor analyzers with an unknown number of components. *Computational Statistics & Data Analysis 124*, 220–234. 9

Perman, M., J. Pitman, and M. Yor (1992). Size-biased sampling of Poisson point processes and excursions. *Probability Theory and Related Fields 92*(1), 21–39. 9

Pitman, J. (1996). Random discrete distributions invariant under size-biased permutation. *Advances in Applied Probability 28*(2), 525–539. 9

Pitman, J. and M. Yor (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability 25*(2), 855–900. 9

R Core Team (2019). *R: a language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. 10

Rousseau, J. and K. Mengersen (2011). Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(5), 689–710. 9

Scrucca, L., M. Fop, T. B. Murphy, and A. E. Raftery (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal 8*(1), 289–317. 4

South, A. (2011). rworldmap: a new R package for mapping global data. *The R Journal 3*(1), 35–43. 11

Tiedeman, D. V. (1955). On the study of types. In S. B. Sells (Ed.), *Symposium on Pattern Analysis*. Randolph Field, Texas: Air University, U.S.A.F. School of Aviation Medicine. 2

Viroli, C. (2010). Dimensionally reduced model-based clustering through mixtures of factor mixture analyzers. *Journal of classification 27*(3), 363–388. 10

Viroli, C. (2011). Finite mixtures of matrix normal distributions for classifying three-way data. *Statistics and Computing 21*(4), 511–522. 10

Ward, J. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association 58*(301), 236–244. 7

Wolfe, J. H. (1965). A computer program for the maximum likelihood analysis of types. Technical report, Technical Bulletin 65–15, U.S. Naval Personnel Research Activity. 2

Zellner, A. (1962). An efficient method of estimating seemingly unrelated regression equations and tests for aggregation bias. *Journal of the American Statistical Association 57*(298), 348–368. 5

Zhou, H., W. Pan, and X. Shen (2009). Penalized model-based clustering with unconstrained covariance matrices. *Electronic Journal of Statistics 3*, 1473–1496. 8

Chapter 2

Gaussian Parsimonious Clustering Models with Covariates and a Noise Component

Abstract

We consider model-based clustering methods for continuous, correlated data that account for external information available in the presence of mixed-type fixed covariates by proposing the MoEClust suite of models. These models allow different subsets of covariates to influence the component weights and/or component densities by modelling the parameters of the mixture as functions of the covariates. A familiar range of constrained eigen-decomposition parameterisations of the component covariance matrices are also accommodated. This paper thus addresses the equivalent aims of including covariates in Gaussian parsimonious clustering models and incorporating parsimonious covariance structures into all special cases of the Gaussian mixture of experts framework. The MoEClust models demonstrate significant improvement from both perspectives in applications to both univariate and multivariate data sets. Novel extensions to include a uniform noise component for capturing outliers and to address initialisation of the EM algorithm, model selection, and the visualisation of results are also proposed.

Keywords: Model-based clustering, mixtures of experts, EM algorithm, parsimony, multivariate response, covariates, noise component.

2.1 Introduction

In many analyses using the standard mixture model framework, a clustering method is typically implemented on the outcome variables only. Reference is not made to the associated covariates until the structure of the produced clustering is investigated in light of the information present in the covariates. Therefore, interpretations of the values of the model parameters within each component are guided by covariates that are not actually used in the construction of the clusters. It is desirable to have covariates incorporated into the clustering process and not only into the interpretation of the clustering structure and model parameters, thereby making them endogenous rather than exogenous to the clustering model. This both informs the construction of the clusters and provides richer insight into the type of observation which characterises each cluster.

When each observation consists of a response variable $\mathbf{y}_i$ on which the clustering is based and covariates $\mathbf{x}_i$ there are, broadly speaking, two main approaches in the literature to having covariates guide construction of the clusters, neatly summarised by Lamont et al. (2016) and compared in Ingrassia et al. (2012). Letting $\mathbf{z}_i$ denote the latent cluster membership indicator vector, where $z_{ig} = 1$ if observation i belongs to cluster g and $z_{ig} = 0$ otherwise, the first approach assumes that $\mathbf{z}_i$ affects the distribution of $\mathbf{x}_i$. In probabilistic terms, this means to replace the actual group-specific conditional distribution $f\left(\mathbf{y}_i \,|\, \mathbf{x}_i, z_{ig} = 1\right)\Pr\left(z_{ig} = 1\right)$ with $f\left(\mathbf{y}_i|\mathbf{x}_i, z_{ig} = 1\right)f\left(\mathbf{x}_i \,|\, z_{ig} = 1\right)\Pr\left(z_{ig} = 1\right)$. The name 'cluster-weighted model' (CWM) is frequently given to this approach, e.g. Dang et al. (2017) and Ingrassia et al. (2015); the latter provides a recent extension allowing for mixed-type covariates, with a further generalisation presented in Punzo and Ingrassia (2016). Noting the use of the alternative term 'mixtures of regressions with *random* covariates' to describe CWMs (e.g. Hennig 2000) provides opportunity to clarify that the remainder of this paper focuses on the second approach, with *fixed* potentially mixed-type covariates affecting cluster membership via $f\left(\mathbf{y}_i \,|\, \mathbf{x}_i, z_{ig} = 1\right)\Pr\left(z_{ig} = 1 \,|\, \mathbf{x}_i\right)$.

This is achieved using the mixture of experts (MoE) paradigm (Dayton and Macready, 1988; Jacobs et al., 1991) in which the parameters of the mixture are modelled as functions of fixed, potentially mixed-type covariates. We present, for finite mixtures of multivariate, continuous, correlated responses, a unifying frame-

work combining all of the special cases of the Gaussian MoE model with the flexibility afforded by the covariance constraints in the Gaussian parsimonious clustering model (GPCM) family (Banfield and Raftery, 1993; Celeux and Govaert, 1995). This has, to date, been lacking for all but the mixture of regressions and the mixture of regressions with concomitant variables where the same covariates enter both parts of the model (Dang and McNicholas, 2015).

Parsimony is obtained in GPCMs by imposing a range of constraints on the elements of an eigen-decomposition of the component covariance matrices. For MoE models, reducing the number of covariance parameters in this manner can help offset the number of regression parameters introduced by covariates, which is particularly advantageous when model selection is conducted using information criteria with penalty terms involving parameter counts. The main contribution of this paper is the development of a framework combining GPCM constraints with all of the special cases of the Gaussian MoE framework whereby different subsets of covariates can enter either, neither, or both the component densities and component weights. We also consider the special cases of the MoE framework for univariate response data with equal and unequal variance across components. Thus, this paper addresses the aim of incorporating potentially mixed-type covariates into the GPCM family and the equivalent aim of bringing GPCM covariance constraints into the Gaussian MoE framework, by proposing the MoEClust model family. The name MoEClust comes from the interest in employing MoE models chiefly for clustering purposes. From both perspectives, MoEClust models show significant improvement in applications to both univariate and multivariate response data.

Other novel contributions include the addition of a noise component for capturing outlying observations, and proposed solutions to initialising the EM algorithm in the presence of covariates sensibly, addressing the issue of model selection given the potentially large model space, and a means for visualising the results of MoEClust models. We also expand the number of special cases in the MoE framework from four to six, by considering more parsimonious counterparts to the standard mixture model and the mixture of regressions by constraining the mixing proportions. In addition, a software implementation for the full suite of MoEClust models is provided by the associated R package `MoEClust` (Murphy and Murphy, 2019), which is available from `www.r-project.org` (R Core Team, 2019),

with which all results were obtained. The syntax of the popular `mclust` package (Scrucca et al., 2016) is closely mimicked, with formula interfaces for specifying covariates in the gating and/or expert networks.

The structure of the paper is as follows. For both Gaussian mixtures of experts and MoEClust models, the modelling frameworks and inferential procedures are described, respectively, in Section 2.2 and Section 2.3. Section 2.3.3 describes the addition of a noise component for capturing outliers. Section 2.4 discusses proposals for addressing some practical issues affecting performance, namely the initialisation of the EM algorithm used to fit the models (Section 2.4.1), and issues around model selection (Section 2.4.2). The performance of the proposed models is illustrated in Section 2.5 with applications to univariate response CO_2 emissions data (Section 2.5.1) and multivariate response data from the Australian Institute of Sports (Section 2.5.2). Finally, the paper concludes with a brief discussion in Section 2.6, with some additional results deferred to the Appendices.

2.2 Modelling

This section builds up the MoEClust models by first describing the mixture of experts (MoE) modelling framework in Section 2.2.1 — elaborating on the special cases of the MoE model in Section 2.2.1.1 — and then extending to the family of MoEClust models comprising Gaussian mixture of experts models with parsimonious covariance structures from the GPCM family in Sections 2.2.2 and 2.2.3. Finally, a brief review of existing models and software is given in Section 2.2.4.

2.2.1 Mixtures of Experts

The mixture of experts model (Dayton and Macready, 1988; Jacobs et al., 1991) extends the mixture model used to cluster response data $\mathbf{y}_i$ by allowing the parameters of the model for observation i to depend on covariates $\mathbf{x}_i$. An independent sample of response/outcome variables of dimension p, denoted by $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$, is modelled by a G-component finite mixture model where the model parameters depend on the associated covariate inputs $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ of dimension d. The MoE model is often referred to as a conditional mixture model (Bishop,

2006) because, given the set of covariates $\mathbf{x}_i$, the distribution of the response variable $\mathbf{y}_i$ is a finite mixture model:

$$f\left(\mathbf{y}_i \mid \mathbf{x}_i\right) = \sum_{g=1}^{G} \tau_g(\mathbf{x}_i)\, f\left(\mathbf{y}_i \mid \boldsymbol{\theta}_g\left(\mathbf{x}_i\right)\right).$$

Each component is modelled by a probability density function $f\left(\mathbf{y}_i \mid \boldsymbol{\theta}_g\left(\mathbf{x}_i\right)\right)$ with component-specific parameters $\boldsymbol{\theta}_g\left(\mathbf{x}_i\right)$ and mixing proportions $\tau_g(\mathbf{x}_i)$; the latter are only allowed to depend on covariates when $G \geq 2$. As usual, $\tau_g(\mathbf{x}_i) > 0$ and $\sum_{g=1}^{G} \tau_g(\mathbf{x}_i) = 1$.

The MoE framework facilitates flexible modelling. While the response variable $\mathbf{y}_i$ is modelled via a finite mixture, model parameters are modelled as functions of related covariates $\mathbf{x}_i$ from the context under study. Both the mixing proportions and the parameters of component densities can depend on covariates. The terminology used to describe MoE models in the machine learning literature often refers to the component densities $f\left(\mathbf{y}_i \mid \boldsymbol{\theta}_g\left(\mathbf{x}_i\right)\right)$ as 'experts' or the 'expert network', and to the mixing proportions $\tau_g(\mathbf{x}_i)$ as 'gates' or the 'gating network', hence the nomenclature *mixture of experts*. Given that covariates can be continuous and/or categorical with multiple levels, we let $d + 1$ denote the number of columns in the corresponding design matrices, accounting also for the intercept term, in contrast to the number of covariates r, with $d \geq r$.

In the original formulation of the MoE model for continuous data (Jacobs et al., 1991), the mixing proportions (gating network) are modelled using multinomial logistic regression (MLR), though this need not strictly be the case; Geweke and Keane (2007) impose a multinomial probit structure here instead. The mixture components (expert networks) are generalised linear models (GLM; McCullagh and Nelder, 1983). Thus,

$$\widehat{\tau}_g(\mathbf{x}_i) = \frac{\exp\left(\widetilde{\mathbf{x}}_i \widehat{\boldsymbol{\beta}}_g\right)}{\sum_{h=1}^{G} \exp\left(\widetilde{\mathbf{x}}_i \widehat{\boldsymbol{\beta}}_h\right)}, \tag{2.1}$$

and

$$\widehat{\boldsymbol{\theta}}_g(\mathbf{x}_i) = \left\{\psi\left(\widetilde{\mathbf{x}}_i \widehat{\boldsymbol{\gamma}}_g\right), \widehat{\boldsymbol{\Sigma}}_g\right\}, \tag{2.2}$$

for some link function $\psi(\cdot)$, with a collection of parameters in the component densities (comprising a $(d+1) \times p$ matrix of expert network regression parameters $\widehat{\gamma}_g$ and the $p \times p$ component covariance matrix $\widehat{\boldsymbol{\Sigma}}_g$), a $(d+1)$-dimensional vector of regression parameters $\widehat{\boldsymbol{\beta}}_g$ in the gates in (2.1), and $\widetilde{\mathbf{x}}_i = (1, \mathbf{x}_i)$. Note that expert network covariates influence only the component means, and not the component covariance matrices. Henceforth, we restrict our attention to continuous outcome variables as per the GPCM family. Therefore, component densities are assumed to be the p-variate Gaussian $\phi(\mathbf{y}_i \,|\, \cdot)$, and the link function $\psi(\cdot)$ in (2.2) is simply the identity, such that covariates are linearly related to the response variables, i.e.

$$f\left(\mathbf{y}_i \,|\, \mathbf{x}_i\right) = \sum_{g=1}^{G} \tau_g(\mathbf{x}_i)\, \phi\big(\mathbf{y}_i \,|\, \boldsymbol{\theta}_g\left(\mathbf{x}_i\right) = \left\{\widetilde{\mathbf{x}}_i \gamma_g, \boldsymbol{\Sigma}_g\right\}\big). \tag{2.3}$$

2.2.1.1 The MoE Family of Models

It is possible that some, none, or all model parameters depend on the covariates. This leads to the four special cases of the Gaussian MoE framework shown in Figure 2.1, with the following interpretations, due to Gormley and Murphy (2011):

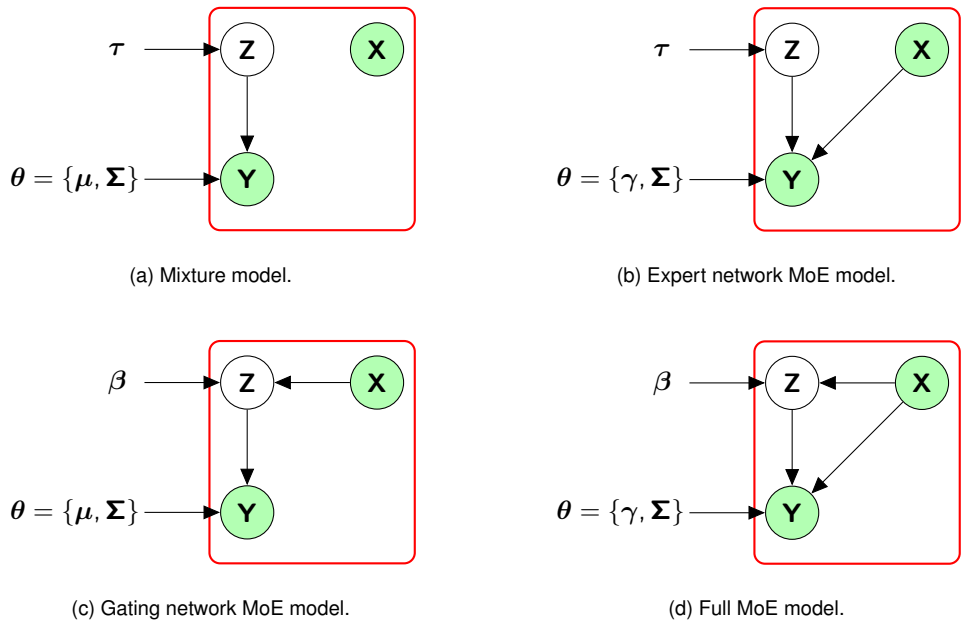
(a) in the *mixture model* the distribution of $\mathbf{y}_i$ depends on the latent cluster membership variable $\mathbf{z}_i$, the distribution of $\mathbf{z}_i$ is independent of the covariates $\mathbf{x}_i$, and $\mathbf{y}_i$ is independent of $\mathbf{x}_i$ conditional on $\mathbf{z}_i$: $f\left(\mathbf{y}_i\right) = \sum_{g=1}^{G} \tau_g \phi\big(\mathbf{y}_i \,|\, \boldsymbol{\theta}_g = \left\{\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\right\}\big)$.

(b) in the *expert network MoE model* the distribution of $\mathbf{y}_i$ depends on the covariates $\mathbf{x}_i$ and the latent cluster membership variable $\mathbf{z}_i$, and the distribution of $\mathbf{z}_i$ is independent of $\mathbf{x}_i$: $f\left(\mathbf{y}_i \,|\, \mathbf{x}_i\right) = \sum_{g=1}^{G} \tau_g \phi\big(\mathbf{y}_i \,|\, \boldsymbol{\theta}_g\left(\mathbf{x}_i\right) = \left\{\widetilde{\mathbf{x}}_i \gamma_g, \boldsymbol{\Sigma}_g\right\}\big)$.

(c) in the *gating network MoE model* the distribution of $\mathbf{y}_i$ depends on the latent cluster membership variable $\mathbf{z}_i$, $\mathbf{z}_i$ depends on the covariates $\mathbf{x}_i$, and $\mathbf{y}_i$ is independent of $\mathbf{x}_i$ conditional on $\mathbf{z}_i$: $f\left(\mathbf{y}_i \,|\, \mathbf{x}_i\right) = \sum_{g=1}^{G} \tau_g\left(\mathbf{x}_i\right) \phi\big(\mathbf{y}_i \,|\, \boldsymbol{\theta}_g = \left\{\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\right\}\big)$.

(d) in the *full MoE model*, given by (2.3), the distribution of $\mathbf{y}_i$ depends on both the covariates $\mathbf{x}_i$ and on the latent cluster membership variable $\mathbf{z}_i$, and the distribution of the latent variable $\mathbf{z}_i$ depends in turn on the covariates $\mathbf{x}_i$.

For models (c) and (d), $\mathbf{z}_i$ has a multinomial distribution with a single trial and probabilities equal to $\tau_g(\mathbf{x}_i)$. The full MoE model thus has the following latent variable representation: $\left(\mathbf{y}_i \,|\, \mathbf{x}_i, z_{ig} = 1\right) \sim \phi\left(\mathbf{y}_i \,|\, \boldsymbol{\theta}_g\left(\mathbf{x}_i\right) = \left\{\widetilde{\mathbf{x}}_i\boldsymbol{\gamma}_g, \boldsymbol{\Sigma}_g\right\}\right)$, $\Pr\left(z_{ig} = 1 \,|\, \mathbf{x}_i\right) = \tau_g(\mathbf{x}_i)$.



(a) Mixture model.

(b) Expert network MoE model.

(c) Gating network MoE model.

(d) Full MoE model.

Figure 2.1: The graphical model representation of the mixture of experts models. The differences between the special cases are due to the presence or absence of edges between the covariates **X** and the latent variables **Z** and/or response variables **Y**. Note that different subsets of the covariates in **X** can enter these two different parts of the full MoE model in (d).

The MoE family can be expanded further, from four to six special cases, by considering the models in (a) and (b), under which covariates do not enter the gating network, by constraining the mixing proportions to be equal across components, i.e. $\tau_g = 1/G \;\forall\; g$. This leads, respectively, to the *equal mixing proportion mixture model* and *equal mixing proportion expert network MoE model*. Such models are more parsimonious than their counterparts with unconstrained $\boldsymbol{\tau}$, as they require estimation of $G - 1$ fewer parameters. Note that the size of a cluster is proportional to τ_g, which is distinct from its volume (Celeux and Govaert, 1995). Thus, situations where $\tau_{ig} = \tau_g(\mathbf{x}_i)$, $\tau_{ig} = \tau_g$, or $\tau_{ig} = 1/G$ can all be accommodated. The six special cases of this MoE framework can be applied to both univariate and multivariate response data.

It is worth noting that CWMs most fundamentally differ from MoE models in their handling of the mixing proportions τ_g and in how the joint density $f(\mathbf{x}_i, z_{ig} = 1)$ is treated, either as $\Pr(z_{ig} = 1 \mid \mathbf{x}_i) = \tau_g(\mathbf{x}_i)$ (MoE) or $f(\mathbf{x}_i \mid z_{ig} = 1) \Pr(z_{ig} = 1)$ (CWM). In other words, the direction of the edge between $\mathbf{X}$ and $\mathbf{Z}$ in the full MoE model in Figure 2.1d is reversed under CWMs (Ingrassia et al., 2012). By virtue of modelling the distribution of the covariates, CWMs are also inherently less parsimonious. The same covariate(s) can enter both parts of full MoE models, in principle. Such models can provide a useful estimation of the conditional density of the outcome given the covariates, but the interpretation of the clustering model and the effect of the covariates becomes more difficult in this case. Conversely, allowing different covariates enter different parts of the model further differentiates MoE models from CWMs. It is common to distinguish among the overall set of covariates between *concomitant* gating network variables and *explanatory* expert network variables. Thus, for clarity, $\mathbf{x}_i^{(G)}$ and $\mathbf{x}_i^{(E)}$ will henceforth refer, respectively, to the possibly overlapping subsets of gating and expert network covariates, such that $\mathbf{x}_i = \{\mathbf{x}_i^{(G)} \cup \mathbf{x}_i^{(E)}\}$, with the dimensions of the associated design matrices given by $d_G + 1$ and $d_E + 1$. Higher order terms, transformations, and interaction effects between covariates are also allowed in both networks.

2.2.2 Gaussian Parsimonious Clustering Models

Parsimony has been considered extensively in the model-based clustering literature. In particular, the volume of work on Gaussian and/or parsimonious mixtures has increased hugely since the work of Banfield and Raftery (1993) and Celeux and Govaert (1995). These works introduced the family of GPCMs, which are implemented in the popular R package `mclust` (Scrucca et al., 2016). The influence of GPCMs is clear on many other works which obtain parsimony in the component covariance matrices; e.g., using constrained factor-analytic structures (McNicholas and Murphy, 2008), the multivariate t-distribution and associated tEIGEN family (Andrews and McNicholas, 2012), and the multivariate contaminated normal distribution (Punzo and McNicholas, 2016).

Parsimonious covariance matrix parameterisations are obtained in GPCMs by means of imposing constraints on the components of an eigen-decomposition of

the form $\boldsymbol{\Sigma}_g = \lambda_g \mathbf{D}_g \mathbf{A}_g \mathbf{D}_g^\top$, where λ_g is a scalar controlling the volume, $\mathbf{A}_g$ is a diagonal matrix, with entries proportional to the eigenvalues of $\boldsymbol{\Sigma}_g$ with $\det(\mathbf{A}_g) = 1$, specifying the shape of the density contours, and $\mathbf{D}_g$ is $p \times p$ orthogonal matrix, the columns of which are the eigenvectors of $\boldsymbol{\Sigma}_g$, governing the corresponding ellipsoid's orientation. Imposing constraints reduces the number of free covariance parameters from $Gp(p+1)/2$ in the unconstrained (VVV) model. This is desirable when p is even moderately large. Thus, GPCMs allow for intermediate component covariance matrices lying between homoscedasticity and heteroscedasticity. Table 2.1 summarises the geometric characteristics of the GPCM constraints, which are then shown in Figure 2.2.

Table 2.1: Nomenclature, descriptions, and parameter counts of the parameterisations of the component covariance matrices $\boldsymbol{\Sigma}_g$ available under GPCMs, all of which are available when there is no dependency in any way on covariates. † indicates availability in the first four special cases of the Gaussian MoE framework shown in Figure 2.1 and the MoEClust family; • indicates other models available in the MoEClust family. While all models are possible when $G = 1$, they are all equivalent to one of the highlighted available models, otherwise missing entries correspond to models which are never available. The other central columns refer to $G > 1$ settings.

Name	Model	$G=1$	$n>p$	$n\leq p$	Distribution	Volume	Shape	Orientation	Covariance Parameters
E	σ	†	•		(univariate)	equal			1
V	σ_g		†		(univariate)	variable			G
EII	$\lambda\mathcal{I}$	†	•	•	spherical	equal	equal	—	1
VII	$\lambda_g\mathcal{I}$		•	•	spherical	variable	equal	—	G
EEI	$\lambda\mathbf{A}$	•	•	•	diagonal	equal	equal	axis-aligned	p
VEI	$\lambda_g\mathbf{A}$		•	•	diagonal	variable	equal	axis-aligned	$G+(p-1)$
EVI	$\lambda\mathbf{A}_g$		•	•	diagonal	equal	variable	axis-aligned	$1+G(p-1)$
VVI	$\lambda_g\mathbf{A}_g$		†	†	diagonal	variable	variable	axis-aligned	Gp
EEE	$\lambda\mathbf{D}\mathbf{A}\mathbf{D}^\top$	•	•		ellipsoidal	equal	equal	equal	$p(p+1)/2$
EVE	$\lambda\mathbf{D}\mathbf{A}_g\mathbf{D}^\top$		•		ellipsoidal	equal	variable	equal	$1+p(p-1)/2+G(p-1)$
VEE	$\lambda_g\mathbf{D}\mathbf{A}\mathbf{D}^\top$		•		ellipsoidal	variable	equal	equal	$G+p(p-1)/2+(p-1)$
EEV	$\lambda\mathbf{D}_g\mathbf{A}\mathbf{D}_g^\top$		•		ellipsoidal	equal	equal	variable	$1+Gp(p-1)/2+(p-1)$
VEV	$\lambda_g\mathbf{D}_g\mathbf{A}\mathbf{D}_g^\top$		•		ellipsoidal	variable	equal	variable	$G+Gp(p-1)/2+(p-1)$
EVV	$\lambda\mathbf{D}_g\mathbf{A}_g\mathbf{D}_g^\top$		•		ellipsoidal	equal	variable	variable	$1+Gp(p-1)/2+(p-1)$
VVE	$\lambda_g\mathbf{D}\mathbf{A}_g\mathbf{D}^\top$		•		ellipsoidal	variable	variable	equal	$G+p(p-1)/2+G(p-1)$
VVV	$\lambda_g\mathbf{D}_g\mathbf{A}_g\mathbf{D}_g^\top$		†		ellipsoidal	variable	variable	variable	$Gp(p+1)/2$

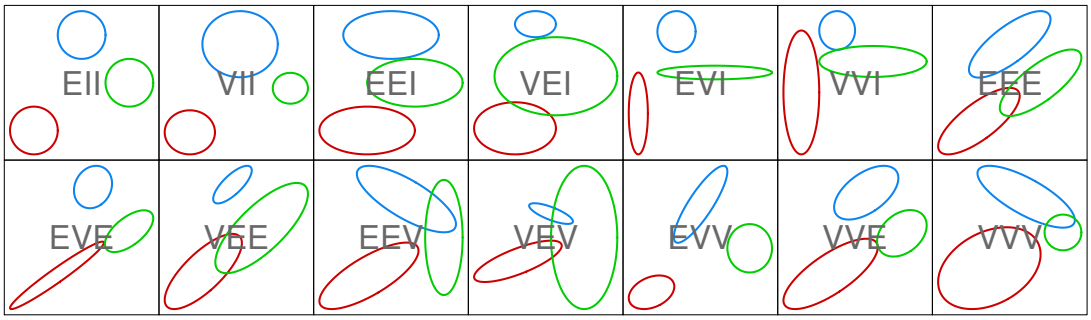


Figure 2.2: Ellipses of isodensity for each of the 14 parsimonious eigen-decomposition covariance parameterisations for multivariate data in GPCMs, with three components in two dimensions.

Note for models with names ending with I that the number of parameters is linear in the data dimension p. Thus, the diagonal models are especially parsimonious and useful in $n \leq p$ settings. While there are 2 variance parameterisations for mixtures of univariate response data, and 14 covariance parameterisations for mixtures of multivariate response data, considering the equal mixing proportion constraint doubles the number of models available in each of these cases.

2.2.3 The MoEClust Family of Models

Interest lies in bringing parsimonious covariance structures to Gaussian MoE models with network-specific subsets of covariates:

$$f\left(\mathbf{y}_i \,|\, \mathbf{x}_i\right) = \sum_{g=1}^{G} \tau_g\left(\mathbf{x}_i^{(G)}\right) \phi\Big(\mathbf{y}_i \;\Big|\; \boldsymbol{\theta}_g\left(\mathbf{x}_i^{(E)}\right) = \left\{\widetilde{\mathbf{x}}_i^{(E)}\boldsymbol{\gamma}_g, \boldsymbol{\Sigma}_g\right\}\Big),$$

where $\boldsymbol{\Sigma}_g$ can follow any of the GPCM constraints outlined in Table 2.1. It is equivalent to say that interest lies in incorporating covariate information into the GPCM model family. Using the covariance constraints, combined with the six special cases of the MoE model described in Section 2.2.1.1, yields the MoEClust family of models, which are capable of dealing with correlated responses and offering additional parsimony in the component densities compared to current implementations of Gaussian MoE models, by virtue of allowing the size, volume, shape, and/or

orientation to be equal or unequal across components. For MoE models, every continuous covariate added to the gating and expert networks introduces $G - 1$ and Gp additional regression parameters, respectively. Parsimonious MoEClust models allow the increase in the number of regression parameters to be offset by the reduction in the number of covariance parameters. This can be advantageous when model selection is conducted using information criteria which include penalty terms based on parameter counts (see Section 2.4.2).

2.2.4 Existing Models and Software

A number of tools for fitting MoE models are available in the R programming environment (R Core Team, 2019). These include `flexmix` (Grün and Leisch, 2007, 2008), `mixtools` (Benaglia et al., 2009), and others. Tools for fitting GPCMs without covariates include `mclust` (Scrucca et al., 2016) and `Rmixmod` (Lebret et al., 2015).

The `flexmix` package (Grün and Leisch, 2007, 2008) can accommodate the full range of MoE models outlined in Section 2.2.1.1, excluding those for which $\boldsymbol{\tau}$ is constrained to be equal, in the case of univariate $\mathbf{y}_i$, though only models with unequal variance can be fitted. The user can specify the form of the GLM and covariates (if any) to be used in the gating and expert networks, for which the package has a similar interface to the `glm` functions within R. In the case of a multivariate continuous response, there is functionality for multivariate Gaussian component distributions though only for models without expert network covariates. Furthermore, only the VVI and VVV constraints and models with unequal mixing proportions or gating concomitants are facilitated.

For univariate data, the `mixtools` package (Benaglia et al., 2009) can accommodate the expert network MoE model with equal or unequal variance; it can also accommodate the full MoE model, though only for $G = 2$, with unequal variance, and with the restriction that all covariates enter both part of the model. The package allows for nonparametric estimation of the functional form for the mixing proportions (gating networks) and the component densities (expert networks), so it offers further flexibility beyond `flexmix` in these cases. However, the multivariate models in `mixtools` use the local independence assumption, so it does not directly offer the

facility to model multivariate Gaussian component densities with non-diagonal covariance matrices. Furthermore, multivariate response models in `mixtools` do not yet incorporate covariates in any way, and the equal mixing proportions constraint is not facilitated in any way either.

The `mclust` package (Scrucca et al., 2016) and `Rmixmod` package (Lebret et al., 2015) can accommodate the full range of covariance constraints in Table 2.1, and are thus examples of existing software which can fit GPCMs, but only using the standard finite mixture model (model (a) in Figure 2.1) or the equal mixing proportions mixture model; i.e., they do not facilitate dependency on covariates in any way.

Another important contribution in this area is by Dang and McNicholas (2015). This work introduces eigen-decomposition parsimony to the MoE framework, though only for the expert network MoE model and the full MoE model. However, for the full MoE model, all covariates are assumed to enter into both parts of the model. Thus, the MoEclust model family completes the work of Dang and McNicholas (2015) by considering all six special cases of the MoE framework, whereby different subsets of covariates can enter either, neither, or both the component densities and/or component weights, as well as models with equal mixing proportions. In addition, our unifying MoEClust framework also incorporates such parsimonious models for univariate response data.

Finally, it should be noted that eigen-decomposition parsimony has been introduced to the alternative CWM framework, in which all covariates enter the same part of the model, by Dang et al. (2017), for the multivariate Gaussian distributions of both the response variables and the covariates, assuming only continuous covariates; see also Punzo and Ingrassia (2015) for eigen-decomposition parsimony applied to the covariates only. The `flexCWM` package (Mazza et al., 2018) allows GPCM covariance structures in the distribution of the continuous covariates only, though only univariate responses are accommodated. It also allows, simultaneously or otherwise, covariates of other types, as well as omitting the distribution for the covariates entirely, leading to non-parsimonious mixtures of regressions, with or without concomitant variables.

2.3 Model Fitting via EM

To estimate the parameters of MoEClust models, we focus on maximum likelihood estimation using the EM algorithm (Dempster et al., 1977). This is outlined first for MoE models in Section 2.3.1 and then extended to MoEClust models in Section 2.3.2. Model fitting details are described chiefly for the full MoE model only, for simplicity. A simple trick involving the residuals of the weighted linear regressions in the expert network assists fitting when using GPCM constraints. A uniform noise component to capture outlying non-Gaussian observations is added in Section 2.3.3. When gating concomitants are present, the noise component is treated in two different ways.

2.3.1 Fitting MoE Models

For the full mixture of experts model, the likelihood is of the form

$$\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Sigma} \,|\, \mathbf{Y}, \mathbf{X}) = \prod_{i=1}^{n} \sum_{g=1}^{G} \tau_g\left(\mathbf{x}_i^{(G)}\right) \phi\left(\mathbf{y}_i \,|\, \boldsymbol{\theta}_g\left(\mathbf{x}_i^{(E)}\right)\right),$$

where $\tau_g\left(\mathbf{x}_i^{(G)}\right)$ and $\boldsymbol{\theta}_g\left(\mathbf{x}_i^{(E)}\right)$ are as defined by (2.1). The data are augmented by imputing the latent cluster membership indicator $\mathbf{z}_i = \left(z_{i1}, \ldots, z_{iG}\right)^\top$. Thus, the conditional distribution of $\left(\mathbf{y}_i, \mathbf{z}_i \,|\, \mathbf{x}_i\right)$ is of the form

$$f\left(\mathbf{y}_i, \mathbf{z}_i \,|\, \mathbf{x}_i\right) = \prod_{g=1}^{G} \left[\tau_g\left(\mathbf{x}_i^{(G)}\right) \phi\left(\mathbf{y}_i \,|\, \boldsymbol{\theta}_g\left(\mathbf{x}_i^{(E)}\right)\right)\right]^{z_{ig}}.$$

Hence, the complete data likelihood is of the form

$$\mathcal{L}_c(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Sigma} \,|\, \mathbf{Y}, \mathbf{X}, \mathbf{Z}) = \prod_{i=1}^{n} \prod_{g=1}^{G} \left[\tau_g\left(\mathbf{x}_i^{(G)}\right) \phi\left(\mathbf{y}_i \,|\, \boldsymbol{\theta}_g\left(\mathbf{x}_i^{(E)}\right)\right)\right]^{z_{ig}},$$

and the complete data log-likelihood has the form

$$\ell_c(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Sigma} \,|\, \mathbf{Y}, \mathbf{X}, \mathbf{Z}) = \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \left[\log \tau_g\left(\mathbf{x}_i^{(G)}\right) + \log \phi\left(\mathbf{y}_i \,|\, \boldsymbol{\theta}_g\left(\mathbf{x}_i^{(E)}\right)\right)\right]. \tag{2.4}$$

The iterative EM algorithm for MoE models follows in a similar manner to that for standard mixture models. It consists of an E-step (expectation) which replaces for each observation the missing data $\mathbf{z}_i$ with their expected values $\widehat{\mathbf{z}}_i$, followed by a M-step (maximisation) which maximises the expected complete data log-likelihood, computed with the estimates $\widehat{\mathbf{Z}} = (\widehat{\mathbf{z}}_1, \ldots, \widehat{\mathbf{z}}_n)$, to provide estimates of the component weight parameters $\widehat{\tau}_g\big(\mathbf{x}_i^{(G)}\big)$ and the component parameters $\widehat{\boldsymbol{\theta}}_g\big(\mathbf{x}_i^{(E)}\big)$. Aitken's acceleration criterion is used to assess convergence of the non-decreasing sequence of log-likelihood estimates (Böhning et al., 1994). Parameter estimates produced on convergence achieve at least a local maximum of the likelihood function. Upon convergence, cluster memberships are estimated via the maximum *a posteriori* (MAP) classification. The E-step involves computing

$$\begin{aligned}\widehat{z}_{ig}^{(t+1)} &= \mathbb{E}\Big(z_{ig} \;\Big|\; \mathbf{y}_i, \mathbf{x}_i, \widehat{\boldsymbol{\beta}}^{(t)}, \widehat{\boldsymbol{\gamma}}^{(t)}, \widehat{\boldsymbol{\Sigma}}^{(t)}\Big) \\ &= \frac{\widehat{\tau}_g^{(t)}\big(\mathbf{x}_i^{(G)}\big)\,\phi\big(\mathbf{y}_i \,|\, \widehat{\boldsymbol{\theta}}_g^{(t)}\big(\mathbf{x}_i^{(E)}\big)\big)}{\sum_{h=1}^{G} \widehat{\tau}_h^{(t)}\big(\mathbf{x}_i^{(G)}\big)\,\phi\big(\mathbf{y}_i \,|\, \widehat{\boldsymbol{\theta}}_h^{(t)}\big(\mathbf{x}_i^{(E)}\big)\big)},\end{aligned}$$

where $\big\{\widehat{\boldsymbol{\beta}}^{(t)}, \widehat{\boldsymbol{\gamma}}^{(t)}, \widehat{\boldsymbol{\Sigma}}^{(t)}\big\}$ are the estimates of the parameters in the gating and expert networks on the t-th iteration of the EM algorithm.

For the M-step, we notice that the complete data log-likelihood in (2.4) can be considered as a separation into the portion due to the gating network and the portion due to the expert network. Thus, the expected complete data log-likelihood (2.5) can be maximised separately under the EM framework:

$$\begin{aligned}\mathbb{E}\Big[\ell_c\big(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Sigma} \,|\, \mathbf{Y}, \mathbf{X}, \mathbf{Z}, \widehat{\boldsymbol{\beta}}^{(t)}, \widehat{\boldsymbol{\gamma}}^{(t)}, \widehat{\boldsymbol{\Sigma}}^{(t)}\big)\Big] &= \sum_{i=1}^{n}\sum_{g=1}^{G} \widehat{z}_{ig}^{(t+1)} \log \tau_g\big(\mathbf{x}_i^{(G)}\big) \\ &\quad + \sum_{i=1}^{n}\sum_{g=1}^{G} \widehat{z}_{ig}^{(t+1)} \log \phi\big(\mathbf{y}_i \,|\, \boldsymbol{\theta}_g\big(\mathbf{x}_i^{(E)}\big)\big).\end{aligned} \tag{2.5}$$

The first term is of the same form as a MLR model, here written with component 1 as the baseline reference level, for identifiability reasons:

$$\log \frac{\tau_g\big(\mathbf{x}_i^{(G)}\big)}{\tau_1\big(\mathbf{x}_i^{(G)}\big)} = \log \frac{\Pr\big(\widehat{z}_{ig}^{(t+1)} = 1\big)}{\Pr\big(\widehat{z}_{i1}^{(t+1)} = 1\big)} = \widetilde{\mathbf{x}}_i^{(G)} \boldsymbol{\beta}_g \;\forall\; g \geq 2, \text{ where } \boldsymbol{\beta}_1 = (0, \ldots, 0)^{\top}.$$

Thus, methods for fitting such models can be used to maximise this term and estimate the parameters in the gating network. The second term is of the same form as fitting G separate weighted multivariate linear regressions, and thus methods for fitting such models can be used to estimate the expert network parameters. Note that these are multivariate in the sense of a multivariate outcome $\mathbf{y}_i$; the associated design matrix having $d_E + 1$ columns means these regressions are possibly also multivariate in terms of the explanatory variables. Thus, fitting MoE models is straightforward in principle.

2.3.2 Fitting MoEClust Models

Maximising the second term in (2.5), corresponding to the expert network, gives rise to the following expression

$$\begin{aligned} -\frac{1}{2}\Bigg(p\log 2\pi + \sum_{i=1}^{n}\sum_{g=1}^{G}\widehat{z}_{ig}^{(t+1)}\log|\boldsymbol{\Sigma}_g| + \\ \sum_{i=1}^{n}\sum_{g=1}^{G}\widehat{z}_{ig}^{(t+1)}\big(\mathbf{y}_i - \widetilde{\mathbf{x}}_i^{(E)}\boldsymbol{\gamma}_g\big)^{\top}\boldsymbol{\Sigma}_g^{-1}\big(\mathbf{y}_i - \widetilde{\mathbf{x}}_i^{(E)}\boldsymbol{\gamma}_g\big)\Bigg). \end{aligned} \tag{2.6}$$

When the same set of regressors are used for each dependent variable, as is always the case for MoEClust models, or when $\boldsymbol{\Sigma}_g$ is diagonal, it can be shown that $\boldsymbol{\gamma}_g$ does not depend on $\boldsymbol{\Sigma}_g$, much like a Seemingly Unrelated Regression model (SUR; Zellner, 1962). We first estimate $\widehat{\boldsymbol{\gamma}}_g$ and then $\widehat{\boldsymbol{\Sigma}}_g$. Fitting G separate multivariate regressions (weighted by $\widehat{z}_{ig}$), yields G sets of $n \times p$ SUR residuals $\widehat{\mathbf{r}}_{ig} = \mathbf{y}_i - \widetilde{\mathbf{x}}_i^{(E)}\widehat{\boldsymbol{\gamma}}_g$ which, crucially, satisfy $\sum_{i=1}^{n}\widehat{z}_{ig}\widehat{\mathbf{r}}_{ig} = 0$. Thus, maximising (2.6) is equivalent to minimising

$$\sum_{i=1}^{n}\sum_{g=1}^{G}\widehat{z}_{ig}^{(t+1)}\log|\boldsymbol{\Sigma}_g| + \sum_{i=1}^{n}\sum_{g=1}^{G}\widehat{z}_{ig}^{(t+1)}\widehat{\mathbf{r}}_{ig}^{\top}\boldsymbol{\Sigma}_g^{-1}\widehat{\mathbf{r}}_{ig}\,, \tag{2.7}$$

which is of the same form as the criterion used in the M-step of a standard Gaussian finite mixture model with component covariance matrices $\widehat{\boldsymbol{\Sigma}}$, component means equal to zero, and new augmented data set $\widehat{\mathbf{R}}$. Thus, when estimating the component covariance matrices via (2.7), the same M-step function as used within

`mclust` can be applied to augmented data, constructed so that each observation is represented as follows:

1. Stack the G sets of SUR residuals into the $(n \times G) \times p$ matrix $\widehat{\mathbf{R}}$:

$$\widehat{\mathbf{R}} = \begin{bmatrix} \widehat{r}_{111} & \widehat{r}_{112} & \dots & \widehat{r}_{11p} \\ \widehat{r}_{211} & \widehat{r}_{212} & \dots & \widehat{r}_{21p} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{r}_{n11} & \widehat{r}_{n12} & \dots & \widehat{r}_{n1p} \\ \hline \widehat{r}_{121} & \widehat{r}_{122} & \dots & \widehat{r}_{12p} \\ \widehat{r}_{221} & \widehat{r}_{222} & \dots & \widehat{r}_{22p} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{r}_{n21} & \widehat{r}_{n22} & \dots & \widehat{r}_{n2p} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \widehat{r}_{1G1} & \widehat{r}_{1G2} & \dots & \widehat{r}_{1Gp} \\ \widehat{r}_{2G1} & \widehat{r}_{2G2} & \dots & \widehat{r}_{2Gp} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{r}_{nG1} & \widehat{r}_{nG2} & \dots & \widehat{r}_{nGp} \end{bmatrix}$$

2. Create the $(n \times G) \times G$ block-diagonal matrix $\widehat{\boldsymbol{\zeta}}$ from the columns of $\widehat{\mathbf{Z}}$:

$$\widehat{\boldsymbol{\zeta}} = \begin{bmatrix} \widehat{z}_{11} & 0 & \dots & 0 \\ \widehat{z}_{21} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{z}_{n1} & 0 & \dots & 0 \\ \hline 0 & \widehat{z}_{12} & \dots & 0 \\ 0 & \widehat{z}_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \widehat{z}_{n2} & \dots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \dots & \widehat{z}_{1G} \\ 0 & 0 & \dots & \widehat{z}_{2G} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \widehat{z}_{nG} \end{bmatrix}$$

Structuring the model in this manner allows GPCM covariance structures to be easily imposed on Gaussian MoE models with gating and/or expert network covariates. In the end, the M-step involves three sub-steps, each using the current estimate of $\widehat{\mathbf{Z}}$: i) estimating the gating network parameters $\widehat{\boldsymbol{\beta}}_g$ and hence the component weights $\widehat{\tau}_g\left(\mathbf{x}_i^{(G)}\right)$ via MLR, ii) estimating the expert network parameters $\widehat{\boldsymbol{\gamma}}_g$ and hence the component-specific means via weighted multivariate multiple linear regression, and iii) estimating the constrained component covariance matrices $\widehat{\boldsymbol{\Sigma}}_g$ using the augmented data set comprised of SUR residuals, as outlined above.

In the absence of covariates in the gating and/or expert networks, under the special cases outlined in Section 2.2.1.1, their respective contribution to (2.5) is maximised as per the corresponding term in a standard GPCM. In other words, the gating and expert networks without covariates can be seen as regressions with only an intercept term. Thus, the augmented data structure is not required when there are no expert covariates and the formula for estimating $\boldsymbol{\tau}$ in the absence of concomitant variables is $\widehat{\tau}_g = n^{-1}\sum_{i=1}^{n}\widehat{z}_{ig}$, rather than (2.1). As described in Section 2.2.1.1, it is sometimes useful to expand the model family further by considering

more parsimonious alternatives to the special cases of models (a) and (b) in Figure 2.1, where gating network concomitants are omitted, by constraining the mixing proportions to be equal and fixed, i.e. $\tau_g = 1/G \ \forall \ g$. Similarly, removing the corresponding regression intercept(s) from the part(s) of the model where covariates enter can yield further parsimony in appropriate settings, e.g. when there are strong *a priori* physical reasons for believing $\mathbb{E}\left(\mathbf{Y} \mid \mathbf{X}^{(E)} = \mathbf{0}\right) = \mathbf{0}$ (Eisenhauer, 2003).

2.3.3 Adding a Noise Component

For models with expert network covariates, and/or when the volume and/or shape differ across components, the mixture likelihood is unbounded. We restrict our interest only to solutions for which the log-likelihood at convergence is finite. As per the `eps` argument to the `mclust` R package's `emControl` function (Scrucca et al., 2016), we monitor the conditioning of the covariances and add a tolerance parameter (set to the relative machine precision, i.e. `2.220446e-16` on IEEE compliant machines) to the M-step estimation of the component covariances to control termination of the EM algorithm on the basis of small eigenvalues. For models with unconstrained $\boldsymbol{\Sigma}_g$, each cluster must contain at least $p+1$ units to avoid computational singularity. Thus, in practice, such spurious solutions with infinite likelihood occur especially for higher G values, whereby either solutions with empty components reduce to ones with fewer components, or uninteresting solutions with degenerate components containing too few units or even singletons are found. Sensible initial allocations (see Section 2.4.1) and/or the equal mixing proportion constraint, which help avoid empty or otherwise poorly populated clusters, can help to alleviate this problem. García-Escudero et al. (2018) offer an excellent discussion of the notions of spurious solutions and degenerate components.

Further extending MoEClust models via the inclusion of an additional uniform noise component can also help in addressing these issues, by capturing outlying observations which do not fit the prevailing pattern of Gaussian clusters and thus would otherwise be assigned to (possibly many) small clusters. In particular, the noise component for encompassing clusters with non-Gaussian distributions is here distributed as a homogeneous spatial Poisson process, as per Banfield and Raftery (1993). Such a noise component can be included regardless of where co-

variates (if any) enter, and regardless of the GPCM constraints employed, though this has hitherto only been considered for the standard mixture model with no covariates. Model-fitting via the EM algorithm is not greatly complicated by the addition of a noise component, though it is required to estimate V, the hypervolume of the region from which the response data have been drawn, or to consider V as an independent tuning parameter as per Hennig and Coretto (2008), especially if $n \leq p$. For univariate responses V is given by the range of $y_1, \ldots, y_n$. For multivariate data, V can be estimated by the hypervolume of the convex hull, ellipsoid hull, or smallest hyperrectangle enclosing the data. We focus on the latter method.

For initialisation, a column in which each entry is τ_0 (the guess of the prior probability that observations are noise) is appended to the starting $\mathbf{Z}$ matrix, with other columns corresponding to non-noise components then multiplied by $1 - \tau_0$. The initial τ_0 should not be too high; it is set to 0.1 here. For models with a noise component and no gating concomitants, the mixing proportions can be, as before, either constrained or unconstrained. In the latter case, we estimate τ_0 and then constrain the remaining proportions. We add the extension that concomitants, when present, are allowed to affect (2.8) or not affect (2.9) the mixing proportion of the noise component. Henceforth, for clarity, we refer to these settings as the gated noise (NG) and non-gated noise (NGN) models, respectively. The NGN model assumes τ_0 is constant across observations and covariate patterns. It is thus the more parsimonious model; it requires only 1 extra gating network parameter, rather than $d_G + 1$ under the GN model, relative to models without a noise component, though it is only defined for $G \geq 2$.

$$\text{GN}: \; f\left(\mathbf{y}_i \,|\, \mathbf{x}_i\right) = \sum_{g=1}^{G} \tau_g\left(\mathbf{x}_i^{(G)}\right) \phi\left(\mathbf{y}_i \;\middle|\; \boldsymbol{\theta}_g\left(\mathbf{x}_i^{(E)}\right) = \left\{\widetilde{\mathbf{x}}_i^{(E)} \boldsymbol{\gamma}_g, \boldsymbol{\Sigma}_g\right\}\right) + \frac{\tau_0\left(\mathbf{x}_i^{(G)}\right)}{V}. \tag{2.8}$$

$$\text{NGN}: \; f\left(\mathbf{y}_i \,|\, \mathbf{x}_i\right) = \sum_{g=1}^{G} \tau_g\left(\mathbf{x}_i^{(G)}\right) \phi\left(\mathbf{y}_i \;\middle|\; \boldsymbol{\theta}_g\left(\mathbf{x}_i^{(E)}\right) = \left\{\widetilde{\mathbf{x}}_i^{(E)} \boldsymbol{\gamma}_g, \boldsymbol{\Sigma}_g\right\}\right) + \frac{\tau_0}{V}. \tag{2.9}$$

2.4 Practical Issues

In this section, factors affecting the performance of MoEClust models are discussed; namely, the necessity of a good initial partition to prevent the EM algorithm

from converging to a suboptimal local maximum (Section 2.4.1), and the necessity of model selection with regard to where and what covariates (if any) enter the model to yield further parsimony by reducing the number of gating and/or expert network regression parameters (Section 2.4.2). Novel strategies for dealing with both issues are proposed.

2.4.1 EM Initialisation

With regards to initialisation of the EM algorithm for $G > 1$ MoEClust models, model-based agglomerative hierarchical clustering and quantile-based clustering have been found to be suitable for multivariate and univariate data, respectively. Both `flexmix` and `mixtools` randomly initialise the allocations, despite the obvious computational drawback of the need to run the EM algorithm from multiple random starting points. However, when explanatory variables $\mathbf{x}_i^{(E)}$ enter the expert network, it is useful to use them to augment the initialisation strategy with extra steps. Algorithm 1 outlines the proposed initialisation strategy, similar to that of Ning et al. (2008). It takes the initial partition of the data (whether obtained by hierarchical clustering, random initialisation, or some other method) and iteratively reallocates observations in such a way that each subset can be well-modelled by a single expert.

Algorithm 1: Iterative reallocation initialisation with expert network covariates

0 Concatenate the response data and expert network covariates into a matrix.
1 Obtain some non-overlapping hard starting partition $\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \ldots, \boldsymbol{\Omega}_G$.
2 Estimate the expert network regression $\eta_g(\boldsymbol{\gamma}_g, \cdot)$ on every subset $\{\boldsymbol{\Omega}_g\}_{g=1}^{G}$.
3 Compute the fitted values

$$\widehat{\mathbf{y}}_{ig} = \eta_g\big(\widehat{\boldsymbol{\gamma}}_g, \mathbf{x}_i^{(E)}\big) \;\forall\; (i, g)$$

and hence the residuals $\widehat{\mathbf{r}}_{ig} = \mathbf{y}_i - \widehat{\mathbf{y}}_{ig}$.
4 Compute $\widehat{\boldsymbol{\Psi}}_g = \mathrm{Cov}\big(\widehat{\mathbf{R}}_g\big) = \frac{1}{n-d_E-1}\widehat{\mathbf{R}}_g^{\top}\widehat{\mathbf{R}}_g \;\forall\; g$.
5 Compute the squared Mahalanobis distance $\widehat{M}_{ig} = \mathrm{d}_{\mathrm{M}}^2\big(\mathbf{y}_i, \widehat{\mathbf{y}}_{ig}\big) = \widehat{\mathbf{R}}_g^{\top}\widehat{\boldsymbol{\Psi}}_g^{-1}\widehat{\mathbf{R}}_g$.
6 Let $k_i = \arg\min_g\big(\widehat{M}_{ig}\big)$.
7 Reassign observation i to subset $\boldsymbol{\Omega}_{k_i}$.
8 Repeat Steps 2–7 until convergence is achieved, i.e. the partition ceases to change.

When using a deterministic approach to obtain the starting partition for Algorithm 1, initialisation can be further improved by considering information in the expert network covariates to find a good clustering of the joint distribution of $\left(\mathbf{y}_i, \mathbf{x}_i^{(E)}\right)$. When $\mathbf{x}_i^{(E)}$ includes categorical or ordinal covariates, a model-based approach to clustering mixed-type data (McParland and Gormley, 2016) can be employed in Step 1, though this is not considered further here.

If at any stage a level is dropped from a categorical variable in subset $\boldsymbol{\Omega}_g$ the variable itself is dropped from the corresponding regressor for the observations with missing levels. Convergence of the algorithm is guaranteed and the additional computational burden incurred is negligible. By using the Mahalanobis distance metric (Mahalanobis, 1936), each observation is assigned to the cluster corresponding to the Gaussian ellipsoid to which it is closest. This has the added advantage of potentially speeding up the running of the EM algorithm. The estimates of $\widehat{\gamma}_g$ at convergence are used as starting values for the expert network. The gating network is initialised by considering the partition itself at convergence as a discrete approximation of the gates.

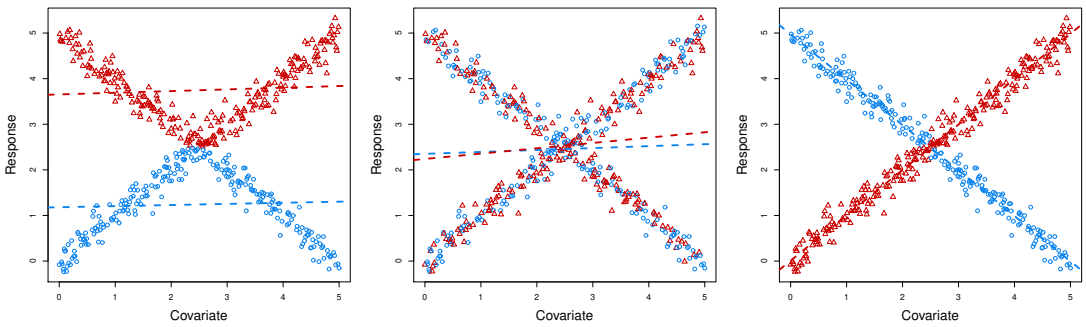
While convergence is monitored via the partition itself, Algorithm 1 implicitly finds the hard partition which minimises the total intra-component regression error criterion

$$\sum_{g=1}^{G} \min_{\{\eta_g, \gamma_g\}} \left(\sum_{i \in \boldsymbol{\Omega}_g} \mathrm{d}_{\mathrm{M}}^2\left(\mathbf{y}_i, \eta_g\left(\gamma_g, \mathbf{x}_i^{(E)}\right)\right)\right). \tag{2.10}$$

However, there are a few small caveats. Firstly, it suffices to use the Euclidean distance in place of the Mahalanobis distance for applications to univariate response data. Secondly, the Moore-Penrose pseudo-inverse (Moore, 1920) or p-dimensional identity matrix $\mathcal{I}_p$ is used in place of $\widehat{\boldsymbol{\Psi}}_g^{-1}$ when $n \leq p$. Finally, we note that Algorithm 1 applies only to the non-noise components; in the presence of a noise component, the $\widehat{\mathbf{Z}}$ matrix outputted by the algorithm at convergence is modified in the usual way.

Figure 2.3 illustrates the necessity of this procedure using a toy data set, with a single continuous covariate and a univariate response clearly arising from a mixture of two linear regressions, which otherwise would not be discerned without including the covariate in the initialisation routine via Algorithm 1. A further demonstration of the utility of this strategy is shown in Appendix 2.B.

Similar to the EM algorithm's susceptibility to local maxima, a limitation of our initialisation strategy is that the result at convergence may represent a suboptimal local minimum. However, the problem is transferred from the difficult task of initialising the EM algorithm to initialising Algorithm 1. Thus, it is feasible to repeat the algorithm with many different partitions and choose the best result — in the sense of minimising the criterion in (2.10) — to initialise one run of the EM algorithm, since Algorithm 1 converges very quickly, requires much less computational effort than the EM algorithm itself, and generally reduces the number of required EM iterations. However, we caution against using the total intra-component regression error criterion to guide the inclusion of expert network covariates.



(a) Hierarchical clustering of the response variables only.

(b) Random allocation.

(c) Partition obtained via Algorithm 1.

Figure 2.3: Initial 2-component hard partitions on univariate data clearly arising from a mixture of two linear regressions, obtained using (a) agglomerative hierarchical clustering, (b) random allocation, and (c) Algorithm 1 applied to the initialisation in (b) upon convergence after 6 iterations, demonstrating the improvement achieved by incorporating expert network covariates into the initialisation strategy. Allocations are distinguished using blue circles and red triangles. Corresponding fitted lines are also shown.

2.4.2 Model Selection

Whether a variable should be considered as a covariate or as part of the responses is usually clear from the context of the data being clustered and the related research question of interest. While the MoEClust model family is explicitly intended for use in such cases, a discussion of cases where it is not known which variables should be treated as response variables and which should be treated as covariates is provided in Appendix 2.E.

However, within the suite of MoE models outlined in Section 2.2.1.1, it is natural to question which variables — among the subset of variables designated as covariates, if any — are to be included, and if so in which part(s) of the MoE model. Unless the manner in which covariates enter is guided by the question of interest in the application under study, this is a challenging problem as the space of MoE models is potentially very large once variable selection for the covariates entering the gating and expert networks is considered. Thus, only models where covariates enter all mixture components or all component weights in a linear manner are typically considered in practice in order to restrict the size of the model search space. However, even within this reduced space, there are 2^r models to consider when $G = 1$ and 2^{2r} models to consider otherwise. Thus, the model space increases further if the number of components G is unknown.

Model comparison for the MoEClust family is even more challenging, especially for multivariate response data for which there are potentially 14 different GPCM covariance constraints to consider for models with $G \geq 2$ and 3 otherwise. When $p = 1$, there are 2 covariance constraints to consider when $G \geq 2$ and 1 otherwise. Considering constraints on the mixing proportions further increases the model search space. However, model selection can still be implemented in a similar manner to other model-based clustering methods: the Bayesian Information Criterion (BIC; Schwarz, 1978) and Integrated Completed Likelihood (ICL; Biernacki et al., 2000) have been shown to give suitable model selection criteria, both for the number of component densities (and thus clusters) required and for selecting covariates to include in the model. The number of free parameters in the penalty term for these criteria of course depends on the included gating and expert network covariates and the GPCM constraints employed.

For MoEClust models involving mixtures of GLMs, stepwise variable selection approaches can be used to find the optimal covariates for inclusion in either the multinomial logistic regression (gating network) or the weighted linear regression (expert network). Indeed, more parsimony can be achieved using variable selection, as there are a total of $G\,(d_G + 1) + Gp\,(d_E + 1)$ intercept and regression coefficients to estimate for a $G > 1$ full MoE model. However, the selected covariates may only be optimal for the given G and the given set of GPCM covariance matrix constraints. MoEClust models also allow for covariates entering only one part of the

model. Thus, we propose a greedy stepwise search whereby each step could involve adding/removing a component or adding/removing a single covariate in either the gating or expert networks. We adopt a forward search, starting from a $G = 1$ model, as backward selection can be particularly cumbersome when r is large. In the considered applications, it sufficed to consider only additions (of components and covariates) rather than additions and removals in the sense that the same final model was obtained despite fewer models being evaluated over the course of the search. Hence, the recommended forward search algorithm proceeds as follows:

Algorithm 2: Greedy forward stepwise search for MoEClust models

1 Choose the best $G = 1$ model with no covariates among all allowable model types.
2 Either:
- increase G by 1,
- add an explanatory variable to the expert network,
- add a concomitant variable to the gating network (only when $G \geq 2$).

3 For every action in Step 2, consider the full range of allowable GPCM constraints.
4 Accept the change which yields the best improvement in terms of BIC or ICL.
5 Repeat Steps 2–4 until there is no further improvement in the selection criterion.

While one could consider changing the GPCM constraints as another potential action in Step 2 of Algorithm 2, our experience suggests that increasing G or adding covariates (especially in the expert network) can radically alter the covariance structure. Thus, we advise changing the GPCM constraints simultaneously and identifying the optimum action by first finding the optimum constraints for each action. While this is more computationally intensive than altering the GPCM constraints as a step in itself, this makes the search less likely to miss optimal models as it traverses the model space. See Appendix 2.A for an example of how to conduct such a stepwise search using code from the `MoEClust` R package (Murphy and Murphy, 2019) and Appendix 2.E for simulation studies examining the performance of Algorithm 2 in the presence of uninformative covariates.

In certain special instances, some extra steps can be considered. When there are no gating network concomitants, a choice can be made, for each action, between fitted models with equal or unequal mixing proportions. We distinguish between G-component models without a noise component and models with $G - 1$ Gaussian components plus an additional noise component. Thus, we recommend treating models with a noise component differently, by running a stepwise search

for models excluding the possibility of a noise component, running a separate stepwise search starting from a $G = 0$ noise-only model, and ultimately choosing between the optimal models with and without a noise component identified by each search. In the presence of a noise component, one can also fit the GN and NGN models, given by (2.8) and (2.9) respectively, when evaluating every action involving models with gating network concomitants.

When r is not so prohibitively large as to render an exhaustive search infeasible, Gormley and Murphy (2010) demonstrate how model selection criteria such as the BIC can be employed to choose the appropriate number of components and guide the inclusion of covariates across the six special cases of the MoE model described in Section 2.2.1.1. Adapting this approach to MoEClust models where GPCM constraints must also be chosen requires fixing the covariates to be included in the component weights and densities and finding the G value and GPCM covariance structure which together optimise some criterion. Different fits with different combinations of covariates are then compared according to the same criterion. However, due to the highlighted computational difficulties when r is large, Algorithm 2 remains the recommended approach.

2.5 Results

The clustering performance of the MoEClust models is illustrated by application to two well-known data sets: univariate CO_2 data (Section 2.5.1) and multivariate data from the Australian Institute of Sports (Section 2.5.2). Additional results are provided for each data set in the Appendices. In particular, code examples (Appendix 2.A), details of the initialisation (Appendix 2.B), and results from a predictive rather than clustering point of view (Appendix 2.D) for the CO_2 data, and results of the stepwise search (Appendix 2.C) for the AIS data, are given. Furthermore, Appendix 2.E examines issues around identifying responses and covariates and identifying the informative subsets of covariates among those variables designated as covariates.

Hereafter, any mention of methods for initialising the allocations, when covariates enter the expert network, refers to finding a single initial partition for Algorithm

1. The BIC and the stepwise search strategy outlined in Algorithm 2 were used to find the optimal number of components, choose the covariance type, and select the best subset of covariates, as well as where to put them. Results of exhaustive searches are also provided for demonstrative purposes. All results were obtained using the R package `MoEClust` (Murphy and Murphy, 2019).

2.5.1 CO_2 Data

As a univariate example of an application of MoEClust, data sourced from the OECD on the CO_2 emissions of $n = 28$ countries in the year 1996 (Hurn et al., 2003) are clustered, with Gaussian component densities. Studying the relationship between CO_2 and the covariate Gross National Product (GNP), both measured *per capita*, is of interest. As consideration is only being given to inclusion/exclusion of a single covariate in the gating and/or expert networks, an exhaustive search is feasible. A range of models ($G \in \{1, \dots, 9\}$) are fitted, with either the equal (E) or unequal variance (V) models from Table 2.1. Quantile-based clustering of the CO_2 values is used to initialise Algorithm 1 when the expert network excludes GNP, otherwise hierarchical clustering of both CO_2 and GNP is used.

Table 2.2 gives BIC and ICL values for the top model under each of the six special cases of the MoE framework. The chosen model had $G = 3$, equal variances (i.e. the E constraint), equal mixing proportions, and GNP in the expert network; thus, this is an *equal mixing proportion expert network MoE model*. This model maximised both the BIC and ICL criteria, and was also identified by the forward stepwise search described in Algorithm 2, starting from a $G = 1$ model (BIC=-163.90), adding a component (BIC=-163.16), adding GNP to the expert network and changing to the V model type (BIC=-157.20), and finally adding a further component, constraining the mixing proportions, and changing back to the E model type (BIC=-155.20). Thereafter, neither adding a component nor adding GNP to the gating network improved the BIC. Code to reproduce both the exhaustive and stepwise searches using the `MoEClust` R package is given in Appendix 2.A.

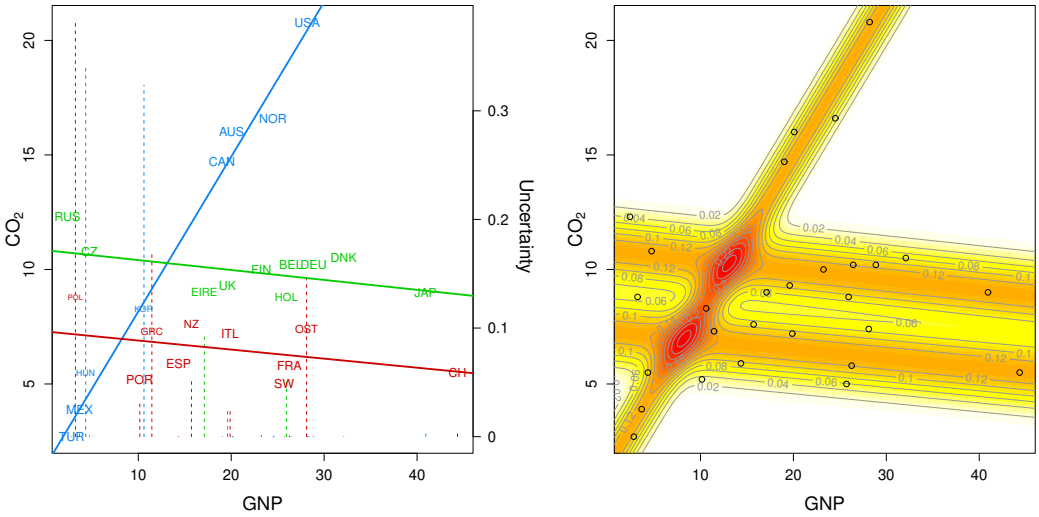
Table 2.2: The MoEClust BIC and ICL values of the top models under the six MoE special cases for the CO_2 data. Each row is optimal with respect to G and GPCM type, given the included covariates.

Special Case	Gating	Expert	G	GPCM	BIC	ICL
Mixture Model			2	E	−163.16	−163.91
Expert Network MoE Model		GNP	2	V	−157.20	−160.04
Gating Network MoE Model	GNP		2	E	−166.05	−166.68
Full MoE Model	GNP	GNP	2	V	−159.25	−161.47
Equal Mixing Proportion Mixture Model	Equal		2	V	−165.19	−184.71
Equal Mixing Proportion Expert Network MoE Model	Equal	GNP	3	E	**−155.20**	**−159.06**

Repeating both the exhaustive and stepwise searches with the addition of a noise component for all models also failed to yield any model with an improved BIC. The fourth row of Table 2.2 corresponds to a *full MoE*, with GNP included in both parts of the model; its sub-optimal BIC highlights the benefits of the model selection approach. The parameters of the optimal model are given in Table 2.3. Its fit is exhibited in Figure 2.4, which shows that the relationship between CO_2 and GNP is clustered around three different linear regression lines; one cluster of 8 countries with a large slope value and two equally-sized clusters, each with different intercepts but similar near-zero slope values. Clustering uncertainties, given by $\widehat{U}_i = \min_{g \in \{1,\ldots,\widehat{G}\}}(1 - \widehat{z}_{ig})$, are also shown.

Table 2.3: Estimated parameters of the optimal MoEClust model fit to the CO_2 data.

Parameter	Component 1	Component 2	Component 3
Proportion	1/3	1/3	1/3
(Intercept)	1.41	7.29	10.84
GNP	0.68	−0.04	−0.04
σ^2_g	0.98	0.98	0.98



(a) Fitted lines of the expert network GLMs. Text label size is proportional to a country's probability of belonging to its assigned cluster. Clustering uncertainty is also indicated by dotted vertical bars relating to the second y-axis. Colours correspond to the MAP classification.

(b) Heat map of the conditional density of the outcome variable CO_2, accounting for the gating and expert networks, the latter of which includes GNP as a covariate.

Figure 2.4: Scatter plots of GNP against CO_2 emissions for $n = 28$ countries with three linear regression components from the optimal MoEClust model with equal variances and mixing proportions.

The optimal model contains GNP in the expert network and has constraints on the component variances and mixing proportions. These are features of the MoEClust models which neither MoE nor GPCM models can fully accommodate. While `flexmix` and `mixtools` can fit the sub-optimal expert network MoE model in row four of Table 2.2, with unequal variances and mixing proportions (which achieves the second highest BIC value), our initialisation strategy ultimately leads to the same or higher BIC estimates. Across 50 random starts, BIC values of -157.29 and -157.20 are achieved using `flexmix` and `mixtools`, respectively. Among these random starts, BIC values as low as -163.67 are obtained. However, the `MoEClust` R package, with Algorithm 1 invoked, achieves a BIC of -157.20 with only a single initial partition. Using `MoEClust` without this initialisation strategy also yields the lower BIC value of -163.67. A further demonstration of the advantages of our initialisation strategy, using instead the optimal model for the the CO_2 data, is provided in Appendix 2.B. Finally, we note that additional results for the CO_2 data from the point of view of predicting the response, rather than clustering, are provided in Appendix 2.D.

2.5.2 Australian Institute of Sport (AIS) Data

Various physical and hematological (blood) measurements were made on 102 male and 100 female athletes at the Australian Institute of Sport (AIS; Cook and Weisberg, 1994). The thirteen variables recorded in the study are detailed in Table 2.4.

Table 2.4: Australian Institute of Sports data variables. The $p = 5$ in the first column are hematological response variables and the others, the $r = 8$ covariates, are physical measurements for the athlete.

Response	Description	Covariate	Description (Units)
RCC	red cell count	BMI	body mass index (kg/m^2)
WCC	white cell count	SSF	sum of skin folds (mm)
Hc	Hematocrit	Bfat	body fat percentage (%)
Hg	Hemoglobin	LBM	lean body mass (kg)
Fe	plasma ferritin	Ht	height (cm)
	concentration	Wt	weight (kg)
		sex	a factor with levels: female, male
		sport	a factor with levels: Basketball, Field, Gymnastics, Netball, Rowing, Swimming, Tennis, Track 400m, Track Sprint, Water Polo

MoEClust models are used to investigate the clustering structure in the athletes' hematological measurements and investigate how covariates may influence these measurements and the clusters. Cluster allocations are initialised using model-based agglomerative hierarchical clustering. Results of the forward stepwise model search described in Algorithm 2, with all covariates considered for inclusion, are given in Appendix 2.C. The optimal model (BIC=-4010.14) is a 2-component EVE *equal mixing proportion expert network MoE model*, which thus has clusters of equal size, volume, and orientation, and unequal shape. Notably, the only covariate (sex), only enters in one part of the model, the expert network.

The sub-optimal BIC values for the best model with all covariates in both parts of the model ($G = 1$, EEE, BIC=-4234.79), which is the same as the best model with all covariates in the expert network only (regardless of τ being constrained or not), and all covariates in the gating network only ($G = 2$, VEE, BIC=-4092.72), highlight the need for the model selection strategy employed. As the optimal model uses the EVE constraints, it has 19 covariance parameters; an otherwise exactly equivalent VVV model, having 30 such parameters, yields a lower BIC

of -4056.19, thus showcasing the benefits of the parsimonious covariance constraints. The difference of 11 covariance parameters between these models is exactly one more than the number of regression parameters introduced by the expert network covariate.

Subsequently, and purely for the purposes of comparing certain special cases of interest, an exhaustive search over a range of MoEClust models is conducted, with $G \in \{1, \dots, 9\}$. This is rendered feasible by only considering the covariates BMI and sex; allowing either, neither, or both to enter either, neither, or both of the gating and expert networks. Note that BMI is itself computed using the covariates measuring weight (Wt) and height (Ht). With 3 permissible covariance parameterisations for the single component models, and 14 otherwise, 16 possible combinations of gating and/or expert network covariate settings, and consideration also being given to models with equal mixing proportions, this still requires fitting $2,252$ MoEClust models. However, some spurious solutions were found, particularly for higher values of G, in the sense that models with empty components or degenerate components with few observations reduced to equivalent models with fewer non-empty components (see Section 2.3.3). Table 2.5 gives the BIC and ICL values of a selection of these fitted models, representing the optimal models for certain special cases of interest.

Table 2.5: The BIC and ICL values for a selection of MoEClust models fitted to the Australian Institute of Sports data. Rows 1 and 2 give the optimal models under settings available in `flexmix`; models without expert network covariates, using either the VVV or VVI covariance constraints. Among the more general MoEClust family, the last row gives the top model according to the ICL criterion and the remaining rows give the top models according to the BIC criterion for each of the six special cases of the MoE framework. Thus, row 3 corresponds to the optimal model according to `mclust`.

Rank (BIC)	Gating	Expert	G	GPCM	BIC	ICL	No. Parameters
206	sex		2	VVV	-4113.31	-4121.32	42
896	sex		5	VVI	-4319.85	-4345.55	58
301			2	EVE	-4146.16	-4201.61	30
3		sex	2	EVE	-4015.35	-4059.54	40
24	sex		3	EVE	-4037.32	-4066.66	42
2	BMI	sex	2	EVE	-4013.40	-4074.11	41
277	Equal		2	EVE	-4140.98	-4192.21	29
1	Equal	sex	2	EVE	$\mathbf{-4010.14}$	-4057.87	39
26	BMI, sex		3	EEE	-4038.75	$\mathbf{-4043.01}$	36

Clearly, the inclusion of covariates improves the fit compared to GPCM models. Similarly, using GPCM covariance constraints improves the fit compared to standard Gaussian MoE models. In particular, it is notable that the optimal models using the VVV and VVI constraints only have covariates enter the gating network. This suggests that the parsimony afforded by the remaining GPCM settings somewhat offsets the number of regression parameters introduced to the expert network.

The top three models according to BIC all have 2 components, the EVE covariance constraints, and the covariate sex in the expert network; they differ only in their treatment of the gating network. Models with equal and unequal mixing proportions, and with BMI as a gating concomitant, have zero, one, and two associated gating network parameters, respectively. The optimal model has equal mixing proportions and was also identified above via Algorithm 2. The full MoE model with BMI in the gating network and sex in the expert network is an interesting case as it does not fit the framework of Dang and McNicholas (2015), which assumes that when covariates enter the model, they enter in both parts. The best such model has 'sex' in both networks ($G = 2$, EVE) and achieves a BIC of -4020.22 with a corresponding rank of 8.

Up to now, models with a noise component have not yet been considered for the AIS data. Thus, another stepwise search is conducted, including a noise component for all candidate models and starting from a $G = 0$ noise-only model (see Appendix 2.C). Consideration was also given to both the GN and NGN model types, in (2.8) and (2.9) respectively, where models included gating concomitants, and to models with equal/unequal mixing proportions for the non-noise components for models without gating concomitants. The optimal full MoE model thus found has two EEE Gaussian clusters and an additional noise component. The covariate 'sex' enters the expert network (see Table 2.6). Both 'SSF' and 'Ht' enter the gating network, though not for the noise component, which has a constant mixing proportion ($\widehat{\tau}_0 \approx 0.08$), as per the NGN model in (2.9). Thus, the Gaussian clusters have equal volume, shape, and orientation, but unequal size. This model achieves a BIC value of -3989.83, which compares favourably to the previously optimal model from Table 2.5, adding a noise component to a model otherwise identical to the optimal model from Table 2.5 (BIC=-3992.81), and to models with a noise component but no stepwise selection of covariates (or no covariates at all).

Table 2.6: Coefficients of the expert network linear regressions for the $G = 2$ Gaussian clusters in the optimal 'full' MoEClust model (with an extra noise component and gating concomitants entering the non-noise clusters only) fit to the AIS data, with female as the reference level for the explanatory variable 'sex'.

	RCC	WCC	Hc	Hg	Fe
Cluster 1					
(Intercept)	4.56	6.89	42.33	14.08	49.73
sexmale	0.42	0.12	2.95	1.30	28.19
Cluster 2					
(Intercept)	4.26	6.93	38.91	13.11	59.70
sexmale	0.86	0.59	7.36	2.80	132.66

The gating network has an intercept of 10.58 and slope coefficients of 0.04 (SSF) and -0.08 (Ht) with corresponding odds ratios of 1.04 and 0.93. Thus, higher SSF values increase the probability of belonging to the second Gaussian cluster, to which taller athletes are less likely to belong, and the probability of belonging to the noise component is constant. Though every observation has its own mean parameter in the presence of expert covariates, given by the fitted values of the expert network (shown in Table 2.6), the means are summarised in Table 2.7 by the posterior mean of the fitted values of the model according to (2.11). The noise component is accounted for by $\overline{\boldsymbol{\mathcal{V}}}$, the p-dimensional centroid of the region used to estimate V:

$$\widehat{\boldsymbol{\mu}}_g = \frac{\sum_{i=1}^{n} \widehat{z}_{ig}\widehat{\mathbf{y}}_i}{\sum_{i=1}^{n} \widehat{z}_{ig}} = \frac{\sum_{i=1}^{n} \widehat{z}_{ig}\left(\sum_{g=1}^{G} \widehat{z}_{ig}\left(\widetilde{\mathbf{x}}_i^{(E)}\widehat{\boldsymbol{\gamma}}_g\right) + \widehat{z}_{i0}\overline{\boldsymbol{\mathcal{V}}}\right)}{\sum_{i=1}^{n} \widehat{z}_{ig}}. \tag{2.11}$$

Given that there exists a binary variable, 'sex', in the expert network for the optimal MoEClust model, there are effectively four Gaussian components plus an additional noise component. By virtue of the EEE constraint on the Gaussian components, all four components and thus both clusters in fact share the same covariance matrix. Components 1 and 2, corresponding to females and males in Cluster 1, share the same covariance matrix but differ according to their means. The same is true for females and males (Components 3 and 4) in Cluster 2. Table 2.7 gives the means and average gates in terms of both components and clusters, as well as the common $\widehat{\boldsymbol{\Sigma}}$ matrix.

Table 2.7: Estimated parameters of the $G = 2$ Gaussian clusters in the optimal 'full' MoEClust model fit to the AIS data (with an extra noise component and gating concomitants entering the non-noise clusters only), with further splitting due to the binary covariate sex in the expert network, giving average gates and component means (for females and males) and the common EEE covariance matrix. While every observation has its own mean parameter, given by the fitted values of the expert network in Table 2.6, the means are summarised by the posterior mean of the model's fitted values, given by (2.11).

	Cluster 1			Cluster 2			$\widehat{\boldsymbol{\Sigma}}$ (EEE)				
	All	Female	Male	All	Female	Male					
$\widehat{\tau}_g$	0.60	0.22	0.38	0.33	0.25	0.07	RCC	WCC	Hc	Hg	Fe
RCC	4.81	4.51	4.98	4.51	4.33	5.12	0.08	0.08	0.46	0.15	−0.83
WCC	7.02	6.95	7.06	7.10	6.96	7.57		2.50	0.60	0.21	5.12
Hc	44.06	41.79	45.35	41.14	39.61	46.29			3.84	1.33	−7.55
Hg	14.88	13.94	15.51	13.91	13.32	15.90				0.57	−1.05
Fe	70.18	53.05	79.87	87.84	58.96	184.67					821.68

Though the plots in Figure 2.4 are suitable for univariate data with a single continuous expert network covariate, visualising MoEClust results for multivariate data with $r > 1$ mixed-type covariates constitutes a much greater challenge. For the optimal full MoE model fit to the AIS data, the data and clustering results are shown using a generalised pairs plot in Figure 2.5. This plot depicts the pairwise relationships between the hematological response variables, the included gating and expert network covariates, and the MAP classification, coloured according to the MAP classification. The marginal distributions of each variable are given along the diagonal. For the hematological response variables, ellipses with axes related to the within-cluster covariances are drawn.

For the purposes of visualising Figure 2.5, owing to the presence of an expert network covariate in the fitted model, the multivariate Gaussian ellipses in panels depicting two response variables are centred on the posterior mean of the fitted values, as described in (2.11). Their shape and size are also modified for the same reason: they are derived by adding the extra variability in the component means to $\widehat{\boldsymbol{\Sigma}}_g$. Thus, the depicted ellipses do not conform to the EEE covariance constraints of the optimal model.

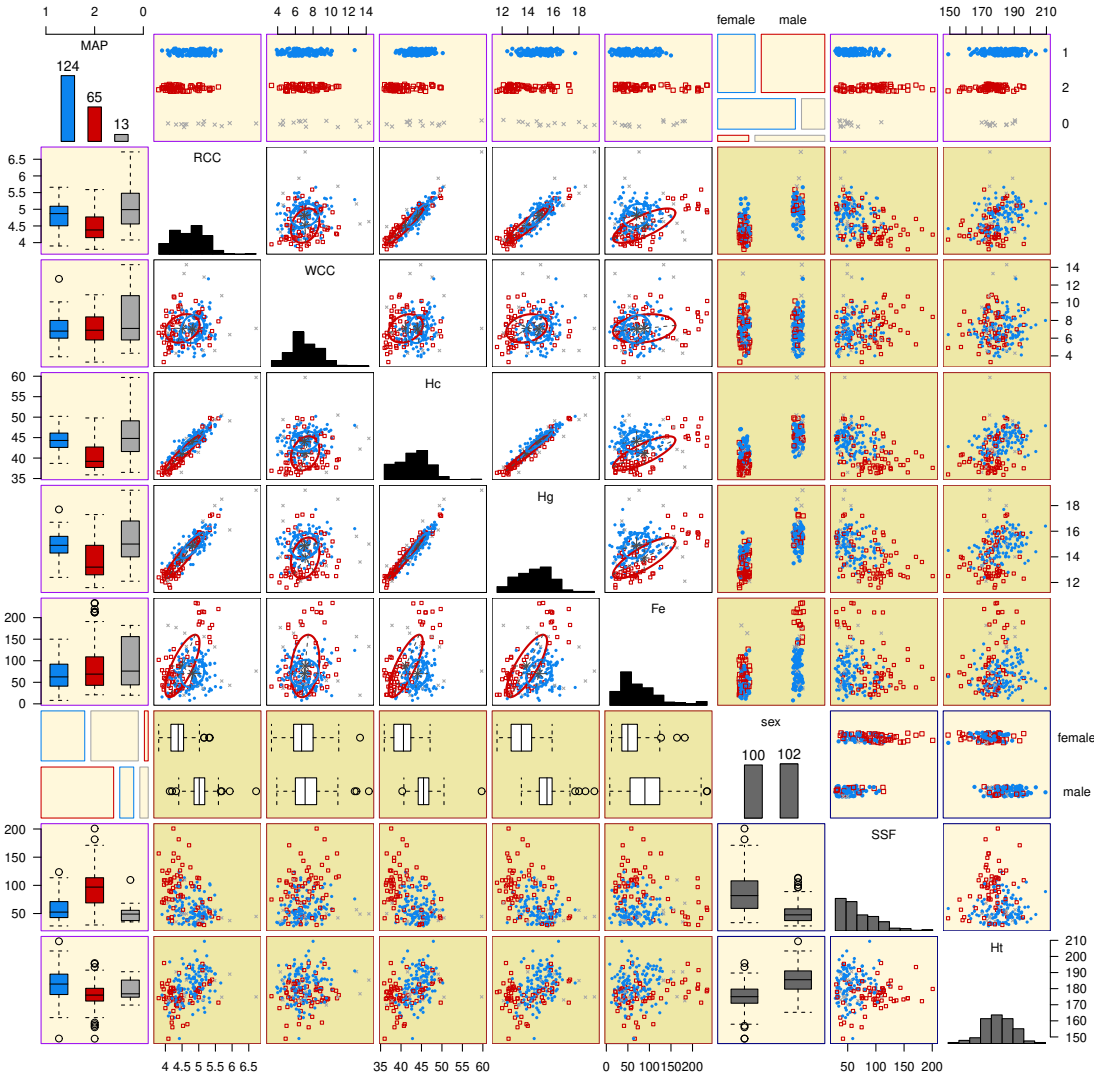


Figure 2.5: Generalised pairs plot for the optimal 'full' MoEClust model fit to the AIS data, depicting pairwise relationships between the hematological response variables, the expert network covariate sex, the gating concomitants SSF and Ht, and the MAP classification. Colours and plotting symbols correspond to the MAP classification: blue circles and red squares for the two Gaussian clusters; grey crosses for the 4 female and 9 male outlying observations assigned to the uniform noise component. Mosaic plots are used to depict two categorical variables, scatter plots are used for panels involving two continuous variables, and a mix of box-plots and jittered strip-plots are used for mixed pairs.

It is clear from Figure 2.5 that the variables 'Hematocrit' (Hc), 'Hemoglobin' (Hg), and 'plasma ferritin concentration' (Fe), and the gating network concomitants 'SSF' and 'Ht', are driving much of the separation between the clusters. On

the other hand, the expert network covariate 'sex' is driving separation within the Gaussian clusters. The correspondence between the MAP classification and the sex label is notably poor, with an adjusted Rand index (ARI; Hubert and Arabie, 1985) of just 0.11 (see Table 2.8a). This is because, under the optimal model, the athletes' size in terms of their SSF and height measurements, rather than their sex, influences the probability of cluster membership, and athletes are divided by sex within each cluster rather than the clusters necessarily capturing their sex. The ARI is higher for models where sex does not enter the expert network, especially when it instead enters the gating network, though such fitted models all have sub-optimal BIC values (see Table 2.5).

Indeed, Table 2.6 implies that males, on average, have elevated levels of all five blood measurements in both Gaussian clusters. However, the magnitude of this effect is more pronounced in Cluster 2, related to athletes with higher average SSF measurements (a proxy for body fat) and lower average height. Interestingly, Figure 2.5 also shows that females have higher average SSF measurements and lower average height; this may explain why there are more males than females in Cluster 1, and the reverse in Cluster 2, given the signs of the gating network coefficients for SSF (0.04) and Ht (-0.08).

Given that one of the unused covariates ('sport') is categorical, the concordance between the MAP classification and the athletes' sport can also be assessed, even if there are many more than $G = 2$ levels to the sport covariate. Table 2.8 comprises three sub-tables showing the cross-tabulation of the MAP classification against 'sex' (Table 2.8a), 'sport' (Table 2.8b), and a new categorical variable obtained by splitting each level of the sport covariate into male and female players (Table 2.8c). In the latter case, it is worth noting that there are no male gymnasts and no female water polo players. Despite the poor ARI values, some interesting conclusions can still be drawn. For instance, all male basketball players are isolated in a single cluster, the majority of the (all female) netball players are isolated in a single cluster, most of the rowers are assigned to the first cluster, and most of the male Track 400m and Water Polo athletes are in the first cluster also. This suggests that collapsing the sport covariate into fewer categorical levels — by, for instance, distinguishing between water-based sports and the others — could prove fruitful in future analyses.

Table 2.8: Cross-tabulations of the MAP classification under the optimal 'full' MoEClust model against the 'sex' covariate (Table 2.8a), the 'sport' covariate (Table 2.8b), and a combination of both covariates (Table 2.8c). Note that 'sex' was used as an expert network covariate in the fitted model, while 'sport' was not selected at all. Corresponding ARI values are given in each case.

(a) ARI $= 0.11$.

'sex'	Female	Male
Cluster 1	46	78
Cluster 2	50	15
Noise	4	9

(b) ARI $= 0.07$.

'sport'	Basketball	Field	Gymnastics	Netball	Rowing	Swimming	Tennis	Track 400m	Track Sprint	Water Polo
Cluster 1	17	7	1	1	29	17	6	25	9	12
Cluster 2	8	8	3	21	8	4	4	3	3	3
Noise	0	4	0	1	0	1	1	1	3	2

(c) ARI $= 0.05$.

'sport'	Basketball		Field		Gymnastics		Netball		Rowing	
'sex'	Female	Male	Female	Male	Female	Male	Female	Male	Female	Male
Cluster 1	5	12	1	6	1	0	1	0	15	14
Cluster 2	8	0	4	4	3	0	21	0	7	1
Noise	0	0	2	2	0	0	1	0	0	0
'sport'	Swimming		Tennis		Track 400m		Track Sprint		Water Polo	
'sex'	Female	Male	Female	Male	Female	Male	Female	Male	Female	Male
Cluster 1	7	10	3	3	9	16	4	5	0	12
Cluster 2	2	2	3	1	2	1	0	3	0	3
Noise	0	1	1	0	0	1	0	3	0	2

2.6 Discussion

The development of a suite of MoEClust models has been outlined, clearly demonstrating the utility of mixture of experts models for parsimonious model-based clustering where covariates are available. A novel means of visualising such models has also been proposed. The ability of MoEClust models to jointly model the response variable(s) and related covariates provides deeper and more principled insight into the relations between such data in a mixture-model based analysis, and provides a principled method for both creating and explaining the clustering, with

reference to information contained in covariates. Their demonstrated use to cluster observations and appropriately capture heterogeneity in cross-sectional data provides only a glimpse of their potential flexibility and utility in a wide range of settings. Indeed, given that general MoE models have been used, under different names, in several fields, including but not limited to statistics (Grün and Leisch, 2007, 2008), biology (Wang et al., 1996), econometrics (Wang et al., 1998), marketing (Wedel and Kamakura, 2012), and medicine (Thompson et al., 1998), MoEClust models could prove useful in many domains.

Improvement over GPCM models has been introduced by accounting for external information available in the presence of potentially mixed-type covariates. Similarly, improvement over Gaussian mixture of experts models which incorporate fixed covariates has been introduced by allowing GPCM family covariance structures in the component densities. MoEClust models are thus Gaussian parsimonious MoE models where the size, volume, shape, and/or orientation can be equal or unequal across components. MoEClust models have been further extended to accommodate the presence of an additional uniform noise component to capture departures from Gaussianity, in such a way that observations are smoothly classified as belonging to Gaussian clusters or as outliers. In particular, two means of doing so have been proposed for models which include gating concomitants. Due to sensitivity of the final solution obtained by the EM algorithm to the initial values, an iterative reallocation procedure based on the Mahalanobis distance has been proposed to mitigate against convergence to suboptimal local maxima for models with expert network covariates. This initialisation algorithm converges quickly and also speeds up convergence of the EM algorithm itself.

Previous parsimonious Gaussian mixtures of experts (Dang and McNicholas, 2015) accommodated only the cases where all covariates enter the expert network MoE model, or the full MoE model with the restriction that all covariates enter both parts of the model. MoEClust constitutes a unifying framework whereby different subsets of covariates can enter either, neither, or both the gating and/or expert networks of Gaussian parsimonious MoE models. Considering the standard mixture model (with no dependence on covariates), or the expert network MoE model, with the equal mixing proportion constraint expands the model family further.

On a cautionary note, care must be exercised in choosing how and where covariates enter when a MoEClust model is used as a clustering tool, as the interpretation of the analysis fundamentally depends on where covariates enter, which of the six special cases of the MoE framework is invoked, and on which GPCM constraints are employed. To this end, a novel greedy forward stepwise search algorithm has been employed for model/variable selection purposes. This strategy has the added advantage of introducing additional parsimony, by potentially reducing the number of regression parameters in the gating and/or expert networks.

Gating network MoEClust models may be of particular interest to users of GPCMs; while concomitants influence the probability of cluster membership, the correspondence thereafter between component densities and clusters has the same interpretation as in standard GPCMs. When covariates enter the component densities, we warn that observations with very different response values can be clustered together, because they are being modelled using the same GLM; similarly, regression distributions with distinct parameters do not necessarily lead to well-separated clusters.

MoEClust models allow the number of parameters introduced by gating and expert network covariates to be offset by a reduction in the number of covariance parameters. This is particularly advantageous when model selection is conducted using the BIC or ICL, which include a penalty term based on the parameter count. Thus, MoEClust models may tend to favour including covariates more than standard Gaussian MoE models would. This is particularly true for explanatory variables in the expert network, which tend to necessitate more regression parameters (Gp) than concomitant variables in the gating network ($G - 1$) per additional continuous covariate or level of categorical covariates included. Thus, in cases where a MoE model might elect to include a concomitant variable in the gating network, a MoEClust model with fewer covariance parameters may elect to include it as an explanatory expert network variable instead. While this does lead to a better fit, it can complicate interpretation.

Possible directions for future work in this area include investigating the utility of nonparametric estimation of the gating network (Young and Hunter, 2010), as well as exploring the use of regularisation penalties in the gating and expert networks to help with variable selection when the number of covariates r is large.

Regularisation in another, Bayesian sense, by specifying a prior on the component variances/covariances in the spirit of Fraley and Raftery (2007), and/or component regression parameters, could also prove useful for avoiding spurious solutions due to computational singularity described in Section 2.3.3. MoEClust models could also be developed in the context of hierarchical mixtures of experts (Jordan and Jacobs, 1994), and/or extended to the supervised or semi-supervised model-based classification settings, where some or all observations are labelled.

Beyond the family of GPCM constraints, MoEClust models could be extended to avail of parsimonious factor-analytic covariance structures for high-dimensional data (McNicholas and Murphy, 2008). These could be incorporated into Gaussian mixture of experts models using residuals in an equivalent fashion to Section 2.3.2 above. Similarly, MoEClust models could benefit from the heavier tails of the multivariate t-distribution, and the robustness to outliers it affords, by considering the associated tEIGEN family of covariance constraints (Andrews and McNicholas, 2012). However, the inclusion of a uniform noise component has the advantage of drawing a clearer distinction between observations belonging to clusters or designated as outliers.

References

Andrews, J. L. and P. D. McNicholas (2012). Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions: the tEIGEN family. *Statistics and Computing 22*(5), 1021–1029. 26, 56

Banfield, J. and A. E. Raftery (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics 49*(3), 803–821. 21, 26, 35

Benaglia, T., D. Chauveau, D. R. Hunter, and D. Young (2009). mixtools: an R package for analyzing finite mixture models. *Journal of Statistical Software 32*(6), 1–29. 29

Biernacki, C., G. Celeux, and G. Govaert (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*(7), 719–725. 40

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer. 22

Böhning, D., E. Dietz, R. Schaub, P. Schlattmann, and B. G. Lindsay (1994). 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. 32

Celeux, G. and G. Govaert (1995). Gaussian parsimonious clustering models. *Pattern Recognition 28*(5), 781–793. 21, 25, 26

Cook, R. D. and S. Weisberg (1994). *An Introduction to Regression Graphics*. New York: John Wiley & Sons. 46

Dang, U. J. and P. D. McNicholas (2015). Families of parsimonious finite mixtures of regression models. In I. Morlini, T. Minerva, and M. Vichi (Eds.), *Advances in Statistical Models for Data Analysis: Studies in Classification, Data Analysis, and Knowledge Organization*, pp. 73–84. Switzerland: Springer. 21, 30, 48, 54

Dang, U. J., A. Punzo, P. D. McNicholas, S. Ingrassia, and R. P. Browne (2017). Multivariate response and parsimony for Gaussian cluster-weighted models. *Journal of Classification 34*(1), 4–34. 20, 30

Dayton, C. M. and G. B. Macready (1988). Concomitant-variable latent-class models. *Journal of the American Statistical Association 83*(401), 173–178. 20, 22

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 39*(1), 1–38. 31

Eisenhauer, J. G. (2003). Regression through the origin. *Teaching Statistics 25*(3), 76–80. 35

Fraley, C. and A. E. Raftery (2007). Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of Classification 24*(2), 155–181. 56

García-Escudero, L. A., A. Gordaliza, F. Greselin, S. Ingrassia, and A. Mayo-Iscar (2018). Eigenvalues and constraints in mixture modeling: geometric and computational issues. *Advances in Data Analysis and Classification 12*(2), 203–233. 35

Geweke, J. and M. Keane (2007). Smoothly mixing regressions. *Journal of Econometrics 138*(1), 252–290. 23

Gormley, I. C. and T. B. Murphy (2010). Clustering ranked preference data using sociodemographic covariates. In S. Hess and A. Daly (Eds.), *Choice Modelling: The State-of-the-art and The State-of-practice – Proceedings from the Inaugural International Choice Modelling Conference*, Chapter 25, pp. 543–569. United Kingdom: Emerald. 42

Gormley, I. C. and T. B. Murphy (2011). Mixture of experts modelling with social science applications. In K. Mengersen, C. Robert, and D. M. Titterington (Eds.), *Mixtures: Estimation and Applications*, Chapter 9, pp. 101–121. New York: John Wiley & Sons. 24

Grün, B. and F. Leisch (2007). Fitting finite mixtures of generalized linear regressions in R. *Computational Statistics & Data Analysis 51*(11), 5247–5252. 29, 54

Grün, B. and F. Leisch (2008). FlexMix version 2: finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software 28*(4), 1–35. 29, 54

Hennig, C. (2000). Identifiability of models for clusterwise linear regression. *Journal of Classification 17*(2), 273–296. 20

Hennig, C. and P. Coretto (2008). The noise component in model-based cluster analysis. In C. Preisach, H. Burkhardt, L. Schmidt-Thieme, and R. Decker (Eds.), *Data Analysis, Machine Learning and Applications: Studies in Classification, Data Analysis, and Knowledge Organization*, pp. 127–138. Berlin, Heidelberg: Springer. 36

Hubert, L. and P. Arabie (1985). Comparing partitions. *Journal of Classification 2*(1), 193–218. 52

Hurn, M., A. Justel, and C. P. Robert (2003). Estimating mixtures of regressions. *Journal of Computational and Graphical Statistics 12*(1), 55–79. 43

Ingrassia, S., S. C. Minotti, and G. Vittadini (2012). Local statistical modeling via the cluster-weighted approach with elliptical distributions. *Journal of Classification 29*(3), 363–401. 20, 26

Ingrassia, S. and A. Punzo (2019). Cluster validation for mixtures of regressions via the total sum of squares decomposition. *Journal of Classification*, 1–22. URL https://doi.org/10.1007/s00357-019-09326-4. 71, 72, 76

Ingrassia, S., A. Punzo, G. Vittadini, and S. C. Minotti (2015). The generalized linear mixed cluster-weighted model. *Journal of Classification 32*(1), 85–113. 20

Jacobs, R. A., M. I. Jordan, S. J. Nowlan, and G. E. Hinton (1991). Adaptive mixtures of local experts. *Neural Computation 3*(1), 79–87. 20, 22, 23

Jordan, M. I. and R. A. Jacobs (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation 6*(2), 181–214. 56

Lamont, A. E., J. K. Vermunt, and M. L. Van Horn (2016). Regression mixture models: does modeling the covariance between independent variables and latent classes improve the results? *Multivariate Behavioural Research 51*(1), 35–52. 20

Lebret, R., S. Iovleff, F. Langrognet, C. Biernacki, G. Celeux, and G. Govaert (2015). Rmixmod: the R package of the model-based unsupervised, supervised, and semi-supervised classification mixmod library. *Journal of Statistical Software 67*(6), 1–29. 29, 30

Mahalanobis, P. C. (1936). On the generalised distance in statistics. *Proceedings National Institute of Science, India 2*(1), 49–55. 38

Mazza, A., A. Punzo, and S. Ingrassia (2018). flexCWM: a flexible framework for cluster-weighted models. *Journal of Statistical Software 86*, 1–27. 30, 75

McCullagh, P. and J. Nelder (1983). *Generalized Linear Models*. London: Chapman and Hall. 23

McNicholas, P. D. and T. B. Murphy (2008). Parsimonious Gaussian mixture models. *Statistics and Computing 18*(3), 285–296. 26, 56

McParland, D. and I. C. Gormley (2016). Model based clustering for mixed data: clustMD. *Advances in Data Analysis and Classification 10*(2), 155–169. 38, 75, 76

Milner, K. and J. Rougier (2014). How to weigh a donkey in the Kenyan countryside. *Significance 11*, 40–43. 74, 76

Moore, E. H. (1920). On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society 26*(9), 394–395. 38

Murphy, K. and T. B. Murphy (2019). `MoEClust`*: Gaussian parsimonious clustering models with covariates and a noise component*. R package version 1.3.0. 21, 41, 43, 63

Ning, H., Y. Hu, and T. S. Huang (2008). Efficient initialization of mixtures of experts for human pose estimation. In *Proceedings of the International Conference on Image Processing, ICIP 2008, October 12-15, 2008, San Diego, California, USA*, pp. 2164–2167. 37

Punzo, A. and S. Ingrassia (2015). Parsimonious generalized linear Gaussian cluster-weighted models. In I. Morlini, T. Minerva, and M. Vichi (Eds.), *Advances in Statistical Models for Data Analysis: Studies in Classification, Data Analysis, and Knowledge Organization*, pp. 201–209. Switzerland: Springer. 30

Punzo, A. and S. Ingrassia (2016). Clustering bivariate mixed-type data via the cluster-weighted model. *Computational Statistics 31*(3), 989–103. 20

Punzo, A. and P. D. McNicholas (2016). Parsimonious mixtures of multivariate contaminated normal distributions. *Biometrical Journal 58*(6), 1506–1537. 26

R Core Team (2019). *R: a language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. 21, 29

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*(2), 461–464. 40

Scrucca, L., M. Fop, T. B. Murphy, and A. E. Raftery (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal 8*(1), 289–317. 22, 26, 29, 30, 35

Thompson, T. J., P. J. Smith, and J. P. Boyle (1998). Finite mixture models with concomitant information: assessing diagnostic criteria for diabetes. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 47*(3), 393–404. 54

Wang, P., M. L. Puterman, and I. Cockburn (1998). Analysis of patent data – a mixed-Poisson regression-model approach. *Journal of Business & Economic Statistics 16*(1), 27–41. 54

Wang, P., M. L. Puterman, I. Cockburn, and N. Le (1996). Mixed Poisson regression models with covariate dependent rates. *Biometrics 52*(2), 381–400. 54

Wedel, M. and W. A. Kamakura (2012). *Market Segmentation: Conceptual and Methodological Foundations*. International Series in Quantitative Marketing. US: Springer. 54

Young, D. S. and D. R. Hunter (2010). Mixtures of regressions with predictor-dependent mixing proportions. *Computational Statistics & Data Analysis 54*(10), 2253–2266. 55

Zellner, A. (1962). An efficient method of estimating seemingly unrelated regression equations and tests for aggregation bias. *Journal of the American Statistical Association 57*(298), 348–368. 33

2.A Appendix 1

CO_2 Data: Code Examples

Code to reproduce both the exhaustive (Listing 2.A.1) and greedy forward stepwise (Listing 2.A.2) searches for the CO_2 data described in Section 2.5.1, using the `MoEClust` R package (Murphy and Murphy, 2019), is provided below. The code in Listing 2.A.1 can be used to reproduce the results in Table 2.2.

Listing 2.A.1: Exhaustive search R code for the CO_2 data.

```
library(MoEClust)
data(CO2data)
CO2   <- CO2data$CO2
GNP   <- CO2data$GNP

# Fit models under the 6 special cases of the MoE framework
m1    <- MoE_clust(CO2, G=1:9)
m2    <- MoE_clust(CO2, G=2:9, gating=~GNP)
m3    <- MoE_clust(CO2, G=1:9, expert=~GNP)
m4    <- MoE_clust(CO2, G=2:9, gating=~GNP, expert=~GNP)
m5    <- MoE_clust(CO2, G=2:9, equalPro=TRUE)
m6    <- MoE_clust(CO2, G=2:9, expert=~GNP, equalPro=TRUE)

# Collate results and rank (by BIC) only the 6 optimal models
res   <- list(m1=m1, m2=m2, m3=m3, m4=m4, m5=m5, m6=m6)
(comp <- MoE_compare(res, optimal.only=TRUE))
```

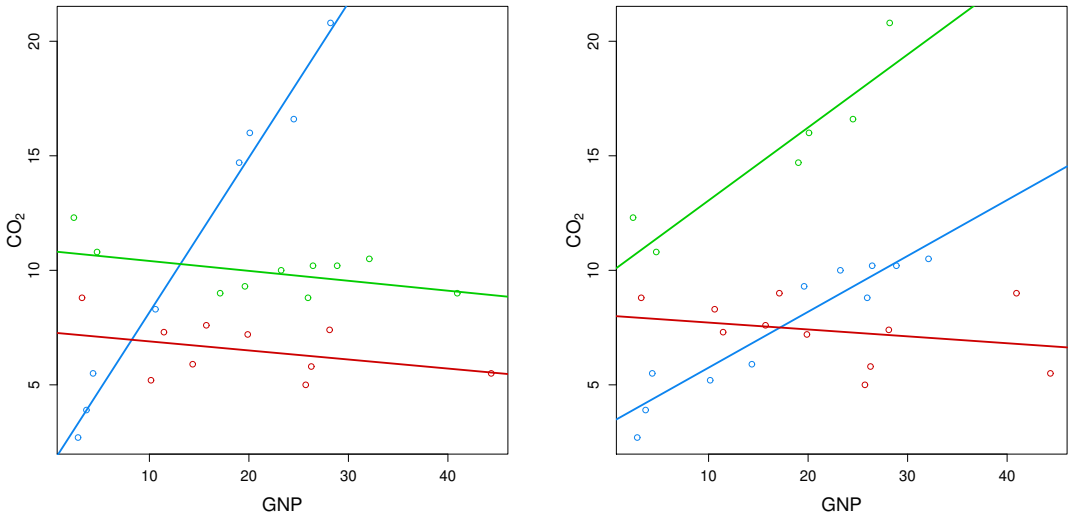
Listing 2.A.2: Stepwise search R code for the CO_2 data.

```
 library(MoEClust)
 data(CO2data)
 CO2   <- CO2data$CO2
 GNP   <- CO2data$GNP

# Conduct a stepwise search
 (mod1 <- MoE_stepwise(CO2, GNP))

# Conduct a stepwise search for models with a noise component
 (mod2 <- MoE_stepwise(CO2, GNP, noise=TRUE))

# Compare both sets of results to choose the optimal model
 (best <- MoE_compare(mod1, mod2, optimal.only=TRUE)$optimal)
```

2.B Appendix 2

CO_2 Data: EM Initialisation

The regression lines for the optimal $G = 3$ equal mixing proportion expert network MoEClust model with equal component variances and the explanatory variable 'GNP' fitted to the CO_2 data with and without the initial partition being passed through Algorithm 1 are shown in Figure 2.B.1. A BIC value of -155.20 is achieved after 21 EM iterations with 6 iterations of our proposed initialisation strategy compared to a value of -161.06 in 28 EM iterations without. While the models differ only in terms of the initialisation strategy employed, Table 2.2 shows that the model would not have been identified as optimal according to the BIC criterion had Algorithm 1 not been used. The superior solution in Figure 2.B.1a has one cluster with a steep slope and two clusters with near-zero slopes but different intercepts.



(a) With Algorithm 1 invoked for initialisation, achieving a BIC value of -155.20.

(b) Without Algorithm 1 invoked for initialisation, achieving a BIC value of -161.06.

Figure 2.B.1: Scatter plots of GNP against CO_2 emissions for $n = 28$ countries showing three coloured linear regression components from the optimal MoEClust model, with equal variances and mixing proportions, with (a) and without (b) the initialisation strategy described in Algorithm 1 invoked.

2.C Appendix 3

AIS Data: Stepwise Model Search

For the AIS data, Table 2.C.1 gives the results of the greedy forward stepwise model selection strategy described in Algorithm 2, showing the action yielding the best improvement in terms of BIC for each step. This forward search is less computationally onerous than its equivalent in the backwards direction. A 2-component EVE *equal mixing proportion expert network MoE* model is chosen, in which the mixing proportions are constrained to be equal and sex enters the expert network. This same model was identified after an exhaustive search over a range of G values, the full range of GPCM covariance constraints, and every possible combination of the BMI and sex covariates in the gating and expert networks (see Table 2.5). Note, however, that the remaining covariates in Table 2.4 are also considered for inclusion here.

To give consideration to outlying observations departing from the prevailing pattern of Gaussianity, a separate stepwise search is conducted, starting from a $G = 0$ noise-only model, with all candidate models having an additional noise component. Thus, a distinction is made between the model found by following the steps shown in Table 2.C.1 with $G = 2$ EVE Gaussian components, and the model found by the second stepwise search described in Table 2.C.2 with three, of which two are EEE Gaussian and one is uniform. Ultimately, the model with the noise component identified in Table 2.C.2 is chosen, based on its superior BIC. Aside from the noise component, it similarly includes 'sex' in the expert network, but differs in its treatment of the gating network and the GPCM constraints employed for the Gaussian clusters. It is a *full MoE* model where the Gaussian clusters have equal volume, shape, and orientation, the expert network includes the covariate 'sex', and the both 'SSF' and 'Ht' influence the probability of belonging to the Gaussian clusters but not the additional noise component, as per (2.9).

Table 2.C.1: Results of the forward stepwise model selection algorithm applied to the AIS data where candidate models do not include a noise component. All covariates in Table 2.4 are considered for inclusion in both parts of the model. The optimal action and associated BIC value is detailed for each step. The resulting models are described in terms of the number of Gaussian components G, the GPCM constraints used, and the treatment of the gating and expert networks.

Step	Optimal Action	G	GPCM	Gating	Expert	BIC
1	—	1	EEE	—		−4202.79
2	Add explanatory variable (Expert)	1	EEE	—	sex	−4050.64
3	Add component and constrain mixing proportions	2	EVE	Equal	sex	−4010.14
4	Stop	2	EVE	Equal	sex	−4010.14

Table 2.C.2: Results of the forward stepwise model selection algorithm applied to the AIS data where all candidate models explicitly include a noise component. All covariates in Table 2.4 are considered for inclusion in both parts of the model. The optimal action and associated BIC value is detailed for each step. The resulting models are described in terms of the number of Gaussian (i.e. non-noise) components G, the GPCM constraints used, and the treatment of the gating and expert networks. When gating concomitants are included, the chosen models here correspond to the NGN model in (2.9). Thus, the noise component's mixing weight is constant and independent of the included concomitants.

Step	Optimal Action	G	GPCM	Gating	Expert	BIC
1	—	0	—	—	—	−4869.82
2	Add component	1	EEE			−4149.46
3	Add explanatory variable (Expert)	1	EEE		sex	−4013.55
4	Add component	2	EVE		sex	−3992.81
5	Add concomitant (Gating)	2	EVE	NGN: SSF	sex	−3990.09
6	Add concomitant (Gating)	2	EEE	NGN: SSF, Ht	sex	−3989.83
7	Stop	2	EEE	NGN: SSF, Ht	sex	−3989.83

2.D Appendix 4

Prediction and Validation for MoEClust Models

In addition to their demonstrated use for clustering, MoEClust models can also be used to make point estimate predictions, which we consider here in the context of a model fitted to a training data set used for predicting newly observed test data. This can be done if both new covariates $\mathbf{x}_i^\star$ and new response data $\mathbf{y}_i^\star$ are observed, or if only $\mathbf{x}_i^\star$ are observed. Thus, for instance, predictions for new athletes using the optimal MoEClust model fit to the AIS data can be made if their sex and their Ht and SSF measurements are known, with or without also having their hematological measurements. Point estimate predictions can be made of the cluster membership probabilities $\widehat{z}_{ig}^\star$ (and hence the MAP classification, here denoted by $\widehat{c}_i^\star$) as well as the response variables $\widehat{\mathbf{y}}_i^\star$. Typically, predicting $\widehat{z}_{ig}^\star$ and $\widehat{c}_i^\star$ will be of most interest when $\mathbf{x}_i^\star$ and $\mathbf{y}_i^\star$ are observed, while predicting $\widehat{\mathbf{y}}_i^\star$ will be of interest when only $\mathbf{x}_i^\star$ are observed. However, we caution that a model considered optimal from a clustering point of view may not necessarily be optimal from a prediction point of view.

Predicting $\widehat{z}_{ig}^\star$ when both $\mathbf{x}_i^\star$ and $\mathbf{y}_i^\star$ are observed amounts to an E-step:

$$\widehat{z}_{ig}^\star = \frac{\widehat{\tau}_g\left(\mathbf{x}_i^{\star(G)}\right)\phi\left(\mathbf{y}_i^\star \,|\, \widehat{\boldsymbol{\theta}}_g\left(\mathbf{x}_i^{\star(E)}\right) = \left\{\widetilde{\mathbf{x}}_i^{\star(E)}\widehat{\boldsymbol{\gamma}}_g, \widehat{\boldsymbol{\Sigma}}_g\right\}\right)}{\sum_{h=1}^G \widehat{\tau}_h\left(\mathbf{x}_i^{\star(G)}\right)\phi\left(\mathbf{y}_i^\star \,|\, \widehat{\boldsymbol{\theta}}_h\left(\mathbf{x}_i^{\star(E)}\right) = \left\{\widetilde{\mathbf{x}}_i^{\star(E)}\widehat{\boldsymbol{\gamma}}_h, \widehat{\boldsymbol{\Sigma}}_h\right\}\right)},$$

whereas $\widehat{z}_{ig}^\star = \widehat{\tau}_g\big(\mathbf{x}_i^{\star(G)}\big)$ when only $\mathbf{x}_i^\star$ are observed. Both expressions are appropriately modified when the GN or NGN settings are adopted in the presence of a noise component and gating concomitants. Similarly, the noise component, if any, can be accounted for in subsequently predicting $\widehat{\mathbf{y}}_i^\star$, as per (2.11), via

$$\widehat{\mathbf{y}}_i^\star = \sum_{g=1}^G \widehat{z}_{ig}^\star\big(\widetilde{\mathbf{x}}_i^{\star(E)}\widehat{\boldsymbol{\gamma}}_g\big) + \widehat{z}_{i0}\overline{\mathcal{V}}, \tag{2.12}$$

where $\overline{\mathcal{V}}$ is the p-dimensional centroid of the region used to estimate the hypervolume V. Here, V is estimated from the smallest hyperrectangle enclosing the data, though we note that $\overline{\mathcal{V}}$ can also be computed for the convex hull or ellipsoid hull, should they be instead used to estimate the hypervolume. Clearly, for models where $\mathbf{x}_i^{(E)} = \varnothing$, $\widehat{\mathbf{y}}_i^\star$ is simply given by the weighted mean of the component means.

Alternatively, the noise component can be discarded by removing the column of $\widehat{\mathbf{Z}}^{\star}$ corresponding to the noise component, if any, and renormalising its rows prior to computing $\widehat{\mathbf{y}}_i^{\star} = \sum_{g=1}^{G} \widehat{z}_{ig}^{\star}\left(\widetilde{\mathbf{x}}_i^{\star(E)}\widehat{\boldsymbol{\gamma}}_g\right)$. We leave this decision as a choice for the interested researcher, with two caveats. Firstly, $\overline{\boldsymbol{\mathcal{V}}}$ is not defined if V is specified as independent tuning parameter, which it must be when $n \leq p$. Secondly, we note that unseen data $\mathbf{y}_i^{\star}$ may lie outside the region used to define the hypervolume and thus lie outside the support of the uniform noise component.

The approach for predicting $\widehat{\mathbf{y}}_i^{\star}$ in (2.12) can be interpreted as an aggregation of the predictions in the component-specific expert networks (accounting also for the centroid of the noise component, if any). While this approach is appealing in that the estimated 'soft' cluster membership probabilities $\widehat{z}_{ig}^{\star}$ are utilised, it can be criticised by virtue of the information lost in reducing the heterogeneous regression functions to the one aggregated function in (2.12). This limitation is pronounced further by the fact that $\widehat{z}_{ig}^{\star} = \widehat{\tau}_g\left(\mathbf{x}_i^{\star(G)}\right)$ amounts only to a prediction of the prior mixing weights when only $\widehat{\mathbf{x}}_i^{\star}$ are observed, though we note that the aggregated regression function in this instance is a curve (hypersurface) rather than a line (hyperplane) for MoEClust models with gating network concomitants.

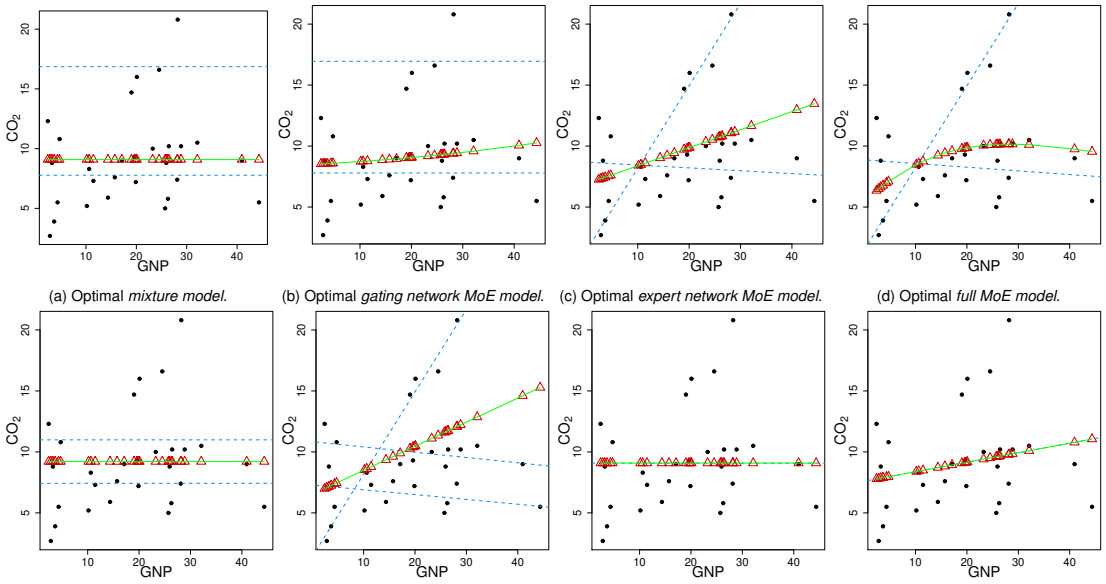
An alternative is to predict $\widehat{\mathbf{y}}_i^{\star}$ using the individual expert network regression of the component to which the new observation is most probably assigned, via

$$\widehat{c}_i^{\star} = \underset{g\in\{0,\ldots,G\}}{\arg\max}\ \left(\widehat{z}_{ig}^{\star}\right), \qquad \widehat{\mathbf{y}}_i^{\star} = \begin{cases} \widetilde{\mathbf{x}}_i^{\star(E)}\widehat{\boldsymbol{\gamma}}_{\widehat{c}_i^{\star}} & \textit{iff}\ \ \widehat{c}_i^{\star} \in \{1,\ldots,G\}, \\ \overline{\boldsymbol{\mathcal{V}}} & \textit{iff}\ \ \widehat{c}_i^{\star} = 0. \end{cases} \tag{2.13}$$

As above, the noise component, if any, can be accounted for either by explicitly including $\overline{\boldsymbol{\mathcal{V}}}$ or by discarding the column of $\widehat{\mathbf{Z}}$ corresponding to the noise component and renormalising its rows prior to computing $\widehat{c}_i^{\star}$, again with the same caveats. In any case, this approach is not without limitations either when only $\mathbf{x}_i^{\star}$ are observed. In particular, $\widehat{c}_i^{\star}$ cannot be determined for models with equal mixing proportions, and $\widehat{c}_i^{\star} = c^{\star}\ \forall\ i$ when the mixing proportions are unequal but not dependent on covariates, such that $\widetilde{\mathbf{y}}_i^{\star}$ is always predicted only by the expert network regression of the largest component. As a result, the aggregation in (2.12) remains the recommended approach. Moreover, when only $\mathbf{x}_i^{\star}$ are observed, the approach in (2.13) is discouraged for MoEClust models which assume assignment independence, i.e. all model types without gating covariates such that $\widehat{c}_i^{\star}$ does not depend on $\mathbf{x}_i^{\star(G)}$.

In order to investigate MoEClust models from a predictive point of view, we turn to the CO_2 data, for which the optimal model identified did not include a noise component. We focus on predicting $\widehat{\mathbf{y}}_i^\star$ when $\widehat{\mathbf{x}}_i$ are observed, rather than predicting $\widehat{z}_{ig}^\star$ and/or $\widehat{c}_i^\star$ when both $\mathbf{y}_i^\star$ and $\mathbf{x}_i^\star$ are observed. In so doing, we use the aggregated regression function in (2.12), with $\widehat{z}_{ig}^\star$ predicted from the mixing proportions, as appropriate. In order to explore whether the optimal model identified in Section 2.5.1 is indeed optimal from a predictive as well as clustering point of view, all 6 models in Table 2.2 (i.e. the optimal models under each special case of the MoE framework) are investigated. Two 1-component models are also considered as comparators — one with no covariates and a linear regression of CO_2 on GNP.

Figure 2.D.1 shows, for all 8 considered models, the values of $\widehat{\mathbf{y}}_i^\star$ predicted via (2.12) using only $\mathbf{x}_i^\star$. Here, the observed data is used as the 'new' data. Component-specific regression lines and corresponding aggregated regression functions are also shown. Notably, the aggregated functions in Figure 2.D.1b and Figure 2.D.1d are curves rather than lines, owing to the inclusion of a covariate in the respective gating networks. Recall that the model shown in Figure 2.D.1f was chosen by BIC.



(a) Optimal *mixture model.* (b) Optimal *gating network MoE model.* (c) Optimal *expert network MoE model.* (d) Optimal *full MoE model.*

(e) Optimal *equal mixing proportion mixture model.* (f) Optimal *equal mixing proportion expert network MoE model.* (g) Optimal $G = 1$ *mixture model.* (h) Optimal $G = 1$ *expert network MoE model.*

Figure 2.D.1: Predicted values $\widehat{\mathbf{y}}_i^\star$ (red triangles), component-specific regression lines (dotted blue), and aggregated regression functions (solid green, where applicable) for a number of MoEClust models fit to the CO_2 data — each one optimal with respect to a (labelled) special case of the MoE framework — with the 'new' data treated as the fitted data (shown, for reference, as black points).

In order to quantify the goodness-of-fit of the 8 considered models, we use the coefficient of determination and other measures recently proposed for mixtures of regressions by Ingrassia and Punzo (2019), which rely on a three-term decomposition of the total sum of squares (TSS). In particular, the TSS is decomposed into the (soft) between-cluster sum of squares (BSS), which measures how well separated the clusters are, and the (soft) within-cluster sum of squares (WSS), which measures how close observations in a cluster are to the regression line of that cluster. Here, 'soft' means that the quantities are weighted by the estimated cluster-membership probabilities. The WSS is further decomposed into EWSS, the (soft) within-cluster sum of squares explained by the model (by virtue of the inclusion of expert network covariates), and RWSS, the (soft) within-cluster residual sum of squares, with the EWSS and RWSS are obtained by summing the cluster-specific (soft) ESS_g and RSS_g values. By definition, BSS is not defined for 1-component models and EWSS is not defined for models without expert network covariates.

By dividing BSS, EWSS, and RWSS by TSS, Ingrassia and Punzo (2019) also obtain the normalised summary measures NEWSS, NBSS, and NRWSS. These measures are hence interpreted as *proportions* of the total variation of the response variable accounted for by the main parts of the model. They can be used to evaluate the fitted model in a number of ways. Firstly, the quantity $\text{NESS} = \text{NBSS} + \text{NEWSS} = 1 - \text{NRWSS}$, which represents the proportion of the TSS explained by the fitted model, indicates a well-fitting model for values close to 1. Secondly, Ingrassia and Punzo (2019) propose visualising the triplet (NEWSS, NBSS, NRWSS) using a ternary diagram. The corresponding points are shown on such a diagram for all 8 models under consideration in Figure 2.D.2. The point labelled 'F', corresponding to the 3-component *equal mixing proportion expert network MoE model* (with equal variances) identified as optimal according to the BIC criterion, achieves the highest NESS (0.94). While the optimal mixture model ('A') and optimal gating network MoE model ('B') both achieve a superior NBSS, their performance is hampered by their vanishing NEWSS values. This confirms that the inclusion of GNP in the expert network improves the model fit. Conversely, linear regression ('H') achieves a much lower NEWSS despite the inclusion of the covariate GNP. This confirms that mixtures with $G \geq 2$ achieve superior fits.

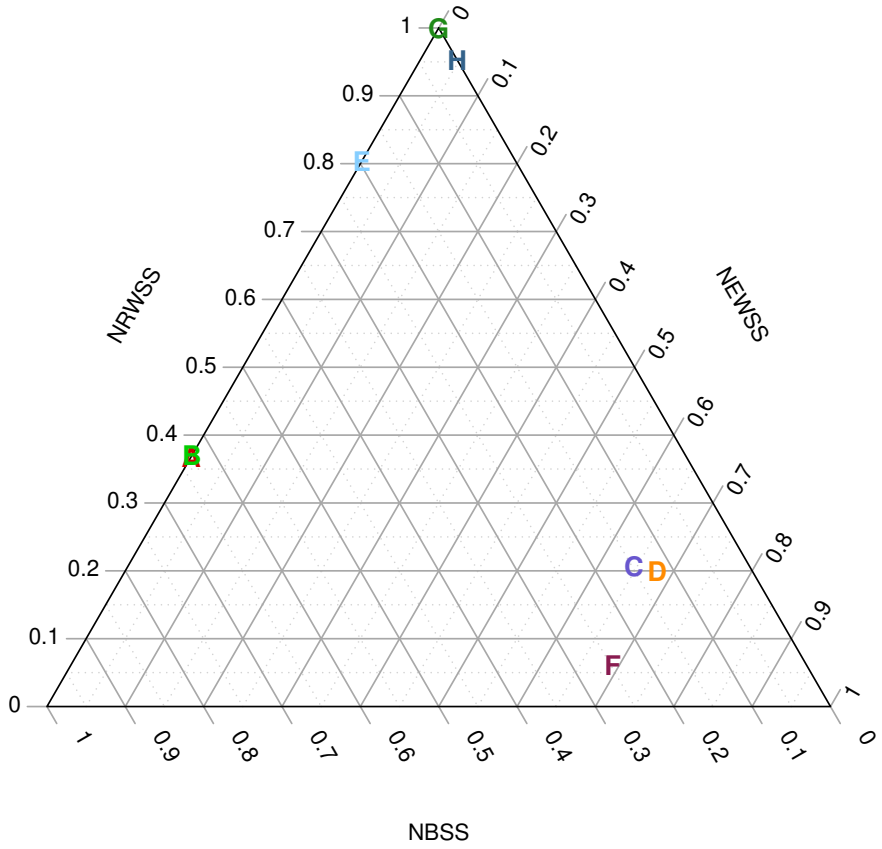


Figure 2.D.2: Ternary diagram of the points (NBSS, NEWSS, NRWSS) for the 8 models fitted to the CO_2 data, with the letters corresponding to the labels of the subfigures in Figure 2.D.1.

Much like the popular coefficient of determination for linear regression models, Ingrassia and Punzo (2019) also define local and global coefficients of determination for finite mixtures of regressions (henceforth referred to as R^2_g and R^2, respectively). The former, given by $R^2_g = \text{ESS}_g/\text{SS}_g$, where $\text{SS}_g = \text{ESS}_g + \text{RSS}_g$, can be interpreted as the proportion of response variation in the g-th cluster explained by the expert network covariates. The latter, given by $R^2 = \text{EWSS}/\text{WSS}$, can be interpreted as the proportion of the within-cluster response variation explained by the fitted model. This global coefficient of determination can be seen as a weighted average of the local coefficients $R^2_1, \ldots, R^2_G$, with weights $\text{SS}_1/\text{WSS}, \ldots, \text{SS}_G/\text{WSS}$ being the proportion of the within-cluster sum of squares due to each cluster.

Figure 2.D.1f and Figure 2.D.1c suggest that the 3-component model chosen by BIC ('F') differs from the optimal $G = 2$ expert network MoE model ('C') in that the component with near-zero slope in model C is split into two components with near-zero slopes but different intercepts under model F. Furthermore, Hungary and South Korea are assigned (according to the MAP classification) to the component with the steep slope under model F while they are assigned to the component with a flat slope under model C. The R^2 for model C is 0.76 while the R^2 for model F is 0.92. The local coefficients of determination and their weights help to explain the superior fit of model F. Under model F, $R_1^2 = 0.98$, $R_2^2 = 0.13$, and $R_3^2 = 0.21$, with weights $\text{SS}_1 = 0.92$, $\text{SS}_2 = 0.04$, and $\text{SS}_3 = 0.04$. Under model C, $R_1^2 = 0.98$ and $R_2^2 = 0.02$, with $\text{SS}_1 = 0.77$ and $\text{SS}_2 = 0.23$. Clearly, therefore, both models fit well to the component with the steep slope, but model F achieves a better fit by virtue of the higher weight attached to this component. A similar conclusion is drawn when comparing model F against the full MoE model ('D'), for which we report, for completeness, $R^2 = 0.77$, $R_1^2 = 0.98$, $R_2^2 = 0.03$, $\text{SS}_1^2 = 0.78$, and $\text{SS}_2^2 = 0.22$.

Overall, we conclude that all special cases of the MoE framework are useful from a clustering point of view, only the gating network MoE model and full MoE model are useful from an out-of-sample prediction point of view, and the expert network MoE model, full MoE model, and equal mixing proportion expert network MoE model are useful from a general explanatory point of view. However, we caution against using these validation measures as model selection tools.

2.E Appendix 5

Distinguishing Responses and Covariates

By design — by virtue of MoEClust being a family of conditional mixture models — it is assumed that the designation of which variables are responses and which variables are to be treated as covariates is always known for MoEClust models, in the sense that the conditioning is guided by the context of the application under study and the related research question of interest. Thus, there is no need to choose the subset of covariates, only to choose — among the known subset of covariates — which further subsets, if any, belong in the gating or expert network.

Hence, two related issues are explored within this Appendix. Firstly, situations in which it is *not* known which variables should be responses and which should be covariates are examined. Secondly, with regard to the proposed stepwise procedure (Algorithm 2) for selecting the informative covariates — and identifying which part of the model they should enter — simulation studies are provided to assess the performance of this procedure, from the point of view of overfitting. Both issues are illustrated through an application to data gathered on a cohort of 544 Kenyan donkeys (Milner and Rougier, 2014). The variables recorded in this study are of mixed type and are detailed in Table 2.E.1.

Table 2.E.1: Variables recorded in the Kenyan donkeys data set.

Variable	Description
Girth	continuous (cm)
Height	continuous (cm)
Length	continuous (cm)
Weight	continuous (kg)
Age[♯]	an ordinal factor with levels:
	< 2, 2–5, 5–10, 10–15, 15–20, > 20 (years)
BCS[♯]	body condition score — an ordinal factor with levels:
	from 1 (emaciated) through 3 (healthy) to 5 (obese) in steps of 0.5
Sex	a factor with levels:
	female, gelding, stallion

[♯] Following Milner and Rougier (2014), three unrepresentative donkeys were excluded from the analysis to leave $n = 541$ donkeys; one was a baby, one had a BCS of 1, and one had a BCS of 4.5. No other donkeys had a BCS of 1, 4.5, or 5, so these levels were dropped from this variable.

Comparing Joint and Conditional Mixture Models

When it is not known which variables should be treated as response variables and which should be treated as covariates, we suggest clustering all variables *jointly*, rather than employing a *conditional* MoEClust mixture model. Under the conditional MoEClust model, all response variables are assumed to be continuous, while the covariates in both networks are allowed to be of mixed type. However, under the joint mixture model, which groups all variables into the set of response variables, MoEClust models without dependence on covariates of any kind (i.e. the mixture model in Figure 2.1) can only be fitted in cases where all variables are continuous. To jointly cluster data of mixed type, including continuous, binary, ordinal, and nominal variables, we propose to use the model-based approach to clustering mixed-type data introduced in McParland and Gormley (2016), henceforth referred to as `clustMD` models after the name of the associated R package.

In addition, we acknowledge that CWMs — here fitted using the `flexCWM` R package (Mazza et al., 2018) — can be used in cases where it is of interest to jointly model both $\mathbf{y}_i$ and $\mathbf{x}_i$ via $f(\mathbf{y}_i \mid \mathbf{x}_i, z_{ig} = 1) f(\mathbf{x}_i \mid z_{ig} = 1)$, even if this still problematically reintroduces the need to distinguish among the overall set of variables between the response(s) and the covariate(s). For this class of models, in the application which follows, a Gaussian distribution with unequal variances across components is assumed for the response variable and a (multivariate) Gaussian distribution is assumed for the continuous covariates. While it is also possible to exploit the local independence assumption in order to model the marginal distribution of categorical covariates under CWMs, such models are not considered here for the sake of brevity.

In order to facilitate a fair comparison against `clustMD` models and CWMs, MoEClust models with a noise component are not considered herein. Neither are models with equal mixing proportions. All three model classes allow for some degree of parsimony in the component covariance matrices: MoEClust models allow GPCM covariance structures in the distribution of the response variables, CWMs allow GPCM covariance structures in the distribution of the continuous covariates, and `clustMD` models allow the 6 spherical and diagonal GPCM covariance structures in Table 2.1 as well as a block diagonal (BD) structure for the underlying

latent variables. As treating the categorical variables 'Age' and 'BCS' as ordinal rather than nominal variables simplifies the fitting of `clustMD` models (McParland and Gormley, 2016), these variables are henceforth coded in that way.

In comparing the three model classes, we note the assertion in Milner and Rougier (2014) that a donkey is effectively an elliptical cylinder with appendages, such that one might expect its weight to be approximately proportional to the product of its length and its squared girth, with its less cylindrical aspects possibly accommodated by including its height as an additional predictor (as well as the other categorical factors). Hence, we pre-process the data by taking the natural log of the Girth and Length variables. Indeed, the motivation behind this study was the difficulty in estimating the weight of a donkey, relative to the ease with which physical measurements of its girth, height, and length can be obtained.

Therefore, conditional MoEClust mixture models are investigated in Table 2.E.2 with weight as the univariate response variable ('A' and 'B'). Secondly, artificial situations in which one is not guided by this research question and instead proceeds to cluster all variables jointly are reflected by the inclusion of a `clustMD` model ('E') and a CWM ('F') in the comparison in Table 2.E.2. Thirdly, MoEClust models where are all continuous variables are treated as the response are also estimated ('C' and 'D'). Thus, different scenarios are modelled whereby no covariates, the subset of continuous covariates only, or all covariates are considered (either as covariates or as part of the responses, as appropriate to the given model class). Finally, Table 2.E.2 also examines the effects of interchanging the role of the (continuous) covariates and the Weight response variable ('G' and 'H').

For MoEClust models with covariates ('B', 'D', and 'H'), a stepwise search using Algorithm 2 and the BIC is conducted to select the optimal model. Otherwise, exhaustive searches over $G \in \{1, \ldots, 9\}$ and all allowable covariance structures are conducted, with the optimal model chosen by BIC reported in each case. However, the three model classes are not all comparable in terms of BIC (Ingrassia and Punzo, 2019); the likelihood function for MoEClust models is a product of conditional probabilities while the likelihood for `clustMD` and `flexCWM` models is a product of joint probabilities. Moreover, CWMs with different subsets of covariates cannot be compared in terms of BIC either, for similar reasons. Hence, covariate selection was not considered for this class of models.

While `clustMD` models with gating concomitants are feasible — despite complicating the designation of response(s) and covariate(s) again — such models are excluded from the comparison as they are not currently implemented in the `clustMD` package. Similarly, multivariate response CWMs are not implemented in the `flexCWM` package and are hence also excluded from the comparison. Finally, we note that all other function arguments were set to their default values when using these two packages.

Table 2.E.2: Results of a comparison between conditional and joint mixture models for the Kenyan donkeys data set, giving — for each model — a name, the R package used, the designation of response(s) and covariate(s), the optimal number of components and covariance decomposition, the included gating and expert network covariates for `MoEClust` models, and the BIC. Horizontal lines separate model classes which are not comparable in terms of BIC, though the BIC for the optimal model within each class is highlighted. A number of abbreviations are used: ALL (all variables), CTS1 (all continuous variables excluding 'Weight'), CTS2 (CTS1 and Weight), CAT (all categorical variables), COV1 (CAT and CTS2), and COV2 (CAT and 'Weight').

Name	Package	Response(s)	Covariate(s)	G	Covariance	Gating	Expert	BIC
A	`MoEClust`	Weight		2	E	—	—	−5039.34
B	`MoEClust`	Weight	COV1	1	E		BCS, CTS1	**−3892.41**
C	`MoEClust`	CTS2		2	EEV	—	—	−3502.03
D	`MoEClust`	CTS2	CAT	1	EEE		Age, BCS	**−3147.32**
E	`clustMD`	ALL	—	2	BD	—	—	**−7647.63**
F	`flexCWM`	Weight	CTS1	2	VEE[♮]	—	—	**−3509.07**
G	`MoEClust`	CTS1		5	VII	—	—	−4564.73
H	`MoEClust`	CTS1	COV2	5	EII		Weight	**−3916.63**

[♮] Note that the stated covariance decomposition for the `flexCWM` model relates to the Gaussian distribution of the covariates only; the univariate Gaussian distribution for the response has unequal variance across components.

Several interesting conclusions can be gleaned from Table 2.E.2. Firstly, note that model B, which includes BCS, Girth, Height, and Length as expert network covariates with Weight as the univariate response amounts to a simple linear regression, while model E, which jointly clusters all variables, and model C, which jointly clusters all continuous variables only, address incomparable questions and both find $G = 2$ clusters. Hence, the lack of conditioning on covariates in models A, D, and E is shown to introduce an extra component relative to models B and D. While models C and E reflect a situation in which the designation of responses and covariates is unknown, clustering all variables jointly still produces sensible results,

albeit addressing a different aim. On the other hand, models G and H, which one might fit if one was not guided by the interest in treating the Weight variable as the response, appear to overestimate the number of components greatly. Notably, the responses in models G and H are treated as covariates under model F. The fact that models with $G > 1$ are identified in each case implies that group information within the set of *random* covariates is captured by CWMs. Overall, this suggests that a conditional mixture model should be used when a specific conditioning is required, while clarifying the recommendation that one should instead cluster all variables jointly when the designation of response(s) and covariate(s) is unknown.

Finally, we note that broadly similar conclusions can be drawn when Length and Girth are not log-transformed, where relevant, with the exceptions of models E, G, and H, for which models with 1, 2, and 1 components are identified, respectively. Given that the decision to pre-process these variables was motivated by the consideration of Weight as the univariate response, this arguably represents a truer approximation of a situation in which one is agnostic as to the designation of response(s) and covariate(s).

Simulation Studies to Assess Overfitting

Henceforth, only MoEClust models for the Kenyan donkeys data set which treat Weight as the response variable are considered. Attention turns to assessing the performance of the novel stepwise selection procedure (Algorithm 2) for identifying informative covariates, as well the part of the model to which they should belong.

To this end, high-dimensional noise is appended to the set of potential covariates used as input to Algorithm 2, in the form of 25 variables containing no clustering information drawn from a $N(\mu_0 = 0, \sigma_0 = 1)$ distribution (see Table 2.E.3). Here, the log-transformed Length and Girth variables are used. Furthermore, a scenario in which only the noisy variables are considered is also assessed (see Table 2.E.4). The rationale for these simulations is to assess situations in which there is an imbalance between the number of responses and covariates, in order to examine whether the approach can lead to overfitting. In contrast to the results shown in Table 2.E.2, models with equal mixing proportions are explored in these stepwise searches, though models with a noise component are not.

Reasssuringly, the algorithm never elects to include a noisy covariate in either experiment. Notably, the final model in Table 2.E.3 is identical to model B in Table 2.E.2, while the final model in Table 2.E.4 differs from model A only in that the mixing proportions are constrained to be equal. This holds for the results obtained using other values of μ_0 and σ_0, which are not shown here for the sake of brevity.

Table 2.E.3: Results of the forward stepwise model search for the Kenyan donkeys data with uninformative covariates appended to all variables other than Weight in Table 2.E.1 used as input.

Step	Optimal Action	G	GPCM	Gating	Expert	BIC
1	—	1	E	—		−5062.91
2	Add explanatory variable (Expert)	1	E	—	Girth	−4161.28
3	Add explanatory variable (Expert)	1	E	—	Girth, Length	−3993.71
4	Add explanatory variable (Expert)	1	E	—	BCS, Girth, Length	−3924.93
5	Add explanatory variable (Expert)	1	E	—	BCS, Girth, Height, Length	−3892.41
6	Stop	1	E	—	BCS, Girth, Height, Length	−3892.41

Table 2.E.4: Results of the forward stepwise model search for the Kenyan donkeys data with only uninformative covariates used as input.

Step	Optimal Action	G	GPCM	Gating	Expert	BIC
1	—	1	E	—		−5062.91
2	Add component	2	V	Equal		−5038.56
3	Stop	2	V	Equal		−5038.56

2.F Appendix 6

MoEClust R Package Vignette

This appendix presents a reproduction of the package vignette[3] of the associated R package `MoEClust` for implementation of the proposed method. Notably, some additional plot types are presented for the CO_2 and AIS data sets.

[3] cran.r-project.org/web/packages/MoEClust/vignettes/MoEClust.html

MoEClust: Gaussian Parsimonious Clustering Models with Gating and Expert Network Covariates and a Noise Component

Keefe Murphy

Introduction

MoEClust is an R package which fits finite Gaussian Mixtures of Experts models using a range of parsimonious covariance parameterisations via the EM/CEM algorithm, i.e. allows incorporation of covariates into the mixing proportions and/or Gaussian densities of finite Gaussian mixture models under the various parsimonious covariance parameterisations in the GPCM family (e.g. **mclust**). These models were introduced by Murphy and Murphy (2019). The package also facilitates the inclusion of an additional noise component, and allows visualisation of Gaussian mixture of experts models with parsimonious covariance parameterisations using generalised pairs plots.

The most important function in the **MoEClust** package is: `MoE_clust`, for fitting the model via the EM/CEM algorithm with gating and/or expert network covariates, supplied via formula interfaces. `MoE_compare` is provided for conducting model selection between different results from `MoE_clust` using different covariate combinations &/or initialisation strategies, etc.

`MoE_stepwise` is provided for conducting a greedy forward stepwise search to identify the optimal model in terms of the number of components, GPCM covariance type, and the subsets of gating/expert network covariates.

`MoE_control` allows supplying additional arguments to `MoE_clust` and `MoE_stepwise` which govern, among other things, controls on the inclusion of an additional noise component and controls on the initialisation of the allocations for the EM/CEM algorithm.

A dedicated plotting function exists for visualising the results using generalised pairs plots, for examining the gating network, and/or log-likelihood, and/or clustering uncertainties, and/or graphing model selection criteria values. The generalised pairs plots (`MoE_gpairs`) visualise all pairwise relationships between

clustered response variables and associated continuous, categorical, and/or ordinal covariates in the gating &/or expert networks, coloured according to the MAP classification, and also give the marginal distributions of each variable (incl. the covariates) along the diagonal.

An `as.Mclust` method is provided to coerce the output of class `"MoEClust"` from `MoE_clust` to the `"Mclust"` class, to facilitate use of plotting and other functions for the `"Mclust"` class within the **mclust** package. As per **mclust**, **MoEClust** also facilitates modelling with an additional noise component (with or without the mixing proportion for the noise component depending on covariates). Finally, a `predict` method is provided for predicting the fitted response and probability of cluster membership (and by extension the MAP classification) for new data, in the form of new covariates and new response data, or new covariates only.

Other functions also exist, e.g. `MoE_crit`, `MoE_dens`, `MoE_estep`, and `aitken`, which are all used within `MoE_clust` but are nonetheless made available for standalone use. The package also contains two data sets: `ais` and `CO2data`.

If you find bugs or want to suggest new features please visit the **MoEClust** GitHub issues page.

This vignette aims to demonstrate the **MoEClust** models via application to well-known univariate and multivariate data sets provided with the package.

Installing MoEClust

MoEClust will run in Windows, Mac OS X or Linux. To install it you first need to install R. Installing Rstudio as a nice desktop environment for using R is also recommended.

Once in R you can type at the R command prompt:

```
install.packages('devtools')
devtools::install_github('Keefe-Murphy/MoEClust')
```

to install the latest development version of the package from the **MoEClust** GitHub page.

To instead install the latest stable official release of the package from CRAN go to R and type:

```
install.packages('MoEClust')
```

In either case, if you then type:

```
library(MoEClust)
```

it will load in all the **MoEClust** functions.

The GitHub version contains a few more features but some of these may not yet be fully tested, and occasionally this version might be liable to break when it is in the process of being updated.

CO2 Data

Load the CO2 data.

```
data(CO2data)
CO2   <- CO2data$CO2; GNP <- CO2data$GNP
```

Fit various MoEClust mixture models to cluster the CO2 data, allowing the GNP variable to enter the gating &/or expert networks, or neither, via a formula interface. Also consider models with equal mixing proportions. Note that for models with covariates in the gating network, or models with equal mixing proportions, we don't need to fit single-component models (though it could be done!) as this would merely duplicate the single-component models within `m1` and `m3`, respectively.

```
m1    <- MoE_clust(CO2, G=1:3, verbose=FALSE)
m2    <- MoE_clust(CO2, G=2:3, gating= ~ GNP, verbose=FALSE)
m3    <- MoE_clust(CO2, G=1:3, expert= ~ GNP, verbose=FALSE)
m4    <- MoE_clust(CO2, G=2:3, gating= ~ GNP, expert= ~ GNP, verbose=FALSE)
m5    <- MoE_clust(CO2, G=2:3, equalPro=TRUE, verbose=FALSE)
m6    <- MoE_clust(CO2, G=2:3, expert= ~ GNP, equalPro=TRUE, verbose=FALSE)
```

Choose the best model among these.

```
comp  <- MoE_compare(m1, m2, m3, m4, m5, m6, optimal.only=TRUE)
```

See if a better model can be found using greedy forward stepwise selection. Conduct a stepwise search on the same data

```
(mod1 <- MoE_stepwise(CO2, GNP, verbose=FALSE))
## ----------------------------------------------------------------------
## Comparison of Gaussian Parsimonious Clustering Models with Covariates
## Data: CO2
## Ranking Criterion: BIC
## Optimal Only: TRUE
```

```
## ----------------------------------------------------------------------
##
##  rank MoENames modelNames G df iters      bic      icl      aic  loglik
gating
##     1   Step_4          E 3  7    21   -155.2 -159.062 -145.875 -65.937
None
##     2   Step_3          V 2  7     7 -157.205 -160.039  -147.88  -66.94
None
##     3   Step_2          E 2  4    19 -163.164 -163.911 -157.835 -74.917
None
##     4   Step_1          E 1  2     1 -163.905 -163.905  -161.24  -78.62
None
##  expert algo equalPro
##    ~GNP   EM     TRUE
##    ~GNP   EM    FALSE
##    None   EM    FALSE
##    None   EM
```

Conduct another stepwise search considering models with a noise component.

```
(mod2 <- MoE_stepwise(CO2, GNP, noise=TRUE, verbose=FALSE))
## ----------------------------------------------------------------------
## Comparison of Gaussian Parsimonious Clustering Models with Covariates
## Data: CO2
## Ranking Criterion: BIC
## Optimal Only: TRUE
## ----------------------------------------------------------------------
##
##  rank MoENames modelNames G df iters      bic      icl      aic  loglik
gating
##     1   Step_2          E 1  4    22 -160.781 -173.158 -155.453 -73.726
None
##     2   Step_1            0  1     1 -165.503 -165.503 -164.171 -81.086
None
##  expert algo  noise
##    None   EM hypvol
##    None   EM hypvol
```

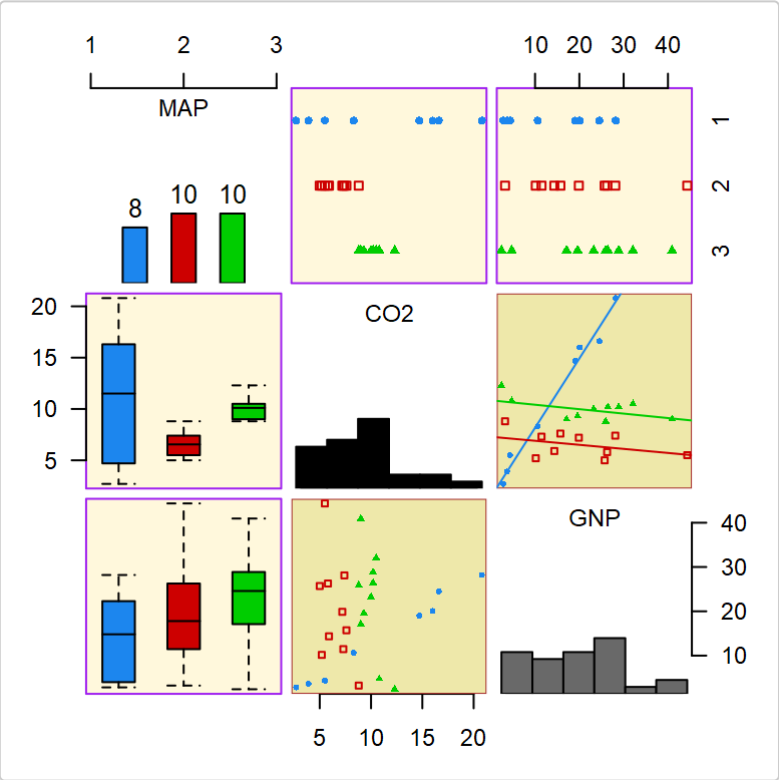
Compare all sets of results to choose the optimal model.

```
(best  <- MoE_compare(mod1, mod2, comp, pick=1)$optimal)
## Call:    MoE_stepwise(data = CO2, network.data = GNP, verbose = FALSE)
##
```

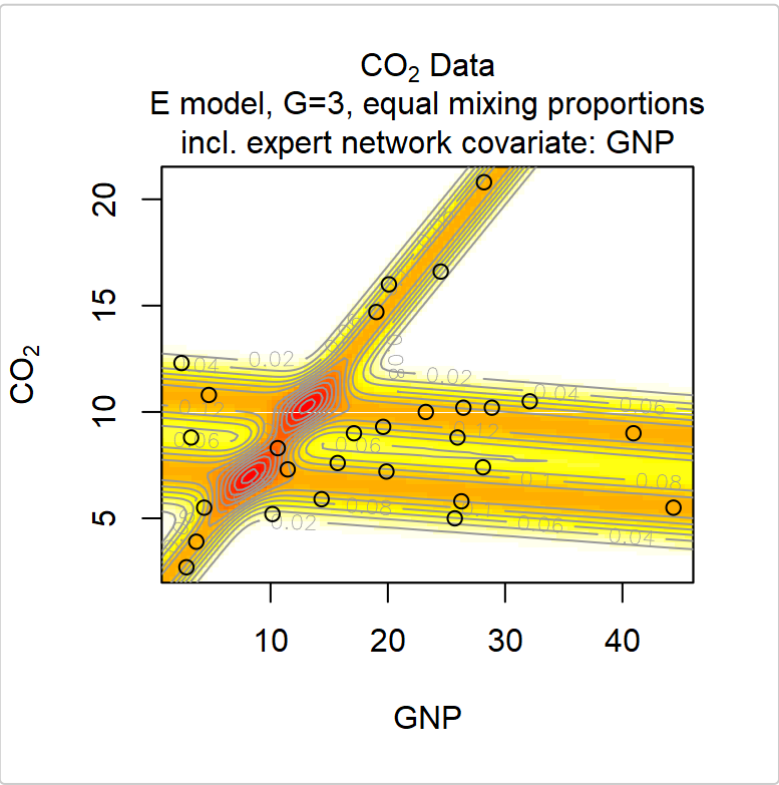
```
## Best Model (according to BIC): univariate, equal variance (E), with 3
components
## Equal Mixing Proportions
## BIC = -155.2 | ICL = -159.062 | AIC = -145.875
## Including expert network covariates:
##  Expert: ~GNP


(summ <- summary(best))


## ------------------------------------------------------
## Gaussian Parsimonious Clustering Model with Covariates
## Data: CO2
## ------------------------------------------------------
##
## MoEClust: E (univariate, equal variance), with 3 components
##
## Gating Network Covariates:  None
## Expert Network Covariates:  ~GNP
## Equal Mixing Proportions:   TRUE
## Noise Component:            FALSE
##
##  log.likelihood  n d df iters    BIC      ICL      AIC Algo
##         -65.937 28 1  7    21 -155.2 -159.062 -145.875   EM
##
## Clustering table:
##  1  2  3
##  8 10 10
```

Visualise the results for the optimal model using a generalised pairs plot.

```
plot(best, what="gpairs", jitter=FALSE)
```

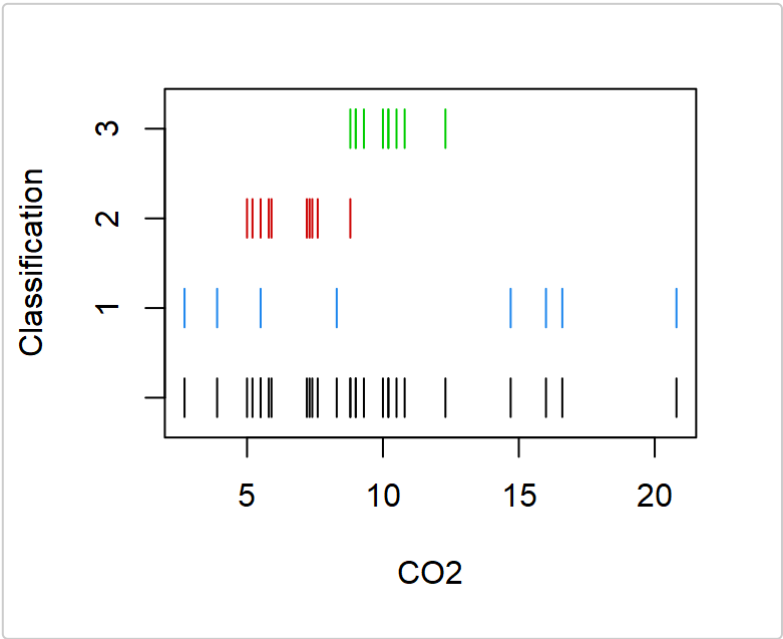


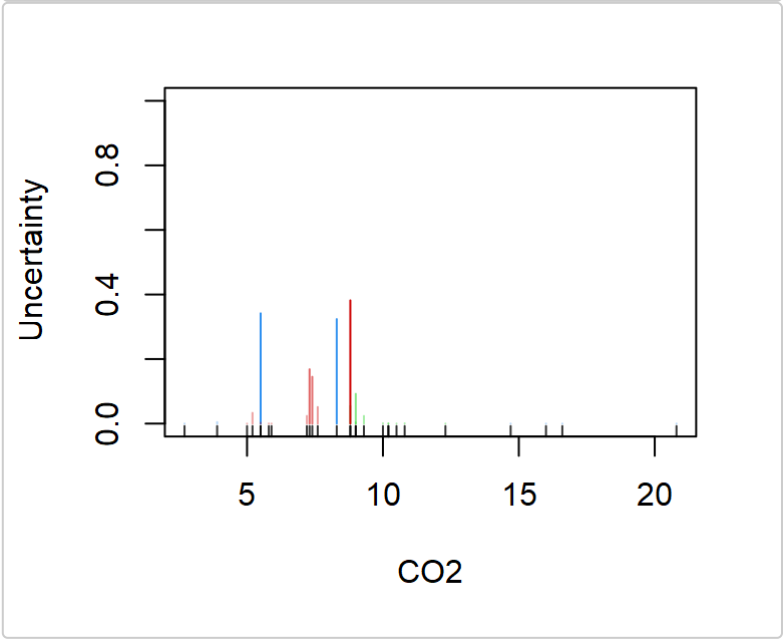
Visualise the density of the mixture distribution.



Convert from the `"MoEClust"` class to the `"Mclust"` class in order to further visualise the results. Examine the `"classification"` and `"uncertainty"` options.

```
mod <- as.Mclust(comp$optimal)
plot(mod, what="classification")
plot(mod, what="uncertainty")
```





Predictions can also be made from `MoEClust` models: the response, probability of cluster membership, and the MAP classification can be predicted for the fitted data or for new data (in the form of new covariates and new response variables, or new covariates only). Let's predict the response variable using the optimal model fit above to the CO2 data.

```
as.vector(predict(comp$optimal)$y)
##  [1] 14.258797  3.901356 20.461833  9.057538  8.292203 14.981863  6.849704
##  [8]  6.679846  9.695510 10.632816  9.451615  9.831188  6.255028  9.590701
## [15]  7.237534  5.289565  9.782900  6.588562  9.531164 17.968480  8.514569
## [22]  6.936316  6.725192  6.275709  5.546887  3.319349  9.910969 10.736908
```

Now let's build a model on some of the CO2 data and retain the indices of the withheld observations:

```
ind     <- sample(1:nrow(CO2data), 2)

res2    <- MoE_clust(CO2data[-ind,]$CO2, G=3, expert=~GNP,
                     equalPro=TRUE, network.data=CO2data[-ind,])
```

Now we can make predictions on the withheld data, either by using the withheld covariates only, or by also using the withheld response variables. Note that `newdata` can be either a list with component(s) `new.x` (and optionally `new.y`) or a single matrix/data.frame with the appropriate columns.

```
# Using new covariates only

predict(res2,
        newdata = CO2data[ind,],
        use.y = FALSE)[1:3]
## y :
##         CO2
## 1  7.254644
## 2 11.543315
##
## classification :
## 1 2
## 2 2
##
## z :
##    Cluster1  Cluster2  Cluster3
## 1 0.3333333 0.3333333 0.3333333
## 2 0.3333333 0.3333333 0.3333333
```

```
# Using both new covariates & new response data

predict(res2,
        newdata = CO2data[ind,])[1:3]
## y :
##         CO2
## 1  3.870231
## 2 18.510889
##
## classification :
## 1 2
## 2 2
##
## z :
##       Cluster1  Cluster2     Cluster3
## 1 1.059569e-11 0.9971625 2.837470e-03
## 2 8.410580e-11 1.0000000 8.891566e-25
```

AIS Data

Load the Australian Institute of Sports data.

```
data(ais)
hema  <- ais[,3:7]
```

Examine the various additional options around initialisation of the algorithm:

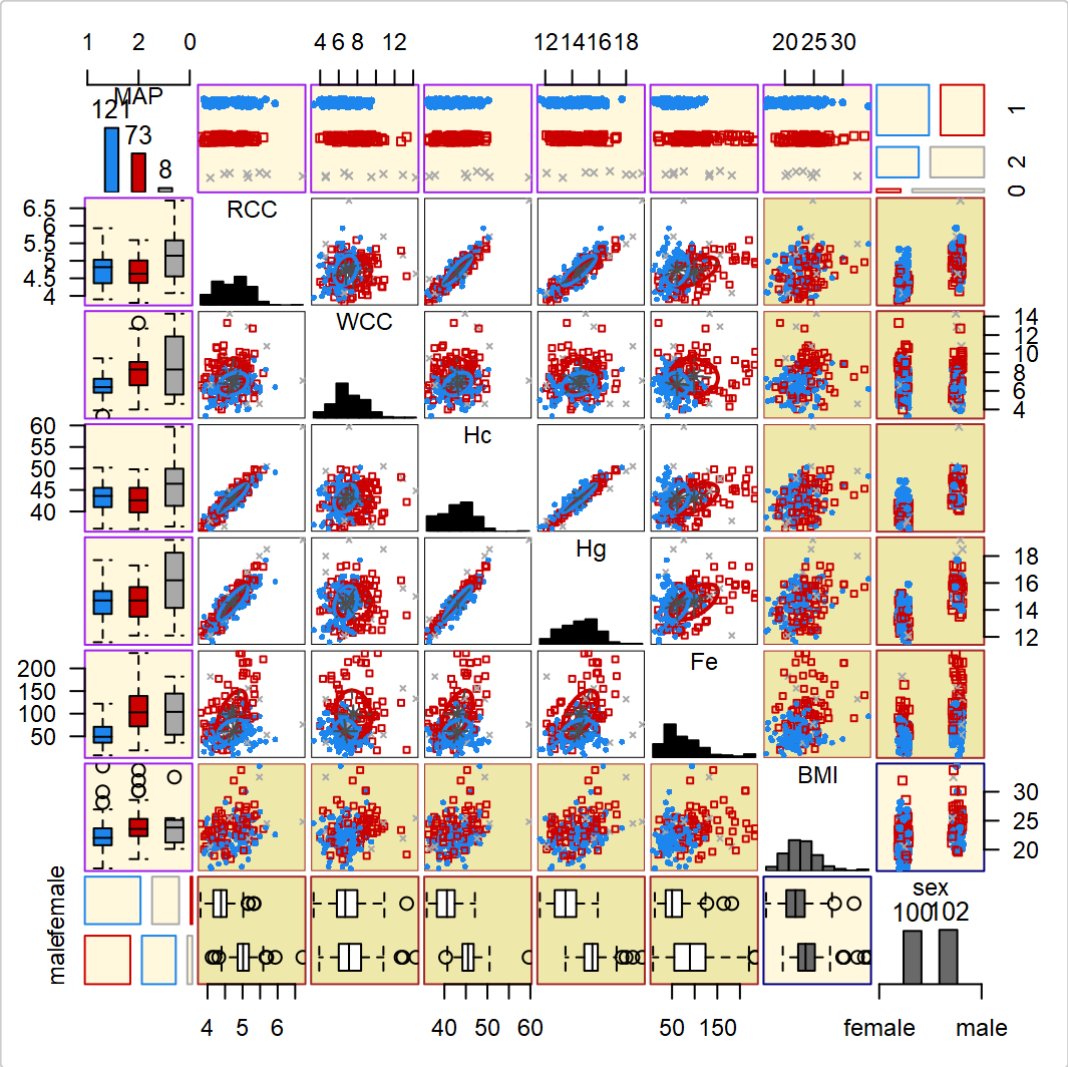
```
?MoE_control
```

Fit a parsimonious Gaussian mixture of experts MoEClust model to the hematological variables within the AIS data, supplying `sex` in the expert network and `BMI` in the gating network via formula interfaces. Include an additional noise component by specifying it's prior mixing proportion `tau0`. Toggle between allowing the mixing proportion for the noise component depend on the gating concomitant or not via the `noise.gate` argument. This time, allow the printing of messages to the screen.

```
mod   <- MoE_clust(hema, G=1:3, expert= ~ sex, gating= ~ BMI,
                   network.data=ais, tau0=0.1, noise.gate=FALSE)
```

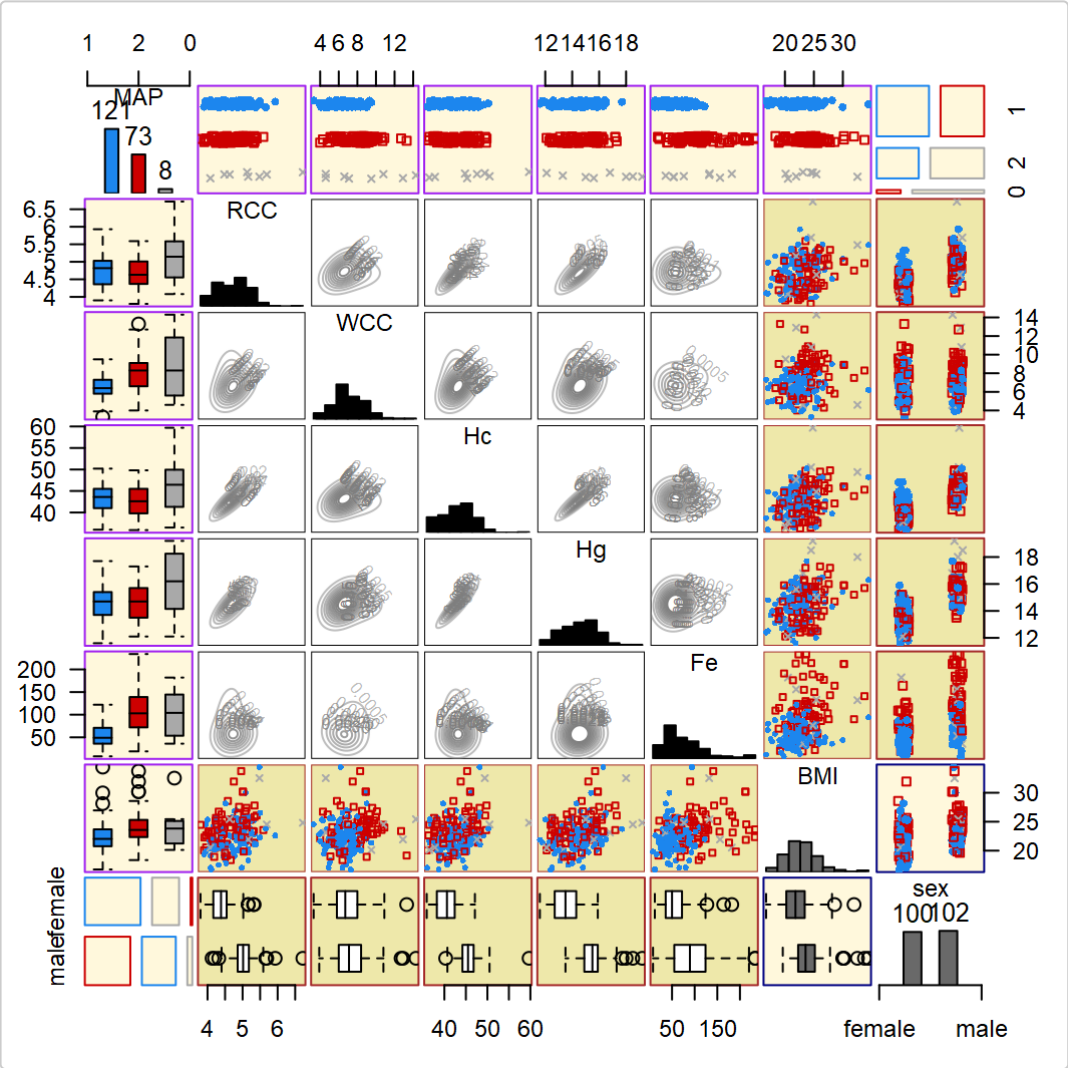
Visualise the results for the optimal model using a generalised pairs plot.

```
plot(mod, what="gpairs")
```



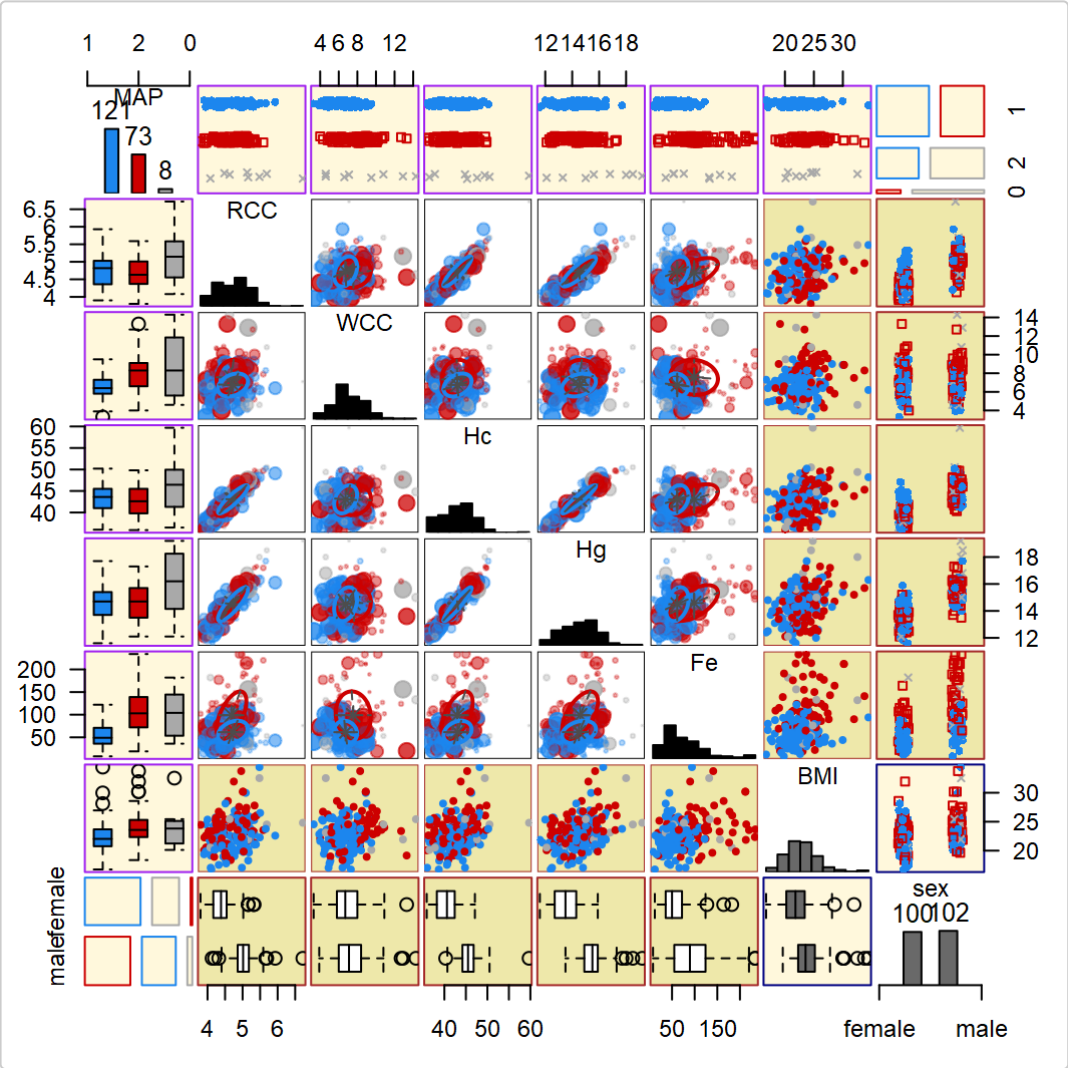
Replace the scatter plots in response vs. response panels with bivariate density contours. Note that this is liable to be slow for models with expert network covariates.

```
plot(mod, what="gpairs", response.type="density")
```



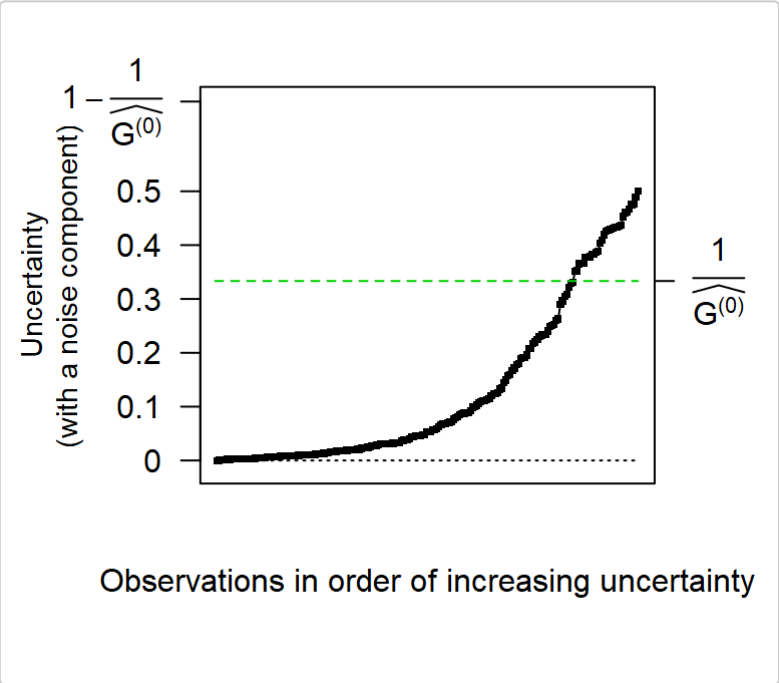
Visualise the clustering uncertainty for the optimal model using a generalised pairs plot.

```
plot(mod, what="gpairs", response.type="uncertainty")
```



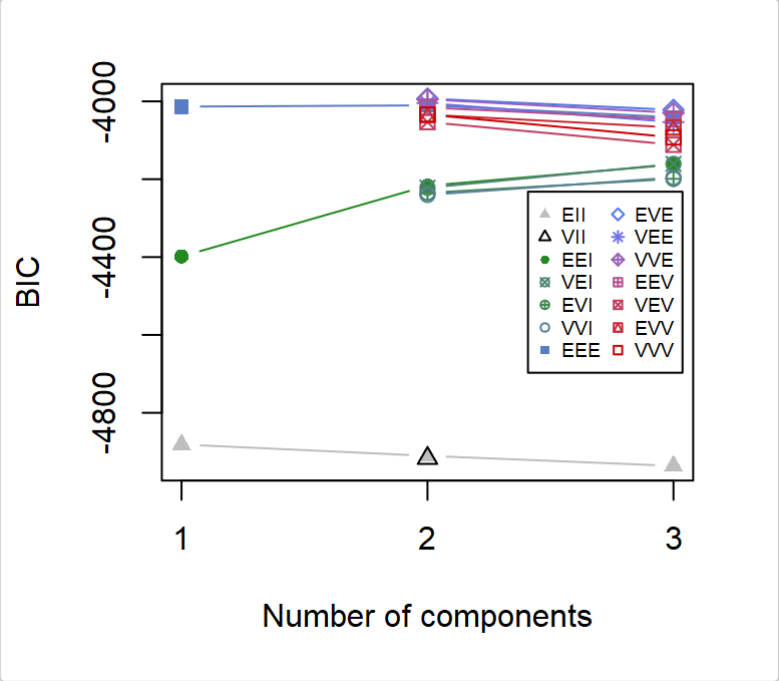
Instead visualise the clustering uncertainty in the form of an ordered profile plot (`type="barplot"` can also be specified here).

```
plot(mod, what="uncertainty", type="profile")
```



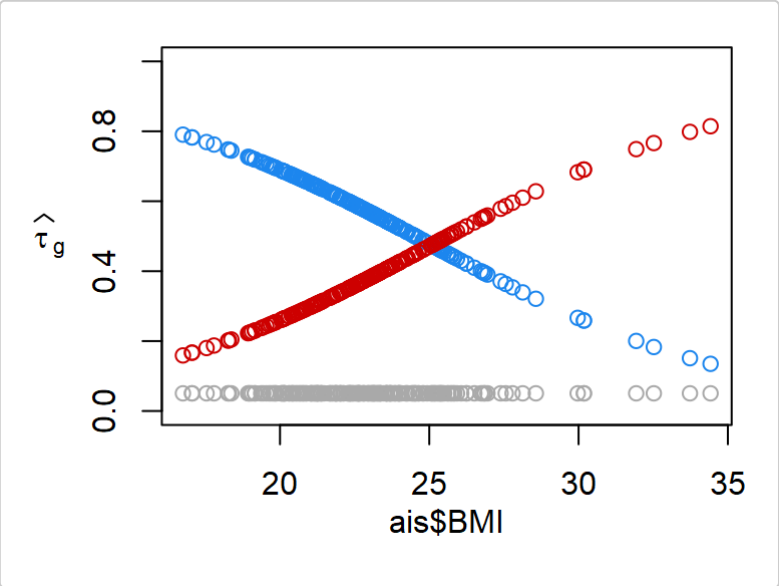
Plot the BIC of the visited models.

```
plot(mod, what="criterion", legendArgs=list(x="right"))
```



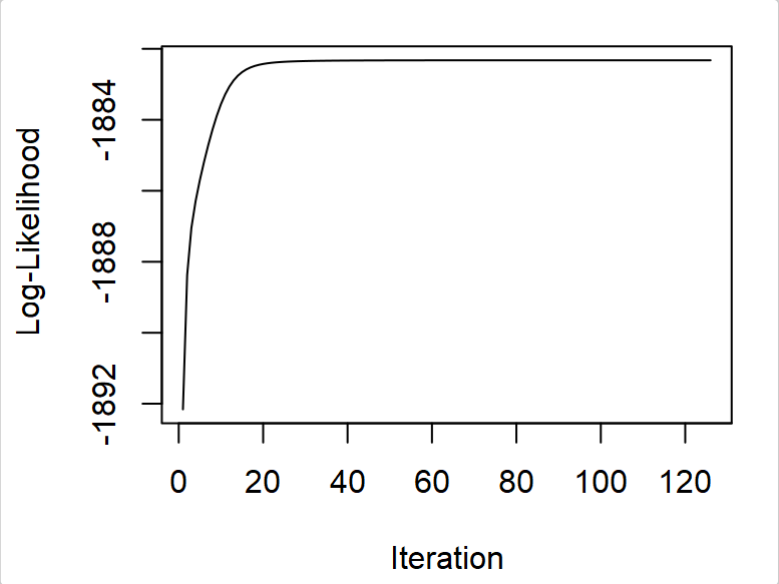
Plot the gating network of the optimal model against the gating concomitant `BMI`. Note the flat horizontal line of grey circles corresponding to the noise component due to the specification of `noise.gate=FALSE` in the original function call.

```
plot(mod, what="gating", x.axis=ais$BMI, type="p", xlab="BMI", pch=1)
```



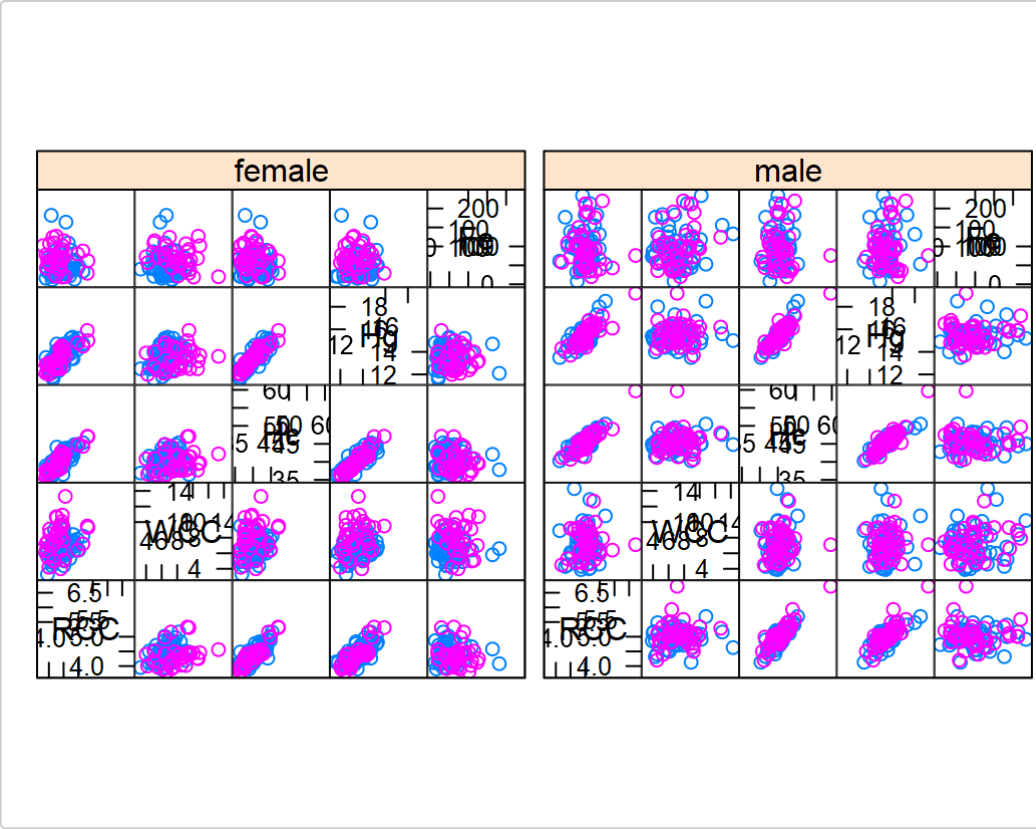
For the optimal model, plot the log-likelihood vs. the number of EM iterations.

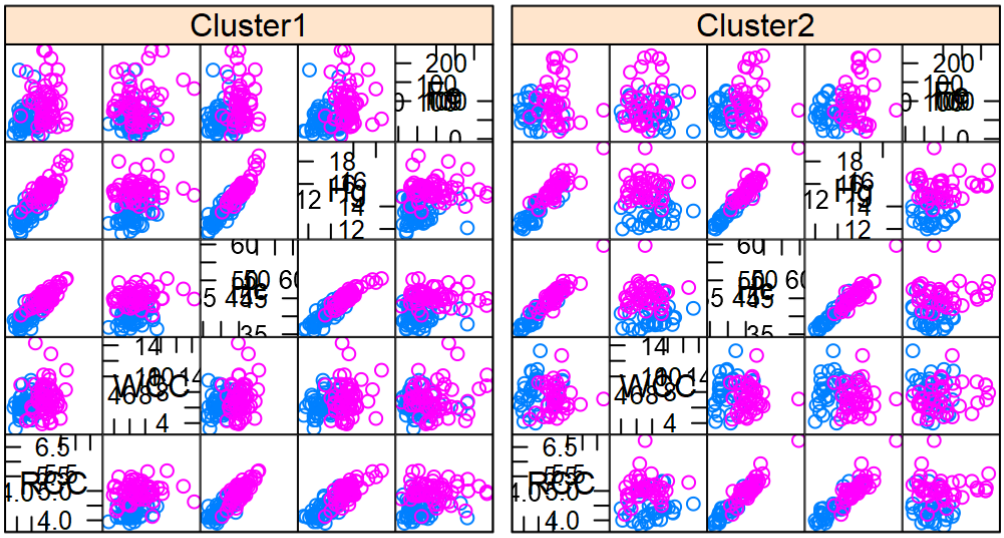
```
plot(mod, what="loglik")
```



Produce further visualisations for the Gaussian components with the aid of the `lattice` library.

```
require("lattice")
z <- factor(mod$classification[mod$classification > 0],
            labels=paste0("Cluster", seq_len(mod$G)))
splom(~ hema | ais$sex, groups=z)
splom(~ hema | z, groups=ais$sex)
```





References

Murphy, K. and T. B. Murphy (2019). Gaussian parismonious clustering models with covariates and a noise component. *Advances in Data Analysis and Classification*, 1-33. URL https://doi.org/10.1007/s11634-019-00373-8.

Fraley, C. and A. E. Raftery (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association 97*(45):611-631.

Chapter 3

Clustering Longitudinal Life-Course Sequences using Mixtures of Exponential-Distance Models

Abstract

Sequence analysis is an increasingly popular approach for the analysis of life courses represented by categorical sequences, i.e. as an ordered collection of activities experienced by subjects over a given time period. Several criteria have been introduced in the literature to measure pairwise dissimilarities among sequences. Typically, dissimilarity matrices are employed as the input to heuristic clustering algorithms, with the aim of identifying the most relevant patterns in the data.

Here, we propose a model-based clustering approach for categorical sequence data. The technique is applied to a survey data set containing information on the career trajectories, in terms of monthly labour market activities, of a cohort of Northern Irish youths tracked from the age of 16 to the age of 22.

Specifically, we develop a family of methods for clustering sequence data directly based on mixtures of exponential-distance models, which we call MEDseq. The Hamming distance, or weighted variants thereof, are employed as the distance metric. The existence of closed-form expressions for the normalising constant using these metrics facilitates the development of an ECM algorithm for model fitting. Additionally, MEDseq models allow the probability of component membership to

depend on fixed covariates. Sampling weights, which are often associated with life-course data arising from surveys, are also accommodated. The simultaneous inclusion of both the weights and the covariates in the clustering process allows new insights to be gleaned from the Northern Irish data.

Keywords: Life-course data, categorical sequences, exponential-distance models, model-based clustering, weighted Hamming distance, gating covariates, survey sampling weights.

3.1 Introduction

Sequence analysis (SA) is an umbrella term for tools defined to explore and describe categorical life-course data. Specifically, attention is focused on the ordered sequence of states (or activities) experienced by individuals over a given time-span (usually at T equally spaced discrete time periods). The goal of analysis is to identify the most relevant patterns in the data. To this end, pairwise dissimilarities among sequences in their entirety are first assessed. Dissimilarity matrices are then employed to identify the most typical trajectories using, in the vast majority of applications, cluster analysis.

Quantifying the distance between categorical sequences is not a trivial task. Optimal matching (OM), developed by Abbott and Forrest (1986) and extended to sociology by Abbott and Hrycak (1990), is popular among the SA community. OM is derived from the edit distance originally proposed in the field of information theory and computer science by Levenshtein (1966). The OM metric assigns costs to the different types of edits, namely insertion, deletion, and substitution. Typically, insertion and deletion are assigned a cost of 1 while substitution costs are allowed to vary. However, specifying these costs involves subjective choices, and may lead to violations of the triangle inequality if not done carefully. Several proposals in the literature introduced criteria to improve or guide the choice of costs in OM. Also, alternative dissimilarity criteria have been introduced to allow control over the importance assigned to the characteristics of the sequences (namely, the collection of experienced states, their timing, or their duration) in the assessment of their differences: see Studer and Ritschard (2016) for an excellent discussion. Even so, there

are no results proving that one procedure is superior to the others, and the choice of dissimilarity measure remains a fundamental choice left to the researcher.

Given a dissimilarity matrix $\mathbf{D}$, obtained from a set of sequences $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_n)$, where n is the number of subjects, cluster analysis is usually applied to group sequences and to identify the most typical trajectories experienced by the sampled individuals. Typically, heuristic clustering algorithms, either hierarchical or partitional, are used. In many applications, it is also of interest to relate the sequences to a set of baseline covariates. Within the described framework, this is solely done by relating the uncovered clustering partition to covariates, using for example multinomial logistic regression (MLR). This approach is questionable from a few points of view. Firstly, the original sequences are substituted by a categorical variable indicating clustering membership, thus disregarding the heterogeneity within clusters. This is clearly only sensible when the clusters are sufficiently homogeneous. However, a clear clustering structure can often be obtained only by increasing the number of clusters (often with some clusters possibly small in size). More importantly, suitable partitions do not necessarily lead to suitable response variables as input for the MLR. It thus seems desirable to cluster sequences and relate the clusters to the covariates simultaneously.

To address these issues, we propose to cluster trajectories in a model-based fashion, allowing the covariates to guide the construction of the clusters, rather than leaving them exogenous to the clustering model. This permits to better understand if and to what extent specific covariates affect the typical sequence patterns characterising each cluster. Model-based clustering methods typically assume that the data arise from a finite mixture of G distributions; Bouveyron et al. (2019) provide an excellent overview. In principle, any distribution can be used, though the term 'model-based clustering' was popularised by Banfield and Raftery (1993), in which the underlying distributions are assumed to be parsimoniously parameterised multivariate Gaussians with component-specific parameters. Such models have been recently extended to the mixture of experts setting (Gormley and Frühwirth-Schnatter, 2019) to facilitate dependence on fixed covariates (Murphy and Murphy, 2019). However, these models can be problematic when applied to dissimilarity matrices, either due to non-identifiability or because the input data are usually far from Gaussian. This problem cannot be addressed by applying mul-

tidimensional scaling to **D** because the resulting low-dimensional configuration is also typically far from Gaussian. Notably, our attempts to fit non-Gaussian mixtures in these settings did not yield useful results.

Another popular framework for clustering categorical data is latent class analysis (LCA; Lazarsfeld and Henry 1968)). Agresti (2002) shows the connection between model-based clustering and LCA. Such models are finite mixtures in which the component distributions are assumed to be multi-way cross-classification tables with all variables mutually independent. Latent class regression models (Dayton and Macready, 1988) are particularly interesting, because their connection to the mixture of experts framework permits the inclusion of covariates to predict the latent class memberships. However, fitting such models is challenging when the sequence length, the number of categories, or the number of latent classes are even moderately large, due to the explosion in the number of parameters.

For the reasons mentioned above, we model the sequences directly, via parsimonious mixtures of exponential-distance models. Exponential-distance models typically depend on a central sequence and a precision parameter in a way that relates to the chosen distance metric. Mostly for reasons of computational convenience, we use dissimilarities based on simple matching, in particular the Hamming distance (Hamming, 1950). This distance is liable to suffer from temporal rigidity, since anticipations and/or postponements of the same choices in life courses are not accounted for. Hence, similar sequences shifted by one time period may be maximally distant from one another. While misalignment is less of a concern for sequences exhibiting long durations in the same state, we address the issue using weighted variants of the Hamming distance, characterised by a range of constraints on the precision parameters in the mixture setting. This leads to the novel MEDseq family of models, which can be seen as similar to a version of the k-medoids/PAM algorithm (Kaufman and Rousseeuw, 1990) based on the Hamming distance with some restrictions relaxed.

Our approach is illustrated using data from the 1999 sweep of the Status Zero Survey (McVicar, 2000; McVicar and Anyadike-Danes, 2002) — henceforth referred to as the MVAD data — on the school-to-work trajectories experienced by a cohort of Northern Irish youths. McVicar and Anyadike-Danes (2002) apply Ward's agglomerative hierarchical clustering algorithm (Ward, 1963) to an OM dissimilar-

ity matrix to obtain $G = 5$ clusters of these trajectories, without performing model selection. Thereafter, they use MLR to relate the hard assignments of trajectories to the clusters to a set of baseline covariates. We instead cluster the MVAD data in a model-based fashion, using the MEDseq model family, and allow the covariates to guide the construction of the clusters by assuming they influence the probability of component membership. Importantly, information is also available on the survey sampling weights, which are only incorporated in the MLR stage of the analysis in McVicar and Anyadike-Danes (2002). While sampling weights can be incorporated into heuristic clustering algorithms, such as Ward's hierarchical clustering (by weighting the linkages between clusters) or k-medoids, and subsequently in the MLR, one of the advantages of our approach is that both the covariates and the weights are incorporated simultaneously.

MEDseq models, like standard SA heuristic clustering algorithms and LCA models, approach the clustering task from the holistic perspective of modelling whole trajectories, in order to uncover groups of similar sequences. In contrast, a number of multistate models employing finite mixtures with Markov components (e.g. Melnykov 2016a; Pamminger and Frühwirth-Schnatter 2010) or with hidden Markov components (Helske et al., 2016) have recently attained popularity for the analysis of categorical sequence data. Such models focus on modelling instantaneous transitions within the life course and on factors that might explain the probability of experiencing them. As described by Wu (2000), this amounts to a difference between considering sequences in their entirety under the MEDseq framework or as time-to-event processes under the Markovian framework.

The remainder of the article is organised as follows. Section 3.2 presents some exploratory analysis of the MVAD data. Section 3.3 develops the MEDseq family of mixtures of exponential-distance models that account for sampling weights and allow potential dependency on covariates. Section 3.4 describes the model fitting procedure and discusses factors affecting performance. Section 3.5 presents results for the MVAD data, including applications of MEDseq models and comparisons to other methods. The insights gleaned from the MVAD data under the optimal MEDseq model are summarised in Section 3.6. The paper concludes with a brief discussion on the MEDseq methodology and potential future extensions in Section 3.7. A software implementation for the full MEDseq model family is provided by

the associated R package `MEDseq` (Murphy et al., 2019), which is available from `www.r-project.org` (R Core Team, 2019), with which all results were obtained.

3.2 Status Zero Survey: MVAD Data

The term 'MVAD data' refers throughout to a cohort of $n = 712$ Northern Irish youths aged 16 and eligible to leave compulsory education as of July 1993 who were observed at monthly intervals until June 1999 as part of the Status Zero Survey (McVicar, 2000; McVicar and Anyadike-Danes, 2002). The subjects were interviewed about the labour market activities they experienced, distinguishing between employment (EM), further education (FE), higher education (HE), joblessness (JL), school (SC), or training (TR). Each observation i is represented by an ordered categorical sequence of length $T = 72$, with an alphabet of $v = 6$ possible categories, e.g. $\mathbf{s}_i = (s_{i,1}, s_{i,2}, \dots, s_{i,72})^\top = (\text{SC,SC,} \dots \text{,TR,TR,} \dots \text{,EM,EM})^\top$. Notably, the transitions HE $\rightsquigarrow$ SC and TR $\rightsquigarrow$ HE are never observed. The sequences share a common length, the time periods are equally spaced, and there are no missing data.

It is of interest to relate the MVAD sequences to covariates in order to understand whether different characteristics — related to gender, community, geographic and social conditions, and personal abilities — impact on the school-to-work trajectories. These covariates are summarised in Table 3.1. All covariates were measured at the age of 16 (i.e. at the start of the study period in July 1993), with the exception of 'Funemp' and 'Livboth', and are thus static background characteristics. The MVAD data also come with associated observation-specific survey sampling weights. Each sample was weighted based on the first state value at age 16, and the 'Grammar' and 'Location' covariates (McVicar and Anyadike-Danes, 2002).

The MVAD data are available in the R packages `MEDseq` and `TraMineR` (Gabadinho et al., 2011). As the data have been used to illustrate some of the functionalities of the `TraMineR` package in its associated vignette[4], interesting features of an exploratory analysis of the data can be found therein. However, we reproduce plots of the transversal state distributions in Figure 3.1 and the transversal Shannon entropies in Figure 3.2, i.e. the entropy of each time point of the state distribution (Billari, 2001). Note that the sampling weights are accounted for in both cases.

[4] `cran.r-project.org/web/packages/TraMineR/vignettes/TraMineR-state-sequence.pdf`

Table 3.1: Available covariates for the MVAD data set. For binary covariates, the event denoted by 1 is indicated. Otherwise, the levels of the categorical covariate 'Location' are grouped in curly brackets.

Covariate	Description
Gender	1=male
Catholic	1=yes
Grammar	Type of secondary education, 1=grammar school
Funemp	Father's employment status as of June 1999, 1=employed
GCSE5eq	Qualifications gained by the end of compulsory education, 1=5+ GCSEs at grades A-C, or equivalent
FMPR	SOC code of father's current or most recent job as of the beginning of the survey, 1=SOC1 (professional, managerial, or related)
Livboth	Living arrangements as of June 1995, 1=living with both parents
Location	{Belfast, N. Eastern, S. Eastern, Southern, Western}

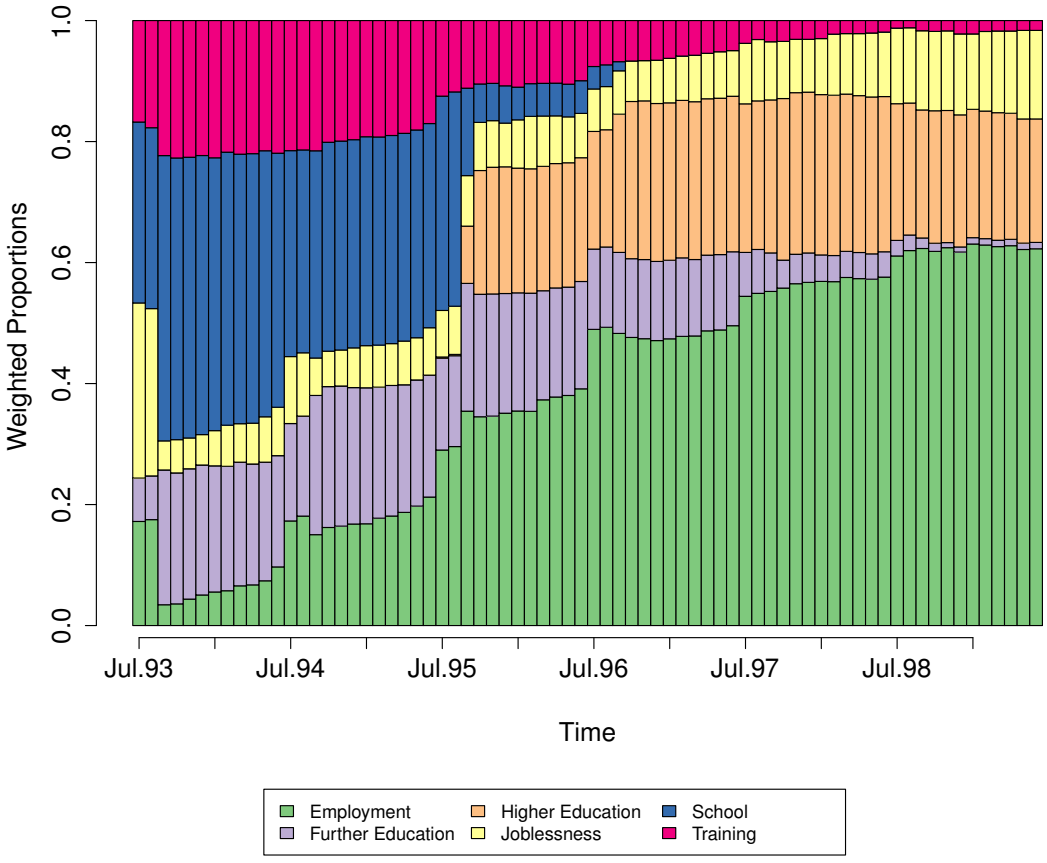


Figure 3.1: Overall state distribution for the weighted MVAD data.

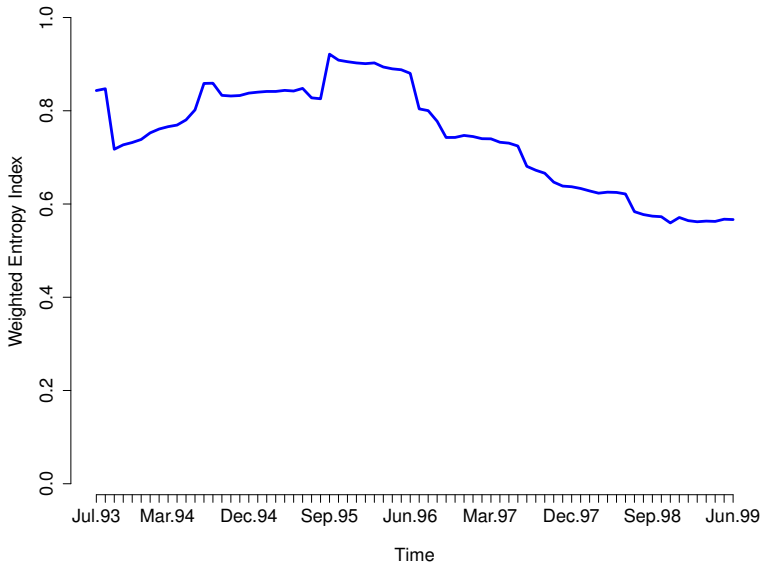


Figure 3.2: Transversal entropy plot for the weighted MVAD data.

Figure 3.1 shows that the number of subjects who found employment increased over time. Conversely, fewer students were in training or further education by the end of the observation period. Most students appear to have entirely left school within 2/3 years of the commencement of the survey. Interestingly, many students were jobless during the first two months of observation, possibly because this period coincided with the summer break from school. Finally, while students only began to pursue higher education from July 1995 onwards, a number of students had already pursued further education during the two preceding years. Figure 3.2 confirms that the heterogeneity of the state distribution varies over time. In particular, the entropy declines after Sep 1995, by which point most students had left school.

3.3 Modelling

In this section, we introduce the family of MEDseq models. The exponential-distance model is described in Section 3.3.1, extended to account for sampling weights in Section 3.3.2, expanded into a family of mixtures in Section 3.3.3, and finally embedded within the mixture of experts framework in Section 3.3.4 in order to accommodate covariates.

3.3.1 Exponential-Distance Models

For an arbitrary distance metric $\mathrm{d}(\cdot, \cdot)$, a location parameter $\boldsymbol{\theta}$, and a precision parameter λ, the probability mass function (PMF) of an exponential-distance model for sequences is

$$f(\mathbf{s}_i \mid \boldsymbol{\theta}, \lambda, \mathrm{d}) = \frac{\exp(-\lambda \mathrm{d}(\mathbf{s}_i, \boldsymbol{\theta}))}{\sum_{\mathbf{s}_i \in \mathbf{S}'} \exp(-\lambda \mathrm{d}(\mathbf{s}_i, \boldsymbol{\theta}))} = \Psi(\lambda, \boldsymbol{\theta} \mid T, v)^{-1} \exp(-\lambda \mathrm{d}(\mathbf{s}_i, \boldsymbol{\theta}))\,, \quad (3.1)$$

with the corresponding log-likelihood function given by

$$\ell(\boldsymbol{\theta}, \lambda \mid \mathbf{S}, \mathrm{d}) = \sum_{i=1}^{n} \log f(\mathbf{s}_i \mid \boldsymbol{\theta}, \lambda, \mathrm{d}) = -\lambda \sum_{i=1}^{n} \mathrm{d}(\mathbf{s}_i, \boldsymbol{\theta}) - n \log \Psi(\lambda, \boldsymbol{\theta} \mid T, v)\,. \quad (3.2)$$

Such a model is analogous to the Gaussian distribution (characterised by the squared Euclidean distance from the mean) and similar to the Mallows model for permutations (Mallows, 1957). Indeed, mixtures of Mallows models have been used to cluster rankings (Murphy and Martin, 2003). We only consider models with $\lambda \geq 0$. When $\lambda = 0$, the distribution of sequences is uniform. For $\lambda > 0$, the central sequence $\boldsymbol{\theta}$ is the mode, i.e. the sequence with highest probability, and the probability of any other sequence decays exponentially as its distance from $\boldsymbol{\theta}$ increases. The precision parameter λ controls the speed of this decay. Larger λ values cause sequences to concentrate around $\boldsymbol{\theta}$, tending toward a point-mass as $\lambda \to \infty$. Notably, λ is not identifiable when all sequences are identical.

The log-likelihood in (3.2) is generally intractable, as the normalising constant $\Psi(\lambda, \boldsymbol{\theta} \mid T, v)$ depends on λ (and possibly also on $\boldsymbol{\theta}$, for some more complicated distances), as well as the fixed constants $T > 1$ and $v > 1$, and requires a sum over all possible sequences. With reference to the MVAD data, for example, the computation of $\Psi(\lambda, \boldsymbol{\theta} \mid T, v)$ is practically infeasible because there are $v^T = 6^{72}$ possible sequences. Fortunately, however, the normalising constant exists in closed form under the Hamming distance metric, $\mathrm{d}_{\mathrm{H}}(\mathbf{s}_i, \mathbf{s}_j) = \sum_{t=1}^{T} \mathbb{1}(s_{i,t} \neq s_{j,t})$, in a manner which facilitates direct enumeration and crucially does not depend on $\boldsymbol{\theta}$. Consider, for example, the Hamming distances between all ternary ($v = 3$) sequences of length $T = 4$. From the arbitrary reference sequence $(0, 0, 0, 0)$, there is 1 instance of a distance of 0, 8 instances of a distance of 1, 24 instances of a dis-

tance of 2, 32 instances of a distance of 3, and 16 instances of a distance of 4. Therefore, $\Psi_{\mathrm{H}}(\lambda \mid T, v) = e^{0} + 8e^{-\lambda} + 24e^{-2\lambda} + 32e^{-3\lambda} + 16e^{-4\lambda}$. Hence, the normalising constant under the Hamming distance metric depends on the parameter λ, the sequence length T, and the number of categories v, and simplifies greatly:

$$\Psi_{\mathrm{H}}(\lambda \mid T, v) = \sum_{p=0}^{T} \binom{T}{p} (v-1)^{p} \exp(-\lambda p) = \left((v-1)\, e^{-\lambda} + 1\right)^{T}. \tag{3.3}$$

Inspired by the generalised Mallows model (Irurozki et al., 2019), the model in (3.1) based on the Hamming distance can be extended to one based on the weighted Hamming distance. By introducing T precision parameters $\lambda_1, \ldots, \lambda_T$, one for each time point (i.e. sequence position), and expressing the exponent in (3.1) as $\mathrm{d}_{\mathrm{WH}}(\mathbf{s}_i, \boldsymbol{\theta}) = \sum_{t=1}^{T} \lambda_t \mathbb{1}(s_{i,t} \neq \theta_t)$ rather than $\lambda \mathrm{d}_{\mathrm{H}}(\mathbf{s}_i, \boldsymbol{\theta}) = \lambda \sum_{t=1}^{T} \mathbb{1}(s_{i,t} \neq \theta_t)$, different time points can contribute differently to the overall distance, weighted according to the period-specific precision parameters. Thus, the distance from a sequence to the central sequence under the weighted Hamming distance becomes a sum of the precision parameters associated with each time point which differs from the corresponding central sequence position. This allows modelling a situation in which there is high consensus regarding the state values of some time period(s), with a large uncertainty about the values of others, and can help to prevent sequences the same distance from $\boldsymbol{\theta}$ from having the same probability. Returning to the MVAD data, the non-constant transversal entropies in Figure 3.2 suggest that such an extension may be fruitful. The extension requires rewriting the log-likelihood in (3.2) with the weighted Hamming distance decomposed into its T components and the normalising constant (3.3) also modified:

$$\ell(\boldsymbol{\theta}, \lambda_1, \ldots, \lambda_T \mid \mathbf{S}, \mathrm{d}_{\mathrm{WH}}) = -\sum_{i=1}^{n} \left[\sum_{t=1}^{T} \left(\lambda_t \mathbb{1}(s_{i,t} \neq \theta_t) + \log\left((v-1)\, e^{-\lambda_t} + 1 \right) \right) \right].$$

Though other dissimilarity measures are available for sequences, we henceforth consider only the Hamming or weighted Hamming distances. In our setting, $\mathrm{d}_{\mathrm{H}}(\cdot, \cdot)$ can be seen as a special case of OM with all substitution costs equal to λ and no insertions or deletions (see Appendix 3.D). In the sense of having time-varying substitution costs, $\mathrm{d}_{\mathrm{WH}}(\cdot, \cdot)$ is similar to the dynamic Hamming distance (DHD; Lesnard, 2010), a prominent alternative to OM. However, the substitution

costs in our model are always assumed to be common with respect to each pair of states. Hence, $\mathrm{d}_{\mathrm{WH}}(\cdot, \cdot)$ corresponds to the Gower distance (Gower, 1971) with equally weighted states and equally or unequally weighted time points.

3.3.2 Incorporating Sampling Weights

Sampling weights are often associated with life-course data, as the data typically arise from surveys where the weights are used to correct for representivity bias or stratified sampling schemes. Following Chambers and Skinner (2003), the sampling weights $\mathbf{w} = w_1, \ldots, w_n$ are incorporated into the exponential-distance model by exponentiating the likelihood of each sampled unit by the attached weight w_i, which is akin to unit i being observed w_i times. The resultant pseudo likelihood $\mathcal{L}^{\mathbf{w}}(\cdot \mid \cdot)$ reweights the likelihood contribution for each unit in order to rebalance the information in the observed sample to approximate the balance of information in the target finite population. The sampling weights $\mathbf{w}$ are thus interpretable as being inversely proportional to the unit inclusion probabilities, remain fixed, and are confined to those included in the sample. Notably, $f(\mathbf{s}_i \mid \boldsymbol{\theta}, \lambda, \mathrm{d})^{w_i} \propto f(\mathbf{s}_i \mid \boldsymbol{\theta}, w_i\lambda, \mathrm{d})$, such that the weights induce a unit-specific rescaling of the precision parameter; it follows that the observed data are independent but not identically distributed.

A secondary benefit of this extension is that it facilitates computational gains in the presence of duplicate observations. Such duplicates are quite likely when dealing with discrete life-course data. Indeed, non-uniqueness can be exploited using likelihood weights for computational efficiency, by fitting models to the subset of unique sequences only, weighted by the sum of the sampling weights (if available, otherwise $w_i = 1 \; \forall \; i$) across each corresponding set of duplicates. In so doing, observations with different sampling weights which are otherwise duplicates are also treated as duplicates, in such a way that the (pseudo) likelihood is unaltered. The number of duplicates clearly lowers when considering both the sequences themselves and their associated covariate patterns. In particular, all observations are unique when there are continuous covariates. Nonetheless, in many applications — e.g. the MVAD data (see Table 3.1) — the covariates are all categorical. Hence, exploiting non-uniqueness in this manner can be extremely computationally convenient, with or without existing sampling weights. For instance, only 557 of the $n = 712$ sequences in the MVAD data set are distinct.

3.3.3 A Family of Mixtures of Exponential-Distance Models

Extending the exponential-distance model with the Hamming distance and sampling weights to the model-based clustering setting yields a weighted pseudo likelihood function of the form

$$\mathcal{L}^{\mathbf{w}}(\lambda, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_G \mid \mathbf{S}, \mathbf{w}, \mathrm{d}_{\mathrm{H}}) = \prod_{i=1}^{n} \left[\sum_{g=1}^{G} \tau_g \frac{\exp(-\lambda \mathrm{d}_{\mathrm{H}}(\mathbf{s}_i, \boldsymbol{\theta}_g))}{\left((v-1)\, e^{-\lambda} + 1\right)^{T}} \right]^{w_i},$$

where the mixing proportions $\tau_1, \ldots, \tau_G$ are positive and sum to 1. Thus, the clustering approach is both model-based and distance-based, thereby bridging the gap between these two 'cultures' in the SA community.

The mixture setting naturally suggests a further extension, whereby the precision parameter λ can be constrained or unconstrained across clusters, in addition to the aforementioned possibility for the precision parameters to be constrained or unconstrained across time points. Within a family of models we term 'MEDseq', we thus define the CC, UC, CU, and UU models, where the first letter denotes whether precision parameters are constrained (C) or unconstrained (U) across clusters and the second denotes the same across time points. Hence, all but the CC model employ different weighted variants of the Hamming distance.

Given the role played by λ when it takes the value 0, whereby the distribution of the sequences is uniform, it is convenient and natural to include a noise component (denoted by N) whose single precision parameter is fixed to 0. This extension can be added to each of the 4 models above, regardless of how the precision parameters are otherwise specified. This completes the MEDseq model family with the CCN, UCN, CUN, and UUN models. When $G = 1$, the CC, CU, and CCN models can be fitted. When $G = 2$, the CCN and CUN models are equivalent to the UCN and UUN models, respectively, as there is only one non-noise component. As the noise component arises naturally from restricting the parameter space, we consider the noise component as one of the G components, denoted hereafter with the subscript 0. All 8 model types are summarised further in Appendix 3.A.

3.3.4 Incorporating Covariates

We now illustrate how to incorporate the available covariate information into the clustering process, both to guide the construction of the clusters and to better interpret the type of observation characterising each cluster. As is typical for model-based clustering analyses, the data are augmented in MEDseq models by introducing a latent cluster membership indicator vector $\mathbf{z}_i = (z_{i,1}, \dots, z_{i,G})^\top$, where $z_{i,g} = 1$ if observation i belongs to cluster g and $z_{i,g} = 0$ otherwise. An advantage of the MEDseq approach is that it can be easily extended to incorporate the possible effects of covariates on the sequence trajectories by allowing the covariates to influence the distribution of the latent variable $\mathbf{z}_i$.

The inclusion of covariates is achieved under the mixture of experts framework (Jacobs et al., 1991; Gormley and Frühwirth-Schnatter, 2019), by extending the mixture model to allow the mixing proportions for observation i to depend on covariates $\mathbf{x}_i$. This is particularly attractive as the interpretation of the remaining component-specific parameters is the same as it would be under a model without covariates. For example, in the case of the CC MEDseq model

$$f(\mathbf{s}_i \mid \lambda, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_G, \mathbf{x}_i, w_i, \mathrm{d_H}) = \left[\sum_{g=1}^{G} \tau_g(\mathbf{x}_i) \frac{\exp(-\lambda \mathrm{d_H}(\mathbf{s}_i, \boldsymbol{\theta}_g))}{((v-1)\, e^{-\lambda} + 1)^T} \right]^{w_i},$$

where the mixing proportions $\tau_g(\mathbf{x}_i)$ are referred to as 'gates' or the 'gating network', with $\tau_g(\mathbf{x}_i) > 0$ and $\sum_{g=1}^{G} \tau_g(\mathbf{x}_i) = 1$, as usual. Such a model can be seen as a conditional mixture model (Bishop, 2006) because, given the covariates $\mathbf{x}_i$, the distribution of the sequences is a finite mixture model under which $\mathbf{z}_i$ has a multinomial distribution with a single trial and probabilities equal to $\tau_g(\mathbf{x}_i)$. The distance-based k-medoids algorithm, though closely related (see Section 3.4.2), does not accommodate the inclusion of covariates.

Incorporating covariates in 'hard' clustering algorithms using MLR, as done by McVicar and Anyadike-Danes (2002), has been criticised because the hard assignment of extraneous cases can negatively impact internal cluster cohesion and the MLR coefficient estimates (Piccarreta and Studer, 2019). An advantage of the noise component in MEDseq models is that it captures uniformly distributed sequences that deviate from those in the other, more defined clusters. Filtering

outliers in this way lessens their impact on the non-noise gating network coefficients, thereby enabling more accurate inference and improving the interpretability of the effects of the covariates. Moreover, the 'soft' partition obtained under the model-based paradigm allows the cluster membership probabilities for sequences lying on the boundary between two neighbouring clusters to be quantified and the effect of such sequences on the gating network coefficients to be mitigated.

As per Murphy and Murphy (2019), the CCN, UCN, CUN, and UUN models which include an explicit noise component can be restricted to having covariates only influence the mixing proportions for the non-noise components, with all observations therefore assumed to have equal probability of belonging to the uniform noise component (i.e. by replacing $\tau_0(\mathbf{x}_i)$ with τ_0). We refer to the former setting as the gated noise (GN) model and to the latter as the non-gated noise (NGN) model. Gating covariates can only be included when $G \geq 2$ under the GN model or when there are 2 or more non-noise components under the NGN model.

3.4 Model Estimation

This section describes the strategy employed for model fitting and some implementation issues that arise in practice. Specifically, Section 3.4.1 outlines the ECM algorithm employed for parameter estimation, Section 3.4.2 discusses the initialisation of the ECM algorithm with reference to the similarities between MEDseq models and the k-medoids algorithm, and the issues of model selection and variable selection are treated in Section 3.4.3.

3.4.1 Model Fitting via ECM

Parameter estimation is greatly simplified by the existence of a closed-form expression for the normalising constant for MEDseq models under the Hamming or weighted Hamming distances. We focus on maximum (pseudo) likelihood estimation using a simple variant of the EM algorithm (Dempster et al., 1977). For simplicity, model fitting details are described chiefly for the CC MEDseq model with sampling weights and gating covariates. Additional details for other model types are deferred to Appendix 3.B. The complete data pseudo likelihood for the CC model is given by

$$\mathcal{L}_c^{\mathbf{w}}(\lambda, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_G \mid \mathbf{S}, \mathbf{X}, \mathbf{Z}, \mathbf{w}, \mathrm{d_H}) = \prod_{i=1}^{n} \left[\prod_{g=1}^{G} \left(\tau_g(\mathbf{x}_i) \frac{\exp(-\lambda \mathrm{d_H}(\mathbf{s}_i, \boldsymbol{\theta}_g))}{((v-1)\, e^{-\lambda} + 1)^T} \right)^{z_{i,g}} \right]^{w_i},$$

and the complete data pseudo log-likelihood hence has the form

$$\begin{aligned} \ell_c^{\mathbf{w}}(\lambda, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_G \mid \mathbf{S}, \mathbf{X}, \mathbf{Z}, \mathbf{w}, \mathrm{d_H}) = \sum_{i=1}^{n} \sum_{g=1}^{G} z_{i,g} w_i \big[& \log \tau_g(\mathbf{x}_i) - \lambda \mathrm{d_H}(\mathbf{s}_i, \boldsymbol{\theta}_g) \\ & - T \log\big((v-1)\, e^{-\lambda} + 1\big) \big]. \end{aligned} \tag{3.4}$$

Under this model, the distribution of $\mathbf{s}_i$ depends on the latent cluster membership variable $\mathbf{z}_i$, which in turn depends on covariates $\mathbf{x}_i$, while $\mathbf{s}_i$ is independent of $\mathbf{x}_i$ conditional on $\mathbf{z}_i$.

The iterative algorithm for MEDseq models follows in a similar manner to that for standard mixture models. It consists of an E-step (expectation) which replaces for each observation the missing data $\mathbf{z}_i$ with their expected values $\widehat{\mathbf{z}}_i$, followed by a M-step (maximisation), which maximises the expected complete data pseudo log-likelihood. The M-step is replaced by a series of conditional maximisation (CM-steps) in which each parameter is maximised individually, conditional on the other parameters remaining fixed. Hence, model fitting is in fact conducted using an expectation conditional maximisation (ECM) algorithm (Meng and Rubin, 1993). Aitken's acceleration criterion is used to assess convergence of the non-decreasing sequence of weighted pseudo log-likelihood estimates (Böhning et al., 1994). Parameter estimates produced on convergence achieve at least a local maximum of the pseudo likelihood function. Upon convergence, cluster memberships are estimated via the maximum *a posteriori* (MAP) classification.

The E-step (with similar expressions when λ is unconstrained across clusters and/or time points) involves computing expression (3.5), where $(m+1)$ is the current iteration number:

$$\begin{aligned} \widehat{z}_{i,g}^{(m+1)} &= \mathbb{E}\left(z_{i,g} \;\middle|\; \mathbf{s}_i, \mathbf{x}_i, \widehat{\boldsymbol{\theta}}_g^{(m)}, \widehat{\lambda}^{(m)}, \widehat{\boldsymbol{\beta}}_g^{(m)}, w_i, \mathrm{d_H} \right) \\ &= \frac{\widehat{\tau}_g^{(m)}(\mathbf{x}_i)\, f\left(\mathbf{s}_i \mid \widehat{\boldsymbol{\theta}}_g^{(m)}, \widehat{\lambda}^{(m)}, w_i, \mathrm{d_H}\right)}{\sum_{g=1}^{G} \widehat{\tau}_g^{(m)}(\mathbf{x}_i)\, f\left(\mathbf{s}_i \mid \widehat{\boldsymbol{\theta}}_g^{(m)}, \widehat{\lambda}^{(m)}, w_i, \mathrm{d_H}\right)}. \end{aligned} \tag{3.5}$$

Note that the weights w_i in the numerator and denominator cancel each other out, leaving the E-step unchanged regardless of the inclusion or exclusion of weights.

Subsequent subsections describe the CM-steps for estimating the remaining parameters in the model. These individual CM-steps rely on the current estimates $\widehat{\mathbf{Z}}^{(m+1)} = \left(\widehat{\mathbf{z}}_1^{(m+1)}, \ldots, \widehat{\mathbf{z}}_n^{(m+1)}\right)$ to provide estimates of the regression coefficients $\widehat{\boldsymbol{\beta}}_g^{(m+1)}$, and hence the mixing proportion parameters $\widehat{\tau}_g^{(m+1)}(\mathbf{x}_i)$, as well as the central sequence(s) $\widehat{\boldsymbol{\theta}}_g^{(m+1)}$ and component precision parameter(s) $\widehat{\lambda}^{(m+1)}$. It is clear from (3.4) that the sampling weights can be accounted for by simply multiplying every $\widehat{z}_i^{(m+1)}$ by the corresponding weight w_i. Conversely, in the CM-steps which follow, corresponding formulas for unweighted MEDseq models can be recovered by replacing $\widehat{z}_{i,g}^{(m+1)} w_i$ with $\widehat{z}_{i,g}^{(m+1)}$. The sampling weights for the MVAD data sum to ≈ 711.52, rather than $n = 712$ though the parameter estimates are not affected by multiplying the weights by a constant value. However, to account for the different characteristics of different weighting systems, all relevant subsequent formulas explicitly account for the sum of the weights, with $W = \sum_{i=1}^{n} w_i$, so as to focus on the relative importance of each case as a representative of cases in the population.

3.4.1.1 Estimating the Gating Network Coefficients

The portion of (3.4) corresponding to the gating network, given by

$$\sum_{i=1}^{n} \sum_{g=1}^{G} z_{i,g} w_i \log \tau_g\left(\mathbf{x}_i\right),$$

is of the same form as a MLR model with weights given by w_i, here written with component 1 as the baseline reference level, for identifiability reasons:

$$\log \frac{\tau_g(\mathbf{x}_i)}{\tau_1(\mathbf{x}_i)} = \log \frac{\Pr(z_{i,g} = 1)}{\Pr(z_{i,1} = 1)} = \widetilde{\mathbf{x}}_i \boldsymbol{\beta}_g \; \forall \; g \geq 2, \text{ with } \boldsymbol{\beta}_1 = (0, \ldots, 0)^\top,$$

where $\widetilde{\mathbf{x}}_i = (1, \mathbf{x}_i)$. Thus, methods for fitting such models can be used to maximise the expectation of this term at each iteration to find estimates of the regression parameters in the gating network $\widehat{\boldsymbol{\beta}}_g^{(m+1)}$ and hence the mixing proportions via

$$\widehat{\tau}_g^{(m+1)}(\mathbf{x}_i) = \frac{\exp\left(\widetilde{\mathbf{x}}_i \widehat{\boldsymbol{\beta}}_g^{(m+1)}\right)}{\sum_{g=1}^{G} \exp\left(\widetilde{\mathbf{x}}_i \widehat{\boldsymbol{\beta}}_g^{(m+1)}\right)}.$$

When there are no gating covariates, the mixing proportions are estimated by $\widehat{\tau}_g^{(m+1)} = W^{-1} \sum_{i=1}^{n} \widehat{z}_{i,g}^{(m+1)} w_i$, i.e. the weighted mean of the g-th column of the matrix $\widehat{\mathbf{Z}}^{(m+1)}$. However, $\boldsymbol{\tau}$ can also be constrained to be equal (i.e. $\tau_g = 1/G \; \forall \; g$) across clusters. Thus, situations where $\tau_{i,g} = \tau_g(\mathbf{x}_i)$, $\tau_{i,g} = \tau_g$, or $\tau_{i,g} = 1/G$ are accommodated.

The standard errors of the MLR in the gating network at convergence are not a valid means of assessing the uncertainty of the coefficient estimates as the cluster membership probabilities are estimated rather than fixed and known. Therefore, we adapt the weighted likelihood bootstrap (WLBS) of O'Hagan et al. (2019) to the MEDseq setting. This is easily implemented by multiplying the sampling weights w_i by draws, for each of B samples, from an n-dimensional symmetric uniform Dirichlet distribution. Here, $B = 1000$ is used to ensure stable estimation of the standard errors. To ensure rapid convergence, the estimated $\widehat{\mathbf{Z}}$ matrix under the optimal model fit to the full data set is used to initialise the ECM algorithm when refitting models to each sample with corresponding new likelihood weights. Finally, the standard errors of the gating network coefficients across the B samples are obtained.

3.4.1.2 Estimating the Central Sequences

The location parameter $\boldsymbol{\theta}$ is sometimes referred to as the Fréchet mean or the central sequence. The k-medoids/PAM algorithm, which is closely related to the MEDseq models with certain restrictions imposed (see Section 3.4.2), fixes the estimate of $\widehat{\boldsymbol{\theta}}_g$ to be the medoid of cluster g (Kaufman and Rousseeuw, 1990), i.e. the observed sequence with minimum distance from the others currently assigned to the same cluster. This estimation approach is especially quick as the Hamming distance matrix for the observed sequences is pre-computed. Notably, this greedy search strategy may fail to find the optimum solution.

However, it can be shown — for a single unweighted exponential-distance model based on the Hamming distance — that $\widehat{\boldsymbol{\theta}}$ is given simply by the modal sequence, which is intuitive when $\mathrm{d_H}(\mathbf{s}_i, \mathbf{s}_j)$ is expressed as $T - \sum_{t=1}^{T} \mathbb{1}(s_{i,t} = s_{j,t})$. Thus, the parameter has a natural interpretation. For more complicated distance metrics, the first-improvement algorithm (Hoos and Stützle, 2004) or a genetic algorithm could be used to estimate $\boldsymbol{\theta}$. Notably, the modal sequence need not be an

observed sequence. It is also notable that the Fréchet mean may be non-unique under any of the proposed estimation strategies.

For the $G > 1$ MEDseq setting, under the ECM framework, central sequence position estimates $\widehat{\theta}_{g,t}^{(m+1)}$ are given by $\arg\min_{\vartheta}\left(\sum_{i=1}^{n}\widehat{z}_{i,g}^{(m+1)}w_i\mathbb{1}(s_{i,t}\neq\vartheta)\right)$. Since this expression is independent of the precision parameter(s), it holds for all MEDseq model types, including those which employ the weighted Hamming distance variants. Thus, $\widehat{\boldsymbol{\theta}}_g$ is estimated easily and exactly via a type of weighted mode, which is composed, for each position in the sequence, by the category corresponding to the maximum of the sum of the weights $\widehat{z}_{i,g}^{(m+1)}w_i$ associated with each of the v observed state values. Similarly, the central sequence under a weighted $G = 1$ model is also estimated via a weighted mode, with the weights given only by w_i. Notably, to estimate the Fréchet mean for a MEDseq model of any type without sampling weights, one need only remove w_i from these terms. Note also that $\boldsymbol{\theta}_0$ does not need to be estimated for the models with an explicit noise component as it does not contribute to the likelihood.

3.4.1.3 Estimating the Precision Parameters

It is worth noting, for the exponential-distance model in general, with any distance metric, that the method of moments estimate for λ is equal to the maximum likelihood estimate (MLE). This is because, for fixed $\boldsymbol{\theta}$, the PMF in (3.1) belongs to the exponential family with natural parameter λ. Hence, with $\widehat{\boldsymbol{\theta}}$ already estimated as per Section 3.4.1.2, $\widehat{\lambda}$ ensures that the expected distance of observations from $\widehat{\boldsymbol{\theta}}$ is equal to the observed average distance from $\widehat{\boldsymbol{\theta}}$, since the solution of

$$\frac{\partial\ell(\lambda\,|\,\mathbf{S},\widehat{\boldsymbol{\theta}},\mathrm{d})}{n\partial\lambda}=\frac{\sum_{\mathbf{s}_i\in\mathbf{S}'}\mathrm{d}\big(\mathbf{s}_i,\widehat{\boldsymbol{\theta}}\big)\exp\big(-\lambda\mathrm{d}\big(\mathbf{s}_i,\widehat{\boldsymbol{\theta}}\big)\big)}{\sum_{\mathbf{s}_i\in\mathbf{S}'}\exp\big(-\lambda\mathrm{d}\big(\mathbf{s}_i,\widehat{\boldsymbol{\theta}}\big)\big)}-\frac{1}{n}\sum_{i=1}^{n}\mathrm{d}\big(\mathbf{s}_i,\widehat{\boldsymbol{\theta}}\big)$$

implies

$$\mathbb{E}_{\lambda}\big(\mathrm{d}\big(\mathbf{S},\widehat{\boldsymbol{\theta}}\big)\big)=\frac{\sum_{\mathbf{s}_i\in\mathbf{S}'}\mathrm{d}\big(\mathbf{s}_i,\widehat{\boldsymbol{\theta}}\big)\exp\big(-\lambda\mathrm{d}\big(\mathbf{s}_i,\widehat{\boldsymbol{\theta}}\big)\big)}{\sum_{\mathbf{s}_i\in\mathbf{S}'}\exp\big(-\lambda\mathrm{d}\big(\mathbf{s}_i,\widehat{\boldsymbol{\theta}}\big)\big)}=\overline{\mathrm{d}}\big(\mathbf{S},\widehat{\boldsymbol{\theta}}\big)=\frac{1}{n}\sum_{i=1}^{n}\mathrm{d}\big(\mathbf{s}_i,\widehat{\boldsymbol{\theta}}\big).\quad(3.6)$$

Under the Hamming distance, the value of the expectation in (3.6) holds with any arbitrary reference sequence in place of $\widehat{\boldsymbol{\theta}}$. Hence, with known $\widehat{\boldsymbol{\theta}}$, the MLE for λ for an unweighted single-component CC model can be obtained as follows:

$$\begin{aligned}
\ell\big(\lambda \,|\, \mathbf{S}, \widehat{\boldsymbol{\theta}}, \mathrm{d}_{\mathrm{H}}\big) &= -\lambda n \overline{\mathrm{d}}_{\mathrm{H}}\big(\mathbf{S}, \widehat{\boldsymbol{\theta}}\big) - nT \log\big((v-1)\, e^{-\lambda} + 1\big)\,, \\
\frac{\partial \ell\,(\cdot)}{\partial \lambda} &= \frac{nT\,(v-1)}{e^{\lambda} + (v-1)} - n\overline{\mathrm{d}}_{\mathrm{H}}\big(\mathbf{S}, \widehat{\boldsymbol{\theta}}\big), \\
\therefore\ \widehat{\lambda} &= \log\left((v-1)\left(\frac{T}{\overline{\mathrm{d}}_{\mathrm{H}}\big(\mathbf{S}, \widehat{\boldsymbol{\theta}}\big)} - 1\right)\right),
\end{aligned}$$

which notably relies on the inverse of the average Hamming distance normalised by the sequence length T. However, this can yield a negative value for $\widehat{\lambda}$. Recall that we only consider $\lambda \geq 0$. Since all distances are non-negative and typically not identical, $\frac{\partial \ell(\cdot)}{\partial \lambda}$ is negative $\forall\ \lambda > 0$ in the case where the sufficient statistic $\overline{\mathrm{d}}_{\mathrm{H}}\big(\mathbf{S}, \widehat{\boldsymbol{\theta}}\big) > v^{-1} T\,(v-1)$, with $\lim_{\lambda \to \infty} \frac{\partial \ell(\cdot)}{\partial \lambda} = -n\overline{\mathrm{d}}_{\mathrm{H}}\big(\mathbf{S}, \widehat{\boldsymbol{\theta}}\big)$. Thus,

$$\widehat{\lambda} = \max\left(0, \log\left((v-1)\left(\frac{T}{\overline{\mathrm{d}}_{\mathrm{H}}\big(\mathbf{S}, \widehat{\boldsymbol{\theta}}\big)} - 1\right)\right)\right).$$

When $\overline{\mathrm{d}}_{\mathrm{H}}\big(\mathbf{S}, \widehat{\boldsymbol{\theta}}\big) < v^{-1} T\,(v-1)$, such that $\widehat{\lambda} > 0$, the identity $\log(c\,(a/b - 1)) = \log(c) + \log(a-b) - \log(b)$ is used for numerical stability, otherwise $\widehat{\lambda}$ is set to 0. When sampling weights are included, following the same steps as above yields the corresponding estimate

$$\widehat{\lambda} = \max\left(0, \log(v-1) + \log\left(\frac{TW}{\sum_{i=1}^{n} w_i \mathrm{d}_{\mathrm{H}}\big(\mathbf{s}_i, \widehat{\boldsymbol{\theta}}\big)} - 1\right)\right). \tag{3.7}$$

While $\widehat{\lambda}$ can potentially be estimated as zero, the inclusion of a noise component in the CCN, UCN, CUN, and UUN models makes this explicit, by restricting one of the clusters to have $\widehat{\lambda}_{g,t} = 0\ \forall\ t = 1, \ldots, T$. When $\widehat{\lambda}_{g,t}$ is either estimated as zero or set to zero, estimating the corresponding $\boldsymbol{\theta}_{g,t}$ parameter has no effect on the likelihood.

The ECM algorithm is employed when $G > 1$, in which case the CM-step for $\widehat{\lambda}^{(m+1)}$ under a CC MEDseq mixture model with sampling weights is given by

$$\frac{\partial \ell_c^{\mathbf{w}}(\cdot)}{\partial \lambda} = \frac{T(v-1)\sum_{i=1}^{n}\sum_{g=1}^{G} z_{i,g} w_i}{e^{\lambda} + (v-1)} - \sum_{i=1}^{n}\sum_{g=1}^{G} z_{i,g} w_i \mathrm{d}_{\mathrm{H}}\left(\mathbf{s}_i, \widehat{\boldsymbol{\theta}}_g\right),$$

$$\therefore\ \widehat{\lambda}^{(m+1)} = \max\left(0, \log(v-1) + \log\left(\frac{T\sum_{i=1}^{n}\sum_{g=1}^{G}\widehat{z}_{i,g}^{(m+1)} w_i}{\sum_{i=1}^{n}\sum_{g=1}^{G}\widehat{z}_{i,g}^{(m+1)} w_i \mathrm{d}_{\mathrm{H}}\left(\mathbf{s}_i, \widehat{\boldsymbol{\theta}}_g^{(m+1)}\right)} - 1\right)\right). \tag{3.8}$$

As per (3.7), this requires the current estimate of each component's central sequence. Again, as each case has $w_i = 1$ when there are no sampling weights, one need only drop the w_i and W terms from (3.7) and (3.8) to estimate the precision parameters of unweighted MEDseq models. The expressions for the weighted complete data pseudo likelihoods and corresponding CM-steps for their precision parameters are given for the remaining MEDseq model types in Appendix 3.B.

3.4.2 ECM Initialisation

The MEDseq models share relevant features with the k-medoids/PAM algorithm based on the Hamming distance. Indeed, MEDseq models differ from PAM only in that i) $\boldsymbol{\theta}_g$ is estimated by the modal sequence rather than the medoid, ii) $\boldsymbol{\tau}$ is estimated, or even dependent on covariates via $\tau_g(\mathbf{x}_i)$, rather than constrained to be equal, iii) λ is allowed to vary across clusters and/or time points, iv) a noise component can be included, and v) the ECM algorithm rather than the classification EM algorithm (CEM; Celeux and Govaert, 1992) is used. The C-step of the CEM algorithm employed by PAM uses deterministic assignments $\widetilde{z}_{i,g}^{(m+1)} = \arg\max_g\left(\widehat{z}_{i,g}^{(m+1)}\right)$, for which the denominator in (3.5) need not be evaluated.

In other words, it can be shown that a CC model fitted by CEM (albeit with conditional maximisation steps), with equal mixing proportions and the central sequences estimated by the medoid rather than the modal sequence, is equivalent to k-medoids based on the Hamming distance. Therefore, we apply the k-medoids algorithm to the Hamming distance matrix to initialise the ECM algorithm by obtaining 'hard' starting values for the allocation matrix $\mathbf{Z}$. In particular, we rely on a

weighted version of PAM available in the R package `WeightedCluster` (Studer, 2013). This strategy is less computationally onerous than using multiple random starts and in our experience also achieves better results than using Ward's hierarchical clustering to inform starting values.

For models with an explicit noise component, it is necessary to supply an initial guess of the prior probability τ_0 that observations are noise, and initialise allocations, assuming the last component is the one associated with $\lambda_g = 0$, by multiplying the initial $\mathbf{Z}$ matrix by $1 - \tau_0$ and appending a column in which each entry is τ_0. We caution that the initial τ_0 should not be too large.

3.4.3 Model Selection

In the MEDseq setting, the notion of model selection refers to identifying the optimal number of components G in the mixture and finding the best MEDseq model type in terms of constraints on the precision parameters. Variable selection on the subset of covariates included in the gating network can also improve the fit. For a given set of covariates, one would typically evaluate all model types over a range of G values and choose simultaneously both the model type and G value according to some criterion. Thereafter, different fits with different covariates can be compared according to the same criterion.

The Bayesian Information Criterion (BIC; Schwarz 1978) includes a penalty term which depends on the number of free parameters. Notably, the penalty term in our setting uses $\log(W)$ rather than $\log(N)$. Preliminary analyses (e.g. Section 3.5.1) suggest that this penalty term is not strict enough. Moreover, it is infeasible to calculate an exact, non-asymptotic expression of this criterion due to the difficulty in computing the marginal likelihood in the presence of discrete central sequence parameters and a normalising constant which depends on the precision parameter(s). Indeed, approaches relying on parameter counts may not be fruitful in general for categorical sequence data, although this may simply be an artefact of the (weighted) Hamming distance metrics employed. Nevertheless, the number of free parameters in the BIC penalty term under each MEDseq model type is summarised in Appendix 3.A.

We turn to silhouette analysis approaches to assess the quality of the clustering in terms of internal cluster cohesion, where high cohesion indicates high between-group distances and strong within-group homogeneity. Typically the silhouette width is defined for clustering methods which produce a 'hard' partition (Rousseeuw, 1987), and the average silhouette width (ASW) or weighted average silhouette width (wASW; Studer 2013) is used as a model selection criterion. However, Menardi (2011) introduces the density-based silhouette (DBS) for model-based clustering methods. This allows the 'soft' assignment information to be used, which would be discarded when using the MAP assignments in the computation of the wASW. The empirical DBS for observation i is given by

$$\widehat{dbs}_i = \frac{\log\left(\frac{\widehat{z}_i^0}{\widehat{z}_i^1}\right)}{\max_{h=1,\dots,n}\left(\left|\log\left(\frac{\widehat{z}_h^0}{\widehat{z}_h^1}\right)\right|\right)}. \tag{3.9}$$

As observations are assigned to clusters based on the MAP classification, $\widehat{dbs}_i$ is proportional to the log-ratio of the posterior probability associated with the MAP assignment of observation i (denoted by $\widehat{z}_i^0$) to the maximum posterior probability that the observation belongs to another cluster (denoted by $\widehat{z}_i^1$). Use of the MAP classification implies $0 \leq \widehat{dbs}_i \leq 1 \; \forall \; i$, with high values indicating a well-clustered data point. Ultimately, the mean or the median of $\left\{\widehat{dbs}_1, \dots, \widehat{dbs}_n\right\}$ can be used both as a global quality measure and as a model selection criterion.

We employ a version of this criterion which is modified in two ways, both to identify optimal models and as a means of validating the chosen model. Firstly, we identify a set of crisply assigned observations having $\widehat{z}_i^1$ lower than a tolerance parameter ϵ, here set equal to 10^{-100}. These observations are given $\widehat{dbs}_i$ values of 1 and are excluded from the computation of the maximum in the denominator of (3.9) for reasons of numerical stability. Secondly, we account for the sampling weights by computing a weighted mean density-based silhouette criterion (wDBS). However, neither the wDBS nor wASW criteria are defined for $G = 1$.

Greedy stepwise selection can be used to further refine the models, in terms of guiding the inclusion/exclusion of gating covariates. We propose a bi-directional

search strategy in which each step can potentially consist of adding or removing a covariate or adding or removing a non-noise component. Every potential action is evaluated over all possible model types at each step, rather than considering changing the model type as an action in itself. Changing the gating covariates or changing the number of components can affect the model type, as observed by Murphy and Murphy (2019). While this makes the stepwise search more computationally intensive, it is less likely to miss optimal models as it explores the model space. For steps involving both gating covariates and a noise component, models with both the GN and NGN settings can be evaluated and potentially selected.

A backward stepwise search starts from the model including all covariates that is considered optimal in terms of the number of components G and of the MEDseq model type. On the other hand, a forward stepwise search uses the optimal model with no covariates included as its starting point. In both cases, the algorithm accepts the action yielding the highest increase in the wDBS criterion at each step. The computational benefits of upweighting unique cases and discarding redundant cases are stronger for the forward search, as early steps with fewer covariates are likely to have fewer unique cases across sequence patterns and covariates.

3.5 Analysing the MVAD Data

Results of fitting MEDseq models to the MVAD data are provided in Section 3.5.1. All results were obtained via the associated R package `MEDseq` (Murphy et al., 2019). A comparison against other approaches, including hierarchical, partitional, and model-based clustering methods, is included in Section 3.5.2. A discussion of the insights gleaned from the solution obtained by the optimal MEDseq model is deferred to Section 3.6.

Due to the weighting scheme used by McVicar and Anyadike-Danes (2002), all results are obtained on a version of the data with the first time point removed. Similarly, the term 'all covariates' henceforth refers to all covariates in Table 3.1 except 'Grammar' and 'Location'. While Murphy and Murphy (2019) show that the same covariate can affect more than one part of a mixture of experts model, and in different ways, removing the quantities used to define the weights eases the interpretability of the results.

3.5.1 Application of MEDseq

Weighted MEDseq models are fit across a range of G values, across all 8 model types, with all covariates included in the gating network. The noise components, where applicable, are treated using the GN setting. Figure 3.3 shows the behaviour of the BIC for these models. Similar behaviour is observed for the ICL criterion (Biernacki et al., 2000). Evidently the penalty terms based on parameter counts for these criteria are not large enough. Values of both criteria do not start to decrease until the number of components is very large and models with too many poorly populated components are identified. Thus, both are deemed inadequate as a means of selecting optimal MEDseq models. The k-fold cross-validated likelihood, a model selection criterion which is free from parameter-counting (Smyth, 2000), also penalises insufficiently (with $k = 10$ folds). The Normalised Entropy Criterion (Celeux and Soromenho, 1996), on the other hand, identifies a model with too few components ($G = 2$).

However, using the wDBS criterion (see Figure 3.4), and again discarding solutions with too few components, a reasonable $G = 10$ UCN model is identified as optimal. Thus, the performance of the wDBS criterion in this setting is found to be superior to the various criteria described above. The same model type and number of components are identified as optimal according to the wDBS criterion when the noise components are treated with the NGN setting, and when the same analysis is repeated with no gating covariates at all. Notably, the wDBS criterion yields the same optimal model in both the GN and NGN settings, and the setting with covariates excluded entirely, regardless of whether the weighted mean or the weighted median of $\left\{\widehat{dbs}_1, \ldots, \widehat{dbs}_n\right\}$ is used. Interestingly, $G = 10$ appears to roughly coincide with the elbow in Figure 3.3 where the BIC values begin to plateau. Hence, the BIC can provide useful, if not rigorous, information.

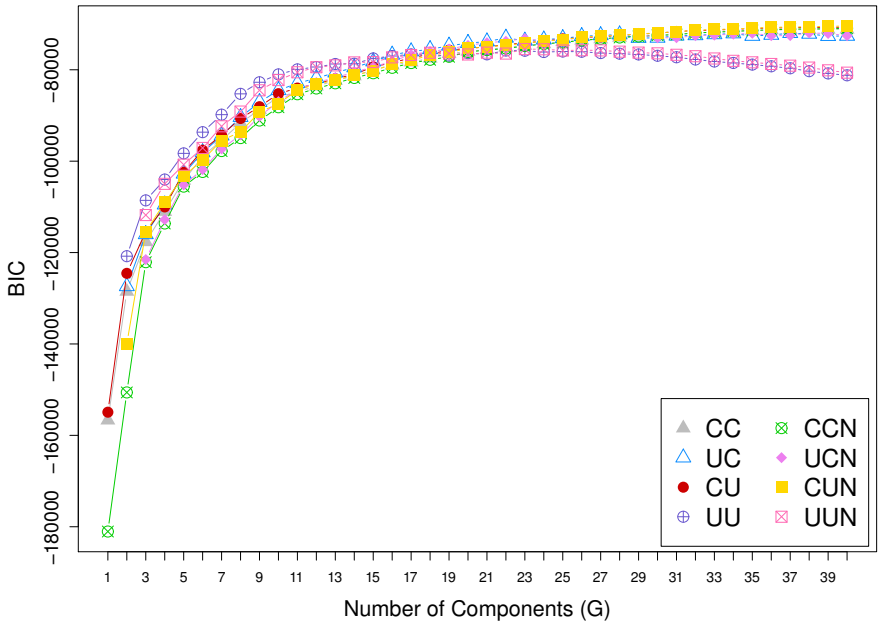


Figure 3.3: BIC values for weighted MEDseq models across a range of G values and model types.

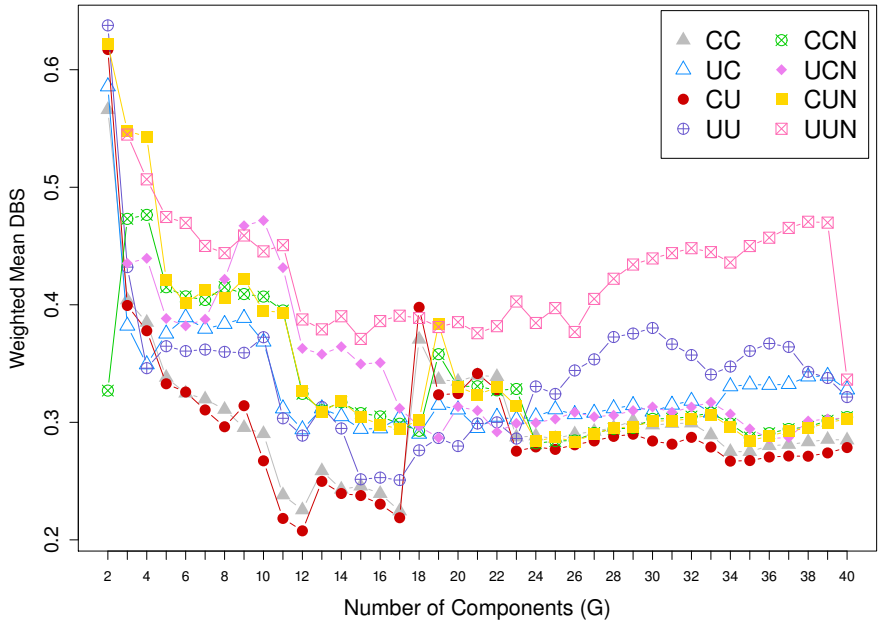


Figure 3.4: wDBS values for weighted MEDseq models across a range of G values and model types.

In refining the model further via greedy stepwise selection, both the forward search (see Table 3.2) and backward search (see Table 3.3) begin with the same number of components and the same model type. Covariates used to define the sampling weights are excluded in both cases. Both searches converge to the same $G = 10$ UCN model with the covariates 'FMPR', 'GCSE5eq', and 'Livboth' in the GN gating network. Under this model, the probability of belonging to the noise component also depends on the included covariates. Notably, the differences between the respective clusterings produced by the models including no covariates, all covariates, and the subset of covariates obtained by stepwise selection are marginal. This can be seen by computing the inner products between all pairs of $\widehat{\mathbf{Z}}$ matrices at convergence. For all three pairwise comparisons, the result, when normalised by its row sums, differs only slightly from the 10-dimensional identity matrix. However, the model uncovered by stepwise selection yields both the highest wDBS value and highest BIC value. In any case, the inclusion of covariates helps inform the interpretation of the clusters, even if not so much their construction.

Table 3.2: Summary of the steps taken to improve the wDBS criterion in the forward direction.

Optimal Step	G	Model Type	Gating Covariates	Gating Type	wDBS
—	10	UCN	—	—	0.4699
Add 'GCSE5eq'	10	UCN	GCSE5eq	GN	0.4724
Add 'Livboth'	10	UCN	FMPR, Livboth	NGN	0.4731
Add 'FMPR'	10	UCN	FMPR, GCSE5eq, Livboth	GN	0.4745
Stop	10	UCN	FMPR, GCSE5eq, Livboth	GN	0.4745

Table 3.3: Summary of the steps taken to improve the wDBS criterion in the backward direction.

Optimal Step	G	Model Type	Gating Covariates	Gating Type	wDBS
—	10	UCN	Catholic, FMPR, Funemp, GCSE5eq, Gender, Livboth	GN	0.4717
Remove 'Catholic'	10	UCN	FMPR, Funemp, GCSE5eq, Gender, Livboth	GN	0.4735
Remove 'Funemp'	10	UCN	FMPR, GCSE5eq, Gender, Livboth	GN	0.4740
Remove 'Gender'	10	UCN	FMPR, GCSE5eq, Livboth	GN	0.4745
Stop	10	UCN	FMPR, GCSE5eq, Livboth	GN	0.4745

These results are not sensitive to the dropping of the first time point or the covariates used to define the sampling weights. Repeating the analysis above with these quantities retained leads to identical inference on the number of components, the MEDseq model type, and the gating covariates identified via stepwise selection. When repeating the analysis with the sampling weights discarded entirely, the

results differ only in that 'Funemp' is identified by stepwise selection rather than 'FMPR'. Finally, in order to ascertain the robustness of the results to a coarsening of the sequences, the analysis was repeated once more with the data subsetted into six-monthly intervals. Again, identical inference was obtained. The ECM algorithm's runtime was not greatly improved in doing so. Indeed, MEDseq models scale more poorly with n rather than T (and also v), as the number of (pseudo) likelihood evaluations for large data sets is more computationally expensive than the number of simple distance evaluations required for long sequences.

3.5.2 Other Clustering Methods

To contrast the MEDseq results with those obtained by other methods, MEDseq models with no covariates and all covariates are compared, in Figure 3.5, against weighted versions of k-medoids, using the R package `WeightedCluster` (Studer, 2013), and Ward's hierarchical clustering, both based on the Hamming distance. Finite mixtures with first-order Markov components, fit via the R package `ClickClust` (Melnykov, 2016b), are also included in the comparison. LCA and latent class regression, fit via the R package `poLCA` (Linzer and Lewis, 2011), are not included, as they encounter computational difficulties due to the explosion in the number of parameters even for $G = 3$. As 'soft' cluster assignment probabilities are not available for k-medoids or Ward's hierarchical clustering, their wDBS values cannot be compared. Thus, Figure 3.5 illustrates a comparison of the wASW values using the MAP classifications where necessary; in so doing, the soft clustering information is discarded.

The `ClickClust` package allows the initial state probabilities to be either estimated or equal to $1/v$ for all categories; both scenarios were considered. Other function arguments were set to their default values. Only the MEDseq models accommodate gating covariates, while all models except the `ClickClust` models accommodate the sampling weights. In all cases, the first time point was dropped. Only the MEDseq model type with the highest wASW for each G value is shown, for clarity. The wASW values for the `ClickClust` models are not shown; they are approximately 0.11 for $G \leq 4$, and negative thereafter. Across all G values, one of the MEDseq model types always outperforms its competitors.

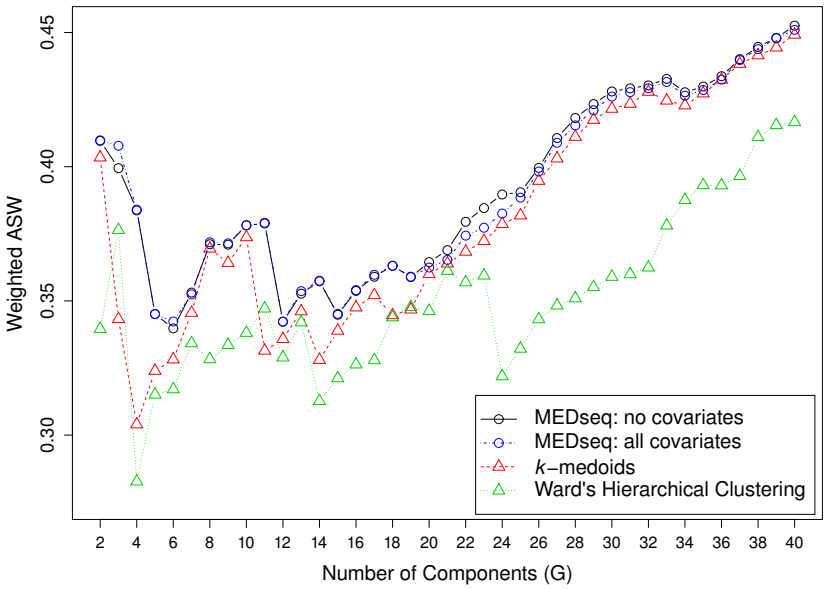


Figure 3.5: Values of the wASW criterion, using Hamming distances, for the best MEDseq model type for each G value with no covariates and all covariates. Corresponding values for weighted k-medoids and weighted Ward's hierarchical clustering are also shown.

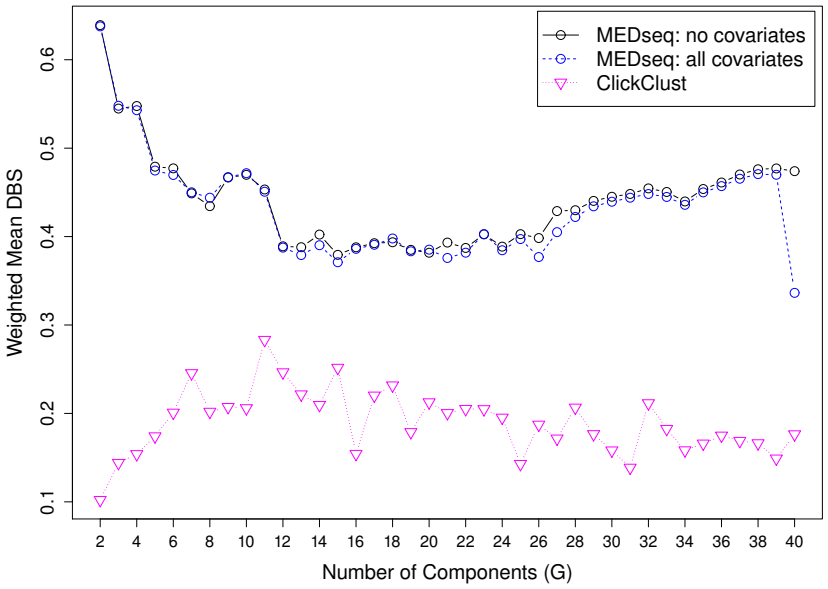


Figure 3.6: Values of the wDBS criterion for the best MEDseq model type at each G value with no covariates and all covariates. Corresponding values for the best `ClickClust` model are also shown.

While the wASW values for the `ClickClust` models being close to zero or even negative shows inferior clustering behaviour, this method also returns a $\widehat{\mathbf{Z}}$ matrix of cluster membership probabilities. Thus, these models can be compared to the MEDseq models in terms of the wDBS criterion also. This is shown in Figure 3.6. Again, only the best model of each type is shown for each G value. The MEDseq models again exhibit the best performance across the entire range of G values. Notably, the optimal `ClickClust` model according to BIC has only $G = 2$ components. An advantage of `ClickClust` is that it allows sequences of unequal lengths, but this is not a concern for the MVAD data.

The R package `seqHMM` (Helske and Helske, 2019) provides tools for fitting mixtures of hidden Markov models, with gating covariates influencing cluster membership probabilities. However, the sampling weights are not accommodated. Such models allow cluster memberships to evolve over time, similar to mixed membership models (Airoldi et al., 2014). They thus cannot be directly compared to MEDseq models. However, we note that the `seqHMM` package provides a pre-fitted model for the MVAD data with 2 clusters — with 3 and 4 hidden states, respectively — and no covariates. Replicating the same model with the first time point omitted and otherwise using the same function arguments yields a model with wDBS=0.50 and wASW=0.23. Otherwise identical `seqHMM` models, including either all covariates or only those deemed optimal for the MEDseq model using stepwise selection, both achieve wDBS=0.47 and wASW=0.23. Notably, these wDBS values are comparable (albeit inferior) to those for MEDseq models with $G = 2$, while the wASW values are much worse.

3.6 Discussion of the MVAD Results

The clusters uncovered by the $G = 10$ UCN model deemed optimal according to the wDBS criterion for the MVAD data are shown in Figure 3.7. Seriation has been applied using the overall Hamming distance matrix (Hahsler et al., 2008) to group observations within clusters for visual clarity. To better inform a discussion of these results, corresponding central sequence estimates are shown in Figure 3.8 and the average time spent in each state by cluster — weighted by the estimated cluster membership probabilities — is shown in Table 3.4, along with the cluster sizes.

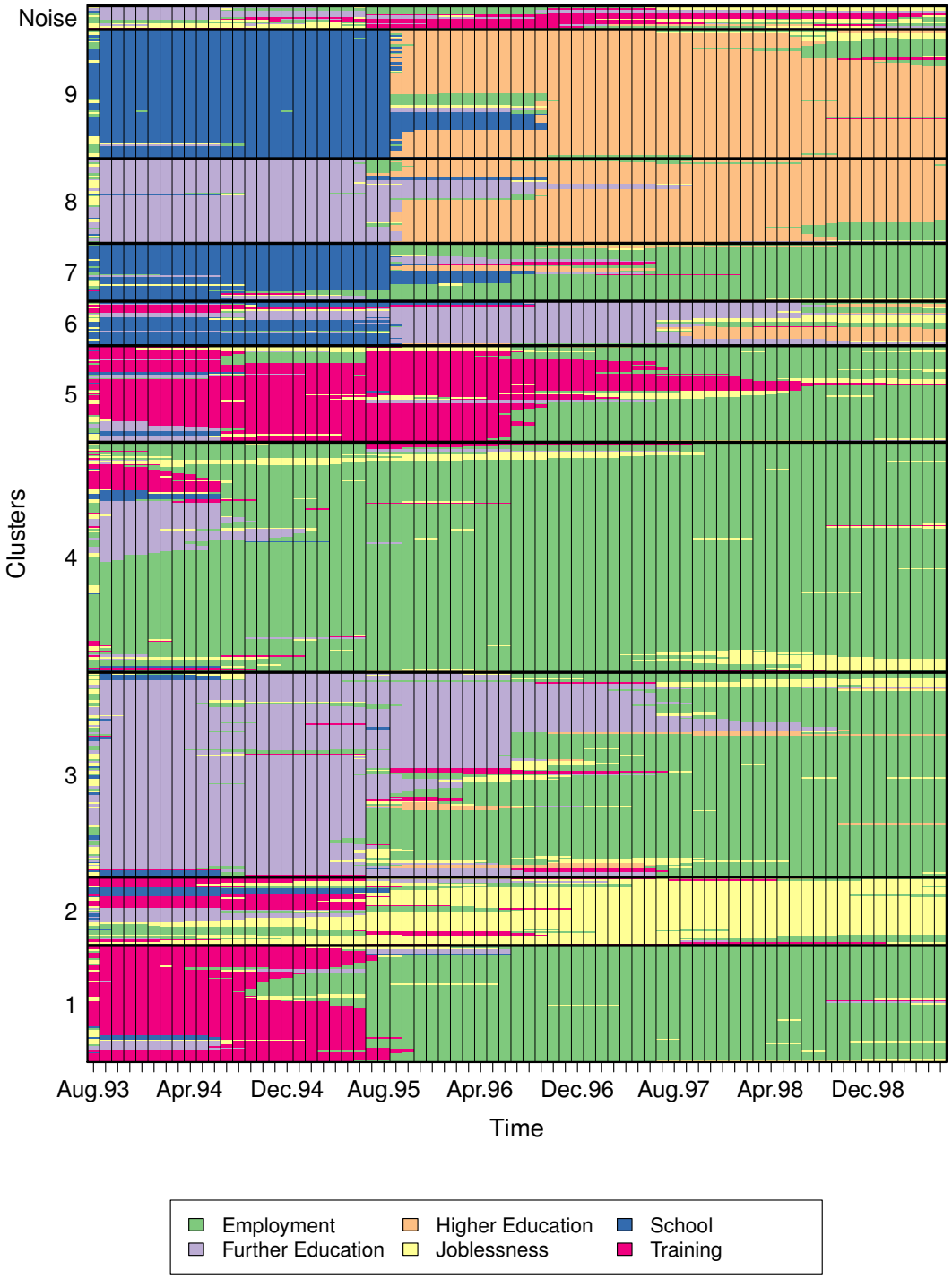


Figure 3.7: Clusters uncovered using the wDBS criterion for the optimal 10-component UCN model with stepwise selection of covariates. Note that the duplicate observations previously discarded during model fitting have been restored for the purposes of this visualisation.

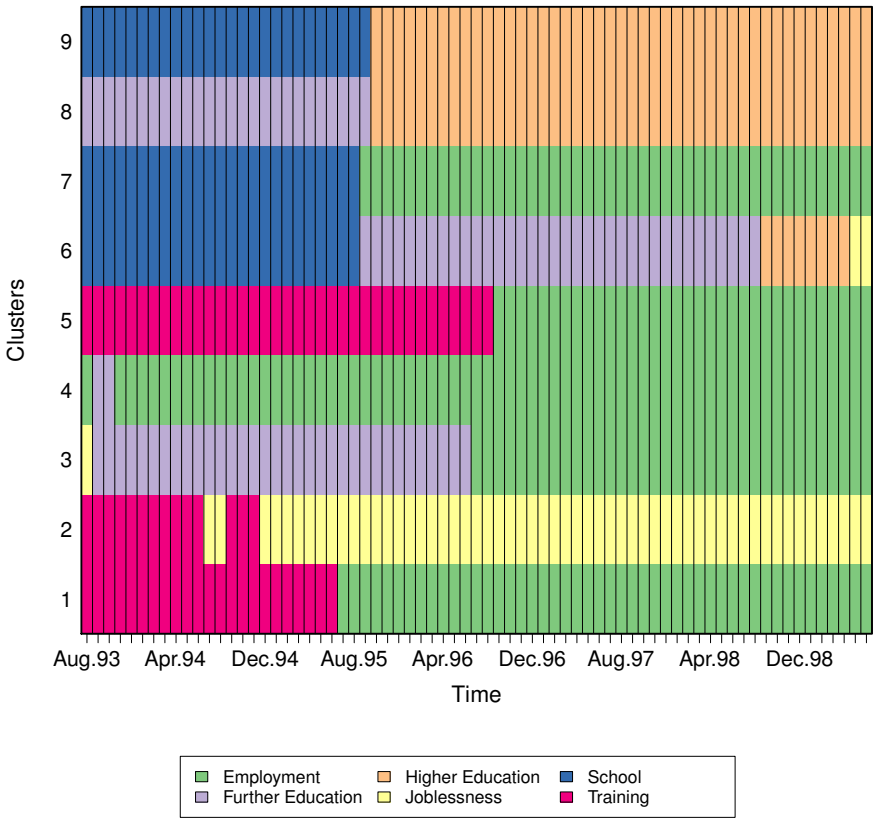


Figure 3.8: Central sequences of the optimal 10-component UCN model with stepwise selection of covariates. The noise component's central sequence is not shown, as it does not contribute to the likelihood.

Table 3.4: Average time (in months) spent in each state by cluster, weighted by the estimated cluster membership probabilities, for the optimal 10-component UCN model with stepwise selection of covariates. Estimated cluster sizes $\widehat{n}_g$ correspond to the MAP partition.

Cluster (g)	$\widehat{n}_g$	EM	FE	HE	JL	SC	TR
1	79	47.79	1.79	0.00	2.29	0.50	18.63
2	46	9.67	4.38	0.00	43.96	2.76	10.23
3	138	33.66	30.96	1.16	3.31	0.73	1.18
4	155	61.83	2.99	0.00	3.55	0.48	2.15
5	65	28.25	2.84	0.00	5.11	0.89	33.91
6	30	6.42	33.27	7.40	4.23	16.36	3.32
7	39	37.68	2.81	2.59	2.68	23.73	1.52
8	57	4.46	27.19	37.79	0.77	0.79	0.00
9	87	4.44	0.50	38.00	1.35	26.41	0.30
Noise	16	14.15	17.66	1.97	14.39	2.37	20.46

This solution tends to group individuals who experienced trajectories that are similar or that differ only for relatively short periods. In particular, the dominating combinations of states experienced over time are clearly identified, and differences in durations and/or age at transition are quite limited in size. Within clusters, substantial reduction of misalignments and/or differences in the durations of states are evident. Ultimately the partition is characterised not only by the sequencing (i.e. the experienced combinations of states), but also by the durations of the states and by the ages at transitions which appear mostly homogeneous within clusters. This can be explained by the fact that cases in the identified groups tended to dedicate the same period of time — 1, 2, or 3 years — to further/higher education and/or training. This is interesting because one might expect the chosen dissimilarity metric to attach higher importance to the sequencing.

The 10-cluster solution for the MVAD data separates individuals who prolonged their studies after the end of compulsory education (clusters 3, 6, 7, 8, and 9) from those who entered the labour market (clusters 1, 4, and 5). The clear division visible for some clusters in Figure 3.7 before and after Sept 1995, when new semesters of further and higher education commenced and the majority of those still remaining in school had eventually left, corresponds to the point in Figure 3.2 after which the entropy declined. Interestingly, individuals who experienced prolonged periods of unemployment are mostly isolated in cluster 2; this is particularly important because the Status Zero Survey originally aimed to identify such 'at risk' subjects.

Notably, the optimal model identified is a UCN model, i.e. one whose precision parameters vary only across clusters, and not across time points. The estimated precision parameters, given in Table 3.5, show that the model captures different degrees of homogeneity in the cluster-specific sequence distributions. The sequences in clusters 1, 8, and 9, for instance, show greater heterogeneity than the more uniformly distributed sequences in clusters 2, 3, and 6. Thus, model selection favours a model based on the simple Hamming distance (albeit weighted differently in each cluster) rather than a more flexible variant which allows different time periods to contribute differently to the overall distance via period-specific weights. Note that the wDBS criterion used to identify the model is not based on parameter counts, meaning the UCN model is not chosen over a more flexible alternative on the basis of parsimony.

Table 3.5: Precision parameters of the optimal 10-component UCN model with stepwise selection of covariates. By definition, $\lambda_g = 0$ characterises the noise component.

Cluster (g)	1	2	3	4	5	6	7	8	9	Noise
$\widehat{\lambda}_g$	3.81	2.22	2.77	3.11	2.84	2.45	3.08	3.49	3.63	0

Clusters 6, 7 and 9 include subjects who continued school for about two years, presumably to retake previously failed examinations or to pursue academic or vocational qualifications. These individuals are split into three groups depending on whether they continued their studies (further education: cluster 6, or higher education: cluster 9) or were employed directly (cluster 7). Clusters 3 and 8 group subjects who entered further education, for about two years (or more, in some cases in the larger cluster 3). Most of the subjects in cluster 3 entered employment directly after further education, whereas the vast majority of those in cluster 8 continued in further education until the end of the observation period.

As for the clusters of individuals who moved quickly to the labour market after the end of compulsory education, it is possible to distinguish between individuals who immediately found a job and remained in employment for most of the observation period (the large cluster 4) and individuals who entered government-supported training schemes (clusters 1 and 5). A further separation is between subjects who were employed after about 2 years of training (cluster 1) and those who participated in training for a much longer period (cluster 5). Importantly, most of the individuals in these two clusters were able to find a job even if some respondents experienced some periods of unemployment.

It is interesting to observe that the cluster of careers dominated by persistent unemployment (cluster 2) is characterized by different experiences at the end of the compulsory education period. Indeed, some subjects entered employment directly after the end of compulsory education but left or lost their job after some months, while some prolonged their education before becoming unemployed. However, the majority entered a training period that did not evolve into steady employment.

The coefficients of the gating network with associated WLBS standard errors are given in Table 3.6, from which a number of interesting effects can be identified. The interpretation of the effects of the covariates is made clearer by virtue of the lower number included after stepwise selection. For completeness, gating network

coefficients and associated WLBS standard errors for the model with all covariates included are provided in Appendix 3.C.

Table 3.6: Multinomial logistic regression coefficients and associated WLBS standard errors (in parentheses) for the gating network of the optimal 10-component UCN model with stepwise selection of covariates.

Cluster	(Intercept)		FMPR		GCSE5eq		Livboth	
2	−0.46	(0.45)	−0.54	(0.56)	−0.22	(0.70)	0.08	(0.51)
3	0.04	(0.39)	0.29	(0.45)	1.30	(0.46)	−0.30	(0.42)
4	0.48	(0.38)	−0.89	(0.43)	−0.25	(0.53)	−0.21	(0.37)
5	−0.16	(0.43)	−0.27	(0.52)	0.17	(0.59)	−0.07	(0.43)
6	−2.38	(0.91)	0.62	(0.63)	2.03	(0.75)	1.43	(0.72)
7	−0.19	(0.49)	−0.66	(0.57)	1.37	(0.59)	−0.03	(0.51)
8	−3.21	(0.50)	0.28	(0.47)	3.34	(0.55)	1.12	(0.49)
9	−1.76	(0.49)	0.71	(0.43)	3.85	(0.50)	0.35	(0.43)
Noise	−1.96	(0.62)	0.37	(0.86)	1.70	(0.93)	−1.07	(0.72)

Relative to the reference cluster (cluster 1), characterised by those who successfully transitioned to stable employment after a short period of training, the positive 'FMPR' coefficients indicate that those whose father's current or most recent job is professional or managerial are more likely to belong to clusters 3, 6, 8, and 9. These clusters are characterised by extended periods of higher education and/or further education. Conversely, clusters 2, 4, 5, and 7 have negative 'FMPR' coefficients. The effect is particularly pronounced for cluster 4, which mostly comprises subjects who immediately entered employment.

Those who achieved 5 or more high GCSE grades are less likely to experience joblessness (cluster 2) or to immediately enter employment (cluster 4). This suggests, as expected, that more academically inclined students tend to further their education in order to improve their job prospects. The largest positive coefficients for this covariate suggest that such students are more likely to pursue higher education after an initial 2-year period in either school (cluster 9) or further education (cluster 8). Additionally, such students are more likely to secure employment immediately after periods of further education (cluster 3) or school (cluster 7), or enter further education after prolonging their time in school (cluster 6).

Unlike the other covariates, 'Livboth' was not measured until June 1995. According to Figure 3.1, this coincides with the point by which most subjects had

turned 18 and left school. Subjects who lived at home with both parents at this point are more likely to have stayed in school beyond the compulsory period and then pursued further education (cluster 6), or to have stayed in school or further education and then pursued higher education (clusters 8 and 9, respectively). Interestingly, such subjects are also more likely to belong to cluster 2, characterised by joblessness. This is the only covariate for which this is the case, perhaps suggesting that subjects who are materially supported by their parents can afford to endure extended periods of unemployment, possibly to research job opportunities in line with their expectations. Conversely, subjects who do not live at home with both parents are more likely to enter the job market sooner, either immediately (cluster 4), after long periods of training (cluster 5), or after short periods in school (cluster 7) or further education (cluster 3). However, the effects of the 'Livboth' coefficients appear to be slight.

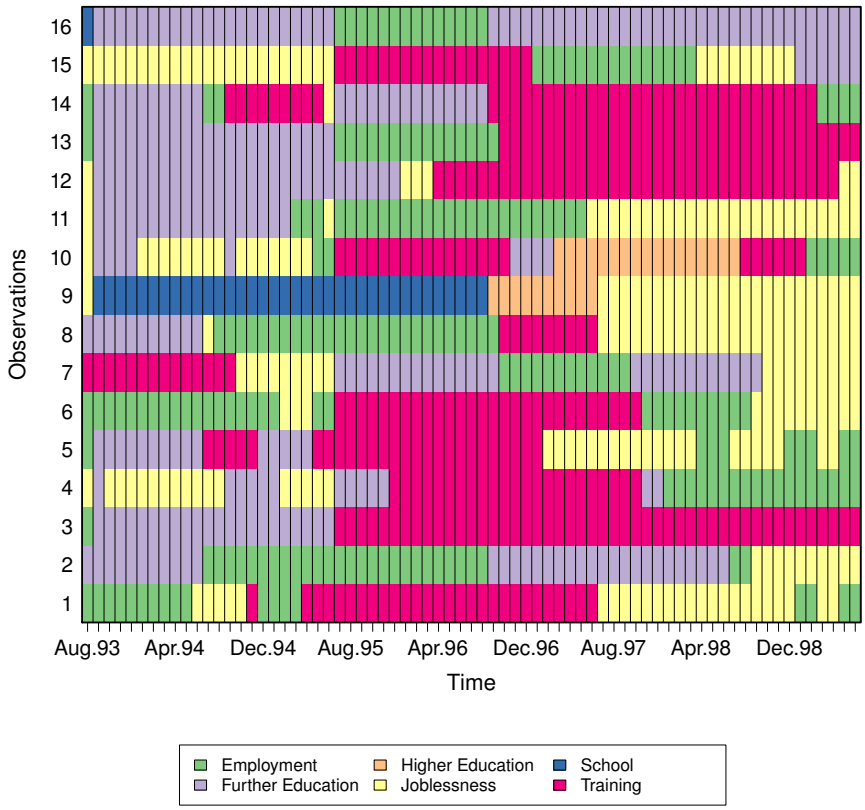


Figure 3.9: Observations assigned to the noise component of the optimal 10-component UCN model with stepwise selection of covariates.

The optimal $G = 10$ UCN model contains a noise component, which allows the remaining non-noise clusters to be modelled more clearly. Figure 3.9 focuses on this noise component, which soaks up subjects who don't neatly fit into any of the defined clusters and transition frequently between states. This includes transitions in and out of education and in and out of employment. The only covariate with a negative coefficient associated with the noise component is 'Livboth'. It is likely that subjects living at home are given a strong sense of direction by the influence of their parents and benefit from familial stability in terms of a lack of disruption to their parents' marriage due to divorce or death.

3.7 Conclusion

In McVicar and Anyadike-Danes (2002), Ward's hierarchical clustering algorithm is applied to an OM dissimilarity matrix to identify relevant patterns in the data. Notably, reference is not made to the associated covariates until the uncovered clustering structure is investigated. In particular, MLR is used to relate the assignments of the trajectories to clusters to a set of baseline covariates. It is also worth noting that the sampling weights are incorporated only in the MLR stage and not in the clustering itself. In other words, weights are incorporated only in the equivalent of the gating network. This is arguably a three-stage approach, comprising the computation of pairwise string distances using OM (or some other distance metric), the hierarchical or partition-based clustering, and the MLR.

MEDseq models, on the other hand, represent a more coherent model-based clustering approach. The sequences are modelled directly using a finite mixture of exponential-distance models, with the Hamming distance and weighted generalisations thereof employed as the distance metric. A range of precision parameter settings have been explored to allow different time points contribute differently to the overall distance. Thus, varying degrees of parsimony are accommodated. Sampling weights are accounted for by weighting each observation's contribution to the pseudo likelihood. Dependency on covariates is introduced by relating the cluster membership probabilities to covariates under the mixture of experts framework. Thus, MEDseq models treat the weights, the relation of covariates to clusters, and

the clustering itself simultaneously. Model selection in the MEDseq setting identifies a reasonable solution for the MVAD data and shows that clustering the sequence trajectories in a holistic manner allows new insights to be gleaned from these data.

Opportunities for future research are varied and plentiful. Co-clustering approaches could be used to simultaneously provide clusters of the observed sequence trajectories and the time periods. While this would require the use of the CEM or stochastic EM algorithms (Govaert and Nadif, 2013), such an approach could be especially useful for the MEDseq models (CU, UU, CUN, and UUN) which weight the Hamming distance by period-specific precision parameters, as it could reduce the number of within-cluster precision parameters required to $1 < T^{\star} \leq T$. Indeed, parsimony has been achieved in a similar fashion by Melnykov (2016a) in the context of finite mixtures with Markov components. In particular, using co-clustering approaches which respect the ordering of the sequences by restricting the column-wise clusters to form contingent blocks would be desirable from an interpretability point of view, though not strictly necessary due to the invariance of the Hamming distance to permutations of the time periods.

It may also be of interest for other applications to extend the MEDseq models to accommodate sequences of different lengths, for which the Hamming distance is not defined. These different lengths could be attributable to missing data, either by virtue of sequences not starting on the same date, shorter follow-up time for some subjects, or non-response for some time points. While the Hamming distance is only defined for equal-length strings, adapting the MEDseq models to such a setting would be greatly simplified if aligning the sequences of different lengths is straightforward. Another limitation of MEDseq models is that time-varying covariates are not accommodated. However, neither of these concerns are relevant for the MVAD data.

MEDseq models implicitly assume substitution-cost matrices with zero along the diagonal and a single value common to all other entries. The relationship between the exponent of an exponential-distance model based on the Hamming distance and the Hamming distance itself (which has a single substitution cost, typically equal to 1) is apparent from the fact that multiplying the substitution-cost matrix by any scalar — as per normalised variants of the Hamming distance (Elzinga,

2007; Gabadinho et al., 2011) — yields the same model, because its value is absorbed into the precision parameter. This is also the case for models employing weighted Hamming distance variants under which the precision parameters, and hence the otherwise common substitution costs, vary across clusters and/or time points. However, all model types in the MEDseq family cannot account for situations in which some states are more different than others — e.g. one where the cost associated with moving from employment to joblessness is assumed to be greater than the cost associated with moving from school to training — as they assume that substitution costs are the same between each pair of states. Such concerns are most pronounced when there is an explicit ordering to the states, e.g. education levels (Studer and Ritschard, 2016).

Hence, another potential extension is to consider MEDseq models with an alternative distance measure, particularly OM. This would require the subjective specification, or estimation, of the $v(v-1)/2$ off-diagonal entries of symmetric substitution-cost matrices. Potentially, as per the range of precision parameter settings in the MEDseq model family, the substitution-cost matrices could also be allowed to vary across clusters and/or time points. However, the normalising constant under an exponential-distance model using OM depends both on the heterogeneous substitution costs and on θ and is not available in closed form, thereby greatly complicating model fitting. Indeed, the dependence on θ renders even offline pre-computation of the normalising constant infeasible for even moderately large T or v. Considering insertions and deletions also would present further challenges. Truncation of the sum over all sequences or an importance sampling approach could be used to address the intractability. In any case, some level of approximation would be required, while the ECM algorithm for MEDseq models based on the Hamming distance is exact.

As well as removing the normalising constant's dependence on θ, another positive consequence of the homogeneity of substitution costs with respect to pairs of states under the Hamming distance is that the ECM algorithm used for parameter estimation scales well with v, the size of the alphabet. Though restrictive, having only one parameter associated with each substitution-cost matrix, regardless of its dimensions, helps address concerns about overparameterisation, especially when the substitution costs implied by the precision parameter(s) vary across clusters

and/or time points (Studer and Ritschard, 2016). A complete characterisation of the implicit substitution costs for the various weighted Hamming distance variants in the MEDseq model family, as well as OM and the DHD, is given in Appendix 3.D.

Furthermore, it is likely that results on the MVAD data would not differ greatly with OM (with state-dependent substitution costs) used in place of the Hamming distance, particularly for models where λ varies across clusters and/or time points, save for a solution with potentially fewer clusters being found. Ultimately, the weighted Hamming distance variants preserve the timing of transitions, by virtue of prohibiting insertions and deletions, but amount to improved substitution costs reflecting replacements of states.

Overall, the MEDseq models appear promising from the perspective of reconciling the distance-based and model-based cultures within the SA community. The results on the MVAD data are encouraging; they seem to suggest that the different precision parameter settings of different MEDseq models adequately address the misalignment problem inherent in the use of the Hamming distance. It remains to be seen if this holds for more turbulent sequence data, e.g. those related to employment activities tracked over longer periods.

References

Abbott, A. and J. Forrest (1986). Optimal matching methods for historical sequences. *Journal of Interdisciplinary History 16*(3), 471–494. 98, 148

Abbott, A. and A. Hrycak (1990). Measuring resemblance in sequence data: an optimal matching analysis of musician's careers. *American Journal of Sociology 96*(1), 145–185. 98, 148

Agresti, A. (2002). *Categorical Data Analysis*. New York: John Wiley & Sons. 100

Airoldi, E. M., D. M. Blei, E. A. Erosheva, and S. E. Fienberg (2014). *Handbook of Mixed Membership Models and Their Applications*. Chapman and Hall/CRC Press. 125

Banfield, J. and A. E. Raftery (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics 49*(3), 803–821. 99

Biernacki, C., G. Celeux, and G. Govaert (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*(7), 719–725. 120

Billari, F. C. (2001). The analysis of early life courses: complex description of the transition to adulthood. *Journal of Population Research 18*(2), 119–142. 102

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer. 109

Böhning, D., E. Dietz, R. Schaub, P. Schlattmann, and B. G. Lindsay (1994). 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. 111

Bouveyron, C., G. Celeux, T. B. Murphy, and A. E. Raftery (2019). *Model-Based Clustering and Classification for Data Science: With Applications in R*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press. 99

Celeux, G. and G. Govaert (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis 14*(3), 315–332. 116

Celeux, G. and G. Soromenho (1996). An entropy criterion for assessing the number of clusters in a mixture model. *Journal of Classification 13*, 195–212. 120

Chambers, R. L. and C. J. Skinner (2003). *Analysis of Survey Data*. Chichester: John Wiley & Sons. 107

Dayton, C. M. and G. B. Macready (1988). Concomitant-variable latent-class models. *Journal of the American Statistical Association 83*(401), 173–178. 100

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 39*(1), 1–38. 110

Elzinga, C. H. (2007). Sequence analysis: metric representations of categorical time series. Technical report, Department of Social Science Research Methods, Vrije Universiteit, Amsterdam. 133

Gabadinho, A., G. Ritschard, N. S. Müller, and M. Studer (2011). Analyzing and visualizing state sequences in R with TraMineR. *Journal of Statistical Software 40*(4), 1–37. 102, 134

Gormley, I. C. and S. Frühwirth-Schnatter (2019). Mixtures of experts models. In S. Frühwirth-Schnatter, G. Celeux, and C. P. Robert (Eds.), *Handbook of Mixture Analysis*, Chapter 12, pp. 279–316. London: Chapman and Hall/CRC Press. 99, 109

Govaert, G. and M. Nadif (2013). *Co-Clustering: models, algorithms and applications*. ISTE-Wiley. 133

Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics 27*(4), 857–871. 107

Hahsler, M., K. Hornik, and C. Buchta (2008). Getting things in order: an introduction to the R package seriation. *Journal of Statistical Software 25*(3), 1–34. 125

Hamming, R. W. (1950). Error detecting and error correcting codes. *The Bell System Technical Journal 29*(2), 147–160. 100

Helske, S. and J. Helske (2019). Mixture hidden Markov models for sequence data: the seqHMM package in R. *Journal of Statistical Software 88*(3), 1–32. 125

Helske, S., J. Helske, and M. Eerola (2016). Analysing complex life sequence data with hidden markov modeling. In G. Ritschard and M. Studer (Eds.), *Proceedings of International Conference on Sequence Analysis and Related Methods*, pp. 209–240. 101

Hoos, H. and T. Stützle (2004). *Stochastic Local Search: Foundations & Applications*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. 113

Irurozki, E., B. Calvo, and J. A. Lozano (2019). Mallows and generalized Mallows model for matchings. *Bernoulli 25*(2), 1160–1188. 106

Jacobs, R. A., M. I. Jordan, S. J. Nowlan, and G. E. Hinton (1991). Adaptive mixtures of local experts. *Neural Computation 3*(1), 79–87. 109

Kaufman, L. and P. J. Rousseeuw (1990). Partitioning around medoids (program PAM). In *Finding Groups in Data: An Introduction to Cluster Analysis*, pp. 68–125. New York: John Wiley & Sons. 100, 113

Lazarsfeld, P. F. and N. W. Henry (1968). *Latent Structure Analysis*. Boston: Houghton Mifflin. 100

Lesnard, L. (2010). Setting cost in optimal matching to uncover contemporaneous socio-temporal patterns. *Sociological Methods & Research 38*(3), 389–419. 106, 148

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady 10*(8), 707–710. 98

Linzer, D. A. and J. B. Lewis (2011). poLCA: an R package for polytomous variable latent class analysis. *Journal of Statistical Software 42*(10), 1–29. 123

Mallows, C. L. (1957). Non-null ranking models. *Biometrika 44*(1/2), 114–130. 105

McVicar, D. (2000). Status 0 four years on: young people and social exclusion in Northern Ireland. *Labour Market Bulletin 14*, 114–119. 100, 102

McVicar, D. and M. Anyadike-Danes (2002). Predicting successful and unsuccessful transitions from school to work by using sequence methods. *Journal of the Royal Statistical Society: Series A (Statistics in Society) 165*(2), 317–334. 100, 101, 102, 109, 119, 132

Melnykov, V. (2016a). Model-based biclustering of clickstream data. *Computational Statistics and Data Analysis 93*(C), 31–45. 101, 133

Melnykov, V. (2016b). ClickClust: an R package for model-based clustering of categorical sequences. *Journal of Statistical Software 74*(9), 1–34. 123

Menardi, G. (2011). Density-based silhouette diagnostics for clustering methods. *Statistics and Computing 21*(3), 295–308. 118

Meng, X. L. and D. R. Rubin (1993). Maximum likelihood estimation via the ECM algorithm: a general framework. *Biometrika 80*(2), 267–278. 111

Müller, N. S., M. Studer, and G. Ritschard (2007). Classification de parcours de vie à l'aide de l'optimal matching. In *XIVe Rencontre de la Société francophone de classification (SFC 2007), Paris, 5–7 septembre 2007*, pp. 157–160. 151

Murphy, K. and T. B. Murphy (2019). Gaussian parsimonious clustering models with covariates and a noise component. *Advances in Data Analysis and Classification*, 1–33. URL https://doi.org/10.1007/s11634-019-00373-8. 99, 110, 119

Murphy, K., T. B. Murphy, R. Piccarreta, and I. C. Gormley (2019). `MEDseq`*: mixtures of exponential-distance models with covariates*. R package version 1.1.0. 102, 119

Murphy, T. B. and D. Martin (2003). Mixtures of distance-based models for ranking data. *Computational Statistics and Data Analysis 41*(3–4), 645–655. 105

O'Hagan, A., T. B. Murphy, L. Scrucca, and I. C. Gormley (2019). Investigation of parameter uncertainty in clustering using a Gaussian mixture model via jackknife, bootstrap and weighted likelihood bootstrap. *Computational Statistics 34*(4), 1779–1813. 113

Pamminger, C. and S. Frühwirth-Schnatter (2010). Model-based clustering of categorical time series. *Bayesian Analysis 5*(2), 345–368. 101

Piccarreta, R. and M. Studer (2019). Holistic analysis of the life course: methodological challenges and new perspectives. *Advances in Life Course Research 41*, 100251. 109

R Core Team (2019). *R: a language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. 102

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics 20*, 53–65. 118

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*(2), 461–464. 117

Smyth, P. (2000). Model selection for probabilistic clustering using cross-validated likelihood. *Statistics and Computing 10*(1), 63–72. 120

Studer, M. (2013). WeightedCluster library manual: a practical guide to creating typologies of trajectories in the social sciences with R. Technical report, LIVES Working Papers 24. 117, 118, 123

Studer, M. and G. Ritschard (2016). What matters in differences between life trajectories: a comparative review of sequence dissimilarity measures. *Journal of the Royal Statistical Society: Series A (Statistics in Society) 179*(2), 481–511. 98, 134, 135

Ward, J. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association 58*(301), 236–244. 100

Wu, L. L. (2000). Some comments on sequence analysis and optimal matching methods in sociology: review and prospect. *Sociological Methods & Research 29*(1), 41–64. 101

3.A Appendix 1

The MEDseq Model Family: Parameter Counts

The models in the MEDseq family differ only in their treatments of the precision parameters, which differentiate the Hamming distance and the weighted variants thereof. While the BIC has been shown to be inadequate as a means of selecting MEDseq models, Table 3.A.1 nevertheless summarises the number of free parameters under each MEDseq model type, in order to demonstrate the increasing level of complexity in moving from the most parsimonious CCN model to the most heavily parameterised UU model. The number of estimated parameters for each component's central sequence is treated as the sequence length T, leading to the strictest possible penalty. Note that central sequence parameters corresponding to time points with estimated or fixed precision parameter values of 0 are not counted. Note also that *estimated* precision parameter values of 0 are counted, but precision parameters fixed at 0 associated with the noise component are not counted. The number of gating network parameters is not accounted for in Table 3.A.1; when there are gating covariates, there are $(r+1) \times G$ extra parameters, where $r+1$ is the dimension of the associated design matrix, including the intercept term. When mixing proportions are constrained to be equal, there are no additional parameters for models without a noise component and one additional parameter for models with a noise component; otherwise there are $G-1$ additional parameters.

Table 3.A.1: Number of estimated parameters under each MEDseq model type. Models with names ending with the letter N, indicating the presence of a noise component for which the single precision parameter is fixed to 0, behave like the corresponding model without this component for all other components. Thus, λ and all subscript variants thereof refer to the non-noise components only.

Model	Precision	λ_g (Clusters)	λ_t (Time Points)	Number of Parameters	
				Central Sequence(s)	Precision
CC	$\lambda_{g,t} = \lambda$	Constrained	Constrained	$GT\mathbb{1}(\lambda \neq 0)$	1
CCN				$(G-1)\,T\mathbb{1}(\lambda \neq 0)$	$\mathbb{1}(G > 1)$
UC	$\lambda_{g,t} = \lambda_g$	Unconstrained	Constrained	$T\sum_{g=1}^{G}\mathbb{1}(\lambda_g \neq 0)$	G
UCN				$T\sum_{g=1}^{G-1}\mathbb{1}(\lambda_g \neq 0)$	$G-1$
CU	$\lambda_{g,t} = \lambda_t$	Constrained	Unconstrained	$G\sum_{t=1}^{T}\mathbb{1}(\lambda_t \neq 0)$	T
CUN				$(G-1)\sum_{t=1}^{T}\mathbb{1}(\lambda_t \neq 0)$	$\mathbb{1}(G > 1)\,T$
UU	$\lambda_{g,t} = \lambda_{g,t}$	Unconstrained	Unconstrained	$\sum_{g=1}^{G}\sum_{t=1}^{T}\mathbb{1}(\lambda_{g,t} \neq 0)$	GT
UUN				$\sum_{g=1}^{G-1}\sum_{t=1}^{T}\mathbb{1}(\lambda_{g,t} \neq 0)$	$(G-1)\,T$

3.B Appendix 2

Further Details on Estimating MEDseq Models

Weighted complete data pseudo likelihood functions for all model types in the MEDseq family are given in Table 3.B.1. Table 3.B.2 outlines the corresponding CM-steps for the precision parameter(s). The sampling weights are accounted for in all cases. The CM-step formulas can be simplified somewhat for unweighted models. Recall that the first letter of the model name denotes whether the precision parameters are constrained/unconstrained across clusters, the second denotes the same across time points (i.e. sequence positions), and model names ending with the letter N include a noise component. All models are written as though gating network covariates $\mathbf{x}_i$ are included. Moreover, models with a noise component are written in the GN rather than NGN form, i.e. it assumed that the covariates affect the mixing proportions of the noise component rather than τ_0 being constant (see Section 3.4.1). All derivations closely follow the same steps as in Section 3.4.1.3 for the CC model. We note again that the corresponding central sequence parameters $\theta_{g,t}$ must be estimated when $\widehat{\lambda}_{g,t}$ is non-zero. In particular — taking the UU model as an example — all state values in the t-th sequence position within component g are identical to $\widehat{\theta}_{g,t}$ when the corresponding denominator in Table 3.B.2 evaluates to zero, such that $\widehat{\lambda}_{g,t} \to \infty$.

Table 3.B.1: Weighted complete data pseudo likelihood functions for all MEDseq model types, which differ according to the constraints imposed on the precision parameters across clusters and/or time points. The expressions for the various weighted Hamming distance metric variants employed are given in full.

Model	Weighted Complete Data Pseudo Likelihood
CC	$\prod_{i=1}^{n}\left[\prod_{g=1}^{G}\left(\tau_g(\mathbf{x}_i)\,\frac{\exp\left(-\lambda\sum_{t=1}^{T}\mathbb{1}\left(s_{i,t}\neq\theta_{g,t}\right)\right)}{\left((v-1)e^{-\lambda}+1\right)^{T}}\right)^{z_{i,g}}\right]^{w_i}$
UC	$\prod_{i=1}^{n}\left[\prod_{g=1}^{G}\left(\tau_g(\mathbf{x}_i)\,\frac{\exp\left(-\lambda_g\sum_{t=1}^{T}\mathbb{1}\left(s_{i,t}\neq\theta_{g,t}\right)\right)}{\left((v-1)e^{-\lambda_g}+1\right)^{T}}\right)^{z_{i,g}}\right]^{w_i}$
CU	$\prod_{i=1}^{n}\left[\prod_{g=1}^{G}\left(\tau_g(\mathbf{x}_i)\,\frac{\exp\left(-\sum_{t=1}^{T}\lambda_t\mathbb{1}\left(s_{i,t}\neq\theta_{g,t}\right)\right)}{\prod_{t=1}^{T}\left((v-1)e^{-\lambda_t}+1\right)}\right)^{z_{i,g}}\right]^{w_i}$
UU	$\prod_{i=1}^{n}\left[\prod_{g=1}^{G}\left(\tau_g(\mathbf{x}_i)\,\frac{\exp\left(-\sum_{t=1}^{T}\lambda_{g,t}\mathbb{1}\left(s_{i,t}\neq\theta_{g,t}\right)\right)}{\prod_{t=1}^{T}\left((v-1)e^{-\lambda_{g,t}}+1\right)}\right)^{z_{i,g}}\right]^{w_i}$
CCN	$\prod_{i=1}^{n}\left[\prod_{g=1}^{G-1}\left(\tau_g(\mathbf{x}_i)\,\frac{\exp\left(-\lambda\sum_{t=1}^{T}\mathbb{1}\left(s_{i,t}\neq\theta_{g,t}\right)\right)}{\left((v-1)e^{-\lambda}+1\right)^{T}}\right)^{z_{i,g}}\left(\frac{\tau_0(\mathbf{x}_i)}{v^{T}}\right)^{z_{i,0}}\right]^{w_i}$
UCN	$\prod_{i=1}^{n}\left[\prod_{g=1}^{G-1}\left(\tau_g(\mathbf{x}_i)\,\frac{\exp\left(-\lambda_g\sum_{t=1}^{T}\mathbb{1}\left(s_{i,t}\neq\theta_{g,t}\right)\right)}{\left((v-1)e^{-\lambda_g}+1\right)^{T}}\right)^{z_{i,g}}\left(\frac{\tau_0(\mathbf{x}_i)}{v^{T}}\right)^{z_{i,0}}\right]^{w_i}$
CUN	$\prod_{i=1}^{n}\left[\prod_{g=1}^{G-1}\left(\tau_g(\mathbf{x}_i)\,\frac{\exp\left(-\sum_{t=1}^{T}\lambda_t\mathbb{1}\left(s_{i,t}\neq\theta_{g,t}\right)\right)}{\prod_{t=1}^{T}\left((v-1)e^{-\lambda_t}+1\right)}\right)^{z_{i,g}}\left(\frac{\tau_0(\mathbf{x}_i)}{v^{T}}\right)^{z_{i,0}}\right]^{w_i}$
UUN	$\prod_{i=1}^{n}\left[\prod_{g=1}^{G-1}\left(\tau_g(\mathbf{x}_i)\,\frac{\exp\left(-\sum_{t=1}^{T}\lambda_{g,t}\mathbb{1}\left(s_{i,t}\neq\theta_{g,t}\right)\right)}{\prod_{t=1}^{T}\left((v-1)e^{-\lambda_{g,t}}+1\right)}\right)^{z_{i,g}}\left(\frac{\tau_0(\mathbf{x}_i)}{v^{T}}\right)^{z_{i,0}}\right]^{w_i}$

Table 3.B.2: CM-steps for the precision parameter(s) for all MEDseq model types, which differ according to the constraints imposed across clusters and/or time points.

Model	Precision Parameter CM-steps
CC	$\widehat{\lambda}^{(m+1)}=\max\left(0,\log(v-1)+\log\left(\frac{T\sum_{i=1}^{n}\sum_{g=1}^{G}\widehat{z}_{i,g}^{(m+1)}w_i}{\sum_{i=1}^{n}\sum_{g=1}^{G}\widehat{z}_{i,g}^{(m+1)}w_i\mathrm{d_H}\left(\mathbf{s}_i,\widehat{\boldsymbol{\theta}}_g^{(m+1)}\right)}-1\right)\right)$
UC	$\widehat{\lambda}_g^{(m+1)}=\max\left(0,\log(v-1)+\log\left(\frac{T\sum_{i=1}^{n}\widehat{z}_{i,g}^{(m+1)}w_i}{\sum_{i=1}^{n}\widehat{z}_{i,g}^{(m+1)}w_i\mathrm{d_H}\left(\mathbf{s}_i,\widehat{\boldsymbol{\theta}}_g^{(m+1)}\right)}-1\right)\right)$
CU	$\widehat{\lambda}_t^{(m+1)}=\max\left(0,\log(v-1)+\log\left(\frac{\sum_{i=1}^{n}\sum_{g=1}^{G}\widehat{z}_{i,g}^{(m+1)}w_i}{\sum_{i=1}^{n}\sum_{g=1}^{G}\widehat{z}_{i,g}^{(m+1)}w_i\mathbb{1}\left(s_{i,t}\neq\widehat{\theta}_{g,t}^{(m+1)}\right)}-1\right)\right)$
UU	$\widehat{\lambda}_{g,t}^{(m+1)}=\max\left(0,\log(v-1)+\log\left(\frac{\sum_{i=1}^{n}\widehat{z}_{i,g}^{(m+1)}w_i}{\sum_{i=1}^{n}\widehat{z}_{i,g}^{(m+1)}w_i\mathbb{1}\left(s_{i,t}\neq\widehat{\theta}_{g,t}^{(m+1)}\right)}-1\right)\right)$
CCN	$\widehat{\lambda}^{(m+1)}=\max\left(0,\log(v-1)+\log\left(\frac{T\sum_{i=1}^{n}\sum_{g=1}^{G-1}\widehat{z}_{i,g}^{(m+1)}w_i}{\sum_{i=1}^{n}\sum_{g=1}^{G-1}\widehat{z}_{i,g}^{(m+1)}w_i\mathrm{d_H}\left(\mathbf{s}_i,\widehat{\boldsymbol{\theta}}_g^{(m+1)}\right)}-1\right)\right)$
UCN	$\widehat{\lambda}_g^{(m+1)}=\max\left(0,\log(v-1)+\log\left(\frac{T\sum_{i=1}^{n}\widehat{z}_{i,g}^{(m+1)}w_i}{\sum_{i=1}^{n}\widehat{z}_{i,g}^{(m+1)}w_i\mathrm{d_H}\left(\mathbf{s}_i,\widehat{\boldsymbol{\theta}}_g^{(m+1)}\right)}-1\right)\right)$
CUN	$\widehat{\lambda}_t^{(m+1)}=\max\left(0,\log(v-1)+\log\left(\frac{\sum_{i=1}^{n}\sum_{g=1}^{G-1}\widehat{z}_{i,g}^{(m+1)}w_i}{\sum_{i=1}^{n}\sum_{g=1}^{G-1}\widehat{z}_{i,g}^{(m+1)}w_i\mathbb{1}\left(s_{i,t}\neq\widehat{\theta}_{g,t}^{(m+1)}\right)}-1\right)\right)$
UUN	$\widehat{\lambda}_{g,t}^{(m+1)}=\max\left(0,\log(v-1)+\log\left(\frac{\sum_{i=1}^{n}\widehat{z}_{i,g}^{(m+1)}w_i}{\sum_{i=1}^{n}\widehat{z}_{i,g}^{(m+1)}w_i\mathbb{1}\left(s_{i,t}\neq\widehat{\theta}_{g,t}^{(m+1)}\right)}-1\right)\right)$

3.C Appendix 3

MVAD Data: Gating Network Coefficients

Multinomial logistic regression coefficients and associated WLBS standard errors for the gating network of a $G = 10$ UCN model with stepwise selection of covariates are provided in Table 3.6. For completeness, coefficients and WLBS standard errors for an otherwise equivalent model with all covariates included (except those used to define the sampling weights) are given in Table 3.C.1. Such a model achieves a wDBS value of 0.4717 (see Table 3.3), compared to 0.4745 for the optimal model with only a subset of covariates detailed in Section 3.5.1. Notably, $G = 10$ and the UCN model type are both also optimal according to the wDBS criterion for the model with all covariates included.

Table 3.C.1: Multinomial logistic regression coefficients and associated WLBS standard errors (in parentheses) for the gating network of the 10-component UCN model with all covariates included.

Cluster	(Intercept)		Gender		Catholic		Funemp		GCSE5eq		FMPR		Livboth	
2	−1.29	(0.58)	−0.57	(0.54)	1.10	(0.43)	1.50	(0.59)	−0.06	(0.68)	0.36	(0.65)	−0.04	(0.53)
3	0.10	(0.49)	−0.55	(0.39)	0.21	(0.40)	0.50	(0.54)	1.25	(0.49)	0.50	(0.47)	−0.27	(0.38)
4	0.66	(0.50)	−0.19	(0.39)	−0.23	(0.39)	−0.09	(0.51)	−0.29	(0.51)	−0.86	(0.42)	−0.16	(0.39)
5	−1.16	(0.57)	1.24	(0.49)	0.39	(0.42)	−0.17	(0.59)	0.24	(0.61)	−0.26	(0.56)	−0.14	(0.44)
6	−2.52	(1.09)	−0.57	(0.61)	0.65	(0.70)	0.41	(1.10)	1.97	(0.77)	0.83	(0.70)	1.46	(0.64)
7	0.10	(0.63)	−0.76	(0.54)	−0.05	(0.53)	0.26	(0.72)	1.32	(0.59)	−0.50	(0.63)	0.03	(0.53)
8	−2.86	(0.63)	−0.60	(0.46)	−0.04	(0.47)	−0.24	(0.71)	3.24	(0.55)	0.31	(0.54)	1.17	(0.48)
9	−1.82	(0.63)	−0.40	(0.42)	0.58	(0.43)	−0.35	(0.70)	3.77	(0.53)	0.82	(0.48)	0.41	(0.46)
Noise	−1.76	(0.67)	0.40	(1.02)	−0.93	(0.97)	0.48	(0.85)	1.34	(0.99)	0.03	(0.98)	−0.65	(0.86)

3.D Appendix 4

Comparison of Distance Measures in Terms of Their Implicit Substitution Costs

The implicit substitution cost matrices for the CC, UC, CU, and UU MEDseq models are given in Table 3.D.1, Table 3.D.2, Table 3.D.3, and Table 3.D.4, respectively. Each table is represented by an array of subtables, with rows of subtables corresponding to components and columns of subtables corresponding to time periods. Within each symmetric subtable, rows and columns correspond to states $1, 2, \ldots, v$. The non-noise components of the corresponding CCN, UCN, CUN, and UUN models behave in a similar fashion, with all precision parameters equal to zero for the uniform noise component. While the precision parameters vary across components and/or time points, there is only a single cost associated with each substitution cost matrix in all cases. Recall, however, that the inclusion of sampling weights further scales the precision parameters for each observation. See Table 3.A.1 for the total numbers of precision parameters under each model type.

Table 3.D.1: Implicit substitution cost matrices for the CC MEDseq model and the non-noise components of the CCN model, such that $\lambda_{g,t} = \lambda$.

	$t = 1$	$t = 2$	$t = T$
$g = 1$	$\begin{array}{c\|cccc} & 1 & 2 & \ldots & v \\ \hline 1 & 0 & \lambda_{1,1} & \ldots & \lambda_{1,1} \\ 2 & \lambda_{1,1} & 0 & \ldots & \lambda_{1,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v & \lambda_{1,1} & \lambda_{1,1} & \ldots & 0 \end{array}$	$\begin{array}{c\|cccc} & 1 & 2 & \ldots & v \\ \hline 1 & 0 & \lambda_{1,1} & \ldots & \lambda_{1,1} \\ 2 & \lambda_{1,1} & 0 & \ldots & \lambda_{1,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v & \lambda_{1,1} & \lambda_{1,1} & \ldots & 0 \end{array}$	$\cdots$ $\begin{array}{c\|cccc} & 1 & 2 & \ldots & v \\ \hline 1 & 0 & \lambda_{1,1} & \ldots & \lambda_{1,1} \\ 2 & \lambda_{1,1} & 0 & \ldots & \lambda_{1,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v & \lambda_{1,1} & \lambda_{1,1} & \ldots & 0 \end{array}$
$g = 2$	$\begin{array}{c\|cccc} & 1 & 2 & \ldots & v \\ \hline 1 & 0 & \lambda_{1,1} & \ldots & \lambda_{1,1} \\ 2 & \lambda_{1,1} & 0 & \ldots & \lambda_{1,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v & \lambda_{1,1} & \lambda_{1,1} & \ldots & 0 \end{array}$	$\begin{array}{c\|cccc} & 1 & 2 & \ldots & v \\ \hline 1 & 0 & \lambda_{1,1} & \ldots & \lambda_{1,1} \\ 2 & \lambda_{1,1} & 0 & \ldots & \lambda_{1,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v & \lambda_{1,1} & \lambda_{1,1} & \ldots & 0 \end{array}$	$\cdots$ $\begin{array}{c\|cccc} & 1 & 2 & \ldots & v \\ \hline 1 & 0 & \lambda_{1,1} & \ldots & \lambda_{1,1} \\ 2 & \lambda_{1,1} & 0 & \ldots & \lambda_{1,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v & \lambda_{1,1} & \lambda_{1,1} & \ldots & 0 \end{array}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$ $\vdots$
$g = G$	$\begin{array}{c\|cccc} & 1 & 2 & \ldots & v \\ \hline 1 & 0 & \lambda_{1,1} & \ldots & \lambda_{1,1} \\ 2 & \lambda_{1,1} & 0 & \ldots & \lambda_{1,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v & \lambda_{1,1} & \lambda_{1,1} & \ldots & 0 \end{array}$	$\begin{array}{c\|cccc} & 1 & 2 & \ldots & v \\ \hline 1 & 0 & \lambda_{1,1} & \ldots & \lambda_{1,1} \\ 2 & \lambda_{1,1} & 0 & \ldots & \lambda_{1,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v & \lambda_{1,1} & \lambda_{1,1} & \ldots & 0 \end{array}$	$\cdots$ $\begin{array}{c\|cccc} & 1 & 2 & \ldots & v \\ \hline 1 & 0 & \lambda_{1,1} & \ldots & \lambda_{1,1} \\ 2 & \lambda_{1,1} & 0 & \ldots & \lambda_{1,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v & \lambda_{1,1} & \lambda_{1,1} & \ldots & 0 \end{array}$

Table 3.D.2: Implicit substitution cost matrices for the UC MEDseq model and the non-noise components of the UCN model, such that $\lambda_{g,t} = \lambda_g$.

$t = 1$

$g = 1$	1	2	...	v
1	0	$\lambda_{1,1}$	...	$\lambda_{1,1}$
2	$\lambda_{1,1}$	0	...	$\lambda_{1,1}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{1,1}$	$\lambda_{1,1}$	...	0

$t = 2$

$g = 1$	1	2	...	v
1	0	$\lambda_{1,1}$	...	$\lambda_{1,1}$
2	$\lambda_{1,1}$	0	...	$\lambda_{1,1}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{1,1}$	$\lambda_{1,1}$	...	0

$\cdots$

$t = T$

$g = 1$	1	2	...	v
1	0	$\lambda_{1,1}$	...	$\lambda_{1,1}$
2	$\lambda_{1,1}$	0	...	$\lambda_{1,1}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{1,1}$	$\lambda_{1,1}$	...	0

$g = 2$	1	2	...	v
1	0	$\lambda_{2,1}$	...	$\lambda_{2,1}$
2	$\lambda_{2,1}$	0	...	$\lambda_{2,1}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{2,1}$	$\lambda_{2,1}$	...	0

$g = 2$	1	2	...	v
1	0	$\lambda_{2,1}$	...	$\lambda_{2,1}$
2	$\lambda_{2,1}$	0	...	$\lambda_{2,1}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{2,1}$	$\lambda_{2,1}$	...	0

$\cdots$

$g = 2$	1	2	...	v
1	0	$\lambda_{2,1}$	...	$\lambda_{2,1}$
2	$\lambda_{2,1}$	0	...	$\lambda_{2,1}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{2,1}$	$\lambda_{2,1}$	...	0

⋮ ⋮ ⋱ ⋮

$g = G$	1	2	...	v
1	0	$\lambda_{G,1}$	...	$\lambda_{G,1}$
2	$\lambda_{G,1}$	0	...	$\lambda_{G,1}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{G,1}$	$\lambda_{G,1}$	...	0

$g = G$	1	2	...	v
1	0	$\lambda_{G,1}$	...	$\lambda_{G,1}$
2	$\lambda_{G,1}$	0	...	$\lambda_{G,1}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{G,1}$	$\lambda_{G,1}$	...	0

$\cdots$

$g = G$	1	2	...	v
1	0	$\lambda_{G,1}$	...	$\lambda_{G,1}$
2	$\lambda_{G,1}$	0	...	$\lambda_{G,1}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{G,1}$	$\lambda_{G,1}$	...	0

Table 3.D.3: Implicit substitution cost matrices for the CU MEDseq model and the non-noise components of the CUN model, such that $\lambda_{g,t} = \lambda_t$.

$t = 1$

$g = 1$	1	2	...	v
1	0	$\lambda_{1,1}$	...	$\lambda_{1,1}$
2	$\lambda_{1,1}$	0	...	$\lambda_{1,1}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{1,1}$	$\lambda_{1,1}$	...	0

$t = 2$

$g = 1$	1	2	...	v
1	0	$\lambda_{1,2}$	...	$\lambda_{1,2}$
2	$\lambda_{1,2}$	0	...	$\lambda_{1,2}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{1,2}$	$\lambda_{1,2}$	...	0

$\cdots$

$t = T$

$g = 1$	1	2	...	v
1	0	$\lambda_{1,T}$	...	$\lambda_{1,T}$
2	$\lambda_{1,T}$	0	...	$\lambda_{1,T}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{1,T}$	$\lambda_{1,T}$	...	0

$g = 2$	1	2	...	v
1	0	$\lambda_{1,1}$	...	$\lambda_{1,1}$
2	$\lambda_{1,1}$	0	...	$\lambda_{1,1}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{1,1}$	$\lambda_{1,1}$	...	0

$g = 2$	1	2	...	v
1	0	$\lambda_{1,2}$	...	$\lambda_{1,2}$
2	$\lambda_{1,2}$	0	...	$\lambda_{1,2}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{1,2}$	$\lambda_{1,2}$	...	0

$\cdots$

$g = 2$	1	2	...	v
1	0	$\lambda_{1,T}$	...	$\lambda_{1,T}$
2	$\lambda_{1,T}$	0	...	$\lambda_{1,T}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{1,T}$	$\lambda_{1,T}$	...	0

⋮ ⋮ ⋱ ⋮

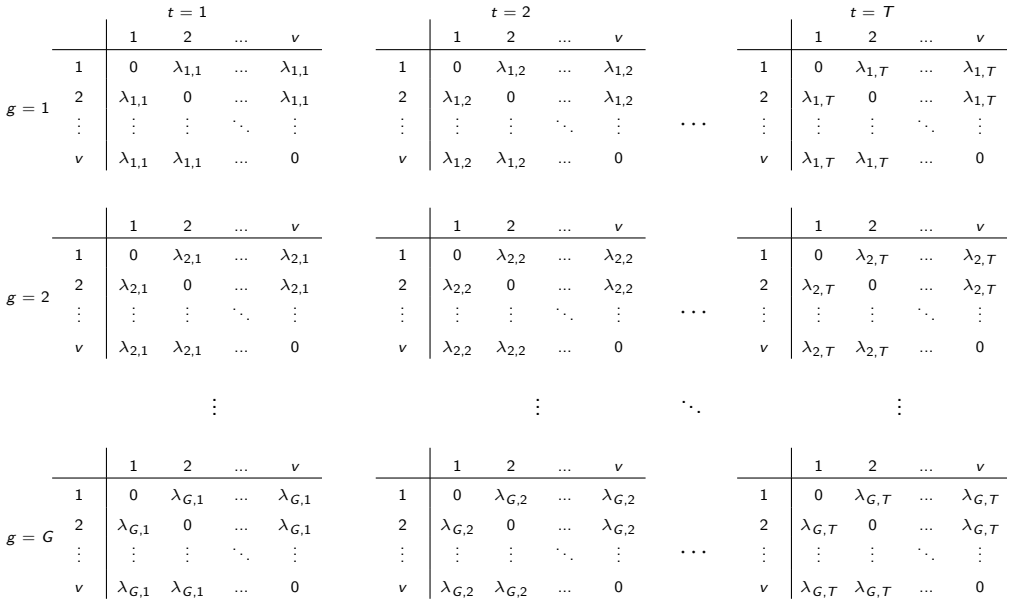
$g = G$	1	2	...	v
1	0	$\lambda_{1,1}$	...	$\lambda_{1,1}$
2	$\lambda_{1,1}$	0	...	$\lambda_{1,1}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{1,1}$	$\lambda_{1,1}$	...	0

$g = G$	1	2	...	v
1	0	$\lambda_{1,2}$	...	$\lambda_{1,2}$
2	$\lambda_{1,2}$	0	...	$\lambda_{1,2}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{1,2}$	$\lambda_{1,2}$	...	0

$\cdots$

$g = G$	1	2	...	v
1	0	$\lambda_{1,T}$	...	$\lambda_{1,T}$
2	$\lambda_{1,T}$	0	...	$\lambda_{1,T}$
⋮	⋮	⋮	⋱	⋮
v	$\lambda_{1,T}$	$\lambda_{1,T}$	...	0

Table 3.D.4: Implicit substitution cost matrices for the UU MEDseq model and the non-noise components of the UUN model, such that $\lambda_{g,t} = \lambda_{g,t}$.



The substitution costs for OM (Abbott and Forrest, 1986; Abbott and Hrycak, 1990) are given in Table 3.D.5. While OM also assigns costs to insertions and deletions, recall that such operations are not permitted when the sequence length T is common to all sequences. Notably, the substitution costs under OM are state-dependent. For the DHD (Lesnard, 2010) in Table 3.D.6, the state-dependent substitution costs also vary across time. We denote state-dependent substitution costs by $\lambda_{g,t}^{(a,b)}$, where the superscript (a, b) reflects the dependence on a particular pair of states a and b. Note that $\lambda_{g,t}^{(a,b)} = \lambda_{g,t}^{(b,a)}$, by symmetry. Recall that the assumption of homogeneous substitution costs between pairs of states leads to a closed-form expression for the normalising constant when the Hamming distance or weighted variants thereof are employed. For both OM and DHD, there are $v(v-1)/2$ parameters per substitution cost matrix, hence there are $Tv(v-1)/2$ parameters for the DHD. Neither OM nor the DHD permit the substitution costs to vary across clusters, in contrast to the UC, UCN, UU, and UUN MEDseq models. Of course, the use of OM or DHD in the MEDseq setting is purely hypothetical at present.

Table 3.D.5: State-dependent substitution cost matrices for OM distances, such that $\lambda_{g,t}^{(a,b)} = \lambda^{(a,b)}$.

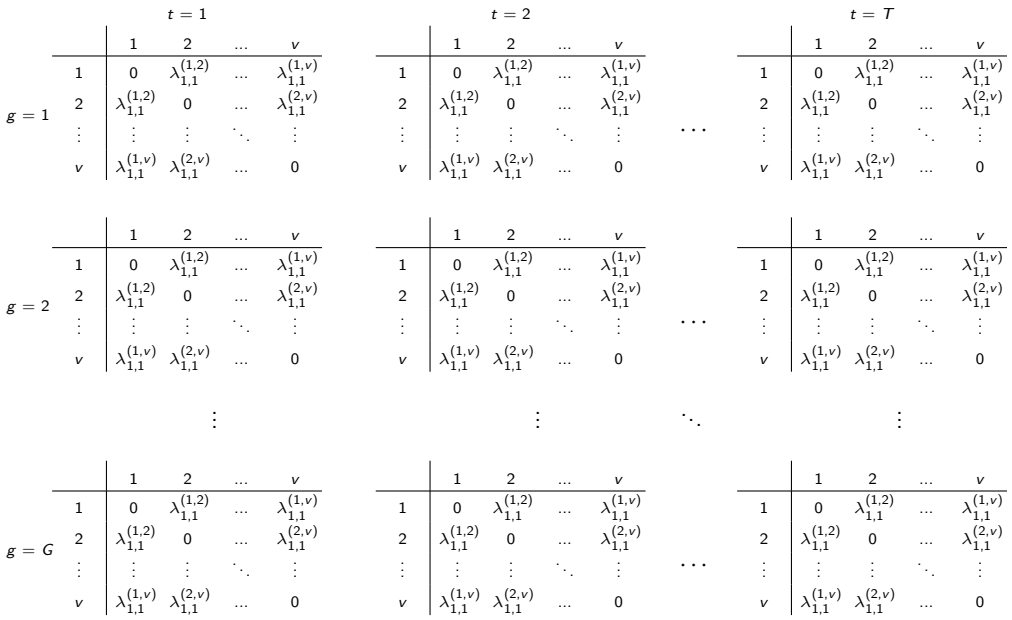
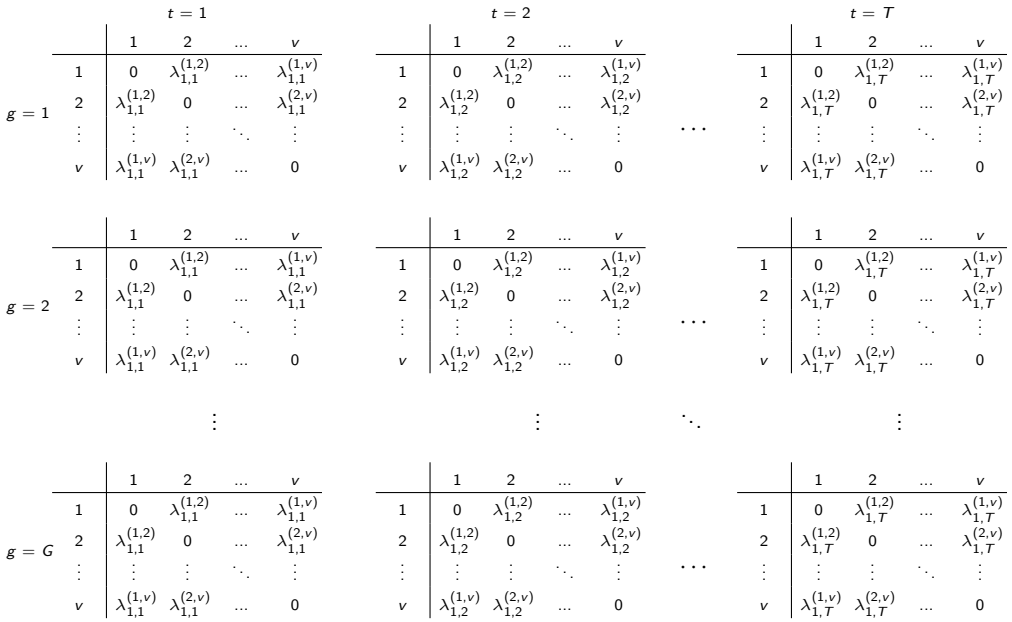


Table 3.D.6: State-dependent substitution cost matrices for the DHD, such that $\lambda_{g,t}^{(a,b)} = \lambda_{t}^{(a,b)}$.



The optimal MEDseq model for the MVAD data is a UCN model, i.e. one with substitution costs of the form $\lambda_{g,t} = \lambda_g$, according to Table 3.D.2. Given that the substitution costs under this particular weighted variant of the Hamming distance are further restricted to be common with respect to each pair of states, which makes model-fitting feasible, it is of interest to investigate how restrictive exactly this assumption may be. As the substitution costs do not vary across time, the closest comparator among the considered distance metrics with state-dependent substitution costs — leaving aside that the costs under a UCN model vary across clusters and an additional noise component is included — is OM, under which $\lambda_{g,t}^{(a,b)} = \lambda^{(a,b)}$ according to Table 3.D.5.

Hence, as but one illustrative example, we demonstrate a method of estimating state-dependent substitution costs from the data which is often used in OM analyses. The method relies on the estimated probabilities of transitioning between states, such that

$$\widehat{\lambda}^{(a,b)} = \mathcal{C} - \Pr\left(s_{i,(t+\ell)} = a \,|\, s_{i,t} = b\right) - \Pr\left(s_{i,(t+\ell)} = b \,|\, s_{i,t} = a\right),$$

where $\mathcal{C}$ is a constant and ℓ is the time lag. Table 3.D.7 gives the costs, estimated thusly with $\mathcal{C} = 2$ and $\ell = 1$, for the MVAD data with the first time point again removed. Encouragingly, the costs do not appear to differ greatly across pairs of states, which suggests that the assumption of state-independent costs under the UCN MEDseq model may in fact not be too detrimental for this particular application. Similar conclusions are drawn for different values of $\mathcal{C}$ and ℓ.

Table 3.D.7: State-dependent substitution costs for the MVAD data, estimated using the observed rates of transition between states.

	EM	FE	HE	JL	SC	TR
EM	0	1.967	1.987	1.951	1.984	1.959
FE	1.967	0	1.993	1.962	1.987	1.992
HE	1.987	1.993	0	1.996	1.982	1.999
JL	1.951	1.962	1.996	0	1.985	1.972
SC	1.984	1.987	1.982	1.985	0	1.995
TR	1.959	1.992	1.999	1.972	1.995	0

3.E Appendix 5

MEDseq R Package Vignette

This appendix presents a reproduction of the package vignette[5] of the associated R package `MEDseq` for implementation of the proposed method.

Notably, this vignette also presents summarised results of a second application of the MEDseq model family, to data (with $n = 2000$ cases, $v = 8$ categories, and $T = 16$ equally spaced time periods) on yearly family life states from the retrospective biographical survey conducted by the Swiss Household Panel in 2002 (Müller et al., 2007). This data set is available in the `MEDseq` package under the name `biofam`. The fitted $G = 10$ UUN model was identified as optimal according to the DBS rather than the wDBS criterion, as no sampling weights were used in the analysis, by otherwise following the same steps as in Section 3.5.1 for the MVAD data. However, the NGN rather than the GN setting was selected in this analysis.

[5] cran.r-project.org/web/packages/MEDseq/vignettes/MEDseq.html

MEDseq: Mixtures of Exponential-Distance Models with Covariates

Keefe Murphy

Introduction

MEDseq is an R package which fits a range of models introduced by Murphy et al. (2019). These are finite mixtures of exponential-distance models for clustering longitudinal/categorical life-course sequence data. A family of parsimonious precision parameter constraints are accommodated. So too are sampling weights. Gating covariates can be supplied via formula interfaces. Visualisation of the results of such models is also facilitated.

The most important function in the **MEDseq** package is: `MEDseq_fit`, for fitting the models via the EM or CEM algorithms. `MEDseq_control` allows supplying additional arguments which govern, among other things, controls on the initialisation of the allocations for the EM/CEM algorithm and the various model selection options. `MEDseq_compare` is provided for conducting model selection between different results from using different covariate combinations &/or initialisation strategies, etc. `MEDseq_stderr` is provided for computing the standard errors of the coefficients for the covariates in the gating network.

A dedicated plotting function exists for visualising various aspects of the results, using new methods as well as some existing methods from the **TraMineR** package. Finally, the package also contains two data sets: `biofam` and `mvad`.

If you find bugs or want to suggest new features please visit the **MEDseq** GitHub issues page.

This vignette mostly aims to demonstrate the **MEDseq** models by reproducing the analysis of the MVAD data in the Murphy et al. (2019) paper. However, an additional example data set is also analysed.

Installing MEDseq

MEDseq will run in Windows, Mac OS X or Linux. To install it you first need to install R. Installing Rstudio as a nice desktop environment for using R is also recommended.

Once in R you can type at the R command prompt:

```
install.packages('devtools')
devtools::install_github('Keefe-Murphy/MEDseq')
```

to install the latest development version of the package from the **MEDseq** GitHub page.

To instead install the latest stable official release of the package from CRAN go to R and type:

```
install.packages('MEDseq')
```

In either case, if you then type:

```
library(MEDseq)
```

it will load in all the **MEDseq** functions.

The GitHub version contains a few more features but some of these may not yet be fully tested, and occasionally this version might be liable to break when it is in the process of being updated.

MVAD Data

Load the MVAD data. The data comes from a study by McVicar and Anyadike-Danes (2002) on transition from school to work. The data consist of static background characteristics (covariates and sampling weights) and a time series sequence of 72 monthly labour market activities for each of 712 individuals in a cohort survey. The individuals were followed up from July 1993 to June 1999. Type `?mvad` for more information. We will drop the first sequence position (i.e. time point) as it (along with the covariates `Grammar` and `Location`) were used to define the sampling weights.

Note that the data set must have equal sequence lengths, and the intervals are assumed to be evenly spaced. The MVAD data meets these criteria. The **TraMineR** function `seqdef` is used to convert the data to the appropriate format.

```
data(mvad, package="MEDseq")
mvad$Location <- factor(apply(mvad[,5L:9L], 1L, function(x) which(x == "yes")),
                        labels = colnames(mvad[,5L:9L]))
mvad          <- list(covariates = mvad[c(3L:4L,10L:14L,87L)],
                      sequences = mvad[,15L:86L],
                      weights = mvad[,2L])
```

```
mvad.cov      <- mvad$covariates
mvad.seq      <- seqdef(mvad$sequences[,-1L],
                        states = c("EM", "FE", "HE",
                                   "JL", "SC", "TR"),
                        labels = c("Employment", "Further Education",
                                   "Higher Education", "Joblessness",
                                   "School", "Training"))
```

The function `MEDseq_control` allows the algorithm used for model fitting, the method used to initialise the allocations (e.g. k-medoids or Ward's hierarchical clustering), the criterion used to identify the optimal model (e.g. density-based silhouette (DBS), average silhouette width (ASW), BIC, etc.), and more to be specified. By default, the EM algorithm is employed, k-medoids is used to obtain starting values, and the (weighted) mean density-based silhouette criterion is used to choose the optimal model (if a range of models are fitted). In this vignette, we will mostly accept these defaults.

The optimal model identified in the Murphy et al. (2019) paper has `G=10` components, the `UCN` model type, sampling weights, and a subset of covariates identified using a stepwise variable selection procedure. The UCN model has a single precision parameter for each cluster. The `gating` covariates can be specified via a formula interface. The argument `covars` tells the formula where to look for the specified covariates. Thus, to fit such a model:

```
mod1 <- MEDseq_fit(mvad.seq, G=10,
                   modtype="UCN", weights=mvad$weights,
                   gating=~ fmpr + gcse5eq + livboth, covars=mvad.cov)
```

The names of the model types are CC, UC, CU, UU, CCN, UCN, CUN, and UUN. The first letter denotes whether the precision parameters are constrained (C) or unconstrained (U) across clusters, the second denotes the same across time periods, and the third letter (N) indicates the precision of a noise component. In this context, a noise component is one wherein the distribution of the sequences is uniform.

Typically, a range of `G` values and `modtype` settings are supplied to a single call to `MEDseq_fit` and chosen from using some criterion. Thus, for a given set of covariates, the model which is optimum in terms of the number of components and the precision parameter configuration can be identified.

To compare different runs using different sets of covariates, the function `MEDseq_compare` can be used. First, let's fit models with all covariates included (except `Grammar` and `Location`) and no covariates included. Let's try different numbers of components and different model types. Note that only the first model here includes a noise component.

```
# 9-component CUN model with no coviarates.
# CUN models have a precision parameter for each sequence position (i.e. time
point),
# though each time point's precision is common across clusters.

mod2 <- MEDseq_fit(mvad.seq, G=9,  modtype="CUN", weights=mvad$weights)

# 11-component CC model with all coviarates.
# CC models have a single precision parameter across all clusters & time points.

mod3 <- MEDseq_fit(mvad.seq, G=11, modtype="CC", weights=mvad$weights,
                   gating=~. - Grammar - Location, covars=mvad.cov)
```

Confirm that the first model is indeed optimal according to the (weighted) mean density-based silhouette. Examine the optimal model in greater detail. Observe how the UCN model type explicitly includes a noise component.

```
(comp <- MEDseq_compare(mod1, mod2, mod3, criterion="dbs"))
opt   <- comp$optimal
```

```
## ---------------------------------------------------------------------
## Comparison of Mixtures of Exponential-Distance Models with Covariates
## Data: mvad.seq
## Ranking Criterion: DBS
## Optimal Only: FALSE
## ---------------------------------------------------------------------
##
##  rank MEDNames modelNames  G  df iters        bic        icl        aic   dbs
##     1     mod1        UCN 10 684    10 -86876.197 -86887.052 -83752.045 0.474
##     2     mod2        CUN  9 647    49 -89379.589 -89388.574 -86424.433  0.42
##     3     mod3         CC 11 852     5 -84801.298 -84816.481  -80909.81 0.238
##    asw     loglik                                                 gating algo
##  0.359 -41192.022                            ~fmpr + gcse5eq + livboth   EM
##   0.37 -42565.217                                                 None   EM
##  0.329 -39602.905 ~male + catholic + funemp + gcse5eq + fmpr + livboth   EM
##  weights noise noise.gate equalPro
##     TRUE  TRUE       TRUE
##     TRUE  TRUE              FALSE
##     TRUE FALSE
```

```
summary(opt)
```

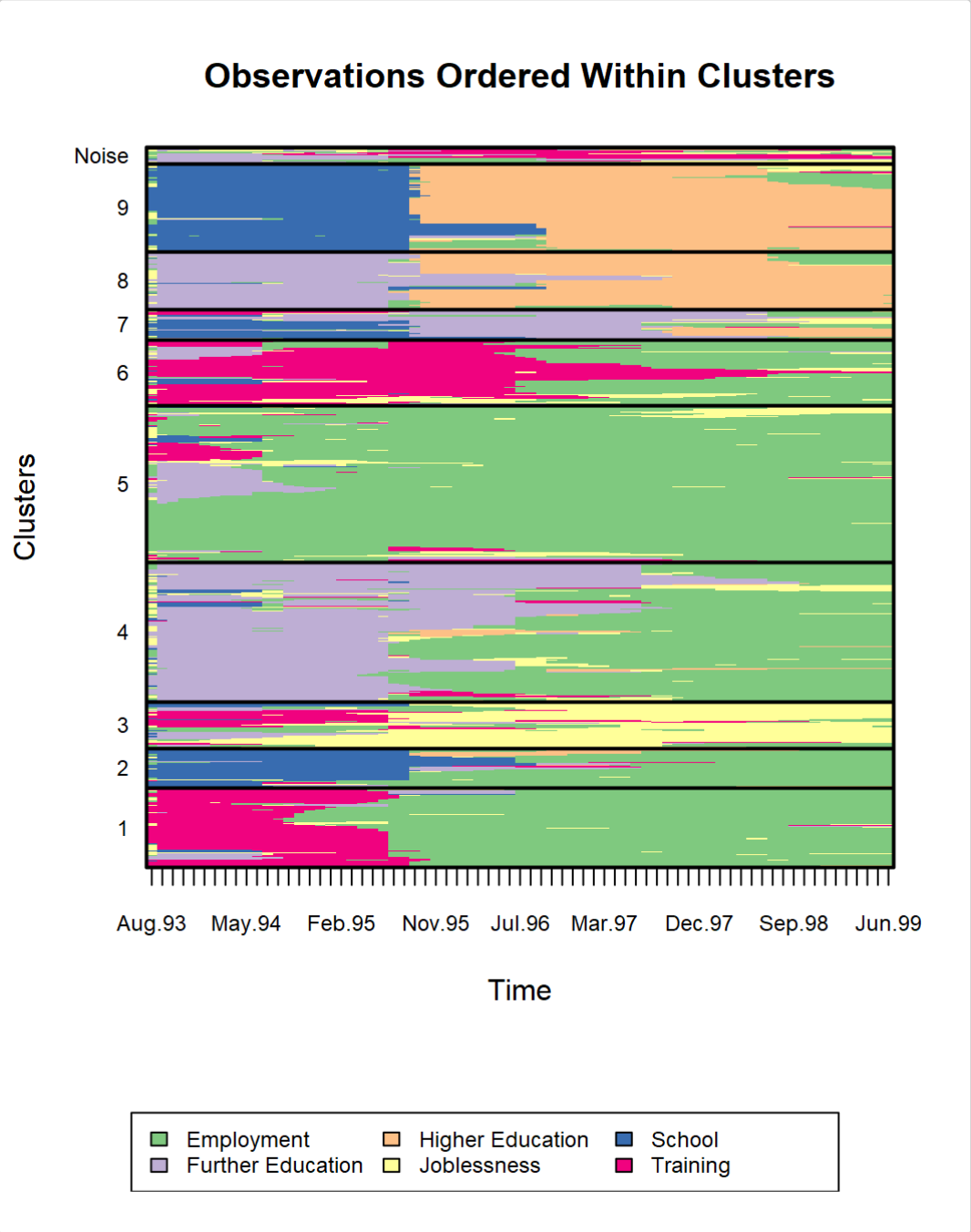
```
## ------------------------------------------------------
## Mixture of Exponential-Distance Models with Covariates
## Data: mvad.seq
## ------------------------------------------------------
##
## MEDseq (UCN), with 10 components
## Gating Network Covariates:  ~fmpr + gcse5eq + livboth
## Noise Component:            TRUE
## Noise Component Gating:     TRUE
##
##  log.likelihood   N  P V  df iters  DBS  ASW      BIC Algo
##       -41192.02 712 71 6 684    10 0.47 0.36 -86876.2   EM
##
## Clustering table:
##   0   1   2   3   4   5   6   7   8   9
##  16  79  39  46 138 155  65  30  57  87
```

Examine the estimated gating network coefficients. Note that standard errors can be computed by calling `MEDseq_stderr` on the `opt` object, to better inform the interpretations of the covariate effects.

```
coef(opt$gating)
##          (Intercept)    fmpryes gcse5eqyes  livbothyes
## Cluster2 -0.18964080 -0.6574700  1.3740424 -0.02973523
## Cluster3 -0.46305507 -0.5411090 -0.2235576  0.08432396
## Cluster4  0.04433461  0.2864898  1.2954436 -0.30215444
## Cluster5  0.48175311 -0.8935825 -0.2534328 -0.21066807
## Cluster6 -0.15643676 -0.2713519  0.1693219 -0.07445400
## Cluster7 -2.37938466  0.6215334  2.0301946  1.43157911
## Cluster8 -3.21496893  0.2758460  3.3367901  1.12349106
## Cluster9 -1.76413515  0.7124812  3.8529731  0.34858309
## Noise    -1.95606029  0.3711763  1.7003713 -1.07290325
```

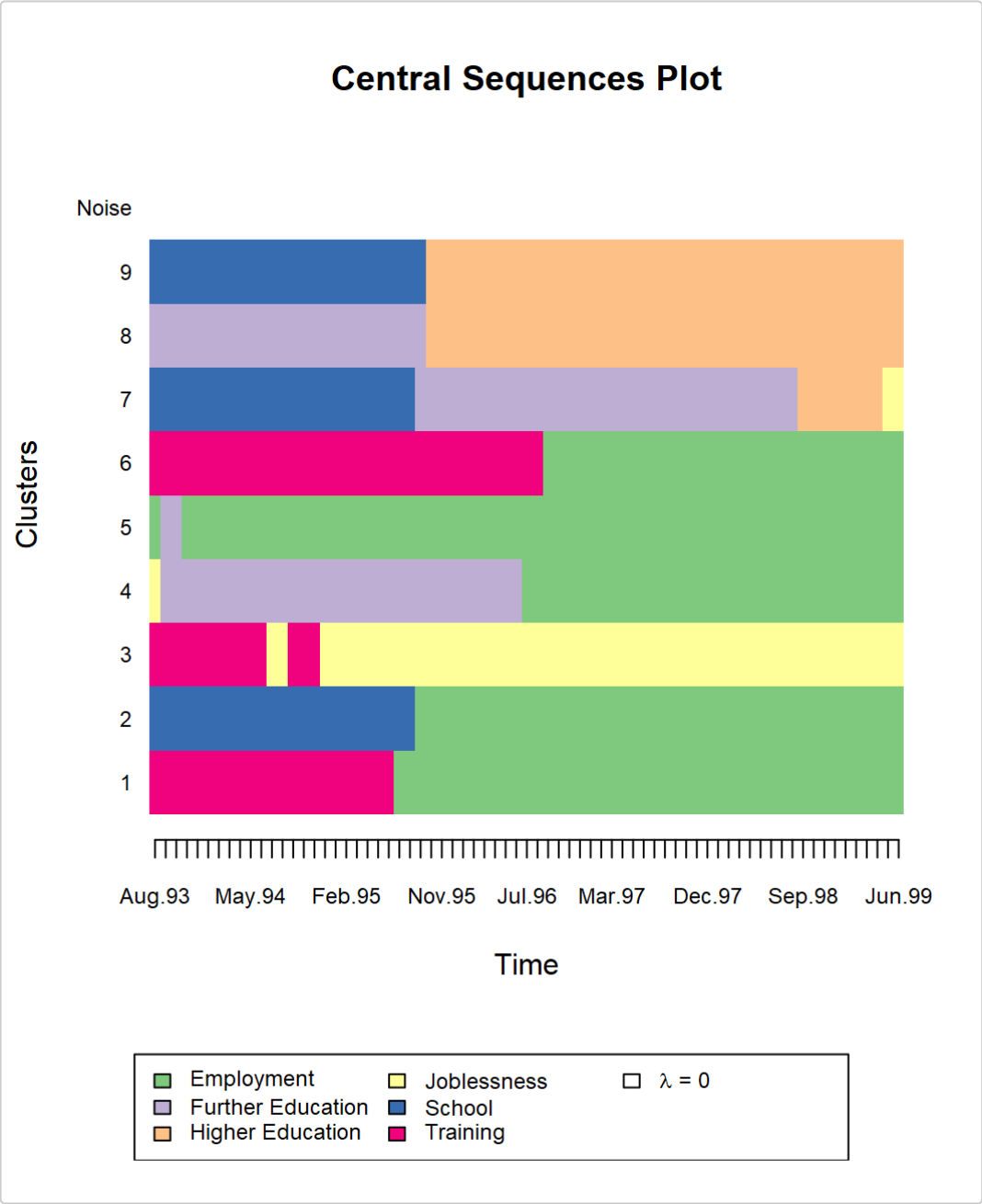
Visualise the clusters uncovered by the optimal model. Note that seriation is applied for visual clarity. The legend indicates which colours correspond to which state categories.

```
plot(opt, type="clusters")
```



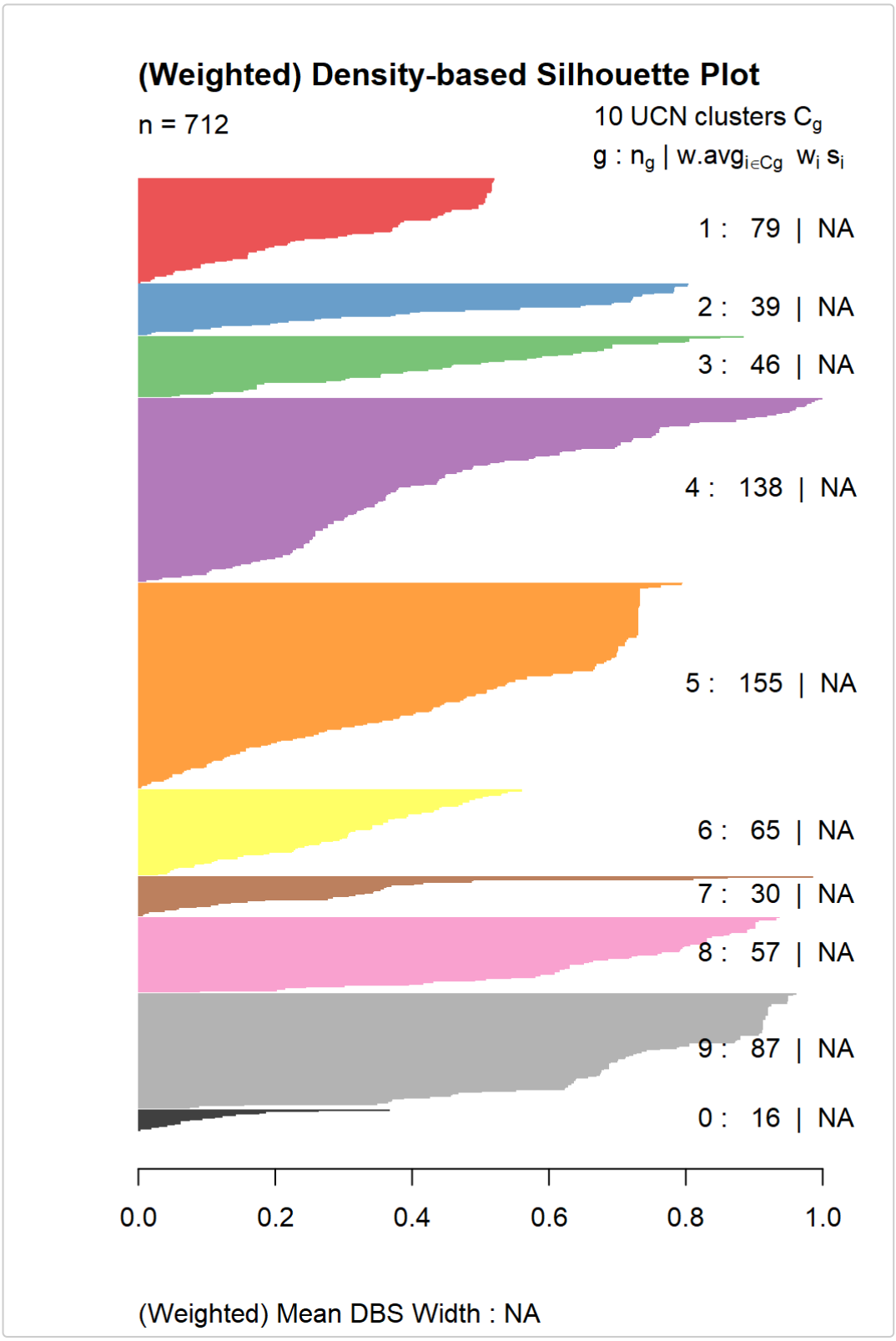
Many other plotting options exist, some of which are adapted from the **TraMineR** package. Use the following code to examine the central sequence parameters. Note that the central sequence of the noise component is not shown as it doesn't contribute to the likelihood.

```
plot(opt, type="mean")
```



Use the following code to see the observation-specific (weighted) density-based silhouette values (coloured by cluster). The (weighted) mean within each cluster is also shown. Note that the low average for the noise component is as expected; we do not expect this cluster to be internally coherent, rather it acts as a filter that allows the other clusters to be captured more clearly.

```
plot(opt, type="dbsvals")
```


(Weighted) Density-based Silhouette Plot
n = 712
10 UCN clusters C_g
g : n_g | w.avg$_{i \in Cg}$ w_i s_i
1 : 79 | NA
2 : 39 | NA
3 : 46 | NA
4 : 138 | NA
5 : 155 | NA
6 : 65 | NA
7 : 30 | NA
8 : 57 | NA
9 : 87 | NA
0 : 16 | NA
0.0
0.2
0.4
0.6
0.8
1.0
(Weighted) Mean DBS Width : NA

Finally, we can quantify the type of observation characterising each cluster by computing the mean time spent in each state within each cluster. By specifying `MAP=TRUE` here, we are computing the mean time based on the hard MAP partition rather than the soft probabilistic assignments. By specifying `norm=TRUE` (which is the default), the mean times are normalised to sum to the sequence length within each cluster. The size of each cluster in terms of the number of observations assigned to it is also reported.

```
MEDseq_meantime(opt,
                MAP=TRUE,
                norm=TRUE)
##          Size        EM         FE        HE         JL         SC         TR
## Cluster1   79 47.721519  1.7974684  0.000000  2.3037975  0.5569620 18.6202532
## Cluster2   39 37.435897  2.7179487  2.769231  2.6923077 24.0000000  1.3846154
## Cluster3   46  9.282609  4.0869565  0.000000 44.5652174  2.8260870 10.2391304
## Cluster4  138 33.644928 30.9782609  1.166667  3.3188406  0.7318841  1.1594203
## Cluster5  155 61.716129  2.9870968  0.000000  3.6387097  0.4967742  2.1612903
## Cluster6   65 28.230769  2.8153846  0.000000  5.1076923  0.8923077 33.9538462
## Cluster7   30  6.666667 33.3000000  7.366667  4.1666667 16.1333333  3.3666667
## Cluster8   57  4.456140 27.1929825 37.789474  0.7719298  0.7894737  0.0000000
## Cluster9   87  4.390805  0.5057471 38.034483  1.3563218 26.4137931  0.2988506
## Noise      16 14.187500 17.7500000  1.687500 14.7500000  2.3125000 20.3125000
```

Biofam Data

As a second example, let's consider data on $N = 2000$ 16 year-long family life state sequences built from the retrospective biographical survey carried out by the Swiss Household Panel (SHP) in 2002. Each of the $v = 8$ states are defined from a combination of five basic states; namely, living with parents (Parent), left home (Left), married (Marr), having children (Child), and Divorced. The data is available in the **MEDseq** package as `biofam`. Type `?biofam` for more information.

```
data(biofam, package="MEDseq")
biofam     <- list(covariates = biofam[2L:9L],
                   sequences = biofam[10L:25L] + 1L)
biofam.cov <- biofam$covariates[,colSums(is.na(biofam$covariates)) == 0]
biofam.seq <- seqdef(biofam$sequences,
                     states = c("P", "L", "M", "L+M",
                                "C", "L+C", "L+M+C", "D"),
                     labels = c("Parent", "Left", "Married",
                                "Left+Marr", "Child", "Left+Child",
                                "Left+Marr+Child", "Divorced"))
```

While the data set contains weights, they are not appropriate for use; `biofam` is merely a subsample of the original data so the weights are not properly adapted. Thus, we will fit a model without supplying the `weights` argument. Secondly, the data set also contain some baseline covariates. As many of them contain missing values, let's only consider the `birthwt` variable, which gives the birth year of each subject.

In the previous example, the model by default assumed that covariates also influenced the mixing proportion of the noise component. We can override this by specifying `noise.gate=FALSE` via `MEDseq_control`. Thus, the noise component's mixing proportion will be constant (though estimated) across all observed sequences and covariate patterns.

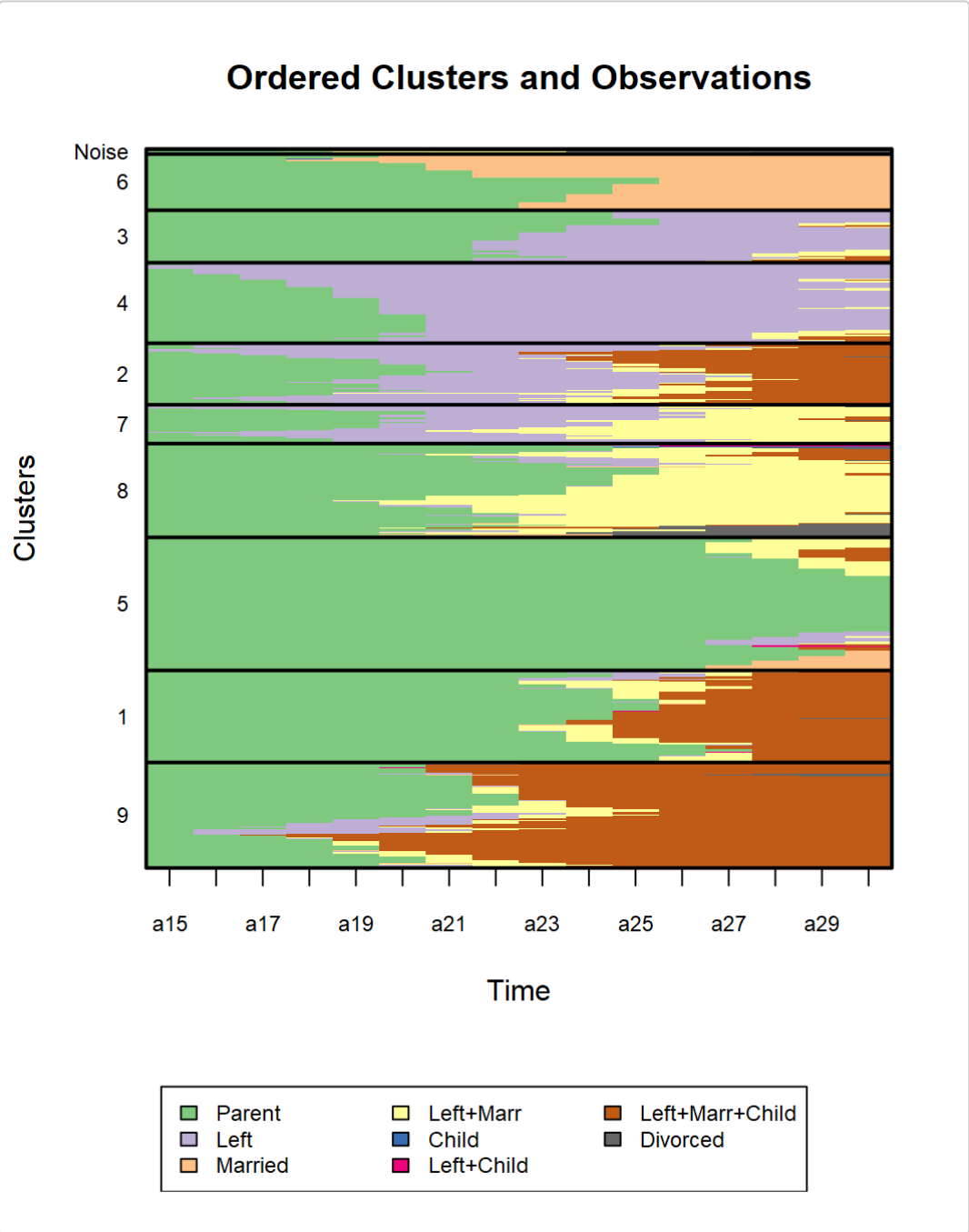
```
# The UUN model includes a noise component.
# Otherwise, there are precision parameters for each time point in each cluster.

bio <- MEDseq_fit(biofam.seq, G=10, modtype="UUN",
                  gating=~ birthyr, covars=biofam.cov,
                  control=MEDseq_control(noise.gate=FALSE))
```

Such a model was identified as optimal according to the weighted DBS criterion following the same steps as the analysis of the MVAD data in the Murphy et al. (2019) paper. Print the details of the model to the screen by typing `print(bio)`:

```
## Call:    MEDseq_fit(seqs = biofam.seq, G = 10, modtype = "UUN", gating =
~birthyr,
##     covars = biofam.cov, control = MEDseq_control(noise.gate = FALSE))
##
## Best Model: UUN, with 10 components and no weights (incl. gating network
covariates)
## DBS = 0.5 | ASW = 0.39 | BIC = -48455.12 | ICL = -48584.23 | AIC = -46741.24
## Gating: ~birthyr
```

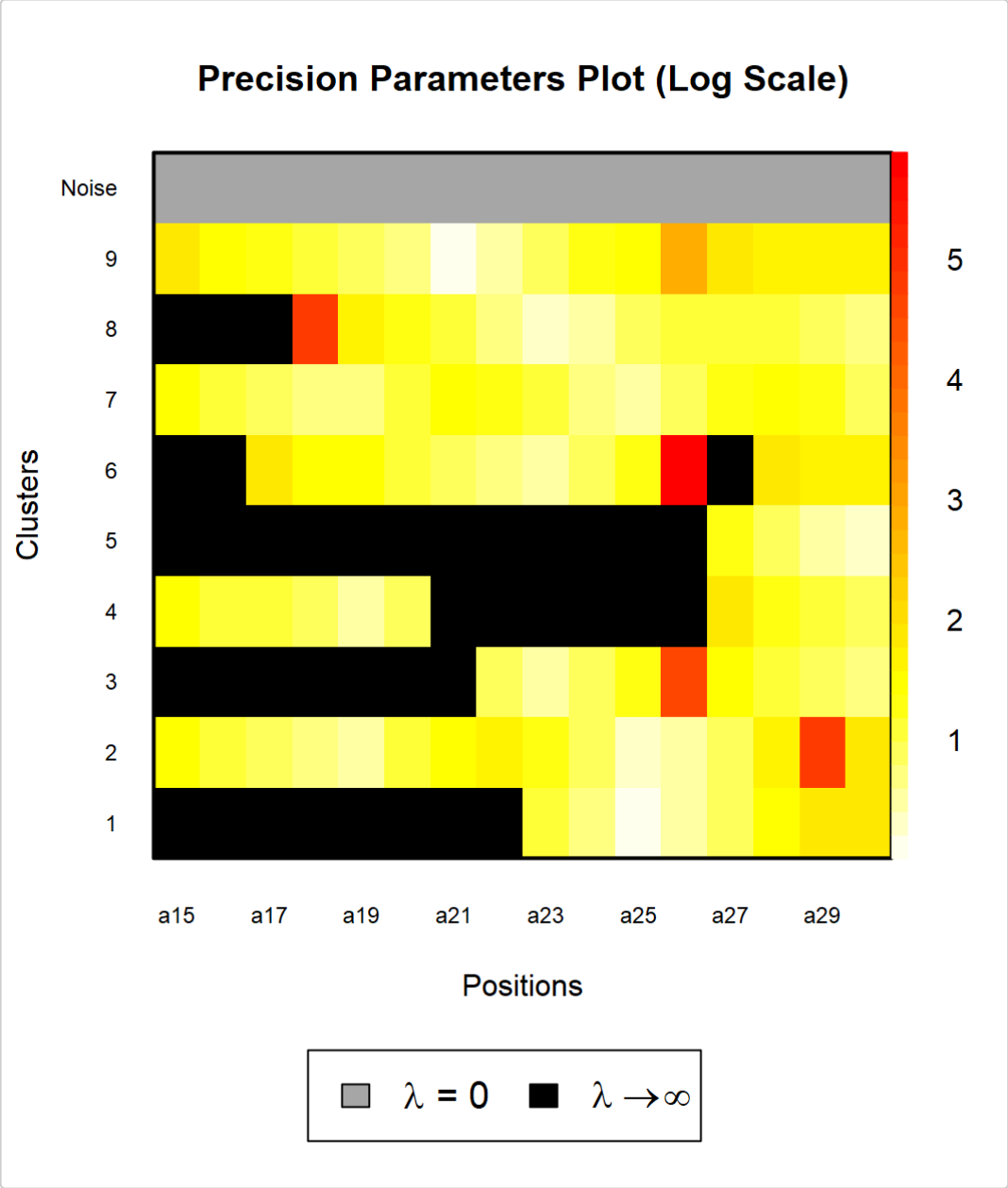
As before, let's look at the clusters uncovered by the model. This time, `seriate="both"` means to order the clusters and the observations within clusters for visual clarity.

```
plot(bio, type="clusters", seriate="both")
```



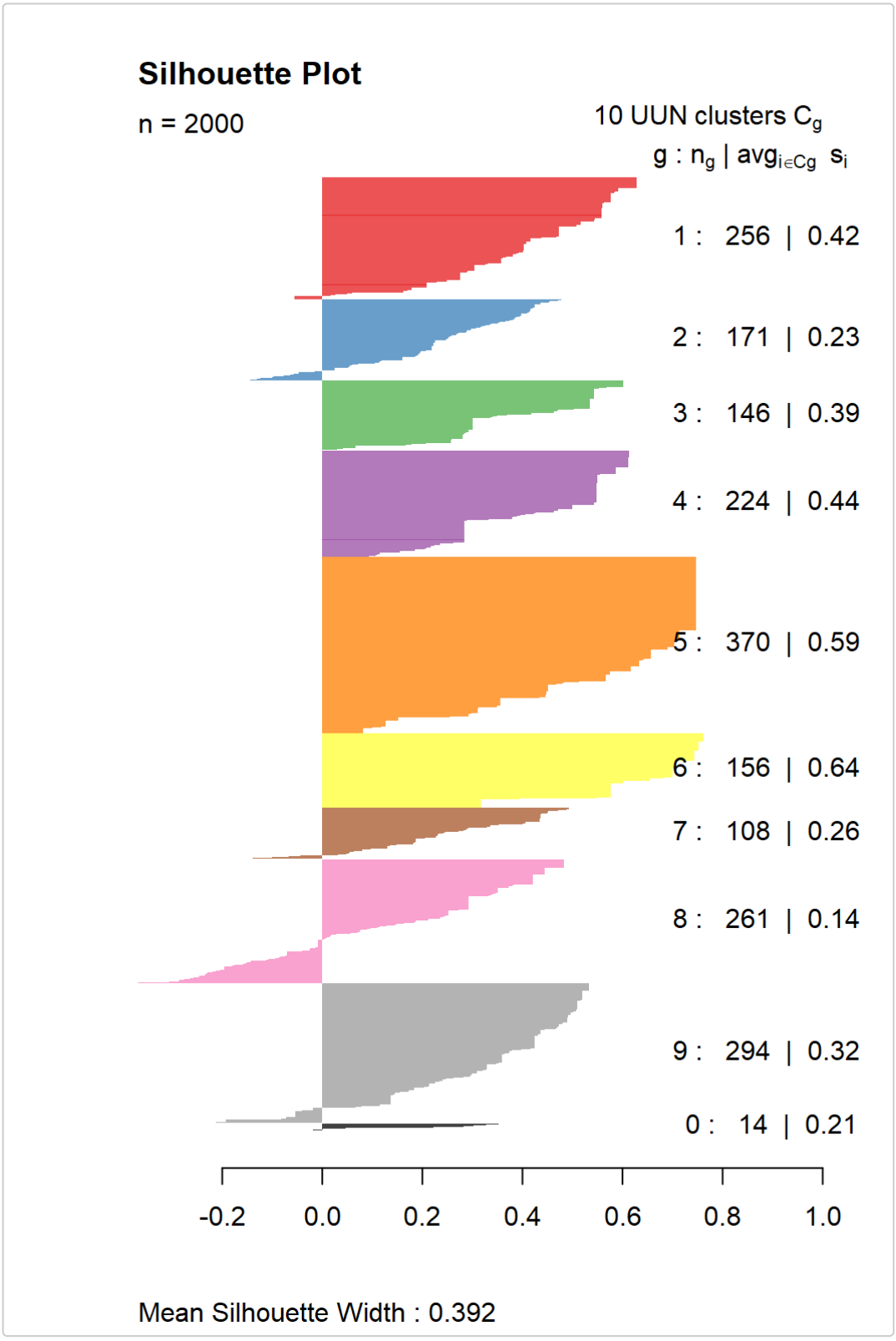
Use the following code to examine the precision parameters. Note how we have a full $16 \times G$ matrix of precision parameters, one for each time point in each cluster. Typically, we would not supply the argument `log.scale=TRUE`. However, in this case there are many quite large precision parameter values which skew the colour scale. Indeed, some are even infinite! Infinite precision under UU or UUN models implies that all values for that time point are identical within the given cluster.

```
plot(bio, type="precision", log.scale=TRUE)
```



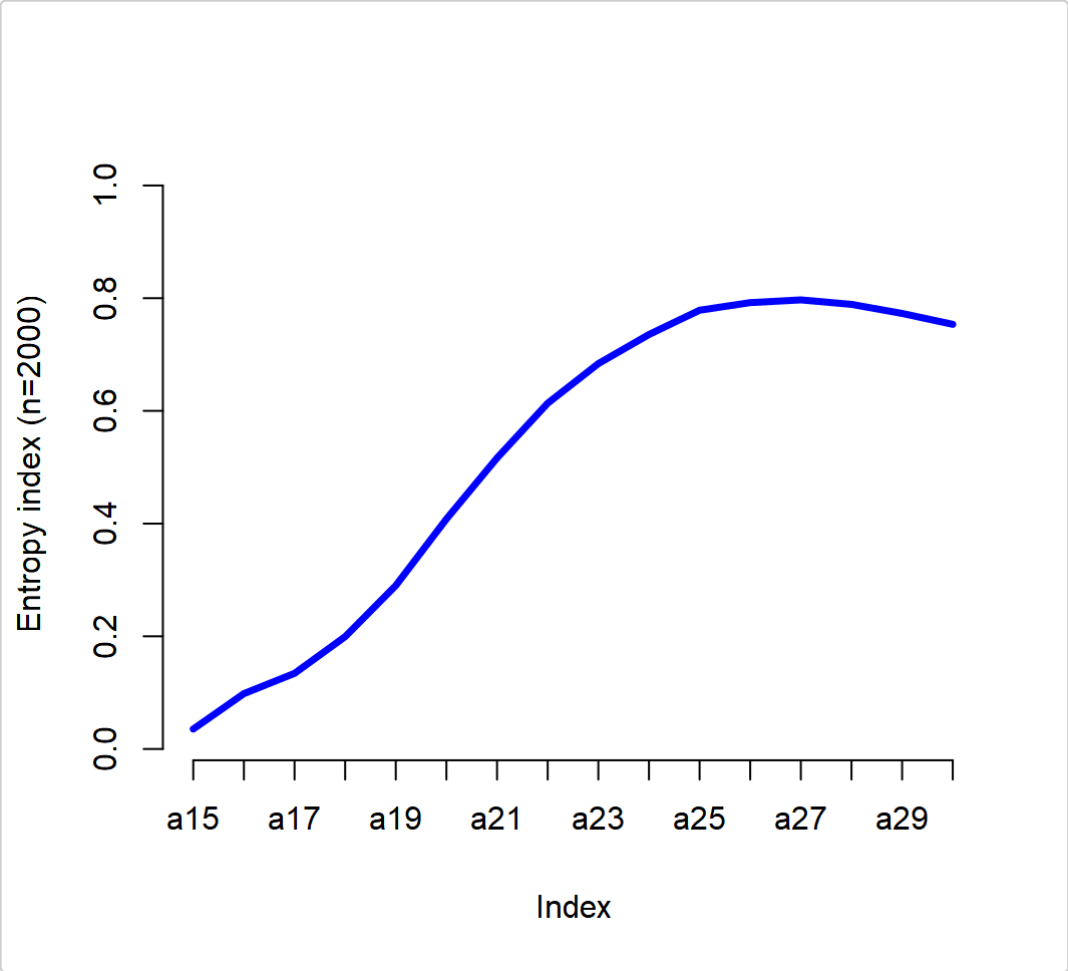
This time, rather than showing the weighted mean DBS values, let's look at observation-specific (unweighted) average silhouette-width values. Note that this relies instead on the MAP partition rather than the soft cluster assignment probabilities.

```
plot(bio, type="aswvals")
```


Silhouette Plot
n = 2000
10 UUN clusters C_g
g : n_g | $avg_{i \in Cg}$ s_i
1 : 256 | 0.42
2 : 171 | 0.23
3 : 146 | 0.39
4 : 224 | 0.44
5 : 370 | 0.59
6 : 156 | 0.64
7 : 108 | 0.26
8 : 261 | 0.14
9 : 294 | 0.32
0 : 14 | 0.21
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
Mean Silhouette Width : 0.392

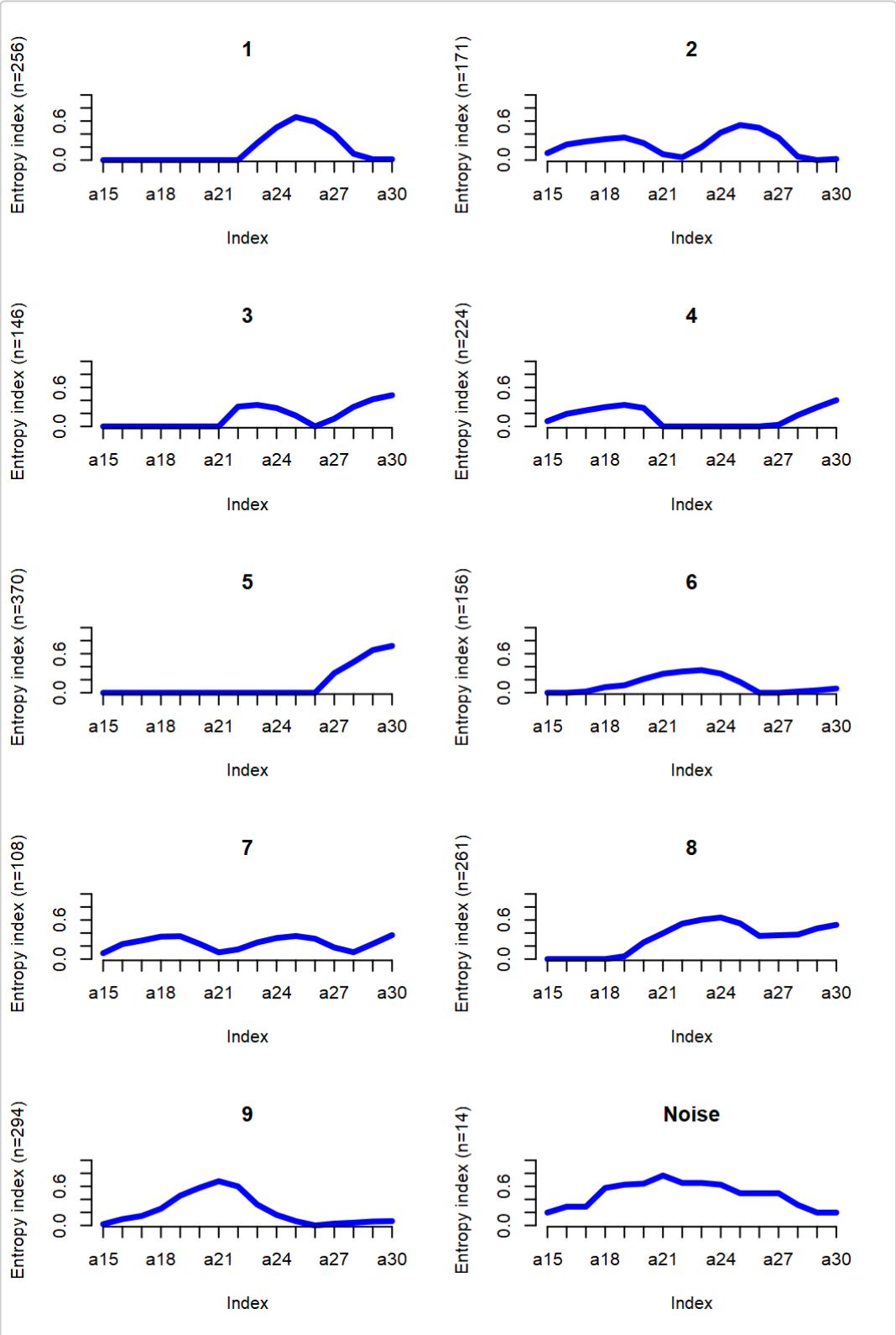
As stated above, some plot types are adapted from **TraMineR**. Let's first look at a plot of the transversal Shannon entropies for the whole data set.

```
seqHtplot(biofam.seq)
```



Now let's use the plot function from **MEDseq** to examine the transversal entropies within each cluster defined by the MAP partition. Here we can see for instance that subjects assigned to Cluster 9, corresponding to those individuals who left the parental home to marry relatively early and had a child on average just one year later, do indeed exhibit greater variability. Conversely, a postponement of the transition to adulthood is evident for subjects in Cluster 5.

```
plot(bio, type="Ht")
```


1
Entropy index (n=256)
2
Entropy index (n=171)
3
Entropy index (n=146)
4
Entropy index (n=224)
5
Entropy index (n=370)
6
Entropy index (n=156)
7
Entropy index (n=108)
8
Entropy index (n=261)
9
Entropy index (n=294)
Noise
Entropy index (n=14)
0.0 0.6
a15 a18 a21 a24 a27 a30
Index

Other plot types adapted from **TraMineR** can be produced using `type="d"` (state distribution plots), `type="f"` (state frequency plots), `type="i"` (selected sequence index plots), and `type="I"` (whole set index plots). Each of these plots are shown on a per-cluster basis. Clustering uncertainties, the gating network, and model selection criteria can also be visualised.

References

Murphy, K., T. B. Murphy, R. Piccarreta, and I. C. Gormley (2019). Clustering longitudinal life-course sequences using mixtures of exponential-distance models. *arXiv* pre-print, 1908.07963.

McVicar, D. and M. Anyadike-Danes (2002). Predicting successful and unsuccessful transitions from school to work by using sequence methods. *Journal of the Royal Statistical Society: Series A (Statistics in Society) 165* (2): 317-334.

Müller, N. S., M. Studer, and G. Ritschard (2007). Classification de parcours de vie à l'aide de l'optimal matching. In *XIVe Rencontre de la Société francophone de classification (SFC 2007), Paris, 5-7 septembre 2007*, pp. 157-160.

Chapter 4

Infinite Mixtures of Infinite Factor Analysers

Abstract

Factor-analytic Gaussian mixtures are often employed as a model-based approach to clustering high-dimensional data. Typically, the numbers of clusters and latent factors must be fixed in advance of model fitting. The pair which optimises some model selection criterion is then chosen. For computational reasons, having the number of factors differ across clusters is rarely considered.

Here the infinite mixture of infinite factor analysers (IMIFA) model is introduced. IMIFA employs a Pitman-Yor process prior to facilitate automatic inference of the number of clusters using the stick-breaking construction and a slice sampler. Automatic inference of the cluster-specific numbers of factors is achieved using multiplicative gamma process shrinkage priors and an adaptive Gibbs sampler. IMIFA is presented as the flagship of a family of factor-analytic mixtures.

Applications to benchmark data, metabolomic spectral data, and a handwritten digit example illustrate the IMIFA model's advantageous features. These include obviating the need for model selection criteria, reducing the computational burden associated with the search of the model space, improving clustering performance by allowing cluster-specific numbers of factors, and uncertainty quantification.

Keywords: Model-based clustering, factor analysis, Pitman-Yor process, multiplicative gamma process, adaptive Markov chain Monte Carlo.

4.1 Introduction

In cases where the number of variables p is comparable to or greater than the number of observations N, many clustering techniques tend to perform poorly or be intractable. Factor analysis (FA; Knott and Bartholomew, 1999) is a well-known approach to parsimoniously modelling data. Bai and Li (2012) outline some computational difficulties which arise when $N \ll p$. Model-based clustering methods which rely on latent factor models have long been successfully utilised to cluster high-dimensional data. Ghahramani and Hinton (1996) propose a mixture of factor analysers model (MFA) with cluster-specific parsimonious covariance matrices and estimate it via an expectation-maximisation algorithm; McLachlan and Peel (2000) provide a succinct overview. Estimation of MFA models has also been considered in a Bayesian framework (Diebolt and Robert, 1994; Richardson and Green, 1997). McNicholas and Murphy (2008) develop a suite of similar parsimonious Gaussian mixture models. Other related developments in this area include Baek et al. (2010) and Viroli (2010), among others.

Clustering using a MFA model typically requires specifying the number of clusters and factors in advance of model fitting. Generally, a range of MFA models with different numbers of clusters and factors are fitted and then compared through the use of information criteria, such as the Bayesian Information Criterion (BIC; Kass and Raftery, 1995) or the Deviance Information Criterion (Spiegelhalter et al., 2002, 2014). Within a Bayesian framework Fokoué and Titterington (2003) use a stochastic model selection approach but do not simultaneously choose the optimal number of clusters and factors. Conducting an exhaustive search of the model space is computationally expensive; the cost is typically reduced by only considering models in which the number of factors is common across clusters. Regardless, even searching the reduced model space can be computationally onerous. The problem of identifying the optimal model is exacerbated by the fraught task of choosing among the range of model selection tools available, which often suggest different optimal models. Moreover, enforcing a common number of factors across clusters may lead to poor clustering performance due to a lack of flexibility.

The infinite mixture of infinite factor analysers (IMIFA) model is introduced here. It theoretically allows infinitely many components and infinitely many factors within

each component. The need to select a model selection criterion is obviated and quantification of the uncertainty in the optimal numbers of non-empty clusters and cluster-specific factors is facilitated. IMIFA relies on an infinite mixture model through the use of a nonparametric Pitman-Yor process (PYP) prior (Perman et al., 1992; Pitman and Yor, 1997), of which the well-known Dirichlet process (DP; Ferguson, 1973) is a special case. The infinite mixture model framework allows the number of clusters present to be automatically inferred; here the stick-breaking construction (Pitman, 1996) and an independent slice-efficient sampler (Kalli et al., 2011) are employed to facilitate this.

By allowing infinitely many factors within each cluster, IMIFA addresses the difficulty in choosing the optimal number of factors. This facilitates fitting factor-analytic models which are more flexible, in the sense that the number of factors may be cluster-specific, thereby potentially improving clustering performance. This is achieved by assuming multiplicative gamma process (MGP) shrinkage priors (Bhattacharya and Dunson, 2011; Durante, 2017) on the cluster-specific factor loading matrices, thus generalising the MGP prior to the mixture setting. Such a prior allows the degree of shrinkage of the factor loadings towards zero to increase as the factor number tends towards infinity. The number of factors with non-negligible loadings can be considered as the 'active' number of factors within each cluster. Following Bhattacharya and Dunson (2011), a computationally efficient adaptive Gibbs sampling algorithm is employed for estimation. Thus, the choice of the numbers of active factors in different clusters is automated.

The IMIFA model with its PYP-MGP priors thus offers a single-pass and therefore computationally efficient approach to clustering high-dimensional data. It can be viewed as the most flexible model at the head of a family of Bayesian factor-analytic mixture models. Section 4.2 develops the hierarchy of the IMIFA model family, beginning with the MFA model and concluding with the flagship IMIFA model. Between these extremes the novel finite mixture of infinite factor analysers model (MIFA) is introduced. Overfitted factor-analytic mixtures (Papastamoulis, 2018) also belong to the IMIFA family; the overfitted mixture of infinite factor analysers (OMIFA) model is also introduced here.

Section 4.3 considers implementation of the IMIFA family of models. A benchmarking experiment is conducted on the well-known Italian olive oil data set. A

real data application follows through the cluster analysis of spectral metabolomic data from an epilepsy study. Finally an illustrative application is provided through clustering United States Postal Service handwritten digit data, a setting for which fitting sub-models of the IMIFA family is practically infeasible. Comparisons against other clustering methods are provided throughout. Simulation studies demonstrating the performance of IMIFA under different scenarios are deferred to Appendix 4.B. Section 4.4 concludes the article with a discussion of IMIFA and thoughts on future research directions.

A software implementation for IMIFA and its family of sub-models is provided by the associated R package `IMIFA` (Murphy et al., 2019), which is freely available from `www.r-project.org` (R Core Team, 2019), with which all results were generated.

4.2 The IMIFA Model Family

The hierarchy of the IMIFA family of models is delineated herein, including a review of extant methodologies, the introduction of novel sub-models, and concluding with the flagship IMIFA model. Prior specifications, Markov chain Monte Carlo (MCMC) inferential procedures, approaches to posterior predictive model checking, and model-specific implementation issues that arise in practice are addressed.

4.2.1 Mixtures of Factor Analysers

Mixtures of factor analysers are Gaussian latent variable models used for clustering high-dimensional data. For each of G clusters in these finite mixtures, the cluster-specific FA model in cluster g is given by $\mathbf{x}_i - \boldsymbol{\mu}_g = \boldsymbol{\Lambda}_g \boldsymbol{\eta}_i + \boldsymbol{\varepsilon}_{ig}$. The observed feature vector $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top$ with mean $\boldsymbol{\mu}_g$ and covariance matrix $\boldsymbol{\Sigma}_g$ is assumed to linearly depend on a q-vector $(q \ll p)$ of latent common factor scores $\boldsymbol{\eta}_i$ and additional sources of variation called specific factors $\boldsymbol{\varepsilon}_{ig}$. It is assumed that $\boldsymbol{\eta}_i$ has a q-variate Gaussian distribution $\mathrm{N}_q(\mathbf{0}, \boldsymbol{\mathcal{I}}_q)$, where $\boldsymbol{\mathcal{I}}_q$ denotes the $q \times q$ identity matrix, and that $\boldsymbol{\varepsilon}_{ig} \sim \mathrm{N}_p(\mathbf{0}, \boldsymbol{\Psi}_g)$, where $\boldsymbol{\Psi}_g$ is a diagonal matrix with non-zero elements $\psi_{1g}, \ldots, \psi_{pg}$ known as uniquenesses. Here, $\boldsymbol{\Lambda}_g$ denotes the $p \times q$ factor loadings matrix of cluster g and notably $q = 0$ is permitted.

To facilitate estimation, a latent cluster indicator vector $\mathbf{z}_i = (z_{i1}, \ldots, z_{iG})^\top$ is introduced such that $z_{ig} = 1$ if observation i belongs to cluster g and $z_{ig} = 0$ otherwise. Hence, $\mathbf{z}_i$ has a $\mathrm{Mult}(1, \boldsymbol{\pi})$ distribution where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_G)^\top$ are the cluster mixing proportions which sum to 1. A symmetric uniform Dirichlet prior $\boldsymbol{\pi} \sim \mathrm{Dir}(\boldsymbol{\alpha} = (\alpha, \ldots, \alpha) = \mathbf{1})$ is assumed. Upon marginalising out $\mathbf{z}_i$ and $\boldsymbol{\eta}_i$, MFA yields a parsimonious finite sum covariance structure for the observed data

$$f(\mathbf{x}_i \mid \boldsymbol{\theta}) = \sum_{g=1}^{G} \pi_g \mathrm{N}_p\big(\mathbf{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g \boldsymbol{\Lambda}_g^\top + \boldsymbol{\Psi}_g\big), \tag{4.1}$$

where $\mathrm{N}_p(\mathbf{x}_i; \cdot, \cdot)$ denotes the density of a p-variate Gaussian distribution evaluated at $\mathbf{x}_i$ and $\boldsymbol{\theta}_g = \{\boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\Psi}_g\}$ are the cluster-specific FA parameters for which inference is straightforward under a Gibbs sampling scheme. Imposing constraints on $\boldsymbol{\Psi}_g$ (McNicholas and Murphy, 2008) and/or fixing $\pi_g = {}^1/_G \ \forall \ g$ may be useful in some settings.

4.2.1.1 Prior Specification and Practical Issues

The conditionally conjugate nature of the various prior distributions detailed below facilitates MCMC sampling via straightforward Gibbs updates. A multivariate Gaussian prior is assumed for the factor loadings of the variable j across the q factors of cluster g:

$$\boldsymbol{\Lambda}_{jg} = (\lambda_{j1g}, \ldots, \lambda_{jqg}) \sim \mathrm{N}_q(\mathbf{0}, \boldsymbol{\mathcal{I}}_q)\,.$$

Similarly, a diffuse multivariate Gaussian prior is assumed for the component means,

$$\boldsymbol{\mu}_g \sim \mathrm{N}_p\big(\widetilde{\boldsymbol{\mu}}, \varphi^{-1}\boldsymbol{\mathcal{I}}_p\big)\,,$$

where $\widetilde{\boldsymbol{\mu}}$ is the overall sample mean and the scalar φ controls the level of diffusion.

An inverse gamma prior $\psi_{jg} \sim \mathrm{IG}(\alpha_0, \beta_j)$ is assumed for the uniquenesses of variable j in cluster g. Guided by Frühwirth-Schnatter and Lopes (2010), hyperparameters are chosen to ensure ψ_{jg} is bounded away from 0, thereby avoiding Heywood problems. With a sufficiently large shape α_0, variable-specific scales are derived from the sample precision matrix $\mathbf{S}^\star = \mathbf{S}^{-1}$ via $\beta_j = (\alpha_0 - 1)/S^\star_{jj}$. However, when ${}^N/_p$ is close to or less than 1, or when $\mathbf{S}^{-1}$ is otherwise unavailable, $\mathbf{S}^\star$ is replaced by a ridge-type estimator $\widehat{\mathbf{S}^{-1}} = \big(\beta_0 + {}^N/_2\big)\big(\beta_0\boldsymbol{\mathcal{I}}_p + 0.5\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^\top\big)^{-1}$,

which combines the the inverse Wishart prior $\mathbf{S}^{-1} \sim \mathrm{W}_p\left(\beta_0, \beta_0 \boldsymbol{\mathcal{I}}_p\right)$ with the sample information, where β_0 is a hyperparameter (Frühwirth-Schnatter and Lopes, 2018). For unstandardised data, this estimator is constructed for the inverse correlation matrix and then appropriately scaled using the diagonal entries of $\mathbf{S}$ (Wang et al., 2015). When the variances are roughly balanced, constraining $\boldsymbol{\Psi}_g$ to $\psi_g \boldsymbol{\mathcal{I}}_p$, and/or using $\beta_j = \beta = (\alpha_0 - 1)/\max(\mathrm{diag}(\mathbf{S}^\star))$, provides additional parsimony. Notably, the isotropic constraint provides the link between factor analysis and probabilistic principal component analysis (Tipping and Bishop, 1999).

The rotational invariance property which makes FA models non-identifiable is well known: most covariance matrices $\boldsymbol{\Sigma}$ cannot be uniquely factored as $\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Psi}$ when $q > 1$. Though identifiability of $\boldsymbol{\Lambda}$ is not strictly necessary for the purposes of clustering or inferring $\boldsymbol{\Sigma}$, addressing the identifiability problem offline using the parameter expanded approach of Ghosh and Dunson (2008) in tandem with Procrustean methods, as in McParland et al. (2014), yields interpretable posterior summaries. Another practical issue is the label switching phenomenon (Frühwirth-Schnatter, 2010) which is addressed offline using the cost-minimising permutation given by the square assignment algorithm (Carpaneto and Toth, 1980). Finally, optimal FA and MFA models are chosen using the BIC-MCMC criterion (Frühwirth-Schnatter, 2011) where necessary in what follows.

4.2.2 Mixtures of Infinite Factor Analysers

To overcome the requirement to specify q, infinite factor analysis (IFA) models are employed (Bhattacharya and Dunson, 2011). The IFA model is a factor analysis model which assumes a multiplicative gamma process (MGP) shrinkage prior on the loadings matrix. This prior allows the degree of shrinkage towards zero to increase as the column index $k \to \infty$, mitigating against the factor splitting phenomenon. Here the IFA model is generalised to the mixture setting, leading to the novel mixture of infinite factor analysers (MIFA) model. Under MIFA, the MGP prior is placed on each parameter expanded $\boldsymbol{\Lambda}_g$ matrix with no restrictions on its entries, thereby making the induced prior on $\boldsymbol{\Sigma}_g$ invariant to the ordering of the variables. The MGP prior is conditionally conjugate, facilitating block Gibbs updates of the loadings and hence rapid mixing. Thus, the MGP prior in mixture settings is given by

$$\lambda_{jkg} \,|\, \phi_{jkg}, \tau_{kg}, \sigma_g \sim \mathrm{N}_1\big(0, \phi_{jkg}^{-1}\tau_{kg}^{-1}\sigma_g^{-1}\big)\,, \qquad \phi_{jkg} \sim \mathrm{Ga}(\nu_1, \nu_2)\,,$$
$$\tau_{kg} = \prod_{h=1}^{k} \delta_{hg}\,, \qquad \sigma_g \sim \mathrm{Ga}(\varrho_1, \varrho_2)\,,$$
$$\delta_{1g} \sim \mathrm{Ga}(\alpha_1, \beta_1)\,, \qquad \delta_{hg} \sim \mathrm{Ga}(\alpha_2, \beta_2) \;\; \forall \;\; h \geq 2,$$

where τ_{kg} is a column shrinkage parameter for the k-th column in the g-th cluster's loadings matrix $\mathbf{\Lambda}_g \;\forall\; k = 1, \ldots, \infty$, and $\mathrm{Ga}(\alpha, \beta)$ denotes the gamma distribution with mean $\alpha\beta$. The role of the local shrinkage parameters $\phi_{1kg}, \ldots, \phi_{pkg}$ for the p elements in column k of $\mathbf{\Lambda}_g$ is to favour sparsity while also preserving the signal of non-zero loadings. Lastly, the cluster shrinkage parameter σ_g reflects the belief that the degree of shrinkage is cluster-specific. A schematic illustration of the MGP prior is given in Figure 4.1; note that loadings can shrink arbitrarily close, but not exactly, to zero.

Bhattacharya and Dunson (2011) fix $\beta_1 = \beta_2 = 1$ and recommend that $\alpha_2 > \beta_2$. However, Durante (2017) elaborates on the cumulative shrinkage properties and roles played by hyperparameters, showing in particular that $\alpha_2 > \beta_2 + 1$ is necessary in order to have column-specific variances τ_{kg}^{-1} that decrease in expectation with growing k. It is also recommended that α_2 be moderately large relative to α_1 (to ensure that the cumulative shrinkage property for which the prior was developed holds) and to avoid excessively high values for α_1 (to avoid over-shrinking to increasingly low-dimensional factorisations). While Bhattacharya and Dunson (2011) assume $\mathrm{Ga}(\nu, \nu)$ priors for the local shrinkage parameters, here more general settings are used to allow control over prior non-informativity. In the spirit of Durante (2017), the expectation $\nu_2/(\nu_1 - 1)$ of the induced inverse gamma prior on ϕ_{jkg}^{-1} is suggested to be ≤ 1 to induce sparsity on average. Furthermore, following the guidelines of Durante (2017), it is generally advisable that all MGP hyperparameters are chosen such that the first two moments of the associated hyperprior are defined, as this leads to superior performance in terms of the expected deviation between the true and estimated covariance matrices. In the mixture setting, α_1 and α_2 may need to be higher than the values suggested by Durante (2017) to enforce a greater degree of shrinkage in clusters with few units; this aspect is highlighted in simulation studies in Appendix 4.B.

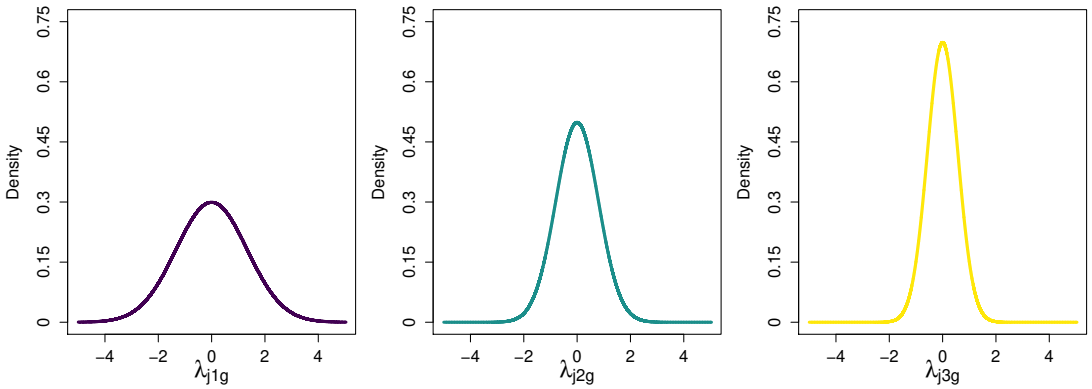


Figure 4.1: Density of a typical element in the first, second, and third columns of a cluster-specific loadings matrix under the MGP shrinkage prior.

4.2.2.1 The Adaptive Gibbs Sampler

An adaptive Gibbs sampler (AGS) is employed when performing inference for MIFA. This dynamically shrinks the loadings matrices (and the infinite scores matrix $\boldsymbol{\eta}$) to have finite numbers of columns, by selecting the number of 'active' factors. This practically facilitates posterior computation while closely approximating the IFA model, without requiring specification of $\mathbf{Q} = (q_1, \ldots, q_G)^\top$. However, a strategy is required for choosing appropriate truncation levels, $\widehat{q}_g$, that strike a balance between missing important factors and wasting computational effort. For computational reasons, a conservatively high upper bound is used, such that $q_g^\star = \min\big(\lfloor 3(p) \rceil, N-1, p-1\big) \ \forall \ g$. The number of factors in each $\boldsymbol{\Lambda}_g$ is then adaptively tuned as the MCMC chain progresses. Adaptation can be made to occur only after the burn-in period, in order to ensure the true posterior distribution is being sampled from before truncating the loadings matrices.

At the t-th iteration, adaptation occurs with probability $\mathrm{p}(t) = \exp(-b_0 - b_1 t)$, with b_0 and b_1 chosen so that adaptation occurs often at the beginning of the chain but then decreases exponentially fast in frequency. Here $b_0 = 0.1$ and $b_1 = 5 \times 10^{-5}$ are used. With probability $\mathrm{p}(t)$, loadings columns having some prespecified proportion of elements ς in a small neighbourhood ϵ of zero are monitored. If there are no such columns, an additional column is added by simulation from the MGP prior. Otherwise redundant columns are discarded and the AGS proceeds with all parameters corresponding to non-redundant columns retained.

Choice of ς and ϵ can be delicate, as there is an implicit trade-off between these two fixed tuning parameters; smaller ς and larger ϵ speed up the algorithm by favouring the discarding of factors during the adaptation step, and *vice versa*. Typically, ς should be kept close to 1 and ϵ should be kept small, relative to the scale of the data. Here, $\varsigma = \lfloor 0.7 \times p \rfloor / p$ and $\epsilon = 0.1$ are found to strike an appropriate balance. The dimension of the matrix $\boldsymbol{\eta}$ of factor scores at a given iteration are set to $p \times \overline{q} = p \times \max(\mathbf{Q}(t))$; rows corresponding to observations currently assigned to a cluster with fewer latent factors than $\overline{q}$ are padded with zeros. Notably, here $\widehat{q}_g$ may shrink to 0 thus allowing diagonal covariance structure within a component. If this occurs, the decision to simulate a new column is based on a binary trial with probability $1 - \varsigma$ as there are no loadings columns to monitor.

The numbers of active factors in each cluster for each retained posterior sample can be used to construct a barchart approximation to the posterior distribution of q_g. The posterior mode is used to estimate each q_g, with credible intervals quantifying uncertainty. Another strategy, which circumvents the need to pre-specify ς and ϵ, is to forego adaptation (provided the computational burden of doing so is tolerable) and estimate $\widehat{q}_g$ from the number of non-redundant columns in the posterior mean loadings matrices. However, this approach is not considered further here.

In any case, the main advantages of MIFA are that different clusters can be modelled by different numbers of factors and that the model search is reduced to one for G only, as q_g is estimated automatically during model fitting. Here, for MIFA models, the optimal G is chosen via the BICM (BIC-Monte (Carlo)) proposed by Raftery et al. (2007), with $\text{BICM} = 2\ln\left(\overline{\mathcal{L}}\right) - 2s_l^2\left(\ln(N) - 1\right)$, where $\overline{\mathcal{L}}$ and s_l^2 are the sample mean and sample variance, respectively, of the log-likelihood values calculated for each retained posterior sample. This criterion is particularly useful in the context of nonparametric models where the number of free parameters is difficult to quantify, though we caution that it may be biased in favour of $G = 1$ models, under which the log-likelihoods tend to exhibit less variability, and that a large number of posterior samples are required to ensure stable estimation of s_l^2.

4.2.2.2 Other Infinite Factor Models

This work offers an extension of the MGP prior and its related AGS routine to the mixture modelling context. Wang et al. (2016) develop a related model employing a multiplicative exponential process prior. Other nonparametric approaches to inferring the number of factors include Knowles and Ghahramani (2007), in which a two-parameter Indian Buffet Process (IBP) prior is assumed on an infinite binary matrix underlying the factor scores, thus selecting features of interest, with associated standard Gaussian weights. A closely related approach using the Beta process (BP) is provided by Paisley and Carin (2009). In Knowles and Ghahramani (2011) and Ročková and George (2016), an IBP prior is instead assumed for sparsifying the loadings. These models assume a single sparse infinite factor model for the whole data set. However, embedding them in a mixture modelling setting, similar to the IMIFA framework, is intuitively feasible.

Indeed, Chen et al. (2010) employ the BP prior, coupled with a Dirichlet process prior, to perform clustering in a manifold learning setting. While the BP and IBP priors achieve exact sparsity, which may be advantageous in certain applications, the MGP prior has a weaker notion of sparsity by virtue of cumulatively shrinking an infinite series arbitrarily close to zero, thereby preserving small signals. The block updates of each row of $\mathbf{\Lambda}_g$ facilitated by the MGP prior and parameter expansion mean the AGS approach is a simpler, more computationally efficient alternative to the BP and IBP priors.

4.2.3 Overfitted Mixtures of (Infinite) Factor Analysers

While MIFA obviates the need to pre-specify $\mathbf{Q}$, the issue of model choice is not yet fully resolved. Overfitted mixtures (Rousseau and Mengersen, 2011; van Havre et al., 2015) are one means of extending MIFA; indeed Papastamoulis (2018) proposes an overfitted mixture of factor analysers (OMFA), albeit with finite factors. Here, the overfitted mixture of infinite factor analysers (OMIFA) model is introduced.

In overfitted mixtures the symmetric Dirichlet prior on π plays an important role. Estimation is approached by initially overfitting the number of clusters expected to be present. Small values of the hyperparameter α encourage emptying out ex-

cess components in the posterior distribution; the uniform prior with $\boldsymbol{\alpha} = \mathbf{1}$ is rather indifferent in this respect. The sampler is initialised with a conservatively high number of components: $G^\star = \max\big(\lceil 3\ln(N)\rceil, 25, N-1\big)$, though this may be too high if it is close to N. While $\widetilde{G} = G^\star$ remains fixed throughout the MCMC chain, the number of non-empty clusters is recorded at each iteration of the sampler as $G_0 = \widetilde{G} - \sum_{g=1}^{\widetilde{G}} \mathbb{1}\big(\sum_{i=1}^{N} z_{ig} = 0\big)$ where $\mathbb{1}(\cdot)$ is the indicator function. The true G is estimated by $\widehat{G}$, the G_0 value visited most often. Cluster-specific inference is conducted only on samples corresponding to those visits. For the OMIFA model, the AGS is modified to handle empty components: the MGP-related parameters are simulated from the relevant priors and each corresponding $\boldsymbol{\Lambda}_g$ matrix is restricted to having $\overline{q}$ factors, i.e. the same number of columns currently in the matrix of factor scores $\boldsymbol{\eta}$, either by truncation or by padding with zeros, as required.

4.2.4 Infinite Mixtures of (Infinite) Factor Analysers

Embedding MFA and MIFA in an infinite mixture setting leads, respectively, to the infinite mixture of finite factor analysers model (IMFA) and the flagship infinite mixture of infinite factor analysers model (IMIFA). These models employ a nonparametric Pitman-Yor process (PYP) prior which is easily incorporated into the MCMC sampling scheme.

The PYP is a stochastic process whose draws are discrete probability measures, whereby $H \sim \mathrm{PYP}(\alpha, d, H_0)$ denotes a PYP probability distribution H, with base distribution H_0 interpreted as the mean of the PYP, discount parameter $d \in [0, 1)$, and concentration parameter $\alpha > -d$. For the PYP mixture model IMFA and the PYP-MGP mixture model IMIFA H_0 comes from the factor-analytic mixture (4.1), hence

$$f(\mathbf{x}_i \,|\, \boldsymbol{\theta}) = \sum_{g=1}^{\infty} \pi_g \mathrm{N}_p\big(\mathbf{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g^\top + \boldsymbol{\Psi}_g\big). \tag{4.2}$$

The stick-breaking representation of the PYP (Pitman, 1996) is used as a prior process for generating the mixing proportions in (4.2). This construction views $\{\pi_1, \pi_2, \ldots\}$ as pieces of a unit-length stick that is sequentially broken in an infinite process, with stick-breaking proportions $\boldsymbol{\Upsilon} = \{\upsilon_1, \upsilon_2, \ldots\}$, summarised as

$$\upsilon_g \sim \text{Beta}(1-d, \alpha+gd)\,, \qquad \boldsymbol{\theta}_g \sim H_0\,,$$
$$\pi_g = \upsilon_g \prod_{l=1}^{g-1}(1-\upsilon_l)\,, \qquad H = \sum_{g=1}^{\infty} \pi_g \delta_{\boldsymbol{\theta}_g} \sim \text{PYP}(\alpha, d, H_0)\,,$$

where $\delta_{\boldsymbol{\theta}_g}$ is the Dirac delta centred at $\boldsymbol{\theta}_g$, such that draws are composed of a sum of infinitely many point masses. The PYP reduces to the DP when $d = 0$, in which case mass shifts to the right with increasing dispersion as α increases, implying an *a priori* larger number of components. However, some important distributional features fundamentally differ when $d \neq 0$ (De Blasi et al., 2015). The PYP exhibits heavier tail behaviour and allows the stick-breaking distribution to vary according to the component index g, without sacrificing much in the way of tractability. In particular, increasing d values have the effect of flattening the prior, controlling its degree of non-informativity (see Appendix 4.E).

Slice sampling (Walker, 2007; Kalli et al., 2011) is used here to yield samples from the PYP by adaptively truncating the number of components needed to be sampled at each iteration. By introducing an auxiliary variable $u_i > 0$ which preserves the marginal distribution of the data, and denoting by $\boldsymbol{\xi} = \{\xi_1, \xi_2, \ldots\}$ a positive sequence of infinite quantities which sum to 1, the joint density of $(\mathbf{x}, \mathbf{u})$ is given by $f(\mathbf{x}, \mathbf{u} \,|\, \boldsymbol{\theta}, \boldsymbol{\xi}) = \sum_{g=1}^{\infty} \pi_g \text{Unif}(\mathbf{u}; 0, \xi_g)\, f(\mathbf{x} \,|\, \boldsymbol{\theta}_g)$. Since only a finite number of ξ_g are greater than $\mathbf{u}$, the conditional density of $\mathbf{x} \,|\, \mathbf{u}$ can be written as a finite mixture with $\widetilde{G} = \max_{\{1 \leq i \leq N\}} (|\mathcal{A}_{\boldsymbol{\xi}}(u_i)|)$ 'active' components at each iteration, where $|\cdot|$ denotes cardinality and $\mathcal{A}_{\boldsymbol{\xi}}(\mathbf{u}) = \{g : \mathbf{u} < \xi_g\}$. Though G is infinite in theory, $\widetilde{G}$ can be at most equal to N. Thus, the infinite mixture of (infinite) factor analysers models can be sampled from.

Typical implementations of the slice sampler arise when $\xi_g = \pi_g$ (Walker, 2007) but independent slice-efficient sampling (Kalli et al., 2011) allows for a deterministic decreasing sequence, e.g. geometric decay, given by $\xi_g = (1-\rho)\,\rho^{g-1}$ where $\rho \in [0, 1)$ is a fixed value to be chosen with care. Higher values generally lead to better mixing but longer run-times, as the average cardinality of $\mathcal{A}_{\boldsymbol{\xi}}(\mathbf{u})$ increases, and *vice versa*. Setting $\rho = 0.75$, in line with the recommendations of Kalli et al. (2011), appears to strike an appropriate balance in the applications considered here.

4.2.4.1 Inference for Infinite Mixtures of Factor Analysers Models

For clarity, what follows focuses on the IMIFA model where inference proceeds via the independent slice-efficient sampler with geometric decay. Inference for other models in the IMIFA family is closely related. The joint density of the IMIFA model is

$$\begin{aligned} f(\mathbf{X}, \boldsymbol{\eta}, \mathbf{Z}, \mathbf{u}, \boldsymbol{\Upsilon}, \boldsymbol{\theta}) &\propto f(\mathbf{X} \mid \boldsymbol{\eta}, \mathbf{Z}, \mathbf{u}, \boldsymbol{\Upsilon}, \boldsymbol{\theta})\, f(\boldsymbol{\eta})\, f(\mathbf{Z}, \mathbf{u} \mid \boldsymbol{\Upsilon}, \boldsymbol{\pi})\, f(\boldsymbol{\Upsilon} \mid \alpha, d)\, f(\boldsymbol{\theta}) \\ &= \left\{ \prod_{i=1}^{N} \prod_{g \in \mathcal{A}_{\boldsymbol{\xi}}(u_i)} \mathrm{N}_p\left(\mathbf{x}_i; \boldsymbol{\mu}_g + \boldsymbol{\Lambda}_g \boldsymbol{\eta}_i, \boldsymbol{\Psi}_g\right)^{z_{ig}} \right\} \left\{ \prod_{i=1}^{N} \mathrm{N}_q\left(\boldsymbol{\eta}_i; \mathbf{0}, \boldsymbol{\mathcal{I}}_q\right) \right\} \\ &\quad \left\{ \prod_{i=1}^{N} \prod_{g=1}^{\infty} \left(\frac{\pi_g}{\xi_g} \mathbb{1}\left(u_i < \xi_g\right) \right)^{z_{ig}} \right\} \left\{ \prod_{g=1}^{\infty} \frac{(1-\upsilon_g)^{\alpha + gd - 1}}{\upsilon_g^d\, \mathrm{B}(1-d, \alpha + gd)} \right\} f(\boldsymbol{\theta}), \end{aligned}$$

where $\mathrm{B}(\cdot)$ is the Beta function and $f(\boldsymbol{\theta})$ is the product of the previously defined collection of conditionally conjugate priors with additional layers for hyperparameters. Only the parameters of the $\widetilde{G}$ active components are sampled at each iteration. The algorithm is initialised with the same $G^\star$ value detailed in Section 4.2.3, typically above the anticipated number to which the algorithm will converge, in the spirit of Hastie et al. (2014). Here, however, $\widetilde{G}$ can theoretically exceed this value. For computational reasons, a finite upper limit is placed on $\widetilde{G}$ with $\max(G^\star, \min(N-1, 50))$ found to be sufficiently large. However, $\widetilde{G}$ is only regarded as a set of proposals as to where to allocate observations; as in Section 4.2.3, it is the subset of non-empty clusters G_0 that is of inferential interest.

Bayesian approaches to clustering are known to be sensitive to initial cluster allocations. While starting values for $\mathbf{z}_i$ can be obtained by any means, model-based agglomerative hierarchical clustering (Scrucca et al., 2016) is used here. Though this is fast and intuitive given that IMIFA models are initialised at a conservatively high number of components, which are then merged as the sampler proceeds, heavily imbalanced initial cluster sizes are cautioned against. By extension, initial cluster means and mixing proportions are computed empirically. Other parameter starting values are simulated from their relevant prior distributions. The adaptive inferential algorithm for IMIFA then proceeds mostly via Gibbs updates. For those which are multivariate Gaussian, using the Cholesky factor of the covariance matrices and employing block updates speeds up the algorithm (Rue and

Held, 2005). The allocations $\mathbf{z}_i$ are sampled in a fast, numerically stable fashion, using the Gumbel-Max trick (Yellott, 1977). Finally, state spaces for applications of IMIFA to real data can be highly multimodal with well-separated regions of high posterior probability coexisting, corresponding to clusterings with different numbers of components. Thus, label switching moves (Papaspiliopoulos and Roberts, 2008) are incorporated in order to improve mixing. Details of the Gibbs updates, Gumbel-Max trick, and label switching moves are provided in Appendix 4.A.

4.2.4.2 Assessing Model Fit and Mixing

As is good statistical practice, posterior predictive model checking (Gelman et al., 2004) is employed. Sampled model parameters from the MCMC chain are used to generate replicate data from the posterior predictive distribution. Valid posterior samples, after conditioning on $\widehat{G}$, are those for which $\max(\mathbf{Q}(t)) \geq \max(\widehat{q}_1, \ldots, \widehat{q}_{\widehat{G}})$ such that the dimension of the estimated scores matrix $\widehat{\eta}$ is preserved. To assess model fit, histograms of the modelled data $\mathbf{X}$ are compared to histograms of the replicate data in a global sense using the Posterior Predictive Reconstruction Error (PPRE), calculated as follows:

1. Gather the histogram bin counts of each variable in $\mathbf{X}$ into the $h \times p$ matrix $\mathcal{H}$, where h is the maximum number of bins across all variables and $\mathcal{H}$ is padded with zeros as required.

2. Generate $r \in \{1, \ldots, R\}$ data sets $\mathcal{X}^{(r)}$ from the posterior predictive distribution.

3. Create a similar matrix of histogram bin counts $\mathcal{H}^{(r)}$ for each $\mathcal{X}^{(r)}$ using the same break-points with which $\mathcal{H}$ was constructed (with endpoint bins extended to $\pm\infty$).

4. Compute the Frobenius norm $\|\cdot\|_{\mathcal{F}}$ between $\mathcal{H}$ and $\mathcal{H}^{(r)}$, standardising to the 0-1 scale using the triangle inequality:

$$\left| \|\mathcal{H}\|_{\mathcal{F}} - \|\mathcal{H}^{(r)}\|_{\mathcal{F}} \right| \leq \|\mathcal{H} - \mathcal{H}^{(r)}\|_{\mathcal{F}} \leq \|\mathcal{H}\|_{\mathcal{F}} + \|\mathcal{H}^{(r)}\|_{\mathcal{F}}.$$

The distribution of PPRE values can be visualised using boxplots and summarised by the median, with credible intervals quantifying uncertainty. This discrepancy measure is well-suited to assessing model adequacy for mixtures of multivariate data: it accounts for inherent multimodality and gives a global quantitative measure of agreement between the distributions of the observed variables and their posterior predictive counterparts.

Convergence of the MCMC chains is assessed using the potential scale reduction factor (PSRF; Brooks and Gelman, 1998; Plummer et al., 2006). Random allocations of the initial cluster labels, resulting in different draws from the relevant priors for parameter initialisation, are used to construct the multiple overdispersed chains required. The MAP labels of each chain are matched to the main chain prior to computing the diagnostics; $\mathbf{\Lambda}_g$ matrices are also rotated to a common template for each cluster. Good convergence is indicated by upper PSRF 95% confidence interval limits close to 1; this is a stricter requirement than the PSRF values themselves being near 1.

4.2.4.3 Comparing the IMIFA Family Models

Though IMIFA and OMIFA come with the computational complexities inherent in nonparametric methods, diminishing adaptation, and extra tuning parameters, their advantages over other models in the IMIFA family are numerous: i) flexibility, in the sense that models where $q_g \neq q_g'$ can be fitted, ii) computational efficiency, in the sense that the burden is reduced relative to searching over a range of fitted MFA or MIFA models, iii) removing the need for model selection criteria, and iv) the ability to quantify the uncertainty in $\widehat{G}$ and $\widehat{q}_g$. Both methods offer simpler alternatives to reversible jump MCMC (Richardson and Green, 1997) and birth-death MCMC (Stephens, 2000). Hence, among the IMIFA family, the infinite factor models are recommended over the finite factor models and the infinite and overfitted mixtures are recommended over the finite mixtures. However, the MIFA model is appropriate if one wishes to fix G but infer q_g.

While infinite mixtures are often used for density estimation, they are also employed to infer the number of components in cluster analyses (e.g. Kim et al. 2006; Xing et al. 2006; Yerebakan et al. 2014). However, Miller and Harrison (2013,

2014) raise concerns about the guarantee of posterior consistency for the number of non-empty clusters, showing the number uncovered is typically greater than or equal to the truth, often with several vanishingly small clusters inferred. These concerns highlight the need for practitioners to pay due consideration to the uncertainty in the number of clusters offered by IMIFA models. Relatedly, Frühwirth-Schnatter and Malsiner-Walli (2019) compare infinite mixtures to overfitted ('sparse finite') mixtures. They highlight that overfitted mixtures are useful for applications in which the data arise from a moderate number of clusters, even as the sample size increases, whereas infinite mixtures are suited to cases where the number of clusters also increases. However, they show that clustering results are driven less by the assumption of whether the data arose from a finite or infinite mixture, but by the hyperprior on the DP parameters or the sparseness of the Dirichlet prior in the overfitted setting. Indeed, they show that overfitted and infinite mixtures yield comparable clustering performance on the observed data when these hyperpriors are matched. This matching leads to 'sparse' infinite mixtures that avoid overfitting the number of clusters. Similar behaviour is observed for the PYP prior in the applications in Section 4.3, where the IMIFA and OMIFA models, with matched hyperpriors, give comparable results.

The issue of choosing α can make implementing overfitted models challenging. With fixed $\alpha = \gamma/G^\star$, the prior approximates a DP with concentration parameter γ as $G^\star$ tends to infinity (Green and Richardson, 2001). Here, following Frühwirth-Schnatter and Malsiner-Walli (2019), a $\mathrm{Ga}(a, bG^\star)$ hyperprior is assumed for α. This favours small values and allows α to be updated via Metropolis-Hastings. In the infinite mixture setting, learning the PYP parameters (which also requires Metropolis-Hastings steps) and adopting the label-switching moves enables accurate inference on G_0. A joint hyperprior $\mathrm{p}(\alpha, d) = \mathrm{p}(\alpha \mid d)\,\mathrm{p}(d)$ is assumed (Carmona et al., 2019) where $\mathrm{p}(\alpha \mid d) = \mathrm{Ga}(\alpha + d; a, b)$; choosing a large b encourages clustering (Müller and Mitra, 2013). A spike-and-slab hyperprior $d \sim \kappa\delta_0 + (1-\kappa)\,\mathrm{Beta}(a', b')$ is assumed. The estimated proportion $\widehat{\kappa}$ can then be used to assess whether the data arose from a DP or a PYP at little extra computational cost. See Appendix 4.A for further details.

4.3 Illustrative Applications

The flexibility and performance of the IMIFA model and its related model family are demonstrated below through application to benchmark and real data sets. All results are obtained through the `IMIFA` R package; code to reproduce many of the results is available in the associated vignette[6]. Appendix 4.B reports on simulation studies demonstrating the performance of IMIFA under different scenarios, including effects of the N/p ratio, the PYP parameters, imbalanced cluster sizes, uncommon q_g and the degree of loadings sparsity, while Appendix 4.C explores the robustness of IMIFA.

MCMC chains were run for $50,000$ iterations, except for Section 4.3.3 in which $20,000$ were run. Every 2^{nd} sample was thinned and the first 20% of iterations were discarded as burn-in. All computations were performed on a Dell Latitude 5491 laptop, equipped with a 6-core 2.60 GHz Intel Core i7-8850H processor and 16 GB of RAM. Where necessary, the optimal finite and infinite factor models are chosen by the BIC-MCMC and BICM criteria, respectively. Throughout, $\widehat{\cdot}$ denotes the posterior mode, posterior mean, or relevant optimal value. Unless otherwise stated, data were mean-centred and unit-scaled and no constraints were imposed on the uniquenesses. Hyperprior specifications are detailed in Table 4.1. While there are many hyperparameters to select, the choices are all reasonably standard. However, poor settings may introduce additional factors or clusters to maintain flexibility and so care in specifying hyperparameters is advised.

Table 4.1: Hyperparameter specifications for the IMIFA model. Note that the specification of the beta distribution in the prior for d amounts to a standard uniform.

Parameter(s)	Hyperparameter(s)	Value(s)
$\boldsymbol{\mu}_g$	φ	0.01
$\boldsymbol{\Psi}_g$	(α_0, β_0)	$(2.5, 3)$
ϕ_{jkg}	(ν_1, ν_2)	$(3, 2)$
δ_{1g}	(α_1, β_1)	$(2.1, 1)$
δ_{kg}	(α_2, β_2)	$(3.1, 1)$
σ_g	(ϱ_1, ϱ_2)	$(3, 2)$
α	(a, b)	$(2, 4)$
d	(a', b', κ)	$(1, 1, 0.5)$

[6] https://cran.r-project.org/web/packages/IMIFA/vignettes/IMIFA.html

4.3.1 Benchmark Data: Italian Olive Oils

The Italian olive oil data (Forina and Tiscornia, 1982; Forina et al., 1983) is often clustered using factor-analytic models, e.g. McNicholas (2010). The data detail the percentage composition of 8 fatty acids in 572 Italian olive oils, known to originate from three areas: southern and northern Italy and Sardinia. Each area is composed of different regions: southern Italy comprises north Apulia, Calabria, south Apulia, and Sicily; Sardinia is divided into inland and coastal Sardinia; and northern Italy comprises Umbria and east and west Liguria. Hence, the true number of clusters is hypothesised to correspond to either 3 areas or 9 regions.

The full family of IMIFA models is fitted to the olive oil data with results detailed in Table 4.2. Models relying on pre-specification of finite ranges of G and/or q are based on $G = 1, \ldots, 9$ and $q = 0, \ldots, 6$. Clustering performance is evaluated using the adjusted Rand index (ARI; Hubert and Arabie, 1985) and the misclassification rate, compared to the 3 area labels. The α parameter is reported as its fixed value or posterior mean, as appropriate. Table 4.2 shows the flexibility and accuracy of the developed model family, and of the IMIFA model in particular which has the best clustering performance. Additionally, IMIFA is the most computationally efficient model considered, among those in the IMIFA family achieving clustering, as it requires only one run. This speed improvement would be exacerbated with larger data sets. However, methods requiring fitting of multiple models were run here in series; parallel implementations would reduce runtimes. Finally, models with different numbers of cluster-specific factors show improved clustering performance compared to the corresponding finite factor model in every case.

Table 4.2 also shows that the performance of the IMIFA model compares favourably to the best parsimonious Gaussian mixture model, fit via the `pgmm` R package (McNicholas et al., 2018) and the best mixture of factor mixture analysers (MFMA) model (Viroli, 2010), evaluated with $1, \ldots, 5$ components in both layers. Models with zero factors were not considered in either case. IMIFA also outperforms the best constrained Gaussian mixture model fitted using `mclust` (Scrucca et al., 2016). These finite mixtures are fit via maximum likelihood and use the BIC for model selection after fitting a large number of candidate models.

Table 4.2: Results of fitting a range of models, including the full IMIFA family, to the Italian olive oil data, detailing the number of candidate models explored, the run-time relative to the IMIFA run (approx. 782 seconds), the posterior mean or fixed value of α, the posterior mean of d, modal estimates of G and $\mathbf{Q}$, and the ARI and misclassification rate as evaluated against the known area labels, under the optimal or modal model as appropriate.

Model	# Models	Relative Time	α	d	G	$\mathbf{Q}$	ARI	Error (%)
IMIFA	1	1.00	0.48	0.01	4	6, 3, 6, 2	0.94	8.39
IMFA	7	4.14	0.62	0.01	5	6, 6, 6, 6, 6	0.91	14.86
OMIFA	1	1.19	0.02	—	4	6, 3, 6, 4	0.93	9.97
MIFA	9	3.41	1	—	5	6, 3, 6, 6, 4	0.92	10.31
MFA	63	13.86	1	—	2	5, 5	0.82	17.13
IFA	1	0.11	—	—	1	6	—	—
FA	7	0.37	—	—	1	6	—	—
`mclust`[†]	115	0.01	—	—	6	—	0.56	38.64
MFMA[†]	1, 350	4.68	—	—	4	5, 5, 5, 5	0.68	20.28
`pgmm`[†,‡]	588	4.46	—	—	5	6, 6, 6, 6, 6	0.53	35.84

[†] Due to the various covariance matrix decompositions considered, the results for `mclust`, MFMA, and `pgmm` are reported for the unstandardised data, for which superior clustering performance in terms of the ARI was achieved in each case.

[‡] The optimal `pgmm` model uses the UCU constraints on the uniquenesses (i.e. $\mathbf{\Psi}_g = \mathbf{\Psi}$). Among the more directly comparable unconstrained UUU models, the optimal one according to BIC has $G = 6$ components, each with 5 factors, and achieves an ARI of 0.43. Notably, the `pgmm` models chosen by BIC both have more components than the IMIFA model.

It is also notable that within the set of IMIFA models relying on information criteria, those deemed optimal were not necessarily optimal in a clustering sense. For instance, the 4-cluster MIFA model yields an ARI of 0.94 and a misclassification rate of 6.99%, with respect to the 3 area labels, despite its sub-optimal BICM. Similarly, the BICM and BIC-MCMC criteria suggest different optimal MFA models. For the IMIFA model $\widehat{\kappa} \approx 0.89$, suggesting similar inference would have resulted under a DP prior. Indeed, the results obtained by the OMIFA and OMFA models are similar to those of their infinite mixture counterparts, though the latter provide a better fit to the data (see Figure 4.5).

Figure 4.2 shows a barchart approximation to the posterior distribution of G under the IMIFA model. The modal value of 4, visited in $\approx 90\%$ of posterior samples, is used as the estimate of the true number of clusters (with 95% credible interval $[4, 5]$). Table 4.3a tabulates the MAP clustering against the 3 area labels and suggests this solution makes geographic sense, in that northern oils are cleanly split into two sub-clusters. Cluster 1 contains all of the 323 southern Italy oils: this large

cluster requires the largest number of factors ($\widehat{q}_1 = 6$ $[5, 6]$, with 95% credible intervals in brackets). Some of the other clusters require notably fewer ($\widehat{q}_2 = 3$ $[1, 6]$, $\widehat{q}_3 = 6$ $[3, 6]$, and $\widehat{q}_4 = 2$ $[1, 4]$). Table 4.3b gives the confusion matrix with oils from the north labelled by their associated region(s), yielding an ARI of 0.994 and a misclassification rate of 0.52%. Figure 4.3 shows the uncertainty in the allocations to these clusters. Only three oils have large probability of belonging to a cluster other than the one to which they were assigned by the IMIFA model.

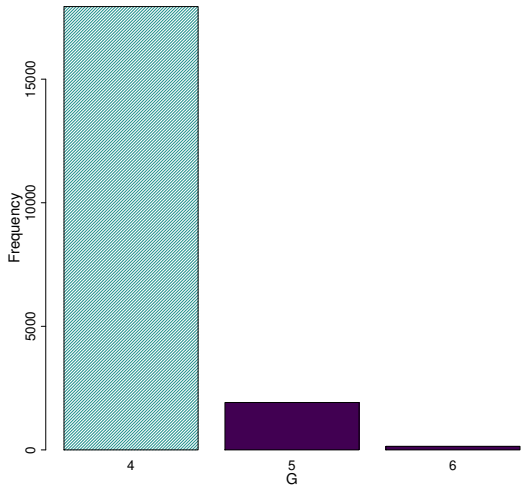


Figure 4.2: Posterior distribution of G under the IMIFA model for the olive oil data. The number of clusters is estimated by the modal value, $\widehat{G} = 4$.

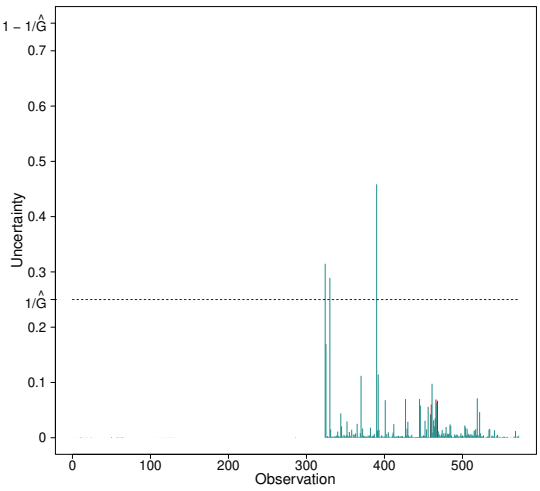


Figure 4.3: Clustering uncertainties for the IMIFA model for the olive oil data. Oils misclassified according to the labels in Table 4.3b are highlighted in red.

Table 4.3: Confusion matrices of the MAP IMIFA clustering of the Italian olive oils against (a) the known 3 area labels and (b) the new labelling in which northern Italy is split into its constituent sub-regions.

(a) 3 area cross tabulation

	1	2	3	4
Southern Italy	323	0	0	0
Sardinia	0	98	0	0
Northern Italy	0	0	103	48

(b) 4 area cross tabulation

	1	2	3	4
Southern Italy	323	0	0	0
Sardinia	0	98	0	0
East Liguria & Umbria	0	0	100	0
West Liguria	0	0	3	48

To assess sensitivity to starting values, the IMIFA model was re-fitted using multiple random initial allocations, implying also different random draws from the

priors for parameter starting values. These runs led to identical inference about $\widehat{G}$ and $\widehat{\mathbf{Q}}$ and equivalent clustering performance. These overdispersed chains were used to compute the upper 95% PSRF confidence limits depicted in Figure 4.4, which indicate good convergence. The PPRE boxplots in Figure 4.5 demonstrate the superior fit of the IMIFA model (with a median PPRE of 0.10) to the olive oil data, compared to the other IMIFA family models. Histograms comparing the bin counts between the modelled and replicate data sets for each variable, under the IMIFA model, are given in Appendix 4.D.

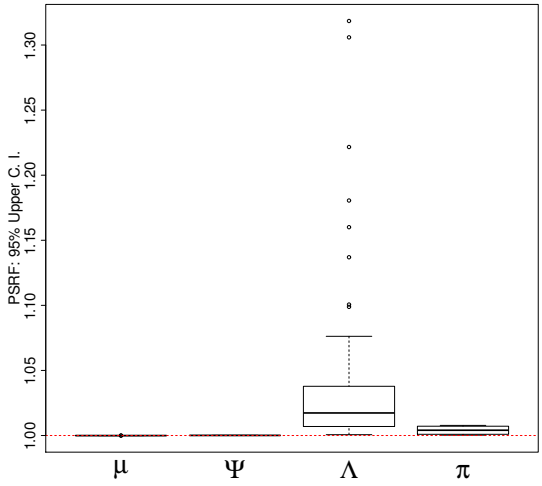


Figure 4.4: Boxplots of the upper PSRF limits for all cluster means, uniquenesses, loadings, and mixing proportions in the overdispersed IMIFA chains fit to the olive oil data, with red reference line at 1.

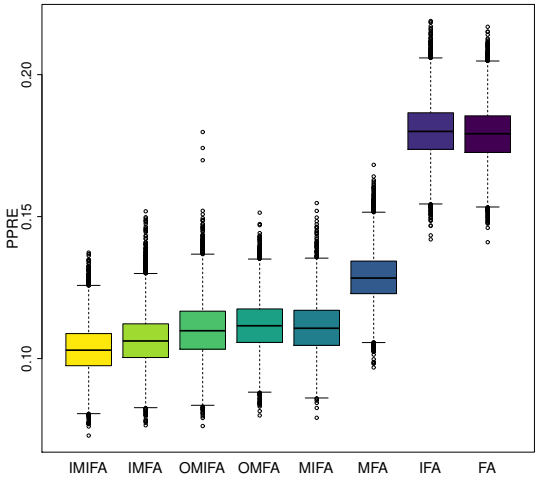


Figure 4.5: Boxplots of the PPRE values for the full family of IMIFA models fit to the olive oil data. Values close to zero indicate good model fit.

4.3.2 Spectral Metabolomic Data

IMIFA is employed to cluster spectral metabolomic data for which $N \ll p$ (Figure 4.6). The data are nuclear magnetic resonance spectra consisting of $p = 189$ spectral peaks from urine samples of $N = 18$ participants, half of which are known to have epilepsy (Carmody and Brennan, 2010; Nyamundanda et al., 2010). Interest lies in uncovering any underlying clustering structure given the $N \ll p$ setting.

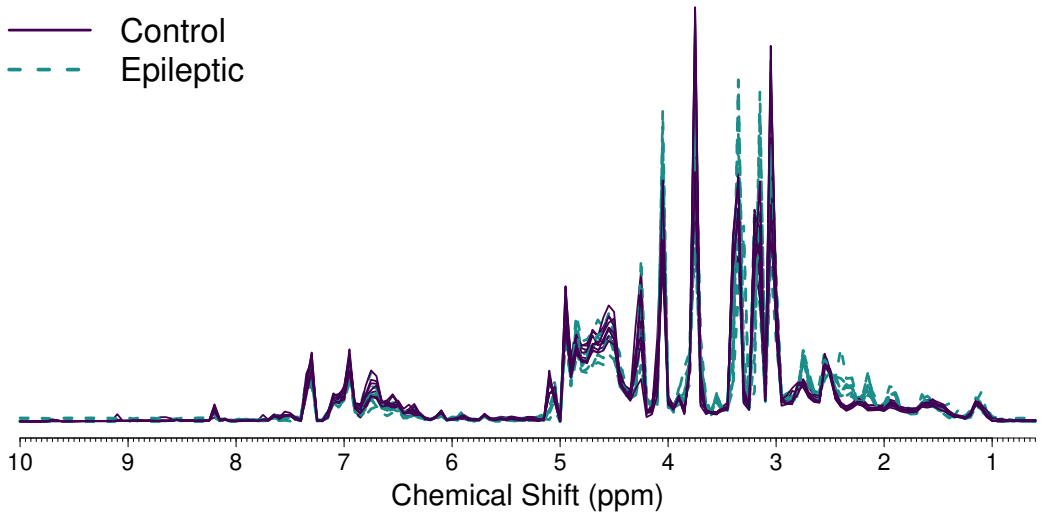


Figure 4.6: Raw spectral metabolomic data.

Data were mean-centred and Pareto scaled (van den Berg et al., 2006). Although $N \ll p$, no restrictions are imposed on the uniquenesses as the sample variances are quite imbalanced. Fitting MIFA models for $G = 1, \ldots, 5$ is feasible as N is small. The BICM criterion chooses $\widehat{G} = 2$ as optimal and one participant is misclassified. IMIFA, however, unanimously visits a 2-cluster model and perfectly uncovers the group structure.

The modal estimates of the number of factors in each IMIFA cluster are $\widehat{q}_1 = 3$ $[2, 9]$ and $\widehat{q}_2 = 5$ $[4, 13]$ (see Figure 4.7). Cluster 1 corresponds to the control group and Cluster 2 to the epileptic participants. Figure 4.8 illustrates the $p \times \widehat{q}_g$ posterior mean loadings matrices, based on retained samples with $\widehat{q}_g$ or more factors, after Procrustes rotation to a common template for both clusters. The sparsity and shrinkage induced by the MGP prior is apparent, as is the greater complexity in Cluster 2, given the greater variation in colour and larger number of factors. For instance, many elevated loadings are visible for chemical shift values between 8 and 10 for the first two factors in Cluster 2; this activity is not present for other factors in either cluster. In general, the distributions of the loadings within a factor exhibit narrow spread around zero, particularly for the cluster of control participants, with the exception of the regions of the spectrum corresponding to the large peaks between chemical shifts of 3 and 5 in Figure 4.6.

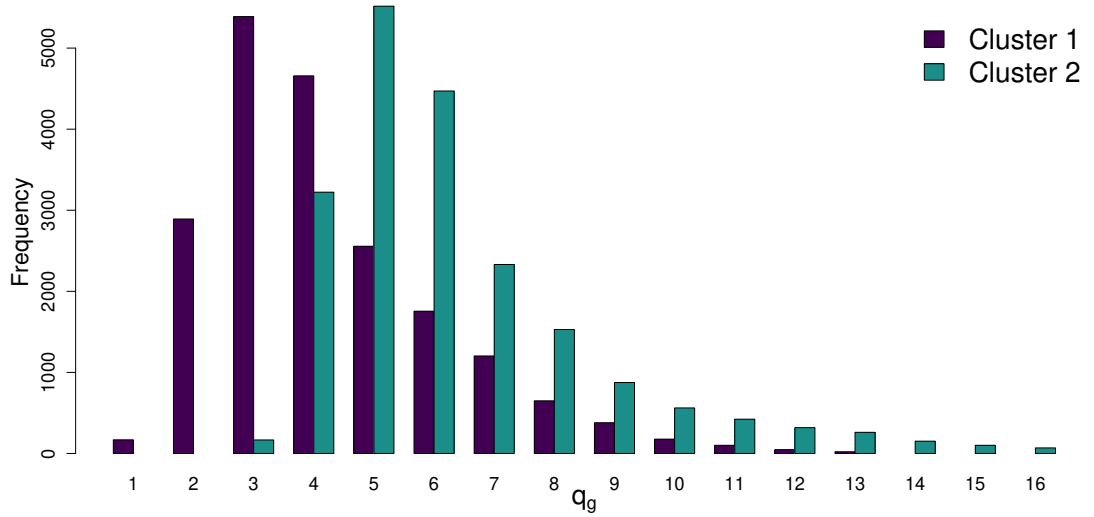


Figure 4.7: Posterior distribution of q_g under the IMIFA model fit to the metabolomic data.

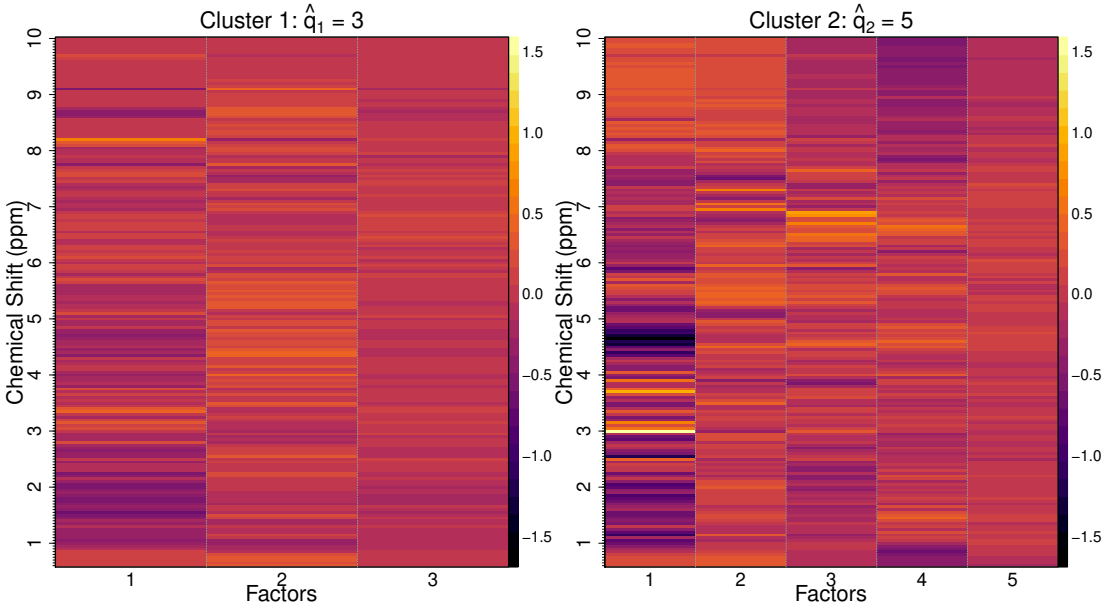


Figure 4.8: Heat maps, calibrated to a common colour scale, of posterior mean loadings matrices in the clusters uncovered by fitting IMIFA to the spectral metabolomic data.

IMIFA outperforms the optimal $\widehat{G} = 3$ `mclust` model and the optimal $\widehat{G} = 2$, $\widehat{q} = 5$ `pgmm` model, with respective ARI values of 0.73 and 0.27. The clustering performance of the optimal MFMA model is identical to the optimal MIFA model described above. Given the $N \ll p$ nature of the data, spectral clustering with

the Gaussian kernel (Ng et al., 2001) is also considered. The eigengap heuristic suggests $\widehat{G} = 2$ and a perfect clustering is achieved almost instantaneously. However, the approach does not characterise the uncovered clusters in an interpretable manner, nor provide estimates of cluster membership uncertainty as given by model-based clustering approaches such as IMIFA.

The median PPRE for the IMIFA model of 0.21 $[0.18, 0.24]$ shows good model fit, given the size and dimensionality of the data. The median PSRF upper 95% confidence limits, using three randomly initialised auxiliary chains, for the cluster means, uniquenesses, loadings, and mixing proportions of $1.01\,(0.01)$, $1.00\,(< 0.01)$, $1.01\,(0.08)$, and $1.00\,(< 0.01)$ respectively, show good mixing also (standard deviations in parentheses). Notably, all chains yield the same inference about $\widehat{G}$ and $\widehat{\mathbf{Q}}$. So too, again, does the OMIFA model, although its model fit is inferior (median PPRE=0.26).

4.3.3 Handwritten Digit Data

A final illustration of IMIFA is given through its application to handwritten digit data from the United States Postal Service (USPS; Hastie et al., 2001). Here $N = 7,291$ images of the digits $0, \ldots, 9$ are considered, taken from handwritten zip codes. The data are not balanced in terms of digit labels. Each image is a 16×16 grayscale grid concatenated into a $p = 256$-dimensional vector; data were mean-centred but not scaled. Such data are often considered in the context of manifold learning, positing that the data dimensionality is artificially high.

Given N and p, fitting a range of MFA or MIFA models is practically infeasible. Results of a single IMIFA run are presented here. For these data, it is reasonable to expect the number of components to grow as the sample size grows. It is anticipated that the flexibility afforded by having cluster-specific numbers of factors will help characterise digits with different geometric features.

The IMIFA model visited a $\widehat{G} = 21$ cluster solution in all posterior samples; Table 4.4 cross-tabulates the MAP clustering against the known digit labels and achieves an ARI of 0.33. The median PPRE of 0.05 $[0.04, 0.06]$ indicates good model fit. The overdispersed chains used to compute the PSRF diagnostics lead to identical inference about the number of clusters but slightly different inference

about the modal numbers of cluster-specific factors. The ARI values between each resulting pair of MAP partitions are all in excess of 0.93. As before, good mixing is indicated by median PSRF upper 95% confidence limits for the cluster means, uniquenesses, and mixing proportions of 1.01 (0.01), 1.01 (0.01), and 1.01 (< 0.01), respectively. In computing the diagnostic for the loadings (1.14 (0.35)), only the first factor (common to all loadings matrices across all clusters in all chains) was considered, for reasons of fairness and computational resource constraints.

Table 4.4: Cross tabulation of the IMIFA model's MAP clustering (rows) against true digit labels (columns) for the USPS data. Cells that are 0 are blank for clarity. Posterior means $\widehat{\pi}_g$ and modal estimates $\widehat{q}_g$, with associated 95% credible intervals, are also given.

	0	1	2	3	4	5	6	7	8	9	$\widehat{\pi}_g$	$\widehat{q}_g$
1	359										0.05	4 [2, 8]
2	58		12			3	2				0.01	3 [2, 7]
3	108										0.01	2 [1, 4]
4	9										0.00	16 [3, 16]
5	95										0.01	4 [1, 8]
6	308					3					0.04	7 [4, 10]
7		844			2						0.12	2 [0, 4]
8		133							1		0.02	1 [0, 4]
9		2	392	10		1					0.05	7 [5, 12]
10	59		121	93	19	91	13	2	25	4	0.06	12 [9, 16]
11				136		64					0.03	5 [2, 9]
12					38	1		1			0.01	2 [0, 8]
13	25		3	7	98	51	2	36	59	28	0.04	8 [5, 12]
14	48		73	61	62	135	32	1	16	6	0.06	8 [6, 12]
15	1						83				0.01	3 [1, 7]
16	1						74				0.01	2 [1, 5]
17		2			4	19	381		2		0.06	2 [1, 6]
18								207			0.03	4 [1, 8]
19	123	8	129	348	247	184	77	26	420	84	0.23	6 [3, 9]
20		16	1	3	120	1		338	19	451	0.13	2 [1, 6]
21					62	3		34		71	0.02	3 [1, 6]

Generally, IMIFA assigns images of the same digit, albeit written differently, to different clusters. Posterior mean images for each cluster are shown in Figure 4.9, ordered, as is Table 4.4, from 0 to 9 according to the digit most frequently assigned to the related cluster. Cluster 7 and the smaller cluster 8 capture the digit 1 written

in a straight and slanted fashion, respectively. Clusters 15, 16, and 17 represent the digit 6 written with extended, medium, and compact loop curvature, respectively. Notably, cluster 15 requires more factors than clusters 16 and 17. A similar interpretation follows for clusters 20 and 21 ($\widehat{q}_{20} = 2, \widehat{q}_{21} = 3$), mostly capturing the digit 9 with a small and large loop, respectively. Cluster 19 appears to mostly represent the digit 8 and has a large number of factors ($\widehat{q}_{19} = 6$) in comparison, say, to clusters 7 and 8 ($\widehat{q}_7 = 2, \widehat{q}_8 = 1$) which capture the digit 1. This is intuitive, as 8 is a more geometrically complex digit than 1. However, some clusters appear to be diluted by the confusion of the so-called 'closed' 4, in contrast to the 'open' 4 in cluster 12, with the digits 3, 5, and 8 (cluster 19) and the digits 7 (written with a horizontal bar) and 9 (clusters 20 and 21). Many clusters capture the most common digit 0, with differing degrees of elongation and border thickness. Of concern here is cluster 4, containing just 9 observations; the fact that $\widehat{q}_4 = 16$, the upper AGS limit, suggests that the model struggles to shrink the number of factors in poorly populated clusters. This difficulty is highlighted further in the simulation studies in Appendix 4.B. Finally, Table 4.4 indicates that clusters 10, 13, and 14 also capture several other digits, all of which are reflected in the blurriness of the resulting posterior mean images and in $\widehat{q}_{10}$, $\widehat{q}_{13}$, and $\widehat{q}_{14}$ being quite large. The cluster-membership uncertainties are visualised in Appendix 4.D.

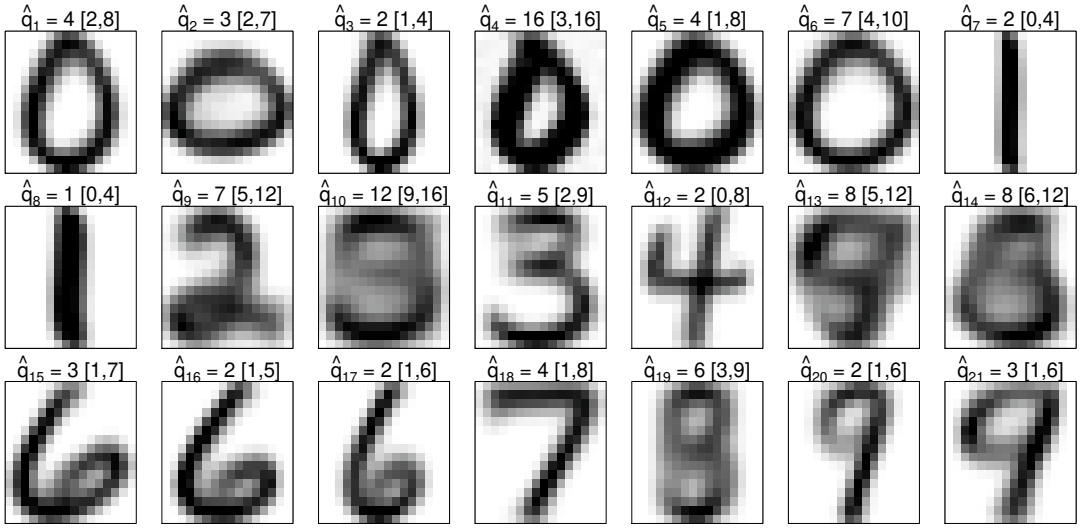


Figure 4.9: Posterior mean images for clusters uncovered by fitting IMIFA to the USPS data. Plots are ordered according to Table 4.4 and labelled with the modal $\widehat{q}_g$.

It is computationally infeasible to run `mclust`, `pgmm`, or MFMA on these large data, as an exhaustive model search would be too vast. For comparative purposes, a DP-BP model (Chen et al., 2010) is fitted; this approach also simultaneously assumes infinitely many components and factors. It finds 43 clusters, each with around 14 factors, and achieves an ARI of 0.32. Cross tabulating this clustering against the 21 clusters of the IMIFA model shows that some of the DP-BP clusters are encapsulated by the larger IMIFA clusters. IMIFA is thus the more parsimonious approach and affords greater cluster-specific factor flexibility. Additionally, a finite mixture of matrix-normal distributions (Viroli, 2011) is also fitted. This approach accounts for the grid nature of the data, but is computationally infeasible for $G > 15$ and requires a model selection strategy. The optimal model according to BIC yields $\widehat{G} = 12$ and ARI $= 0.38$. While neither IMIFA nor the DP-BP model account for the spatial structure in the data, they demonstrate comparative performance without the need for a computationally expensive model search.

4.4 Discussion

The proposed IMIFA model is a Bayesian nonparametric approach clustering high-dimensional data using factor-analytic mixture models. By extending the MGP prior (Bhattacharya and Dunson, 2011) to the PYP-MGP setting, the model sidesteps the fraught and computationally intensive task of determining the optimal number of clusters and factors using model selection criteria. Thus, the IMIFA model is recommended when fitting factor-analytic mixtures in settings where an exhaustive model search is computationally infeasible. Though IMIFA is not entirely choice-free, it achieves improved clustering results by allowing factor-analytic models of different dimensions in different clusters. If small clusters are inferred, one may wish to prune or merge small clusters with the larger clusters (West et al., 1994) or assess whether the small clusters are in fact of domain-specific interest. While comparative performance can be achieved by the IMIFA and OMIFA models, one may wish to fit a MIFA or OMIFA model when the expectation is that the number of clusters is fixed or unlikely to grow with N, respectively.

Future research directions are varied and plentiful. Incorporating covariates, in the spirit of Bayesian factor regression models (West, 2003; Carvalho et al., 2008), would allow for direct inclusion of the weight and urine pH covariates available with the metabolomic data, for example. Furthermore, the models could be extended to the (semi-)supervised model-based classification setting where all (or some) of the data are labelled. While constraints on the uniquenesses across variables and/or clusters are allowed, there is scope for also constraining the loadings across clusters. Though the number of factors would no longer be cluster-specific, the common number of loadings columns would be estimated in a similarly automatic fashion. However, incorporating covariance matrix constraints in the IMIFA model family problematically reintroduces the need for model selection strategies, in order to choose between them, though the BICM criterion could feasibly be used for this purpose also.

As proposed by Bhattacharya and Dunson (2011), the MGP hyperparameters could be learned via Metropolis-Hastings, and thus also be made cluster-specific. This could help combat some difficulties identified in the simulation studies in Appendix 4.B. For example, learning those related to local shrinkage may help when loadings are notably dense. Learning those related to column shrinkage may help in settings with many small clusters, where IMIFA struggles to adaptively truncate loadings columns. In principle, a further global shrinkage parameter ϖ could be added to the MGP prior to borrow information across clusters, i.e. $\lambda_{jkg} \mid \ldots \sim \mathrm{N}_1\big(0, \phi_{jkg}^{-1}\tau_{kg}^{-1}\sigma_g^{-1}\varpi^{-1}\big)$. Alternatively, the infinite factor prior of Legramanti et al. (2019) could be employed, which decouples control over the shrinkage rate and the active loadings terms. Finally, the IMIFA family can in fact be considered as wider than the range of models presented here. For example, the IBP prior (Knowles and Ghahramani, 2007, 2011; Ročková and George, 2016) could be extended to the infinite mixture setting, as per the DP-BP model of Chen et al. (2010).

For applied problems, a mismatch between the assumed model and the data distribution will impact inference. Miller and Harrison (2013, 2014) highlight that posterior consistency for the number of non-empty clusters in infinite mixtures is contingent on correct specification of the component distributions. While they do not discourage the use of infinite mixtures for clustering, they show that a few tiny extra clusters are typically fitted and suggest robustifying inference. If the data dis-

tribution is close to but not exactly a finite mixture of Gaussians, an infinite Gaussian mixture will introduce more components as the amount of data increases. Potential avenues of exploration thus include considering the IMIFA model with the heavy tailed multivariate t-distribution (Peel and McLachlan, 2000). Similarly, modelling of complex component distributions can be achieved by considering the MFMA approach in the context of infinite factor models. Defining robust inference functions as in Lee and MacEachern (2014) or using nonparametric unimodal component distributions as in Rodriguez and Walker (2014) may also prove fruitful. Another means of robustifying inference is to explicitly include a noise component with zero factors to capture outliers which depart from the component multivariate normality assumption. Finally, a 'coarsened' posterior (Miller and Dunson, 2018) could be used for addressing misspecification, by conditioning on the event that the model generates data close to the observed data in a distributional sense.

References

Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics 2*(6), 1152–1174. 224

Baek, J., G. J. McLachlan, and L. K. Flack (2010). Mixtures of factor analyzers with common factor loadings: applications to the clustering and visualization of high-dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*(7), 1298–1309. 169

Bai, J. and K. Li (2012). Statistical analysis of factor models of high dimension. *The Annals of Statistics 40*(1), 436–465. 169

Bhattacharya, A. and D. B. Dunson (2011). Sparse Bayesian infinite factor models. *Biometrika 98*(2), 291–306. 170, 173, 174, 194, 195, 211, 217

Brooks, S. P. and A. Gelman (1998). Generative methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics 7*(4), 434–455. 182

Carmody, S. and L. Brennan (2010). Effects of pentylenetetrazole-induced seizures on metabolomic profiles of rat brain. *Neurochemistry International 56*(2), 340–344. 188

Carmona, C., L. Nieto-barajas, and A. Canale (2019). Model based approach for household clustering with mixed scale variables. *Advances in Data Analysis and Classification 13*(2), 559–583. 183, 209

Carpaneto, G. and P. Toth (1980). Solution of the assignment problem. *ACM Transactions on Mathematical Software 6*(1), 104–111. 173

Carvalho, C. M., J. Chang, J. E. Lucas, J. R. Nevins, Q. Wang, and M. West (2008). High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association 103*(484), 1438–1456. 195

Chen, M., J. Silva, J. Paisley, C. Wang, D. B. Dunson, and L. Carin (2010). Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: algorithm and performance bounds. *IEEE Transactions on Signal Processing 58*(12), 6140–6155. 177, 194, 195

De Blasi, P., S. Favaro, A. Lijoi, R. H. Mena, I. Prünster, and M. Ruggiero (2015). Are Gibbs-type priors the most natural generalization of the Dirichlet process? *IEEE Transactions on Pattern Analysis and Machine Intelligence 37*(2), 212–229. 179, 224

Diebolt, J. and C. P. Robert (1994). Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 56*(2), 363–375. 169

Durante, D. (2017). A note on the multiplicative gamma process. *Statistics & Probability Letters 122*, 198–204. 170, 174

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics 1*(2), 209–230. 170

Fokoué, E. and D. M. Titterington (2003). Mixtures of factor analysers. Bayesian estimation and inference by stochastic simulation. *Machine Learning 50*(1), 73–94. 169

Forina, M., C. Armanino, S. Lanteri, and E. Tiscornia (1983). Classification of olive oils from their fatty acid composition. In H. Martens and H. Russrum, Jr. (Eds.), *Food Research and Data Analysis*, pp. 189–214. Applied Science Publishers, London. 185

Forina, M. and E. Tiscornia (1982). Pattern recognition methods in the prediction of Italian olive oil by their fatty acid content. *Annali di Chimica 72*, 143–155. 185

Frühwirth-Schnatter, S. (2010). *Finite Mixture and Markov Switching Models*. Series in Statistics. New York: Springer. 173

Frühwirth-Schnatter, S. (2011). Dealing with label switching under model uncertainty. In K. L. Mengersen, C. P. Robert, and D. M. Titterington (Eds.), *Mixtures: Estimation and Applications*, Wiley Series in Probability and Statistics, pp. 193–218. Chichester: John Wiley & Sons. 173

Frühwirth-Schnatter, S. and H. F. Lopes (2010). Parsimonious Bayesian factor analysis when the number of factors is unknown. Technical report, The University of Chicago Booth School of Business. 172

Frühwirth-Schnatter, S. and H. F. Lopes (2018). Sparse bayesian factor analysis when the number of factors is unknown. *arXiv* pre-print, 1804.04231. 173

Frühwirth-Schnatter, S. and G. Malsiner-Walli (2019). From here to infinity: sparse finite versus Dirichlet process mixtures in model-based clustering. *Advances in Data Analysis and Classification 13*(1), 33–63. 183, 210

Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin (2004). *Bayesian Data Analysis* (Third ed.). Chapman and Hall/CRC Press. 181

Ghahramani, Z. and G. E. Hinton (1996). The EM algorithm for mixtures of factor analyzers. Technical report, Department of Computer Science, University of Toronto. 169

Ghosh, J. and D. B. Dunson (2008). Default prior distributions and efficient posterior computation in Bayesian factor analysis. *Journal of Computational and Graphical Statistics 18*(2), 306–320. 173

Green, P. J. and S. Richardson (2001). Modelling heterogeneity with and without the Dirichlet process. *Scandinavian Journal of Statistics 28*(2), 355–375. 183

Hastie, D. I., S. Liverani, and S. Richardson (2014). Sampling from Dirichlet process mixture models with unknown concentration parameter: mixing issues in large data implementations. *Statistics and Computing 25*(5), 1023–1037. 180

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning* (Second ed.). Springer Series in Statistics. New York: Springer. 191

Hubert, L. and P. Arabie (1985). Comparing partitions. *Journal of Classification 2*(1), 193–218. 185

Jara, M., E. Lesaffre, M. De Iorio, and F. Quintana (2010). Bayesian semiparametric inference for multivariate doubly-interval-censored data. *The Annals of Applied Statistics 4*(4), 2126–2149. 209

Kalli, M., J. E. Griffin, and S. G. Walker (2011). Slice sampling mixture models. *Statistics and Computing 21*(1), 93–105. 170, 179

Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association 90*(430), 773–795. 169

Kim, S., M. G. Tadesse, and M. Vannucci (2006). Variable selection in clustering via Dirichlet process mixture models. *Biometrika 93*(4), 877–893. 182

Knott, M. and D. J. Bartholomew (1999). *Latent Variable Models and Factor Analysis* (Second ed.). Number 7 in Kendall's library of statistics. London: Edward Arnold. 169

Knowles, D. and Z. Ghahramani (2007). Infinite sparse factor analysis and infinite independent components analysis. In M. E. Davies, C. J. James, S. A. Abdallah, and M. D. Plumbley (Eds.), *Independent Component Analysis and Signal Separation*, pp. 381–388. Berlin, Heidelberg: Springer. 177, 195

Knowles, D. and Z. Ghahramani (2011). Nonparametric bayesian sparse factor models with application to gene expression modeling. *The Annals of Applied Statistics 5*(2B), 1534–1552. 177, 195

Lee, J. and S. N. MacEachern (2014). Inference functions in high dimensional Bayesian inference. *Statistics and Its Interface 7*(4), 477–486. 196

Legramanti, S., D. Durante, and D. B. Dunson (2019). Bayesian cumulative shrinkage for infinite factorizations. *arXiv* pre-print, 1902.04349. 195

Malsiner-Walli, G., S. Frühwirth-Schnatter, and B. Grün (2016). Model-based clustering based on sparse finite Gaussian mixtures. *Statistics and Computing 26*(1), 303–324. 210

McLachlan, G. J. and D. Peel (2000). *Finite Mixture Models*. Wiley Series in Probability and Statistics. New York: John Wiley & Sons. 169

McNicholas, P. D. (2010). Model-based classification using latent Gaussian mixture models. *Journal of Statistical Planning and Inference 140*(5), 1175–1181. 185

McNicholas, P. D., A. ElSherbiny, A. F. McDaid, and T. B. Murphy (2018). `pgmm`*: parsimonious Gaussian mixture models*. R package version 1.2.3. 185

McNicholas, P. D. and T. B. Murphy (2008). Parsimonious Gaussian mixture models. *Statistics and Computing 18*(3), 285–296. 169, 172, 208

McParland, D., I. C. Gormley, T. H. McCormick, S. J. Clark, C. W. Kabudula, and M. A. Collinson (2014). Clustering South African households based on their asset status using latent variable models. *The Annals of Applied Statistics 8*(2), 747–767. 173

Miller, J. W. and D. B. Dunson (2018). Robust Bayesian inference via coarsening. *Journal of the American Statistical Association 114*(527), 1113–1125. 196

Miller, J. W. and M. T. Harrison (2013). A simple example of Dirichlet process mixture inconsistency for the number of components. *Advances in Neural Information Processing Systems 26*, 199–206. 182, 195

Miller, J. W. and M. T. Harrison (2014). Inconsistency of Pitman-Yor process mixtures for the number of components. *The Journal of Machine Learning Research 15*(1), 3333–3370. 183, 195

Müller, P. and R. Mitra (2013). Bayesian nonparametric inference – why and how. *Bayesian Analysis 8*(2), 269–360. 183, 209

Murphy, K., C. Viroli, and I. C. Gormley (2019). `IMIFA`*: infinite mixtures of infinite factor analysers and related models*. R package version 2.1.2. 171, 206, 221

Ng, A. Y., M. I. Jordan, and Y. Weiss (2001). On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems*, Cambridge, MA, USA, pp. 849–856. MIT Press. 191

Nyamundanda, G., L. Brennan, and I. C. Gormley (2010). Probabilistic principle component analysis for metabolomic data. *BMC Bioinformatics 11*(571), 1–11. 188

Paisley, J. and L. Carin (2009). Nonparametric factor analysis with Beta process priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, New York, NY, USA, pp. 777–784. ACM. 177

Papaspiliopoulos, O. and G. O. Roberts (2008). Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika 95*(1), 169–186. 181, 210

Papastamoulis, P. (2018). Overfitting Bayesian mixtures of factor analyzers with an unknown number of components. *Computational Statistics & Data Analysis 124*, 220–234. 170, 177

Peel, D. and G. J. McLachlan (2000). Robust mixture modelling using the t distribution. *Statistics and Computing 10*(4), 339–348. 196

Perman, M., J. Pitman, and M. Yor (1992). Size-biased sampling of Poisson point processes and excursions. *Probability Theory and Related Fields 92*(1), 21–39. 170

Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields 102*(2), 145–158. 209

Pitman, J. (1996). Random discrete distributions invariant under size-biased permutation. *Advances in Applied Probability 28*(2), 525–539. 170, 178, 224

Pitman, J. and M. Yor (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability 25*(2), 855–900. 170

Plummer, M., N. Best, K. Cowles, and K. Vines (2006). CODA: convergence diagnosis and output analysis for MCMC. *R News 6*(1), 7–11. 182

Raftery, A. E., M. Newton, J. Satagopan, and P. Krivitsky (2007). Estimating the integrated likelihood via posterior simulation using the harmonic mean identity. In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West (Eds.), *Bayesian Statistics 8: Proceedings of the Eighth Valencia International Meeting, June 2–6, 2006*, Oxford; New York, pp. 1–45. Oxford University Press. 176

R Core Team (2019). *R: a language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. 171

Richardson, S. and P. J. Green (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 59*(4), 731–792. 169, 182

Ročková, V. and E. I. George (2016). Fast Bayesian factor analysis via automatic rotations to sparsity. *Journal of the American Statistical Association 111*(516), 1608–1622. 177, 195

Rodriguez, C. E. and S. G. Walker (2014). Univariate Bayesian nonparametric mixture modeling with unimodal kernels. *Statistics and Computing 24*(1), 35–49. 196

Rousseau, J. and K. Mengersen (2011). Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(5), 689–710. 177

Rue, H. and L. Held (2005). *Gaussian Markov Random Fields: Theory and Applications*, Volume 104 of *Monographs on statistics and applied probability*. London: Chapman and Hall/CRC Press. 180

Scrucca, L., M. Fop, T. B. Murphy, and A. E. Raftery (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal 8*(1), 289–317. 180, 185, 212

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica 4*(2), 639–650. 224

Spiegelhalter, D. J., N. G. Best, B. P. Carlin, and A. Van Der Linde (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64*(4), 583–639. 169

Spiegelhalter, D. J., N. G. Best, B. P. Carlin, and A. Van Der Linde (2014). The deviance information criterion: 12 years on. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76*(3), 485–493. 169

Stephens, M. (2000). Bayesian analysis of mixture models with an unknown number of components - an alternative to reversible jump methods. *The Annals of Statistics 28*(1), 40–74. 182

Tipping, M. E. and C. M. Bishop (1999). Mixtures of probabilistic principal component analyzers. *Neural Computation 11*(2), 443–482. 173, 207

van den Berg, R. A., H. C. Hoefsloot, J. A. Westerhuis, A. K. Smilde, and M. J. van der Werf (2006). Centering, scaling, and transformations: improving the biological information content of metabolomics data. *BMC Genomics 7*(1), 142. 189

van Havre, Z., N. White, J. Rousseau, and K. Mengersen (2015). Overfitting Bayesian mixture models with an unknown number of components. *PloS one 10*(7), e0131739. 177

Viroli, C. (2010). Dimensionally reduced model-based clustering through mixtures of factor mixture analyzers. *Journal of classification 27*(3), 363–388. 169, 185

Viroli, C. (2011). Finite mixtures of matrix normal distributions for classifying three-way data. *Statistics and Computing 21*(4), 511–522. 194

Walker, S. G. (2007). Sampling the Dirichlet mixture model with slices. *Communications in Statistics - Simulation and Computation 36*(1), 45–54. 179

Wang, C., G. Pan, T. Tong, and Z. L (2015). Shrinkage estimation of large dimensional precision matrix using random matrix theory. *Statistica Sinica 25*(3), 993–1008. 173

Wang, Y., A. Canale, and D. B. Dunson (2016). Scalable geometric density estimation. In A. Gretton and C. P. Robert (Eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, Volume 51 of *Proceedings of Machine Learning Research*, Cadiz, Spain, pp. 857–865. PMLR. 177

West, M. (1992). Hyperparameter estimation in Dirichlet process mixture models. Technical report, ISDS Discussion Paper 92–A03, Duke University. 210, 211

West, M. (2003). Bayesian factor regression models in the “large p, small n" paradigm. In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West (Eds.), *Bayesian Statistics 7: Proceedings of the 7th Valencia International Meeting, June 2–7, 2002*, Oxford; New York, pp. 723–732. Oxford University Press. 195

West, M., P. Müller, and M. D. Escobar (1994). Hierarchical priors and mixture models, with applications in regression and density estimation. In A. F. M. Smith and P. R. Freeman (Eds.), *Aspects of Uncertainty: A Tribute to D. V. Lindley*, pp. 363–386. New York: John Wiley & Sons. 194

Xing, E. P., K. A. Sohn, M. I. Jordan, and Y. W. Teh (2006). Bayesian multipopulation haplotype inference via a hierarchical Dirichlet process mixture. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 1049–1056. ACM. 182

Yellott, Jr., J. I. (1977). The relationship between Luce’s choice axiom, Thurstone’s theory of comparative judgment, and the double exponential distribution. *Journal of Mathematical Psychology 15*(2), 109–144. 181, 208

Yerebakan, H. Z., B. Rajwa, and M. Dundar (2014). The infinite mixture of infinite Gaussian mixtures. In *Advances in Neural Information Processing Systems*, pp. 28–36. 182

4.A Appendix 1

Posterior Conditional Distributions: Technical details for sampling from the IMIFA model

The structure of the Metropolis-within-Gibbs sampler to conduct inference for the IMIFA model and the exact forms of the required conditional distributions are detailed below. Note that $\mathrm{Ga}(\alpha, \beta)$ refers throughout to the gamma distribution with mean α/β. The number of observations in a component is denoted by n_g, where $\mathbf{n} = \left(n_1, \ldots, n_{\widetilde{G}}\right)^\top$ sums to N, and $\widetilde{q}_g$ is the current sample of the cluster-specific number of active factors. Algorithms for sampling other models in the IMIFA family can all be considered as special cases of what follows. The algorithm is implemented in the associated R package `IMIFA` (Murphy et al., 2019). For $g = 1, \ldots, \widetilde{G}$, where $\widetilde{G}$ is the current number of active components (of which some may be empty):

$$\boldsymbol{\mu}_g \,|\, \ldots \sim \mathrm{N}_p\left(\boldsymbol{\Omega}_{\boldsymbol{\mu}}^{-1}\left(\boldsymbol{\Psi}_g^{-1}\Big(\sum_{i:\, z_{ig}=1} \mathbf{x}_i - \sum_{i:\, z_{ig}=1} \boldsymbol{\Lambda}_g \boldsymbol{\eta}_i\Big) + \varphi \boldsymbol{\mathcal{I}}_p \widetilde{\boldsymbol{\mu}}\right), \boldsymbol{\Omega}_{\boldsymbol{\mu}}^{-1}\right),$$

$$\boldsymbol{\eta}_i \,|\, z_{ig} = 1, \ldots \sim \mathrm{N}_{\widetilde{q}_g}\left(\boldsymbol{\Omega}_{\boldsymbol{\eta}}^{-1}\boldsymbol{\Lambda}_g^\top \boldsymbol{\Psi}_g^{-1}\left(\mathbf{x}_{i:\, z_{ig}=1} - \boldsymbol{\mu}_g\right), \boldsymbol{\Omega}_{\boldsymbol{\eta}}\right) \quad \text{for } i = 1, \ldots, n_g\,,$$

$$\psi_{jg} \,|\, \ldots \sim \mathrm{IG}\left(\alpha_0 + \frac{n_g}{2}, \beta_j + \frac{\mathcal{S}_{jg}}{2}\right) \quad \text{for } j = 1, \ldots, p,$$

$$\boldsymbol{\Lambda}_{jg} \,|\, \ldots \sim \mathrm{N}_{\widetilde{q}_g}\left(\boldsymbol{\Omega}_{\boldsymbol{\lambda}}^{-1}\boldsymbol{\eta}_{i:\, z_{ig}=1}^\top \psi_{jg}^{-1}\left(\mathbf{x}_{i:\, z_{ig}=1}^{(j)} - \mu_{jg}\right), \boldsymbol{\Omega}_{\boldsymbol{\lambda}}^{-1}\right) \quad \text{for } j = 1, \ldots, p,$$

$$\phi_{jkg} \,|\, \ldots \sim \mathrm{Ga}\left(\nu_1 + \frac{1}{2}, \nu_2 + \frac{\sigma_g \tau_{kg} \lambda_{jkg}^2}{2}\right) \quad \text{for } j = 1, \ldots, p \text{ and } k = 1, \ldots, \widetilde{q}_g\,,$$

$$\delta_{1g} \,|\, \ldots \sim \mathrm{Ga}\left(\alpha_1 + \frac{p\widetilde{q}_g}{2}, \beta_1 + \frac{\sigma_g}{2}\sum_{h=1}^{\widetilde{q}_g} \tau_{hg}^{(1)} \sum_{j=1}^{p} \phi_{jhg}\lambda_{jhg}^2\right),$$

$$\delta_{kg} \,|\, \ldots \sim \mathrm{Ga}\left(\alpha_2 + \frac{p}{2}\left(\widetilde{q}_g - k + 1\right), \beta_2 + \frac{\sigma_g}{2}\sum_{h=k}^{q_g} \tau_{hg}^{(k)} \sum_{j=1}^{p} \phi_{jhg}\lambda_{jhg}^2\right) \quad \text{for } k = 2, \ldots, \widetilde{q}_g\,,$$

$$\sigma_g \,|\, \ldots \sim \mathrm{Ga}\left(\varrho_1 + \frac{p\widetilde{q}_g}{2}, \varrho_2 + \frac{\sum_{k=1}^{\widetilde{q}_g} \tau_{kg} \sum_{j=1}^{p} \phi_{jkg} \lambda_{jkg}^2}{2}\right),$$

$$\upsilon_g \,|\, \ldots \sim \mathrm{Beta}\Big(1 - d + n_g, \alpha + gd + N - \sum_{l=1}^{g} n_l\Big),$$

$$u_i \,|\, z_{ig} = 1, \ldots \sim \mathrm{Unif}(0, \xi_g) \quad \text{for } i = 1, \ldots, N,$$

where

$$\boldsymbol{\Omega}_{\boldsymbol{\mu}} = \varphi \boldsymbol{\mathcal{I}}_p + n_g \boldsymbol{\Psi}_g^{-1},$$

$$\boldsymbol{\Omega}_{\boldsymbol{\eta}} = \boldsymbol{\mathcal{I}}_{\widetilde{q}_g} + \boldsymbol{\Lambda}_g^{\top} \boldsymbol{\Psi}_g^{-1} \boldsymbol{\Lambda}_g \,,$$

$$\boldsymbol{\Omega}_{\boldsymbol{\lambda}} = \mathrm{diag}\big(\phi_{j1g}\tau_{1g}\sigma_g, \ldots, \phi_{j\widetilde{q}_g g}\tau_{\widetilde{q}_g g}\sigma_g\big) + \psi_{jg}^{-1} \boldsymbol{\eta}_{i:\, z_{ig}=1}^{\top} \boldsymbol{\eta}_{i:\, z_{ig}=1} \,,$$

$$\tau_{hg}^{(k)} = \prod_{t=1}^{h} \frac{\delta_{tg}}{\delta_{kg}} \,,$$

$$\pi_g = \upsilon_g \prod_{l=1}^{g-1} (1 - \upsilon_l) \,,$$

and

$$\mathcal{S}_{jg} = \sum_{i:\, z_{ig}=1} \big(x_{ij} - \mu_{jg} - \boldsymbol{\Lambda}_{jg} \boldsymbol{\eta}_i\big)^{\top} \big(x_{ij} - \mu_{jg} - \boldsymbol{\Lambda}_{jg} \boldsymbol{\eta}_i\big).$$

Here $\mathbf{x}^{(j)}$ denotes the j-th column of the data matrix, λ_{jkg}^2 denotes a single squared loading, and $\tau_{kg} = \prod_{h=1}^{k} \delta_{hg}$ is updated after every update of δ_{hg}.

Parsimonious parameterisations of the component covariance matrices are easily incorporated. Uniquenesses can be constrained to be isotropic, with $\boldsymbol{\Psi}_g = \mathrm{diag}(\psi_g, \ldots, \psi_g)$, leading to a model that corresponds to an infinite mixture and infinite-dimensional extension of probabilistic principal components analysers (Tipping and Bishop, 1999). Uniquenesses can also be constrained across clusters, with or without the isotropic constraint across variables. These restrictions de-

fine the models in the pgmm family (McNicholas and Murphy, 2008) named UUC, UCU, and UCC, respectively, to which the following Gibbs updates are Bayesian analogues

$$\psi_g \,|\, \ldots \sim \mathrm{IG}\left(\alpha_0 + \frac{pn_g}{2}, \beta + \frac{\mathrm{tr}\left(\boldsymbol{\mathcal{S}}_g\right)}{2}\right),$$

$$\psi_j \,|\, \ldots \sim \mathrm{IG}\left(\alpha_0 + \frac{N}{2}, \beta_j + \frac{\sum_{g=1}^{G} \mathcal{S}_{jg}}{2}\right),$$

$$\psi \,|\, \ldots \sim \mathrm{IG}\left(\alpha_0 + \frac{pN}{2}, \beta + \frac{\sum_{g=1}^{G} \mathrm{tr}\left(\boldsymbol{\mathcal{S}}_g\right)}{2}\right).$$

In the contexts of finite and overfitted mixtures (i.e. MFA, MIFA, OMFA, and OMIFA) $\mathbf{z}_i \,|\, \mathbf{x}_i, \ldots \sim \mathrm{Mult}\left(1, p_{i1}, \ldots, p_{i\widetilde{G}}\right)$, with

$$p_{ig} = \Pr(z_{ig} = 1 \,|\, \mathbf{x}_i, \ldots) = \frac{\pi_g \mathrm{N}_p\left(\mathbf{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g^\top + \boldsymbol{\Psi}_g\right)}{\sum_{g=1}^{\widetilde{G}} \pi_g \mathrm{N}_p\left(\mathbf{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g^\top + \boldsymbol{\Psi}_g\right)},$$

whereas under the IMIFA and IMFA models

$$p_{ig} \propto \mathrm{N}_p\left(\mathbf{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g^\top + \boldsymbol{\Psi}_g\right) \frac{\pi_g}{\xi_g} \mathbb{1}\left(u_i < \xi_g\right). \tag{4.3}$$

The allocations $\mathbf{z}_i$ are sampled in a fast, numerically stable fashion, using the unnormalised log-probabilities and independent draws from the standard Gumbel distribution (Yellott, 1977) via $s_{ig} = -\ln(m_{ig})$, with $m_{ig} \sim \mathrm{Exp}(\lambda = 1)$. Observation i is assigned the label g satisfying

$$\underset{g \in \{1,\ldots,\widetilde{G}\}}{\arg\max} \left(\ln(p_{ig}) + s_{ig}\right).$$

For the IMIFA and IMFA models, the sampler need only find the maximum over, and only draw Gumbel noise for, log-probabilities for which the indicator function in (4.3) evaluates to 1.

Sampling the parameters of the PYP for non-zero d values requires the introduction of Metropolis-Hastings steps within the Gibbs sampler. A joint hyperprior of the form $\mathrm{p}(\alpha, d) = \mathrm{p}(d)\,\mathrm{p}(\alpha \mid d)$ is assumed, as per Jara et al. (2010). Firstly, the hyperprior for the discount parameter d is similar to the one assumed by Carmona et al. (2019); a mixture of a point-mass at zero and a continuous beta distribution, in order to consider the DP special case with $d = 0$ with positive probability, i.e. $d \sim \kappa\delta_0 + (1-\kappa)\,\mathrm{Beta}(a', b')$. This facilitates explicit comparison between DP models and encompassing PYP alternatives. Secondly, the hyperprior for α is given conditionally on d, s.t. $(\alpha \mid d) \sim \mathrm{Ga}(\alpha + d; a, b)$, and includes the constraint $\alpha > -d$ by shifting the support of the gamma density to the interval $(-d, \infty)$; choosing a large b value is particularly relevant as it encourages clustering (Müller and Mitra, 2013).

The likelihood for α and d is given by the exchangeable partition probability function induced by the PYP (Pitman, 1995). Thus, the required conditional posterior distributions are

$$\alpha \mid d, \ldots \propto \frac{\Gamma(\alpha+1)}{\Gamma(\alpha+N)} \left\{ \prod_{g=1}^{G_0-1} (\alpha + gd) \right\} p(\alpha \mid d), \tag{4.4}$$

$$d \mid \alpha, \ldots \propto \left\{ \prod_{g=1}^{G_0-1} (\alpha + gd) \right\} \left\{ \prod_{g=1}^{G_0} \frac{\Gamma(n_g - \alpha)}{\Gamma(1-\alpha)} \right\} p(d). \tag{4.5}$$

Sampling from the distributions in (4.4) and (4.5), while always considering the support $\alpha > -d$, proceeds as per Carmona et al. (2019); a Metropolis-Hastings step is implemented for the discount parameter with independent proposal distribution $0.5\delta_0 + 0.5\mathrm{Beta}(d; 1, 1)$, and a random walk Metropolis-Hastings step with proposal distribution given by $\alpha^\star \mid \alpha \sim \mathrm{Unif}(\alpha - \zeta, \alpha + \zeta)$ is implemented for the concentration parameter, where ζ (= 2 in our implementation) is used to control the acceptance rate. For d, the mutation rate is considered rather than the acceptance rate, whereby a move is only considered accepted if the proposal differs from the current value.

However, when the DP prior is assumed, or when the sampled value of d is exactly zero under the PYP prior, α is updated according to the auxiliary variable routine of West (1992), with Gibbs updates by simulation from a weighted mixture of two gamma distributions, via

$$\alpha \,|\, G_0, \chi, \ldots \sim \omega_\chi \mathrm{Ga}\big(a + G_0, b - \ln(\chi)\big) + (1 - \omega_\chi)\,\mathrm{Ga}\big(a + G_0 - 1, b - \ln(\chi)\big),$$

where G_0 denotes the current number of non-empty clusters,

$$(\chi \,|\, \alpha, G_0) \sim \mathrm{Beta}(\alpha + 1, N)\,,$$

and the mixing weights ω_χ are defined by

$$\frac{\omega_\chi}{1 - \omega_\chi} = \frac{(a + G_0 - 1)}{N\big(b - \ln(\chi)\big)}.$$

The complementary label switching moves of Papaspiliopoulos and Roberts (2008), which are effective at swapping similar and unequal clusters, respectively, are also incorporated. Firstly, the labels of two randomly chosen non-empty clusters g and h are swapped with probability

$$\min\left(1, \left(\pi_h/\pi_g\right)^{n_g - n_h}\right).$$

Secondly, the labels of neighbouring active components l and $l+1$ are swapped with probability

$$\min\left(1, \left(1 - v_{l+1}\right)^{n_l} / \left(1 - v_l\right)^{n_{l+1}}\right);$$

if accepted, v_l and v_{l+1} are also swapped. Cluster-specific parameters are re-ordered accordingly after each accepted move. Finally, for updating α under the sparse finite OMIFA or OMFA models, a random walk Metropolis-Hastings step is implemented, with a Gaussian proposal distribution, where

$$\alpha \,|\, \mathbf{Z}, \widetilde{G}, \ldots \propto \frac{\Gamma\left(\alpha \widetilde{G}\right)}{\Gamma\left(N + \alpha \widetilde{G}\right)} \left\{ \prod_{g\,:\,n_g > 0} \frac{\Gamma\left(n_g + \alpha\right)}{\Gamma\left(\alpha\right)} \right\} p\left(\alpha\right).$$

Further details of this update can be found in Malsiner-Walli et al. (2016) and Frühwirth-Schnatter and Malsiner-Walli (2019).

4.B Appendix 2

Simulation Studies

The performance of the novel IMIFA model with its PYP-MGP priors, in terms of inferring both the number of clusters and the cluster-specific numbers of factors, is assessed here through simulation studies. Section 4.B explores sensitivity to the PYP parameters in a range of dimensionality scenarios, with balanced cluster sizes and a common number of factors. The simulation study in Section 4.B is more challenging; a larger number of clusters (many of which are small) are simulated for $N < p$ data, with different numbers of cluster-specific factors (some of which are large). The final simulation study in Section 4.B mirrors the design in Section 4.B, only here the true $\boldsymbol{\Lambda}_g$ matrices used to generate the data are sparse.

Simulation Study 1

Firstly, data with $G = 3$ clusters and $p = 50$ variables are simulated with $q_g = 4 \ \forall \ g$, and with $\boldsymbol{\pi} = (1/3, 1/3, 1/3)$ so that clusters are roughly equally sized. Other model parameters are simulated, with $\boldsymbol{\eta}_i \sim \mathrm{N}_q(\mathbf{0}, \mathcal{I}_q)$, $\psi_{jg} \sim \mathrm{IG}(2, 1)$, and $\boldsymbol{\Lambda}_{jg} \sim \mathrm{N}_q(\mathbf{0}, \mathcal{I}_q)$. Notably, loadings are not drawn from the MGP prior (Bhattacharya and Dunson, 2011) underpinning the IMIFA model. To ensure clusters are reasonably closely located, $\boldsymbol{\mu}_g \sim \mathrm{N}_p((2g - G - 1)\,\mathbf{1}, \mathcal{I}_p)$. The data are then simulated according to the conditional mixture model

$$f\left(\mathbf{x}_i \,|\, \boldsymbol{\eta}_i, \boldsymbol{\theta}\right) = \sum_{g=1}^{G} \pi_g \mathrm{N}_p\left(\mathbf{x}_i; \boldsymbol{\mu}_g + \boldsymbol{\Lambda}_g \boldsymbol{\eta}_i, \boldsymbol{\Psi}_g\right).$$

To evaluate performance in different settings, sample sizes less than, equal to, and greater than p are considered, i.e. $N = 25, 50$, and 300. Sensitivity to the PYP and DP parameters is explored by firstly assuming a DP prior with various values of α less than, equal to, and greater than 1, and by allowing α to be learned as per West (1992), and secondly by incorporating Metropolis-Hastings steps to learn both α and d, assuming a PYP prior.

Results, provided in Table 4.B.1, are based on 10 replicate data sets, standardised prior to model fitting, for each scenario. MCMC chains were run for $25,000$ iterations, with every 2^{nd} sample thinned and the first 20% of iterations discarded as burn-in. Cluster labels were initialised using `mclust` (Scrucca et al., 2016), as hierarchical clustering gave poor, heavily imbalanced starting values. As the cluster-specific $\mathbf{\Lambda}_g$ and $\mathbf{\Psi}_g$ parameters could still induce separation among clusters, pairwise scatterplots from one randomly chosen raw replicate data set under the $N > p$ scenario are shown in Figure 4.B.1 to demonstrate the extent of the overlap; for visual clarity, only 5 randomly chosen variables are depicted.

Table 4.B.1: Aggregated simulation study results for the IMIFA model under different dimensionality scenarios and settings of the concentration and discount parameters α and d (posterior mean estimates thereof in parentheses where appropriate). The modal estimates of G and associated estimates of $q_g \ \forall \ g$ are reported (with 95% credible intervals in brackets). Clustering performance is assessed through the average percentage error rate against the known cluster labels.

Dimension	α	d	G	q_1	q_2	q_3	Error (%)
$N = 25$ ($N < p$)	0.5	0	3 [3, 3]	5 [3, 9]	5 [3, 9]	5 [3, 9]	0
	1	0	3 [3, 3]	5 [3, 9]	5 [3, 9]	5 [3, 9]	0
	5	0	3 [3, 4]	5 [3, 9]	5 [3, 9]	5 [3, 9]	6.4
	(0.57)	0	3 [3, 3]	5 [3, 9]	5 [3, 9]	5 [3, 9]	0
	(0.51)	(0.05)	3 [3, 3]	5 [3, 9]	5 [3, 9]	5 [3, 9]	0
$N = 50$ ($N = p$)	0.5	0	3 [3, 3]	5 [4, 7]	5 [4, 7]	5 [4, 7]	0
	1	0	3 [3, 3]	5 [4, 7]	5 [4, 7]	5 [4, 7]	0
	5	0	3 [3, 3]	5 [4, 7]	5 [4, 7]	5 [4, 7]	0
	(0.52)	0	3 [3, 3]	5 [4, 7]	5 [4, 7]	5 [4, 7]	0
	(0.48)	(0.03)	3 [3, 3]	5 [4, 7]	5 [4, 7]	5 [4, 7]	0
$N = 300$ ($N > p$)	0.5	0	3 [3, 3]	5 [4, 6]	5 [4, 6]	5 [4, 6]	0
	1	0	3 [3, 3]	5 [4, 6]	5 [4, 6]	5 [4, 6]	0
	5	0	3 [3, 3]	5 [4, 6]	5 [4, 6]	5 [4, 6]	0
	(0.42)	0	3 [3, 3]	5 [4, 6]	5 [4, 6]	5 [4, 6]	0
	(0.39)	(0.02)	3 [3, 3]	5 [4, 6]	5 [4, 6]	5 [4, 6]	0

Table 4.B.1 clearly demonstrates that the IMIFA model performs well overall for these data, exhibiting capability to uncover the structure within the simulated data sets regardless of dimensionality. The modal estimate of G is equal to the truth in all cases, with only the $N < p$, $\alpha = 5$ scenario showing some deviation in the 95% credible interval. Perhaps surprisingly, given the closeness of the cluster means, and the equality of the clusters in terms of their mixing proportions and numbers of factors, G is never underestimated. Indeed, clustering performance is mostly

perfect. Furthermore, in every case, the true value of $q_g = 4$ is within the limits of the associated credible intervals, which intuitively become narrower as more data accumulates. While the modal estimates $\widehat{q}_g$ are consistently greater than the truth throughout Table 4.B.1, overestimation should be preferred to underestimation; a less parsimonious model which nevertheless fits well and uncovers the true clustering structure is better than one which loses information and fits poorly due to having too few factors. Recall that the loadings were drawn from a standard multivariate Gaussian, rather than the MGP prior underpinning the IMIFA model, i.e. entries in the true $\mathbf{\Lambda}_g$ matrices did not shrink with the column index, nor were the loadings sparse. Thus, there is evidence to suggest the model is liable to overestimate the number of factors when the $\mathbf{\Lambda}_g$ matrices, and by extension the cluster-specific marginal covariance matrices, are dense. This is explored further in the subsequent simulation studies.

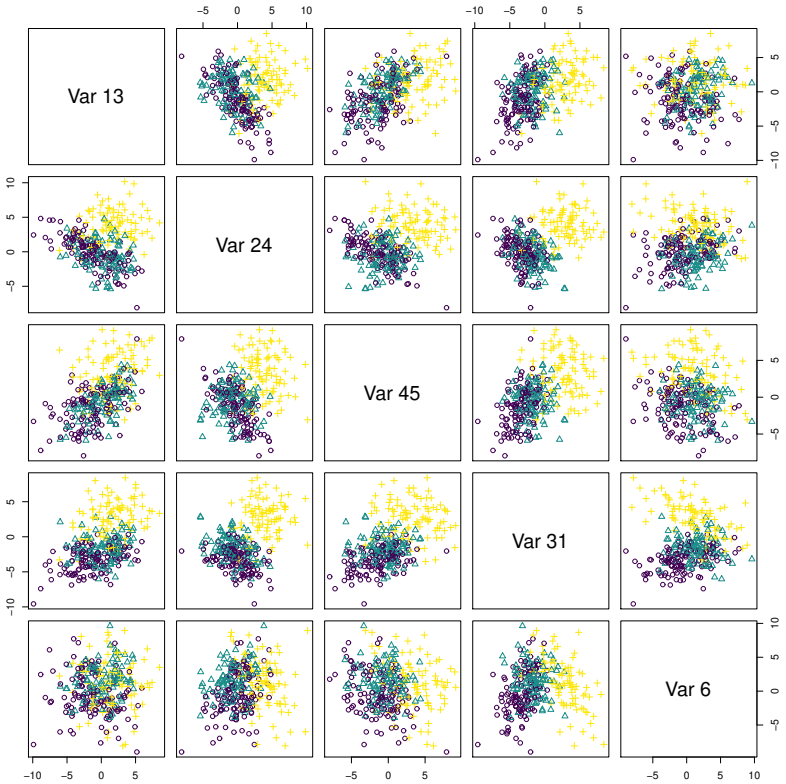


Figure 4.B.1: Pairwise scatterplots of 5 randomly chosen variables from one of the raw replicate data sets under the $N > p$ scenario in Table 4.B.1, demonstrating the overlap between the 3 clusters.

Simulation Study 2

Results of a more challenging simulation study are presented in Figure 4.B.3; here, $N < p$ data ($N = 200, p = 250$) are simulated with a large number of clusters and uncommon numbers of cluster-specific factors. In particular, many of the $G = 10$ clusters are small (a setting often studied in Bayesian nonparametric modelling), with $\boldsymbol{\pi} = (0.25, 0.2, 0.15, 0.1, 0.05, \ldots, 0.05)^{\top}$. The numbers of factors $q_1, \ldots, q_g$ are drawn randomly from $0, \ldots, \min(15, n_g - 1)$, where the upper limit ensures that no cluster has more factors than observations. Otherwise, the same parameter settings as Simulation Study 1 above (Section 4.B) were used to generate the data.

Results of fitting an IMIFA model assuming a PYP prior, allowing both α and d to be learned, and otherwise using the same sampler settings as in Section 4.B above, are given for 5 replicates of this scenario, with the $\boldsymbol{\pi}$ vector ordered randomly for each data set. To demonstrate the extent of the challenge these settings represent, pairwise scatterplots are again shown for 5 randomly chosen variables for the first replicate data set in Figure 4.B.2.

Figure 4.B.3 shows that the model over-estimates the number of clusters, though in some cases the ARI values are nonetheless quite good, as the larger clusters are generally uncovered well. However, the smaller clusters are further divided, albeit cleanly, into smaller sub-clusters with, in some cases, just 1 or 2 units inside. In these cases, the modal $\widehat{q}_g$ estimates are close or equal to the upper limit of the adaptive Gibbs sampler ($3\ln(p)$), and hence or otherwise greater than the corresponding estimated cluster sizes $\widehat{n}_g$. Thus, there is evidence that the model has difficulty in adaptively shrinking the $\boldsymbol{\Lambda}_g$ matrices when there are many clusters with few units.

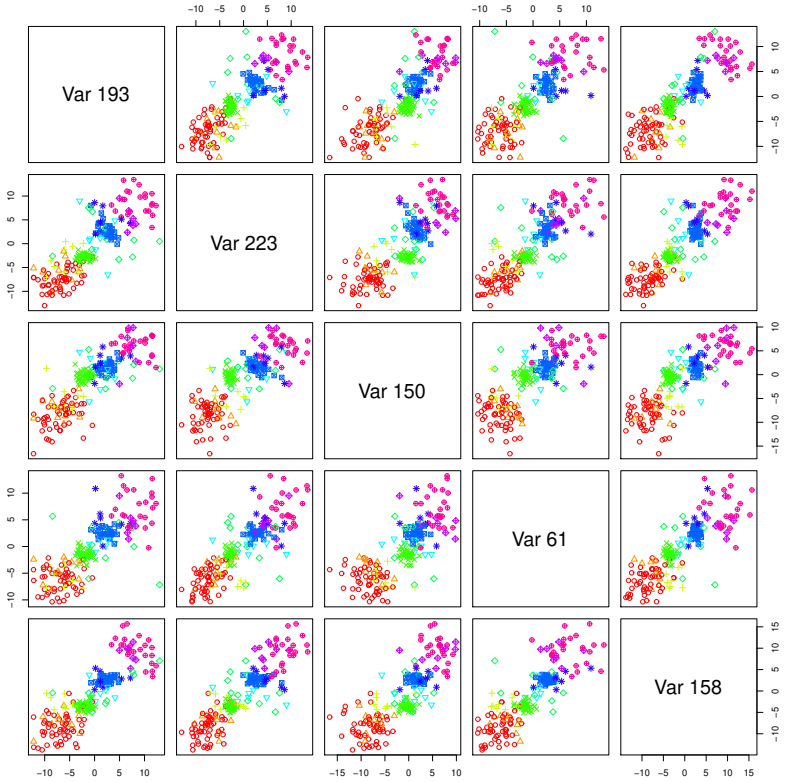
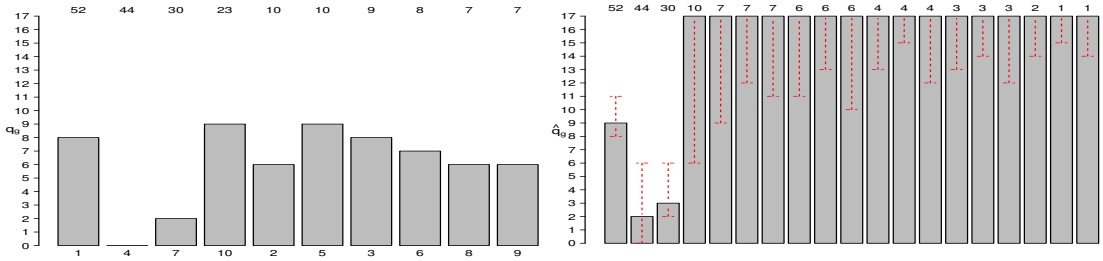
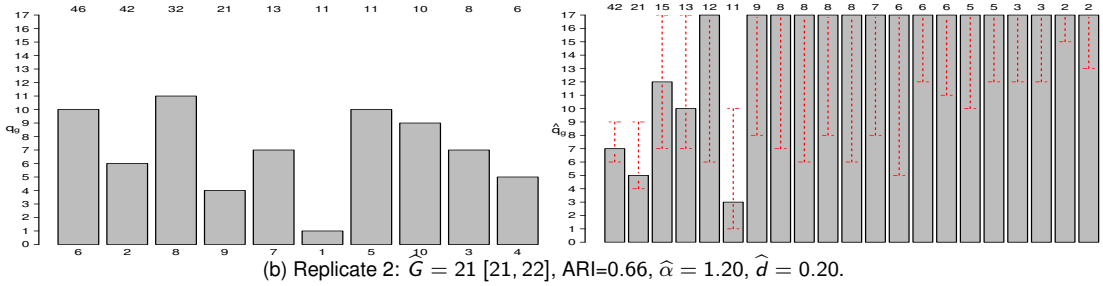


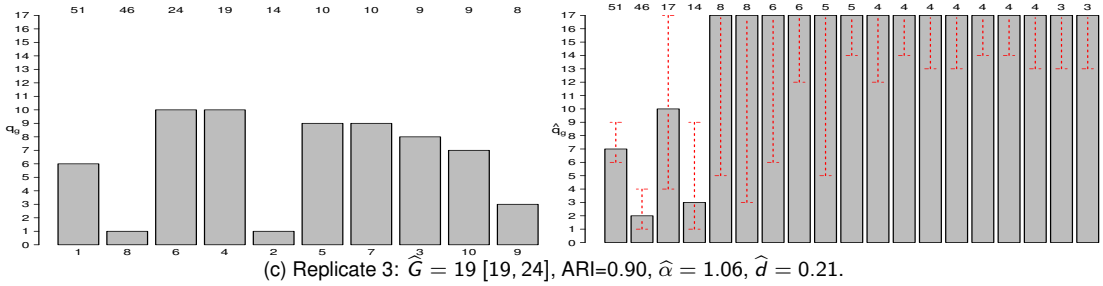
Figure 4.B.2: Pairwise scatterplots of 5 randomly chosen variables from the first raw replicate data set in Simulation Study 2 (Section 4.B), demonstrating the overlap between the 10 clusters.



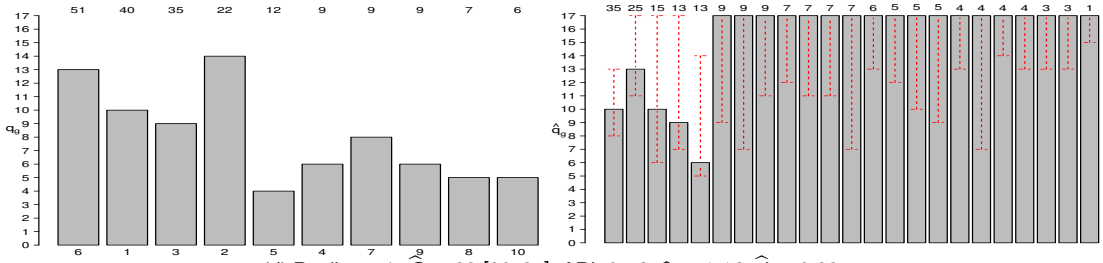
(a) Replicate 1: $\widehat{G} = 19$ [19, 19], ARI=0.94, $\widehat{\alpha} = 0.80$, $\widehat{d} = 0.26$.



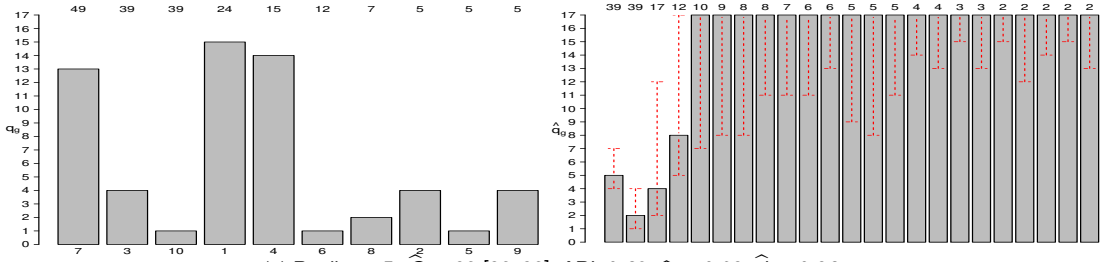
(b) Replicate 2: $\widehat{G} = 21$ [21, 22], ARI=0.66, $\widehat{\alpha} = 1.20$, $\widehat{d} = 0.20$.



(c) Replicate 3: $\widehat{G} = 19$ [19, 24], ARI=0.90, $\widehat{\alpha} = 1.06$, $\widehat{d} = 0.21$.



(d) Replicate 4: $\widehat{G} = 23$ [20, 25], ARI=0.56, $\widehat{\alpha} = 1.16$, $\widehat{d} = 0.23$.



(e) Replicate 5: $\widehat{G} = 23$ [23, 23], ARI=0.68, $\widehat{\alpha} = 0.92$, $\widehat{d} = 0.26$.

Figure 4.B.3: Barplots of the true number of cluster-specific factors q_g (left) and estimates $\widehat{q}_g$ (right) for each replicate data set and corresponding fitted IMIFA model comprising Simulation Study 2. Bars are sorted in descending order of n_g and $\widehat{n}_g$, respectively, and labelled above with these true and estimated cluster sizes. The plots on the left are also labelled below with the cluster indices. Vertical red lines in the plots on the right show 95% credible intervals for $\widehat{q}_g$. Modal $\widehat{G}$ estimates (with 95% credible intervals in brackets), ARI values, and posterior mean estimates $\widehat{\alpha}$ and $\widehat{d}$ are given for each replicate.

Simulation Study 3

In both previous simulation studies, the true loadings were dense, having been drawn from a standard multivariate Gaussian, rather than the MGP prior underpinning the model. The design of this final simulation study exactly mirrors the parameter and sampler settings used in Section 4.B with the sole exception that, as per the simulation study design in Bhattacharya and Dunson (2011), the true loadings matrices used to generate the data are sparse.

Specifically, the number of non-zero loadings in each $\mathbf{\Lambda}_g$ matrix begins at p in column 1, and successively decays by 10% for each subsequent column. The locations of the zeros in each column are allocated randomly and non-zero elements are drawn from a standard multivariate Gaussian. Again, pairwise scatterplots are shown for 5 randomly chosen variables for the first of the five replicate data sets in Figure 4.B.4, to demonstrate the extent of the overlap between clusters.

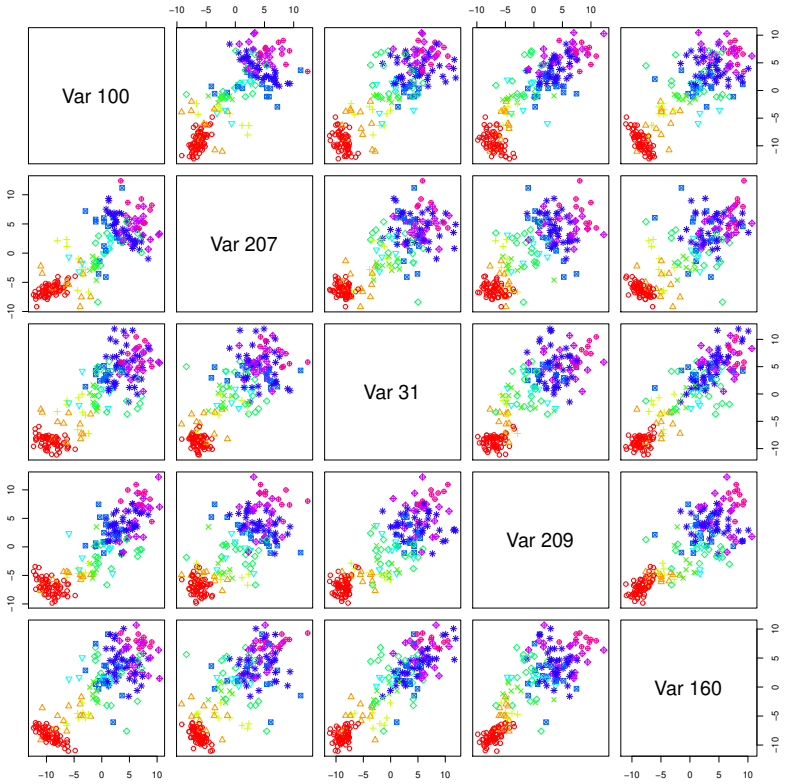
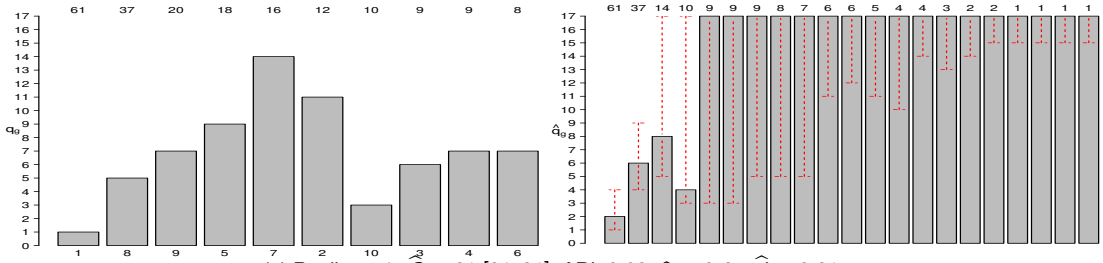


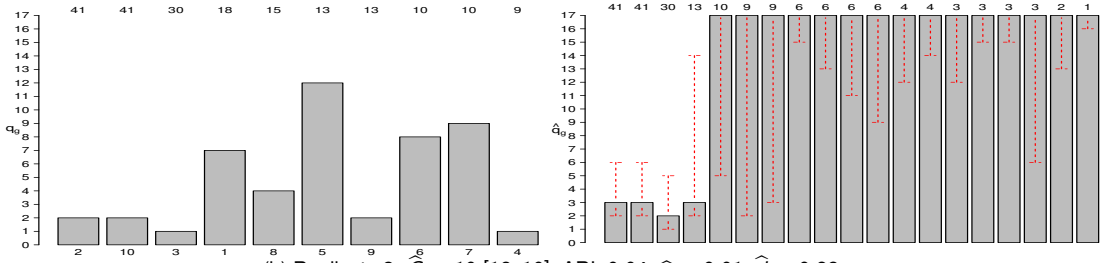
Figure 4.B.4: Pairwise scatterplots of 5 randomly chosen variables from the first raw replicate data set in Simulation Study 3 (Section 4.B), demonstrating the overlap between the 10 clusters.

Results are presented in Figure 4.B.5. Performance is comparable to the results of Simulation Study 2, in the sense that, again, the number of clusters is over-estimated. ARI values are nonetheless acceptable. Small clusters are divided into even smaller sub-clusters for which the model struggles to adaptively shrink the number of factors. The comparability of the results of these experiments suggests that performance is being driven not by whether the loadings used to generate the data exhibit increasing levels of sparsity across columns, in line with the MGP prior underpinning the model, but by the presence of many small clusters.

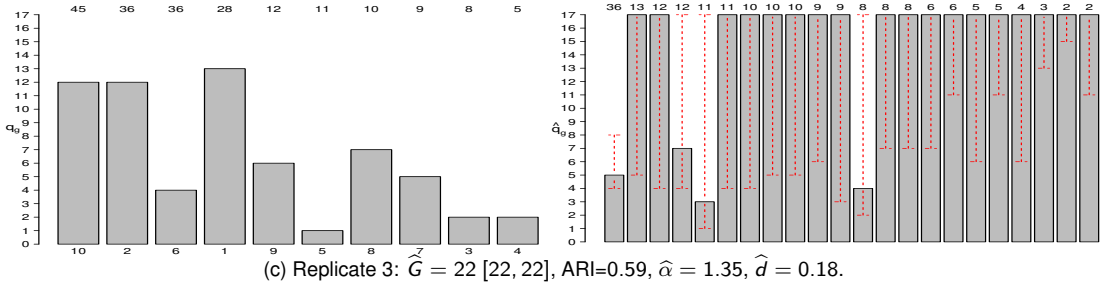
The over-estimation of $\widehat{q}_g$ in the small clusters in simulation studies 2 and 3 suggests that the hyperparameters α_1 and α_2 related to the MGP column shrinkage parameters may need to be higher in mixture settings to enforce a greater degree of shrinkage as there will be fewer data in each cluster from which local and global shrinkage parameters can be learned, compared to fitting an IFA model on the full data set. Introducing Metropolis-Hastings steps to allow these hyperparameters be cluster-specific and learned from the data, rather than fixed, may also help in this regard.



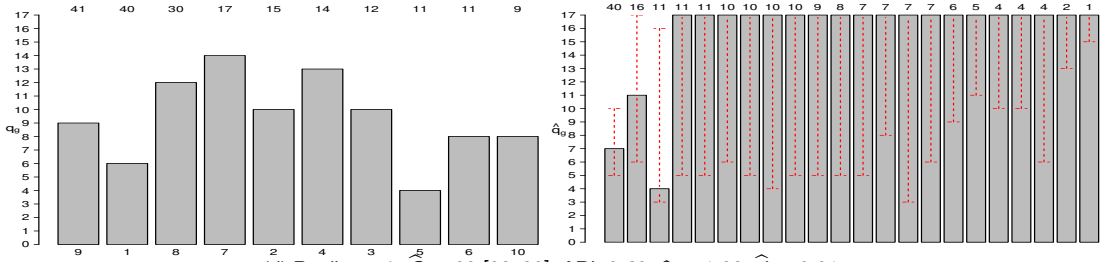
(a) Replicate 1: $\widehat{G} = 21$ [21, 21], ARI=0.93, $\widehat{\alpha} = 0.65$, $\widehat{d} = 0.31$.



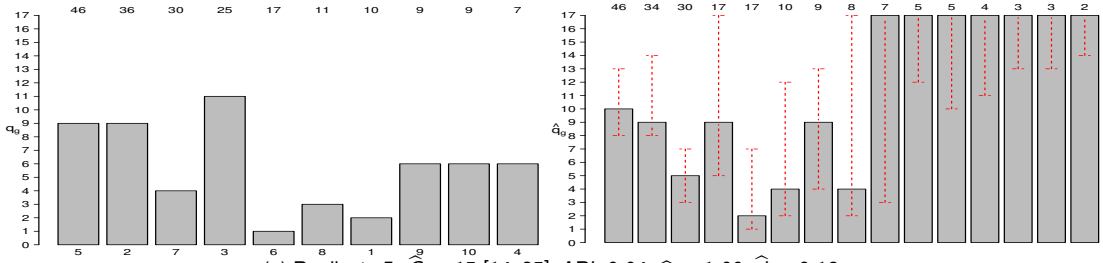
(b) Replicate 2: $\widehat{G} = 19$ [18, 19], ARI=0.94, $\widehat{\alpha} = 0.91$, $\widehat{d} = 0.23$.



(c) Replicate 3: $\widehat{G} = 22$ [22, 22], ARI=0.59, $\widehat{\alpha} = 1.35$, $\widehat{d} = 0.18$.



(d) Replicate 4: $\widehat{G} = 22$ [22, 22], ARI=0.69, $\widehat{\alpha} = 1.22$, $\widehat{d} = 0.21$.



(e) Replicate 5: $\widehat{G} = 15$ [14, 25], ARI=0.94, $\widehat{\alpha} = 1.09$, $\widehat{d} = 0.18$.

Figure 4.B.5: Barplots of the true number of cluster-specific factors q_g (left) and estimates $\widehat{q}_g$ (right) for each replicate data set and corresponding fitted IMIFA model comprising Simulation Study 3. Bars are sorted in descending order of n_g and $\widehat{n}_g$, respectively, and labelled above with these true and estimated cluster sizes. The plots on the left are also labelled below with the cluster indices. Vertical red lines in the plots on the right show 95% credible intervals for $\widehat{q}_g$. Modal $\widehat{G}$ estimates (with 95% credible intervals in brackets), ARI values, and posterior mean estimates $\widehat{\alpha}$ and $\widehat{d}$ are given for each replicate.

4.C Appendix 3

Assessing Robustness of the IMIFA Model

In order to assess the robustness of the IMIFA model, $N(0, 1)$ noise with no clustering information was appended separately to the rows and columns of the olive oil data set. Six new scenarios were generated with 10, 50, and 100 extra variables, and the same numbers of extra observations. Cluster validity is evaluated in Table 4.C.1 with respect to the 4 area relabelling in Table 4.3b. In the case of extra observations, noise observations are labelled as though they belong to a fifth cluster. Data were mean-centred and unit-scaled only after expansion.

As the number of irrelevant variables increases, the clustering structure can still be uncovered quite well, however mixing becomes slower and there is increasing support for clusters with only one or no factors as the signal-to-noise ratio decreases. As such, variable selection, or at least data pre-processing, may still be required. As rows of noise are appended, IMIFA generally has no difficulty in assigning these observations to a cluster of their own. Interestingly, clusters corresponding to noise observations correctly require no latent factor structure.

Table 4.C.1: Clustering performance of the IMIFA model on expanded noisy versions of the Italian olive oil data. The run-time relative to running IMIFA on the original data, posterior mean of the PYP parameters α and d, modal estimates of G and $\mathbf{Q}$, ARI, and percentage error rate are all given.

Scenario	Relative Time	α	d	G	$\mathbf{Q}$	ARI	Error (%)
$N = 572, p = 18$	1.86	0.48	0.01	4	3, 4, 4, 3	0.85	12.59
$N = 572, p = 58$	3.14	0.47	0.01	4	1, 2, 2, 2	0.74	14.69
$N = 572, p = 108$	5.64	0.46	0.02	4	0, 1, 0, 2	0.73	17.66
$N = 582, p = 8$	1.10	0.57	0.01	5	6, 2, 2, 2, 0	0.94	6.87
$N = 622, p = 8$	1.09	0.56	0.01	5	4, 1, 1, 2, 0	0.95	6.59
$N = 672, p = 8$	1.07	0.53	0.01	5	4, 1, 2, 2, 0	1.00	0.45

4.D Appendix 4

Additional Results and Visualisations

In this Section, some additional visualisations of the results of the illustrative applications are provided. Specifically, more details are provided on the posterior predictive model fit assessment and the observation-specific cluster membership uncertainties. All plots were produced using the associated R package `IMIFA` (Murphy et al., 2019).

The Posterior Predictive Reconstruction Error (PPRE) has been proposed as a posterior predictive checking strategy for models in the IMIFA family. In short, this involves computing the standardised Frobenius norm of the difference between a matrix of histogram bin counts for the modelled data set and similar matrices constructed using replicate data drawn from the posterior predictive distribution. While the median PPRE value or boxplots of the distribution of PPRE values have been shown to yield useful global measures of model fit in multivariate settings, the histograms themselves can also be studied on a variable-by-variable basis.

In high-dimensional settings, such as the spectral metabolomic ($p = 189$) and USPS digits ($p = 256$) data sets, it is only feasible to examine the histograms for a subset of the variables. Nonetheless, the global median PPRE measures for these data sets are quite good (0.21 and 0.05, respectively). Hence, Figure 4.D.1 shows only the histograms comparing bin counts for the $p = 8$ variables in the standardised Italian olive oil data, to which an IMIFA model was fitted, against corresponding counts for the replicate data under the fitted IMIFA model. The true bin counts are within the 95% credible intervals of the replicate data bin counts in the vast majority of cases, indicating good model fit: recall that this IMIFA model achieves a median PPRE of just 0.10.

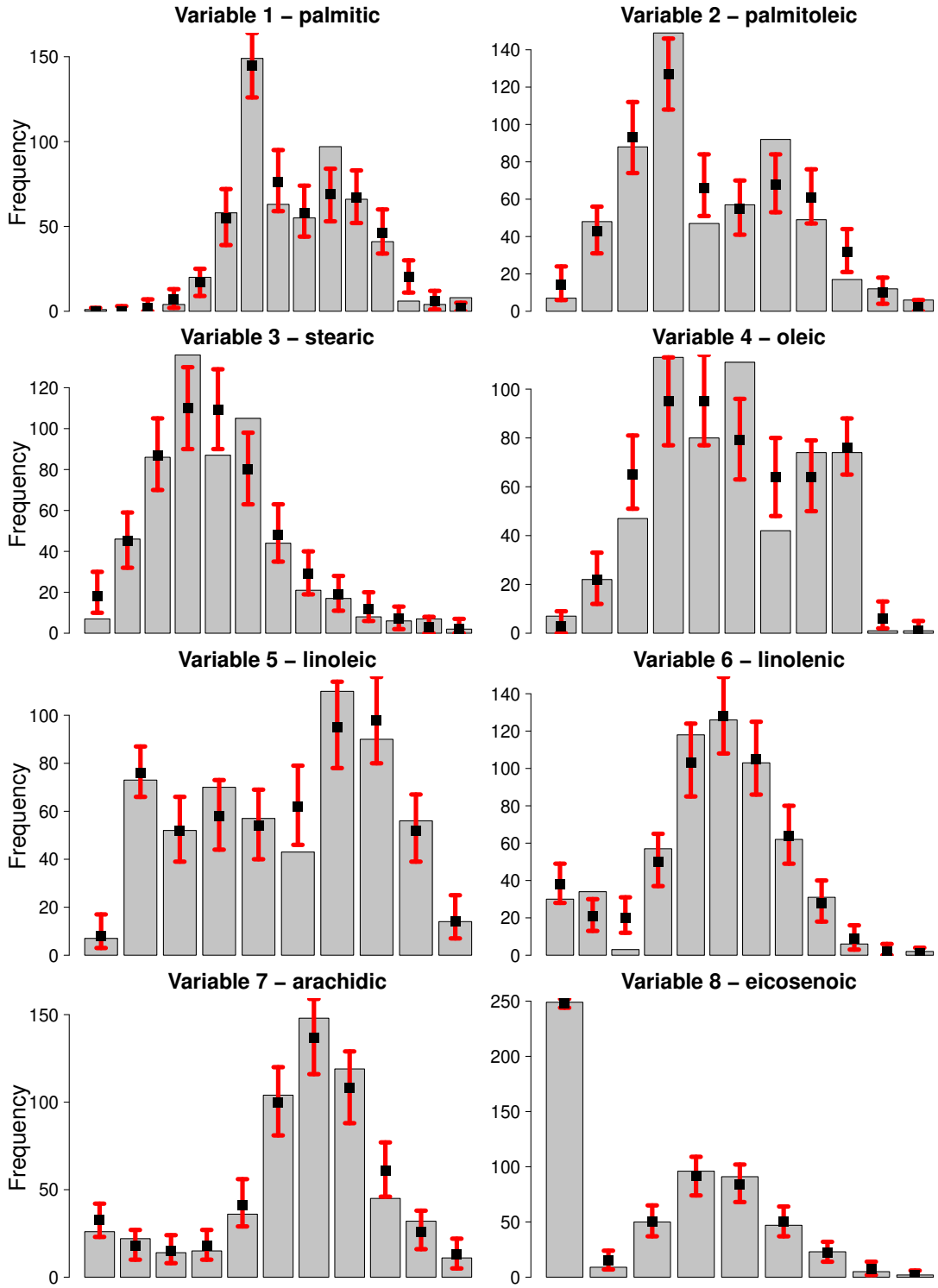


Figure 4.D.1: Histograms of the $p = 8$ variables in the standardised Italian olive oil data set. The height of each bar corresponds to the modelled data set, while the black squares correspond to the median bin counts of the replicate data sets drawn from the posterior predictive distribution of the fitted IMIFA model (with associated 95% credible intervals given by vertical red lines).

The IMIFA model fitted to the USPS digits data set uncovers $\widehat{G} = 21$ clusters. Regarding the uncertainty in the allocations to these clusters, the model-based nature of IMIFA facilitates estimation of the uncertainty with which observation i is assigned to its cluster g via

$$\widehat{U}_i = \min_{g \in \left\{1,\ldots,\widehat{G}\right\}} \left(1 - \widehat{z}_{ig}\right),$$

where $\widehat{z}_{ig}$ is the estimated probability that observation i belongs to cluster g. Figure 4.D.2 shows that the observation-specific cluster membership uncertainties are generally quite low, with the mean uncertainty being just 0.02 and 92% of observations being assigned with uncertainty less than $1/\widehat{G}$. A similar plot for the olive oil data is shown in the main text (Figure 4.3); uncertainties for the spectral metabolomic data are not shown, as there was no uncertainty in the assignments under the fitted IMIFA model (i.e. $\widehat{U}_i = 0 \ \forall \ i = 1, \ldots, N$).

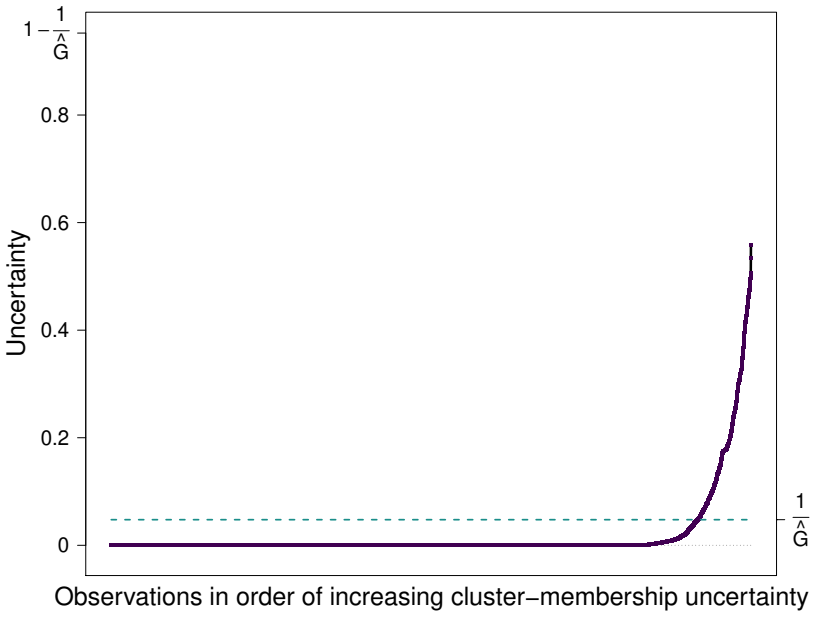


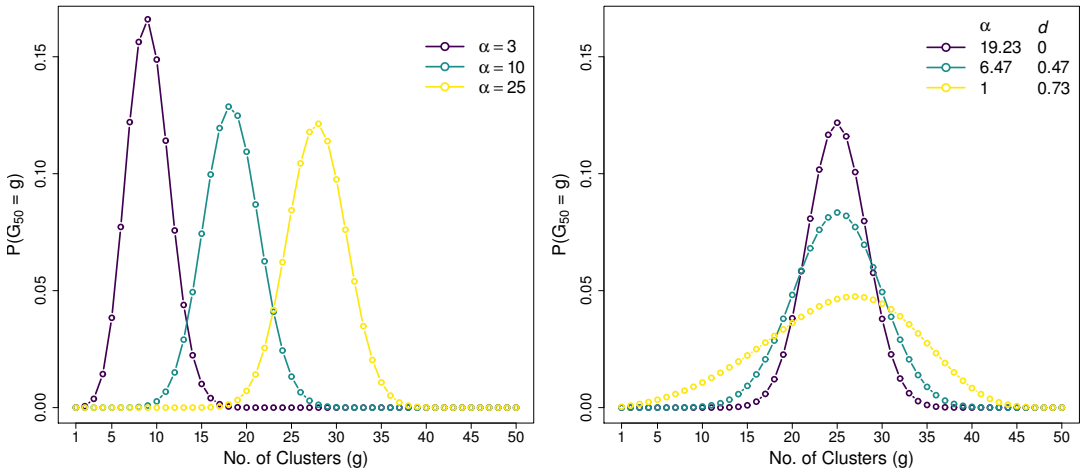
Figure 4.D.2: Uncertainty profile plot for the 21-cluster IMIFA model fitted to the USPS digits data, showing observation-specific uncertainties in increasing order, most of which are below the line at $1/\widehat{G}$.

4.E Appendix 5

Comparing the PYP and DP Priors

The PYP and DP priors differ in their construction of the stick-breaking proportions, with $v_g \sim \text{Beta}(1-d, \alpha+gd)$ under the PYP prior (Pitman, 1996) and $v_g \sim \text{Beta}(1, \alpha)$ under the DP prior (Sethuraman, 1994). While the DP can be seen as a special case of the PYP with $d = 0$ (indeed, the PYP is often referred to as the two-parameter Poisson-Dirichlet process), some important distributional features differ for non-zero d values. Notably, the growth rate of $\mathbb{E}(G)$ is logarithmic in N under a DP prior (Antoniak, 1974), while it is Zipfian under a PYP prior.

Figure 4.E.1, adapted from De Blasi et al. (2015), illustrates the utility of the extra parameter in the IMIFA setting. In particular, note that the DP with $d = 0$ is also included as a comparator in Figure 4.E.1b and exhibits a highly peaked distribution. In terms of prior specification, this implies the need for reliable prior information on the number of clusters, which is often unavailable, as the high-peakedness prevents the wrong prior information from being overruled. As stated, the joint hyperprior assumed on α and d in the IMIFA setting achieves further flexibility, by allowing the PYP parameters to be learned from the data rather than fixed.



(a) Dirichlet process prior

(b) Pitman-Yor process prior

Figure 4.E.1: DP and PYP priors when $N = 50$, under different concentration and discount parameter settings. Under the DP prior, mass shifts to the right with increasing dispersion as α increases. Under the PYP prior, with parameters fixed so that $\mathbb{E}(G_{50}) = 25$, a heavier-tailed, less informative prior is obtained as d increases.

4.F Appendix 6

IMIFA R Package Vignette

This appendix presents a reproduction of the package vignette[7] of the associated R package `IMIFA` for implementation of the proposed method. Notably, some additional results summaries and diagnostic plots are presented for the IMIFA model fitted to the Italian olive oil.

[7] cran.r-project.org/web/packages/IMIFA/vignettes/IMIFA.html

IMIFA: Infinite Mixtures of Infinite Factor Analysers and Related Models

Keefe Murphy

Introduction

IMIFA is an R package that provides flexible and efficient functions for fitting Infinite Mixtures of Infinite Factor Analysers (**IMIFA**) and related models. The main model, IMIFA itself, conducts Bayesian nonparametric clustering via latent Gaussian models. While the package offers a Bayesian implementation of the Factor Analysis (FA) and Mixtures of Factor Analysers (MFA) models, among others, these require pre-specification of the number of latent factors &/or the number of components, which must remain fixed, the main advantages of the IMIFA model are that a) the model search is dramatically reduced, b) these quantities are estimated automatically, c) the number of latent factors is allowed to be cluster-specific, and d) uncertainty in the number of clusters and numbers of cluster-specific factors can be quantified.

Typically, one would run FA or MFA models over ranges of values for the numbers of clusters and factors, with the pair which optimises some model selection criterion typically chosen. **IMIFA** instead enables Bayesian nonparametric model-based clustering with factor analytic covariance structures, *without* recourse to model selection criteria, to automatically choose the number of clusters &/or cluster-specific latent factors.

The main features of the IMIFA model are the multiplicative gamma process shrinkage prior on the factor loadings, which allows theoretically infinitely many factors (this can also be employed in a MIFA context, for instance, where the number of clusters is fixed but cluster-specific factors are estimated), an adaptive Gibbs sampler which dynamically truncates the infinite loadings matrices, and the Dirichlet process prior, which utilises the stick-breaking construction and slice-efficient sampling, and allows theoretically infinitely many clusters. Tools are provided for soliciting sensible hyperparameters for these priors. As of IMIFA v1.2.0, a Pitman-Yor process prior on the number of mixture components is assumed by default, and its concentration and discount parameters are learned via Metropolis-Hastings updates.

Model-specific diagnostic tools are also provided, as well as many extensive options for plotting results, conducting posterior inference on parameters of interest, and quantifying uncertainty. The functions are typically verbose, offering plenty of messages and warnings to the user where appropriate. Please see Murphy et al. (2019) for more info.

If you find bugs or want to suggest new features please visit the **IMIFA** GitHub issues page.

This vignette aims to reproduce some results in the Murphy et al. (2019) paper using the `mcmc_IMIFA()` and `get_IMIFA_results()` functions and demonstrates how the plots therein were created using the dedicated S3 `plot` method, while also demonstrating how to fit other models in the IMIFA family.

Installing IMIFA

IMIFA will run in Windows, Mac OS X or Linux. To install **IMIFA** you first need to install R. Installing Rstudio as a nice desktop environment for using R is also recommended.

Once in R you can type:

```
install.packages('devtools')

devtools::install_github('Keefe-Murphy/IMIFA')
```

at the R command prompt to install the latest development version of the package from the **IMIFA** GitHub page.

To instead install the latest stable official release of the package from CRAN go to R and type:

```
install.packages('IMIFA')
```

In either case, if you then type:

```
library(IMIFA)
```

it will load in all the **IMIFA** functions.

The GitHub version contains a few more features but some of these may not yet be fully tested, and occasionally this version might be liable to break when it is in the process of being updated.

The three main functions

There exist several utility functions in the package to solicit good prior hyperparameters (e.g. `G_priorDensity()`, `psi_hyper()`, `MGP_check()`) and to prepare results for producing nice plots (e.g. `mat2cols()`), this vignette focuses only on the three most important functions:

1. `mcmc_IMIFA()`
2. `get_IMIFA_results()`
3. and a dedicated S3 `plot()` method for objects of class "Results_IMIFA"

While it is possible to simulate data from a factor analytic mixture using the `sim_IMIFA_data()` function, specifying, among other things, the sample size `N`, the number of clusters `G`, and the number of variables `P`, with true parameters either supplied or also simulated, e.g.

```
# Simulate 100 observations from 3 balanced clusters with cluster-specific
numbers of latent factors
psi      <- matrix(rgamma(60, 2, 1), nrow=20, ncol=3)
mu       <- matrix(rnorm(60, -2 + 1:3, 1), nrow=20, ncol=3, byrow=TRUE)
sim_data <- sim_IMIFA_data(N=100, G=3, P=20, Q=c(2, 2, 5),
                           psi=psi, mu=mu)
```

the well-known Italian olive oil data set will be used throughout this vignette instead. You can load this data set after loading the **IMIFA** package by typing

```
data(olive)
```

and learn more about this data set by typing

```
?olive
```

Fitting the model & running the MCMC chain

The `mcmc_IMIFA` function provides an adaptive Gibbs/Metropolis-within-Gibbs sampler for nonparametric model-based clustering using models from the **IMIFA** family. The function facilitates model-based clustering with dimensionally reduced factor-analytic covariance structures, with automatic estimation of the number of clusters and cluster-specific factors as appropriate to the method employed. Factor analysis with one group (FA/IFA), finite mixtures (MFA/MIFA), overfitted mixtures

(OMFA/OMIFA), infinite factor models which employ the multiplicative gamma process (MGP) shrinkage prior (IFA/MIFA/OMIFA/IMIFA), and infinite mixtures which employ Pitman-Yor / Dirichlet Process Mixture Models (IMFA/IMIFA) are all provided. The function creates a raw object of class `'IMIFA'` from which the optimal/modal model can be extracted by `get_IMIFA_results`.

There are many, many options for specifying hyperparameters, specifying running conditions and pre-processing the data. These are documented both within `mcmc_IMIFA` and within various control functions (`mixfaControl`, `mgpControl`, `bnpControl`, etc.), are deferred, for brevity, to these functions' help files. Great care was taken to ensure the default function arguments governed by the control functions would be appropriate in most applications, but you can nevertheless access further helpful instructions by typing

```
?mcmc_IMIFA
```

and `?mixfaControl`, `?mgpControl`, `?bnpControl` etc. as needed. Arguments to these control functions can actually be supplied, provided they are named, directly to `mcmc_IMIFA`, and this convention is adopted throughout this document.

Be warned that the `mcmc_IMIFA` function calls in this section may take quite some time to run. Let's begin by fitting a Mixture of Factor Analysers model (MFA) to the unit-scaled `olive` data. For this, we must specify sequences of values for `range.G`, the number of clusters, and `range.Q`, the number of latent factors. Let's assume that uniquenesses are `isotropic` rather than `unconstrained`. This isotropic constraint provides the link between factor analysis and the probabilistic principal component analysis model (PPCA): note that we could also constrain uniqueness across clusters (but still be diagonal within each cluster) by specifying `uni.type="constrained"` or constrain uniquenesses to a single value (i.e. equal across all clusters and all variables) by specifying `uni.type="single"`. Let's elect not to store the latent factor scores, as this can be a huge drain on memory, with the caveat that posterior inference on the scores won't be possible. Let's also allow diagonal covariance as a special case where `range.Q` is `0`, and accept all other defaults (for instance, cluster labels will be initialised by `mclust`).

```
simMFA   <- mcmc_IMIFA(olive, method="MFA", n.iters=10000,
                       range.G=3:6, range.Q=0:3,
                       centering=FALSE, scaling="unit",
                       uni.type="isotropic", score.switch=FALSE)
```

Now let's instead have the numbers of cluster-specific latent factors be estimated automatically using a Mixture of Infinite Factor Analysers model (MIFA). This time, we'll also mean-centre the data and initialise the cluster labels using `kmeans` instead. Note that `range.Q` no longer needs to be specified, but it can be given as a

conservatively high starting value and upper limit. Let's accept the default, and also include the Infinite Factor Analysis model (IFA) as a special case where `range.G` is `1`.

```
simMIFA  <- mcmc_IMIFA(olive, method="MIFA", n.iters=10000, centering=TRUE,
                       range.G=1:3, z.init="kmeans")
```

MIFA doesn't entirely solve the issue of model choice, as you can see; `range.G` still needs to be specified. We can allow the number of clusters to instead/also be estimated automatically by fitting one of the overfitted mixture models (OMFA/OMIFA) or one of the infinite mixture models (IMFA/IMIFA). Let's fit an Overfitted Mixture of Infinite Factor Analysers, and override the default value for the starting value / upper limit for the number of clusters (`range.G`) and supply a sufficiently small Dirichlet hyperparameter (`alpha`) for the cluster mixing proportions. We can enforce additional shrinkage by varying other MGP hyperparameters, using arguments from `mgpControl()`.

```
simOMIFA <- mcmc_IMIFA(olive, method="OMIFA", n.iters=10000,
                       range.G=10, alpha=0.8, alpha.d1=3.5, nu=3,
                       alpha.d2=7, prop=0.6, epsilon=0.12)
```

Finally, let's run the flagship IMIFA model, on which all subsequent demonstrations and results will be based, for a greater number of iterations, accepting the defaults for most arguments. Note that the `verbose` argument, which defaults to `TRUE` will ordinarily print a progress bar to the console window. The default implementation uses the independent rather than dependent slice-efficient sampler; we could override the default for the parameter governing the rate of geometric decay by specifying `rho`.

```
simIMIFA <- mcmc_IMIFA(olive, method="IMIFA", n.iters=50000, verbose=FALSE)
```

Postprocessing and extracting optimum results

In order to extract results, conduct posterior inference and compute performance metrics for MCMC samples of models from the **IMIFA** family, we can pass the output of `mcmc_IMIFA` to the function `get_IMIFA_results()`. If, for instance, `simMFA` above was supplied, this function would find the pair of G and Q values which optimises a model selection criterion of our choosing and prepare results from that model only. If `simIMIFA` is supplied, this function finds the *modal* estimates of G and each q_g (the cluster-specific number of latent factors), and likewise prepares results accordingly.

This function can be re-ran at little computational cost in order to extract different models explored by the sampler used by `mcmc_IMIFA`, without having to re-run the model itself. New results objects using different numbers of clusters and different numbers of factors (if visited by the model in question), or using different model selection criteria (if necessary) can be generated with ease. The function also performs post-hoc corrections for label switching, as well as post-hoc Procrustes rotation to ensure sensible posterior parameter estimates, computes error metrics, and constructs credible intervals, the average similarity matrix, and the posterior confusion matrix.

Please see the function's help manual by typing `?get_IMIFA_results` for further assistance with the various function arguments.

If we wanted to choose the optimum MFA model, we would simply type

```
resMFA  <- get_IMIFA_results(simMFA)
```

If we instead wanted to explore the 3-cluster solution, construct 90% credible intervals, and have the number of latent factors chosen by another criterion, we could try

```
resMFA2 <- get_IMIFA_results(simMFA, G=3, criterion="aic.mcmc")
```

For now, let's just extract results from our IMIFA run above so we can proceed to visually examine them. Though the IMIFA model obviates the need for model selection criteria, the syntax for extracting results is exactly the same. However, this time, let's also summarise the clustering via the N x N similarity matrix obtained by averaging the adjacency matrices (admittedly at the expense of slightly slowing the function down!), so that we can visualise it later.

```
resIMIFA <- get_IMIFA_results(simIMIFA, z.avgsim=TRUE)
```

Before we examine the results in great detail, we can quickly summarise the solution as follows

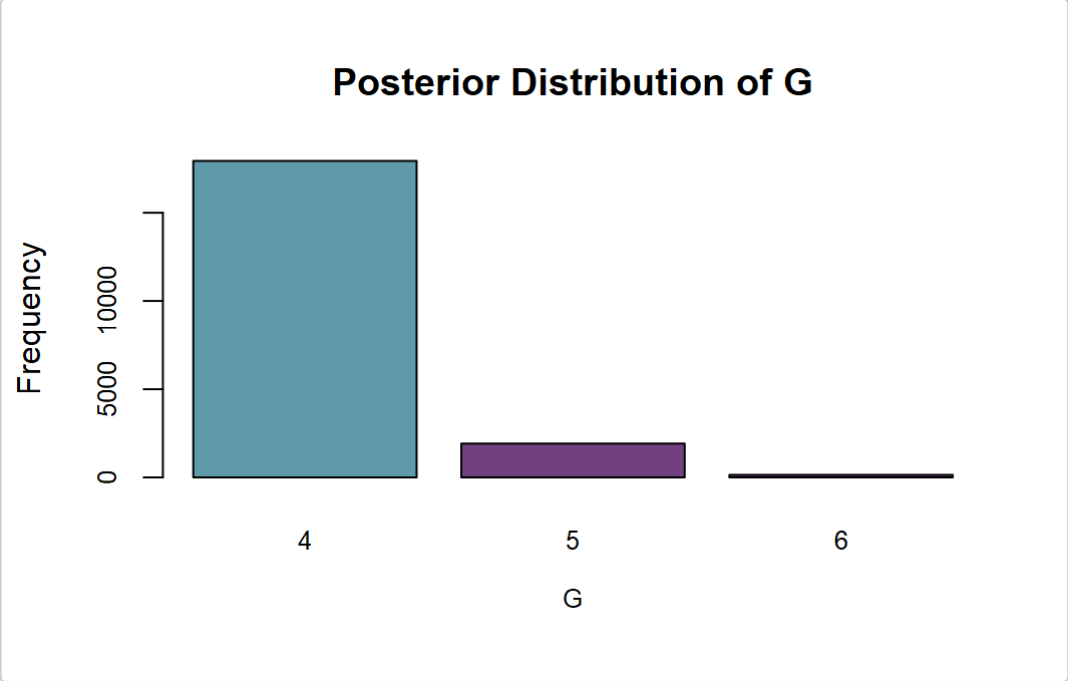
```
summary(resIMIFA)
## Call:    get_IMIFA_results.IMIFA(sims = simIMIFA, z.avgsim = TRUE, zlabels =
olive$area)
##
## The chosen IMIFA model has 4 groups with 6, 3, 6 and 2 factors respectively:
this Results_IMIFA object can be passed to plot(...)
```

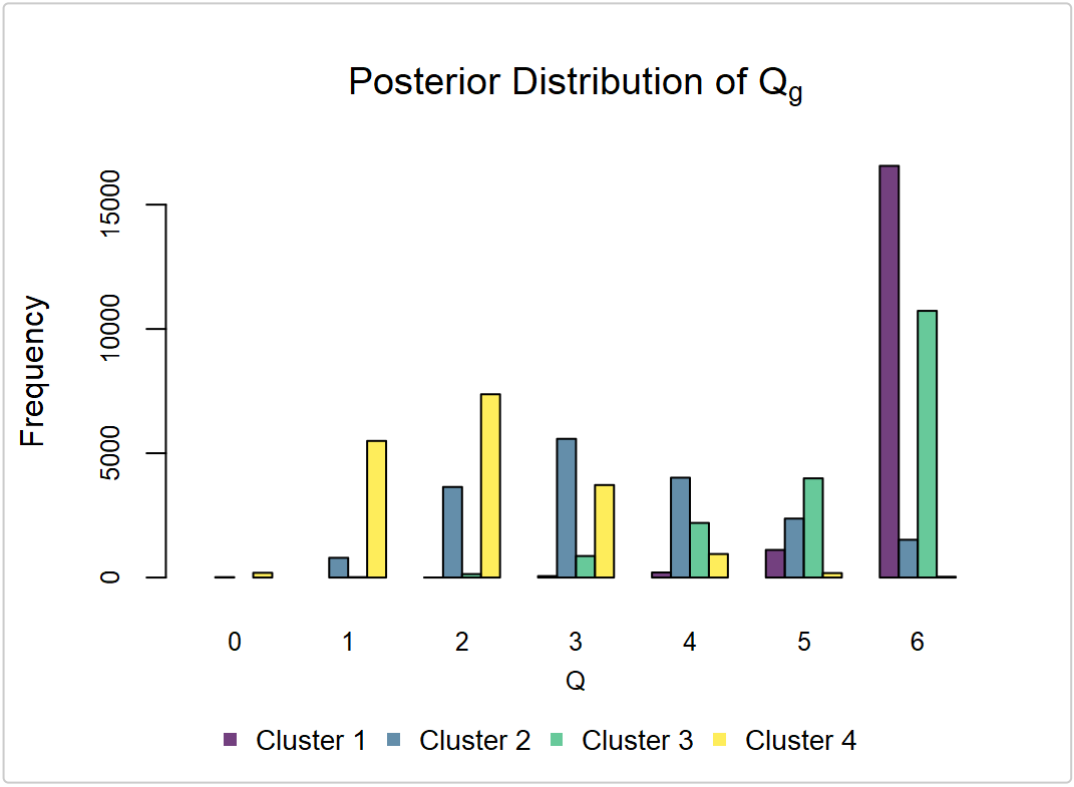
Visualing IMIFA results

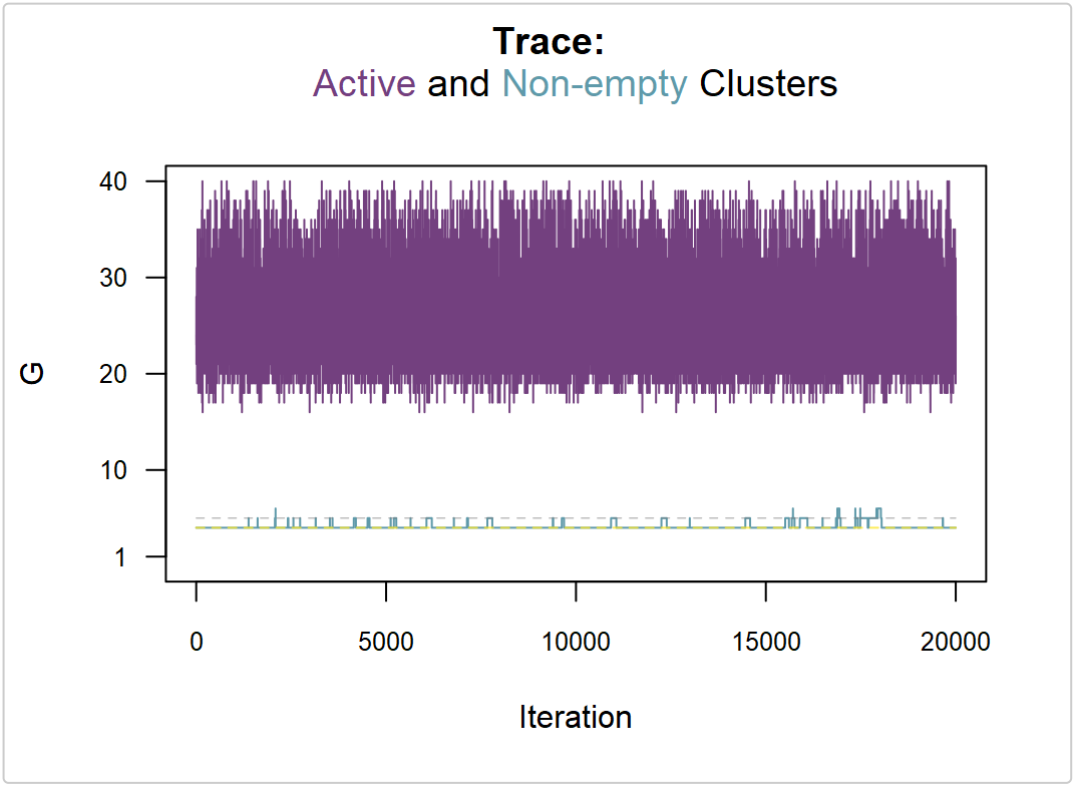
The object `resIMIFA` above is of the class `Results_IMIFA`. We can call `plot` on objects of this class to access a dedicated function for visualising output and parameters of inferential interest for IMIFA and related models. The two most important arguments, beyond the `Results_IMIFA` object itself, are `plot.meth` and `param`, where the former dictates the type of plot to be produced (one of `c("all", "correlation", "density", "errors", "GQ", "means", "parallel.coords", "trace", "zlabels")`, depending on the `method` employed originally by `mcmc_IMIFA`) for the `param`eter of interest (one of `c("means", "scores", "loadings", "uniquenesses", "pis", "alpha")`, depending on the `method` employed originally by `mcmc_IMIFA`). Note that `"all"` refers here to the options `"trace"`, `"density"`, `"means"`, and `"correlation"`. Note also that many of the function calls below will also print relevant output to the console window that is not always shown here. Please see the function's help manual by typing `plot.Results_IMIFA` for further assistance with the various function arguments.

Let's examine the posterior distribution of G and the posterior distribution of q_g for each of the 4 clusters. The third plot below, depicting the trace of the numbers of active and non-empty clusters, allows us to examine mixing of the chain with respect to G. The true number of clusters is estimated by $G = 4$, the modal value.

```
plot(resIMIFA, plot.meth="GQ")
```

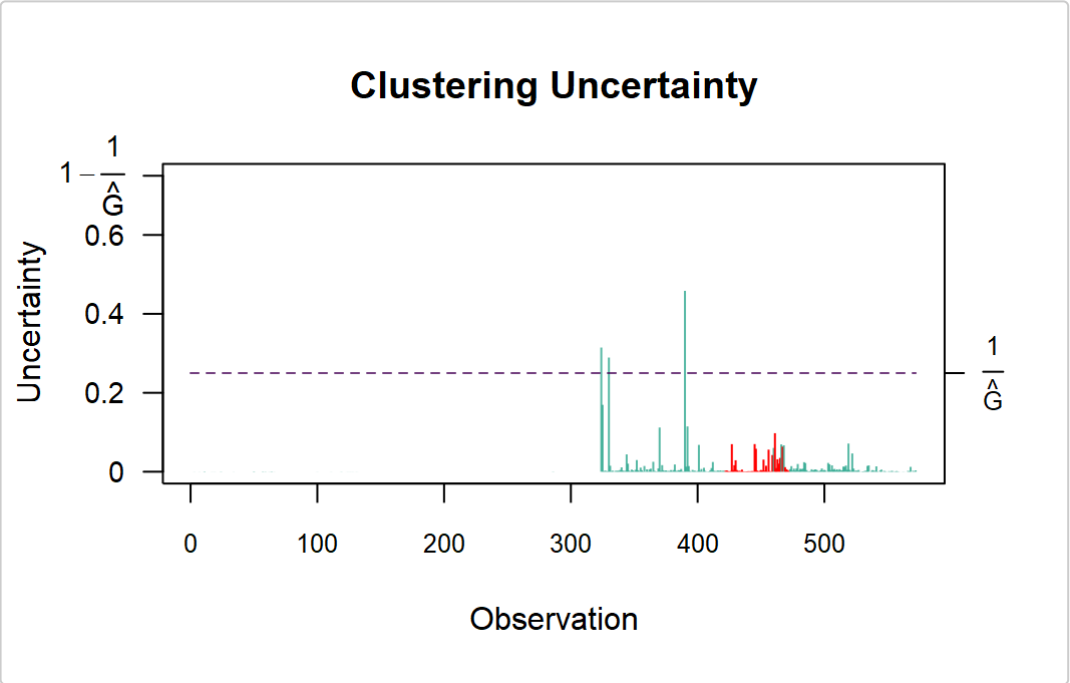



Posterior Distribution of Q_g
Frequency
0
5000
10000
15000
0
1
2
3
4
5
6
Q
Cluster 1
Cluster 2
Cluster 3
Cluster 4


Trace:
Active and Non-empty Clusters
G
40
30
20
10
1
0
5000
10000
15000
20000
Iteration

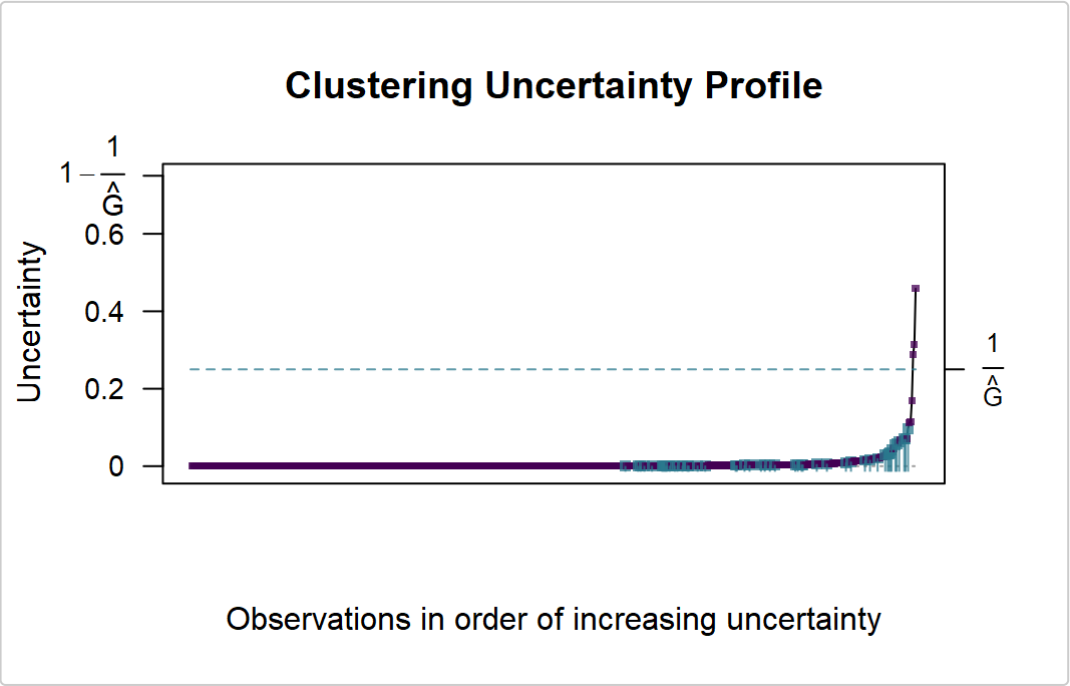
Let's examine clustering performance against the known cluster labels. Note that the cluster labels could have been already supplied to `get_IMIFA_results` too, but plotting allows clustering performance to be evaluated against new labels without having to extract the full results object all over again. More specifically, the code below allows us to visualise the clustering uncertainty (with or without the labels being supplied, in fact; when they are supplied, misclassified observations are highlighted, otherwise observations with uncertainty exceeding the inverse of the number of clusters are highlighted).

```
plot(resIMIFA, plot.meth="zlabels", zlabels=olive$area, g=1)
## confusion.matrix :
##           Observed
## Predicted Southern Italy Sardinia Northern Italy Sum
##       1              323        0              0 323
##       2                0       98              0  98
##       3                0        0            103 103
##       4                0        0             48  48
##       Sum            323       98            151 572
##
## rand :
## [1] 0.9697255
##
## crand :
## [1] 0.9370795
##
## misclassified :
##  [1] 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439
440
## [20] 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458
459
## [39] 461 462 463 464 465 467 469 470 471 472
##
## error.rate :
## [1] "8.39%"
```



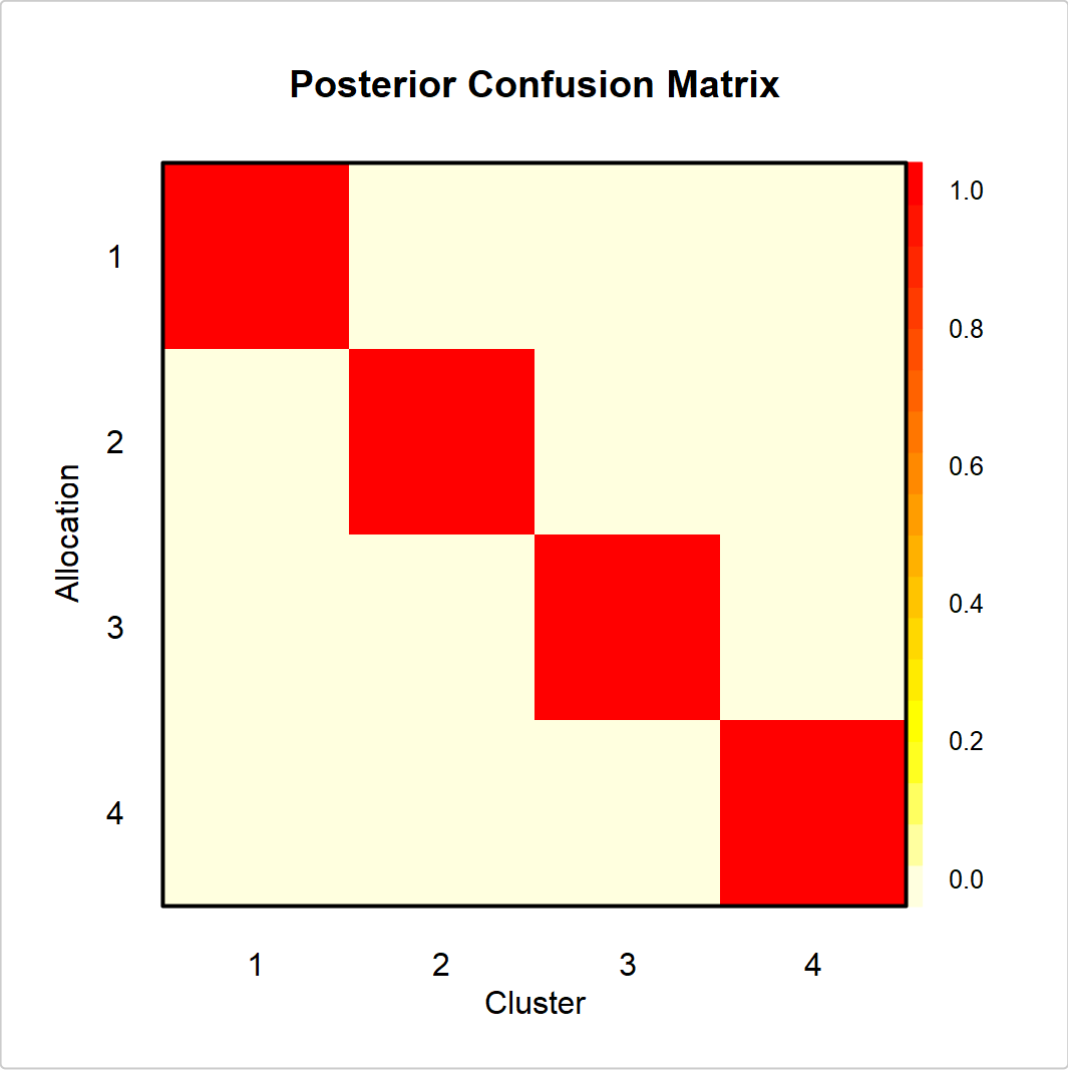
We can also plot a clustering uncertainty profile, also highlighting misclassified observations, with `g=2`. When `plot.meth="zlabels"`, `g=3` would mean to instead visualise the uncertainties in the form of a histogram.

```
plot(resIMIFA, plot.meth="zlabels", zlabels=olive$area, g=2)
```



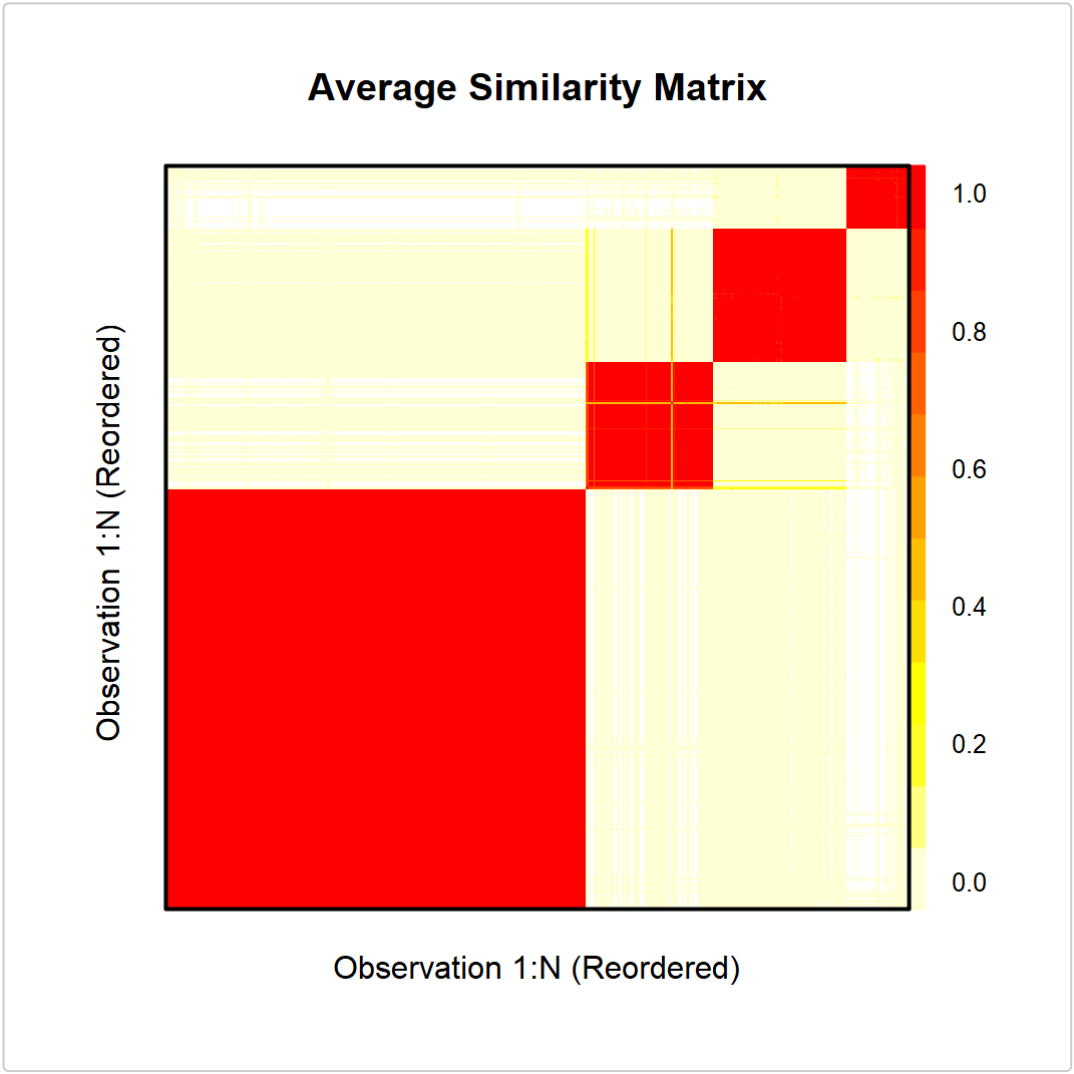
With `g=4`, we can visualise the posterior confusion matrix. The benchmark matrix for comparison is the identity matrix of order $G = 4$, corresponding to a situation with no uncertainty in the clustering.

```
plot(resIMIFA, plot.meth="zlabels", g=4)
```



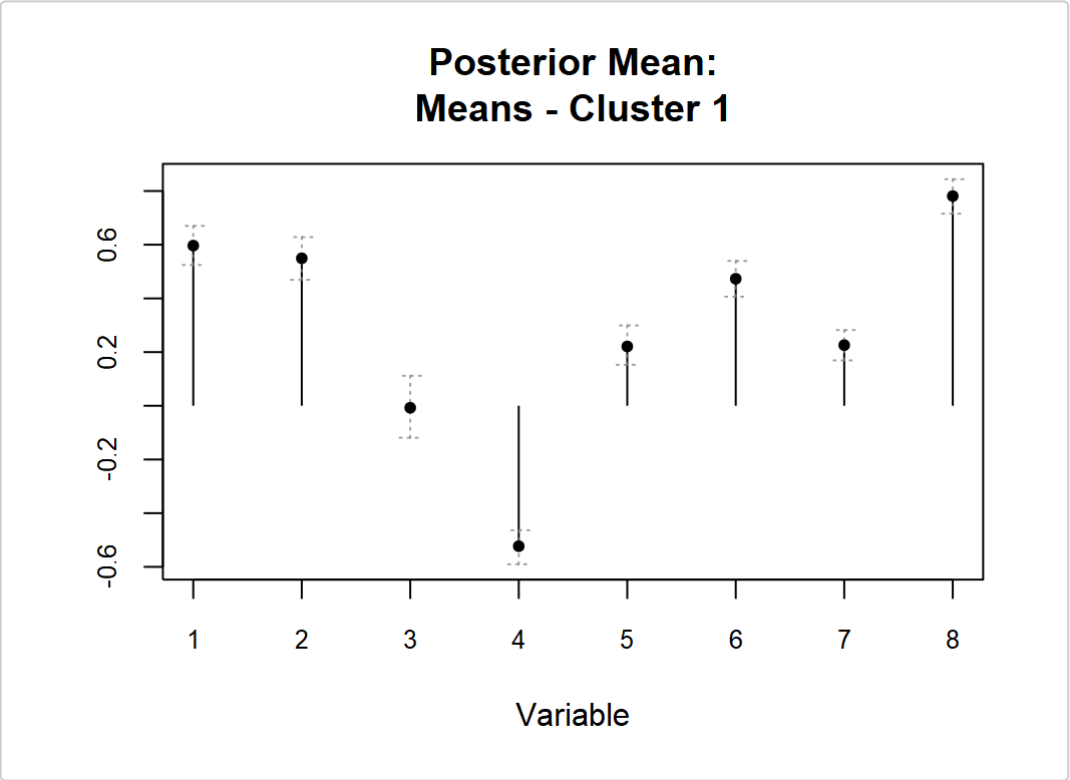
Finally, we can visualise the $N * N$ similarity matrix by supplying `g=5`. Had we not specified `g`, the function would cycle through the available plots.

```
plot(resIMIFA, plot.meth="zlabels", g=5)
```



To examine the posterior mean estimates of the cluster means in a given cluster (say the 1st), we can set both `plot.meth` and `param` to `"means"`. If the cluster isn't specified using the argument `g`, the user will be prompted to hit <Return> at the onset of each plot in order to cycle through similar plots for all clusters. In the code below `mat=TRUE` means, in this case, to plot all variables simultaneously. By default, credible intervals are also plotted. Note that the data were originally mean-centred and unit-scaled when `mcmc_IMIFA` was called.

```
plot(resIMIFA, plot.meth="means", param="means", mat=TRUE, g=1)
```



Had the factor scores been stored, we could examine the trace plots for them using the code below, where `mat=TRUE` and `by.fac=FALSE` (the default) means, in this case, to plot all factors simultaneously for a given observation: `ind=1` specifies that the observation of interest is the first (however this is not shown, as the scores were not stored).

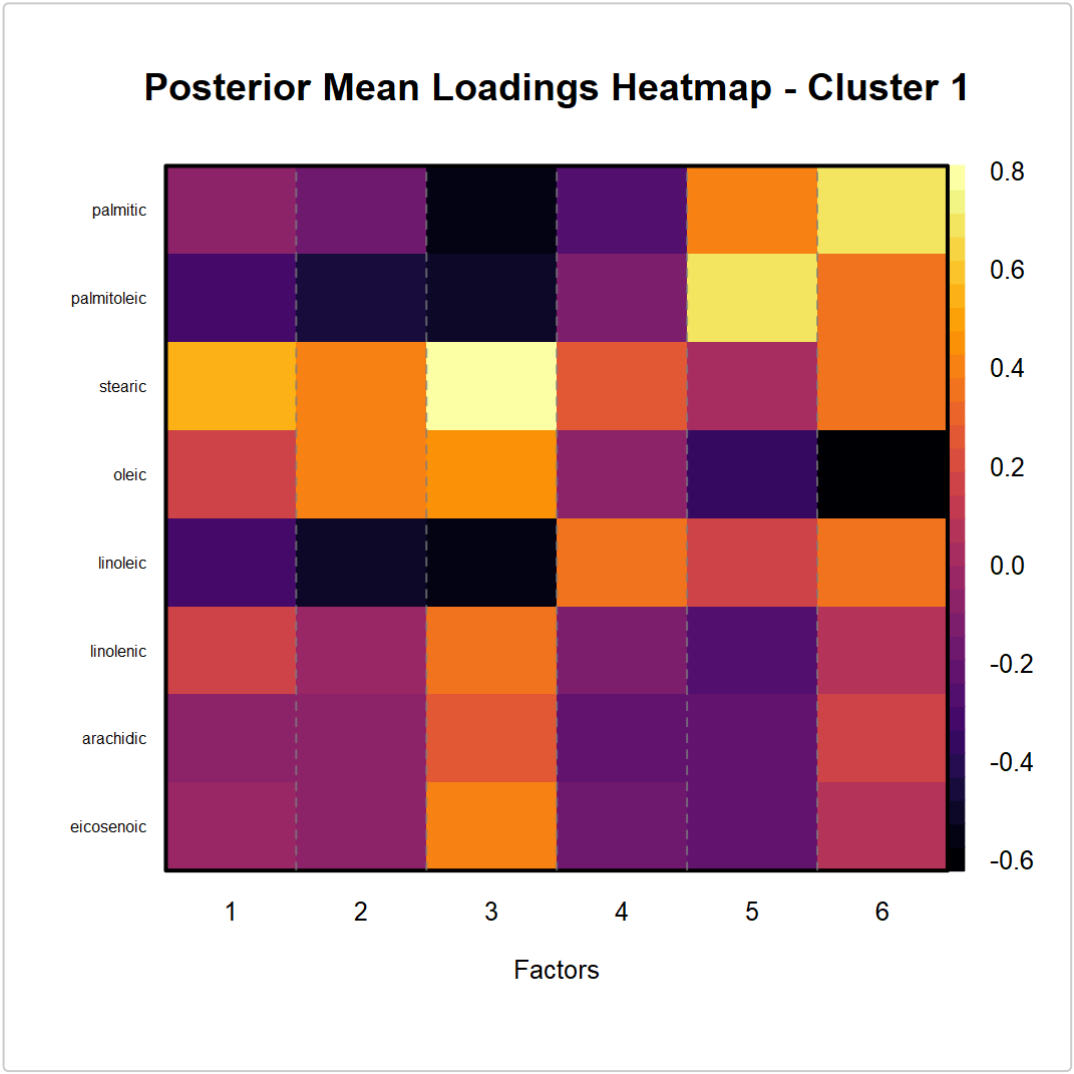
```
plot(resIMIFA, plot.meth="trace", param="scores", mat=TRUE, ind=1)
```

We could instead plot all observations simultaneously for a given factor, say the 2nd (also not shown).

```
plot(resIMIFA, plot.meth="trace", param="scores", mat=TRUE, by.fac=TRUE, fac=2)
```

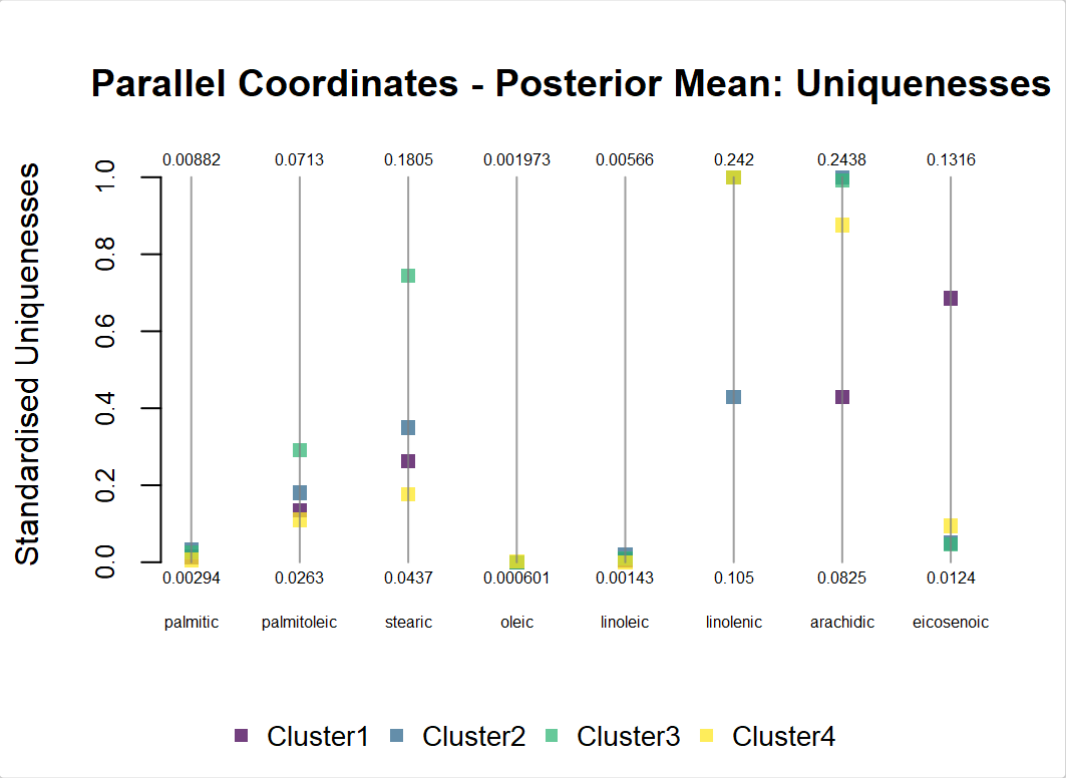
The code below will produce only a heatmap of the loadings matrix in the first cluster: note, however, that `heat.map=TRUE` by default for loadings (whereas the opposite is true for the scores). Darker colours correspond to entries which are more negatively loaded and *vice versa*.

```
plot(resIMIFA, plot.meth="means", param="loadings", heat.map=TRUE, g=1)
```



To examine posterior mean uniquenesses from all clusters in the form of a parallel coordinates plot, type

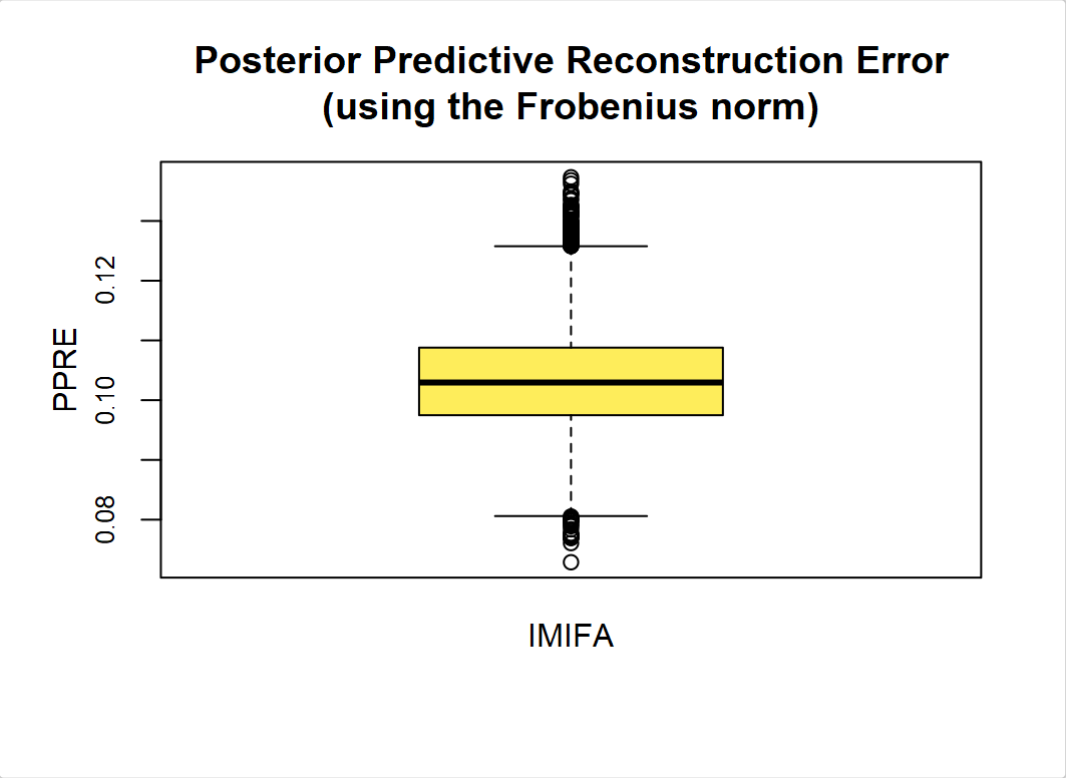
```
plot(resIMIFA, plot.meth="parallel.coords", param="uniquenesses")
```



We could have also used the argument `show.last` to visualise the last valid sample of uniquenesses from all clusters instead. This argument can be used to replace the posterior mean by the corresponding last valid sample for any combination of arguments to `plot.Results_IMIFA` that shows a posterior mean.

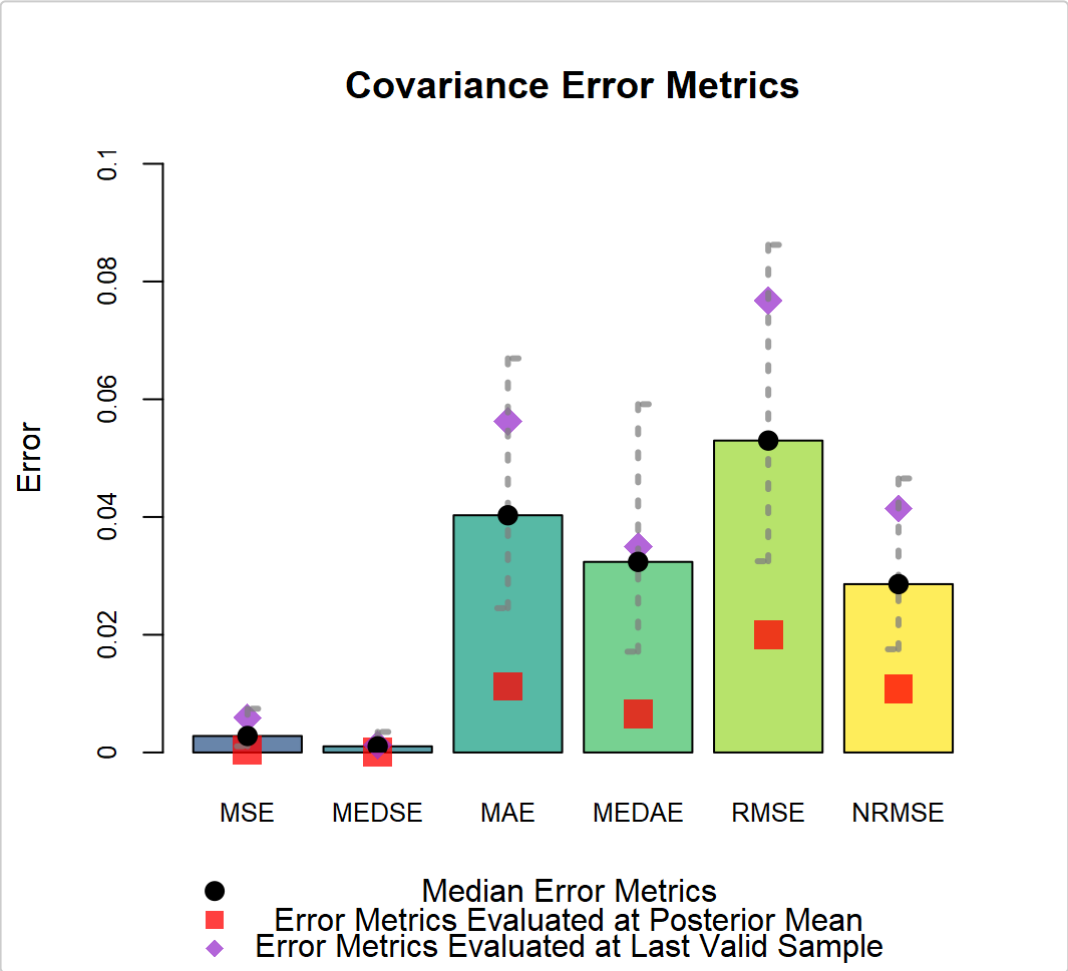
Posterior predictive checking to assess the appropriateness of the fitted model is also facilitated. The posterior predictive reconstruction error, obtained by comparing bin counts of the data against bin counts of replicate draws from the posterior distribution, can be visualised as follows. Setting `g=2` allows individual histograms to be depicted, while the PPRE offers an overall perspective across variables.

```
plot(resIMIFA, plot.meth="errors", g=1)
##              2.5%              Mean            Median Last Valid Sample
##        0.08767887        0.10327357        0.10295660        0.08807199
##             97.5%
##        0.12035785
```



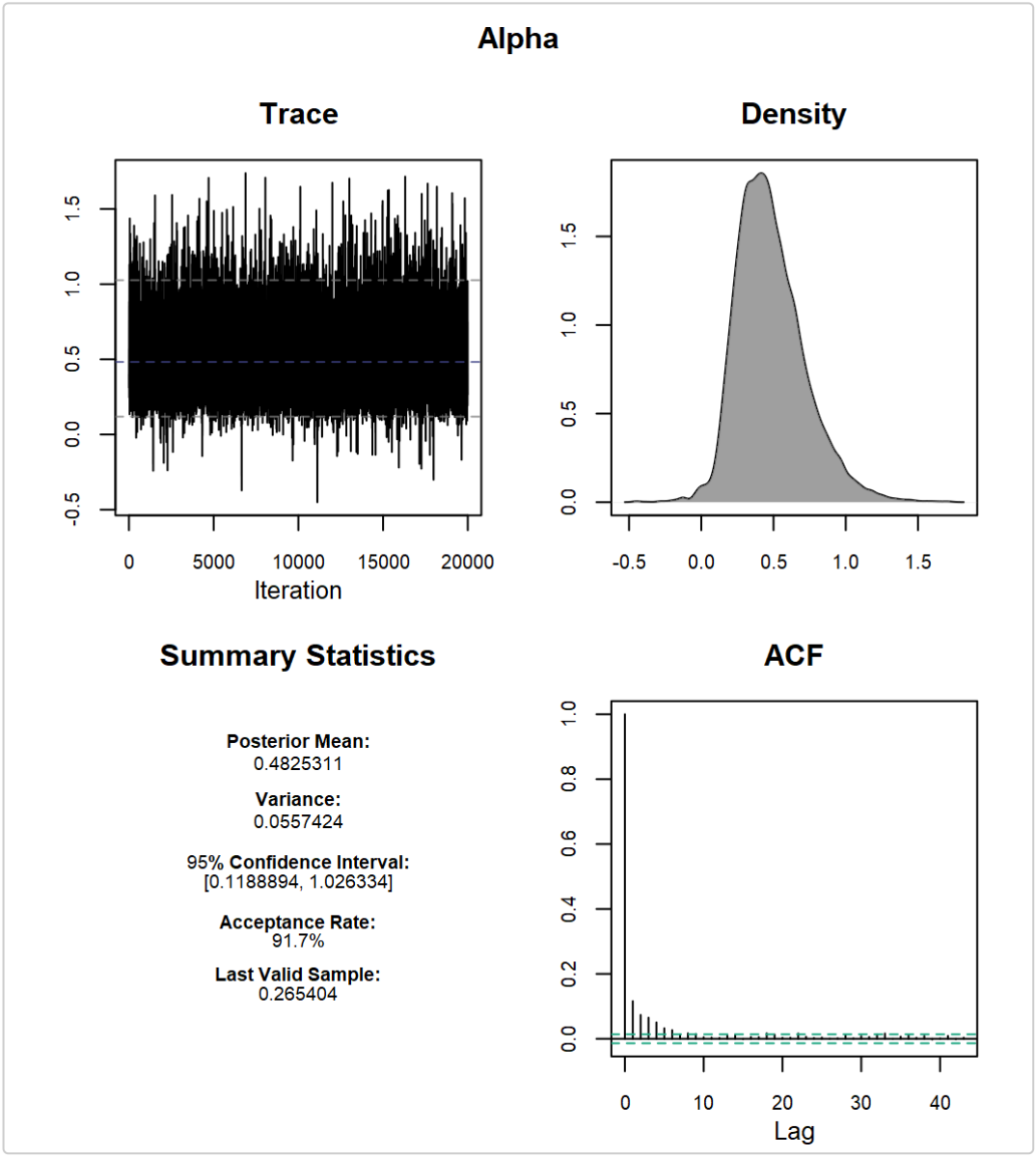
Quantifying the error between the empirical and estimated covariance matrices - and the uncertainty associated with those metrics - can also provide a useful indicator of the validity of the solution. For models which achieve clustering, the overall estimated covariance matrix is constructed from the cluster-specific estimated covariance matrices. A visualisation can be produced as follows:

```
plot(resIMIFA, plot.meth="errors", g=3)
##                                          MSE        MEDSE        MAE
MEDAE
## Medians                        0.0028070738 1.049235e-03 0.04029407
0.03237337
## Evaluated at Posterior Mean    0.0003993847 4.316984e-05 0.01124250
0.00654860
## Evaluated at Last Valid Sample 0.0058939475 1.232595e-03 0.05627498
0.03494658
##                                       RMSE      NRMSE
## Medians                        0.05298182 0.02860113
## Evaluated at Posterior Mean    0.01998461 0.01078827
## Evaluated at Last Valid Sample 0.07677205 0.04144378
```

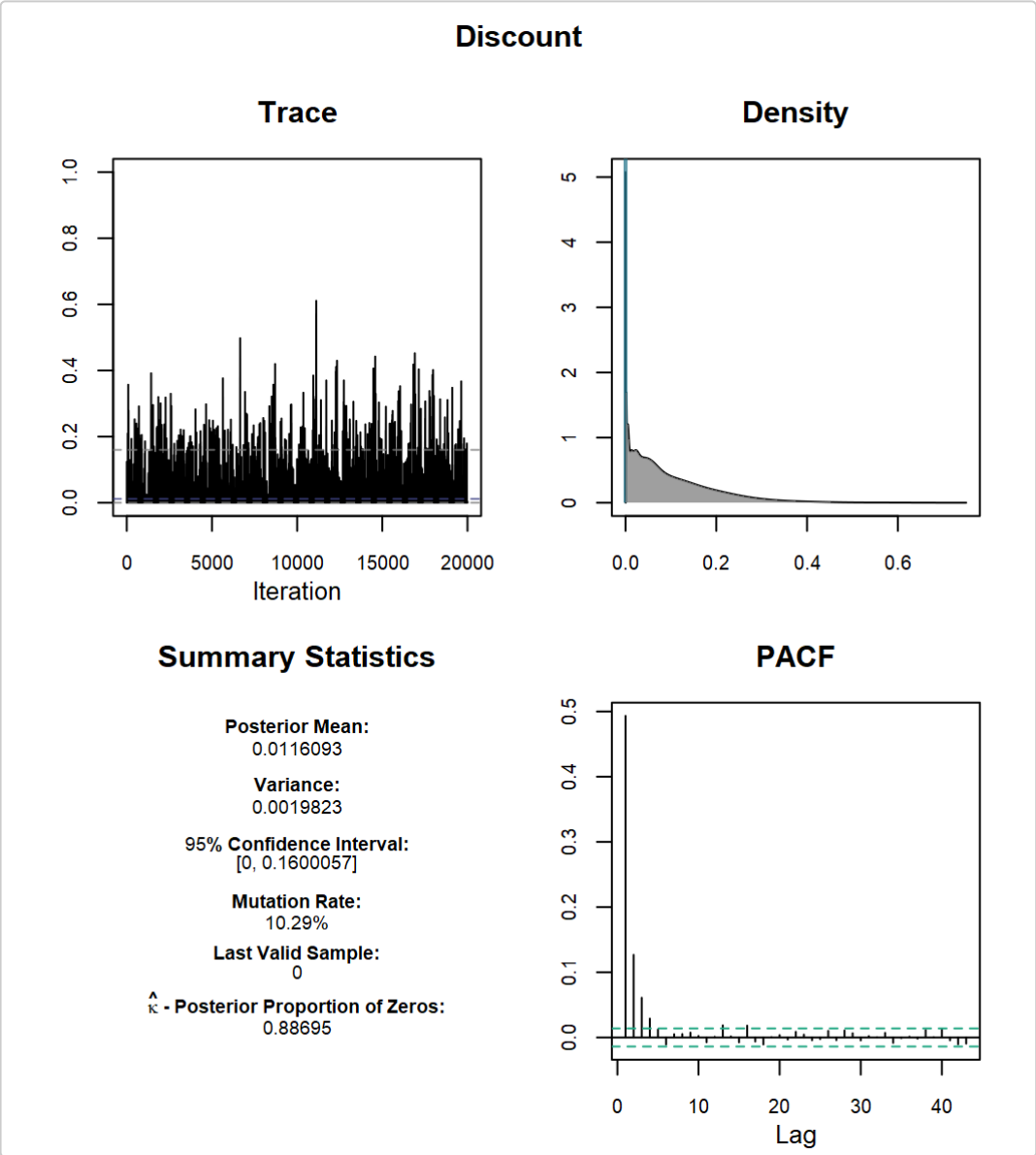


Finally, inference can be conducted on the Pitman-Yor concentration parameter α and discount parameter d (assuming `learn.alpha` and `learn.d`, respectively, were *not* set to `FALSE` when `mcmc_IMIFA` was initially run using the IMFA/IMIFA methods), where `all` below refers to `trace`, `density`, posterior `means`, and ACF/PACF (`correlation`) plots. The type of `correlation` plot can be toggled via the logical argument `partial`. Note that the `density` for `discount` accounts for the point-mass at zero built into its prior.

```
plot(resIMIFA, plot.meth="all", param="alpha")
```




```
plot(resIMIFA, plot.meth="all", param="discount", partial=TRUE)
```



References

Murphy, K., C. Viroli, and I. C. Gormley (2019). Infinite mixtures of infinite factor analysers. *Bayesian Analysis*, 1-27 URL https://projecteuclid.org/euclid.ba/1570586978.

Bhattacharya, A. and D. B. Dunson (2011). Sparse Bayesian infinite factor models. *Biometrika 98* (2): 291-306.

Chapter 5

Conclusions and Future Work

This final chapter summarises and discusses the novel model families introduced in Chapters 2–4, outlining commonalities between the proposed approaches and elaborating on general limitations and future research opportunities related to the overall theme. For some extensions already suggested within the chapters, additional details of the proposals are provided. Firstly, however, the main contributions of each chapter are briefly restated.

In this thesis, different parsimonious families of model-based clustering methods are described for addressing three main limitations of the standard finite Gaussian mixture model. Namely, such a model does not incorporate covariates in any way, its assumption that the underlying component distributions are multivariate Gaussian is inappropriate for categorical sequence data, and the model is generally not well suited to high-dimensional data settings.

In Chapter 2, the MoEClust model family is developed in order to address the equivalent aims of incorporating covariates in GPCMs and introducing parsimony to the special cases of the (Gaussian) MoE framework. Applications to univariate and multivariate data demonstrate improvements from both perspectives and richer insights into the type of observations characterising each cluster are provided.

In Chapter 3, the MEDseq model family is proposed for reconciling the distance-based and model-based approaches for clustering categorical sequence trajectories, guided by an application to data on the monthly employment activities of a cohort of Northern Irish youths. Weighted variants of the Hamming distance are employed as the distance measure, amounting to improved substitution costs re-

flecting replacements of states. Sampling weights and concomitant variables are included in the clustering process in a coherent manner.

In Chapter 4, the IMIFA model is presented as the flagship model in a family of factor-analytic mixtures. In particular, the IMIFA model is a fully Bayesian non-parametric approach to clustering high-dimensional data. A PYP prior is employed which assumes infinitely many mixture components. The MGP shrinkage prior is generalised to the mixture setting to allow infinitely many latent factors within each component. An adaptive inferential MCMC algorithm is developed to enable automatic estimation of these quantities. Hence, the need to select and employ model selection criteria in a mixture of factor analysers is obviated and further flexibility is achieved by allowing the number of factors to be cluster-specific.

Associated with each body of work are distributed software packages for the statistical software platform R, namely the `MoEClust`, `MEDseq`, and `IMIFA` libraries. Thus, the results in this thesis are reproducible and the proposed model families are easily accessible to interested researchers.

By virtue of each chapter introducing a *family* of models, the issue of model selection for identifying the one which best fits the data is an important common theme throughout this thesis. In any unsupervised clustering problem, there is a need to choose the optimal number of components G. In Chapters 2 and 3 there is also a need to choose the optimal set of constraints, in terms of the component covariance matrices and precision parameter settings, respectively.

As both the MoEClust and MEDseq model families are characterised as mixtures of experts frameworks, there is the further need in these chapters to identify the optimal subset of covariates to include. While MEDseq models allow concomitant variables to influence the cluster-membership probabilities, the issue is complicated further for MoEClust models as different subsets of the related covariates are also allowed to influence the component mean parameters.

Optimal MoEClust models are identified among a range of fitted models using the BIC or ICL criteria and a novel greedy forward stepwise variable selection procedure. This procedure guides the inclusion of covariates and hence implicitly guides the choice of the model type in terms of the special cases of the MoE framework. Each step of the procedure considers several potential sub-steps, starting from a model with $G = 1$ and no covariates; adding a component, adding a single

covariate to the component mixing proportions, or adding a single covariate to the component distributions. Each sub-step is evaluated over all available GPCM constraints and the step which yields the best improvement in the chosen criterion is accepted until no step yields a further improvement. In comparison to exhaustively searching over all possible models, the stepwise procedure is shown to converge to the optimal model in both applications. Throughout Chapter 2, it sufficed to consider only additions, rather than additions and removals, though this may not hold true in settings with larger numbers of covariates. Presently, the `MoE_stepwise` function in the `MoEClust` R package allows only additions; the searches with both additions and removals were conducted manually, but automating this would be a reasonably straightforward extension to the code. As an alternative approach to stepwise covariate selection, a LASSO-like L_1 penalty could be imposed in the gating and/or expert network regressions. This would allow covariate selection and model estimation to be conducted simultaneously, albeit with the caveat that the penalty parameter typically needs to be chosen via cross-validation, which brings an associated computational burden of its own. Other future work involves implementing this stepwise procedure in such a way that the sub-steps evaluated in each step are done so in parallel rather than in series in order to speed up the algorithm. Indeed, exhaustive searches could also be similarly parallelised.

Model selection criteria based on parameter count penalties such as the BIC and ICL are shown to perform poorly for selecting optimal MEDseq models, either by virtue of the discrete nature of the data or the properties of the distance metrics employed. Moreover, calculating exact, non-asymptotic posterior model probabilities is not feasible in this setting, due to the difficulties in computing the marginal likelihood integrals. For MEDseq models, there is a need to choose the optimal precision parameter settings, as well as the number of components and the subset of covariates to include. To this end, a model-selection tool which is free from parameter counts is used, namely the (weighted) mean DBS criterion, in tandem with a stepwise selection strategy similar to that proposed for MoEClust models in Chapter 2. The DBS criterion is comparable to the ASW criterion. The ASW criterion uses the 'hard' cluster labels as one of its inputs, while the DBS criterion is particularly apt in the model-based clustering context as it preserves the information contained in the estimated matrix of 'soft' cluster-membership probabilities.

The IMIFA model, on the other hand, obviates the need for model-selection criteria entirely and requires only a single run to estimate both the number of components and the numbers of component-specific factors, thus making it the most computationally efficient model in the IMIFA family. This has the advantage of circumventing the need to choose the model-selection criterion; indeed, in many applications and settings, different criteria often suggest different optimal models. For models in the IMIFA family which do rely on model-selection criteria, the BICM criterion, a posterior simulation based version of the BIC, is used. Such models include the MIFA model, for instance, which is appropriate in cases where one wishes to fix the number of clusters but estimate the numbers of cluster-specific factors. If one wishes to fit a range of MIFA models with different G values, the BICM criterion is particularly useful for nonparametric models where the number of free parameters is difficult to quantify.

The IMIFA model family can be expanded further by imposing constraints on the elements of $\mathbf{\Lambda}_g$ and/or $\mathbf{\Psi}_g$, as per McNicholas and Murphy (2008), to yield additional parsimony. Indeed, imposing isotropic constraints on $\mathbf{\Psi}_g$, leading to a model that corresponds to an infinite mixture and infinite-dimensional extension of probabilistic principal components analysers (Tipping and Bishop, 1999), has been shown to be trivial in this setting. However, this naturally and problematically reintroduces the need to employ model selection criteria — such as the BICM — to identify the optimal model, as appropriate constraints must be chosen.

There is also scope for further expanding the MoEClust model family. In contrast to MEDseq models, which allow covariates to enter the gating network only, covariates are additionally allowed to enter the component distributions of MoEClust models. However, covariates only enter via the component means.Firstly, therefore, it is of potential interest to instead or also allow covariate-dependence in the covariance matrices. Pouhramadi (1999) and extensions in Xu and MacKenzie (2012) provide a framework for doing so, albeit with applications to longitudinal data. While this would capture heterogeneity in the dependencies among response variables with respect to the space of covariates, such an extension to MoEClust would, again, further complicate the issue of model selection. Moreover, the issue of how to allow the covariance matrices to depend on covariates across the full range of constrained GPCM settings for $\mathbf{\Sigma}_g$ and σ_g remains an open question.

Relatedly, it should be noted that the maximum likelihood estimates of the component covariance matrices, i.e. the M-steps in the EM algorithm, are biased: the divisor is n rather than $n-1$. An investigation into the ramifications of this bias, particularly for small data sets, is of interest despite unbiased estimation of $\boldsymbol{\Sigma}_g$ by definition no longer maximising the likelihood. Furthermore, while on the subject of the MLE of the component covariance matrices, modifying the various M-step functions in the `MoEClust` R package to implement the eigenvalue constraints considered in García-Escudero et al. (2018) would be a valuable and reasonably straightforward extension, which would mitigate against the problems of spurious solutions and degenerate components.

A second potential expansion beyond the 6 special cases of the MoE framework considered here is to allow different subsets of covariates to affect each dependent variable. Such models have been recently introduced (Galimberti and Soffritti, 2019), in a setting where the covariates only enter the parsimoniously parameterised Gaussian component densities, under the name 'seemingly unrelated clusterwise linear regression models', and can be seen as a generalisation of the expert network MoE model under which the same set of regressors is used for each dependent variable. Compared to the expert network MoE model, further parsimony can be achieved under this mixture of SUR models framework if a given covariate is only relevant for fewer than G mixture components. It would thus be of interest to also incorporate gating covariates in such a setting, to yield an analogue of the full MoE model, or to constrain the mixing proportions across components to yield an analogue of the equal mixing proportion MoE model. However, these extensions would, as above, make the already difficult task of covariate selection even more complicated. Galimberti and Soffritti (2019) also consider constraints on the expert network regression coefficients, such that $\boldsymbol{\gamma}_g = \boldsymbol{\gamma} \, \forall \, g$. This would be of interest also in the existing MoEClust models with expert network covariates.

With regard to Appendix 2.D, in which MoEClust models are discussed from the point of view of prediction rather than clustering, a number of limitations can be identified. Firstly, as the noise component follows a uniform distribution, predictions of new data may suffer if the new data falls outside the region used to define the hypervolume. To this end, the approach in Leisch (2008), under which the noise component instead follows an inflated Gaussian distribution with an intercept-only

expert network, may prove fruitful. However, there may be some sensitivity to the degree of inflation, especially for models which already omit covariates in the expert networks of the non-noise components. Secondly, the predictions of $\widehat{\mathbf{y}}_i^{\star}$ and $\widehat{z}_{ig}^{\star}$ are merely point estimates about which no uncertainty is provided. This raises the issue of parameter uncertainty in general for MoEClust models. To this end, the weighted likelihood bootstrap, already used for MEDseq models in Chapter 3, though originally proposed for Gaussian mixture models by O'Hagan et al. (2019), could be employed. Thirdly, it would be of interest to extend the validation measures proposed for finite mixtures of regressions by Ingrassia and Punzo (2019) — particularly the local and global coefficients of determination and the normalised explained sum of squares — to MoEClust models for multivariate response data and/or MoEClust models with a noise component. Finally, it remains to assess whether the CWM framework outperforms the MoEClust model family in terms of prediction. We conjecture that this would typically be the case; as CWMs explicitly model the marginal density of the they covariates, they allow for *assignment dependence* (in the sense that the distribution of $\mathbf{x}_i$ affects the cluster assignment of observation i). This can only be achieved in the MoEClust paradigm for models with gating network concomitants.

Throughout this thesis, maximum likelihood estimation via the EM and ECM algorithms (Chapters 2 and 3, respectively) and Bayesian estimation (Chapter 4) have been employed as appropriate to the task at hand. However, estimating MoEClust and MEDseq models in a Bayesian fashion is entirely feasible. Firstly, it may be necessary to do so if expanding the MEDseq model family to other, more complicated distance metrics, namely OM. Secondly, sparsity-inducing LASSO-like regularisation priors on the regression coefficients could help guide the inclusion of covariates under both model families. For the gating network in particular, which relates concomitant variables to the cluster mixing proportions, considering the probit rather than logit link in the Bayesian setting could be beneficial, as the conditional distributions for the regression parameters for the probit model achieve conjugacy, which is not the case for the logit model (Geweke and Keane, 2007).

The issue of choosing the subset of covariates to include in the mixtures of experts model families given by MoEClust and MEDseq is an issue of *covariate* selection in terms of $\mathbf{x}_i$ rather than an issue of *variable* selection in terms of the

response variables $\mathbf{y}_i$ or observed sequences $\mathbf{s}_i$. Considering the 'variables' in categorical sequence data as the sequence positions, some MEDseq models — those for which the weighted Hamming distance metric allows specific precision parameters for each time point — implicitly include variable selection, by virtue of weighting the contribution of each time point to the overall distance differently. However, no variable selection in terms of $\mathbf{y}_i$ has been conducted for MoEClust models. In situations where there exist some variables which carry no information about the group structure, this may adversely affect overall clustering performance.

An excellent review of variable selection methods for model-based clustering is provided by Fop and Murphy (2018), in which a distinction is drawn between so-called filter approaches and wrapper approaches. The former set of approaches amount to pre-screening (or post-screening) the variables before (or after) estimating the model and are easy to implement and computationally efficient. Wrapper approaches, on the other hand, simultaneously conduct model estimation and variable selection. A natural candidate strategy in the MoEClust setting, given the stepwise approach already adopted for covariate selection, would be to adapt the greedy search algorithms implemented in the R package `clustvarsel` (Scrucca and Raftery, 2018) for GPCMs, without dependency in any way on covariates, to the Gaussian MoE framework. This approach recasts variable selection as a model selection problem, whereby the decision to include or exclude a response variable is made on the basis of an information criterion (e.g. BIC). It is of particular interest given the demonstration of its superiority over alternative methods in a comparison conducted in Fop and Murphy (2018).

Variable selection for models in the IMIFA family is also of interest. While the robustness study in Appendix 4.C shows that noisy *observations* are isolated in a cluster of their own, noisy *variables* adversely affecting performance is also shown. A clearer example of the need to incorporate variable selection in IMIFA models is provided by the USPS data application. When discarding peripheral pixels around the edges of the box bounding the digits with low standard deviations (< 0.7), a cleaner $\widehat{G} = 20$ solution is obtained, which achieves an adjusted Rand index of 0.41, compared to the solution obtained on the full data set ($\widehat{G} = 21$, ARI=0.33). While this approach — a filter approach in the terminology of Fop and Murphy (2018) — is intuitive, as Bouveyron and Brunet-Saumard (2014) point out that such

pixels are unlikely to be discriminatory given the geometry of the digits, a more principled approach beyond a naïve pre-processing of the data is clearly desirable.

Indeed, despite the stated intention for models in the IMIFA family to be used for clustering high-dimensional data, we must caution that factor-analytic mixture models are not really suitable for *very* high-dimensional settings. While the number of covariance parameters is admittedly greatly reduced, there can potentially still be many loadings and uniqueness parameters to estimate when p is extremely large. Moreover, there are still p component mean parameters in each cluster, regardless of the covariance decomposition employed. One potential avenue of exploration in such settings is to consider factor-analytic co-clustering models; in principle, a PYP prior could be assumed for the row and/or column clusters. Secondly — in the spirit of Bouveyron and Brunet-Saumard (2014), in which variable selection and dimension reduction are achieved simultaneously by imposing a penalty term on the component means — a shrinkage prior could be imposed on $\boldsymbol{\mu}_g$. If $\mu_{jg} = 0 \, \forall \, g$, the j-th variable would be deemed irrelevant. Relatedly, the link between the rows of $\boldsymbol{\Lambda}_g$ and the variables in the data matrix could be exploited by extending the MGP prior to include a row-wise *variable-specific* shrinkage parameter, in addition to the current shrinkage parameters on the local, column, and cluster levels.

Beyond the notions of model selection and covariate selection, another commonality between Chapters 2 and 3 is the inclusion of a uniform noise component for capturing outliers. In the MoEClust model family, the noise component is conceived of as an 'additional' component in a mixture where the remaining components are otherwise Gaussian, whereas with MEDseq models, the noise component is considered as one of the G components as it arises naturally from restricting all precision parameters in the component to be equal to zero. While covariates are allowed to influence (or not influence) the probability of belonging to the noise component in both cases, the noise component in MoEClust models does not otherwise account for observations being outliers with respect to the covariates $\mathbf{x}_i$, i.e. leverage points. To this end, the aforementioned approach of Leisch (2008) may again prove fruitful. Similarly, including an explicit noise component in the IMIFA model family to robustify inference is also feasible. This could be achieved using a mixture of mixtures approach, whereby

$$f(\mathbf{x}_i) = \pi_0 \mathrm{N}_p(\mathbf{x}_i; \boldsymbol{\mu}_0, \boldsymbol{\Psi}_0) + (1 - \pi_0) \sum_{g=1}^{\infty} \pi_g \mathrm{N}_p\left(\mathbf{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g \boldsymbol{\Lambda}_g^\top + \boldsymbol{\Psi}_g\right),$$

by imposing the restriction that $q_0 = 0$ and inflating the entries of the hypercovariance matrix $\boldsymbol{\Psi}_0$ *a priori*.

Including a noise component in IMIFA models is one sense in which the developed model families could be unified. Another step in this direction is to consider IMIFA models with dependence on covariates or MoEClust models with factor-analytic covariance structures. The former would amount to an infinite mixture and infinite factor extension of the MFA model with covariates proposed by Fokoué (2005), under which the zero-mean assumption in the prior for each latent factor score is replaced by a covariate-dependent non-zero hypermean. The latter could be feasibly achieved using the same trick relying on the residuals from the weighted multivariate linear regressions in the component densities already exploited by MoEClust for GPCM covariance structures. Thus, a mixtures of experts equivalent to the model family of McNicholas and Murphy (2008) could be easily developed. Indeed, this same trick could be used to consider an alternative to the underlying multivariate Gaussian distribution in MoEClust models, namely the multivariate t-distribution and the associated tEIGEN family of covariance matrix constraints (Andrews and McNicholas, 2012). The assumption of a multivariate t-distribution in each component is also of interest for the IMIFA model family.

Relatedly, considering MEDseq models with an alternative distance measure, namely OM, could improve the results on the MVAD data. While the intractable normalising constant under an exponential-distance model based on OM greatly complicates model fitting, it remains a potential avenue of future exploration. OM can accommodate sequences of unequal length by assigning costs to insertions and deletions. However, there is potential for accommodating unequal lengths in MEDseq models still based on the Hamming distance and weighted variants thereof, provided the time points are temporally aligned. This would involve computing the Hamming distance for the overlapping positions and adding the number of non-overlapping positions, thus assuming the worst-case scenario for the non-overlapping positions, i.e. that they are not equal. While it remains to be seen how the normalising constant would be affected under such a model, estimating

the precision parameters in this setting would be a straightforward extension. Estimating the modal sequence(s) would require finding the mode for each sequence position across only the partially overlapping set of available observations.

While unequal lengths are often attributable to missing data, missingness in terms of internal gaps can still arise for sequences of equal length (though there is no missing information for the MVAD data application considered). In this instance, the naïve solution implemented in the `TraMineR` package (Gabadinho et al., 2011) for computing pairwise dissimilarities using the Hamming distance or other measures — rendered feasible by the categorical nature of the data — may prove useful in the MEDseq setting; namely, to add 'missing' to the alphabet as another distinct state and increment v by 1. Depending on the level of missingness, this approach could result in modal sequence position estimates $\widehat{\theta}_{g,t}$ which are 'missing'.

An altogether more sophisticated approach for handling missing data in model-based clustering analyses in general is suggested in the unpublished work of Biernacki et al. (2019), which embeds management of the missing data mechanism into the model by jointly modelling the observed data (here $\mathbf{S} = (\mathbf{s}_i, \ldots, \mathbf{s}_n)$) and its pattern of missingness ($\mathbf{C} = (\mathbf{c}_i, \ldots, \mathbf{c}_n) \in \{0, 1\}^T$, where $c_{i,t} = 1$ when $s_{i,t}$ is missing and $c_{i,t} = 0$ otherwise). The expected complete data log-likelihood separates into a portion due to $\mathbf{S}$ and a portion due to $\mathbf{C}$ and can be maximised using a stochastic variant of the EM algorithm. In particular, Biernacki et al. (2019) propose a missing not-at-random mechanism (MNAR) using a logistic model for $\mathbf{C}$ which is allowed to depend on either or both the data themselves and the latent cluster membership indicator variables. The MNAR mechanism appears to be particularly apt for life-course sequences given that missingness tends to be consecutive, particularly volatile sequences are likely to exhibit more missingness, and missingness is often attributable to shorter follow-up times for some study subjects (Halpin, 2016).

Another, related problem is when the intervals between time points are irregular. Fortunately, MEDseq models can be used in such settings (by virtue of the Hamming distance and the weighted variants thereof implicitly modelling sequences as discrete, whole strings, rather than as time-to-event processes as per the Markovian framework), as long as the time points are aligned, i.e. the intervals are irregular but common for all observations. In essence, this amounts to a situation in which information is missing for certain time periods for all subjects. In such

instances, the MEDseq models with time-varying precision parameters are likely to be especially appropriate and useful. However, accounting for intervals which are different for different subjects remains an open problem in the sequence analysis community, even if such situations are not typical of life-course data. A potentially useful first step would be to appropriately pad out the data with missing values and then treat the problem using the suggested strategies for accounting for missingness using the Hamming distance and its weighted variants discussed above.

Moreover, as the exponent of an exponential-distance model based on the Hamming distance, given by $\lambda \sum_{t=1}^{T} \mathbb{1}(s_{i,t} \neq \theta_{g,t})$, implies substitution costs of λ, arguably none of the models in the MEDseq family actually employ the simple matching Hamming distance. Thus, allowing λ to be fixed to 1, or indeed some other value(s), may be of interest for researchers who wish to truly assume the Hamming distance (or normalised variants thereof) or estimate substitution costs by other means. This would greatly reduce the number of estimable parameters, especially for versions of the model which allow the precision parameters to vary across time points.

Another point related to the substitution costs, given that the use of sampling weights induces an observation-specific rescaling of the precision parameter(s), is that it may be of interest to allow random rather than fixed likelihood weights. This is achieved in Gebru et al. (2016) by assuming a gamma distribution for each w_i and using two sequential E-steps, one for the cluster-membership probabilities z_{ig} and one for the random weights w_i.

Furthermore, the rescaling of the precision parameter(s) induced by w_i raises the prospect of allowing covariates to additionally (or instead) affect the precision parameter(s), such that the MEDseq models would become — in the terminology of the MoEClust model family — 'full' mixture of experts models (or expert network mixture of experts models), rather than gating network mixture of experts models. Allowing the precision parameter(s) to depend on covariates represents a more appealing alternative to allowing the central sequence parameters to depend on covariates, given that the latter would necessitate estimation of $G \times T$ weighted multinomial logistic regressions. In so doing, there would be $G \sum_{t=1}^{T} (d+1)(v_t - 1)$ regression coefficients to estimate at each iteration — where v_t is number of states in the alphabet represented across all observations for the t-th time point and $(d+1)$

is the dimension of the associated design matrix (accounting also for the intercept) — which would be computationally infeasible for even moderately long sequences. In any case, however, it is usual in cluster analyses of life-course data to allow only gating network concomitants, e.g. the Markovian methods and latent class regression models considered as comparators in Section 3.5.2.

Another challenge in the area of clustering categorical sequences, which MEDseq models could feasibly be extended to address, are so called 'multichannel' sequences; i.e. situations where different sequences arise from multiple different domains on the same subjects. With regard to analyses of multichannel life-course sequences, the standard approach is again to apply heuristic or partitional clustering algorithms to a matrix of pairwise dissimilarities, with the difference in the multichannel scenario being that the dissimilarities for the whole data set are computed using a substitution cost matrix obtained by summing over the substitution costs derived for the individual channels (Pollock, 2007). Thus, the cost of a particular multichannel sequence changing to another is calculated by summing each of the relevant substitution costs. However, as this yields a $v \times v$ aggregate substitution cost matrix with state-specific entries, this approach to constructing a combined dissimilarity measure is not possible under the current MEDseq framework — under which the precision parameters are assumed to be constant with respect to pairs of states — because it would, again, render the normalising constant intractable. Hence, our alternative proposal is to rely on the local independence assumption, via

$$f\left(\mathbf{s}_i^1, \mathbf{s}_i^2, \ldots \mathbf{s}_i^M \mid \ldots\right) = \sum_{g=1}^{G} \tau_g(\mathbf{x}_i) \prod_{m=1}^{M} f(\mathbf{s}_i^m \mid \ldots),$$

whereby separate, tractable models based on the Hamming distance or one of its weighted variants are fitted to each of M channels, such that, for a given observation, its individual channels are independent given its component membership. For simplicity, the same set of precision parameter constraints could be employed across each channel-specific model. Relatedly, the local independence assumption could also be relied on to jointly cluster sequences and mixed-type covariates, as discussed in the MoEClust setting in Appendix 2.E.

Four penultimate comments relate to the IMIFA model family. Firstly, recall that versions which overfit the number of components are included, having either infinite or finite numbers of factors. Frühwirth-Schnatter and Malsiner-Walli (2019) show that such sparse finite mixtures also elicit a stick-breaking representation, truncated at G components. Thus, inferential tools used to estimate IMIFA and IMFA models such as the independent slice-efficient sampler and the adopted label-switching moves are feasible in the overfitted setting also. Frühwirth-Schnatter and Malsiner-Walli (2019) also draw a distinction between the number of non-empty clusters, which is of inferential interest, and the number of mixture components, characterising sparse finite mixtures as having G components and infinite mixtures as having infinitely many. This paper ultimately shows that sparse finite and infinite mixture models under a DP prior differ only in their construction of the mixing proportions. At least empirically, the comparability of the two model classes when the PYP prior is assumed is borne out by the applications in Chapter 4. In particular, matching the hyperpriors on the PYP parameters in the infinite mixture setting to the prior on the mixing proportions in the overfitted setting is shown to yield 'sparse' infinite mixtures. This helps mitigate against concerns regarding posterior consistency for the number of non-empty clusters in infinite mixtures assuming the PYP or DP priors.

A graphical justification for the extension to a PYP prior with $d \in [0, 1)$ and $\alpha > 0$ is provided in Appendix 4.E. However, De Blasi et al. (2015) describe an alternative formulation of the PYP with $d < 0$ and $\alpha = m|d|$, where m is a positive integer, and characterises this formulation as one which concentrates mass on a *finite* number of components such that the stick-breaking proportion $v_m = 1$. Miller and Harrison (2018) describe placing a prior on m in this scenario as equivalent to assuming a symmetric Dirichlet prior of variable dimension on the mixing proportions. Thus, this alternative formulation is of great interest in terms of bridging the gap between overfitted (i.e. sparse finite) mixtures and infinite mixtures.

Secondly, with respect to the variant of the PYP adopted here, two recently proposed sampling strategies, which exhibit superior mixing properties compared to the independent slice-efficient sampler employed, have potential utility in the IMIFA setting, namely the thresholded exchangeable slice sampler (Fall and Barat, 2014) and the importance conditional sampler (Canale et al., 2019). The former is partic-

ularly interesting as it overcomes the limitation of the stick-breaking prior not being invariant to the ordering of the cluster labels (Papaspiliopoulos and Roberts, 2008; Hastie et al., 2014). At present, the mixing proportions and their corresponding cluster-specific parameters are reordered at each iteration, such that the mixing proportions form a decreasing sequence, and label-switching moves are incorporated in order to improve mixing over the space of clustering labels. These steps could be avoided with the use of the exchangeable slice sampler. More simply, incorporating the threshold of Fall and Barat (2014) in the independent slice-efficient sampler, as they suggest themselves, is also of interest.

Thirdly, any potentially wider IMIFA model could not only include versions with different priors on the number of components but also versions underpinned by other infinite factor priors. A recent development in this area is provided by Srivastava et al. (2017), in which a multiscale generalised double Pareto prior is proposed. This elicits a soft-thresholding rule which estimates loadings entries in such a way that those with small magnitude are automatically set to zero. Recall that the MGP prior employed by the IMIFA model shrinks loadings entries arbitrarily close but not exactly to zero. Another particularly attractive candidate for consideration in the IMIFA setting is the cumulative shrinkage prior recently introduced by Legramanti et al. (2019). This prior effectively amounts to a spike-and-slab prior on the *a priori* variances of the loadings entries. It achieves shrinkage in distribution as the loadings' dimensionality increases, whereas the MGP prior only achieves shrinkage in expectation. The prior also decouples the parameters governing shrinkage and sparsity from the parameters governing the non-zero loadings entries, while the MGP does not. By virtue of achieving exact sparsity on the loadings, and by extension the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Psi}$, these alternative priors could be especially beneficial for the handwritten digits application, wherein the pixel representation of the data themselves is notably sparse. At present, neither prior has been generalised to the mixture setting, infinite or otherwise.

Fourthly, as per Roy et al. (2019), the prior on the factor scores, $\boldsymbol{\eta}_i \sim \mathrm{N}_q(\mathbf{0}, \boldsymbol{\mathcal{I}}_q)$, could be replaced by $\boldsymbol{\eta}_i \sim \mathrm{N}_q(\mathbf{0}, \mathbf{H})$, where $\mathbf{H}$ is a general diagonal matrix with non-identical entries following their own inverse gamma prior distributions, such that the latent factors are assumed to be heteroscedastic. This yields a covariance matrix decomposition of the form $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\mathbf{H}\boldsymbol{\Lambda}^\top + \boldsymbol{\Psi}$ and has the effect of removing the rota-

tional ambiguity in the loadings matrix, except for permutations, thereby removing the need for Procrustean post-processing to correct for non-identifiability. Moreover, Roy et al. (2019) demonstrate improved performance using heteroscedastic factors in the context of an IFA model, compared to an otherwise identical model with homoscedastic factors, in terms of more accurate estimation of $\mathbf{\Lambda}$, as well as the $\mathbf{\Sigma}$ matrix itself. In extending this approach to the mixture setting, however, it may be advantageous to allow $\mathbf{H}$ be cluster-specific, particularly for infinite factor models for which the MGP prior is assumed and the numbers of cluster-specific factors are adaptively truncated, via $\boldsymbol{\eta}_i \,|\, z_{ig} = 1 \sim \mathrm{N}_{\tilde{q}_g}(\mathbf{0}, \mathbf{H}_g)$, where $\mathbf{H}_g \overset{d}{=} \mathbf{H}_{g'}$.

Finally, an overarching limitation of this thesis is that all of the proposed parsimonious model families have only been considered in entirely unsupervised settings. Hence, a principal area of future research is to extend these models to supervised settings, where all observations are labelled according to the group to which they belong, as well as to semi-supervised settings, where only some proportion of observations are labelled. While scenarios in which observations are labelled are scarce in analyses of life-course sequences, model-based approaches to supervised or semi-supervised classification are feasible across all three model families.

Among other things, Appendix 2.D discusses various issues around predicting the class labels of unseen data using MoEClust models when new covariates are observed, with or without also observing associated new responses, under the familiar framework of designating the labelled observations as a training data set and the unlabelled observations as a test data set. In contrast, the model-based classification paradigm instead jointly models both the labelled and unlabelled data, while keeping the known labels fixed and estimating the unknown labels (McNicholas, 2010). By virtue of the component-specific parameters being learned from the entire data set, this approach often outperforms the training/test split approach from the point of view of classification accuracy. Furthermore, a model-based approach to predicting the unknown labels would certainly improve on the out-of-sample prediction approaches discussed in Appendix 2.D for those MoEClust model types which assume assignment dependence (i.e. those without gating concomitants). Hence, the extension to the semi-supervised setting is of particular interest for the MoEClust model family, given that only the training/test split approach is presently implemented in the associated `MoEClust` R package.

With regard to the IMIFA model family, we note that the Italian olive oil data analysed in Section 4.3.1 has previously been considered in a factor-analytic setting under an artificially constructed semi-supervised scenario (McNicholas, 2010). Impressive classification accuracy is achieved with even a moderate 50% level of supervision. We also note that the USPS digits data analysed in Section 4.3.3 comes with a hitherto unused test data set. While the MIFA model might appear to be a natural choice in a fully supervised analysis, given that the number of components is explicitly defined by the known group labels, the IMIFA model could be particularly useful in semi-supervised analyses. An issue which plagues many semi-supervised analyses, discussed at length in Cappozzo et al. (2019) is that the unlabelled data could suggest the presence of additional components. An infinite mixture model, with the assignments of the labelled data fixed, could in principle detect these extra components, if any, by virtue of allowing G to vary beyond the number of components implied by the known labels. By jointly modelling both the labelled and unlabelled data under an infinite factor mixture model (MIFA, OMIFA, and IMIFA), both sets of data would contribute to the estimation of the component-specific numbers of factors.

Overall, the parsimonious model-based clustering methods proposed in this thesis address some key limitations of the standard finite mixture model in ways that are evidently necessary and advantageous. Results of applications in each case appear promising and the diversity of each model family is conjectured to be flexibly adaptable to a wide range of situations. Finally, as shown in this concluding chapter, there are many possible extensions representing fertile grounds for future research which could potentially be incorporated into the published R packages.

References

Andrews, J. L. and P. D. McNicholas (2012). Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions: the tEIGEN family. *Statistics and Computing 22*(5), 1021–1029. 253

Biernacki, C., G. Celeux, J. Josse, and F. Laporte (2019). Dealing with missing data in model-based clustering through a MNAR model. In *CRoNos & MDA 2019 - Meeting and Workshop on Multivariate Data Analysis and Software*, Limassol, Cyprus. *HAL* pre-print, 02103347. 254

Bouveyron, C. and C. Brunet-Saumard (2014). Model-based clustering of high-dimensional data: a review. *Computational Statistics & Data Analysis 71*, 52–78. 251, 252

Canale, A., R. Corradin, and B. Nipoti (2019). Importance conditional sampling for Bayesian nonparametric mixtures. *arXiv* pre-print, 1906.08147. 257

Cappozzo, A., F. Greselin, and T. B. Murphy (2019). A robust approach to model-based classification based on trimming and constraints. *Advances in Data Analysis and Classification*, 1–28. URL https://doi.org/10.1007/s11634-019-00371-w. 260

De Blasi, P., S. Favaro, A. Lijoi, R. H. Mena, I. Prünster, and M. Ruggiero (2015). Are Gibbs-type priors the most natural generalization of the Dirichlet process? *IEEE Transactions on Pattern Analysis and Machine Intelligence 37*(2), 212–229. 257

Fall, M. D. and É. Barat (2014). Gibbs sampling methods for Pitman-Yor mixture models. *HAL* pre-print, 00740770v2. 257, 258

Fokoué, E. (2005). Mixtures of factor analyzers: an extension with covariates. *Journal of Multivariate Analysis 95*(2), 370–384. 253

Fop, M. and T. B. Murphy (2018). Variable selection methods for model-based clustering. *Statistics Surveys 12*, 18–65. 251

Frühwirth-Schnatter, S. and G. Malsiner-Walli (2019). From here to infinity: sparse finite versus Dirichlet process mixtures in model-based clustering. *Advances in Data Analysis and Classification 13*(1), 33–63. 257

Gabadinho, A., G. Ritschard, N. S. Müller, and M. Studer (2011). Analyzing and visualizing state sequences in R with TraMineR. *Journal of Statistical Software 40*(4), 1–37. 254

Galimberti, G. and G. Soffritti (2019). Seemingly unrelated clusterwise linear regression. *Advances in Data Analysis and Classification*, 1–26. URL https://doi.org/10.1007/s11634-019-00369-4. 249

García-Escudero, L. A., A. Gordaliza, F. Greselin, S. Ingrassia, and A. Mayo-Iscar (2018). Eigenvalues and constraints in mixture modeling: geometric and computational issues. *Advances in Data Analysis and Classification 12*(2), 203–233. 249

Gebru, I. D., X. Alameda-Pineda, F. Forbes, and R. Horaud (2016). EM algorithms for weighted-data clustering with application to audio-visual scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 38*(12), 2402–2415. 255

Geweke, J. and M. Keane (2007). Smoothly mixing regressions. *Journal of Econometrics 138*(1), 252–290. 250

Halpin, B. (2016). Multiple imputation for categorical time series. *The Stata Journal 16*(3), 590–612. 254

Hastie, D. I., S. Liverani, and S. Richardson (2014). Sampling from Dirichlet process mixture models with unknown concentration parameter: mixing issues in large data implementations. *Statistics and Computing 25*(5), 1023–1037. 258

Ingrassia, S. and A. Punzo (2019). Cluster validation for mixtures of regressions via the total sum of squares decomposition. *Journal of Classification*, 1–22. URL https://doi.org/10.1007/s00357-019-09326-4. 250

Legramanti, S., D. Durante, and D. B. Dunson (2019). Bayesian cumulative shrinkage for infinite factorizations. *arXiv* pre-print, 1902.04349. 258

Leisch, F. (2008). Modelling background noise in finite mixtures of generalized linear regression models. In P. Brito (Ed.), *COMPSTAT2008*. Physica-Verlag HD. 249, 252

McNicholas, P. D. (2010). Model-based classification using latent Gaussian mixture models. *Journal of Statistical Planning and Inference 140*(5), 1175–1181. 259, 260

McNicholas, P. D. and T. B. Murphy (2008). Parsimonious Gaussian mixture models. *Statistics and Computing 18*(3), 285–296. 248, 253

Miller, J. W. and M. T. Harrison (2018). Mixture models with a prior on the number of components. *Journal of the American Statistical Association 113*(521), 340–356. 257

O'Hagan, A., T. B. Murphy, L. Scrucca, and I. C. Gormley (2019). Investigation of parameter uncertainty in clustering using a Gaussian mixture model via jackknife, bootstrap and weighted likelihood bootstrap. *Computational Statistics 34*(4), 1779–1813. 250

Papaspiliopoulos, O. and G. O. Roberts (2008). Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika 95*(1), 169–186. 258

Pollock, G. (2007). Holistic trajectories: a study of combined employment, housing and family careers by using multiple-sequence analysis. *Journal of the Royal Statistical Society: Series A (Statistics in Society) 170*(1), 167–183. 256

Pouhramadi, M. (1999). Joint mean-covariance models with applications to longitudinal data: unconstrained parameterisation. *Biometrika 86*(3), 677–690. 248

Roy, A., I. Lavine, A. H. Herring, and D. B. Dunson (2019). Perturbed factor analysis: improving generalizability across studies. *arXiv* pre-print, 1910.03021. 258, 259

Scrucca, L. and A. E. Raftery (2018). clustvarsel: a package implementing variable selection for Gaussian model-based clustering in R. *Journal of Statistical Software 84*(1), 1–28. 251

Srivastava, S., B. E. Engelhardt, and D. B. Dunson (2017). Expandable factor analysis. *Biometrika 104*(3), 649–663. 258

Tipping, M. E. and C. M. Bishop (1999). Mixtures of probabilistic principal component analyzers. *Neural Computation 11*(2), 443–482. 248

Xu, J. and G. MacKenzie (2012). Modelling covariance structure in bivariate marginal models for longitudinal data. *Biometrika 99*(3), 649–662. 248