
University of Washington
Working Group on Model-Based Clustering
30th Summer Session

Università degli Studi di Bologna
Bertinoro, July 22–26, 2024



Organized by

Charles Bouveyron, Gilles Celeux, Bettina Grün, T. Brendan Murphy, Rebecca Nugent,
Adrian E. Raftery, Luca Scrucca, Angela Montanari, Cinzia Viroli, and Laura Anderlucci

WORKING GROUP BOOK

DO NOT CITE OR DISTRIBUTE ANY UNPUBLISHED MATERIAL

Contents

Program	3
Michael Fop	4
Daniele Durante	32
Laura Anderlucci	96
Julien Jaques	97
Qing Mai	190
Gertraud Malsiner-Walli	201
Angela Montanari	222
Adrian Raftery	232
Pierre Latouche	261
Christian Hennig	330
Keefe Murphy	359
Brendan Murphy	449
Dimitris Karlis	489
Sylvia Frühwirth-Schnatter	510
Aaron Molstad	557
Luca Scrucca	604
Alessandro Casa	633
Claire Gormley	693
Poster session	746
Software session	746

Program

Day (Chair)	Time	Speaker	Title
Monday (A. Montanari)	09:00-10:20	Michael Fop University College Dublin	Model-based clustering of networks with compositional edges
	10:40-12:00	Daniele Durante Università Bocconi	Bayesian nonparametric stochastic block modeling of criminal networks
Tuesday (A. Raftery)	09:00-10:20	Laura Anderlucci Università degli Studi di Bologna	Model-based clustering of high dimensional data via Random Projections
	10:40-10:55	Julien Jacques Université de Lyon	Mixture of regression for functional data
	11:00-11:15	Qing Mai Florida State University	Model-based low-rank tensor clustering
	11:20-11:35	Gertraud Malsiner-Walli WU Vienna University	Multivariate mixed type longitudinal data and model-based clustering
	11:40-11:55	Angela Montanari Università degli Studi di Bologna	The written Italian of university students: identification of tendencies via mixtures of GLLVM for count data
	14:00-16:00	Poster session	
Wednesday (B. Murphy)	9:00-10:20	Adrian Raftery University of Washington	Marginal likelihoods for model selection in mixture models from MCMC using the THAMES estimator
	10:40-10:55	Pierre Latouche Université Clermont Auvergne	The deep-latent position block model for network visualisations compatible with block structures
	11:00-11:15	Christian Hennig Università degli Studi di Bologna	Nonparametric consistency for maximum likelihood estimation and clustering based on mixtures of elliptically-symmetric distributions
	11:20-11:35	Keefe Murphy Maynooth University	Automatic mixtures of regularised experts
	11:40-11:55	Bettina Grün WU Vienna University	CLIPS - Clustering in the parameter space
	14:00-16:00	Software session	
Thursday (B. Grün)	09:00-10:20	Brendan Murphy University College Dublin	An unsupervised record linkage approach using household information to enhance individual matching across different databases
	10:40-10:55	Dimitris Karlis Athens University of Economics and Business	Dyadic multivariate mixture models for the analysis of elite swimmers
	11:00-11:15	Sylvia Frühwirth-Schnatter WU Vienna University	A Bayesian approach toward mixtures of factor analyser where the number of clusters and factors is unknown
	11:20-11:35	Aaron Molstad University of Minnesota	Kernelized discriminant analysis for multivariate categorical response regression
	11:40-11:55	Luca Scrucca Università degli Studi di Perugia	A model-based approach to shot charts estimation in basketball
Friday (L. Scrucca)	09:00-10:20	Alessandro Casa Università di Bergamo	Sparse model-based clustering of three-way data via lasso-type penalties
	10:40-12:00	Claire Gormley University College Dublin	Constrained model-based clustering for hyperspectral images

Michael Fop

Material list:

Promskaia I., O'Hagan A., and Fop M. (2024) A Dirichlet stochastic block model for composition-weighted networks.
Unpublished manuscript.

A Dirichlet stochastic block model for composition-weighted networks

Iuliia Promskaia^{1,2}, Adrian O'Hagan^{1,2}, and Michael Fop²

¹*SFI Insight Centre for Data Analytics, Dublin, Ireland*

²*School of Mathematics and Statistics, University College Dublin, Dublin, Ireland*

iuliia.promskaia@insight-centre.org

[ORCID: 0009-0009-8704-2046](https://orcid.org/0009-0009-8704-2046)

Abstract

Network data are observed in various applications where the individual entities of the system interact with or are connected to each other, and very often these interactions are defined by their associated strength or importance. Clustering is a common task in network analysis that involves finding groups of nodes that display similarities in the way they interact with the rest of the network. However, most clustering methods use the strengths of connections between entities in their original form, ignoring the possible differences in the capacities of individual nodes to send or receive edges. This often leads to clustering solutions that are heavily influenced by the nodes' capacities. One way to overcome this is to analyse the strengths of connections in relative rather than absolute terms, expressing each edge weight as a proportion of the sending (or receiving) capacity of the respective node. This, however, induces additional modelling constraints that most existing clustering methods are not designed to handle. In this work we propose a stochastic block model (SBM) for composition-weighted networks based on direct modelling of compositional, i.e. proportional, weight vectors using a Dirichlet mixture, with the parameters determined by the cluster labels of the sender and the receivers. We address the inference problem via an extension of the classification expectation-maximisation algorithm that uses a working independence assumption, to express the complete data likelihood of each node of the network as a function of fixed cluster labels of the remaining nodes. A model selection criterion is derived to aid the choice of the number of clusters. An alternative approach to clustering in composition-weighted networks based on a mapping to Euclidean space is also provided. The Dirichlet SBM (DirSBM) is validated using a number of simulation studies, assessing the effect of various initialisation strategies on the model's performance, latent structure recovery, parameter estimation quality and model selection, and is applied to real-world data sets, namely the Erasmus network data and the London bike sharing network data.

Keywords Compositional data, hybrid likelihood, statistical network analysis, stochastic block model, weighted networks

Acknowledgements This work was supported by the Science Foundation Ireland funded Insight Centre for Data Analytics (Grant number 12/RC/2289_P2).

1 Introduction

Networks are often used to represent relationships between entities, and in the recent past there has been a rapid increase in the collection and availability of such relational data. Network data appear in a diverse range of applications, such as the social sciences (Rastelli and Fop, 2020), biology (Daudin, 2011), transportation (Ng and Murphy, 2021), and finance (Hledik and Rastelli, 2023). Given the wide availability of network data, the statistical analysis and modelling of networks is an active area of research, driven by the unique challenges that these data present, largely due to the dependence between entities. For a concise overview of the main models and methods for network data see Salter-Townshend et al. (2012).

One common task in network analysis is clustering, which involves finding groups of nodes that share similar connectivity patterns or have similar characteristics. The two main families of

approaches to clustering in networks are algorithm-based and model-based (Leger et al., 2013). Algorithm-based methods, as the name suggests, involve a set of operations that partition the network into clusters. These methods are often based on some notion of distance or similarity between nodes (Pons and Latapy, 2006), spectral properties of the Laplacian matrix (von Luxburg, 2007), or an optimisation procedure (Guimerà et al., 2010; Clauset et al., 2005). Model-based methods assume the presence of sub-populations in the data, corresponding to the clusters, which are then modelled by means of probability distributions. As opposed to algorithm-based methods, model-based clustering approaches have the advantage of providing a principled framework where the clustering task is framed as an inferential task for a generative probability model (Daudin, 2011; Bouveyron et al., 2019).

Stochastic block modelling is a popular model-based clustering framework for network data, which models the connectivity patterns between pairs of nodes, assuming their stochastic equivalence: the presence and the strength of a connection between two nodes is determined by their cluster membership rather than by the nodes themselves. The definition of a stochastic block model as a way to describe the a priori community structure in a network was introduced in Holland et al. (1983) and Wang and Wong (1987), and the first attempt to use this model as a clustering tool was made in Wasserman and Anderson (1987).

The stochastic block model (SBM) was originally developed for the analysis and clustering of binary network data. For this type of data, binary interactions are often defined on a pre-specified threshold or on a subjective evaluation of what constitutes an interaction between two nodes, resulting in a potential information loss. With network data becoming more widespread and complex in the recent years, a number of extensions of the SBM to networks with different edge types and characteristics have been proposed. Particular efforts have been focused on clustering models for weighted networks. For instance, Nowicki and Snijders (2001) extended the SBM to allow for finite set valued edges, such as those taking values in $\{-1, 0, 1\}$. Count edge attributes or multiple edges between pairs of nodes were considered in El Haj et al. (2022), Karrer and Newman (2011), Zanghi et al. (2010), and Mariadassou et al. (2010). To cluster nodes in networks with continuous edge weights, Ng and Murphy (2021) propose a weighted SBM based on Gamma distributions. More general frameworks encompassing a wider range of edge types and attribute distributions have been proposed by Aicher et al. (2013, 2014), and Ludkin (2020). We point the reader to Lee and Wilkinson (2019) for a comprehensive review of extensions of the SBM for different data types as well as inferential approaches. An approach related to weighted SBMs is the one introduced by Melnykov et al. (2020). In Melnykov et al. (2020), the authors propose a general framework for performing model-based clustering on multi-layer networks. The approach uses a multivariate Gaussian distribution to model the weights, allowing the correlation between weights connecting pairs of nodes as well as between edge weights across different layers to be accounted for. This work can be seen as a generalisation of a weighted SBM (Aicher et al., 2013, 2014; Ng and Murphy, 2021) that takes into account the correlation structure between the weights.

All of the aforementioned approaches propose models for the analysis of the weights in their original form. However, in some applications, in particular those modelling flows in the network, such as transportation or trade, the observed edge weights are influenced by the nodes' capacities to send and receive edges, which can differ significantly across the network. This often results in clustering solutions that mimic the patterns related to the capacities of the sending and receiving nodes, rather than the weights connecting them. Taking as an example the Erasmus programme network presented in Section 6.1, suppose we are interested in grouping European countries based on similarities in students' preferences for exchange destinations. With the participating countries having very different populations, we expect larger countries, such as Germany, France and the UK, to have larger student populations and so larger volumes of students participating in the programme. These countries are also able to accommodate more incoming students. By contrast, the Nordic countries, like Denmark and Finland, are much smaller in population, and hence they send and receive significantly fewer students. If we were to perform clustering using the raw counts of students going on Erasmus exchange between these countries without taking the differences in population into account, the countries would be roughly divided into groups based on their population sizes, i.e. Germany, France and the UK as one cluster, and Denmark and Finland as another, due to differences in the scale of the weights. Even if the preference profile of the UK students is closer to those of the smaller Nordic countries, the volume of students the country sends is far too large for it to be assigned to the same cluster as Finland and Denmark. Similar issue can arise in flight networks, with small airports only being able to service a limited number of flights per day as opposed to international hubs with multiple runways, or international trade networks, where the value of goods traded is linked to country's GDP (Melnykov et al., 2020). The differences in the magnitudes of edge weights can significantly affect the results, making it more

challenging to draw conclusions.

One way to address this is by using relative rather than absolute weights in the network. In the aforementioned cases, this can be done by calculating proportions of flow alongside each edge with respect to a sending (or receiving) node, leading to edge weights of relative size. In other applications, the absolute weights can simply be unavailable for the analysis as the interactions between nodes are measured in relative terms to begin with. For example, in epidemiological networks the edge weights could represent probabilities of interaction between agents, or in infrastructure networks they could correspond to proportions of liquid flowing between units. Networks with relative edge weights can also arise as a result of privacy protection policies, such as in [Hledík and Rastelli \(2023\)](#), where the absolute information could potentially be used to identify individual actors, banks in this case.

The main challenge associated with the relative nature of edge weights in the network is their interdependence. Proportional data imposes a constant-sum constraint, making the standard tools used for networks with independent weights invalid. To appropriately address this constraint, tools from compositional data analysis can be employed. Compositional data describe parts of some whole, such as percentages, probabilities, and proportions, and they can be naturally observed or constructed from other types of data, such as continuous or count data, by dividing individual components by their sum. There are two main families of approaches to compositional data analysis. The first involves mapping the compositions onto the real line, allowing one to use the standard machinery on the transformed data. Log-ratio transformations are a popular choice in the literature due to their sound theoretical properties, and other alternatives, such as standardisation, are also sometimes used in practice ([Aitchison, 1986](#); [Baxter, 1995](#)). The second approach is based on direct modelling of compositional observations, which often uses the Dirichlet distribution to model the compositions in their original form ([Pal and Heumann, 2022](#); [Tsagris and Stewart, 2018](#)). Both approaches have their relative benefits, such as the scale-invariance of the log-ratio transformations, or desirable distributional properties that come with direct modelling, and so the choice of the most suitable approach is context dependent. For the discussion of the main methods and challenges in compositional data analysis, we suggest the review paper by [Alenazi \(2023\)](#).

The literature on models and methods for networks with weights of relative nature is very scarce, with most works addressing unique application-specific challenges, often related to biological networks. Examples of such works are [Yuan et al. \(2019\)](#) and [Ha et al. \(2020\)](#), where the centered log-ratio transformation is applied to the compositional data to eliminate the constant-sum constraint in an attempt to recover the microbiome interaction network. Another notable work by [Hledík and Rastelli \(2023\)](#), which is driven by a financial market application, uses direct modelling of proportions via the Dirichlet distribution and proposes a dynamic latent variable modelling framework for the analysis of interbank networks.

To the best of our knowledge, there have been no attempts to develop a general model-based clustering framework for composition-weighted networks. In this work, we propose an extension of the stochastic block model that utilises the relative rather than absolute strength of connections, by leveraging ideas from compositional data analysis. We model the set of edge weights from each node using a Dirichlet distribution, the parameters of which are determined by the cluster labels of the sender and each of the receivers. For inference, we consider the hybrid maximum likelihood approach developed by [Marino and Pandolfi \(2022\)](#), based on [Daudin et al. \(2008\)](#), using a variant of the classification expectation-maximisation (EM) algorithm ([Celeux and Govaert, 1992](#)). To address model selection, an integrated completed likelihood (ICL) criterion is derived ([Daudin et al., 2008](#)). We also briefly introduce an alternative approach to cluster nodes in composition-weighted networks by making use of the log-ratio transformation that maps compositional data onto the real line ([Aitchison, 1982](#)). The code implementing the modelling framework in R ([R Core Team, 2019](#)) is available on [GitHub](#).

The structure of this paper is as follows: we start by describing the SBM with compositional Dirichlet-distributed edge weights in Section 2, and outline the inferential procedure based on a classification EM algorithm in Section 3; Section 4 presents an alternative method to cluster nodes in the network with compositional edge weights by using the log-ratio transformation of the compositions and then utilising a standard weighted stochastic block model; we test the model performance on synthetic data in Section 5, assessing different initialisation strategies, parameter estimation quality, and clustering and model selection performance, also in comparison to alternative approaches; we then illustrate the use of the proposed model in application to the Erasmus programme data from [Gadár et al. \(2020\)](#) and the London bike sharing data from [Transport for London \(2016\)](#) in Section 6; Section 7 concludes the paper with a discussion, highlighting the work's impact and the potential for further developments.

2 Dirichlet stochastic block model

This section introduces the Dirichlet stochastic block model (DirSBM) for networks with compositional edge weights.

Let \mathbf{Y} be the weighted adjacency matrix of a directed network on n nodes with no self-loops with strictly positive entries, i.e. $y_{ij} > 0$ for all $i \neq j$ and $y_{ii} = 0$ for all i . Define the compositional counterpart of \mathbf{Y} , denoted \mathbf{X} , whose row entries \mathbf{x}_i are given by

$$\mathbf{x}_i = \left(\frac{y_{i1}}{\sum_{j \neq i} y_{ij}}, \frac{y_{i2}}{\sum_{j \neq i} y_{ij}}, \dots, \frac{y_{i(i-1)}}{\sum_{j \neq i} y_{ij}}, 0, \frac{y_{i(i+1)}}{\sum_{j \neq i} y_{ij}}, \dots, \frac{y_{in}}{\sum_{j \neq i} y_{ij}} \right),$$

i.e. $x_{ij} > 0$ for $i \neq j$ and $\sum_l x_{il} = 1 \forall i$. Note that $x_{ii} = 0$ is due to the absence of self-loops in the original weighted network. Denote with \mathbf{x}_i^* a subvector of \mathbf{x}_i that excludes the zero entry x_{ii} , and with \mathbf{X}^* the matrix formed by row vectors $\mathbf{x}_i^*, i = 1, \dots, n$. In practice, in some applications, the weighted adjacency matrix \mathbf{Y} may be unavailable, with the compositional matrix \mathbf{X} being the only piece of data provided. The Dirichlet stochastic block model with K blocks assumes the following generative process for a network with compositional edge weights \mathbf{X}^* :

1. Given a K -dimensional vector of cluster membership proportions $\boldsymbol{\theta}$, generate binary cluster allocations $\mathbf{z}_i \sim \text{Multinom}(1, \boldsymbol{\theta})$, for $i = 1, \dots, n$. The entries of these vectors, i.e. $z_{ik} = 1$ when node i is assigned to cluster C_k and 0 otherwise. Denote with \mathbf{Z} an $n \times K$ matrix of binary cluster allocations of all nodes in the network, and with \mathbf{Z}_{-i} the $(n-1) \times K$ -dimensional matrix of cluster allocations of all nodes in the network that are not i .
2. Let $\mathbf{A} = \{\alpha_{kh}\}_{k,h=1}^K$ be a $K \times K$ matrix with strictly positive values, corresponding to the Dirichlet distribution concentration parameters. Given the cluster allocations, generate $(n-1)$ -dimensional compositional observations as

$$\mathbf{x}_i^* | \mathbf{z}_i, \mathbf{Z}_{-i} \sim \text{Dir}(\mathbf{z}_i \mathbf{A} \mathbf{Z}_{-i}^\top),$$

for $i = 1, \dots, n$.

To aid understanding, consider the vector \mathbf{c} denoting the cluster labels of the nodes in the network, such that $c_i = k$ if i is a member of cluster C_k . The above distribution can also be written as

$$\mathbf{x}_i^* | \mathbf{c} \sim \text{Dir}(\alpha_{c_i c_1}, \dots, \alpha_{c_i c_{i-1}}, \alpha_{c_i c_{i+1}}, \dots, \alpha_{c_i c_n}). \quad (1)$$

The intuition behind this formulation is as follows: for any sender i , the distribution of weights of the edges connecting it to other vertices depends on the cluster memberships of i itself and of all of the receiver nodes $j \neq i$. If the sender i and the receiver j are members of the same cluster C_k , the corresponding parameter of the Dirichlet distribution is α_{kk} , and this parameter value is shared for all such receivers $j \in C_k$. If instead the receiver node is in a different cluster, $j \in C_h$, the corresponding parameter is α_{kh} .

2.1 Properties of DirSBM

The model statement in Section 2 implies that the distribution of weight vectors conditional on cluster labels in the whole network is shared among the nodes assigned to the same cluster, up to a permutation of the ordering of the nodes, which is assumed to be invariant for simplicity.

To illustrate this, consider, for example, the distribution of compositional weights of node $i \in C_1$ and rearrange the order of the dimensions so that they are sorted by cluster assignment index in ascending order for ease of notation. Every vector of compositional weights of node i belonging to cluster C_1 , denoted $\mathbf{x}_{i \in C_1}^*$, conditional on the cluster assignments of all the remaining $(n-1)$ nodes, is distributed as

$$\mathbf{x}_{i \in C_1}^* | \mathbf{Z} \sim \text{Dir}(\underbrace{\alpha_{11}, \dots, \alpha_{11}}_{(n_1 - 1) \text{ times}}, \underbrace{\alpha_{12}, \dots, \alpha_{12}}_{n_2 \text{ times}}, \dots, \underbrace{\alpha_{1h}, \dots, \alpha_{1h}}_{n_h \text{ times}}, \dots, \underbrace{\alpha_{1K}, \dots, \alpha_{1K}}_{n_K \text{ times}}),$$

where n_h denotes the membership count of cluster C_h . This is because any member of cluster C_1 is connected to the remaining $(n_1 - 1)$ members of its own cluster, all n_2 members of cluster C_2 , and so on.

Generalising for a generic node in cluster C_k , $k = 1, \dots, K$, we obtain:

$$\mathbf{x}_{i \in C_k}^* | \mathbf{Z} \sim \text{Dir}(\underbrace{\alpha_{k1}, \dots, \alpha_{k1}}_{n_k \text{ times}}, \dots, \underbrace{\alpha_{kk}, \dots, \alpha_{kk}}_{(n_k - 1) \text{ times}}, \dots, \underbrace{\alpha_{kh}, \dots, \alpha_{kh}}_{n_h \text{ times}}, \dots, \underbrace{\alpha_{kK}, \dots, \alpha_{kK}}_{n_K \text{ times}}). \quad (2)$$

Therefore, all the nodes belonging to the same cluster are identically distributed and share the same Dirichlet distribution, whose parameter vector is determined by the cluster allocations of the remaining nodes. This property of the DirSBM can be seen as a counterpart of stochastic equivalence in other stochastic block models (Lee and Wilkinson, 2019; Nowicki and Snijders, 2001; Snijders and Nowicki, 1997). The nodes in the network are called stochastically equivalent if they have the same probability of being connected by an edge to any other node in the network, conditional on their cluster labels. In weighted SBMs, such as the ones proposed by Aicher et al. (2014) and Ng and Murphy (2021), this leads to identical distributions of edge weights for any pair of nodes $i \in C_k$ and $j \in C_h$ as the parameters of such distributions are only determined by the cluster labels of nodes. The DirSBM cannot be said to strictly exhibit stochastic equivalence as it does not model the probability of edge existence between nodes, but it does share the benefits that arise as a consequence of stochastic equivalence as the identical distribution property of a set of edge weights conditional on cluster labels still holds.

2.1.1 Interpretation of parameters

The parameters of the Dirichlet distribution in their original form are usually difficult to interpret as they are not measured on the same scale as the compositional data. To relate the parameter values to the original compositions, we can compute the expected proportion sent from any node $i \in C_k$ to any node $j \in C_h$, denoted w_{kh} . In our case, as the parameter values are shared among the pairs of nodes that belong to the same cluster pair, this expected value is given by:

$$w_{kh} = \mathbb{E}[X_{ij}^*] = \frac{\alpha_{kh}}{(n_k - 1)\alpha_{kk} + \sum_{g \neq k} n_g \alpha_{kg}} \quad i \in C_k, j \in C_h. \quad (3)$$

We can use these values to construct a $K \times K$ matrix of expected exchange proportions, denoted \mathbf{W} , that is more intuitive to interpret than the original Dirichlet parameter matrix \mathbf{A} . Note that, although the entries of this matrix are proportions, the rows do not sum to 1, since these proportions are shared among all pairs of nodes with the same cluster labels.

It may also be insightful to learn the expected total shares of exchanges between clusters. To calculate these, we make use of the aggregation property of the Dirichlet distribution, which states that if the parts of the Dirichlet observation are added together, the distribution remains Dirichlet, with the updated parameter values being equal to the sum of the corresponding original values (Frigyik et al., 2010). Using Equation (2), we aggregate the parts from the same cluster pair, resulting in a compositional vector

$$\mathbf{q}_i|\mathbf{Z} = \left(\sum_{\substack{j \neq i \\ j \in C_1}} x_{ij}, \dots, \sum_{\substack{j \neq i \\ j \in C_k}} x_{ij}, \dots, \sum_{\substack{j \neq i \\ j \in C_K}} x_{ij} \right) \sim Dir(n_1\alpha_{k1}, \dots, (n_k - 1)\alpha_{kk}, \dots, n_K\alpha_{kK}),$$

for $i \in C_k$. Hence, the expected cluster-to-cluster exchange shares, denoted with v_{kh} , are given by:

$$v_{kh} = \mathbb{E}[q_{il}] \begin{cases} \frac{(n_k - 1)\alpha_{kk}}{(n_k - 1)\alpha_{kk} + \sum_{g \neq k} n_g \alpha_{kg}} & \text{if } h = k, \\ \frac{n_h \alpha_{kh}}{(n_k - 1)\alpha_{kk} + \sum_{g \neq k} n_g \alpha_{kg}} & \text{if } h \neq k. \end{cases} \quad (4)$$

The resulting matrix $\mathbf{V} = \{v_{kh}\}_{k,h=1}^K$ exhibits a unit-row-sum property.

3 Inference

The distribution of a set of edge weights \mathbf{X} , conditional on the cluster labels is

$$p(\mathbf{X}|\mathbf{Z}) = \prod_{i=1}^n p(\mathbf{x}_i|\mathbf{z}_i, \mathbf{Z}_{-i}) = \prod_{i=1}^n \prod_{k=1}^K \left[\frac{\Gamma(\sum_{j \neq i}^n \alpha_j)}{\prod_{j \neq i}^n \Gamma(\alpha_j)} \prod_{j \neq i}^n x_{ij}^{\alpha_j - 1} \right]^{z_{ik}}, \quad \text{where } \alpha_j = \sum_{h=1}^K z_{jh} \alpha_{kh}.$$

This is a product of probability densities of $(n - 1)$ -dimensional Dirichlet distributions with the respective parameter vectors determined by the cluster label of the sender node i and of the remaining nodes in the network.

As usual in model-based clustering, inference proceeds by maximizing the marginal log-likelihood of the data \mathbf{X} with respect to the model parameters. However, this quantity involves summing over the set of all possible cluster allocations and is computationally intractable (Daudin et al., 2008).

A popular algorithm used in model-based clustering is the expectation-maximisation (EM) algorithm (Dempster et al., 1977). It is employed to find the maximum of the marginal log-likelihood of the data by introducing latent variables corresponding to cluster labels and defining the joint log-likelihood of the data and the cluster labels, the complete data log-likelihood. Then, inference for the model consists of estimation of parameters and of the latent cluster assignments. The EM algorithm works by iteratively finding the expectation of the complete data log-likelihood with respect to the posterior distribution of the latent variables (E-step) and maximising this expectation with respect to the model parameters (M-step).

In the case of the DirSBM, the complete data log-likelihood is:

$$\begin{aligned} l_c(\mathbf{A}, \boldsymbol{\theta}) &= \log p(\mathbf{X}, \mathbf{Z}) \\ &= \sum_{i=1}^n \sum_{k=1}^K z_{ik} \left[\log \Gamma\left(\sum_{j \neq i}^n \alpha_j\right) - \sum_{j \neq i}^n \log \Gamma(\alpha_j) + \sum_{j \neq i}^n (\alpha_j - 1) \log x_{ij} \right] + \sum_{i=1}^n \sum_{k=1}^K z_{ik} \log \theta_k. \end{aligned} \quad (5)$$

Further details are provided in Appendix A.

In stochastic block modelling, the E-step is often computationally intractable due to the fact that the latent variables are not independent a posteriori. To overcome this problem, the variational EM is often employed for inference in SBMs, which is based on an approximation of the posterior distribution of the latent variables (Daudin et al., 2008).

3.1 Hybrid log-likelihood

In the DirSBM, a new challenge arises when trying to employ the variational EM for inference, as is standard in SBMs. This is due to the fact that, in order to derive the evidence lower bound (Daudin et al., 2008) one is required to compute the expectation:

$$\mathbb{E}_{\mathbf{Z} \sim q(\mathbf{Z})} \left[\log \Gamma\left(\sum_{j \neq i}^n \sum_{h=1}^K z_{jh} \alpha_{kh}\right) - \sum_{j \neq i}^n \log \Gamma\left(\sum_{h=1}^K z_{jh} \alpha_{kh}\right) \right],$$

with respect to $q(\mathbf{Z})$, the mean field approximation of the posterior distribution of the latent cluster allocations. This expectation is intractable, because, unlike in standard SBM, even conditional on their cluster labels, the distribution of node i is not independent of that of node j . In fact, due to the model formulation in Equation (5), one needs to know the cluster assignments of all the receiving nodes contained in \mathbf{Z}_{-i} to define the distribution of \mathbf{x}_i given \mathbf{z}_i .

For this reason, we employ the inferential approach based on a hybrid log-likelihood proposed by Marino and Pandolfi (2022). The approach uses a working independence assumption, whereby each of the latent variables corresponding to the cluster label c_i is seen as a function of fixed values of the remaining cluster labels \mathbf{c}_{-i} , allowing factorization of the likelihood over the nodes. The observed hybrid log-likelihood, using the notation from Equation (1), is given by

$$l^{hyb}(\mathbf{A}, \boldsymbol{\theta}) = \sum_{i=1}^n \log \left(\sum_{k=1}^K \theta_k p(\mathbf{x}_i | c_i = k, \mathbf{c}_{-i} = \tilde{\mathbf{c}}_{-i}) \right)$$

where the notation $\tilde{\mathbf{c}}_{-i}$ is employed to indicate the fixed nature of the cluster labels of the remaining $(n - 1)$ nodes in the network when considering the contribution of node i to the log-likelihood.

Introducing the set of latent variables \mathbf{Z} , we arrive at the complete data hybrid log-likelihood:

$$\begin{aligned} l_c^{hyb}(\mathbf{A}, \boldsymbol{\theta}) &= \sum_{i=1}^n \sum_{k=1}^K z_{ik} \log p(\mathbf{x}_i | z_{ik} = 1, \mathbf{Z}_{-i} = \tilde{\mathbf{Z}}_{-i}) + \sum_{i=1}^n \sum_{k=1}^K z_{ik} \log \theta_k \\ &= \sum_{i=1}^n \sum_{k=1}^K z_{ik} \left[\log \Gamma\left(\sum_{j \neq i}^n \tilde{\alpha}_j\right) - \sum_{j \neq i}^n \log \Gamma(\tilde{\alpha}_j) + \sum_{j \neq i}^n (\tilde{\alpha}_j - 1) \log x_{ij} \right] + \sum_{i=1}^n \sum_{k=1}^K z_{ik} \log \theta_k, \end{aligned} \quad (6)$$

where $\tilde{\mathbf{Z}}$ is a binary matrix with entries $\tilde{z}_{jk} = \mathbb{1}\{\tilde{c}_i = k\}$ and $\tilde{\alpha}_j = \sum_{h=1}^K \tilde{z}_{jh} \alpha_{kh}$.

With the complete data hybrid log-likelihood, a working independence assumption allows one to implement a variant of the classification EM algorithm (Celeux and Govaert, 1992), where the hard partition of the nodes that are not i is used to compute the cluster assignment probability for i . The steps of the algorithm are outlined below.

E-step: compute the probability of assignment of node i to cluster k at iteration t using

$$\hat{z}_{ik}^{(t)} = \widehat{\Pr}(z_{ik} = 1 | \mathbf{x}_i, \tilde{\mathbf{Z}}_{-i}^{(t-1)}) \propto \hat{\theta}_k^{(t-1)} \frac{\Gamma(\sum_{j \neq i}^n \tilde{\alpha}_j^{(t-1)})}{\prod_{j \neq i}^n \Gamma(\tilde{\alpha}_j^{(t-1)})} \prod_{j \neq i}^n x_{ij}^{\tilde{\alpha}_j^{(t-1)}},$$

with $\tilde{\alpha}_j^{(t-1)} = \sum_{h=1}^K \tilde{z}_{jh}^{(t-1)} \hat{\alpha}_{kh}^{(t-1)}$. The quantity on the right hand side is normalised to fulfill the constraint $\sum_{k=1}^K \hat{z}_{ik}^{(t)} = 1$.

C-step: following Marino and Pandolfi (2022), a greedy approach to the C-step is adopted. That is, for each node $i = 1, \dots, n$, every label swap is considered and i is assigned to the cluster label producing the highest value of the observed hybrid log-likelihood, i.e.

$$c_i^{(t)} = \arg \max_{k=1, \dots, K} \sum_{l=1}^n \log \left(\sum_{h=1}^K \hat{\theta}_h^{(t-1)} p(\mathbf{x}_l | c_l = k, \mathbf{c}_{-l} = \tilde{\mathbf{c}}_{-l}) \right),$$

then $\mathbf{c}^{(t)}$ and $\tilde{\mathbf{Z}}^{(t)}$ are updated. The update of cluster labels happens sequentially for the nodes in the network rather than simultaneously, so any changes in the cluster assignments of nodes before i affect $\tilde{\mathbf{c}}_{-i}$ and hence the cluster label for i .

M-step: maximise the expectation of the complete data hybrid log-likelihood

$$\begin{aligned} \mathbb{E}[l_c^{hyb}(\mathbf{A}, \boldsymbol{\theta})] &= \sum_{i=1}^n \sum_{k=1}^K \hat{z}_{ik}^{(t)} \left(\log \Gamma(\sum_{j \neq i}^n \tilde{\alpha}_j^{(t-1)}) - \sum_{j \neq i}^n \log \Gamma(\tilde{\alpha}_j^{(t-1)}) + \sum_{j \neq i}^n (\tilde{\alpha}_j^{(t-1)} - 1) \log x_{ij} \right) \\ &\quad + \sum_{i=1}^n \sum_{k=1}^K \hat{z}_{ik}^{(t)} \log \hat{\theta}_k^{(t-1)}. \end{aligned}$$

Closed form solution is available for the mixing proportions, so these are updated using

$$\hat{\theta}_k^{(t)} = \frac{\sum_{i=1}^n \hat{z}_{ik}^{(t)}}{n}. \quad (7)$$

The updates for Dirichlet parameters are not available in closed form, so the set of estimates $\{\hat{\alpha}_{kh}^{(t)}\}_{k,h=1}^K$ are found numerically using the R function *optim* (R Core Team, 2019). Since the parameters of the Dirichlet distribution are strictly positive, we use the L-BFGS-B optimisation routine, which allows to restrict the range of the parameter values (Byrd et al., 1995).

The steps above are iterated until the following convergence criterion is met:

$$\left| \frac{l^{hyb}(\hat{\mathbf{A}}^{(t)}, \hat{\boldsymbol{\theta}}^{(t)}) - l^{hyb}(\hat{\mathbf{A}}^{(t-1)}, \hat{\boldsymbol{\theta}}^{(t-1)})}{l^{hyb}(\hat{\mathbf{A}}^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})} \right| < \epsilon$$

In this work, $\epsilon = 10^{-5}$ was used. More details on the algorithm can be found in Appendix B.

3.2 Identifiability of parameter order

The DirSBM exhibits a minor non-identifiability issue related to the ordering of clusters and the order of parameters. In particular, once we obtain the final node partition and the parameter estimates using the algorithm described in Section 3.1, the ordering of rows of the estimated Dirichlet concentration matrix $\hat{\mathbf{A}}$ and of the entries of the mixing proportions vector $\hat{\boldsymbol{\theta}}$ is not guaranteed to match the ordering of clusters in the partition.

To aid understanding, consider an example with

$$\mathbf{c} = (1, 1, 2, 1, 2), \quad \mathbf{A} = \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix} \text{ and } \boldsymbol{\theta} = (\theta_1, \theta_2).$$

For observation 1, $\mathbf{x}_1^* \sim \theta_1 Dir(\alpha_{11}, \alpha_{12}, \alpha_{11}, \alpha_{12}) + \theta_2 Dir(\alpha_{21}, \alpha_{22}, \alpha_{21}, \alpha_{22})$. Now, consider the same partition of nodes with the same cluster order and reorder the rows of the parameter matrix \mathbf{A} and the entries of $\boldsymbol{\theta}$, i.e.

$$\mathbf{c} = (1, 1, 2, 1, 2), \quad \mathbf{A}' = \begin{pmatrix} \alpha_{21} & \alpha_{22} \\ \alpha_{11} & \alpha_{12} \end{pmatrix} = \begin{pmatrix} \alpha'_{11} & \alpha'_{12} \\ \alpha'_{21} & \alpha'_{22} \end{pmatrix} \text{ and } \boldsymbol{\theta}' = (\theta_2, \theta_1) = (\theta'_1, \theta'_2).$$

Then,

$$\begin{aligned} \mathbf{x}_1^* &\sim \theta'_1 Dir(\alpha'_{11}, \alpha'_{12}, \alpha'_{11}, \alpha'_{12}) + \theta'_2 Dir(\alpha'_{21}, \alpha'_{22}, \alpha'_{21}, \alpha'_{22}) \\ &= \theta_2 Dir(\alpha_{21}, \alpha_{22}, \alpha_{21}, \alpha_{22}) + \theta_1 Dir(\alpha_{11}, \alpha_{12}, \alpha_{11}, \alpha_{12}), \end{aligned}$$

which will clearly give rise to the same likelihood value as the originally ordered parameters.

To address this, we create a hard partition based on probabilities $\hat{\mathbf{Z}}$ from the E-step and use the order of clusters in this hard partition as reference since column entries of $\hat{\mathbf{Z}}$ are tied to the individual terms of the Dirichlet mixture. Then we use the *matchClasses()* function in R (*e1071* package, Dimitriadou et al., 2009) to match the order of clusters in the E-step to that in the C-step, and use this match to reorder the rows of $\hat{\mathbf{A}}$ and entries of $\hat{\boldsymbol{\theta}}$ at the convergence of the algorithm.

3.3 Algorithm initialisation

As in most mixture models, extra attention should be paid to the choice of the initialisation strategy since the objective function in question is not unimodal, and the algorithm is only guaranteed to converge to a local optimum (Wu, 1983).

In this paper, we consider the following initialisation strategies, which are compared in the simulation study in Section 5.1:

- **Random:** each node is assigned to one of K groups randomly with equal probability. Starting with a random partition of nodes gives us an idea of the performance of the algorithm when no information on the data is used when the initial partition is created. To reduce the risk of convergence to a poor local maximum, we run the algorithm with a number of random starting partitions and select the best based on the value of the observed hybrid log-likelihood. In the simulation study in Section 5.1, we set the number of random starting partitions equal to 5 as we want to strike a balance between performance and computational costs. A larger number of random starting partitions have been tested, providing only a marginal improvement on average, but at a higher computational cost.
- **K-means:** the k-means algorithm (Hartigan and Wong, 1979) is repeatedly used to perform clustering on the data matrix \mathbf{X}^* to reduce the risk of convergence to a local maximum, and the clustering solution of the best run is used as an initial partition. This is a widely used initialisation strategy as it is computationally inexpensive. Since the k-means algorithm itself is initialised at random, we run it 50 times in order to find the initial cluster labels as part of the simulation study.
- **K-means on CLR-transformed data (CLR+K-means):** as our data are compositional in nature, the centered log-ratio transformation is first applied to the data matrix \mathbf{X}^* (Aitchison, 1982), which maps the compositions to the real line, then the k-means algorithm is used in the same fashion as above to find an initial partition (Godichon-Baggioni et al., 2017).
- **Spectral clustering:** eigendecomposition of \mathbf{X}^* is performed and used for dimensionality reduction, then the k-means algorithm is used on the reduced data to get a set of initial cluster labels (von Luxburg, 2007). Spectral clustering is widely used for initialisation as it can handle more complex cluster shapes than k-means itself whilst maintaining low computational costs.
- **Binary SBM (BinSBM):** a binary version of the network is created from \mathbf{X} by assigning binary edges where the weighted edge exceeds a pre-specified threshold. After empirical investigation, we selected as a threshold the mean weight, as it tends to provide a reasonably well-connected binary network that retains clustering structure. Once such a binary network is created, a Bernoulli stochastic block model is fitted using the R package *blockmodels* (Léger, 2016) and the clustering partition is used to initialise the algorithm.
- **Gaussian SBM (GausSBM):** a Gaussian stochastic block model is fitted directly to \mathbf{X} using the R package *blockmodels* (Léger, 2016) and the obtained clustering solution is used for initialisation.

3.4 Model selection

The modelling approach detailed in Section 2 is only applicable when the number of clusters is set in advance. However, in reality, in many situations the number of clusters is unknown and needs to be inferred from the data. Therefore, a model selection criterion indicating how well models with varying numbers of clusters fit the data is required in order to choose the number of groups appropriately.

One popular choice in stochastic block modelling is the integrated completed likelihood criterion (ICL, Biernacki et al., 2000; Daudin et al., 2008), typically employed for selecting the number of groups in this context; see for example Rastelli and Fop (2020), Ng and Murphy (2021), and El Haj et al. (2022). In the case of DirSBM, the ICL is given by

$$ICL(K) = l_c^{hyb}(\hat{\mathbf{A}}, \hat{\boldsymbol{\theta}} | \mathbf{X}, \hat{\mathbf{Z}}, K) - \frac{1}{2}K^2 \log n(n-1) - \frac{1}{2}(K-1) \log n,$$

where $l_c^{hyb}(\hat{\mathbf{A}}, \hat{\boldsymbol{\theta}} | \mathbf{X}, \hat{\mathbf{Z}}, K)$ is the complete data hybrid log-likelihood from Equation (6), evaluated using parameter estimates at convergence of the algorithm, used in place of the complete data log-likelihood following Marino and Pandolfi (2022). We fit the DirSBM with different numbers of clusters, and compute the respective values of the criterion. The highest ICL value corresponds to the optimal number of groups.

Appendix C contains derivations of the ICL that follow closely the derivations of the original ICL for stochastic block models proposed in Daudin et al. (2008). The simulation study concerning model selection performance using the proposed framework is contained in Section 5.4.

4 An alternative approach

As discussed in Section 1, there are two main approaches to working with compositions in compositional data analysis. One involves applying a transformation to the data that maps them to the Euclidean space, followed by the use of standard tools and models for independent data. The second is based on direct modelling of compositions (Alenazi, 2023). The methodology proposed in Section 2 aims to model the compositions directly, but since both approaches have their relative benefits, we propose an alternative that falls under the transformation-based approach, assessing its performance in Section 5.3 for comparative purposes.

We first map the compositions onto the real line using the centered log-ratio (CLR) transformation proposed in Aitchison (1982):

$$\mathbf{u}_i = CLR(\mathbf{x}_i) = \left(\log \frac{x_{i1}}{\bar{x}}, \dots, 0, \dots, \log \frac{x_{in}}{\bar{x}} \right),$$

where $\bar{x} = \left(\prod_{j \neq i} x_{ij} \right)^{\frac{1}{n-1}}$ is the geometric mean of \mathbf{x} , then apply SBM with Gaussian weights (Aicher et al., 2013, 2014) to the transformed data, assuming that each individual edge weight follows

$$u_{ij} | (z_{ik} = 1, z_{jh} = 1) \sim N(\mu_{kh}, \sigma^2). \quad (8)$$

The centered log-ratio transformation has been chosen for its property of preserving the dimensionality of the observations. That is, it maps the Aitchison simplex sample space \mathbb{S}^n to the standard Euclidean space \mathbb{R}^n , unlike many other transformations that map to \mathbb{R}^{n-1} (Greenacre, 2021). In this case, each dimension of \mathbf{u}_i corresponds to a transformed weight of an edge connecting node i to each of the other nodes in the network, so the reduction of dimensionality would require a choice of an edge to be dropped, leading to information loss.

The approach of fitting the Gaussian SBM on CLR-transformed data is implemented using the *blockmodels* and *compositions* packages in R (Léger, 2016; Van den Boogaart and Tolosana-Delgado, 2008), and its clustering performance is assessed alongside the DirSBM in Section 5.3. This approach is a rather strong competitor for the DirSBM proposed in Section 2, in particular in simpler and smaller cases, such as in networks with relatively few nodes with a small number of clusters.

There are two major drawbacks of the alternative approach in comparison to the DirSBM. First of all, the normality assumption in Equation (8) may not be appropriate as the mapping of compositional vectors to the real line (whether by using CLR or any other transformation) only eliminates a unit-sum constraint and does not guarantee convenient distributional behaviour of the resulting transformed data. This means that, in order to utilise this approach, one needs to check

the appropriateness of the transformation and the type of weighted SBM to fit to the transformed data. The DirSBM, on the other hand, is a ready-to-use model that takes into account the unique behaviour of the compositional data as it models them directly. The second disadvantage of the transformation-based approach is interpretability. The parameters of the distribution in Equation (8) (or of any other distribution selected to model the transformed data) refer to the auxiliary Euclidean space, and it is unclear how one could use these to understand the patterns present in the original simplex space. As demonstrated in Section 2.1.1, the parameters of the DirSBM can be used to compute the expected exchange proportions between nodes as well as clusters, making the interpretation of results intuitive.

5 Simulation studies

In this section, we consider various simulation studies to assess the performance of our model with respect to clustering, parameter estimation, and model selection. We also investigate the effect of the different initialisation strategies listed in Section 3.3 on the quality of the final inferred clustering of the nodes.

Artificial data are generated according to the DirSBM presented in Section 2. Different numbers of clusters, $K = \{2, 3, 5\}$, and network sizes, $n = \{30, 50, 70, 100\}$ are considered. The parameter matrices \mathbf{A} are defined so that the entries have varying levels of homogeneity, resulting in indirect modifications to the degree of overlap between clusters in the compositions. The term “level of homogeneity” is used to describe the extent to which the between-cluster weight parameters, i.e. $\{\alpha_{kh}\}_{h \neq k}$, are close to the respective within-cluster weight parameters, α_{kk} . When the values of the between- and the within-cluster parameters are set to be more similar, the resulting compositional weights in the network become more homogeneous, thus making separating the clusters more challenging. On the other hand, the lower the homogeneity, the more separated the clusters. Further details on the parameter values used in the simulation studies can be found in Appendix D.

Throughout this section, where relevant, clustering performance is evaluated using the adjusted Rand Index (ARI, [Rand, 1971](#); [Hubert and Arabie, 1985](#)), which measures the agreement between two partitions, adjusted for random chance, by comparing the estimated clustering against the true classification of the nodes.

5.1 Choice of initialisation strategy

We consider the different initialisation strategies of Section 3.3 and evaluate the model performance with respect to cluster label recovery on simulated data. We generate 50 networks of size 50 with 3 similarly-sized clusters and the parameter matrix \mathbf{A} is chosen so that the resulting within- and between-cluster edge weights are reasonably, but not excessively, distinct:

$$\mathbf{A} = \begin{pmatrix} 1.0 & 0.7 & 0.5 \\ 0.9 & 1.5 & 0.6 \\ 0.4 & 0.5 & 1.2 \end{pmatrix}.$$

We initialise the algorithm using the strategies outlined in Section 3.3 and compute the ARI. The results of this simulation study are shown in Figure 1. From the figure, random initialisation seems to work best on average, with a mean ARI of 0.805, despite being the only strategy that is not informed by the data. It also exhibits relatively low variability in comparison to all but one other strategy, which is k-means on log-ratio transformed data that is clearly inferior, with a mean ARI of only 0.513. Further investigations also reveal that, as well as having the best average ARI and low variability, random initialisation has led to the highest observed hybrid log-likelihood value in 33 out of 50 instances. Given that the number of random partitions considered was only 5, this result also shows that the proposed algorithm is capable of recovering the clustering patterns fairly successfully even when the initial cluster labels are assigned randomly and are not based on some educated guess.

It may come as a surprise to the reader that out of all the strategies considered, random initialisation performed best, as it gives a starting cluster partition that is completely uninformed. However, there are two aspects of the data modelled by the DirSBM that are worth remembering. Firstly, the data matrix \mathbf{X}^* introduced in Section 2 is quite unusual in its structure in the sense that the ordering of the dimensions in the compositional edge weight vectors is not fixed and is tied to the index of the sender node. For example, the first entry of observation \mathbf{x}_1^* , i.e. x_{11}^* , refers to the weight of an edge from node 1 to node 2 (as there are no self-loops and so there is no weight

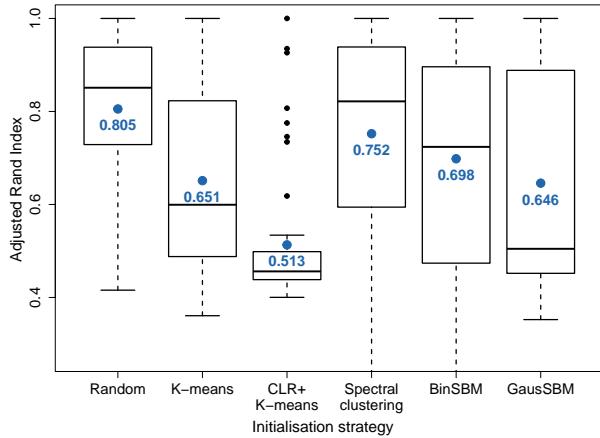


Fig. 1 Adjusted Rand Index (ARI) of DirSBM with different initialisation strategies with mean ARI in blue.

corresponding to an edge from node 1 to itself), whereas, for any other node $i \neq 1$, x_{1i}^* is the edge weight from node i to node 1. The same shift along the dimensions happens across the whole set of compositional weight vectors as we take each node in turn to be the sender. This means that distance-based algorithms, such as k-means, would not necessarily compute the distances between the matching pairs of nodes' edge weights, e.g. the first set of distances for node 1 would involve the edge weight from node 1 to node 2, and for any other node i , it would involve the edge weight from node i to node 1. This can potentially lead to finding spurious clustering patterns, producing initial partitions that result in convergence to local maxima.

The second aspect is the fact that the edges in the networks are not independent, because the weights are defined in terms of compositions. Using initialisation strategies that assume independence between the edge weights, we risk identifying false patterns in the network and getting stuck in a local maximum as a result. For instance, the Gaussian SBM initialisation could give rise to a misleading initial partition as the unit-sum constraint is ignored. With repeated random initialisation, we are less likely to get stuck in one of the local maxima as we can control the exploration of the partition space by changing the number of random starts.

Throughout the simulation studies, we use random initialisation with 5 starts, and to increase the reliability of results for the real world data in Sections 6.1 and 6.2 we increase the number of random partitions to 20.

5.2 Parameter estimation performance

In this section we examine parameter estimation quality when the number of clusters is known in advance. We consider the cases with 2, 3 and 5 clusters as well as low, medium and high level of parameter homogeneity, as described at the start of Section 5. The size of the networks is also varied to inspect its effect on parameter estimation quality.

The Frobenius distance (Golub and Van Loan, 1996) between the true parameter matrix \mathbf{A} and its estimate $\hat{\mathbf{A}}$ is calculated for 50 synthetic data sets in each scenario, and the results are presented in Figure 2. As expected, larger networks lead to better parameter estimates, showing that the proposed hybrid log-likelihood classification EM leads to consistent parameter estimates as the size of the network increases. Notably, the biggest improvement occurs when we move from $n = 30$ to $n = 50$. This comes as no surprise since the performance of the model with regards to the recovery of latent clustering improves significantly when the number of nodes increases from 30 to 50 and above, as seen in Section 5.3. We can also note that higher homogeneity of Dirichlet parameters makes the estimation more challenging as a result of higher difficulty in separating the clusters, and the parameter estimates are closer to their true counterparts when the number of clusters is lower. This is again an expected result, as, when the number of clusters increases, recovering the underlying partition of the nodes is more challenging and each individual parameter is estimated from a smaller share of the data.

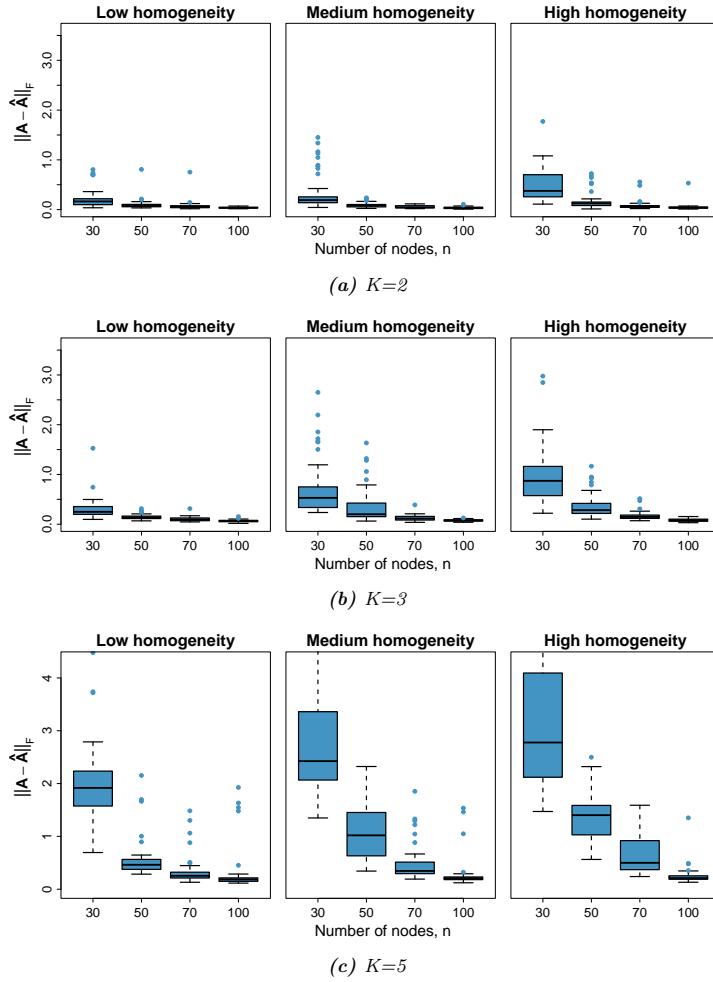


Fig. 2 Frobenius distance between the true parameter matrix \mathbf{A} and its estimate $\hat{\mathbf{A}}$, based on 50 artificial data sets. The number of clusters K is fixed at the respective true values of (a) 2, (b) 3 and (c) 5.

5.3 Clustering structure recovery

To assess the quality of the clustering results, we compare the clustering performance of the following models using the ARI:

1. **Binary SBM fitted on the binary version of the network (BinSBM):**

We test the approach that is often taken in practice where a weighted network is transformed into a binary one. Following closely the description of using binary SBM as an initialisation strategy from Section 3.3, we construct a binary network and fit the Bernoulli SBM using the *blockmodels* package in R (Léger, 2016).

2. **Naive Gaussian SBM (GausSBM):**

We fit SBM with Gaussian edge weights (Aicher et al., 2014) directly to the data. This model is used to test the importance of the constant-sum constraint on the sender edge weights in different scenarios, as similar naive approaches are sometimes seen in real-world applications.

3. **Gaussian SBM on log-ratio transformed data (CLR+GausSBM):**

The data are mapped onto the real line using a centered log-ratio transformation, then SBM with Gaussian weights is fitted to the transformed data, as described in Section 4.

4. **DirSBM (DirSBM):**

We fit the DirSBM described in Section 2 directly to the data.

Figure 3 illustrates the performance of all 4 models on artificial data sets with 2, 3 and 5 clusters across multiple network sizes and varying levels of Dirichlet parameter homogeneity. Each boxplot

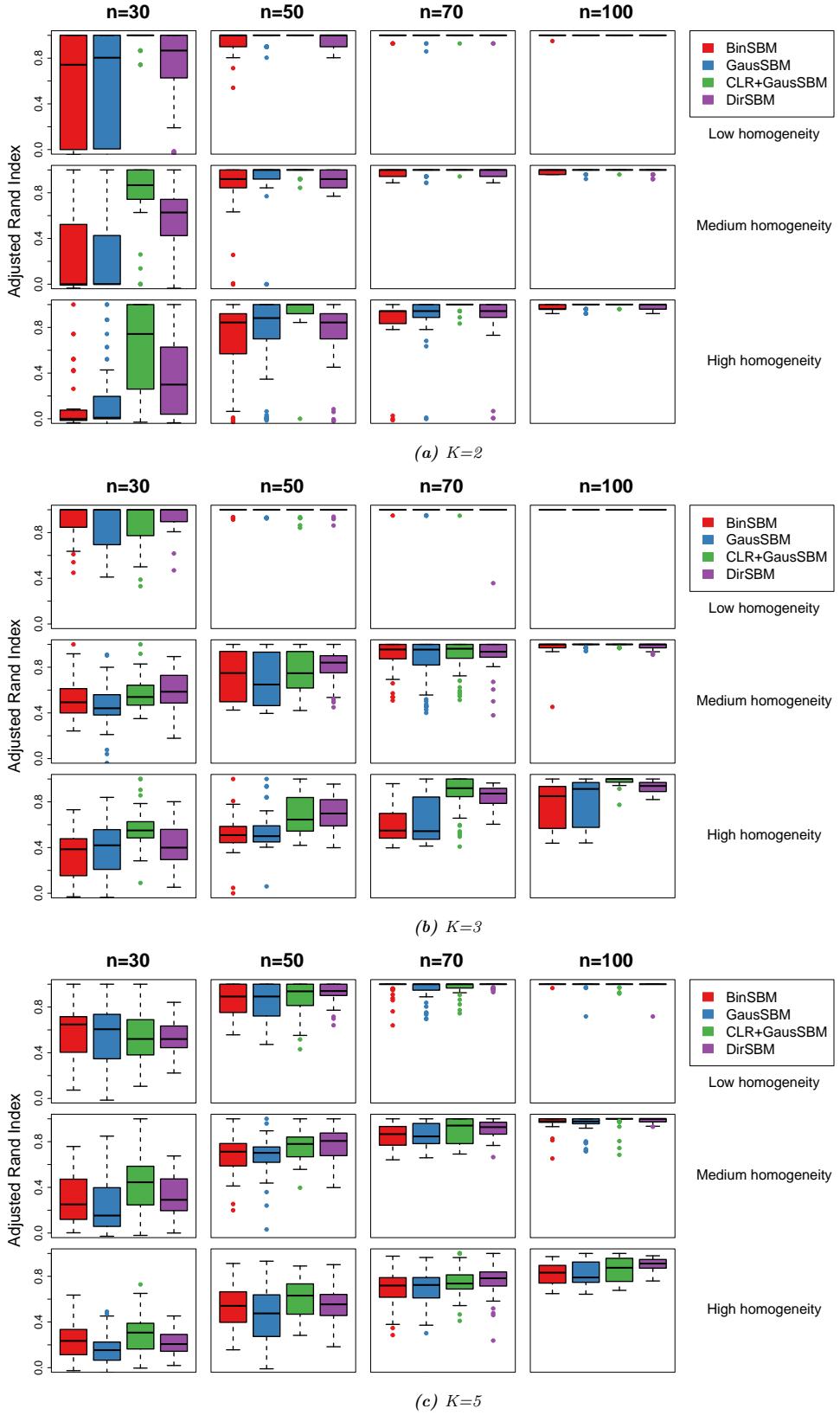


Fig. 3 Adjusted Rand Index between the true partition and clustering solutions estimated by binary SBM, Gaussian SBM, Gaussian SBM on centered log-ratio transformed data and DirSBM, with different network sizes and levels of Dirichlet parameters homogeneity. Results are based on 50 artificial data sets with (a) $K = 2$, (b) $K = 3$ and (c) $K = 5$ clusters.

Table 1 Choice of the number of clusters using integrated completed likelihood (ICL). Results indicate the number of times each number of clusters has been selected by the criterion, based on 50 artificial networks.

K		2				3				5					
\hat{K}		1	2	3	4	1	2	3	4	1	2	3	4	5	6
Low homogeneity	$n = 30$	1	45	3	1	3	46	1		2	30	18			
	50		50				50				10	40			
	70		48	1	1		50				1	49			
	100		45	5			50					45	5		
Medium homogeneity	$n = 30$		44	6		37	13		1	20	27	2			
	50		47	3		10	40			3	37	10			
	70		47	3			48	2			19	31			
	100		42	7	1		49	1		2	47	1			
High homogeneity	$n = 30$	21	28	1		1	35	14	9	25	16				
	50	1	41	8			15	35	1	31	18				
	70		45	5		2	47	1		40	10				
	100		46	4			49	1		7	43				

is based on 50 networks. The DirSBM was initialised with 5 random cluster partitions based on the results of the simulation study in Section 5.1.

It is evident that the performance of all models improves with sample size, often approaching an ARI of 1 as we reach 100 nodes. In the case of BinSBM and GausSBM, this can be due to the fact that, in larger networks, the compositional samples are rather high-dimensional, and so the constant-sum constraint does not induce as much dependence between the individual dimensions as in smaller networks, resulting in a good fit of the models that assume independence between the weights. In the case of CLR+GausSBM and DirSBM, the improvement of performance with sample size can be associated with more accurate parameter estimation. Smaller networks present a greater challenge, due to both stronger dependence between the dimensions of compositions and the limited amount of data available for parameter estimation.

The CLR+GausSBM and DirSBM tend to outperform the competitors in terms of clustering structure recovery, and the gap in performance is significantly wider for smaller networks and higher homogeneity of Dirichlet parameters. It is also notable that the performance of BinSBM and the GausSBM on the raw compositional data often exhibits much higher variability indicated by wider boxplots.

With regards to the comparison between the DirSBM and the CLR+GausSBM, the two methods tend to perform similarly well. Looking at Figures 3(b) and 3(c), we observe that in many scenarios the DirSBM performs better and with lower overall variability, making it a more suitable model in the cases with larger number of clusters in the data. Considering Figure 3(a), where the number of underlying clusters is 2, we note that CLR+GausSBM tends to perform better than DirSBM. Albeit a simplistic approach, CLR+GausSBM seems to present a valid alternative when modelling compositional networks if it is expected that the number of clusters is small.

5.4 Model selection performance

We assess the model selection performance using the integrated completed likelihood (ICL) detailed in Section 3.4. Table 1 reports the confusion matrices of the true number of clusters K and the optimal number of clusters \hat{K} as selected by ICL.

We observe that ICL is very successful in selecting the correct number of clusters in larger networks with any level of parameter homogeneity, which is to be expected provided that both the latent structure recovery and the parameter estimation improve with network size. It is also not surprising that, in smaller networks with higher levels of homogeneity and/or higher numbers of clusters, the model selection becomes a much more complex task, and the number of clusters tends to be underestimated by the ICL. This is due to the fact that in cases with medium and high levels of homogeneity the clusters are much harder to separate, coupled with the difficulty of accurately estimating the parameters in smaller networks, as seen in Sections 5.3 and 5.2 respectively.

6 Application to real world network data

6.1 Erasmus programme data

The Erasmus programme is the most popular university student exchange programme in Europe, spanning thousands of Higher Education institutions in more than 30 countries. [Gadár et al. \(2020\)](#) have produced an extensive data set of student, staff and teacher mobility in the Erasmus network by combining data from a number of different sources.

The aim of our analysis is to understand global mobility patterns in the Erasmus network, focusing on student exchanges. Due to significant differences in the student populations of the European countries, analysis of raw counts of students per pair of countries may not provide particular insights regarding the popularity of destinations, as the biggest countries dominate simply due to their overall hosting capacity.

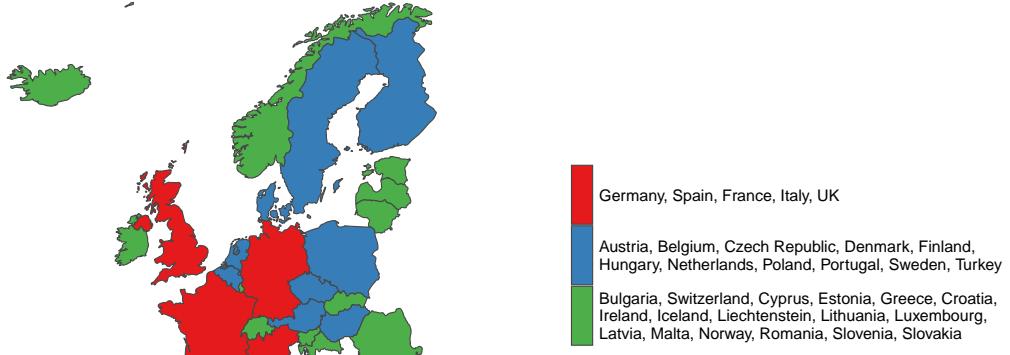
After aggregating the student numbers to country level for the 2012-2013 academic year, a directed network with countries as nodes and proportions of students as edge weights is constructed, with the proportions summing to 1 for the sending countries. The Erasmus network is very dense ($\approx 87\%$ of all possible edges are present), and there is no reason to believe that the remaining edges are not allowed to exist since all countries in the Erasmus network are allowed to exchange students. Therefore, instead of treating the edges as truly missing, we assign a value of 0.001 to the interactions with zero count and treat the network as fully connected. In compositional data analysis, zero values present a great challenge and are an active area of research since standard tools are not designed to handle zero-valued compositional parts ([Filzmoser et al., 2018](#)). Replacing zeros by some small value is a standard approach in the case of zero counts that works generally well, especially considering its simplicity. Since in our case 0.001 is added to the original count data rather than to the compositions themselves, the final results should be robust relative to the magnitude of the added constant.

We implement DirSBM with K ranging from 1 to 6 and use the ICL to select the number of clusters. According to the ICL, the optimal number of clusters is 3 (ICL= 3764), followed by a solution with 5 clusters (ICL= 3718). As the Erasmus network is fairly small (33 nodes), the ICL might underestimate the number of clusters due to the penalty being too strong in the context of small n , as we saw in Section 5.4, therefore we examine the results of both $K = 3$ and $K = 5$ cases.

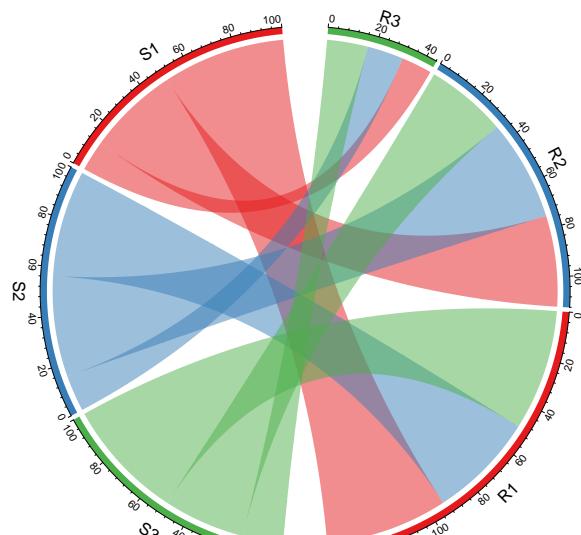
Figure 4(a) illustrates the solution with 3 clusters on a map of Europe, noting the membership of each cluster on the side as smaller countries are difficult to see on the plot. Figure 4(b) represents the total percentage flows between the clusters, based on the estimated matrix $\hat{\mathbf{V}}$. The estimate of the parameter matrix \mathbf{A} as well as the expected node-to-node and cluster-to-cluster exchange matrices $\hat{\mathbf{W}}$ and $\hat{\mathbf{V}}$ from Equations (3) and (4) respectively are provided in Appendix E.

The clustered compositions are reasonably interpretable in a real-world context. The red group seems to dominate the student preference list across all clusters as indicated by high country-wise exchange percentages (12.6, 8.7 and 10 respectively) from the first column of $\hat{\mathbf{W}}$, and it consists of Germany, Spain, France, Italy and the UK. The red cluster collectively also notably receives the highest total percentages. The blue cluster can be described as containing countries that are still very popular exchange destinations, but not as popular as the likes of Germany and France, possibly due to their relatively small size or more challenging national language. Although roughly equal percentages of students are expected to be retained by the blue cluster and to be sent to the countries in the red cluster (41.4% and 43.7% respectively), these shares are split between 11 countries instead of 5, so each individual country in the blue cluster is expected to receive smaller shares in comparison to countries in the red cluster; for instance, any red cluster country is expected to receive 10% of students from any green cluster country, whereas for blue cluster countries this share is only 3%. The green cluster contains the less popular exchange destinations, such as Bulgaria and Slovakia, as indicated by significantly lower expected country-wise and cluster-wise exchange shares. Collectively, the green cluster also retains the least of students, as only 16.8% go on exchange within the cluster itself, while the remaining 83.2% go on exchange to the countries in the red and blue clusters. It can also be noted that the countries in the red cluster are generally closely co-located in geographical terms, as are most countries in the blue cluster, except for Portugal and Turkey, and the countries in the green cluster are more spread out.

The second best solution according to ICL with $K = 5$ is presented in Figure 5, and the parameter estimates as well as matrices $\hat{\mathbf{V}}$ and $\hat{\mathbf{W}}$ are available in Appendix E. The solution with 5 clusters captures more complex patterns in comparison to the solution with 3 clusters. The red cluster, now consisting of only Germany, Spain and France, is the cluster of countries receiving the highest percentages of students from the countries of all clusters, ranging between the expected 8.1% to 19.6% per country pair, and the second largest total shares, dominated by the blue cluster



(a)



(b)

Fig. 4 Clustering solution for DirSBM with 3 clusters: (a) map of Europe with countries coloured by cluster assignment; (b) chord diagram of total percentage flows between clusters of countries based on \hat{V} . On the left, S_1 , S_2 and S_3 denote the sending cluster (1, 2 and 3). Percentages departing from the sender sum up to 100. On the right, R_1 , R_2 and R_3 denote the receiving cluster (1, 2 and 3).

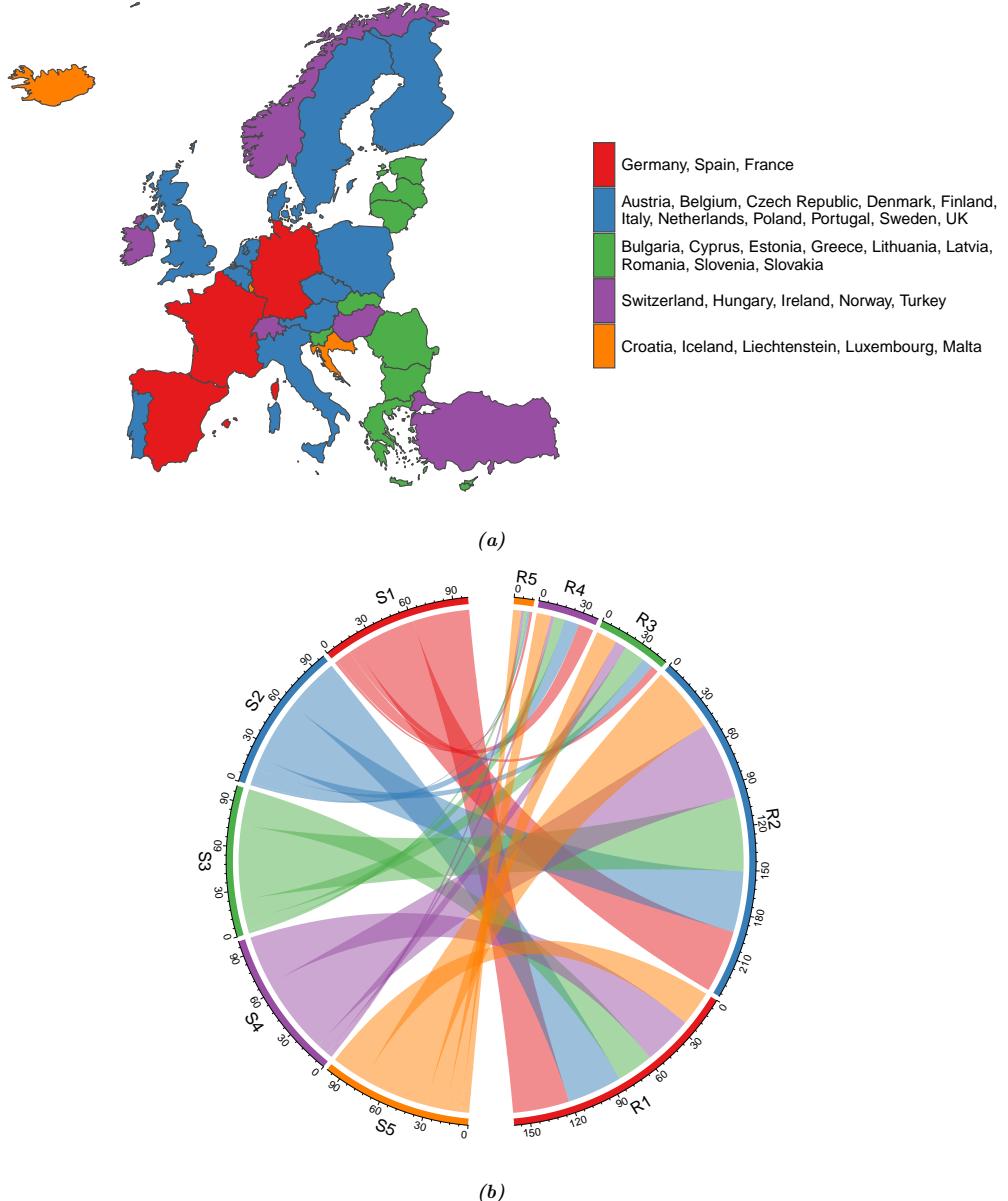


Fig. 5 Clustering solution for DirSBM with 5 clusters: **(a)** map of Europe with countries coloured by cluster assignment; **(b)** chord diagram of total percentage flows between clusters of countries based on \hat{V} . On the left, S_1 - S_5 denote the sending cluster (numbered 1 to 5). Percentages departing from the sender sum up to 100. On the right, R_1 - R_5 denote the receiving cluster (numbered 1 to 5).

by a small margin. It should be noted that the total shares received by the red cluster are only split between 3 countries instead of 11 like in the case of the blue cluster. This cluster is also characterised by retaining 39.1% of its students, and sending 43.3% to the blue cluster, implying a lack of diversity of exchange destinations among its students. The blue cluster can be described as containing reasonably popular exchange destinations, but not quite as popular as the red countries, and, similarly to the red cluster, the majority of students go on exchange within the cluster itself (42.2%) or to the red cluster (38.1%). The green cluster, despite consisting of 9 countries (8 Eastern European countries and Cyprus), does not attract large shares of students (total cluster exchange shares are between 2.4% and 10.6%), which may be due to the fact that such countries are less well known exchange destinations. Green countries have a somewhat more diverse preference profile in comparison to the first two clusters, as the parameter estimates are much closer to each other, but there is still a strong preference for the red and blue cluster countries. Purple countries, which are Switzerland, Hungary, Ireland, Norway and Turkey, can be described as medium-preference exchange destinations. They have a strong preference for the red and blue cluster countries, and very limited interest in countries within its own cluster, as well as in green and orange clusters. The orange cluster is the one having the most homogeneous preferences, as indicated by the smallest differences between the parameter estimates, and its countries tend to be among the least popular destinations for exchange. This can possibly be explained by the fact that these countries are either very small, are located very far from the centre of Europe or are not members of the European Union (as of the 2012-2013 academic year). The green, purple and orange clusters retain very small shares or transfers, 14.2%, 2.4% and 5.6% respectively, while the majority of students from these clusters tend to go on exchange to countries in the red and blue clusters. A final observation is that there seems to be a connection between the exchange preferences of Erasmus students and the level of economic development of the hosting country or the prestige of going on exchange to specific destinations. The countries with stronger economies tend to be grouped together and to exhibit higher expected country-to-country exchange proportions, whereas countries with weaker economies or newer members of the programme, such as Turkey and Croatia, tend to be separated from the more established members.

6.2 London bike sharing data

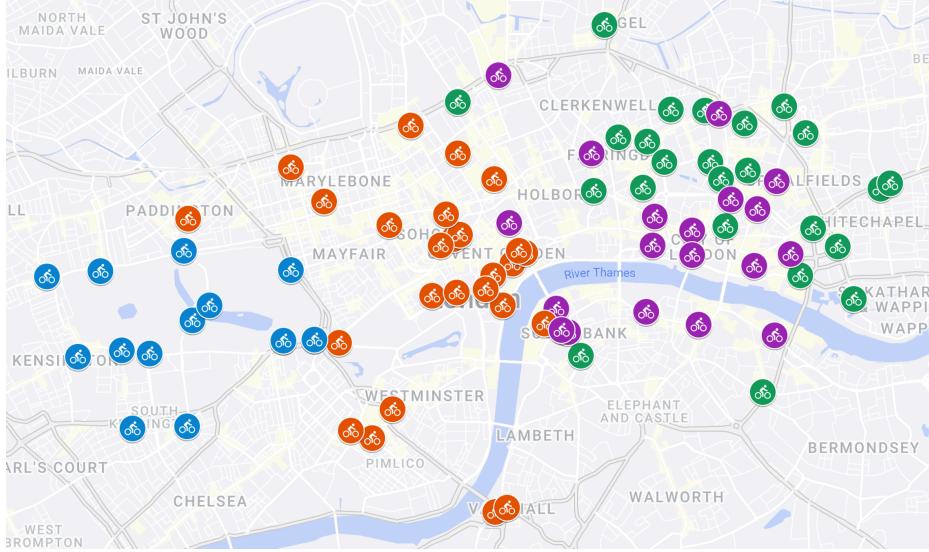
We test DirSBM on the London bike sharing data from 2014 provided by the [Transport for London \(2016\)](#). We start by counting the number of trips taken between pairs of stations and identifying the busiest stations for our analysis. To do so, we look at the top 100 start and end stations in terms of volume of bikes exchanged and consider their intersection, i.e. the stations that are both among the most popular start and end points of the journey, resulting in an overall network of 85 stations. The trips only between these 85 stations account for over 10% of the total year's trips. The network is very dense, with just under 2% of edges having zero weights, hence, similarly to the Erasmus programme network from Section 6.1, we set the weights of zero-weighted edges equal to 0.001. We then compute the compositional edge weights by dividing the count weights by the total number of trips taken from the start station.

We fit the DirSBMs with number of clusters between 1 and 8 clusters and select the optimal number of clusters based on the ICL. According to the criterion, the solution with $K = 4$ is optimal (ICL= 27866).

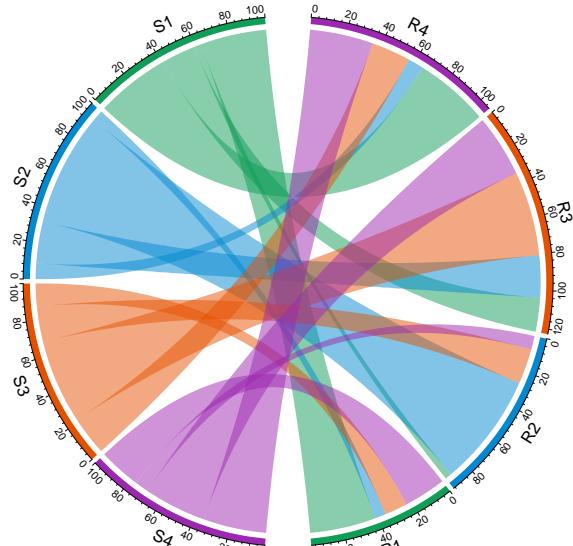
Figure 6 illustrates the solution with 4 clusters on a map of central London (an interactive version of the map is available [here](#)) as well as the chord diagram for total exchange proportions between clusters. The Dirichlet parameter matrix estimate $\hat{\mathbf{A}}$ as well as matrices $\hat{\mathbf{V}}$ and $\hat{\mathbf{W}}$ are provided in Appendix F.

Unlike the Erasmus network data presented in Section 6.1, the clustering structure is largely driven by stronger connections within clusters than those between clusters as the diagonal elements of $\hat{\mathbf{A}}$ tend to be dominant. The clusters are also much more balanced in terms of number of observations, with 24, 15, 26 and 20 stations in the green, blue, red and purple cluster, respectively. There is a geographical component to the clustering structure recovered, in the sense that the clusters are quite compact, indicating that more trips take place within the neighbouring areas of the city, and the neighboring clusters tend to exchange larger shares of trips in comparison to more distant ones.

The blue cluster is centered around Hyde Park and Kensington, which has the least geographical spread, and it retains the majority of its flow within the cluster (62.3% of trips), possibly due to the purpose of the journey being recreational cycling. A portion of 23.2% of trips departing from the blue cluster terminate in the neighboring red cluster, and significantly smaller shares arrive to the purple (8.6%) or the green clusters (5.9%). The red cluster contains bike stations around Soho



(a)



(b)

Fig. 6 Clustering solution for DirSBM with 4 clusters, as selected by ICL, on London bike sharing data: (a) map of Central London with bike stations coloured by cluster assignment. Image created using [Google My Maps](#) software, with the interactive map available [here](#); (b) chord diagram of total percentage flows between clusters of stations based on \hat{V} . On the left, S_1 - S_4 denote the sending cluster (numbered 1 to 4). Percentages departing from the sender sum up to 100. On the right, R_1 - R_4 denote the receiving cluster (numbered 1 to 4).

and Covent Garden regions of London that are known for entertainment and shopping, as well as stations near major tourist attractions (such as the British Museum) or large train stations (e.g. Euston station, King’s Cross station). A significant 46.4% of trips take place within the cluster, and the green cluster receives the least trips starting from the red cluster, possibly due to the geographical distance between the two. The green and the purple clusters have strong connections with each other, exchanging 41.1% and 23.2% of their trips (in comparison to retaining 36.5% and 34.9% respectively), which can be due to the concentration of corporate buildings in this region of London and the employees living within cycling distance of their workplaces. Neither cluster is well connected to the blue cluster, likely due to the distance and availability of faster public transport options as well as the purpose of travel. A proportion of 34.3% of trips from stations in a purple cluster arrive in the red cluster, whilst 21.6% travel in the opposite direction. In summary, the clusters detected by DirSBM by modelling the proportions of transfers between the stations are interpretable in terms of geographical locations and neighbourhoods of London.

7 Discussion

In this paper, we have introduced DirSBM, an extension of the stochastic block model to directed weighted networks with compositional edge weights that is based on direct modelling of compositional weights vectors using the Dirichlet distribution. To the best of our knowledge, no counterparts currently exist in the literature, making the proposed clustering methodology the first of its kind. We have developed an inferential procedure based on a variant of the classification expectation-maximisation algorithm for hybrid likelihood. Model selection is addressed using the integrated completed likelihood criterion. We have also proposed an alternative framework for clustering for composition-weighted networks via a log-ratio transformation of the data and subsequent application of the existing weighted SBM to the transformed data. In the simulation studies, we have shown the effectiveness of DirSBM in terms of parameter estimation and recovery of the clustering structure in a variety of settings, and have explored the use of ICL for model selection, which has proven to work reasonably well. The code implementing the modelling framework in R can be found on [GitHub](#).

The modelling approach can be extended in multiple directions. One important extension to the DirSBM is introducing structural zeros, to enable the ability to model the absence of guaranteed edges in the network. One could use the idea from [Tsagris and Stewart \(2018\)](#) that concerns Dirichlet regression with zeros, where indicators are used to “discard” zero entries of compositional vectors and subsequently fit the model to the non-zero subset of compositional data. As well as making the model more realistic and applicable to a wider range of real world data, such an extension could lead to numerically more stable performance, since the dimension of the Dirichlet observations would not scale with the number of nodes.

The use of the centered log-ratio transformation approach presented in Section 4 has been shown to be a simple valid alternative to the DirSBM in simple scenarios with few clusters in small networks. With regards to this approach, transformations other than the centered log-ratio could be explored as well as other weighted SBMs. As for the incorporation of structural zeros, it is unclear how one could include such information in this methodology as the edges are treated as independent and the respective parameters of the distribution are estimated in the Euclidean space ([Filzmoser et al., 2018](#), Chapter 13.5). This implies that even if two nodes have similar weights in a compositional sense, the absence of edges for one of them can lead to assignment to different clusters, because the absent edges would make them appear more different than they actually are when mapped to the Euclidean space. To give a simple illustrative example, suppose we have two compositions, $x_1 = (0.2, 0.5, 0.28, 0.01, 0.01)$ and $x_2 = (0.2, 0.5, 0.3, 0, 0)$, and let the zero values be structural zeros, i.e. indicating a missing edge in the network. Then, applying the centered log-ratio transformation, $u_1 = \text{clr}(x_1) = (0.95, 1.86, 1.28, -2.05, -2.05)$ and $u_2 = \text{clr}(x_2) = (-0.44, 0.48, -0.035, 0, 0)$ (using the convention $\log 0 = 0$), a false impression is created that the compositions are quite different.

Other developments could include investigating more educated and efficient initialisation strategies for the proposed algorithm. Although random initialisation allows to explore the solution space rather well by considering the most diverse set of starting points, thus reducing the risk of convergence to a local maximum, it does so at a computational cost. This cost could potentially be reduced if the algorithm was to be run from fewer, or ideally a single, more informed initial clustering allocation set. Other distributions for compositional data could be considered to model the sets of proportional edge weights in the network, such as the generalised Dirichlet distribution, which would be less restrictive on the compositional variance ([Connor and Mosimann, 1969](#)).

A Complete data log-likelihood

Recall that given the binary matrix of cluster allocations \mathbf{Z} , the compositional data \mathbf{X} has the following probability density:

$$p(\mathbf{X}|\mathbf{Z}) = \prod_{i=1}^n p(\mathbf{x}_i|\mathbf{z}_i, \mathbf{Z}_{-i}) = \prod_{i=1}^n \prod_{k=1}^K \left[\frac{\Gamma(\sum_{j \neq i} \alpha_j)}{\prod_{j \neq i} \Gamma(\alpha_j)} \prod_{j \neq i} x_{ij}^{\alpha_j - 1} \right]^{z_{ik}}, \text{ where } \alpha_j = \sum_{h=1}^K z_{jh} \alpha_{kh}.$$

The latent variables have probability distribution:

$$p(\mathbf{z}_i) = \prod_{k=1}^K \theta_k^{z_{ik}}.$$

This results in the following complete data likelihood:

$$\mathcal{L}_c(\mathbf{A}, \boldsymbol{\theta}) = \prod_{i=1}^n \prod_{k=1}^K p(\mathbf{x}_i, \mathbf{z}_i, \mathbf{Z}_{-i}) = \prod_{i=1}^n \prod_{k=1}^K \left[\theta_k \frac{\Gamma(\sum_{j \neq i} \alpha_j)}{\prod_{j \neq i} \Gamma(\alpha_j)} \prod_{j \neq i} x_{ij}^{\alpha_j - 1} \right]^{z_{ik}} \quad (9)$$

Taking the natural logarithm of Equation (9), we arrive at the expression for the complete data log-likelihood from Equation (5).

B Classification EM with hybrid log-likelihood

The expectation of the complete data hybrid log-likelihood with respect to a single latent variable \mathbf{z}_i is given by:

$$\begin{aligned} & \mathbb{E}[l_c^{hyb}(\mathbf{A}, \boldsymbol{\theta})] \\ &= \sum_{i=1}^n \sum_{k=1}^K \mathbb{E}[z_{ik}] \left(\log \Gamma(\sum_{j \neq i} \tilde{\alpha}_j) - \sum_{j \neq i} \log \Gamma(\tilde{\alpha}_j) + \sum_{j \neq i} (\tilde{\alpha}_j - 1) \log x_{ij} \right) + \sum_{i=1}^n \sum_{k=1}^K \mathbb{E}[z_{ik}] \log \theta_k. \end{aligned}$$

The E-step can be found in standard EM fashion by using the Bayes rule:

$$\begin{aligned} \hat{z}_{ik} &= \widehat{\Pr}(z_{ik} = 1 | \mathbf{x}_i, \tilde{\mathbf{Z}}_{-i}) = \frac{p(z_{ik} = 1 | \tilde{\mathbf{Z}}_{-i}) p(\mathbf{x}_i | z_{ik} = 1, \tilde{\mathbf{Z}}_{-i})}{p(\mathbf{x}_i | \tilde{\mathbf{Z}}_{-i})} \\ &= \frac{p(z_{ik} = 1) p(\mathbf{x}_i | z_{ik} = 1, \tilde{\mathbf{Z}}_{-i})}{\sum_h p(z_{ih} = 1) p(\mathbf{x}_i | z_{ih} = 1, \tilde{\mathbf{Z}}_{-i})} \\ &\propto p(z_{ik} = 1) p(\mathbf{x}_i | z_{ik} = 1, \tilde{\mathbf{Z}}_{-i}) \\ &= \theta_k \prod_{j \neq i}^n x_{ij}^{\tilde{\alpha}_j} \frac{\Gamma(\sum_{j \neq i} \tilde{\alpha}_j)}{\prod_{j \neq i} \Gamma(\tilde{\alpha}_j)}, \end{aligned}$$

as $p(z_{ik} = 1 | \tilde{\mathbf{Z}}_{-i}) = p(z_{ik} = 1)$ due to the working independence assumption.

The M-step involves finding the estimates for the mixing proportions $\boldsymbol{\theta} = (\theta_1, \dots, \theta_K)$ and the Dirichlet connectivity matrix \mathbf{A} . The closed form solution for the mixing proportions is derived similarly to that in the standard EM algorithm by maximising $\mathbb{E}[l_c^{hyb}(\mathbf{A}, \boldsymbol{\theta})]$, subject to constraint $\sum_k \theta_k = 1$. Let

$$f = \mathbb{E}[l_c^{hyb}(\mathbf{A}, \boldsymbol{\theta})] - \lambda \left(\sum_{k=1}^K \theta_k - 1 \right).$$

The partial derivative of f with respect to θ_k is

$$\frac{1}{\theta_k} \sum_{i=1}^n \hat{z}_{ik} - \lambda = 0,$$

which gives $\lambda \theta_k = \sum_{i=1}^n \hat{z}_{ik}$. Summing over k and using the unit-sum constraint for $(\theta_1, \dots, \theta_K)$, we find that $\lambda = n$, producing the update for the mixing proportions from Equation (7).

The updates for the Dirichlet concentration parameter matrix \mathbf{A} are not available in closed form and so are found numerically using the R function *optim* (R Core Team, 2019). L-BFGS-B optimisation procedure (Byrd et al., 1995) is used to update the set of parameters $\{\alpha_{kh}\}_{k,h=1}^K$ as it allows a permitted range of values to be set; in the case of DirSBM, we are only interested in strictly positive solutions.

C Integrated completed likelihood

From Biernacki et al. (2000), the exact complete-data integrated log-likelihood is given by

$$\log p(\mathbf{X}, \mathbf{Z}|K) = \log p(\mathbf{X}|\mathbf{Z}, K) + \log p(\mathbf{Z}|K) \quad (10)$$

Following closely the derivations of the ICL for random graphs in Daudin et al. (2008), we can find the first term of Equation (10) using a BIC approximation:

$$\log p(\mathbf{X}|\mathbf{Z}, K) = \max_{\mathbf{A}} \log p(\mathbf{X}|\mathbf{Z}, \mathbf{A}, K) - \frac{1}{2}K^2 \log n(n-1),$$

where K^2 is the number of parameters in the model (elements of non-symmetric \mathbf{A} matrix), and $n(n-1)$ is the number of edge weights in the network (with no self-loops).

The second term of Equation (10) is derived in the same fashion as in Daudin et al. (2008), using a non-informative Jeffrey prior and Stirling formula for an approximation of a Gamma function, leading to

$$\log p(\mathbf{Z}|K) = \max_{\boldsymbol{\theta}} \log p(\mathbf{Z}|\boldsymbol{\theta}, K) - \frac{1}{2}(K-1) \log n,$$

with $(K-1)$ being the number of free parameters in $\boldsymbol{\theta}$ and n being the number of latent variables in the model.

Substituting the approximation results back and then replacing the complete data log-likelihood with its hybrid counterpart in the fashion of Marino and Pandolfi (2022), we arrive at

$$\begin{aligned} ICL(K) &= \log p(\mathbf{X}|\hat{\mathbf{Z}}, \hat{\mathbf{A}}, K) - \frac{1}{2}K^2 \log n(n-1) + \log p(\hat{\mathbf{Z}}|\hat{\boldsymbol{\theta}}, K) - \frac{1}{2}(K-1) \log n \\ &= \log p(\mathbf{X}, \hat{\mathbf{Z}}|\hat{\mathbf{A}}, \hat{\boldsymbol{\theta}}, K) - \frac{1}{2}K^2 \log n(n-1) - \frac{1}{2}(K-1) \log n \\ &= l_c^{hyb}(\hat{\mathbf{A}}, \hat{\boldsymbol{\theta}}|\mathbf{X}, \hat{\mathbf{Z}}, K) - \frac{1}{2}K^2 \log n(n-1) - \frac{1}{2}(K-1) \log n. \end{aligned}$$

D Simulation studies: additional notes

Parameters used to generate data sets with different levels of parameter homogeneity, low, medium and high, denoted with subscripts L , M and H , respectively:

$K = 2$

$$\mathbf{A}_L = \begin{pmatrix} 1.0 & 0.6 \\ 0.8 & 1.5 \end{pmatrix}, \mathbf{A}_M = \begin{pmatrix} 1.0 & 0.6 \\ 0.9 & 1.4 \end{pmatrix}, \mathbf{A}_H = \begin{pmatrix} 1.0 & 0.8 \\ 0.9 & 1.5 \end{pmatrix}$$

$K = 3$

$$\mathbf{A}_L = \begin{pmatrix} 1.0 & 0.6 & 0.2 \\ 0.6 & 1.5 & 0.5 \\ 0.3 & 0.4 & 1.2 \end{pmatrix}, \mathbf{A}_M = \begin{pmatrix} 1.0 & 0.7 & 0.5 \\ 0.9 & 1.5 & 0.6 \\ 0.4 & 0.5 & 1.2 \end{pmatrix}, \mathbf{A}_H = \begin{pmatrix} 1.0 & 0.7 & 0.5 \\ 0.9 & 1.3 & 0.7 \\ 0.6 & 0.5 & 1.2 \end{pmatrix}$$

$K = 5$

$$\begin{aligned} \mathbf{A}_L &= \begin{pmatrix} 1.0 & 0.6 & 0.2 & 0.3 & 0.5 \\ 0.6 & 1.5 & 0.5 & 0.4 & 0.7 \\ 0.3 & 0.4 & 1.2 & 0.5 & 0.2 \\ 0.7 & 0.5 & 0.3 & 1.4 & 0.4 \\ 0.5 & 0.7 & 0.8 & 0.6 & 1.7 \end{pmatrix}, \mathbf{A}_M = \begin{pmatrix} 1.0 & 0.7 & 0.5 & 0.4 & 0.6 \\ 0.9 & 1.5 & 0.6 & 0.5 & 0.7 \\ 0.4 & 0.5 & 1.2 & 0.6 & 0.3 \\ 0.8 & 0.6 & 0.4 & 1.4 & 0.5 \\ 0.5 & 0.8 & 0.9 & 0.7 & 1.7 \end{pmatrix}, \\ \mathbf{A}_H &= \begin{pmatrix} 1.0 & 0.7 & 0.5 & 0.4 & 0.6 \\ 0.9 & 1.3 & 0.7 & 0.5 & 0.8 \\ 0.6 & 0.7 & 1.2 & 0.8 & 0.5 \\ 0.8 & 0.6 & 0.4 & 1.4 & 0.7 \\ 0.7 & 0.8 & 0.9 & 0.6 & 1.6 \end{pmatrix} \end{aligned}$$

The parameter values are constrained to be positive, but some care is needed when the parameter matrices for synthetic data sets are chosen if the intention is to use such data for model testing. Due to specification of the model based on the Dirichlet distribution whose dimensions grow with the size of the network, they cannot be too large, especially in the case of large number of nodes. The expression for the complete data hybrid log-likelihood involves a Gamma function term evaluated at the sum of all parameter values, and large parameters cause it to become too large to compute. The parameters also cannot be too close to 0 as we are required to evaluate the Gamma function at each individual parameter value.

E Erasmus programme data: parameter estimates

E.1 K=3

The estimate of the Dirichlet parameter matrix \mathbf{A} of the 3-cluster DirSBM on the Erasmus programme data is

$$\hat{\mathbf{A}} = \begin{pmatrix} 7.019 & 1.893 & 0.410 \\ 5.325 & 2.521 & 0.535 \\ 1.347 & 0.410 & 0.142 \end{pmatrix}.$$

The expected total cluster-to-cluster exchange share and the expected node-to-node exchange share matrices are (multiplied by 100 for convenience):

$$\hat{\mathbf{V}} = \begin{pmatrix} 50.3 & 37.3 & 12.5 \\ 43.7 & 41.4 & 14.9 \\ 49.8 & 33.4 & 16.8 \end{pmatrix}, \quad \hat{\mathbf{W}} = \begin{pmatrix} 12.6 & 3.4 & 0.7 \\ 8.7 & 4.1 & 0.9 \\ 10.0 & 3.0 & 1.1 \end{pmatrix}.$$

The rows of these matrices correspond to the sending clusters, and the receiving clusters are along the columns. To give an example in the context of the Erasmus programme network, the entry $\hat{v}_{12} = 37.3$ indicates that 37.3% of students of cluster 1 collectively (which contains Germany, Spain, France, Italy and the UK) go on exchange to cluster 2 collectively (which includes countries like Austria and Belgium). The entry $\hat{w}_{31} = 10.0$ can be read as 10.0% of Irish students (cluster 3) are expected to go to Germany (cluster 1), and the same percentage is expected to go from Norway (cluster 3) to Spain (cluster 1).

E.2 K=5

Similarly, for the $K = 5$ solution,

$$\hat{\mathbf{A}} = \begin{pmatrix} 11.004 & 2.214 & 0.323 & 1.189 & 0.213 \\ 12.197 & 4.057 & 0.847 & 1.984 & 0.282 \\ 5.539 & 3.083 & 1.174 & 0.959 & 0.301 \\ 3.035 & 1.298 & 0.231 & 0.158 & 0.105 \\ 0.630 & 0.323 & 0.124 & 0.155 & 0.108 \end{pmatrix}.$$

and

$$\hat{\mathbf{V}} = \begin{pmatrix} 39.1 & 43.3 & 5.2 & 10.6 & 1.9 \\ 38.1 & 42.2 & 7.9 & 10.3 & 1.5 \\ 25.1 & 51.2 & 14.2 & 7.2 & 2.3 \\ 34.2 & 53.6 & 7.8 & 2.4 & 2.0 \\ 24.3 & 45.8 & 14.4 & 10.0 & 5.6 \end{pmatrix}, \quad \hat{\mathbf{W}} = \begin{pmatrix} 19.6 & 3.9 & 0.6 & 2.1 & 0.4 \\ 12.7 & 4.2 & 0.9 & 2.1 & 0.3 \\ 8.4 & 4.7 & 1.8 & 1.4 & 0.5 \\ 11.4 & 4.9 & 0.9 & 0.6 & 0.4 \\ 8.1 & 4.2 & 1.6 & 2.0 & 1.4 \end{pmatrix}.$$

F Bike sharing data: parameter estimates

Dirichlet parameter matrix \mathbf{A} estimate of the 4-cluster DirSBM on the bike sharing data is

$$\hat{\mathbf{A}} = \begin{pmatrix} 1.58 & 0.25 & 0.71 & 2.04 \\ 0.22 & 3.96 & 0.79 & 0.38 \\ 0.61 & 1.51 & 2.16 & 1.26 \\ 1.23 & 0.64 & 1.67 & 2.33 \end{pmatrix},$$

and the expected cluster-to-cluster exchange shares and station-to-station exchange shares matrices are

$$\hat{\mathbf{V}} = \begin{pmatrix} 36.5 & 3.7 & 18.7 & 41.1 \\ 5.9 & 62.3 & 23.2 & 8.6 \\ 12.6 & 19.4 & 46.4 & 21.6 \\ 23.2 & 7.6 & 34.3 & 34.9 \end{pmatrix}, \quad \hat{\mathbf{W}} = \begin{pmatrix} 1.6 & 0.2 & 0.7 & 2.1 \\ 0.2 & 4.5 & 0.9 & 0.4 \\ 0.5 & 1.3 & 1.9 & 1.1 \\ 1.0 & 0.5 & 1.3 & 1.8 \end{pmatrix}.$$

References

- Aicher, C., Jacobs, A. Z., and Clauset, A. (2013). Adapting the stochastic block model to edge-weighted networks. <https://doi.org/10.48550/arXiv.1305.5782>.
- Aicher, C., Jacobs, A. Z., and Clauset, A. (2014). Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248.
- Aitchison, J. (1982). The statistical analysis of compositional data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):139–160.
- Aitchison, J. (1986). *The statistical analysis of compositional data*. Chapman & Hall, Ltd., GBR.
- Alenazi, A. (2023). A review of compositional data analysis and recent advances. *Communications in Statistics - Theory & Methods*, 52(16):5535–5567.
- Baxter, M. J. (1995). Standardization and transformation in principal component analysis, with applications to archaeometry. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 44(4):513–527.
- Biernacki, C., Celeux, G., and Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725.
- Bouveyron, C., Celeux, G., Murphy, T. B., and Raftery, A. E. (2019). *Model-Based Clustering and Classification for Data Science: with Applications in R*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.
- Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16:1190–1208.
- Celeux, G. and Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics & Data Analysis*, 14(3):315–332.
- Clauset, A., Newman, M., and Moore, C. (2005). Finding community structure in very large networks. *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, 70:066111.
- Connor, R. J. and Mosimann, J. E. (1969). Concepts of independence for proportions with a generalization of the Dirichlet distribution. *Journal of the American Statistical Association*, 64(325):194–206.
- Daudin, J. J. (2011). A review of statistical models for clustering networks with an application to a PPI network. *Journal de la Societe Francaise de Statistique*, 152(2):111–125.
- Daudin, J. J., Picard, F., and Robin, S. (2008). A mixture model for random graphs. *Statistics & Computing*, 18:173–183.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.
- Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., and Weingessel, A. (2009). E1071: Misc functions of the department of statistics (e1071), tu wien. *R package version 1.5-24*.
- El Haj, A., Slaoui, Y., Louis, P.-Y., and Khraibani, Z. (2022). Estimation in a binomial stochastic blockmodel for a weighted graph by a variational expectation maximization algorithm. *Communication in Statistics - Simulation & Computation*, 51:4450–4469.
- Filzmoser, P., Hron, K., and Templ, M. (2018). *Applied Compositional Data Analysis. With Worked Examples in R*. Springer Series in Statistics. Springer.
- Frigyik, A. B., Kapila, A., and Gupta, M. R. (2010). Introduction to the Dirichlet distribution and related processes. <https://api.semanticscholar.org/CorpusID:8763665>.
- Gadár, L., Kosztyán, Z., Telcs, A., and Abonyi, J. (2020). A multilayer and spatial description of the Erasmus mobility network. *Scientific Data*, 7.

- Godichon-Baggioni, A., Maugis-Rabusseau, C., and Rau, A. (2017). Clustering transformed compositional data using k-means, with applications in gene expression and bicycle sharing system data. *Journal of Applied Statistics*, 46.
- Golub, G. H. and Van Loan, C. F. (1996). *Matrix computations*. Johns Hopkins University Press, USA, 3 edition.
- Greenacre, M. (2021). Compositional data analysis. *Annual Review of Statistics & its Applications*, 8(1):271–299.
- Guimerà, R., Stouffer, D. B., Sales-Pardo, M., Leicht, E. A., Newman, M. E. J., and Amaral, L. A. N. (2010). Origin of compartmentalization in food webs. *Ecology*, 91(10):2941–2951.
- Ha, M. J., Kim, J., Galloway-Peña, J., Do, K.-A., and Peterson, C. B. (2020). Compositional zero-inflated network estimation for microbiome data. *BMC Bioinformatics*, 21:1–20.
- Hartigan, J. A. and Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.
- Hledík, J. and Rastelli, R. (2023). A dynamic network model to measure exposure concentration in the Austrian interbank market. *Statistical Methods & Applications*.
- Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: first steps. *Social Networks*, 5(2):109–137.
- Hubert, L. J. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.
- Karrer, B. and Newman, M. E. J. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E*, 83:016107.
- Lee, C. and Wilkinson, D. J. (2019). A review of stochastic block models and extensions for graph clustering. *Applied Network Science*, 4(1).
- Léger, J.-B. (2016). Blockmodels: A R-package for estimating in latent block model and stochastic block model, with various probability functions, with or without covariates. <https://doi.org/10.48550/arXiv.1602.07587>.
- Léger, J.-B., Vacher, C., and Daudin, J. J. (2013). Detection of structurally homogeneous subsets in graphs. *Statistics & Computing*, 24.
- Ludkin, M. (2020). Inference for a generalised stochastic block model with unknown number of blocks and non-conjugate edge models. *Computational Statistics & Data Analysis*, 152:107051.
- Mariadassou, M., Robin, S., and Vacher, C. (2010). Uncovering latent structure in valued graphs: a variational approach. *The Annals of Applied Statistics*, 4.
- Marino, M. F. and Pandolfi, S. (2022). Hybrid maximum likelihood inference for stochastic block models. *Computational Statistics & Data Analysis*, 171:107449.
- Melnykov, V., Sarkar, S., and Melnykov, Y. (2020). On finite mixture modeling and model-based clustering of directed weighted multilayer networks. *Pattern Recognition*, 112:107641.
- Ng, T. L. J. and Murphy, T. B. (2021). Weighted stochastic block model. *Statistical Methods & Applications*, 30(5):1365–1398.
- Nowicki, K. and Snijders, T. A. B. (2001). Estimation and prediction for stochastic block structures. *Journal of the American Statistical Association*, 96(455):1077–1087.
- Pal, S. and Heumann, C. (2022). Clustering compositional data using dirichlet mixture model. *PLOS ONE*, 17(5):1–24.
- Pons, P. and Latapy, M. (2006). Computing communities in large networks using random walks. *Journal of Graph Algorithms & Applications*, 10:191–218.
- R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing.
- Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

- Rastelli, R. and Fop, M. (2020). A stochastic block model for interaction lengths. *Advances in Data Analysis & Classification*, 14.
- Salter-Townshend, M., White, A., Gollini, I., and Murphy, T. B. (2012). Review of statistical network analysis: models, algorithms, and software. *Statistical Analysis & Data Mining*, 5:243–264.
- Snijders, T. A. and Nowicki, K. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100.
- Transport for London (2016). <https://cycling.data.tfl.gov.uk>.
- Tsagris, M. and Stewart, C. (2018). A Dirichlet regression model for compositional data with zeros. *Lobachevskii Journal of Mathematics*, 39:398–412.
- Van den Boogaart, K. and Tolosana-Delgado, R. (2008). “compositions”: a unified R package to analyze compositional data. *Computers & Geosciences*, 34:320–338.
- von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics & Computing*, 17(4):395–416.
- Wang, Y. J. and Wong, G. Y. C. (1987). Stochastic Blockmodels for Directed Graphs. *Journal of the American Statistical Association*, 82:8–19.
- Wasserman, S. and Anderson, C. (1987). Stochastic a posteriori blockmodels: construction and assessment. *Social Networks*, 9(1):1–36.
- Wu, C. F. J. (1983). On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1):95 – 103.
- Yuan, H., He, S., and Deng, M. (2019). Compositional data network analysis via lasso penalized D-trace loss. *Bioinformatics*, 35(18):3404–3411.
- Zanghi, H., Picard, F., Miele, V., and Ambroise, C. (2010). Strategies for online inference of model-based clustering in large and growing networks. *The Annals of Applied Statistics*, 4(2).

Daniele Durante

Material list:

Durante, D. (2024) Bayesian nonparametric stochastic block modeling of criminal networks. WG Slides.

Legramanti, S., Rigon, T., Durante, D. and Dunson, D.B. (2022) Extended stochastic block models with application to criminal networks. *Annals of Applied Statistics*. 16, 2369—2395.

Lu, C., Durante, D., and Friel, N.B. (2024). Zero-inflated stochastic block modeling of efficiency-security tradeoffs in weighted criminal networks.

Bayesian nonparametric stochastic block modeling of criminal networks – Useful additional references.

Bayesian nonparametric stochastic block modeling of criminal networks

Working Group on Model-Based Clustering Summer Session



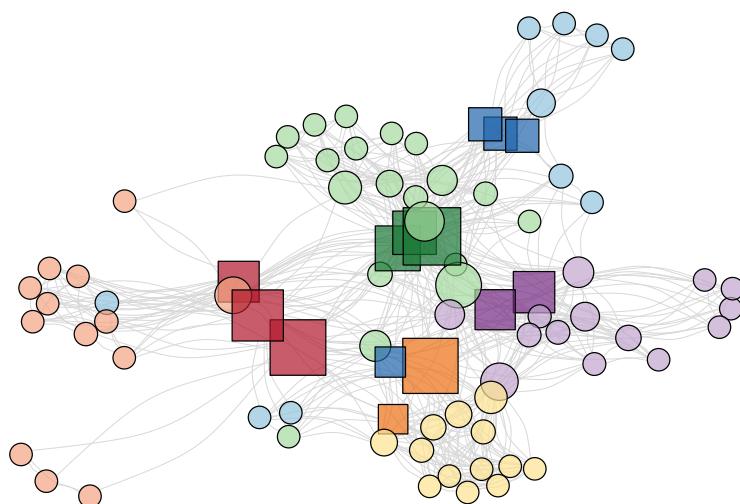
Funded by ERC, project number: 101116718 (NEMESIS)

26.07.2024

Daniele Durante

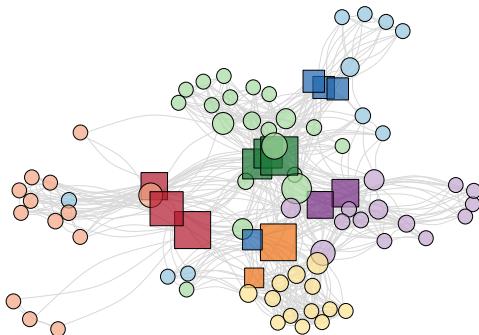
daniele.durante@unibocconi.it

Criminal networks



Data refer to a large law-enforcement operation, named *Operazione Infinito*, aimed at disrupting the 'Ndrangheta mafia in Lombardy (Italy). This criminal organization is a key example of a deeply rooted, highly structured and hard to untangle covert architecture with disruptive impact, both local and international.

Stylized facts about organized crime

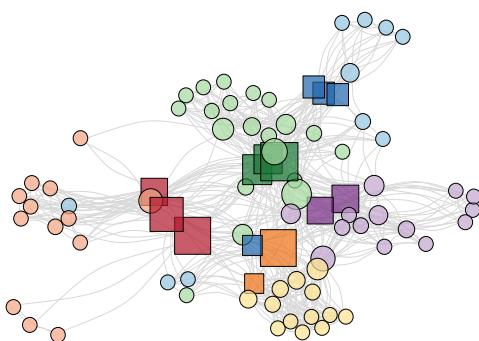


□ **Question:** To what extent is it possible to reconstruct the architecture of a criminal organization from the observed (often noisy) patterns of interaction among its criminals?

Criminology relies on descriptive analyses and community detection algorithms, but to answer this question we require **models** which **incorporate (and learn) structure**.

- **Redundancy (resilience):** Presence of criminals with similar connectivity patterns in the network → **group structures** [stochastic equivalence]
- **Efficiency–Security:** Connectivity patterns that facilitate criminal activities while preserving security → **complex/noisy group interactions**
- **Hierarchies:** 'Ndrangheta revolves around blood family relations, which are also aggregated at the territorial level → **inclusion of layer information**

From a statistical modeling perspective . . .



The previously-mentioned stylized facts suggest that **block structures** in criminal networks can be a key to reveal the complex **modular architectures** of organized crime via formal inference on **shared connectivity patterns** among interacting suspects, as observed during investigations.

Such criminal networks exhibit key features that require **careful innovations**.

- criminal networks are subject to **measurement errors** (often structured).
- exhibit complex combinations of an unknown number of **block structures**, covering community, core–periphery and disassortative behaviors.
- Informative **node attributes**, often defining **layers**, should be leveraged.

Stochastic block models [formulation]

Given a $V \times V$ symmetric adjacency matrix \mathbf{Y} , with elements $y_{vu} = y_{uv} = 1$ if nodes v and u are connected, and $y_{vu} = y_{uv} = 0$ otherwise, SBM assumes that the presence of an edge between two generic nodes v and u depends only on the group memberships z_v and z_u of such nodes and on block probabilities.

Denoting with θ_{hk} the **block probability** between group h and group k , the **likelihood** for the adjacency matrix \mathbf{Y} is

$$\begin{aligned} p(\mathbf{Y} | \mathbf{z}, \Theta) &= \prod_{v=2}^V \prod_{u=1}^{v-1} \theta_{z_v z_u}^{y_{vu}} (1 - \theta_{z_v z_u})^{1-y_{vu}} \\ &= \prod_{h=1}^H \prod_{k=1}^h \theta_{hk}^{m_{hk}} (1 - \theta_{hk})^{\bar{m}_{hk}}, \end{aligned}$$

where m_{hk} and \bar{m}_{hk} are the number of edges and non-edges between groups h and k . The θ_{hk} 's are usually given **independent Beta(a, b) priors** and can be integrated out in $p(\mathbf{Y} | \mathbf{z}, \Theta)$ to obtain the **marginal likelihood** $p(\mathbf{Y} | \mathbf{z})$.

$$p(\mathbf{Y} | \mathbf{z}) = \prod_{h=1}^H \prod_{k=1}^h \frac{B(a + m_{hk}, b + \bar{m}_{hk})}{B(a, b)}.$$

Variants of SBM

SBM variants mostly differ in the choice of the **prior for $\mathbf{z} = (z_1, \dots, z_V)^\top$** .

Key examples [1. Nowicki and Snijders (2001), 2. Geng et al. (2019), 3. Kemp et al. (2006)]

- 1 Classical SBM. $\mathbf{z} \sim \text{Dirichlet-Multinomial}(\bar{H}, \beta)$. \bar{H} finite and fixed.
- 2 MFM-SBM. $\mathbf{z} \sim \text{Dirichlet-Multinomial}(\bar{H}, \beta)$. \bar{H} random on \mathbb{N} .
- 3 Infinite relational SBM. $\mathbf{z} \sim \text{CRP}(\alpha)$. \bar{H} infinite.

Note that **these are all Gibbs-type priors** [Gnedin and Pitman (2005), De Blasi et al. (2013)]. A probability mass function $p(\mathbf{z})$ is of **Gibbs-type** if and only if

$$p(\mathbf{z}) = \mathcal{W}_{V,H} \prod_{h=1}^H (1 - \sigma)_{n_h - 1},$$

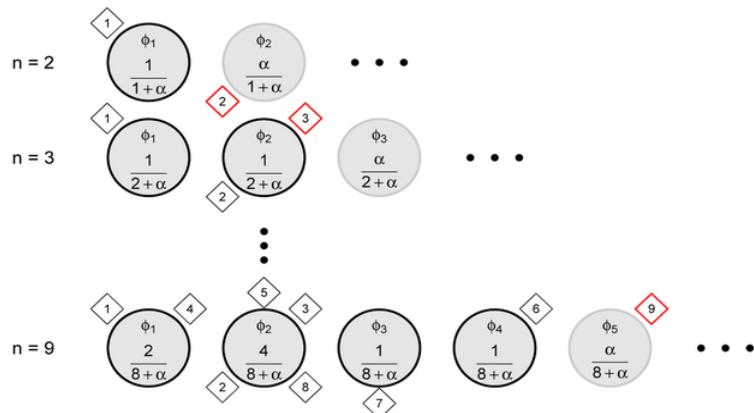
where n_h denotes the **number of nodes in cluster h** , $\sigma < 1$ is the **discount parameter** and $\{\mathcal{W}_{V,H} : 1 \leq H \leq V\}$ is a **collection of non-negative weights** satisfying the recursion $\mathcal{W}_{V,H} = (V - H\sigma)\mathcal{W}_{V+1,H} + \mathcal{W}_{V+1,H+1}$, with $\mathcal{W}_{1,1} = 1$

- ESBM. $\mathbf{z} \sim \text{Gibbs-type}(\mathcal{W}_{V,H}, \sigma)$. \bar{H} finite-fixed, finite-random, infinite.

Gibbs-type priors

Under **Gibbs-type priors**, the group membership indicators \mathbf{z} can be generated in a sequential and interpretable manner.

$$\text{pr}(z_{V+1} = h \mid \mathbf{z}) \propto \begin{cases} \mathcal{W}_{V+1, H}(n_h - \sigma) & \text{for } h = 1, \dots, H, \\ \mathcal{W}_{V+1, H+1} & \text{for } h = H + 1. \end{cases}$$



Credit: <https://europepmc.org/article/med/31217637>

Gibbs-type priors

Relevant examples of **Gibbs-type priors**.

Dirichlet–multinomial, DM (\bar{H} finite and fixed)

- $\text{pr}(z_{V+1} = h \mid \mathbf{z}) \propto n_h + \beta$ for $h = 1, \dots, H$
- $\text{pr}(z_{V+1} = H + 1 \mid \mathbf{z}) \propto \beta(\bar{H} - H)\mathbb{1}(H \leq \bar{H})$

Pitman–Yor process, PY. When $\sigma = 0 \rightarrow$ Dirichlet process, DP (\bar{H} infinite)

- $\text{pr}(z_{V+1} = h \mid \mathbf{z}) \propto n_h - \sigma$ for $h = 1, \dots, H$
- $\text{pr}(z_{V+1} = H + 1 \mid \mathbf{z}) \propto \alpha + H\sigma$

Gnedin process, GN (\bar{H} finite and random). **New in SBM!**

- $\text{pr}(z_{V+1} = h \mid \mathbf{z}) \propto (n_h + 1)(V - H + \gamma)$ for $h = 1, \dots, H$
- $\text{pr}(z_{V+1} = H + 1 \mid \mathbf{z}) \propto H^2 - H\gamma$

Layer-assisted Gibbs-type priors

To include layer information $x_v \in \{1, \dots, L\}$, $v = 1, \dots, V$ we can rely on the **PPM** structure of Gibbs-type priors [Müller et al. (2011)]. Replace $p(\mathbf{z})$ with

$$p(\mathbf{z} | \mathbf{x}) \propto \mathcal{W}_{V,H} \prod_{h=1}^H p(\mathbf{x}_h)(1 - \sigma)_{n_h - 1},$$

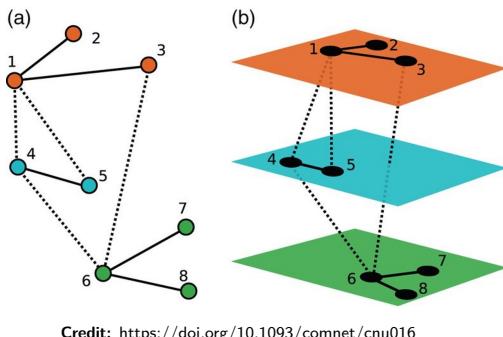
where $\mathbf{x} = (x_1, \dots, x_V)$, whereas $\mathbf{x}_h = \{\mathbf{x}_v : z_v = h\}$ are the layer indicators for the nodes in cluster h . The function $p(\mathbf{x}_h)$ controls the contribution of \mathbf{x} to cluster cohesion by favoring groups that are homogeneous with respect to layer.

For $p(\cdot)$, we use the **Dirichlet–Multinomial cohesion function** which yields to

$$\text{pr}(z_{V+1} = h | x_{V+1}, \mathbf{z}, \mathbf{x}) \propto \begin{cases} \frac{n_h x_{V+1} + \alpha_{x_{V+1}}}{n_h + \alpha_0} \cdot \mathcal{W}_{V+1,H}(n_h - \sigma) & \text{for } h = 1, \dots, H, \\ \frac{\alpha_{x_{V+1}}}{\alpha_0} \cdot \mathcal{W}_{V+1,H+1} & \text{for } h = H+1. \end{cases}$$

where $n_h x_{V+1}$ is the number of nodes in cluster h with the same layer indicator $l = x_{V+1}$ as node $V+1$, and n_h is the number of nodes in cluster h . Note that $p(\mathbf{z} | \mathbf{Y}, \mathbf{x}) \propto p(\mathbf{z} | \mathbf{x})p(\mathbf{Y} | \mathbf{z}) \propto [p(\mathbf{z})p(\mathbf{x} | \mathbf{z})]p(\mathbf{Y} | \mathbf{z})$ [error-prone layers].

Partially-exchangeable SBM [motivation]



Credit: <https://doi.org/10.1093/comnet/cnu016>

The previous layer-assisted Gibbs-type prior requires the choice of a cohesion function and, in general, does not necessarily guarantee a consistent sequence of random partitions on the growing network size.

Solution: Random partitions induced by H-NRMI [partial exchangeability].

Intuitively

Intuitively, we are assuming that every node $v_l = 1, \dots, V_l$ within each layer $l = 1, \dots, L$ has a latent attribute w_{l,v_l} and, on the joint distribution of such attributes, we impose a partially exchangeable structure. E.g., if $L = 2$,

$$\begin{aligned} (\textcolor{red}{w_{11}}, \textcolor{red}{w_{12}}, \textcolor{blue}{w_{21}}, \textcolor{blue}{w_{22}}, \textcolor{blue}{w_{23}}) &\stackrel{d}{=} (\textcolor{red}{w_{12}}, \textcolor{red}{w_{11}}, \textcolor{blue}{w_{22}}, \textcolor{blue}{w_{23}}, \textcolor{blue}{w_{21}}) \\ (\textcolor{red}{w_{11}}, \textcolor{red}{w_{12}}, \textcolor{blue}{w_{21}}, \textcolor{blue}{w_{22}}, \textcolor{blue}{w_{23}}) &\neq (\textcolor{red}{w_{11}}, \textcolor{blue}{w_{23}}, \textcolor{blue}{w_{21}}, \textcolor{blue}{w_{22}}, \textcolor{red}{w_{12}}) \end{aligned}$$

Partially-exchangeable SBM

The above structure is present in, e.g., H-NRMIS [Camerlenghi et al. (2019)].

$$\begin{aligned} w_{11}, \dots, w_{lV_l} \mid \tilde{p}_l &\stackrel{iid}{\sim} \tilde{p}_l \quad l = 1, \dots, L \\ \tilde{p}_1, \dots, \tilde{p}_L \mid \tilde{p}_0 &\stackrel{iid}{\sim} \text{NRMI}(\rho, c, \tilde{p}_0) \\ \tilde{p}_0 &\sim \text{NRMI}(\rho_0, c_0, P_0) \end{aligned}$$

Since \tilde{p}_j and \tilde{p}_0 are almost surely discrete (w_{l,v_l}) has ties both within and across layers. I.e., $\text{pr}(w_{l,v_l} = w_{l',v'_{l'}}) > 0$ for $l, l' \in [L]$, $v_l \in [V_l]$ and $v'_{l'} \in [V'_{l'}]$. If we consider the random partition of $[V]$ induced by these ties — i.e., $z_{l,v_l} = h$ if and only if $w_{l,v_l} = w_h^*$ — its distribution is actually $\mathbf{z} \sim \text{pEPPF}(\rho, \rho_0, c, c_0)$.

Replace $\mathbf{z} \sim \text{Gibbs-type-}x(\mathcal{W}_{V,H}, \sigma, p(\mathbf{x}))$ with

$$\mathbf{z} \sim \text{pEPPF}(\rho, \rho_0, c, c_0)$$

where $\text{pEPPF}(\rho, \rho_0, c, c_0)$ is the partially exchangeable partition probability function induced by a hierarchical normalized completely random measure (H-NRMI) with parameters ρ, ρ_0, c, c_0 .

Partially-exchangeable SBM [CRF]

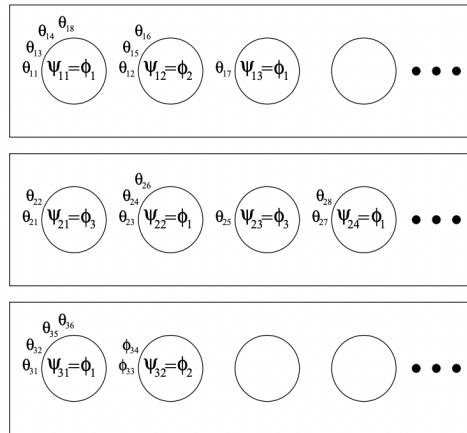


Figure 2: A depiction of a Chinese restaurant franchise. Each restaurant is represented by a rectangle. Customers (θ_{ji} 's) are seated at tables (circles) in the restaurants. At each table a dish is served. The dish is served from a global menu (ϕ_k), whereas the parameter ψ_{jt} is a table-specific indicator that serves to index items on the global menu. The customer θ_{ji} sits at the table to which it has been assigned in (24).

Credit: <https://www.jstor.org/stable/27639773>

Posterior computation and inference

Algorithm 1: Gibbs sampler for ESBM

At each iteration, update the cluster assignments as follows:

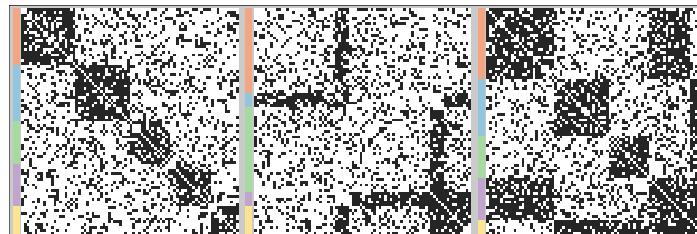
For $v = 1, \dots, V$ **do**:

1. Remove node v from the network;
 2. If the cluster which contained node v becomes empty, discard it (so that clusters $1, \dots, H^-$ are non-empty);
 3. Sample z_v from the categorical variable with $\text{pr}(z_v = h \mid \mathbf{Y}, \mathbf{X}, \mathbf{z}_{-v}) \propto \text{pr}(z_v = h \mid \mathbf{X}, \mathbf{z}_{-v}) p(\mathbf{Y} \mid z_v = h, \mathbf{z}_{-v})$ for each $h = 1, \dots, H^- + 1$, where $p(\mathbf{Y} \mid z_v = h, \mathbf{z}_{-v})$ is the closed-form Beta-Binomial likelihood evaluated at $(z_v = h, \mathbf{z}_{-v})$, whereas $\text{pr}(z_v = h \mid \mathbf{X}, \mathbf{z}_{-v})$ are the closed-form probabilities of the chosen urn scheme.
-

To fully [exploit the posterior and perform inference directly on partitions](#), we adapt the **decision-theoretic approach** of Wade and Ghahramani [2018].

- **Point estimate of \mathbf{z} .** Partition with lowest posterior averaged VI distance from the other clusterings $\hat{\mathbf{z}} = \arg \min_{\mathbf{z}'} \mathbb{E}_{\mathbf{z}}[\text{VI}(\mathbf{z}, \mathbf{z}') \mid \mathbf{Y}]$.
- **$(1 - \alpha)$ credible ball around $\hat{\mathbf{z}}$.** Set of all those partitions having less than a threshold distance from $\hat{\mathbf{z}}$, with the threshold chosen to minimize the size of the ball while ensuring it contains at least $(1 - \alpha)$ posterior probability.
- **Model comparison.** $\mathcal{B}_{\mathcal{M}, \mathcal{M}'} = [\sum_{\mathbf{z}} p(\mathbf{Y} \mid \mathbf{z}) p(\mathbf{z} \mid \mathcal{M})][\sum_{\mathbf{z}} p(\mathbf{Y} \mid \mathbf{z}) p(\mathbf{z} \mid \mathcal{M}')]^{-1}$. Can use harmonic mean estimator, but we prefer WAIC.

Simulation ESBM [performance]



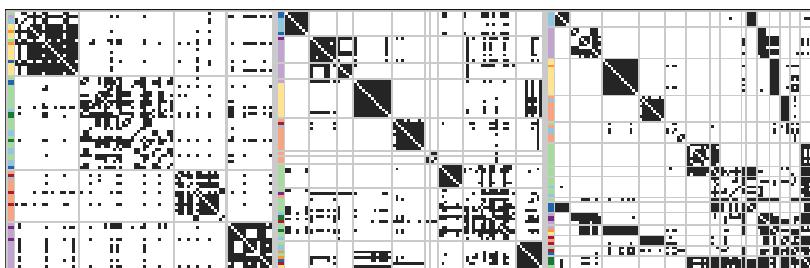
SCENARIO	WAIC			$\mathbb{E}[\text{VI}(\mathbf{z}, \mathbf{z}_0) \mid \mathbf{Y}]$			H	$\text{VI}(\hat{\mathbf{z}}, \mathbf{z}_b)$		
	1	2	3	1	2	3		1	2	3
[UNSUP] DM	3551.0	3559.8	3540.3	0.420	0.746	0.517	8 [7,8]	6 [5,7]	6 [5,6]	0.702 0.971 0.691
[UNSUP] DP	3550.7	3559.5	3540.4	0.414	0.736	0.514	7 [7,8]	6 [5,7]	6 [5,6]	0.694 0.955 0.692
[UNSUP] PY	3551.4	3559.0	3540.2	0.376	0.708	0.498	7 [6,9]	6 [5,7]	6 [5,6]	0.696 0.884 0.645
[UNSUP] GN	3550.1	3554.3	3535.9	0.292	0.642	0.455	5 [5,6]	5 [5,5]	5 [5,5]	0.592 0.827 0.601
[SUP] GN	3521.3	3510.4	3515.2	0.041	0.139	0.122	5 [5,5]	5 [5,5]	5 [5,5]	0.139 0.297 0.284

SCENARIO	H			$\text{VI}(\hat{\mathbf{z}}, \mathbf{z}_0)$			ERROR [EST]		
	1	2	3	1	2	3	1	2	3
[UNSUP] ESBM (GN)	5	5	5	0.126	0.404	0.374	0.030	0.028	0.031
[UNSUP] Louvain	4	4	3	0.303	2.904	0.810	0.040	0.124	0.051
[UNSUP] Spectral	4	4	3	0.557	2.806	0.810	0.045	0.132	0.051
[UNSUP] greed (SBM)	4	5	4	0.412	0.267	0.477	0.044	0.027	0.028
[SUP] ESBM (GN)	5	5	5	0.000	0.159	0.000	0.022	0.026	0.023
[SUP] JCDC ($w_n = 1.5$)	4	4	3	0.303	2.024	0.703	0.040	0.112	0.047

Application [performance]

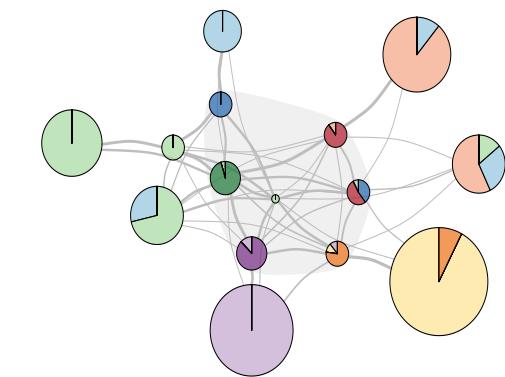
Comparison among Gibbs-type priors, and against popular competitors

	WAIC		H		VI(\hat{z}, z_b)	
	UNSUP	SUP	UNSUP	SUP	UNSUP	SUP
DM	1228.5	1199.0	14 [14,15]	15 [15,15]	0.279	0.163
DP	1256.2	1198.5	14 [14,14]	15 [15,16]	0.219	0.279
PY	1279.9	1225.5	14 [14,14]	15 [14,15]	0.299	0.199
GN	1204.7	1194.1	15 [15,15]	15 [15,16]	0.317	0.221

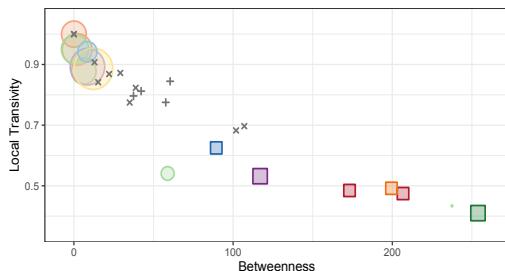


Within the Gibbs-type class, the Gnedin process prior has better performance and assisting block-modeling with *locali-role* information further improves inference. Moreover, ESBM learn block patterns revealing a complex hierarchical structure of the organization, mostly hidden from state-of-the-art alternatives.

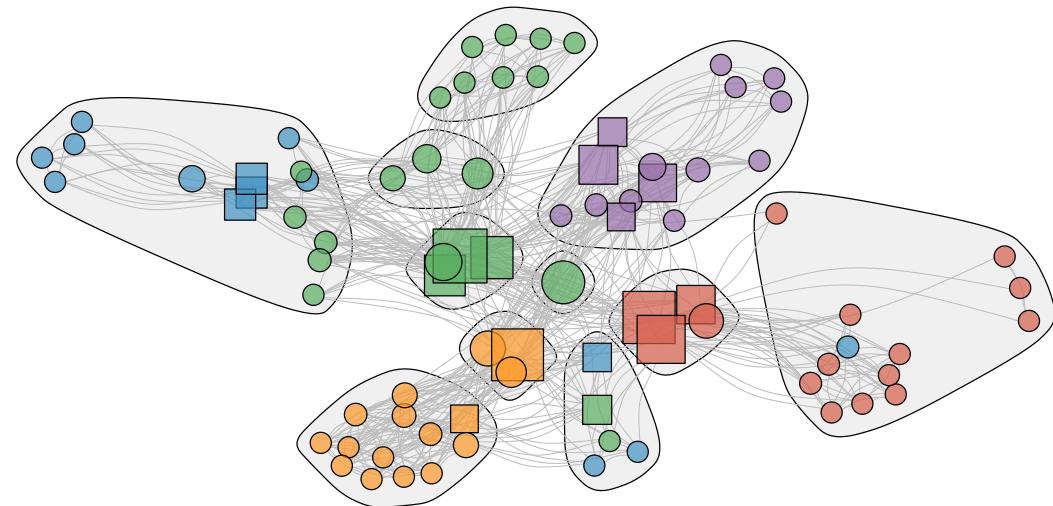
Application [interpretations]



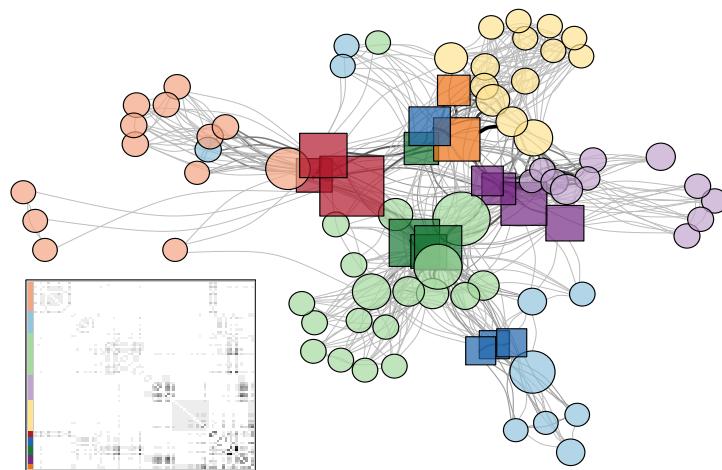
The affiliates' groups typically exhibit **community patterns** and connect to the **hidden core** mainly through the **bosses** of the corresponding *locale*, which in turn display **weak assortativity** in the higher-level coordinating architecture among bosses of different *locali*.



Application [PEx–SBM]



Weighted criminal networks



Inclusion of **weighted edges** (counts of meetings co-attended) might give new opportunities to identify **complex patterns in the excess of zero ties** in each block (**security strategy**) relative to the distribution of the observed weighted ties within that block (**efficiency strategy**).

Zero-inflated stochastic block models

We define the entries $w_{vu} \in \mathbb{N}$ of the $V \times V$ symmetric weighted adjacency matrix \mathbf{W} , as $w_{vu} = (1 - w_{0,vu}) \cdot w_{1,vu}$ for $v, u = 1, \dots, V$, with

$$(w_{0,vu} | z_v, z_u, \Theta) \sim \text{Bern}(\theta_{z_v z_u}), \quad \text{independently for } v, u = 1, \dots, V,$$

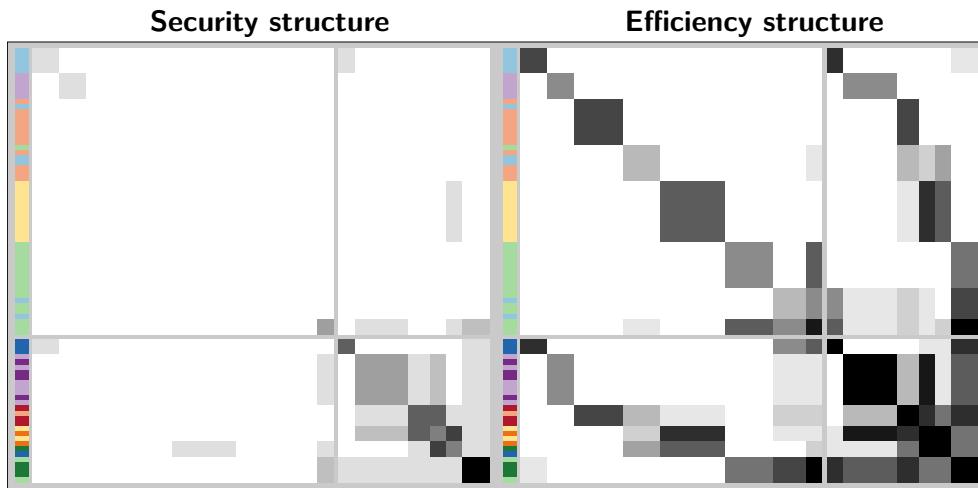
$$(w_{1,vu} | z_v, z_u, \Lambda) \sim \text{Pois}(\lambda_{z_v z_u}), \quad \text{independently for } v, u = 1, \dots, V,$$

and assign to the entries in Θ and Λ , independent $\text{Beta}(a, b)$ and $\text{Gamma}(\nu, \gamma)$ priors, respectively. In this way $p(\mathbf{W}_0 | \mathbf{z})$ and $p(\mathbf{W}_1 | \mathbf{z})$ correspond to products of independent **Beta–Binomial** and **Negative–Binomial** likelihoods, respectively.

Interpretation: Intuitively, \mathbf{W}_1 measures the observed or latent strength of interaction, whereas \mathbf{W}_0 refers to the decision to either obscure or not ties
→ This may unveil structured secrecy strategies.

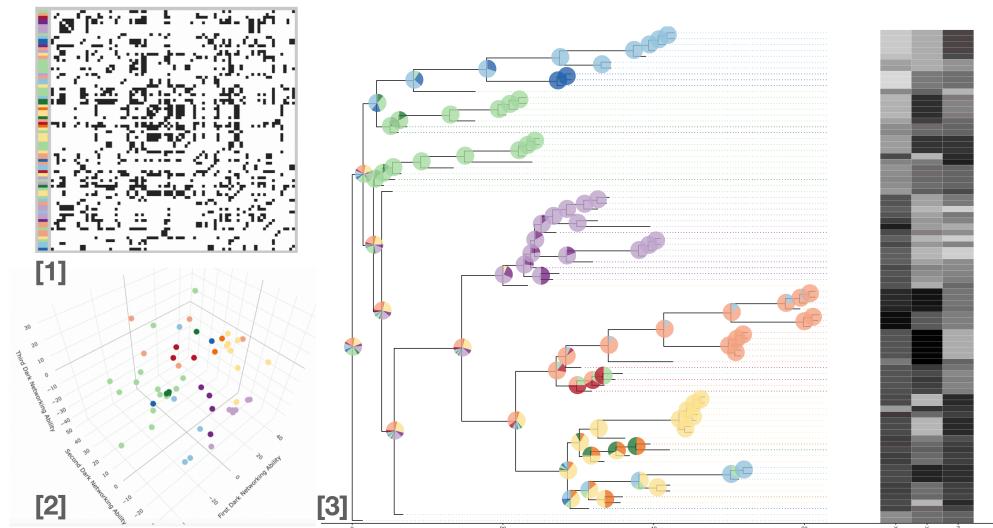
Inference: We can again consider either a **Gibbs-type** or **pEPPF** prior for \mathbf{z} and slightly adapt the previous collapsed Gibbs–sampler via data augmentation step.

Application [results]



Using a Gnedenko process prior for \mathbf{z} , we learn similar blocks as those inferred under the binary network. However, inclusion of weighted ties, allows also to detect **block structures among bosses of different locali with unusual excess of zeros** that suggest strategies aimed at balancing security–efficiency.

Phylogenetic latent space model [main idea]

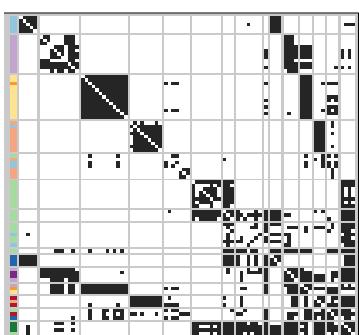


Can we learn the evolutionary history of the criminal organization from Y?

Some remarks and references

Criminal networks provide a fundamental field of application where the advancements in modern statistics can have a major societal impact. However, despite the relevance of such studies, this area has been somewhat overlooked. The goal of this talk was to illustrate the potentials of creating this bridge.

Some directions of future research: Phylogenetic latent space models — Extension to multiplex criminal network data — Inclusion of degree-correction



References

- Legramanti, Rigon, Durante and Dunson. [2022]. *Extended stochastic block models with application to criminal networks*. Annals of Applied Statistics
- Durante, Gaffi, Lijoi and Pruenster. [2024+]. *Partially-exchangeable multilayer stochastic block models*. In review.
- Lu, Durante and Friel. [2024+]. *Zero-inflated stochastic block modeling of efficiency-security tradeoffs in weighted criminal networks*. In review.

EXTENDED STOCHASTIC BLOCK MODELS WITH APPLICATION TO CRIMINAL NETWORKS

BY SIRIO LEGRAMANTI^{1,a}, TOMMASO RIGON^{2,c}, DANIELE DURANTE^{1,b} AND DAVID B. DUNSON^{3,d}

¹*Department Decision Sciences and Institute for Data Science and Analytics, Bocconi University,*
^asirio.legramanti@unibocconi.it, ^bdaniele.durante@unibocconi.it

²*Department of Economics, Management and Statistics, University of Milano-Bicocca,* ^ctommaso.rigon@unimib.it

³*Department of Statistical Science, Duke University,* ^ddunson@duke.edu

Reliably learning group structures among nodes in network data is challenging in several applications. We are particularly motivated by studying covert networks that encode relationships among criminals. These data are subject to measurement errors, and exhibit a complex combination of an unknown number of core-periphery, assortative and disassortative structures that may unveil key architectures of the criminal organization. The coexistence of these noisy block patterns limits the reliability of routinely-used community detection algorithms, and requires extensions of model-based solutions to realistically characterize the node partition process, incorporate information from node attributes, and provide improved strategies for estimation and uncertainty quantification. To cover these gaps, we develop a new class of extended stochastic block models (ESBM) that infer groups of nodes having common connectivity patterns via Gibbs-type priors on the partition process. This choice encompasses many realistic priors for criminal networks, covering solutions with fixed, random and infinite number of possible groups, and facilitates the inclusion of node attributes in a principled manner. Among the new alternatives in our class, we focus on the Gneden process as a realistic prior that allows the number of groups to be finite, random and subject to a reinforcement process coherent with criminal networks. A collapsed Gibbs sampler is proposed for the whole ESBM class, and refined strategies for estimation, prediction, uncertainty quantification and model selection are outlined. The ESBM performance is illustrated in realistic simulations and in an application to an Italian mafia network, where we unveil key complex block structures, mostly hidden from state-of-the-art alternatives.

1. Introduction. Network data are ubiquitous in modern applications, and there is a recurring interest in block structures defined by groups of nodes that share similar connectivity patterns (e.g., Fortunato and Hric (2016)). Our focus is on studying networks of individuals involved in organizing crime. In this setting, it is of considerable interest to infer shared connectivity patterns among different suspects, based on data provided by investigations, in order to obtain key insights into the hierarchical structure of criminal organizations (e.g., Campana (2016), Campana and Varese (2022), Diviák (2022), Faust and Tita (2019)).

The relevance of this endeavor has motivated an increasing shift in modern forensic studies away from classical descriptive analyses of criminal networks (e.g., Agreste et al. (2016), Carley, Lee and Krackhardt (2002), Cavallaro et al. (2020), Grassi et al. (2019), Krebs (2002), Malm and Bichler (2011), Morselli (2009)), and toward studying more complex group structures which involve the monitored suspects (e.g., Calderoni, Brunetto and Piccardi (2017), Calderoni and Piccardi (2014), Ferrara et al. (2014), Liu et al. (2018), Magalingam, Davis and Rao (2015), Sangkaran, Abdullah and Jhanjhi (2020)). These contributions have provided

Received January 2021; revised November 2021.

Key words and phrases. Bayesian nonparametrics, Gibbs-type prior, network, product partition model.

valuable initial insights into the structure and functioning of several criminal organizations. However, the focus has been on classical community detection algorithms (Blondel et al. (2008), Girvan and Newman (2002), Newman and Girvan (2004), Newman (2006)), which infer groups of criminals characterized by dense within-block connectivity and sparser connections between different blocks (Fortunato and Hric (2016)). Such approaches are overly simplified and ignore other fundamental block structures, such as core-periphery, disassortative and weak community patterns (e.g., Fortunato and Hric (2016)). These more nuanced structures are inherent to criminal organizations, which exhibit an intricate combination of vertical and horizontal hierarchies of block interactions (Catino (2014), Le (2012), Morselli, Giguère and Petit (2007), Paoli (2007)). Disentangling such complex architectures is fundamental to inform preventive and repressive operations. However, this task requires improved methods combined with more realistic representations of criminal networks that incorporate a broader set of recurring block structures, beyond assortative communities.

An initial strategy for addressing the above objectives is to consider spectral clustering algorithms (von Luxburg (2007)) and stochastic block models (Holland, Laskey and Leinhardt (1983), Nowicki and Snijders (2001)). Both methods learn more general block architectures in network data, and hence, despite their limited use in forensic studies, are expected to unveil criminal structures currently hidden to community detection algorithms. Nonetheless, as clarified in Sections 1.1–1.2, several aspects of criminal network studies still require careful statistical innovations. A crucial one is the coexistence of several community, core-periphery and disassortative architectures whose number, size and structure are unknown and partially obscured by the measurement errors arising from the investigations. To ensure accurate learning in these challenging settings, it is fundamental to rely on an extended, yet interpretable, class of model-based solutions encompassing a variety of flexible mechanisms for the formation of suspect groups. Such processes should also allow structured inclusion of external information and facilitate the adoption of principled methods for estimation, prediction, uncertainty quantification and model selection, within a single realistic modeling framework.

1.1. The Infinito network. Our motivation is drawn from a large law-enforcement operation, named *Operazione Infinito*, that was conducted in Italy from 2007 to 2009 for disentangling and disrupting the core structure of the 'Ndrangheta mafia in Lombardy, north of Italy. According to the pretrial detention order produced by the preliminary investigation judge of Milan,¹ such a criminal organization, also referred to as *La Lombardia*, is a key example of a deeply rooted, highly structured and hard to untangle covert architecture with a disruptive and pervasive impact, both locally and internationally (Catino (2014), Paoli (2007)). This motivates our efforts to provide an improved understanding of its hidden hierarchical structures via innovative block modeling of the relationships among its monitored affiliates.

The raw data can be found at <https://sites.google.com/site/ucinetsoftware/datasets/covert-networks> and comprise information on the coparticipation of 156 suspects at 47 monitored summits of the criminal organization, as reported in the judicial acts¹ that were issued upon request by the prosecution. Consistent with our main goal of shedding light on the internal structure of *La Lombardia* via inference on the block connectivity patterns among its affiliates, we focus on the reduced set of 118 suspects that attended at least one summit and were classified in the judicial acts as members of this specific criminal organization. Since only 18% of these affiliate pairs coattended at least one of the summits and, among them, just 5% coparticipated in more than one meeting, we consider here the binary adjacency matrix indicating the presence or absence of a coattendance in at least one of the monitored summits.

¹Tribunale di Milano, 2011. Ordinanza di applicazione di misura coercitiva con mandato di cattura — art. 292 c.p.p. (Operazione Infinito). Ufficio del giudice per le indagini preliminari (in Italian).

Due to the sparse and almost-binary form of the original counts, this dichotomization leads to a negligible loss of information and is beneficial in reducing the noise that may arise from investigations of multiple summits. Moreover, because of the highly regulated 'Ndrangheta coordinating processes (Catino (2014), Paoli (2007)), the coattendance of at least one summit is arguably sufficient to declare the presence of a connection among two affiliates.

More problematic is the possible presence of false negatives, which may arise in such studies as a result of covering strategies implemented by the criminal organization to carefully balance the tradeoff between efficiency and security (Morselli, Giguère and Petit (2007)). These covert patterns are not altered by the dichotomization procedure, and further motivate the development of improved methods for principled uncertainty quantification and structured borrowing of information among affiliates via the inclusion of available knowledge on the criminal organization and on suspects' external attributes. For example, current forensic theories (e.g., Catino (2014), Paoli (2007)) and initial quantitative analyses (e.g., Calderoni, Brunetto and Piccardi (2017), Calderoni and Piccardi (2014)) suggest that the internal organization of 'Ndrangheta revolves around specific blood family relations, which may be further aggregated at the territorial level in structural coordinated units, named *locali*. Each *locale* controls a specific territory, and has a further layer of regulated hierarchy defined by a group of affiliates, and comparatively fewer bosses that are in charge of leading the *locale*, managing the funds, overseeing violent actions and guaranteeing the communication flows. Information on presumed *locale* membership and role can be retrieved, for each suspect of interest, from the judicial acts¹ of *Operazione Infinito* and, as shown in the graphical representation of the *Infinito network* in Figure 1, could help in assisting inference on the hidden underlying block structure, thus reducing the impact of covering strategies.

The inclusion of the aforementioned node attributes motivates careful and principled probabilistic representations accounting for the fact that these sources of external information are produced by an investigation process and, therefore, may be prone to measurement errors. Despite its relevance and potential benefits, this endeavor has been largely neglected in the

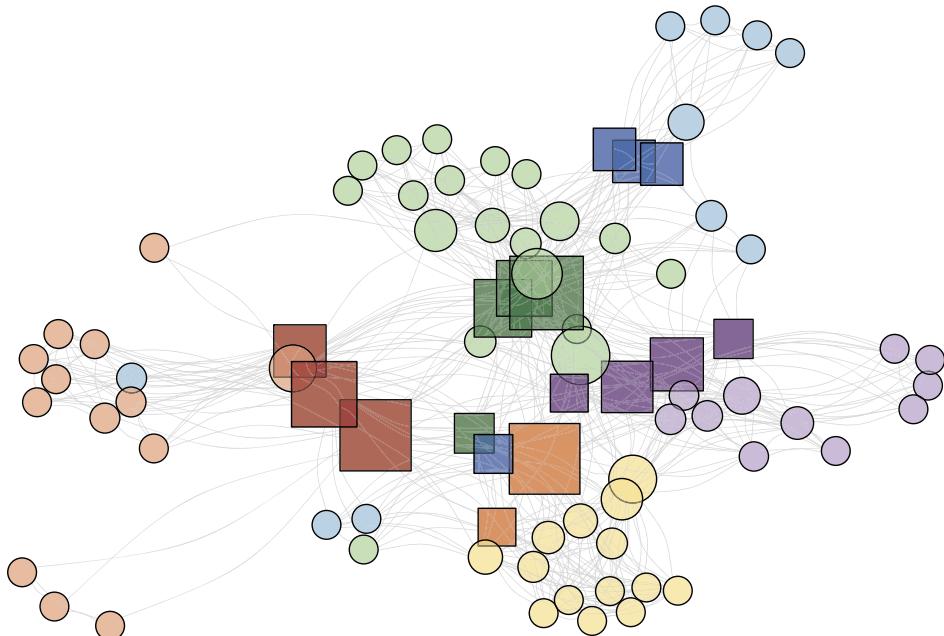


FIG. 1. Graphical representation of the *Infinito* network. Node positions are obtained via force-directed placement (Fruchterman and Reingold (1991)). Node size is proportional to the corresponding betweenness, whereas colors indicate the presumed *locale* membership. Darker square nodes represent the bosses of each *locale*, while lighter circles indicate the affiliates. See the online article for the color version of this figure.

analysis of criminal networks, and, as discussed in Section 1.2, state-of-the-art methods for block modeling lack a general solution to include error-prone node attribute effects in the partition process and quantify the magnitude of the improvements relative to no supervision. To cover this gap and flexibly learn the complex variety of block structures in noisy criminal networks, we develop a novel class of extended stochastic block models (ESBM) that formally quantify uncertainty in the suspects' grouping structure—including in the number, size and composition of the blocks—via Gibbs-type priors (Gnedin and Pitman (2005), Lijoi, Mena and Prünster (2007a, 2007b), Lijoi, Prünster and Walker (2008)) for the underlying partition; see also De Blasi et al. (2015) for a recent review.

As clarified in Section 2, although the block-modeling literature has focused on a much less extensive set of priors, the general Gibbs-type class is well motivated since it provides broad, interpretable and realistic probabilistic generative mechanisms for the formation of suspects' groups. This allows careful incorporation of probabilistic structure within a single modeling framework, which is amenable to novel extensions for the inclusion of probabilistic homophily with respect to error-prone external attributes, and for careful model-based inference on the partition structure via refined methods for estimation, uncertainty quantification, model selection and prediction. To assess out-of-sample predictive performance, we perform inference on the $V = 84$ suspects affiliated to the 5 most populated *locali*, and hold out as a test set the 34 members of those smaller-sized *locali* with ≤ 6 monitored affiliates. As discussed in Calderoni and Piccardi (2014), such a choice is also beneficial in reducing potential issues arising from the incomplete identification of low-sized *locali* during investigations, and, due to the modular organization of 'Ndrangheta (e.g., Catino (2014), Paoli (2007)), it arguably leads to a more accurate learning of its core recurring hierarchies.

1.2. Relevant literature. The relevance of learning block structures within networks has motivated a collective effort by various disciplines toward the development of methodologies for detecting node groups, ranging from algorithmic strategies (Blondel et al. (2008), Girvan and Newman (2002), Newman and Girvan (2004), Newman (2006), von Luxburg (2007)) to model-based solutions (Airoldi et al. (2008), Athreya et al. (2017), Geng, Bhattacharya and Pati (2019), Holland, Laskey and Leinhardt (1983), Karrer and Newman (2011), Kemp et al. (2006), Nowicki and Snijders (2001)); see Fortunato and Hric (2016), Abbe (2017) and Lee and Wilkinson (2019) for an overview.

Despite being routinely implemented in criminal network analyses, most algorithmic approaches focus on detecting communities characterized by a dense connectivity within each block and sparser connections between different blocks (Blondel et al. (2008), Girvan and Newman (2002), Newman and Girvan (2004), Newman (2006)). This constrained search is expected to provide a limited and possibly biased view of the key modules that are hidden in criminal networks. For instance, Figure 1 clearly highlights a core-periphery structure underlying the *Infinito network*, with communities of affiliates in peripheral positions and groups of bosses at the core. According to panel (a) in Figure 2, state-of-the-art algorithms for community detection (Blondel et al. (2008)) applied to the *Infinito network* obscure such patterns by overcollapsing some *locali*, while failing to separate affiliates from bosses.

These issues motivate a focus on alternative solutions aimed at grouping nodes which are characterized by common connectivity patterns within the network, rather than just exhibiting community structures. One possibility to address this goal from an algorithmic perspective is to rely on spectral clustering (von Luxburg (2007)). This strategy accounts for general block structures and possesses desirable properties, including consistency in estimation of the partition structure underlying various model-based representations (Athreya et al. (2017), Lei and Rinaldo (2015), Rohe, Chatterjee and Yu (2011), Sarkar and Bickel (2015), Sussman et al. (2012), Zhou and Amini (2019)). As shown in panel (b) of Figure 2, this yields improvements

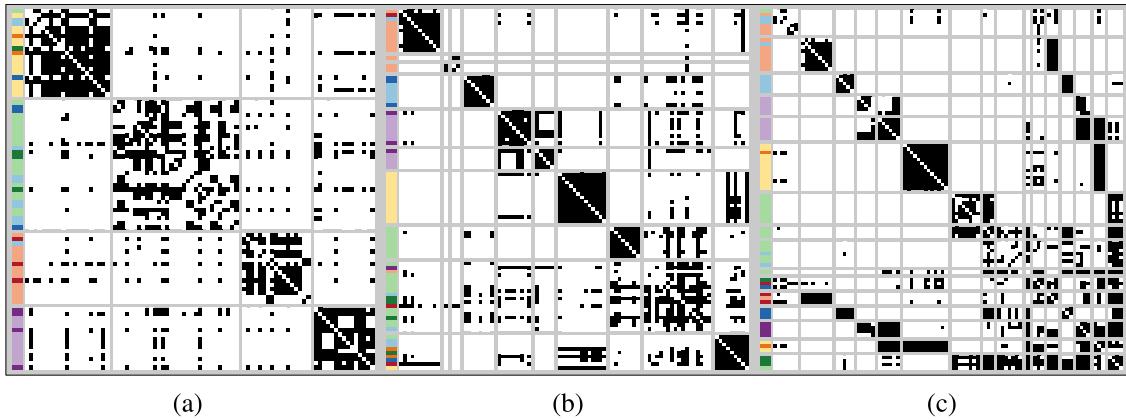


FIG. 2. *Adjacency matrix of the Infinito network with nodes reordered and partitioned in blocks according to the clustering structure estimated under three different methods: (a) community detection via the Louvain algorithm (Blondel et al. (2008)); (b) spectral clustering (von Luxburg (2007)) with the number of groups obtained via a combination of the model selection procedures in the R package `randsn`; (c) ESBM with supervised Gnedin process prior. Black and white cells represent edges and nonedges, respectively. Side colors correspond to the different locali, with darker and lighter shades denoting bosses and affiliates, respectively. See the online article for the color version of this figure.*

in learning complex block structures within the *Infinito network* relative to classical community detection algorithms. Nonetheless, spectral clustering lacks extensive methods for inference beyond point estimation, requires prespecification or heuristic algorithms to choose the unknown number of groups, and faces practical instabilities. As a result, this strategy is sub-optimal relative to carefully chosen model-based approaches; see panel (c) in Figure 2 for an example of the gains that can be obtained over spectral clustering by the methods developed in Sections 2–3.

Among the generative models for learning groups of nodes in network data, the stochastic block model (SBM) (Holland, Laskey and Leinhardt (1983), Nowicki and Snijders (2001)) is arguably the most widely implemented and well-established formulation, owing also to its careful balance between simplicity and flexibility (Abbe (2017), Lee and Wilkinson (2019)). In SBMs, the probability of each edge only depends on the cluster memberships of the two involved nodes, thereby allowing efficient inference on node groups and on block probabilities, which can also characterize disassortative, core-periphery or weak community patterns, and combinations of such structures (Fortunato and Hric (2016)). These desirable properties have motivated extensive theory (Bickel et al. (2013), Noroozi and Pensky (2020), Olhede and Wolfe (2014), Zhao, Levina and Zhu (2012)) and various generalizations of the SBM (Airoldi et al. (2008), Fosdick et al. (2019), Geng, Bhattacharya and Pati (2019), Handcock, Raftery and Tantrum (2007), Karrer and Newman (2011), Kemp et al. (2006), Newman and Clauset (2016), Rastelli, Latouche and Friel (2018), Schmidt and Morup (2013), Sengupta and Chen (2018), Stanley et al. (2019), Tallberg (2004), White and Murphy (2016)).

Part of the above extensions aim at addressing two fundamental open problems with classical SBMs, that also arise in criminal networks. First, in real-world applications the number of underlying groups is typically not known and has to be inferred from the data. Therefore, classical SBM formulations based on a fixed and prespecified number of groups (Holland, Laskey and Leinhardt (1983), Nowicki and Snijders (2001)) are conceptually unappealing in precluding uncertainty quantification on the unknown number of nonempty clusters, while state-of-the-art model selection procedures for choosing this quantity (Chen and Lei (2018), Le and Levina (2015), Li, Levina and Zhu (2020), Saldaña, Yu and Feng (2017), Wang and Bickel (2017)) led to mixed and biased results when applied to realistic criminal networks

in the simulation studies reported in Section 4. The second important problem is that, as discussed in Section 1, it is common to observe external and possibly error-prone node attributes that may effectively inform the grouping mechanism. Hence, SBMs require extensions to include such information in the partitioning process.

A successful answer to the first open issue has been provided by Bayesian nonparametric solutions replacing the original Dirichlet-multinomial process for node partitioning (Nowicki and Snijders (2001)) with alternative priors that allow the number of groups to either grow adaptively with the network size via the Chinese restaurant process (CRP) (Kemp et al. (2006), Schmidt and Morup (2013)) or to be finite and random under a mixture-of-finite-mixtures representation (Geng, Bhattacharya and Pati (2019)). Unfortunately, all these extensions have been developed separately and SBMs still lack a unifying framework, which would be conceptually and practically useful to clarify common properties, develop broad computational and inferential strategies, and identify novel solutions that may effectively address the problems arising from the criminal network discussed in Section 1.1. To address this gap, we unify in Section 2 most of the aforementioned formulations within an extended stochastic block model (ESBM) framework based on Gibbs-type priors.

As clarified in Section 2.2.2, this broad family of prior distributions also allows the natural inclusion of error-prone node attributes in a principled manner via product partition models (PPMs) (Hartigan (1990), Lijoi, Mena and Prünster (2007a), Müller, Quintana and Rosner (2011), Quintana and Iglesias (2003)), which favor the formation of groups that are homogeneous with respect to attributes, thereby incorporating probabilistic homophily. This property is known to play an important role in the formation of blocks within networks, and has inspired modifications of algorithmic strategies to learn group patterns coherent not only with network structure but also with pairwise similarities among node attributes (e.g., Binkiewicz, Vogelstein and Rohe (2017), Zhang, Levina and Zhu (2016)). These solutions often yield to practical gains, but inherit the inferential limitations of the unsupervised counterparts. Available model-based strategies (e.g., Gormley and Murphy (2010), Kim, Hughes and Sudderth (2012), Newman and Clauset (2016), Tallberg (2004), White and Murphy (2016), Zhao, Du and Buntine (2017)) commonly treat node membership variables as categorical responses whose formation depends on attributes via a higher-level regression model which, however, does not explicitly incorporate measurement errors in the attributes. Closer to the methods developed in Section 2.2.2 are mixture representations defining a joint model for the network and the node attributes, under the assumption of a shared underlying partition which influences the formation of both data structures (e.g., Stanley et al. (2019), Xu et al. (2012), Yang, McAuley and Leskovec (2013)). These models arise as attribute-assisted versions of the original SBM by Nowicki and Snijders (2001), but lack a broader modeling, inferential and computational framework to incorporate and compare more general priors on the random partition, beyond the Dirichlet-multinomial. Section 2.2.2 covers this gap by leveraging the connection between Gibbs-type priors and PPMs, which further provides a direct and principled characterization of homophily.

Within the Gibbs-type class, we will mainly focus on the Gnedenko process (De Blasi, Lijoi and Prünster (2013), Gnedenko (2010)) as an example of prior which has not yet been employed in SBMs, but exhibits analytical tractability, desirable properties, theoretical guarantees and promising empirical performance in applications; see panel (c) in Figure 2. As clarified in Section 3, our framework allows posterior computation via an easy-to-implement collapsed Gibbs sampler, and motivates general strategies for uncertainty quantification, prediction and model assessment, thus fully exploiting the advantages of a model-based approach over algorithmic strategies. The performance of key priors within the ESBM class and the magnitude of the improvements relative to state-of-the-art competitors are illustrated in Section 4 with extensive simulations focusing on realistic criminal network structures. In light of these results,

we opt for a supervised Gneden process to analyze the *Infinito network* in Section 5, obtaining a novel in-depth view of the modular organization of 'Ndrangheta that was hidden to previous quantitative studies. Concluding remarks are provided in Section 6, where we also mention possible extensions to degree-corrected stochastic block models (Karrer and Newman (2011)) and mixed membership stochastic block models (Airoldi et al. (2008)). Codes and data to reproduce all our results are available at <https://github.com/danieledurante/ESBM>; see also the Supplementary Material (Legramanti et al. (2022)).

2. Extended stochastic block models. Consider a binary undirected network having V nodes, and let \mathbf{Y} denote its $V \times V$ symmetric adjacency matrix, with elements $y_{vu} = y_{uv} = 1$ if nodes v and u are connected, and $y_{vu} = y_{uv} = 0$ otherwise. In our criminal network application self-loops are not allowed, and hence, are not included in the generative model. In Section 2.1, we first present the statistical model relying on classical SBM representations, and then characterize, in Section 2.2.1, the prior on the node partition via Gibbs-type processes leading to our general ESBM class. Such a unified representation is further extended in Section 2.2.2 to include information from error-prone node attributes. Consistent with our motivating application, we focus on binary undirected edges and categorical attributes, but our approach can be naturally extended to other types of networks and covariates, as highlighted in the final discussion.

2.1. Model formulation. SBMs (Holland, Laskey and Leinhardt (1983), Nowicki and Snijders (2001)) partition the nodes into H mutually exclusive and exhaustive groups, with nodes in the same cluster sharing common connectivity patterns. More specifically, SBMs assume that the sub-diagonal entries y_{vu} , for $v = 2, \dots, V$, $u = 1, \dots, v - 1$, of the symmetric adjacency matrix \mathbf{Y} are conditionally independent Bernoulli random variables with associated probabilities $\theta_{z_v, z_u} \in (0, 1)$ depending only on the group memberships z_v and z_u of the two involved nodes v and u . Let $\mathbf{z} = (z_1, \dots, z_V)^\top \in \{1, \dots, H\}^V$ be the node membership vector associated to the generic node partition $\{Z_1, \dots, Z_H\}$, so that $z_v = h$ if and only if $v \in Z_h$, and denote with Θ the $H \times H$ symmetric matrix whose generic element $\theta_{hk} \in (0, 1)$ measures the probability of an edge between a node in group h and a node in group k . Then, the likelihood for \mathbf{Y} is $p(\mathbf{Y}|\mathbf{z}, \Theta) = \prod_{h=1}^H \prod_{k=1}^h \theta_{hk}^{m_{hk}} (1 - \theta_{hk})^{\bar{m}_{hk}}$, where m_{hk} and \bar{m}_{hk} denote the number of edges and nonedges between nodes in groups h and k , respectively.

Classical SBMs (Holland, Laskey and Leinhardt (1983), Nowicki and Snijders (2001)) assume independent Beta(a, b) priors for the block probabilities θ_{hk} . Therefore, the joint density for the diagonal and the subdiagonal elements of Θ is $p(\Theta) = \prod_{h=1}^H \prod_{k=1}^h [\theta_{hk}^{a-1} (1 - \theta_{hk})^{b-1}] B(a, b)^{-1}$, where $B(\cdot, \cdot)$ is the Beta function. Although quantifying uncertainty in the block probabilities is important, the overarching goal in SBMs is to infer the node partition. Consistent with this focus, Θ is commonly treated as a nuisance parameter, which is marginalized out in $p(\mathbf{Y}|\mathbf{z}, \Theta)$ via beta-binomial conjugacy, obtaining

$$(1) \quad p(\mathbf{Y}|\mathbf{z}) = \prod_{h=1}^H \prod_{k=1}^h \frac{B(a + m_{hk}, b + \bar{m}_{hk})}{B(a, b)}.$$

As we will clarify in Section 3, such a marginalization is also useful for computation and inference. The likelihood in (1) is common to several SBM extensions, which then differ in the choice of the probabilistic mechanism underlying \mathbf{z} . Let $\bar{H} \geq H$ be the total number of possible groups in the whole population of nodes, and denote with $\bar{\mathbf{z}} = (\bar{z}_1, \dots, \bar{z}_V)^\top \in \{1, \dots, \bar{H}\}^V$ the indicators which define the population clusters for the V observed nodes. A natural option to characterize the generative process for the partition is to consider a Dirichlet-multinomial prior distribution for $\bar{\mathbf{z}}$, obtained by marginalizing the vector of group probabilities $\boldsymbol{\pi} = (\pi_1, \dots, \pi_{\bar{H}}) \sim \text{Dirichlet}(\boldsymbol{\beta})$ out of a multinomial likelihood for $\bar{\mathbf{z}}$, in which

$\text{pr}(\bar{z}_v = h | \pi) = \pi_h$, for $v = 1, \dots, V$. If \bar{H} is fixed and finite, this leads to the original Bayesian SBM (Nowicki and Snijders (2001)). However, as already discussed, the number of groups in criminal networks is typically unknown and has to be inferred from the data. A possible solution consists in placing a prior on \bar{H} , which leads to the mixture-of-finite-mixtures (MFM) version of the SBM in Geng, Bhattacharya and Pati (2019). Another option is a Dirichlet process partition mechanism, corresponding to the infinite relational model (Kemp et al. (2006)). Such an infinite mixture model differs from MFM in that $\bar{H} = \infty$, meaning that infinitely many nodes would give rise to infinitely many groups. Note that the total number of possible clusters \bar{H} should not be confused with the number of occupied clusters H . The latter is defined as the number of distinct labels in $\bar{\mathbf{z}}$, and is upper bounded by $\min\{V, \bar{H}\}$.

Notably, all the above solutions are specific examples of Gibbs-type priors (e.g., De Blasi et al. (2015)), thus motivating our unified ESBM class presented in Section 2.2. Before introducing this extension it is worth noticing that $\bar{\mathbf{z}}$ identifies labeled clusters. Hence, a vector $\bar{\mathbf{z}}$ and its relabelings are regarded as distinct objects, even though they identify the same partition. Throughout the rest of the paper, we will rely on the previously-defined vector \mathbf{z} , which denotes all relabelings of $\bar{\mathbf{z}}$ that lead to the same partition. For simplicity, we assume that $z_v \in \{1, \dots, H\}$, which corresponds to avoiding empty groups. This does not modify likelihood (1), which is invariant under relabeling, that is, $p(\mathbf{Y}|\mathbf{z}) = p(\mathbf{Y}|\bar{\mathbf{z}})$.

2.2. Prior specification. As illustrated in Section 2.1, several priors for \mathbf{z} have been considered in the context of SBMs, including the Dirichlet-multinomial (Nowicki and Snijders (2001)), the Dirichlet process (Kemp et al. (2006)), and mixtures of finite Dirichlet mixtures (Geng, Bhattacharya and Pati (2019)). Interestingly, these are all examples of Gibbs-type priors, which stand out for their analytical and computational tractability; see De Blasi et al. (2015) for a comprehensive overview. In Section 2.2.1, we propose the ESBM as a unifying framework characterized by the choice of a Gibbs-type prior for \mathbf{z} . This formulation includes the previously-mentioned SBMs as special cases and offers new alternatives by exploring the whole Gibbs-type class and its direct relation with PPMs (Hartigan (1990), Lijoi, Mena and Prünster (2007a), Quintana and Iglesias (2003)). This connection with PPMs is exploited in Section 2.2.2 to supervise the prior via possibly error-prone node attributes.

2.2.1. Unsupervised Gibbs-type priors. Gibbs-type prior distributions are defined on the space of unlabeled group indicators \mathbf{z} . For $a > 0$, denote the ascending factorial with $(a)_n = a(a+1)\cdots(a+n-1)$ for every $n \geq 1$, and set $(a)_0 = 1$. A probability distribution function $p(\mathbf{z})$ is of Gibbs-type if and only if

$$(2) \quad p(\mathbf{z}) = \mathcal{W}_{V,H} \prod_{h=1}^H (1 - \sigma)_{n_h-1},$$

where n_h is the number of nodes in cluster h , $\sigma < 1$ denotes the so-called *discount parameter* and $\{\mathcal{W}_{V,H} : 1 \leq H \leq V\}$ is a collection of nonnegative weights that satisfy the recursion $\mathcal{W}_{V,H} = (V - H\sigma)\mathcal{W}_{V+1,H} + \mathcal{W}_{V+1,H+1}$, with $\mathcal{W}_{1,1} = 1$. As shown in Lijoi, Mena and Prünster (2007a), the class of random partitions induced by Gibbs-type priors coincides with exchangeable PPMs, which are probability models for random partitions \mathbf{z} of the form $p(\mathbf{z}) \propto c(Z_1) \cdots c(Z_H)$, where $\{Z_1, \dots, Z_H\}$ is the partition associated to \mathbf{z} , whereas $c(\cdot)$ is a nonnegative *cohesion function* measuring the homogeneity within each cluster. Such a connection will be useful to incorporate node-specific attributes in ESBMs. Interestingly, Gibbs-type priors represent a broad, yet tractable, class whose predictive distribution (Lijoi, Mena and Prünster (2007b)) implies that membership indicators \mathbf{z} can be obtained in a sequential and interpretable manner according to

$$(3) \quad \text{pr}(z_{V+1} = h | \mathbf{z}) \propto \begin{cases} \mathcal{W}_{V+1,H}(n_h - \sigma) & \text{for } h = 1, \dots, H, \\ \mathcal{W}_{V+1,H+1} & \text{for } h = H + 1. \end{cases}$$

TABLE 1
A classification of Gibbs-type priors

\bar{H}	σ	H (growth)	Example
I	Fixed	$\sigma < 0$	–
II	Random	$\sigma < 0$	–
III.a	Infinite	$\sigma = 0$	$\mathcal{O}(\log V)$
III.b	Infinite	$\sigma \in (0, 1)$	$\mathcal{O}(V^\sigma)$
			Pitman–Yor process (PY)

Hence, the group assignment process can be interpreted as a simple seating mechanism in which a new node is assigned to an existing cluster h with probability proportional to the current size n_h of that cluster, discounted by a global factor σ and further rescaled by a weight $\mathcal{W}_{V+1, H}$, which may depend both on the size V of the network and on the current number H of nonempty groups. Alternatively, the incoming node is assigned to a new cluster with probability proportional to $\mathcal{W}_{V+1, H+1}$. Such a general mechanism is conceptually appealing in our application to criminal networks since it realistically accounts for group sizes n_h , network size V and complexity H in the formation process of the modular structure underlying the criminal organization, while providing a variety of possible generative mechanisms under a single modeling framework. In the examples below, we show how commonly used priors in SBMs and unexplored alternatives of interest in criminal network studies can be obtained as special cases of (3).

EXAMPLE 1 (DM – Dirichlet-multinomial). Let $\sigma < 0$ and consider the collection of weights $\mathcal{W}_{V, H} = [\beta^{H-1}/(\beta\bar{H} + 1)_{V-1}] \prod_{h=1}^{H-1} (\bar{H} - h) \mathbb{1}(H \leq \bar{H})$ for some $\beta = -\sigma$ and $\bar{H} \in \{1, 2, \dots\}$. Then (3) coincides with the DM urn scheme: $\text{pr}(z_{V+1} = h | \mathbf{z}) \propto n_h + \beta$ for $h = 1, \dots, H$, and $\text{pr}(z_{V+1} = H + 1 | \mathbf{z}) \propto \beta(\bar{H} - H) \mathbb{1}(H \leq \bar{H})$.

EXAMPLE 2 (DP – Dirichlet process). Let $\sigma = 0$ and set $\mathcal{W}_{V, H} = \alpha^H/(\alpha)_V$ for some $\alpha > 0$. Then (3) leads to the CRP urn scheme: $\text{pr}(z_{V+1} = h | \mathbf{z}) \propto n_h$ for $h = 1, \dots, H$, and $\text{pr}(z_{V+1} = H + 1 | \mathbf{z}) \propto \alpha$. CRP can be obtained as a limiting DM with $\beta = \alpha/\bar{H}$, as $\bar{H} \rightarrow \infty$.

EXAMPLE 3 (PY – Pitman–Yor process). Let $\sigma \in [0, 1)$ and set $\mathcal{W}_{V, H} = [\prod_{h=1}^{H-1} (\alpha + h\sigma)]/(\alpha + 1)_{V-1}$ for some $\alpha > -\sigma$. Then (3) characterizes the PY process: $\text{pr}(z_{V+1} = h | \mathbf{z}) \propto n_h - \sigma$ for $h = 1, \dots, H$, and $\text{pr}(z_{V+1} = H + 1 | \mathbf{z}) \propto \alpha + H\sigma$. PY reduces to a DP when $\sigma = 0$.

EXAMPLE 4 (GN – Gneden process). Let $\sigma = -1$ and set $\mathcal{W}_{V, H} = [(\gamma)_{V-H} \prod_{h=1}^{H-1} (h^2 - \gamma h)]/\prod_{v=1}^{V-1} (v^2 + \gamma v)$ for some $\gamma \in (0, 1)$. Then (3) identifies the GN process: $\text{pr}(z_{V+1} = h | \mathbf{z}) \propto (n_h + 1)(V - H + \gamma)$ for $h = 1, \dots, H$, and $\text{pr}(z_{V+1} = H + 1 | \mathbf{z}) \propto H^2 - H\gamma$.

Other popular examples of tractable Gibbs-type priors can be found in De Blasi, Lijoi and Prünster (2013), De Blasi et al. (2015), Lijoi, Mena and Prünster (2007a, 2007b) and Miller and Harrison (2018).

Priors DM, DP, PY and GN provide various realistic generative mechanisms for the grouping structure in criminal networks, thus allowing analysts to choose the most suitable one for a given study, or possibly test different specifications under a single modeling framework. For example, DP and PY (Kemp et al. (2006)) may provide useful constructions in the analysis of relatively unstable and fragmented criminal organizations, such as terrorist networks, which are characterized by multiple small cells and even *lone wolves*. As shown in Table 1, when the growth is expected to be rapid, that is, $\mathcal{O}(V^\sigma)$, and possibly favoring the formation of

low-sized groups, PY may be a more sensible choice relative to DP, which in turn would be recommended in regimes with slower increments, that is, $\mathcal{O}(\log V)$. Organized crime, such as 'Ndrangheta, is instead characterized by a more stable and highly regulated modular architecture, which might support the use of priors with a finite number \bar{H} of population clusters, such as DM (Nowicki and Snijders (2001)) and GN. Clearly, in most forensic studies, \bar{H} is unknown, and hence, quantifying uncertainty in \bar{H} under GN provides a more realistic choice than fixing \bar{H} as in DM. In fact, the GN process can be derived from DM by placing a prior on \bar{H} , thus making it random. Specifically, the distribution $p_{\text{GN}}(\mathbf{z})$ of \mathbf{z} under the GN process can be expressed as

$$p_{\text{GN}}(\mathbf{z}) = \sum_{h=1}^{\infty} \text{pr}_{\text{GN}}(\bar{H} = h) p_{\text{DM}}(\mathbf{z}; 1, h),$$

where $p_{\text{DM}}(\mathbf{z}; 1, h)$ is the Dirichlet-multinomial distribution in the first Example, with $\beta = 1$ and $\bar{H} = h$, whereas $\text{pr}_{\text{GN}}(\bar{H} = h) = \gamma(1 - \gamma)^{h-1}/h!$ can be interpreted as the prior on \bar{H} under GN. Although different prior choices for \bar{H} might be considered (De Blasi et al. (2015), Geng, Bhattacharya and Pati (2019), Miller and Harrison (2018)), the GN process has conceptual and practical advantages in applications to criminal networks. First, the sequential mechanism described in the fourth Example has a simple analytical expression that facilitates posterior inference and prediction. Moreover, the distribution $\text{pr}_{\text{GN}}(\bar{H} = h) = \gamma(1 - \gamma)^{h-1}/h!$ has the mode at 1, heavy tail and infinite expectation (Gnedin (2010)). Hence, the associated MFM favors parsimonious representations of the block structures in criminal organizations which facilitate repressive operations, but preserves robustness to \bar{H} due to heavy tails.

The prior on \bar{H} quantifies the uncertainty in the total number of clusters that one would expect if $V \rightarrow \infty$. However, in applications, the number of nonempty groups H occupied by the observed V nodes is of more direct interest and can also guide the choice of the prior hyperparameters. Under Gibbs-type priors, this quantity has a closed-form probability distribution function, which coincides with $\text{pr}(H = h) = \mathcal{W}_{V,h}\mathcal{C}(V, h; \sigma)\sigma^{-h}$ for every $h = 1, \dots, V$, where $\mathcal{C}(V, h; \sigma)$ denotes the so-called generalized factorial coefficient (Gnedin and Pitman (2005)). The DP case is recovered when $\sigma \rightarrow 0$. In <https://github.com/danieledurante/ESBM>, we provide codes to evaluate such quantities under the Gibbs-type priors in Table 1, and then exploit these values for guiding the choice of the hyperparameters, a strategy first proposed in Lijoi, Mena and Prünster (2007a, 2007b). In Sections 4–5, this is accomplished by combining a visual inspection of the prior induced on H with the analysis of its moments. This strategy provides a practically effective solution in a broad set of applications where expert knowledge can be directly quantified through prior information on H , which naturally translates into specific hyperparameters for the four examples of Gibbs-type priors in Table 1.

In addition to its practical relevance, the above result clarifies also the asymptotic behavior of H . Indeed, the prior for H converges to a point mass in scenario I, to a proper distribution in scenario II and to a point mass at infinity in scenario III. For instance, under GN we have

$$\text{pr}_{\text{GN}}(H = h) = \binom{V}{h} \frac{(1 - \gamma)^{h-1}(\gamma)^{V-h}}{(1 + \gamma)^{V-1}}, \quad \text{for } h = 1, \dots, V,$$

and hence the expectation can be easily computed via $\mathbb{E}_{\text{GN}}(H) = \sum_{h=1}^V h \cdot \text{pr}_{\text{GN}}(H = h)$. Note that $\lim_{V \rightarrow \infty} \text{pr}_{\text{GN}}(H = h) = \text{pr}_{\text{GN}}(\bar{H} = h) = \gamma(1 - \gamma)^{h-1}/h!$.

The prior on \bar{H} induced by GN also ensures posterior consistency for the estimated grouping structure. This follows from the theory for MFM in Geng, Bhattacharya and Pati (2019), that actually applies to any DM with prior on \bar{H} supported on all positive integers. In particular, this holds for GN, thus giving further support for the use of such a prior in the motivating criminal network application. Instead, DP and PY unsurprisingly lead to inconsistent estimates for \bar{H} if the data are generated from a model with $\bar{H}_0 < \infty$ (Miller and Harrison (2014)). Intuitively, this happens because DP and PY assume $\bar{H} = \infty$. Hence, we suggest Gibbs-type

priors with $\sigma \geq 0$ only if the analyst believes that $\overline{H}_0 = \infty$, that is, when the true number of groups grows without bound with the number of nodes; see also Sections 3.2, 4 and 5 for data-driven strategies to select among the different priors via WAIC (Watanabe (2010, 2013)).

2.2.2. Supervised Gibbs-type priors. If node attributes $\mathbf{x}_v = (x_{v1}, \dots, x_{vd})^\top$ are available for each $v = 1, \dots, V$, this external information may support inference on block structures, both in terms of point estimation and in reduction of posterior uncertainty. As mentioned in Section 1, this is particularly relevant in applications to criminal networks where specific block structures could be purposely blurred by covering strategies and, therefore, inclusion of informative attributes might help in revealing obscured modules. This solution should also account for the fact that node attributes collected in investigations may be error prone.

An option to address such goals in a principled manner within ESBMs is to rely on the PPM structure of Gibbs-type priors. Adapting Park and Dunson (2010) and Müller, Quintana and Rosner (2011) to our network setting, this solution is based on the idea of replacing (2) with

$$(4) \quad p(\mathbf{z}|\mathbf{X}) \propto \mathcal{W}_{V,H} \prod_{h=1}^H p(\mathbf{X}_h)(1-\sigma)_{n_h-1},$$

where $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_V)^\top$, whereas $\mathbf{X}_h = \{\mathbf{x}_v : z_v = h\}$ are the attributes for the nodes in cluster h . In (4), $p(\mathbf{X}_h)$ controls the contribution of \mathbf{X} to the cluster cohesion by favoring groups that are homogeneous with respect to attribute values, while including uncertainty in \mathbf{X} . Motivated by the *Infinito network* application, we consider the case in which each node attribute $\mathbf{x}_v = x_v \in \{1, \dots, C\}$ is a single categorical variable denoting a suitable combination between *locale* affiliation and role in the criminal organization. This is a common setting in criminology, where node attributes often come in the form of exogenous partitions defined by the forensic agencies as a result of the investigation process. In these categorical settings, the recommended practice within the PPM framework (Müller, Quintana and Rosner (2011)) is to rely on the Dirichlet-multinomial (without multinomial coefficient) cohesion function

$$(5) \quad p(\mathbf{X}_h) \propto \frac{1}{\Gamma(n_h + \alpha_0)} \prod_{c=1}^C \Gamma(n_{hc} + \alpha_c),$$

where n_{hc} is the number of nodes in cluster h with attribute value c , while $\alpha_0 = \sum_{c=1}^C \alpha_c$, with $\alpha_c > 0$ for $c = 1, \dots, C$. Including this cohesion in equation (4) leads to the urn scheme:

$$(6) \quad \text{pr}(z_{V+1} = h | \mathbf{X}, x_{V+1}, \mathbf{z}) \propto \begin{cases} \frac{n_{hx_{V+1}} + \alpha_{x_{V+1}}}{n_h + \alpha_0} \mathcal{W}_{V+1,H}(n_h - \sigma) & \text{for } h = 1, \dots, H, \\ \frac{\alpha_{x_{V+1}}}{\alpha_0} \mathcal{W}_{V+1,H+1} & \text{for } h = H + 1, \end{cases}$$

where $n_{hx_{V+1}}$ is the number of nodes in cluster h having the same covariate value $c = x_{V+1}$ as node $V + 1$, n_h is the total number of nodes in cluster h , whereas $\alpha_{x_{V+1}}$ is the parameter associated with the category $c = x_{V+1}$ of node $V + 1$. As clarified in (6), the introduction of a $p(\mathbf{X}_h)$, defined as in (5), induces probabilistic homophily favoring attribution of a new node to those groups that have a higher fraction of existing nodes with its same attribute value.

Besides including realistic homophily structures, the above representation effectively accounts for possible noise in the attributes. Indeed, the expression for $p(\mathbf{X}_h)$ in (5) coincides with the marginal likelihood for the attributes of the nodes in group h under the assumption that the model underlying all these quantities is defined by a multinomial with group-specific class probabilities $\boldsymbol{\nu}_h = (\nu_{1h}, \dots, \nu_{Ch})^\top$, which are assigned a Dirichlet prior with parameters $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_C)^\top$. Under this interpretation, the supervised Gibbs-type prior in equation (4) can be reexpressed as $p(\mathbf{z}|\mathbf{X}) \propto [\mathcal{W}_{V,H} \prod_{h=1}^H (1-\sigma)_{n_h-1}] \prod_{h=1}^H p(\mathbf{X}_h) \propto p(\mathbf{z}) p(\mathbf{X}|\mathbf{z})$, where $p(\mathbf{z})$ is the unsupervised Gibbs-type prior in Section 2.2.1, whereas $p(\mathbf{X}|\mathbf{z})$ is the likelihood induced by the Dirichlet-multinomial model for the observed node attributes. Hence, learning

block structures in \mathbf{Y} under the supervised Gibbs-type prior can be interpreted as a two-step Bayesian procedure in which the unsupervised prior on \mathbf{z} is first updated with the likelihood for the attributes in \mathbf{X} , and then such a first-step posterior enters as a new prior in the second step to be updated with the information from the observed network \mathbf{Y} . Under the assumption of conditional independence between \mathbf{Y} and \mathbf{X} given \mathbf{z} , such a two-step process yields the actual posterior for \mathbf{z} , since $p(\mathbf{z}|\mathbf{Y}, \mathbf{X}) \propto [p(\mathbf{z})p(\mathbf{X}|\mathbf{z})]p(\mathbf{Y}|\mathbf{z}) \propto p(\mathbf{z}|\mathbf{X})p(\mathbf{Y}|\mathbf{z})$.

As mentioned in Section 1.2, while the induced joint model for \mathbf{Y} and \mathbf{X} is reminiscent of earlier constructions (Stanley et al. (2019), Xu et al. (2012), Yang, McAuley and Leskovec (2013)), our solution crucially extends these ideas to the whole ESBM class, well beyond the original Bayesian SBM by Nowicki and Snijders (2001).

3. Posterior computation and inference. In Section 3.1, we derive a collapsed Gibbs sampler which holds for the entire ESBM class presented in Section 2. Then, in Section 3.2 we provide extensive tools not only for point estimation of the group structure, but also for uncertainty quantification, model selection and prediction. Despite their relevance in routine studies including, for example, the *Infinito network* motivating application presented in Section 1.1, these aspects have been partially neglected in the SBM literature.

3.1. Collapsed Gibbs sampler. The availability of the urn schemes in (3) and (6) for the whole class of Gibbs-type priors allows the derivation of a general collapsed Gibbs sampler that holds for any ESBM; see Algorithm 1. At every iteration, this routine samples the group assignment of each node $v = 1, \dots, V$ from its full conditional distribution given the adjacency matrix \mathbf{Y} and the vector \mathbf{z}_{-v} of the cluster assignments of all the other nodes, excluding v . By direct application of the Bayes rule, these full conditional probabilities are

$$(7) \quad \text{pr}(z_v = h | \mathbf{Y}, \mathbf{X}, \mathbf{z}_{-v}) \propto \text{pr}(z_v = h | \mathbf{X}, \mathbf{z}_{-v}) \frac{p(\mathbf{Y}|z_v = h, \mathbf{z}_{-v})}{p(\mathbf{Y}_{-v}|\mathbf{z}_{-v})},$$

where \mathbf{Y}_{-v} is the $(V - 1) \times (V - 1)$ adjacency matrix without the row and column referring to node v . Recalling Schmidt and Morup (2013), the last term in (7) can be simplified as

$$(8) \quad \frac{p(\mathbf{Y}|z_v = h, \mathbf{z}_{-v})}{p(\mathbf{Y}_{-v}|\mathbf{z}_{-v})} = \prod_{k=1}^H \frac{\text{B}(a + m_{hk}^-, r_{vk}, b + \bar{m}_{hk}^-, \bar{r}_{vk})}{\text{B}(a + m_{hk}^-, b + \bar{m}_{hk}^-)},$$

where m_{hk}^- and \bar{m}_{hk}^- denote the number of edges and nonedges between clusters h and k , without counting node v , while r_{vk} and \bar{r}_{vk} define the number of edges and nonedges between node v and the nodes in cluster k . The prior term $\text{pr}(z_v = h | \mathbf{X}, \mathbf{z}_{-v})$ in (7) is directly available from either (3) or (6), depending on whether node attributes are excluded or included, respectively. In particular, the unsupervised Gibbs-type priors discussed in Section 2.2.1 yield

$$(9) \quad \text{pr}(z_v = h | \mathbf{X}, \mathbf{z}_{-v}) = \text{pr}(z_v = h | \mathbf{z}_{-v}) \propto \begin{cases} \mathcal{W}_{V, H^-}(n_h^- - \sigma) & \text{for } h \leq H^-, \\ \mathcal{W}_{V, H^-+1} & \text{for } h = H^- + 1, \end{cases}$$

where n_h^- and H^- are the cardinality of cluster h and the total number of occupied clusters, respectively, after removing node v . Whereas, the supervised extension in Section 2.2.2 yields

$$(10) \quad \text{pr}(z_v = h | \mathbf{X}, \mathbf{z}_{-v}) \propto \begin{cases} \frac{n_{hx_v}^- + \alpha_{x_v}}{n_h^- + \alpha_0} \mathcal{W}_{V, H^-}(n_h^- - \sigma) & \text{for } h \leq H^-, \\ \frac{\alpha_{x_v}}{\alpha_0} \mathcal{W}_{V, H^-+1} & \text{for } h = H^- + 1, \end{cases}$$

where $n_{hx_v}^-$ is the number of nodes in cluster h with covariate value $c = x_v$, without counting node v , whereas α_{x_v} coincides with the parameter for the category $c = x_v$ of node v . Under

Algorithm 1: Gibbs sampler for ESBM

At each iteration update the cluster assignments z_1, \dots, z_V as follows:

For $v = 1, \dots, V$ **do:**

1. Remove node v from the network;
2. If the cluster which contained node v becomes empty, discard it and relable the group indicators (so that clusters $1, \dots, H^-$ are nonempty);
3. Sample z_v from the categorical random variable with probabilities as in (7) for $h = 1, \dots, H^- + 1$, where $p(\mathbf{Y}|z_v = h, \mathbf{z}_{-v})/p(\mathbf{Y}_{-v}|\mathbf{z}_{-v})$ is defined in (8), while $\text{pr}(z_v = h|\mathbf{X}, \mathbf{z}_{-v})$ coincides with either (9) or (10) depending on whether node attributes are excluded or included, respectively.

the priors presented in Table 1, both (9) and (10) admit the simple expressions reported in the four Examples in Section 2.2.1.

Although Algorithm 1 leverages likelihood (1) with the block probabilities θ_{hk} integrated out, a plug-in estimate for each θ_{hk} can be easily obtained. In particular, since $(\theta_{hk}|\mathbf{Y}, \mathbf{z}) \sim \text{Beta}(a + m_{hk}, b + \bar{m}_{hk})$, a reasonable point estimate for θ_{hk} is

$$(11) \quad \hat{\theta}_{hk} = \mathbb{E}(\theta_{hk}|\mathbf{Y}, \mathbf{z} = \hat{\mathbf{z}}) = \frac{a + \hat{m}_{hk}}{a + \hat{m}_{hk} + b + \hat{\bar{m}}_{hk}},$$

for every $h = 1, \dots, \hat{H}$ and $k = 1, \dots, h$, where \hat{m}_{hk} and $\hat{\bar{m}}_{hk}$ denote the number of edges and nonedges between nodes in groups h and k , computed from the estimated $\hat{\mathbf{z}}$. In the next subsection, we describe improved methods for estimation of \mathbf{z} , uncertainty quantification in group detection, model selection and prediction.

3.2. Estimation, uncertainty quantification, model selection, prediction. While algorithmic methods return a single estimated partition, ESBM provides the whole posterior distribution over the space of node partitions. To fully exploit such a posterior and perform inference directly on the space of partitions, we adapt to the block modeling setting the decision-theoretic approach proposed by Wade and Ghahramani (2018). In this way, we summarize posterior distributions on partitions leveraging the *variation of information* (VI) metric (Meilă (2007)), that quantifies distances between two clusterings by comparing their individual and joint entropies, and ranges from 0 to $\log_2 V$. Intuitively, VI measures the amount of information in two clusterings relative to the information shared between them, thus providing a metric that decreases to 0 as the overlap between two partitions grows; see Wade and Ghahramani (2018) for a discussion of the key properties of VI. Under this framework, a formal Bayesian point estimate for \mathbf{z} is that partition with the lowest posterior averaged VI distance from the other clusterings, thus obtaining

$$(12) \quad \hat{\mathbf{z}} = \arg \min_{\mathbf{z}'} \mathbb{E}_{\mathbf{z}}[\text{VI}(\mathbf{z}, \mathbf{z}')|\mathbf{Y}],$$

where the expected value is taken with respect to the posterior distribution of \mathbf{z} . Due to the huge cardinality of the space of partitions, even for moderate V , the optimization in (12) is typically carried out through a greedy algorithm (Wade and Ghahramani (2018)), as in the R package `mcclust.ext`.

The VI distance also provides natural strategies to construct credible sets around point estimates. In particular, one can define a $1 - \alpha$ credible ball around $\hat{\mathbf{z}}$ by ordering the partitions according to their VI distance from $\hat{\mathbf{z}}$, and defining the ball as containing all the partitions having less than a threshold distance from $\hat{\mathbf{z}}$, with this threshold chosen to minimize the size of the ball while ensuring that it contains at least $1 - \alpha$ posterior probability. Summarizing

such a ball is nontrivial given the high-dimensional and discrete nature of the space of partitions. In practice, as illustrated in our studies, one can report the partition at the edge of the ball, which we call credible bound. This form of uncertainty quantification complements the commonly-studied *posterior similarity matrix* that measures, for each pair of nodes, the relative frequency of MCMC samples in which such nodes are assigned to the same group (e.g., Wade and Ghahramani (2018)). Relative to this quantity, the additional inference methods we propose are conceptually and practically more appealing as they allow estimation and uncertainty quantification directly on the space of partitions.

Another important inference task is the selection among several candidate models—which mainly arise in our context from the choice among different priors for \mathbf{z} in Section 2.2. One possibility to formally address this goal is through the Bayes factor (e.g., Kass and Raftery (1995)). However, this strategy requires calculation of the marginal likelihood $p(\mathbf{Y}|\mathcal{M}) = \sum_{\mathbf{z}} p(\mathbf{Y}|\mathbf{z})p(\mathbf{z}|\mathcal{M})$ for a generic model \mathcal{M} , which is not available analytically under the priors in Section 2.2. Although simple strategies, such as the harmonic mean estimate (Raftery et al. (2007)), can be employed to compute $p(\mathbf{Y}|\mathcal{M})$ in SBMs (e.g., Legramanti, Rigon and Durante (2020)), these solutions may face instabilities and slow convergence in general settings (e.g., Lenk (2009), Pajor (2017), Wang et al. (2018)). To overcome these shortcomings and provide a general-use model selection strategy, we opt for the WAIC information criterion (Gelman, Hwang and Vehtari (2014), Watanabe (2010, 2013)). Relative to other information criteria commonly employed also in the SBM framework and its extensions (e.g., Côme and Latouche (2015), Gormley and Murphy (2010), Lee and Wilkinson (2019), Rastelli, Latouche and Friel (2018), Saldaña, Yu and Feng (2017)), the WAIC yields practical and theoretical advantages (Gelman, Hwang and Vehtari (2014)), and has direct connections with Bayesian leave-one-out cross-validation (Watanabe (2010)), thus providing also a measure of edge predictive accuracy. In addition, calculation of the WAIC only requires posterior samples of the log-likelihoods for the edges, $\log p(y_{vu}|\mathbf{z}, \Theta) = y_{vu} \log \theta_{z_v, z_u} + (1 - y_{vu}) \log(1 - \theta_{z_v, z_u})$, $v = 2, \dots, V$, $u = 1, \dots, v - 1$. These quantities can be readily obtained by combining the posterior samples for \mathbf{z} from Algorithm 1, with those for the block probabilities in Θ , which can be easily simulated from the conjugate full conditional distributions $(\theta_{hk}|\mathbf{Y}, \mathbf{z}) \sim \text{Beta}(a + m_{hk}, b + \bar{m}_{hk})$ for $h = 1, \dots, H$ and $k = 1, \dots, h$ via a separate algorithm that can be run in parallel across blocks and samples; see Section 3.4 in Gelman, Hwang and Vehtari (2014) for details on the WAIC, and refer to the WAIC function in the R package LaplacesDemon for practical implementation. As a global measure of goodness-of-fit, we also study the misclassification error when predicting each y_{vu} with $\hat{\theta}_{\hat{z}_v \hat{z}_u}$ from (11).

Recalling the criminal network application presented in Section 1.1, predicting the group membership z_{V+1} for a newly observed suspect $V + 1$ is also of fundamental interest in these contexts. While common algorithmic strategies would require heuristic procedures, the urn scheme representation (3) of the Gibbs-type priors provides a natural construction to obtain formal estimates of group probabilities for incoming suspects, without conditioning on external attributes that are typically unavailable in early investigations of such new individuals. Combining equations (7)–(8) with the urn scheme in (3), a plug-in estimate for the predictive probabilities of the cluster allocations for node $V + 1$ is

$$(13) \quad \begin{aligned} & \text{pr}(z_{V+1} = h|\mathbf{Y}, \mathbf{y}_{V+1}, \hat{\mathbf{z}}) \\ & \propto \text{pr}(z_{V+1} = h|\hat{\mathbf{z}}) \prod_{k=1}^{\hat{H}} \frac{\text{B}(a + \hat{m}_{hk} + \hat{r}_{V+1,k}, b + \hat{\bar{m}}_{hk} + \hat{\bar{r}}_{V+1,k})}{\text{B}(a + \hat{m}_{hk}, b + \hat{\bar{m}}_{hk})}, \end{aligned}$$

for every $h = 1, \dots, \hat{H} + 1$, with $\text{pr}(z_{V+1} = h|\hat{\mathbf{z}})$ as in (3). In (13), $\mathbf{y}_{V+1} = (y_{V+1,1}, \dots, y_{V+1,V})^\top$ is the vector of newly observed edges between node $V + 1$ and those already in network \mathbf{Y} . The frequencies \hat{m}_{hk} and $\hat{\bar{m}}_{hk}$ denote instead the number of edges and nonedges

between the existing nodes in groups h and k computed from the estimated cluster assignments in $\hat{\mathbf{z}}$, whereas $\hat{r}_{V+1,k}$ and $\hat{r}_{V+1,k}$ define the number of edges and nonedges between the incoming node $V + 1$ and the existing nodes in cluster k , still evaluated at the estimated partition $\hat{\mathbf{z}}$. Note that, under the priors in Table 1, the quantity $\text{pr}(z_{V+1} = h | \hat{\mathbf{z}})$ admits the closed-form expressions reported in the four Examples of Gibbs-type priors in Section 2.2.1.

4. Simulation studies. To assess the performance of ESBM in settings mimicking our motivating application, and quantify the advantages over state-of-the-art alternatives (Amini et al. (2013), Blondel et al. (2008), Côme et al. (2021), von Luxburg (2007), Zhang, Levina and Zhu (2016)), we consider three simulated networks of $V = 80$ nodes displaying different criminal block structures sampled from a SBM with $H_0 = 5$ groups, and block probabilities equal to either 0.75 or 0.25. As shown in Figure 3, the first network defines a horizontal criminal organization characterized by classical community structures of varying size. The second network provides, instead, a more challenging scenario, which exhibits a nested hierarchy of core-periphery, weak-community and disassortative patterns characterizing a vertical criminal organization. In particular, we assume the presence of two equally-sized macro-groups, each having a small fraction of bosses that interact with all the affiliates of their associated group and also with an additional cluster of higher-level bosses. Finally, the last simulated network resembles more closely the block structures of the *Infinito network*, where we expect community patterns among the affiliates in each *locale*, core-periphery structures between such affiliates and the corresponding bosses, and assortative behaviors among the bosses of the different *locali*, resulting from covering strategies.

As we will clarify in Table 3, state-of-the-art methodologies (Amini et al. (2013), Blondel et al. (2008), Côme et al. (2021), von Luxburg (2007), Zhang, Levina and Zhu (2016)) applied to these three networks mostly fail in recovering the true underlying blocks and show a general tendency to over-collapse different groups, possibly due to their inability to incorporate unbalanced noisy partitions and effectively exploit attribute information. Such results motivate implementation of ESBM, both without and with node attributes coinciding, in this case, with the true partition \mathbf{z}_0 . Such a choice is useful for assessing to what extent the supervised Gibbs-type priors and relevant competitors can effectively exploit truly informative node attributes.

Within the Gibbs-type class, we first assess the four representative unsupervised priors for \mathbf{z} presented in Table 1, and then check whether introducing informative node attributes further improves the performance in each scenario. The hyperparameters are specified so that the

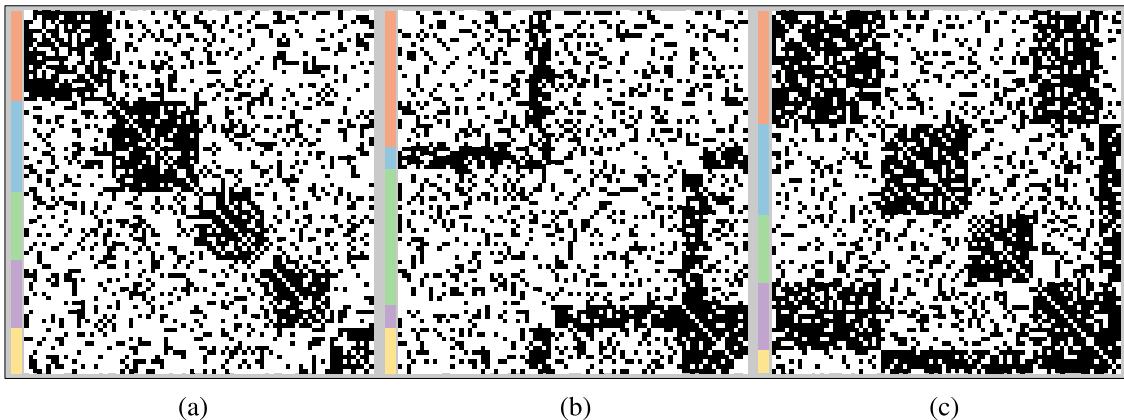


FIG. 3. Simulated adjacency matrices under the first (a), second (b) and third (c) scenario. Side colors correspond to the true partition \mathbf{z}_0 . Black cells refer to edges, whereas white cells denote nonedges. See the online article for the color version of this figure.

TABLE 2

Performance of ESBM in the three scenarios with $H_0 = 5$, when excluding attributes (UNSUP), and when supervising each prior with the true partition \mathbf{z}_0 as attribute (SUP). Performance is measured by the WAIC, the posterior mean $\mathbb{E}[\text{VI}(\mathbf{z}, \mathbf{z}_0)|\mathbf{Y}]$ of the VI distance from \mathbf{z}_0 , the posterior median number of nonempty clusters H (first and third quartiles in brackets) and the distance $\text{VI}(\hat{\mathbf{z}}, \mathbf{z}_b)$ among the estimated partition $\hat{\mathbf{z}}$ and the 95% credible bound \mathbf{z}_b . Bolded values denote best performances among UNSUP priors within each column. Bolded gray cells denote best overall performance in each column

Scenario	WAIC			$\mathbb{E}[\text{VI}(\mathbf{z}, \mathbf{z}_0) \mathbf{Y}]$			H			$\text{VI}(\hat{\mathbf{z}}, \mathbf{z}_b)$		
	1	2	3	1	2	3	1	2	3	1	2	3
[UNSUP]												
DM	3551	3560	3540	0.42	0.75	0.52	8 [7,8]	6 [5,7]	6 [5,6]	0.70	0.97	0.69
DP	3551	3560	3540	0.41	0.74	0.51	7 [7,8]	6 [5,7]	6 [5,6]	0.69	0.96	0.69
PY	3551	3559	3540	0.38	0.71	0.50	7 [6,9]	6 [5,7]	6 [5,6]	0.70	0.88	0.65
GN	3550	3554	3536	0.29	0.64	0.46	5 [5,6]	5 [5,5]	5 [5,5]	0.59	0.83	0.60
[SUP]												
DM	3523	3513	3517	0.09	0.16	0.13	6 [5,6]	5 [5,6]	5 [5,5]	0.25	0.32	0.33
DP	3523	3513	3517	0.09	0.16	0.14	6 [5,6]	5 [5,6]	5 [5,5]	0.25	0.32	0.33
PY	3522	3512	3517	0.07	0.15	0.13	6 [5,6]	5 [5,5]	5 [5,5]	0.20	0.32	0.31
GN	3521	3510	3515	0.04	0.14	0.12	5 [5,5]	5 [5,5]	5 [5,5]	0.14	0.30	0.28

prior expected number $\mathbb{E}_{\text{DM}}(H)$, $\mathbb{E}_{\text{DP}}(H)$, $\mathbb{E}_{\text{PY}}(H)$ and $\mathbb{E}_{\text{GN}}(H)$ of nonempty groups under the different priors is close to $10 > H_0$, whereas Algorithm 1 is initialized with every node in a different cluster. In this way we can check robustness of the results to hyperparameter settings and to the initialization of the Gibbs sampler. Specifically, we set $\bar{H} = 50$ and $\beta = 3.5/50$ for the DM, $\alpha = 3$ in the DP, $\sigma = 0.6$ and $\alpha = -0.3$ under the PY, and $\gamma = 0.45$ for the GN. In implementing such models, we consider the default uniform setting $a = b = 1$ for the prior on the block probabilities (e.g., Geng, Bhattacharya and Pati (2019), Nowicki and Snijders (2001)), and let $\alpha_1 = \dots = \alpha_C = 1$ in (5), when including node attributes.

From Algorithm 1, we obtain 40,000 samples for \mathbf{z} , after a conservative burn-in of 10,000. In our experiments, inference has proven robust to different initializations of \mathbf{z} in Algorithm 1, including extreme settings with all the nodes in a single group. Nonetheless, starting with one cluster for every node provides the best overall mixing, when monitored on the chain for the likelihood in (1) evaluated at the MCMC samples of \mathbf{z} . Graphical analysis of the traceplots for such a chain suggests rapid convergence and effective mixing under all models. Algorithm 1 provides 150 samples of \mathbf{z} per second when executed on an iMac with 1 Intel Core i5 3.4 GHZ processor and 8 GB RAM, thus showing good efficiency. Table 2 summarizes the performance of the four priors.

Among all the unsupervised Gibbs-type priors considered for \mathbf{z} , the Gneden process always yields slightly improved performance in terms of WAIC and posterior mean of the VI distance from the true partition \mathbf{z}_0 . In addition, it also offers a more accurate learning of the number of groups, with tighter interquartile ranges that always include the true $H_0 = 5$, and tighter credible balls around the VI-optimal posterior point estimate $\hat{\mathbf{z}}$. In our experiments, the GN prior was also the less sensitive to hyperparameter settings, although comparable robustness was observed even for DM, DP and PY under moderate changes of the hyperparameters. For instance, setting these hyperparameters to induce an expected value on H under all priors of $5 = H_0$ instead of 10, did not change the final conclusions provided by Table 2.

As expected, including informative attributes further improves performance of all unsupervised priors in each scenario, effectively lowering $\mathbb{E}[\text{VI}(\mathbf{z}, \mathbf{z}_0)|\mathbf{Y}]$, and further shrinking the credible balls. In a sense, this is the best setting, since we consider the true \mathbf{z}_0 as a node

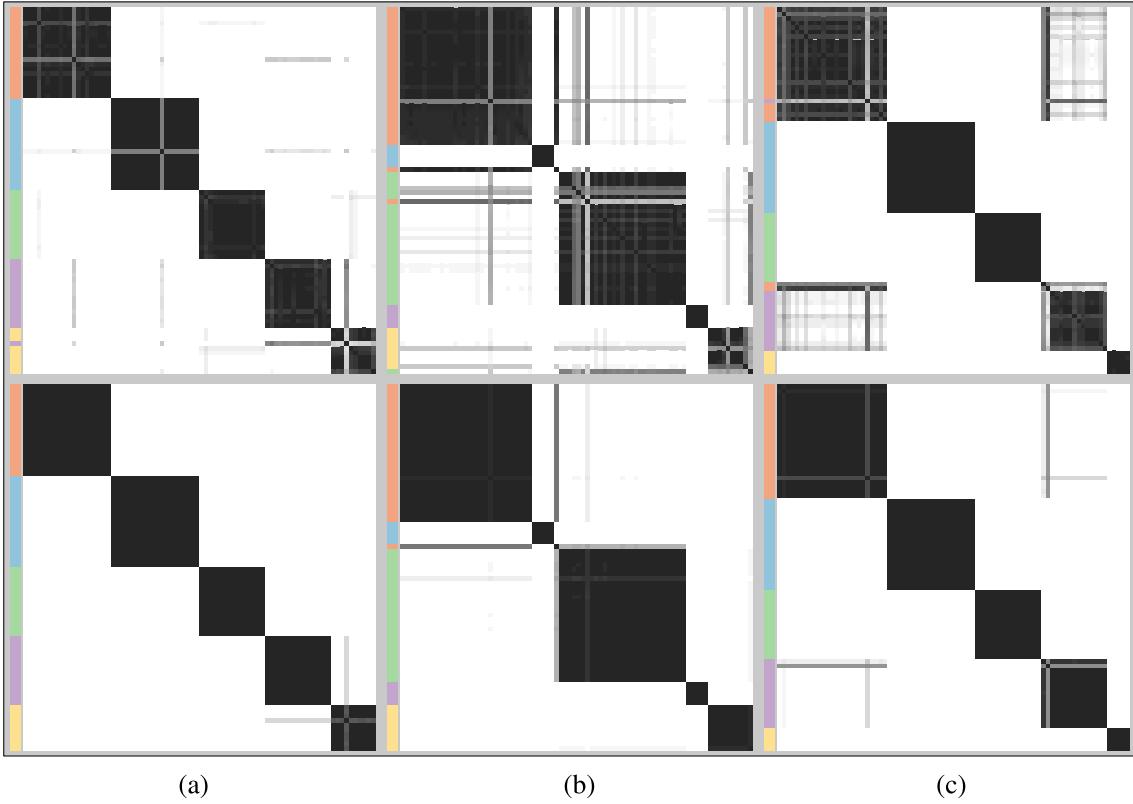


FIG. 4. For the first (a), second (b) and third (c) scenario, posterior similarity matrices under the Gnedin process from the ESBM without (first row) and with (second row) node attributes, respectively. Cell colors range from white to black as the estimated co-clustering probability of the associated pair of nodes goes from 0 to 1. Side colors correspond to estimated partitions $\hat{\mathbf{z}}$. See the online article for the color version of this figure.

attribute. We also tried supervising with a random permutation of \mathbf{z}_0 . This resulted in a slight performance deterioration relative to the unsupervised GN prior, which is doubly reassuring. In fact, on one hand it shows that, under the proposed model selection criteria, an unsupervised prior would be preferred to one with noninformative attributes. On the other, the fact that performance deterioration is not dramatic suggests robustness in learning. According to the *posterior similarity matrices* in Figure 4, unbalanced partitions are harder to infer, especially without attributes. However, this gap vanishes when including informative attributes that can successfully support inference and reduce posterior uncertainty. All misclassification errors for in-sample edge prediction are about 0.24, almost matching the one expected under the true model. This suggests accurate calibration and a tendency to avoid overfitting in ESBMs. Such a property is further confirmed by the performance in predicting, via (13), the group membership for 300 new nodes, among which 50 are simulated from a cluster not yet observed in the original networks. For this task, the missclassification errors under the supervised GN prior are 0.01, 0.08 and 0.04 in the first, second and third scenario, respectively.

To further clarify the magnitude of the improvements provided by the ESBM, Table 3 compares the performance of GN prior—which proved the more accurate in Table 2—with the results obtained under the state-of-the-art alternative methods (Amini et al. (2013), Blondel et al. (2008), Côme et al. (2021), von Luxburg (2007), Zhang, Levina and Zhu (2016)) discussed in Section 1.2. Since most of these competitors are non-Bayesian and only provide a point estimate $\hat{\mathbf{z}}$ of \mathbf{z} , Table 3 focuses on measures of accuracy in point estimation to facilitate comparison among the different methods. In estimating H under spectral clustering, we consider a variety of model selection criteria available in the R library `randnet`, and set

TABLE 3

For the three simulation scenarios, performance comparison between ESBM with GN prior, and state-of-the-art unsupervised and supervised competitors in the R libraries `igraph`, `randnet`, `greed` and `JCDC`. These include the Louvain algorithm ([Blondel et al. \(2008\)](#)), Spectral clustering ([von Luxburg \(2007\)](#)), Regularized Spectral clustering ([Amini et al. \(2013\)](#)), the greed clustering algorithm for SBM and degree corrected SBM (DC–SBM) ([Côme et al. \(2021\)](#)) and the attribute-assisted `JCDC` community detection algorithm ([Zhang, Levina and Zhu \(2016\)](#)). The assessment focuses on the estimated number \hat{H} of nonempty groups, the VI distance $VI(\hat{\mathbf{z}}, \mathbf{z}_0)$ between the estimated and true partitions, and the absolute error (multiplied by 10) between the estimated and true edge probabilities, averaged across the $V(V - 1)/2$ node pairs. Bolded values denote best performances among unsupervised methods within each column, whereas bolded gray cells denote best overall performance within each column

Scenario	\hat{H}			$VI(\hat{\mathbf{z}}, \mathbf{z}_0)$			ERROR [EST] ($\times 10$)		
	1	2	3	1	2	3	1	2	3
[UNSUP]									
ESBM (GN)	5	5	5	0.13	0.40	0.37	0.30	0.28	0.31
Louvain	4	4	3	0.30	2.90	0.81	0.40	1.24	0.51
Spectral	4	4	3	0.56	2.81	0.81	0.45	1.32	0.51
Reg. Spectral	4	4	3	0.56	2.63	0.81	0.45	1.21	0.51
greed (SBM)	4	5	4	0.41	0.27	0.48	0.44	0.27	0.28
greed (DC–SBM)	2	1	2	1.47	1.94	1.18	1.05	1.26	0.84
[SUP]									
ESBM (GN)	5	5	5	0.00	0.16	0.00	0.22	0.26	0.23
JCDC ($w_n = 5$)	4	4	3	0.42	2.83	0.81	0.40	1.16	0.51
JCDC ($w_n = 1.5$)	4	4	3	0.30	2.02	0.70	0.40	1.12	0.47

\hat{H} equal to the median of the values of H estimated under the different strategies. These include the Beth–Hessian solution from [Le and Levina \(2015\)](#), the likelihood ratio strategy by [Wang and Bickel \(2017\)](#) and the cross-validation methods developed in [Chen and Lei \(2018\)](#) and [Li, Levina and Zhu \(2020\)](#). This estimate for H is also used as a sensible starting value to initialize the greedy clustering algorithm for SBM and DC–SBM in the R library `greed` ([Côme et al. \(2021\)](#)). As shown in the R manual of the `greed` library, this strategy estimates \mathbf{z} under a Dirichlet-multinomial prior for the group membership indicators. Hence, to make results comparable with the proposed ESBM class, we set the Dirichlet hyperparameter in `greed` equal to 3.5/50, as done for the DM prior under ESBM. Among the available methods that leverage attribute information, we consider the community detection algorithm proposed by [Zhang, Levina and Zhu \(2016\)](#), under different default values for the tuning parameters, and setting again $H = \hat{H}$. This strategy has been shown in [Zhang, Levina and Zhu \(2016\)](#) to yield improved empirical performance relative to other powerful attribute-assisted solutions, thereby providing a suitable benchmark competitor.

As illustrated in Table 3, the above competitors display a tendency to systematically underestimate the true number of nonempty groups, and exhibit reduced accuracy in learning the true partition and the exact edge probabilities, relative to ESBM with GN prior. This accuracy reduction is further affected by the difficulties in learning more complex block structures beyond communities, which affect performance even when supervising the algorithms with the true underlying partition \mathbf{z}_0 . The `greed` clustering algorithm for SBM ([Côme et al. \(2021\)](#)) is overall the closest in performance to the proposed ESBM with GN prior and, in additional studies, we found that its performance can be typically improved by setting hyperparameters and starting values more extreme than those underlying the true data generative process. While this choice is possible, in practice the truth is unknown, and hence, a more data-driven strategy to set these quantities, as the one we consider for the `greed` algorithm evaluated in

Table 3, is more desirable in general. The unsupervised and supervised ESBM with GN prior always yield accurate point estimates of \mathbf{z} in all scenarios, and unlike the competitors under analysis, further allow principled uncertainty quantification and not just point estimation. As expected, the output of the greedy clustering algorithm by Côme et al. (2021) in Table 3 points clearly toward SBM rather than DC–SBM in all the three scenarios. This result is further confirmed by the state-of-the-art model selection strategies implemented in the functions `NCV.select` (Chen and Lei (2018)) and `ECV.block` (Li, Levina and Zhu (2020)) of the R library `randnet`.

5. Application to the *Infinito network*. We apply the approach developed in Sections 2–3 to the *Infinito network* presented in Section 1.1. Despite its potential in unveiling the internal organization of 'Ndrangheta, such a network has received little attention within the statistical literature, apart from some initial analyses in Calderoni and Piccardi (2014) and Calderoni, Brunetto and Piccardi (2017). These two contributions have the merit of providing early results on the relevance of block structures as key sources of knowledge to shed light on the internal architecture of criminal organizations. However, the overarching focus is on classical community structures and their relation with suspect attributes, such as *locali* affiliation and role. As clarified in Section 1 and in the simulation studies in Section 4, this approach rules out recurring block structures in criminal networks, fails to formally include error-prone attributes in the modeling process, and lacks extensive methods for uncertainty quantification, model selection and prediction.

To address these issues and obtain a deeper understanding of the internal structure behind *La Lombardia*, we provide an in-depth analysis of the *Infinito network* under the ESBM class. As for the simulations in Section 4, we first identify a suitable candidate model by comparing the performance of the unsupervised and supervised priors for \mathbf{z} presented in Sections 2.2.1–2.2.2, with hyperparameters inducing 20 expected clusters a priori. This value is four times the number of *locali* in the network, which seems reasonably conservative. In particular, we let $\bar{H} = 50$ and $\beta = 12/50$ for the DM, $\alpha = 8$ in the DP, $\sigma = 0.725$ and $\alpha = -0.350$ for the PY, and $\gamma = 0.3$ under GN. Posterior inference relies again on 40,000 MCMC samples produced by Algorithm 1, after a burn-in of 10,000. The traceplots for the likelihood in (1) suggest adequate mixing and rapid convergence as in the simulations, with similar running times. Also in this case, the results were overall robust to initialization and moderate changes in the hyperparameter settings.

As clarified in Table 4, GN yields the best performance also in the *Infinito network*, relative to the other examples of Gibbs-type priors commonly implemented in network studies. This provides quantitative support for the conjecture in Section 2.2.1 on the suitability of GN as a realistic prior for the grouping structures in organized crime. Moreover, as seen in Table 4, supervising the priors with the additional information on role and *locale* affiliation leads to a further reduction in the WAIC and lower posterior uncertainty, meaning that such attributes carry information about 'Ndrangheta modules. Calderoni and Piccardi (2014) and Calderoni, Brunetto and Piccardi (2017) investigated similar effects, but with a main focus on descriptive analyses of classical community structures, thus obtaining results that partially depart from the expected vertical architecture of 'Ndrangheta (Catino (2014), Paoli (2007)). In fact, the authors obtain communities defined by unions of multiple *locali*, and seem unable to separate affiliates from bosses throughout the partition process. As shown in panel (a) of Figure 2, this tendency is confirmed when applying the Louvain algorithm (Blondel et al. (2008)) to the *Infinito network*. Compared to the ESBM in the panel (c) of Figure 2, the Louvain algorithm provides an overly coarsened view of the block structures in the *Infinito network*.

Recalling major forensic theories on organized crime (e.g., Catino (2014), Paoli (2007)), our conjecture is that 'Ndrangheta displays more complex block structures in which the pure

TABLE 4

Performance of ESBM in the Infinito network, when excluding attributes (UNSUP), and when supervising each prior with role-locale information (SUP). Performance is measured by the WAIC. Bolded values denote best performance among the UNSUP priors. Bolded gray cells indicate best overall performance. We also provide the posterior median number of nonempty clusters H (first and third quartiles in brackets), and the distance $VI(\hat{\mathbf{z}}, \mathbf{z}_b)$ among the estimated partition $\hat{\mathbf{z}}$ and the 95% credible bound \mathbf{z}_b

	WAIC		H		VI($\hat{\mathbf{z}}, \mathbf{z}_b$)	
	UNSUP	SUP	UNSUP	SUP	UNSUP	SUP
DM	1229	1199	14 [14,15]	15 [15,15]	0.28	0.16
DP	1256	1199	14 [14,14]	15 [15,16]	0.22	0.28
PY	1280	1226	14 [14,14]	15 [14,15]	0.30	0.20
GN	1205	1194	15 [15,15]	15 [15,16]	0.32	0.22

communities among the affiliates within each *locale* are combined with higher-level coordinating block structures between the bosses. Unlike classical community detection algorithms, the ESBM crucially accounts for these architectures, thus providing unprecedented empirical evidence in support of such forensic theories, as illustrated in Figures 5–6. These graphical assessments are based on a point estimate $\hat{\mathbf{z}}$ of the partition structure under the supervised GN process prior, which we consider in the subsequent analyses of the *Infinito network*, due to its superior performance in Table 4 and the relatively low posterior uncertainty around the estimated partition $\hat{\mathbf{z}}$ —the radius of the credible ball is far below the maximum achievable VI distance of $\log_2 84 \approx 6.39$. To formally confirm the forensic hypotheses, we compute the difference in WAIC between the unsupervised and supervised GN prior, with suspects' attribute \mathbf{X} defining the conjectured structure. In particular, the class of each affiliate corresponds to the associated *locale*, whereas all the bosses share a common label indicating that such members have a leadership role in the organization. Moreover, a subset of the affiliates of the purple *locale* who are known from the judicial acts¹ to cover a peripheral role are assigned a distinct label. The resulting difference in WAIC is ≈ 11 , which provides a strong evidence in favor of our conjecture, when compared with the thresholds suggested for related information criteria (e.g., Gelman, Hwang and Vehtari (2014), Spiegelhalter et al. (2002)).

As illustrated in Figure 2, these fundamental structures are hidden not only to community detection algorithms (Blondel et al. (2008)), but also to spectral clustering solutions (von Luxburg (2007)), which account for more complex block structures. This is further confirmed by the higher values for the deviance $\mathcal{D} = -2 \log p(\mathbf{Y}|\hat{\mathbf{z}})$ under the state-of-the-art competitors discussed in Section 1, and evaluated in Section 4. More specifically, the estimated partitions under Louvain (Blondel et al. (2008)), Spectral (von Luxburg (2007)), Reg. Spectral (Amini et al. (2013)), greed (SBM) (Côme et al. (2021)), JCDC ($w_n = 5$) and JCDC ($w_n = 1.5$) (Zhang, Levina and Zhu (2016)) yield deviances of 2371, 2109, 1954, 1601, 2105 and 2163, respectively, whereas those obtained under the unsupervised and supervised GN process prior are 1553 and 1549, respectively. As for the simulation study, the Dirichlet hyperparameter for the greed algorithm is set equal to the same value 12/50 considered for the DM prior under ESBM in the application. Similarly, any time an estimate or a starting value for H is required to implement one of the competitors, we set it equal to the median of the values of H given by different selection strategies (Chen and Lei (2018), Le and Levina (2015), Li, Levina and Zhu (2020), Wang and Bickel (2017)). Since this estimate of H is lower than the one obtained under the GN prior, we also compute the deviances leveraging the same number of nonempty clusters $\hat{H} = 16$ inferred by the GN process, thus providing an assessment not affected by the different model complexities. This alternative implementation yields the same conclusions, thereby confirming the superior performance of

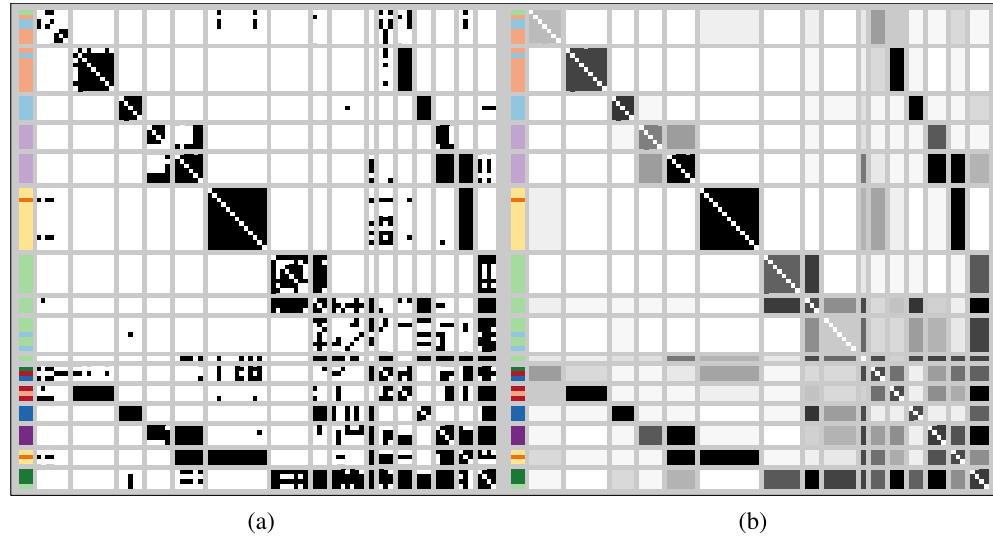


FIG. 5. *Adjacency matrix (a) and estimated edge-probability matrix (b) of the Infinito network with nodes reordered and partitioned in blocks according to the clustering structure estimated under ESBM with supervised GN prior. Side colors correspond to the different locali, with darker and lighter shades denoting bosses and affiliates, respectively. See the online article for the color version of this figure.*

the ESBM class also in this application. To evaluate the plausibility of the SBM assumption relative to its degree-corrected version (Karrer and Newman (2011)), we further studied the output of the R functions `NCV.select` (Chen and Lei (2018)) and `ECV.block` (Li, Levina and Zhu (2020)) in the R library `randnet`, which allow to formally select between SBM and DC-SBM. Both strategies provide support in favor of SBM in this specific application.

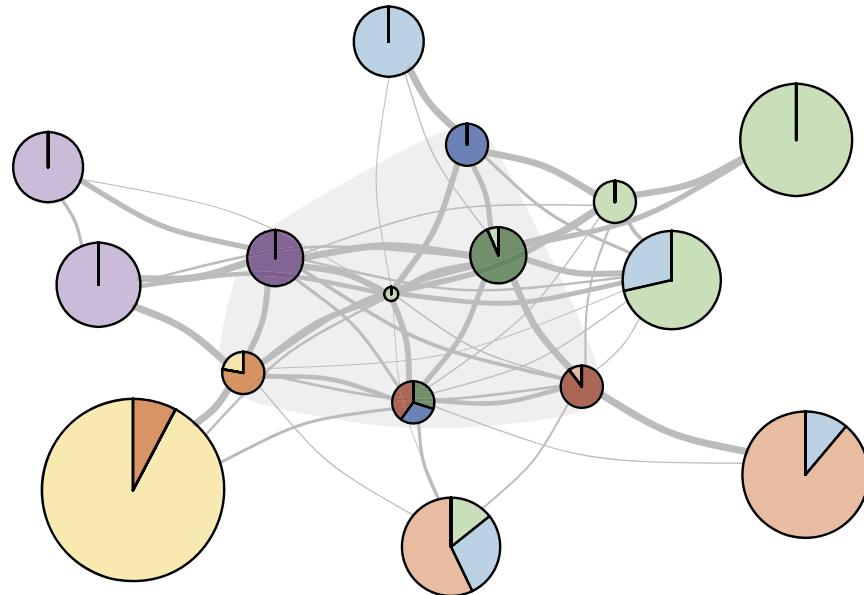


FIG. 6. *Network representation of the inferred clusters in the Infinito network. Each node denotes one cluster and edges are weighted by the estimated block probabilities. Node sizes are proportional to cluster cardinalities, while the pie charts represent compositions with respect to locale affiliation and leadership role; colors are the same as in Figure 5. To provide more direct insights, the composition with respect to role in the smaller-sized pie charts is reweighted to account for the fact that bosses are less frequent in the network relative to affiliates. Node positions are obtained via force-directed placement (Fruchterman and Reingold (1991)) to reflect strength of connections. See the online article for the color version of this figure.*

The above results are also confirmed in Figure 2, which clearly highlights the improved ability of the supervised GN prior in learning the block structures that characterize the *Infinito network*. According to Figures 5 and 6, these modules suggest a nested partition structure mainly defined by the two macro-blocks of affiliates and bosses, which are further partitioned in sub-groups mostly coherent with the *locale* affiliation. The affiliates' groups typically exhibit community patterns and connect to the hidden core mainly through the bosses of the corresponding *locale*, which in turn display weak assortative structures in the higher-level coordinating architecture among bosses of different *locali*.

Figure 7 confirms this result by showing how affiliates' groups are typically characterized by high local transitivity and low betweenness, whereas clusters of bosses display the opposite behavior. This is a fundamental finding, which provides new empirical evidence on the attempt of 'Ndrangheta bosses to address the tradeoff between efficiency and security (Morselli, Giguère and Petit (2007)) via the creation of low-sized, sparse and secure core groups with a high betweenness that favors the flow of information toward larger and dense groups of affiliates, which guarantee efficiency. Besides these recurring architectures, the flexibility of ESBM is also able to account for other informative local deviations. For instance, the first group in Figure 5 comprises affiliates from different *locali*, who were found in judicial acts¹ to have peripheral roles. Similarly, the moderate block-connectivity patterns between the purple *locale* and the yellow one in Figures 5–6, are consistent with the fact that the latter was created as a branching of the former.¹ The green *locale* has instead more complex block structures among affiliates, with a fragmentation in various subgroups denoting middle-level leadership positions. According to the judicial acts,¹ these positions typically refer to authority roles in overseeing criminal actions or in guaranteeing coordination between *La Lombardia* and the leading 'Ndrangheta families in Calabria. Similar roles are covered also by the small fraction of affiliates allocated to groups of bosses. Among these affiliates, it is worth highlighting the suspect allocated to the single-node cluster with the most central position in Figure 6. While not being classified as a boss in the judicial acts,¹ such a suspect is a senior member of high rank in the organization with fundamental mediating roles between all the *locali*, and with the leading 'Ndrangheta families in Calabria. Hence, the actual position of such an affiliate in the vertical structure of *La Lombardia* may be much higher than currently reported.

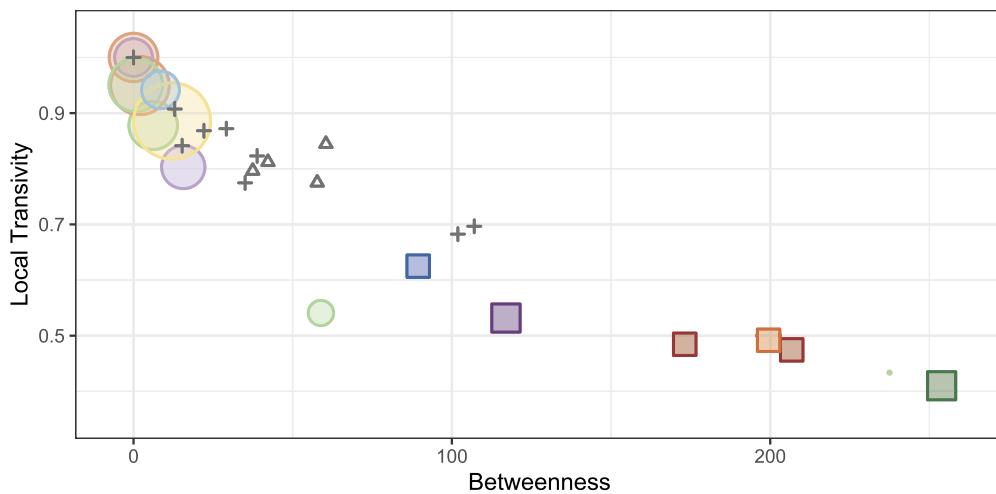


FIG. 7. Scatter plot of average betweenness and local transitivity for each estimated cluster under the supervised GN prior. Sizes are proportional to cluster cardinalities, whereas the color of each point is set equal to the one occupying the largest portion of the associated pie chart in Figure 6. Circles and squares represent groups mostly referring to affiliates and bosses, respectively, while the Δ and + symbols denote cluster-specific measures computed from the partitions estimated under the Louvain algorithm (Blondel et al. (2008)) (Δ) and spectral clustering (von Luxburg (2007)) (+). See the online article for the color version of this figure.

As clarified in Figures 2 and 7, all the above structures cannot be inferred under state-of-the-art alternatives and, therefore, open new avenues to obtain a substantially improved understanding of the criminal network organization under ESBM, along with refined predictive strategies for incoming affiliates. In particular, the predictive methods in Section 3.2 applied to the 34 held-out suspects in the *Infinito network* crucially allow to recognize the role of incoming criminals without the need to use external information, that may not be available when a suspect is first observed. In fact, classifying the 34 held-out suspects via (13) with an unsupervised GN prior favors allocation of new affiliates to current clusters characterized by high normalized local transitivity and low normalized betweenness, whereas incoming bosses are assigned to groups with much lower difference among these quantities. More specifically, the average difference between the two measures is 0.88 for the held-out affiliates, and 0.20 for the held-out bosses.

6. Discussion and future research directions. Criminal networks provide a fundamental field of application where the advancements in network science can have a major societal impact. However, despite the relevance of such studies, there has been limited consideration of criminal networks in the statistical literature, and the focus has been largely on restrictive methods that offer limited knowledge on the internal structure of criminal organizations. To cover this gap, we proposed ESBMs as a broad class of realistic models that unifies most existing SBMs via Gibbs-type priors. Besides providing a single methodological, theoretical and computational framework for various SBMs, such a generalization facilitates the proposal of new models by exploring alternative options within the Gibbs-type class, and allows natural inclusion of attributes via connections with PPMs. Both aspects are fundamental to investigate criminal networks. For example, we have shown in simulations that the Gneden process, which to the best of our knowledge had never been used in SBMs, yields a suitable prior for partition structures in organized crime, and can improve the performance of already implemented DP, PY and DM in various realistic criminal networks where routine strategies, such as community detection and spectral clustering, fail. The motivating *Infinito network* application clarifies the benefits of our extended class of models and methods, providing formal unprecedented empirical evidence to several forensic theories on the internal functioning of complex criminal organizations, such as 'Ndrangheta.

The present work offers also many future directions of research. For example, the general and modular structure of ESBMs motivates application to modern real-world networks beyond criminal ones, and facilitates extensions to directed, bipartite and weighted networks. To address this goal, it is sufficient to substitute the beta-binomial likelihood in (1) with suitable ones, such as gamma-Poisson for count edges and Gaussian–Gaussian for continuous ones. Other types of suspect attributes beyond categorical ones can also be easily included leveraging the default choices suggested by Müller, Quintana and Rosner (2011) for $p(\cdot)$ in (4) under continuous, ordinal and count-type attributes. Additional applications to other criminal networks and further extensions to alternative representations, such as the mixed membership SBM (Airoldi et al. (2008), Ranciati, Vinciotti and Wit (2020)) and degree corrected SBM (Karrer and Newman (2011)), are also worthy of exploration. Despite the relevance of such constructions, we shall emphasize that while ESBM preserves interpretability and parsimony by avoiding mixed membership structures, it still allows quantification of uncertainty in the degree of affiliation to different groups via formal inference on the posterior similarity matrix and on the credible bounds. Finally, although studying the coverage properties of the credible balls presented in Section 3.2 is still an ongoing area of research that goes beyond the scope of the present article (Wade and Ghahramani (2018)), it would be of interest to empirically check such properties within the ESBM context.

Acknowledgments. We are grateful to the Editor, the Associate Editor and the anonymous referees for the valuable comments and the constructive feedbacks, which helped us in improving the preliminary version of this article. We also thank Omiros Papaspiliopoulos for the insightful discussion on model selection techniques.

SUPPLEMENTARY MATERIAL

R code and data to reproduce the analyses in “Extended stochastic block models with application to criminal networks.” (DOI: [10.1214/21-AOAS1595SUPP](https://doi.org/10.1214/21-AOAS1595SUPP); .zip). This directory contains R code, data and detailed guidelines to implement the ESBM class and reproduce all the analyses in Sections 4 and 5. For the most recent and updated version of the code, see <https://github.com/danieledurante/ESBM>.

REFERENCES

- ABBE, E. (2017). Community detection and stochastic block models: Recent developments. *J. Mach. Learn. Res.* **18** 1–86. [MR3827065](#)
- AGRESTE, S., CATANESE, S., DE MEO, P., FERRARA, E. and FIUMARA, G. (2016). Network structure and resilience of mafia syndicates. *Inform. Sci.* **351** 30–47.
- AIROLDI, E. M., BLEI, D. M., FIENBERG, S. E. and XING, E. P. (2008). Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.* **9** 1981–2014.
- AMINI, A., CHEN, A., BICKEL, P. J. and LEVINA, E. (2013). Pseudo-likelihood methods for community detection in large sparse networks. *Ann. Statist.* **41** 2097–2122. [MR3127859](#) <https://doi.org/10.1214/13-AOS1138>
- ATHREYA, A., FISHKIND, D. E., TANG, M., PRIEBE, C. E., PARK, Y., VOGELSTEIN, J. T., LEVIN, K., LYZINSKI, V., QIN, Y. and SUSSMAN, D. L. (2017). Statistical inference on random dot product graphs: A survey. *J. Mach. Learn. Res.* **18** 1–92. [MR3827114](#)
- BICKEL, P. J., CHOI, D., CHANG, X. and ZHANG, H. (2013). Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *Ann. Statist.* **41** 1922–1943. [MR3127853](#) <https://doi.org/10.1214/13-AOS1124>
- BINKIEWICZ, N., VOGELSTEIN, J. T. and ROHE, K. (2017). Covariate-assisted spectral clustering. *Biometrika* **104** 361–377. [MR3698259](#) <https://doi.org/10.1093/biomet/asx008>
- BLONDEL, V. D., GUILLAUME, J. L., LAMBIOTTE, R. and LEFEBVRE, E. (2008). Fast unfolding of communities in large networks. *J. Stat. Mech.* **10** P10008.
- CALDERONI, F., BRUNETTO, D. and PICCARDI, C. (2017). Communities in criminal networks: A case study. *Soc. Netw.* **48** 116–125.
- CALDERONI, F. and PICCARDI, C. (2014). Uncovering the structure of criminal organizations by community analysis: The Infinito network. In 2014 *Tenth International Conference on Signal–Image Technology and Internet-Based Systems* 301–308. IEEE, Marrakech, Morocco.
- CAMPANA, P. (2016). Explaining criminal networks: Strategies and potential pitfalls. *Methodol. Innov.* **9** 1–10.
- CAMPANA, P. and VARESE, F. (2022). Studying organized crime networks: Data sources, boundaries and the limits of structural measures. *Soc. Netw.* **69** 149–159.
- CARLEY, K. M., LEE, J.-S. and KRACKHARDT, D. (2002). Destabilizing networks. *Connections* **24** 79–92.
- CATINO, M. (2014). How do mafias organize? Conflict and violence in three mafia organizations. *Eur. J. Sociol.* **55** 177–220.
- CAVALLARO, L., FICARA, A., DE MEO, P., FIUMARA, G., CATANESE, S., BAGDASAR, O., SONG, W. and LIOTTA, A. (2020). Disrupting resilient criminal networks through data analysis: The case of Sicilian Mafia. *PLoS ONE* **15** 1–22.
- CHEN, K. and LEI, J. (2018). Network cross-validation for determining the number of communities in network data. *J. Amer. Statist. Assoc.* **113** 241–251. [MR3803461](#) <https://doi.org/10.1080/01621459.2016.1246365>
- CÔME, E. and LATOUCHE, P. (2015). Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood. *Stat. Model.* **15** 564–589. [MR3441229](#) <https://doi.org/10.1177/1471082X15577017>
- CÔME, E., JOUVIN, N., LATOUCHE, P. and BOUVEYRON, C. (2021). Hierarchical clustering with discrete latent variable models and the integrated classification likelihood. *Adv. Data Anal. Classif.* **15** 957–986. [MR4333226](#) <https://doi.org/10.1007/s11634-021-00440-z>
- DE BLASI, P., LIJOI, A. and PRÜNSTER, I. (2013). An asymptotic analysis of a class of discrete nonparametric priors. *Statist. Sinica* **23** 1299–1321. [MR3114715](#)

- DE BLASI, P., FAVARO, S., LIJOI, A., MENA, R. H., PRÜNSTNER, I. and RUGGIERO, M. (2015). Are Gibbs-type priors the most natural generalization of the Dirichlet process? *IEEE Trans. Pattern Anal. Mach. Intell.* **37** 212–229.
- DIVIÁK, T. (2022). Key aspects of covert networks data collection: Problems, challenges, and opportunities. *Soc. Netw.* **69** 160–169.
- FAUST, K. and TITA, G. E. (2019). Social networks and crime: Pitfalls and promises for advancing the field. *Annu. Rev. Criminol.* **2** 99–122.
- FERRARA, E., DE MEO, P., CATANESE, S. and FIUMARA, G. (2014). Detecting criminal organizations in mobile phone networks. *Expert Syst. Appl.* **41** 5733–5750.
- FORTUNATO, S. and HRIC, D. (2016). Community detection in networks: A user guide. *Phys. Rep.* **659** 1–44. MR3566093 <https://doi.org/10.1016/j.physrep.2016.09.002>
- FOSDICK, B. K., MCCORMICK, T. H., MURPHY, T. B., NG, T. L. J. and WESTLING, T. (2019). Multiresolution network models. *J. Comput. Graph. Statist.* **28** 185–196. MR3939381 <https://doi.org/10.1080/10618600.2018.1505633>
- FRUCHTERMAN, T. M. and REINGOLD, E. M. (1991). Graph drawing by force-directed placement. *Softw. Pract. Exp.* **21** 1129–1164.
- GELMAN, A., HWANG, J. and VEHTARI, A. (2014). Understanding predictive information criteria for Bayesian models. *Stat. Comput.* **24** 997–1016. MR3253850 <https://doi.org/10.1007/s11222-013-9416-2>
- GENG, J., BHATTACHARYA, A. and PATI, D. (2019). Probabilistic community detection with unknown number of communities. *J. Amer. Statist. Assoc.* **114** 893–905. MR3963189 <https://doi.org/10.1080/01621459.2018.1458618>
- GIRVAN, M. and NEWMAN, M. E. J. (2002). Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **99** 7821–7826. MR1908073 <https://doi.org/10.1073/pnas.122653799>
- GNEDIN, A. (2010). A species sampling model with finitely many types. *Electron. Commun. Probab.* **15** 79–88. MR2606505 <https://doi.org/10.1214/ECP.v15-1532>
- GNEDIN, A. and PITMAN, J. (2005). Exchangeable Gibbs partitions and Stirling triangles. *Zap. Nauchn. Sem. (POMI) S.-Peterburg.* **325** 83–102. MR2160320 <https://doi.org/10.1007/s10958-006-0335-z>
- GORMLEY, I. C. and MURPHY, T. B. (2010). A mixture of experts latent position cluster model for social network data. *Stat. Methodol.* **7** 385–405. MR2643609 <https://doi.org/10.1016/j.stamet.2010.01.002>
- GRASSI, R., CALDERONI, F., BIANCHI, M. and TORRIERO, A. (2019). Betweenness to assess leaders in criminal networks: New evidence using the dual projection approach. *Soc. Netw.* **56** 23–32.
- HANDCOCK, M. S., RAFTERY, A. E. and TANTRUM, J. M. (2007). Model-based clustering for social networks. *J. Roy. Statist. Soc. Ser. A* **170** 301–354. MR2364300 <https://doi.org/10.1111/j.1467-985X.2007.00471.x>
- HARTIGAN, J. A. (1990). Partition models. *Comm. Statist. Theory and Methods* **19** 2745–2756. MR1088047 <https://doi.org/10.1080/03610929008830345>
- HOLLAND, P. W., LASKEY, K. B. and LEINHARDT, S. (1983). Stochastic blockmodels: First steps. *Soc. Netw.* **5** 109–137. MR0718088 [https://doi.org/10.1016/0378-8733\(83\)90021-7](https://doi.org/10.1016/0378-8733(83)90021-7)
- KARRER, B. and NEWMAN, M. E. J. (2011). Stochastic blockmodels and community structure in networks. *Phys. Rev. E* (3) **83** 016107, 10. MR2788206 <https://doi.org/10.1103/PhysRevE.83.016107>
- KASS, R. E. and RAFTERY, A. E. (1995). Bayes factors. *J. Amer. Statist. Assoc.* **90** 773–795. MR3363402 <https://doi.org/10.1080/01621459.1995.10476572>
- KEMP, C., TENENBAUM, J. B., GRIFFITHS, T. L., YAMADA, T. and UEDA, N. (2006). Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence* 381–388.
- KIM, D., HUGHES, M. and SUDDERTH, E. (2012). The nonparametric metadata dependent relational model. In *ICML'12: Proceedings of the 29th International Conference on Machine Learning* 1411–1418. IEEE, Edinburgh, UK.
- KREBS, V. E. (2002). Mapping networks of terrorist cells. *Connections* **24** 43–52.
- LE, V. (2012). Organised crime typologies: Structure, activities and conditions. *Int. J. Criminol. Sociol.* **1** 121–131.
- LE, C. M. and LEVINA, E. (2015). Estimating the number of communities in networks by spectral methods. Preprint. Available at [arXiv:1507.00827](https://arxiv.org/abs/1507.00827).
- LEE, C. and WILKINSON, D. J. (2019). A review of stochastic block models and extensions for graph clustering. *Appl. Netw. Sci.* **4** 1–50.
- LEGARAMANTI, S., RIGON, T. and DURANTE, D. (2020). Bayesian testing for exogenous partition structures in stochastic block models. *Sankhya A*. In press.
- LEGARAMANTI, S., RIGON, T., DURANTE, D. and DUNSON, D. B (2022). Supplement to “Extended stochastic block models with application to criminal networks.” <https://doi.org/10.1214/21-AOAS1595SUPP>
- LEI, J. and RINALDO, A. (2015). Consistency of spectral clustering in stochastic block models. *Ann. Statist.* **43** 215–237. MR3285605 <https://doi.org/10.1214/14-AOS1274>

- LENK, P. (2009). Simulation pseudo-bias correction to the harmonic mean estimator of integrated likelihoods. *J. Comput. Graph. Statist.* **18** 941–960. [MR2750446](#) <https://doi.org/10.1198/jcgs.2009.08022>
- LI, T., LEVINA, E. and ZHU, J. (2020). Network cross-validation by edge sampling. *Biometrika* **107** 257–276. [MR4108931](#) <https://doi.org/10.1093/biomet/asaa006>
- LIJOI, A., MENA, R. H. and PRÜNSTNER, I. (2007a). Controlling the reinforcement in Bayesian non-parametric mixture models. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **69** 715–740. [MR2370077](#) <https://doi.org/10.1111/j.1467-9868.2007.00609.x>
- LIJOI, A., MENA, R. H. and PRÜNSTNER, I. (2007b). Bayesian nonparametric estimation of the probability of discovering new species. *Biometrika* **94** 769–786. [MR2416792](#) <https://doi.org/10.1093/biomet/asm061>
- LIJOI, A., PRÜNSTNER, I. and WALKER, S. G. (2008). Bayesian nonparametric estimators derived from conditional Gibbs structures. *Ann. Appl. Probab.* **18** 1519–1547. [MR2434179](#) <https://doi.org/10.1214/07-AAP495>
- LIU, F., CHOI, D., XIE, L. and ROEDER, K. (2018). Global spectral clustering in dynamic networks. *Proc. Natl. Acad. Sci. USA* **115** 927–932. [MR3763702](#) <https://doi.org/10.1073/pnas.1718449115>
- MAGALINGAM, P., DAVIS, S. and RAO, A. (2015). Using shortest path to discover criminal community. *Digit. Investig.* **15** 1–17.
- MALM, A. and BICHLER, G. (2011). Networks of collaborating criminals: Assessing the structural vulnerability of drug markets. *J. Res. Crime Delinq.* **48** 271–297.
- MEILÄ, M. (2007). Comparing clusterings—an information based distance. *J. Multivariate Anal.* **98** 873–895. [MR2325412](#) <https://doi.org/10.1016/j.jmva.2006.11.013>
- MILLER, J. W. and HARRISON, M. T. (2014). Inconsistency of Pitman-Yor process mixtures for the number of components. *J. Mach. Learn. Res.* **15** 3333–3370. [MR3277163](#)
- MILLER, J. W. and HARRISON, M. T. (2018). Mixture models with a prior on the number of components. *J. Amer. Statist. Assoc.* **113** 340–356. [MR3803469](#) <https://doi.org/10.1080/01621459.2016.1255636>
- MORSELLI, C. (2009). Hells Angels in springtime. *Trends Organ. Crime* **12** 145–158.
- MORSELLI, C., GIGUÈRE, C. and PETIT, K. (2007). The efficiency/security trade-off in criminal networks. *Soc. Netw.* **29** 143–153.
- MÜLLER, P., QUINTANA, F. and ROSNER, G. L. (2011). A product partition model with regression on covariates. *J. Comput. Graph. Statist.* **20** 260–278. [MR2816548](#) <https://doi.org/10.1198/jcgs.2011.09066>
- NEWMAN, M. E. J. and GIRVAN, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E* **69** 026113.
- NEWMAN, M. E. J. (2006). Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **103** 8577–8582.
- NEWMAN, M. E. J. and CLAUSET, A. (2016). Structure and inference in annotated networks. *Nat. Commun.* **7** 1–11.
- NOROOZI, M. and PENSKY, M. (2020). Statistical inference in heterogeneous block model. Preprint. Available at [arXiv:2002.02610](https://arxiv.org/abs/2002.02610).
- NOWICKI, K. and SNIJders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *J. Amer. Statist. Assoc.* **96** 1077–1087. [MR1947255](#) <https://doi.org/10.1198/016214501753208735>
- OLHEDE, S. C. and WOLFE, P. J. (2014). Network histograms and universality of blockmodel approximation. *Proc. Natl. Acad. Sci. USA* **111** 14722–14727.
- PAJOR, A. (2017). Estimating the marginal likelihood using the arithmetic mean identity. *Bayesian Anal.* **12** 261–287. [MR3620130](#) <https://doi.org/10.1214/16-BA1001>
- PAOLI, L. (2007). Mafia and organised crime in Italy: The unacknowledged successes of law enforcement. *West Eur. Polit.* **30** 854–880.
- PARK, J.-H. and DUNSON, D. B. (2010). Bayesian generalized product partition model. *Statist. Sinica* **20** 1203–1226. [MR2730180](#)
- QUINTANA, F. A. and IGLESIAS, P. L. (2003). Bayesian clustering and product partition models. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **65** 557–574. [MR1983764](#) <https://doi.org/10.1111/1467-9868.00402>
- RAFTERY, A. E., NEWTON, M. A., SATAGOPAN, J. M. and KRIVITSKY, P. N. (2007). Estimating the integrated likelihood via posterior simulation using the harmonic mean identity. In *Bayesian Statistics 8*. 1–45. Oxford Univ. Press, Oxford. [MR2433201](#)
- RANCIATI, S., VINCIOTTI, V. and WIT, E. C. (2020). Identifying overlapping terrorist cells from the Noordin Top actor-event network. *Ann. Appl. Stat.* **14** 1516–1534. [MR4152144](#) <https://doi.org/10.1214/20-AOAS1358>
- RASTELLI, R., LATOUCHE, P. and FRIEL, N. (2018). Choosing the number of groups in a latent stochastic blockmodel for dynamic networks. *Netw. Sci.* **6** 469–493.
- ROHE, K., CHATTERJEE, S. and YU, B. (2011). Spectral clustering and the high-dimensional stochastic block-model. *Ann. Statist.* **39** 1878–1915. [MR2893856](#) <https://doi.org/10.1214/11-AOS887>
- SALDAÑA, D. F., YU, Y. and FENG, Y. (2017). How many communities are there? *J. Comput. Graph. Statist.* **26** 171–181. [MR3610418](#) <https://doi.org/10.1080/10618600.2015.1096790>

- SANGKARAN, T., ABDULLAH, A. and JHANJI, N. (2020). Criminal community detection based on isomorphic subgraph analytics. *Open Comput. Sci.* **10** 164–174.
- SARKAR, P. and BICKEL, P. J. (2015). Role of normalization in spectral clustering for stochastic blockmodels. *Ann. Statist.* **43** 962–990. MR3346694 <https://doi.org/10.1214/14-AOS1285>
- SCHMIDT, M. N. and MORUP, M. (2013). Nonparametric Bayesian modeling of complex networks: An introduction. *IEEE Signal Process. Mag.* **30** 110–128.
- SENGUPTA, S. and CHEN, Y. (2018). A block model for node popularity in networks with community structure. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **80** 365–386. MR3763696 <https://doi.org/10.1111/rssb.12245>
- SPIEGELHALTER, D. J., BEST, N. G., CARLIN, B. P. and VAN DER LINDE, A. (2002). Bayesian measures of model complexity and fit. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **64** 583–639. MR1979380 <https://doi.org/10.1111/1467-9868.00353>
- STANLEY, N., BONACCI, T., KWITT, R., NIETHAMMER, M. and MUCHA, P. J. (2019). Stochastic block models with multiple continuous attributes. *Appl. Netw. Sci.* **4** 1–22.
- SUSSMAN, D. L., TANG, M., FISHKIND, D. and PRIEBE, C. E. (2012). A consistent adjacency spectral embedding for stochastic blockmodel graphs. *J. Amer. Statist. Assoc.* **107** 1119–1128. MR3010899 <https://doi.org/10.1080/01621459.2012.699795>
- TALLBERG, C. (2004). A Bayesian approach to modeling stochastic blockstructures with covariates. *J. Math. Sociol.* **29** 1–23.
- VON LUXBURG, U. (2007). A tutorial on spectral clustering. *Stat. Comput.* **17** 395–416. MR2409803 <https://doi.org/10.1007/s11222-007-9033-z>
- WADE, S. and GHAHRAMANI, Z. (2018). Bayesian cluster analysis: Point estimation and credible balls (with discussion). *Bayesian Anal.* **13** 559–626. MR3807860 <https://doi.org/10.1214/17-BA1073>
- WANG, Y. X. R. and BICKEL, P. J. (2017). Likelihood-based model selection for stochastic block models. *Ann. Statist.* **45** 500–528. MR3650391 <https://doi.org/10.1214/16-AOS1457>
- WANG, Y.-B., CHEN, M.-H., KUO, L. and LEWIS, P. O. (2018). A new Monte Carlo method for estimating marginal likelihoods. *Bayesian Anal.* **13** 311–333. MR3780425 <https://doi.org/10.1214/17-BA1049>
- WATANABE, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *J. Mach. Learn. Res.* **11** 3571–3594. MR2756194
- WATANABE, S. (2013). A widely applicable Bayesian information criterion. *J. Mach. Learn. Res.* **14** 867–897. MR3049492
- WHITE, A. and MURPHY, T. B. (2016). Mixed-membership of experts stochastic blockmodel. *Netw. Sci.* **4** 48–80.
- XU, Z., KE, Y., WANG, Y., CHENG, H. and CHENG, J. (2012). A model-based approach to attributed graph clustering. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data* 505–516.
- YANG, J., MCAULEY, J. and LESKOVEC, J. (2013). Community detection in networks with node attributes. In *2013 IEEE 13th International Conference on Data Mining* 1151–1156.
- ZHANG, Y., LEVINA, E. and ZHU, J. (2016). Community detection in networks with node features. *Electron. J. Stat.* **10** 3153–3178. MR3571965 <https://doi.org/10.1214/16-EJS1206>
- ZHAO, H., DU, L. and BUNTINE, W. (2017). Leveraging node attributes for incomplete relational data. In *International Conference on Machine Learning* 4072–4081.
- ZHAO, Y., LEVINA, E. and ZHU, J. (2012). Consistency of community detection in networks under degree corrected stochastic block models. *Ann. Statist.* **40** 2266–2292. MR3059083 <https://doi.org/10.1214/12-AOS1036>
- ZHOU, Z. and AMINI, A. (2019). Analysis of spectral clustering algorithms for community detection: The general bipartite setting. *J. Mach. Learn. Res.* **20** 1–47. MR3948087

PAPER

Zero–Inflated Stochastic Block Modeling of Efficiency–Security Tradeoffs in Weighted Criminal Networks

Chaoyi Lu,¹ Daniele Durante*² and Nial Friel¹

¹School of Mathematics and Statistics, University College Dublin, Ireland and ²Department of Decision Sciences, Bocconi University, Italy
*Corresponding author. daniele.durante@unibocconi.it

Abstract

Criminal networks arise from the unique attempt to balance a need of establishing frequent ties among affiliates to facilitate the coordination of illegal activities, with the necessity to sparsify the overall connectivity architecture to hide from law enforcement. This efficiency–security tradeoff is also combined with the creation of groups of redundant criminals that exhibit similar connectivity patterns, thus guaranteeing resilient network architectures. State-of-the-art models for such data are not designed to infer these unique structures. In contrast to these solutions we develop a computationally-tractable Bayesian zero-inflated Poisson stochastic block model (ZIP-SBM), which identifies groups of redundant criminals with similar connectivity patterns, and infers both overt and covert block interactions within and across such groups. This is accomplished by modeling weighted ties (corresponding to counts of interactions among pairs of criminals) via zero-inflated Poisson distributions with block-specific parameters that quantify complex patterns in the excess of zero ties in each block (security) relative to the distribution of the observed weighted ties within that block (efficiency). The performance of ZIP-SBM is illustrated in simulations and in a study of summits co-attendances in a complex Mafia organization, where we unveil efficiency–security structures adopted by the criminal organization that were hidden to previous analyses.

Key words: Collapsed MCMC, Crime, Gneden process, Stochastic block model, Zero–inflated Poisson

¹ 1. Introduction

The “EU Serious and Organised Crime Threat Assessment” report recently released by Europol in 2021 defines modern criminal networks as complex systems of interactions with incomplete information and a multifaceted combination of hierarchical overt and covert architectures. The challenges posed by these data incompletenesses and complexities undermine the effectiveness of law–enforcement policies, and present obstacles toward expanding knowledge on efficiency–security tradeoffs (Morselli et al., 2007) of criminal organizations from the analysis of the interactions among the suspects observed during investigations (Lindquist and Zenou, 2019; Faust and Tita, 2019; Campana and Varese, 2022; Diviák, 2022). Although substantial advancements have been made over the past thirty years in addressing these goals (e.g., Sparrow, 1991; Klerks, 2001; Von Lampe, 2006; Morselli, 2009; Papachristos, 2014; Calderoni et al., 2017; Bright et al., 2021), as organized crime evolves towards more nuanced structures, the ability of current solutions to infer more complex architectures is undermined by a general reliance in criminology on overly–simplified representations and an overarching focus on descriptive analyses.

Recalling state–of–the–art reviews in the field (e.g., Campana, 2016; Lindquist and Zenou, 2019; Faust and Tita, 2019; Campana and Varese, 2022; Diviák, 2022), there are at least two fundamental challenges which hinder advancements in the analysis of modern criminal networks. First, the covert nature of criminal organizations implies an excess of zero ties in the observed criminal network, which yield a partial view of the actual connectivity architecture underlying the illicit organization. Second, these zero ties are not randomly located, but often arise as the result of specific, yet unknown, secrecy and redundancy strategies to address efficiency–security tradeoffs (Morselli et al., 2007; Catino, 2015; Bouchard and Malm, 2016; Cavallaro et al., 2020). As a consequence, these strategies lead to unique group structures formed by redundant criminals, along with complex block–interactions among such groups which incorporate a combination of dense community patterns, core–periphery structures and weakly assortative modules in both the overt and covert connectivity architectures. This means that, if properly

modeled, every block has the potential to disentangle both efficiency and security structures by studying the excess of zero ties within such a block (security) relative to the distribution of the observed weighted interactions among criminals belonging to the two groups which identify the block (efficiency).

This article is motivated by the above intuition and aims at translating the aforementioned challenges into opportunities for inferring more nuanced and yet-unaddressed efficiency and security architectures of criminal organizations rooted within the complex interactions among the corresponding members. This is accomplished through the development of a new Bayesian zero-inflated Poisson stochastic block model (ZIP-SBM) for weighted, yet sparse, criminal networks which combines (i) species sampling processes (e.g., De Blasi et al., 2015; Gnedin, 2010) to rigorously characterize the mechanisms of redundant groups formation and criminals affiliation to these underlying groups, (ii) stochastic block models (e.g., Holland et al., 1983; Nowicki and Snijders, 2001) to define the observed network as a function of the criminals' allocations to the different groups and flexible block interactions among such groups and, finally, (iii) zero-inflated Poisson distributions (Lambert, 1992; Ghosh et al., 2006) to infer *unusual* excess of zeros in the distribution of the weighted ties within the blocks defining the different pairs of groups. Such a model crucially exploits structural and regular equivalence patterns inherent to known endogenous and exogenous redundancies in organized crime (Sparrow, 1991) to express the observed network as a combination of two underlying ones. The first reconstructs, for each pair of groups, the actual strength of the ties among the allocated criminals (efficiency), whereas the second unveils the propensity of the illicit organization to either obscure or not these ties within the block corresponding to that pair of groups (security). Section 1.1 clarifies the methodological and applied advancements of the proposed approach.

1.1. Relevant literature

Although there has been a recent adoption of sophisticated statistical methods to study criminal networks (e.g., Malm et al., 2017; Charette and Papachristos, 2017; Calderoni et al., 2017; Bright et al., 2019; Diviák et al., 2019; Gollini et al., 2020; Cavallaro et al., 2020; Legramanti et al., 2022), none of the currently-available solutions provides a generative model that can flexibly incorporate, and infer, core structures of illicit organizations not only in the strength of the observed ties among criminals, but also in systematic sparsity patterns, to ultimately unveil the nature of the efficiency–security tradeoffs. In fact, popular link–prediction methods mainly rely on descriptive solutions applied to a dichotomized version of the observed weighted criminal networks (Berlusconi et al., 2016; Calderoni et al., 2020). These strategies lack a model–based perspective that would enable inference, uncertainty quantification and inclusion of those measurement errors which may occur in investigations. Moreover, the loss of information arising from dichotomization crucially fails to infer efficiency structures encoded in the strength of the weighted ties — often corresponding to counts of interactions among pairs of criminals — and hinders the possibility to unveil more nuanced security architectures from the analysis of *unusual* patterns of zero connections relative to the distribution of weighted ties.

A noteworthy attempt to move towards more structured model–based representations of the complex group interactions in modern criminal networks can be found in the extended stochastic block model of Legramanti et al. (2022). Albeit providing important advancements in inference on redundancy patterns relative to previous studies relying on community detection algorithms (Girvan and Newman, 2002; Newman, 2006; Blondel et al., 2008) and spectral clustering methods (Von Luxburg, 2007), this perspective still focuses on dichotomized versions of weighted criminal networks. Hence, it faces the same conceptual barriers of link–prediction methods when the goal is to disentangle and quantify security architectures from efficiency structures. In fact, in order to infer an excess of zero ties pointing towards systematic obscuration mechanisms it is necessary to possess a benchmark distribution for the weighted interactions which allows one to quantify the extent to which the total number of observed zero connections is *unusual* relative to those expected under such a distribution. As clarified in Section 2, the proposed ZIP-SBM addresses these challenges via a novel generalization of extended stochastic block models (Legramanti et al., 2022) in the context of weighted and sparse criminal networks where the focus is to quantify efficiency–security tradeoff architectures. This is accomplished by avoiding data dichotomization prior to statistical modeling while relying on block-specific zero-inflated Poisson distributions — rather than Bernoulli ones — for the ties among groups of redundant criminals.

Although the potentials of related constructions have been never explored in the context of criminal network analysis, from a methodological perspective there has been some research involving stochastic block models in combination with zero-inflations (Aicher et al., 2015; Mariadassou and Matias, 2015; Ng and Murphy, 2021). The proposed ZIP-SBM yields key advancements relative to these and other contributions. In fact, these formulations address the simpler, and very different, settings which either consider all zero ties to denote a truly observed non-existing interaction or assume knowledge about which zero ties are associated to a truly non-existing interaction

and which simply denote the lack of knowledge about the value of such an interaction. This information is not available in commonly-analyzed criminal networks. In fact, the proposed ZIP-SBM aims to uncover this type of information rather than presuming it. In addition, the above procedures rely on multinomial distributions for the group allocations which imply that the total number of modules underlying the criminal network is finite and pre-specified. In fact, such a quantity is unknown in criminal network studies and, hence, it is conceptually and practically useful to incorporate uncertainty also on the total number of groups and learn it as part of the inference process. As detailed in Section 2.2, the ZIP-SBM addresses this aspect via a Gneden process prior (Gneden, 2010) for the allocation of redundant criminals to groups (Legramanti et al., 2022). Besides incorporating uncertainty in the total number of modules within the network, this prior also yields a natural characterization for the affiliation mechanism of criminals to redundant groups via a scheme that depends both on the size of the criminal network and also on the current number, dimension and composition of such groups.

From a more general perspective, the proposed ZIP-SBM has also connections with a relevant line of research that considers the observed network as a corrupted measurement of a “true” underlying one, and aims at recovering such an underlying network, while providing inference on its structures; see e.g., Chatterjee (2015), Priebe et al. (2015), Young et al. (2020) and the references therein. Recasting our proposal within this framework, the “true” underlying network could be regarded as the actual count of interactions among pairs of criminals (efficiency), for which we observe a corrupted version due to the zero inflation (possibly operated by a security strategy). However, unlike Chatterjee (2015) and Young et al. (2020), ZIP-SBM crucially includes and learns block structures among criminals both in the “true” underlying network and also in the mechanism yielding the corrupted measurements, while allowing the model parameters to change across such blocks. These block structures are inherent to criminal networks and are also of key interest in criminology, thus motivating a focus on those models that can incorporate these structures. To this end, the contribution by Priebe et al. (2015) is more in line with the scope of ZIP-SBM, in that it also accounts for group structures among nodes in networks observed with errors. However, Priebe et al. (2015) focus on the simpler node classification problem in which the goal is to identify the group membership of a single distinguished node, assuming that those of all the others are already known. This is not the case in criminal networks, where the group allocation defining redundancy patterns is unknown for all the criminals and must be inferred from a network with contamination of zeros.

Albeit providing a sophisticated representation of criminal networks, the proposed ZIP-SBM is amenable to tractable posterior inference for the allocation of criminals to groups and for the parameters of the zero-inflated Poisson distribution within each block. This is accomplished via a data-augmentation collapsed Gibbs-sampler derived in Section 3. The output of this algorithm yields highly accurate reconstructions of redundant patterns and efficiency-security architectures both in extensive simulation analyses and also in a study of the complex Mafia network reconstructed from the judicial records of “Operazione Infinito” (e.g., Calderoni et al., 2017; Legramanti et al., 2022); see Section 1.2 for details about the motivating application. These analyses are illustrated within Sections 4–5, respectively, and clarify the potential of a ZIP-SBM in uncovering relevant structures that state-of-the-art methods cannot unveil. For instance, unlike the analyses of the *Infinito* network in Calderoni et al. (2017) and Legramanti et al. (2022), the proposed ZIP-SBM unveils a yet-discovered combination of communities and core-periphery architectures, in both overt and covert connectivity patterns. While moving from the periphery to the core, such structures are characterized by a reduced redundancy combined with a tendency of establishing increasingly strong ties and a preference to progressively sparsify these connections to guarantee security of those criminals at the top of the pyramid. We refer the reader to Section 6 for a final discussion and future directions.

1.2. The 'Ndrangheta network from “Operazione Infinito”

This article is motivated by an attempt to unveil unexplored knowledge on the efficiency and security structures of the 'Ndrangheta Mafia operating in the area of Milan (north of Italy) from the analysis of a network among its members that have been monitored during a law-enforcement operation named “Operazione Infinito” (Calderoni et al., 2017; Legramanti et al., 2022); see Figure 1. According to Interpol, 'Ndrangheta is currently one of the most widespread, proliferated and powerful criminal organizations worldwide, with a strong tendency towards transnational crime and a specific ability to generate highly structured, pervasive and resilient networks that can penetrate not only illegal activities, but also regular business and politics (Paoli, 2007; Catino, 2014; Sergi and Lavorgna, 2016). The alarming threat and the unique challenges posed by such an organization are further evident from the recent establishment of the I-CAN project¹, a three year (2020–2023) initiative whose aim is to favor across-country cooperation and coordination in a multilateral response against 'Ndrangheta.

¹ <https://www.interpol.int/Crimes/Organized-crime/INTERPOL-Cooperation-Against-Ndrangheta-I-CAN>.

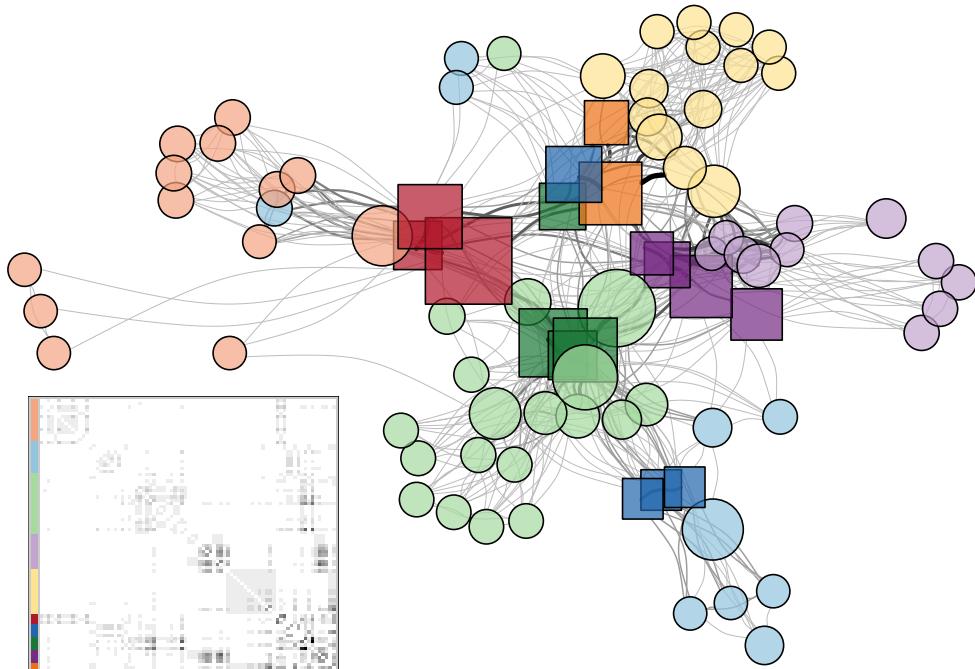


Fig. 1. Representation of the *Infinito* network (e.g., Calderoni et al., 2017) and its adjacency matrix. Each node is a criminal and ties denote the number of co-attended summits of the 'Ndrangheta organization, as monitored during investigations. Node positions are obtained via force directed placement (Fruchterman and Reingold, 1991), whereas colors, both for the network and for its adjacency matrix representation, define the presumed *locale* membership. Darker square nodes indicate the suspected bosses of each *locale*, while lighter circles represent simple affiliates. Node size is proportional to the corresponding betweenness, while tie color and thickness is proportional to its weight.

Quoting the Interpol Secretary General, Jürgen Stock: “*I-CAN is about building a global early warning system against an invisible enemy*”. This description clarifies the importance of obtaining improved knowledge on the source structures of 'Ndrangheta, and crucially highlights the core challenge in addressing this goal, namely the security architecture underlying such a covert organization. As clarified within Section 1, our goal is to cover this fundamental gap via a careful model-based approach capable of learning structure also in *invisible* connectivity patterns, with a focus on the analysis of the 'Ndrangheta *Infinito* network retrieved from the judicial documents of “Operazione *Infinito*”².

Recalling Calderoni et al. (2017) and Legramanti et al. (2022) “Operazione *Infinito*” has been a massive law-enforcement operation spanning six years with the aim of monitoring and disrupting the core architecture of *La Lombardia*, the highly-pervasive branch of the 'Ndrangheta Mafia operating in the area of Milan. As shown in Figure 1, each node is a member of the organization, while the ties denote the number of summits (or meetings) of the criminal organization co-attended by each pair of nodes, as monitored throughout the investigations. The original data are available at <https://sites.google.com/site/ucinetsoftware/datasets/covert-networks> and have been retrieved from the pre-trial detention order that was issued upon a request by the prosecution. To clarify the substantial advancements in inference on efficiency and security structures resulting from the proposed ZIP-SBM relative to state-of-the-art studies of such data, we consider here the pre-processed network analyzed in Legramanti et al. (2022), but with a crucial difference. While Legramanti et al. (2022) focus on dichotomized ties encoding presence or absence of at least one co-attendance, we crucially avoid such a dichotomization and leverage both sparsity and information on weighted ties to substantially expand the knowledge about 'Ndrangheta architectures and security strategies. In fact, in addition to providing a graphical evidence of grouping structures among redundant criminals, Figure 1 also suggests the presence of systematic patterns both in the strength of weighted ties and in the positioning of the zero interactions. In particular, while moving from the periphery to the core, there seems to be a tendency to increasingly sparsify ties, while strengthening the non-zero ones. The

² Tribunale di Milano, 2011. Ordinanza di applicazione di misura coercitiva con mandato di cattura — art. 292 c.p.p. (Operazione *Infinito*). Ufficio del giudice per le indagini preliminari (in Italian)

152 dichotomization operated by Legramanti et al. (2022) and the community detection approach in Calderoni et al.
153 (2017) rule out these seemingly fundamental architectures, whereas, as clarified in Section 5, ZIP–SBM exploits
154 such additional sources of information to extract deeper knowledge about the organization.

155 **Remark 1** Although the newly–proposed ZIP–SBM is general and can be applied to networks measuring any
156 form of count interaction among criminals, our focus on summit co–attendances is motivated by two main reasons.
157 Firstly, the judicial documents of “Operazione *Infinito*” provide a highly detailed report of criminal attendances to
158 monitored ’Ndrangheta summits. Secondly, following Calderoni et al. (2017) and Calderoni and Superchi (2019),
159 the co–attendance patterns to summits are often more informative about the underlying structure and function
160 of the networked system among the members of the criminal organization than other forms of interactions, such
161 as, e.g., phone calls. This motivates our focus on the one–mode criminal–criminal projection of the original two–
162 mode criminal–summit data. Such a projection (i) is common in related studies (Calderoni et al., 2017; Calderoni
163 and Superchi, 2019; Cavallaro et al., 2020; Ficara et al., 2021; Legramanti et al., 2022), (ii) facilitates comparison
164 with the previous analyses of the same one–mode *Infinito* network (e.g., Calderoni et al., 2017; Legramanti et al.,
165 2022) and (iii) provides access to weighted ties useful for disentangling efficiency–security structures.

166 **Remark 2** As a consequence of Remark 1, a zero tie among two generic criminals in the network implies that
167 these criminals were never recorded, during the investigations, to attend a same summit. Such an event might be
168 either due to the actual absence of ties among these two criminals or to the fact that possible interactions have
169 occurred through more secure mechanisms, including additional, yet secret, summits, hidden to law–enforcement
170 investigations. The ZIP–SBM aims to disentangle these two alternatives and quantify the strength of interaction
171 behind obscured ties. These advances are a key to (i) improve network reconstruction, (ii) uncover the efficiency–
172 security tradeoffs of the criminal organization analyzed, and (iii) guide the investigations towards monitoring the
173 inferred obscured ties. Notice that, although the law–enforcement may fail to monitor specific summits, the data
174 analyzed are the result of careful investigations spanning across six years, and the arrest warrant documents that
175 produced the network in Figure 1 contain a comprehensive report of summit attendances. If the zero ties were only
176 due to a general inability of law–enforcement to monitor ’Ndrangheta summits, one would expect no systematic
177 security patterns. In fact, as discussed within Section 5, the proposed ZIP–SBM uncovers structured obscuration
178 mechanisms related to peculiar hierarchies within the criminal organization which are, therefore, highly unlikely
179 to be simply the result of limited investigations.

180 Figure 1 also suggests that the block–connectivity patterns among redundant groups might be related to the
181 available criminal attributes encoding presumed *locale* affiliations and roles. Such an empirical finding is in line
182 with forensic theories which suggest that the ’Ndrangheta organization revolves around blood family relations,
183 often aggregated at the territorial level in structural units, known as *locali* (Paoli, 2007; Catino, 2014; Sergi and
184 Lavorgna, 2016). These units administer crime in different territories and are characterized by an additional level
185 of internal hierarchy comprising a set of affiliates and comparatively fewer bosses that oversee the illicit and licit
186 activities within each *locale*, and coordinate interactions across *locali*. Due to this, it is reasonable to expect that
187 ’Ndrangheta organizations are subject to a preference toward creating redundancies within *locali*, rather than
188 across such territorial units, and, at a more nested level, with respect to similarities in the role. Nonetheless, since
189 these attributes are also a result of error–prone investigations, the inclusion of this general notion of *homophily*
190 should be treated with care. As clarified in Section 3, under the proposed ZIP–SBM this is accomplished via a
191 probabilistic reinforcement of the prior on criminal allocations to groups which favors the creation of modules
192 that are homogenous with respect to the external attributes, but does not prevent from inferring more nuanced
193 group structures comprising heterogenous criminals. In fact, as illustrated in Section 5, a ZIP–SBM unveils also
194 highly peculiar modules that depart from the *locale* and role division, while pointing toward potential instabilities
195 and obscured dynamics within the ’Ndrangheta organization.

196 2. Bayesian Zero–Inflated Stochastic Block Models

197 Let \mathbf{Y} be the $V \times V$ symmetric adjacency matrix comprising the weighted, yet sparse, undirected ties among the
198 V criminals (nodes) in the network. More specifically, each $y_{vu} = y_{uv} \in \mathbb{N}$ measures the count relation between
199 nodes v and u , for every $v = 2, \dots, V$ and $u = 1, \dots, v - 1$. Notice that, since our focus is on undirected networks,
200 y_{vu} is equal to y_{uv} by definition. Following the discussion in Section 1, we aim to define a statistical model for
201 \mathbf{Y} which incorporates, and infers, (i) underlying grouping structures defined by redundant nodes having similar
202 connectivity patterns, (ii) flexible overt and covert block interactions among these groups, and (iii) structured
203 sparsity to disentangle efficiency and security architectures.

204 Equations (1)–(3) summarize the newly–proposed ZIP–SBM, whose interpretation, properties and generative
 205 construction are discussed in Sections 2.1–2.2. More specifically, denote by $\mathbf{z} = (z_1, \dots, z_V)$ the vector of criminal
 206 allocations to redundant groups, where $z_v = h$ indicates that the v –th criminal belongs to group $h \in \{1, \dots, H\}$,
 207 for each $v = 1, \dots, V$. Moreover, let $\bar{\Pi}$ and $\bar{\Lambda}$ be two $H \times H$ symmetric matrices whose entries $\bar{\pi}_{hk} \in (0, 1)$ and
 208 $\bar{\lambda}_{hk} \in \mathbb{R}^+$ correspond to the zero–inflation probability (security) and the rate (efficiency), respectively, of the ties
 209 among generic criminals in groups h and k , for each $h = 1, \dots, H$ and $k = 1, \dots, h$. Then, the proposed Bayesian
 210 ZIP–SBM is defined as

$$(y_{vu} \mid z_v = h, z_u = k, \bar{\pi}_{hk}, \bar{\lambda}_{hk}) \sim \text{ZIP}(\bar{\pi}_{hk}, \bar{\lambda}_{hk}), \quad \text{independently for } v = 2, \dots, V, u = 1, \dots, v-1, \quad (1)$$

$$\bar{\pi}_{hk} \sim \text{Beta}(a, b), \quad \bar{\lambda}_{hk} \sim \text{Gamma}(a_1, a_2), \quad \text{independently for } h = 1, \dots, H, k = 1, \dots, h, \quad (2)$$

$$(\mathbf{z} = (z_1, \dots, z_V) \mid \mathbf{c}) \sim \text{GN}(\gamma; \mathbf{c}), \quad (3)$$

211 where $\text{ZIP}(\bar{\pi}_{hk}, \bar{\lambda}_{hk})$ in (1) denotes the zero–inflated Poisson distribution (e.g., Lambert, 1992; Ghosh et al., 2006)
 212 with probability mass function $p(y) = \bar{\pi}_{hk}\mathbb{1}(y = 0) + (1 - \bar{\pi}_{hk})\bar{\lambda}_{hk}^y e^{-\bar{\lambda}_{hk}}/y!$, ($y \in \mathbb{N}$), for the ties among pairs
 213 of criminals in groups h and k , respectively, whereas equations (2)–(3) clarify the selected prior distributions for
 214 the block–specific parameters in $\bar{\Pi}$ and $\bar{\Lambda}$, and for the allocation vector \mathbf{z} encoding membership of criminals to
 215 the underlying groups. In particular, in equation (2) we follow the recommended practice in Bayesian stochastic
 216 block models for binary (e.g., Nowicki and Snijders, 2001; Geng et al., 2019) and weighted (e.g., McDaid et al.,
 217 2013) networks, and assume independent Beta(a, b) and Gamma(a_1, a_2) priors for the entries $\bar{\pi}_{hk}$ in $\bar{\Pi}$, and $\bar{\lambda}_{hk}$
 218 in $\bar{\Lambda}$, respectively. As discussed in Section 2.1 this choice guarantees conjugacy properties that facilitate posterior
 219 inference. In addition, it introduces suitable dependence structures among the ties, both overt and covert, between
 220 criminals in the network. For the allocation vector we adopt in (3) a supervised version $\text{GN}(\gamma; \mathbf{c})$ of the Gneden
 221 process prior (Gneden, 2010) which belongs to the general Gibbs–type class (e.g., De Blasi et al., 2015) employed
 222 in Legramanti et al. (2022). As clarified in Section 2.2, this prior naturally characterizes the affiliation mechanism
 223 of criminals to the redundant groups via a sequential scheme, which depends both on the network size and also
 224 on the current number, dimension and, possibly, the attribute composition \mathbf{c} of such groups. In the study of the
 225 *Infinito* network in Figure 1, the external attribute vector $\mathbf{c} = (c_1, \dots, c_V) \in \{1, \dots, C\}^V$ encodes memberships of
 226 criminals to a known exogenous partition that corresponds to a combination of *locale*–role information and, hence,
 227 is expected to influence the formation of the redundant endogenous groups encoded in \mathbf{z} . Notice that, as discussed
 228 in Section 2.2, when \mathbf{c} is not available, the ZIP–SBM can be still implemented by replacing the supervised Gneden
 229 process prior $\text{GN}(\gamma; \mathbf{c})$ with its unsupervised counterpart $\text{GN}(\gamma)$.

230 As outlined within Section 1.1, the ZIP–SBM model in (1)–(3) generalizes the extended stochastic block model
 231 of Legramanti et al. (2022), originally developed for binary networks, in order to include weighted ties while
 232 accounting for possible zero–inflation mechanisms pointing to systematic sparsity patterns that may inform on
 233 security strategies. This is accomplished by replacing the commonly–assumed Bernoulli distribution within each
 234 block with a zero–inflated Poisson. Albeit providing a natural methodological extension, this choice substantially
 235 expands the potential for improved inference within the context of criminal networks and opens new avenues
 236 to infer yet–unexplored generative architectures produced by efficiency–security tradeoffs (Morselli et al., 2007;
 237 Catino, 2015). In fact, as discussed within Section 1.1, unlike other stochastic block models incorporating zero–
 238 inflation (Aicher et al., 2015; Mariadassou and Matias, 2015; Ng and Murphy, 2021), the proposed ZIP–SBM does
 239 not require knowledge about the nature of zero ties, but rather aims to uncover it as part of the inference process;
 240 see Section 2.1 for additional details, and refer to the empirical studies in Sections 4–5 for a clear illustration of
 241 the applied potential of the ZIP–SBM in (1)–(3) relative to state–of–the–art alternatives.

2.1. Generative construction, interpretation and properties

242 Adapting classical augmented–data representations of zero–inflated Poisson distributions (e.g., Lambert, 1992;
 243 Ghosh et al., 2006) to our network setting, model (1) can be readily obtained by marginalizing out the latent
 244 variables $w_{vu} \in \mathbb{N}$ and $x_{vu} \in \{0; 1\}$ in the following generative representation

$$\begin{aligned} y_{vu} &= w_{vu}(1 - x_{vu}), \\ (w_{vu} \mid z_v = h, z_u = k, \bar{\lambda}_{hk}) &\sim \text{Poisson}(\bar{\lambda}_{hk}), \quad (x_{vu} \mid z_v = h, z_u = k, \bar{\pi}_{hk}) \sim \text{Bern}(\bar{\pi}_{hk}), \end{aligned} \quad (4)$$

246 independently for $v = 2, \dots, V$ and $u = 1, \dots, v-1$. Combining (4) with (2)–(3) clarifies that the proposed ZIP–
 247 SBM can be alternatively reinterpreted as the combination of two underlying stochastic block models for the

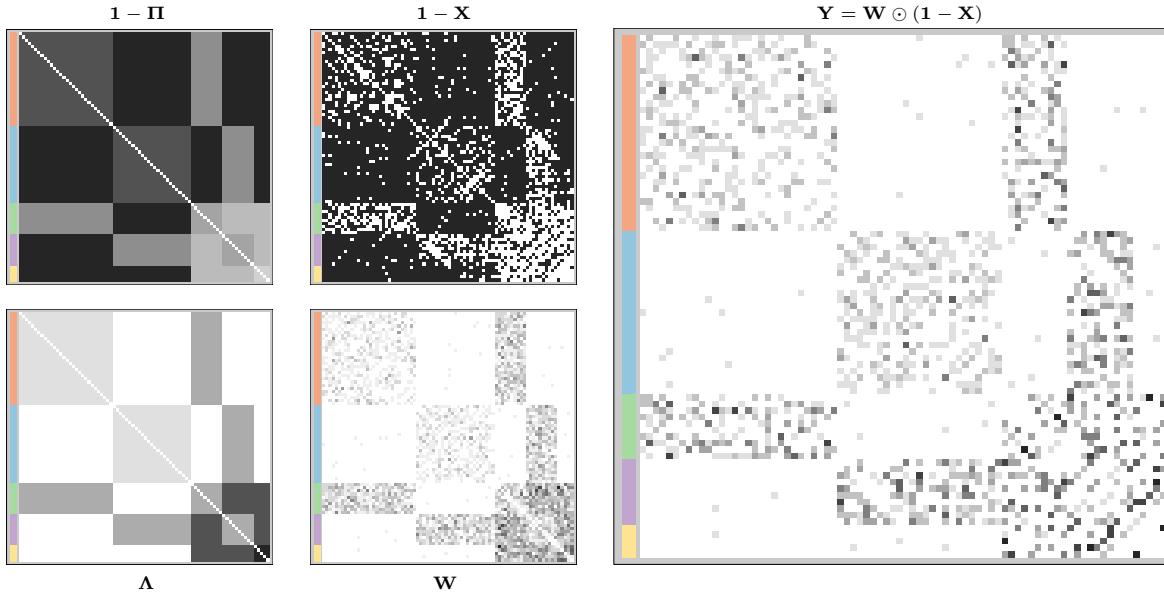


Fig. 2. Graphical representation of the generative mechanism underlying ZIP-SBM. Side colors correspond to the grouping structure encoded in \mathbf{z} . Criminals allocated to the same group are redundant from a connectivity perspective and, hence, the associated rows in $\mathbf{1} - \boldsymbol{\Pi} = \mathbf{1} - \mathbf{Z}\bar{\boldsymbol{\Pi}}\mathbf{Z}^\top$ and $\boldsymbol{\Lambda} = \mathbf{Z}\bar{\boldsymbol{\Lambda}}\mathbf{Z}^\top$ are equal by construction. Notice that \mathbf{Z} is the $V \times H$ matrix with rows $\mathbf{z}_v = [\mathbb{1}(z_v = 1), \dots, \mathbb{1}(z_v = H)]$ for $v = 1, \dots, V$. The matrices $\mathbf{1} - \mathbf{X}$ and \mathbf{W} have generic entries $1 - x_{vu}$ and w_{vu} , respectively, simulated as in (4), where $1 - \pi_{vu} = 1 - \mathbf{z}_v \bar{\boldsymbol{\Pi}} \mathbf{z}_u^\top$ and $\lambda_{vu} = \mathbf{z}_v \bar{\boldsymbol{\Lambda}} \mathbf{z}_u^\top$ correspond to the entries in position (v, u) of $\mathbf{1} - \boldsymbol{\Pi}$ and $\boldsymbol{\Lambda}$, respectively. Recalling (4), the element-wise Hadamard product \odot between \mathbf{W} and $\mathbf{1} - \mathbf{X}$ yields the observed network \mathbf{Y} .

augmented networks \mathbf{W} and \mathbf{X} with entries w_{vu} and x_{vu} , respectively, for each $v = 2, \dots, V$ and $u = 1, \dots, v - 1$. The former encodes the observed or latent weighted ties among each generic pair of criminals (v, u) , for every $v = 2, \dots, V$ and $u = 1, \dots, v - 1$, whereas the latter characterizes the excess of zeros relative to those expected under the distribution of such weighted ties, thus informing on possible security architectures.

As is clear from representation (4), the distribution of both quantities w_{vu} and x_{vu} only depends on the group allocations z_v and z_u of the two involved nodes v and u , and on the block-specific parameters defining the ties among such groups. As a consequence, these parameters are shared among all ties involving criminals from the same pair of groups (h, k) , and only change as a function of (h, k) . This construction crucially leverages, formalizes, and infers redundancy structures made by groups of criminals with a similar position within the network topology and, hence, redundant from a connectivity perspective (e.g., Bouchard and Malm, 2016; Cavallaro et al., 2020). To clarify this point, notice that as a consequence of (4), the individual parameters λ_{vu} and π_{vu} indexing the distribution of w_{vu} and x_{vu} , respectively, for each $v = 2, \dots, V$ and $u = 1, \dots, v - 1$, can be derived as

$$\lambda_{vu} = \bar{\lambda}_{z_v, z_u} = \mathbf{z}_v \bar{\boldsymbol{\Lambda}} \mathbf{z}_u^\top \quad \text{and} \quad \pi_{vu} = \bar{\pi}_{z_v, z_u} = \mathbf{z}_v \bar{\boldsymbol{\Pi}} \mathbf{z}_u^\top, \quad \text{for } v = 2, \dots, V, u = 1, \dots, v - 1, \quad (5)$$

where $\mathbf{z}_v = [\mathbb{1}(z_v = 1), \dots, \mathbb{1}(z_v = H)]$ and $\mathbf{z}_u = [\mathbb{1}(z_u = 1), \dots, \mathbb{1}(z_u = H)]$ comprise vectors of all zero entries except for a single 1 in the position corresponding to the group indicator to which the nodes v and u have been allocated, respectively. Let $\boldsymbol{\Lambda}$ and $\boldsymbol{\Pi}$ denote the $V \times V$ symmetric matrices with entries λ_{vu} and π_{vu} , respectively, for each $v = 2, \dots, V$ and $u = 1, \dots, v - 1$. Then the above result implies that the rows of $\boldsymbol{\Lambda}$ and $\boldsymbol{\Pi}$ corresponding to criminals allocated to the same redundant group are equal, respectively, meaning that any two criminals v and v' within the same group share the same rates of interaction and zero-inflation probabilities in the ties with others, i.e., $\lambda_{vu} = \lambda_{v'u}$ and $\pi_{vu} = \pi_{v'u}$ for every u different from v and v' . This enforces, in turn, equality in distribution among the associated rows of \mathbf{W} and \mathbf{X} , respectively, thereby incorporating the precise notion of stochastic equivalence (e.g., Nowicki and Snijders, 2001). Due to the definition of y_{vu} in (4), the same holds for the observed network \mathbf{Y} . See Figure 2 for a graphical illustration of the generative model underlying \mathbf{Y} , which further stresses these concepts. Such a figure also clarifies that the overall sparsity of the network is controlled by both the zero-inflation probabilities in $\boldsymbol{\Pi}$ and the Poisson rates in $\boldsymbol{\Lambda}$. High values of the entries in $\boldsymbol{\Pi}$ and/or low

272 rates in Λ yield sparser networks. As clarified in the simulation studies within Section 4, although the parameters
 273 associated with sparser blocks are more difficult to learn due to the presence of relatively few weighted ties, the
 274 proposed ZIP–SBM generally achieves accurate performance also in these settings.

275 Besides providing a simple augmented–data representation for the formation process of the ties in \mathbf{Y} which is
 276 useful for both estimation and inference, the above formulation characterizes also a natural and interpretable
 277 generative model for criminal networks. In particular, according to representation (4) and Figure 2, the observed
 278 tie $y_{vu} \in \mathbb{N}$ among criminals v and u , depends on the decision to either obscure or not ties among such criminals,
 279 encoded in $x_{vu} \in \{0; 1\}$, and on the corresponding, observed or latent, strength of interaction measured by $w_{vu} \in$
 280 \mathbb{N} . Following this line of reasoning, if $y_{vu} > 0$ then $(x_{vu} = 0, w_{vu} = y_{vu})$, meaning that no security strategy has
 281 been adopted for the pair (v, u) and the weighted tie among v and u has been actually observed. A zero count
 282 $y_{vu} = 0$ can be instead associated with two different scenarios, namely $(x_{vu} = 1, w_{vu} \in \mathbb{N})$ or $(x_{vu} = 0, w_{vu} = 0)$.
 283 In the first case, ties among v and u have been arguably obscured — no matter whether the strength of these ties
 284 is high or low. In the second situation, no security strategy has been adopted, but the actual count of interactions
 285 among v and u is effectively zero. Recalling (4), these two cases can be disentangled under the proposed ZIP–SBM
 286 via the conditional probability

$$\begin{aligned}\text{pr}(x_{vu} = 1 \mid y_{vu} = 0, z_v = h, z_u = k, \bar{\pi}_{hk}, \bar{\lambda}_{hk}) &= \frac{\text{pr}(x_{vu} = 1, y_{vu} = 0 \mid z_v = h, z_u = k, \bar{\pi}_{hk}, \bar{\lambda}_{hk})}{\text{pr}(y_{vu} = 0 \mid z_v = h, z_u = k, \bar{\pi}_{hk}, \bar{\lambda}_{hk})}, \\ &= \frac{\bar{\pi}_{hk}}{\bar{\pi}_{hk} + (1 - \bar{\pi}_{hk})e^{-\bar{\lambda}_{hk}}},\end{aligned}\quad (6)$$

287 for every $v = 2, \dots, V$ and $u = 1, \dots, v - 1$. When $y_{vu} = 0$, it is also of interest to assess to what extent such
 288 a zero tie corresponds to an underlying non–zero interaction. Recalling our previous discussion, when a security
 289 strategy is implemented, i.e., $x_{vu} = 1$, then $y_{vu} = 0$ no matter whether an actual underlying tie w_{vu} is effectively
 290 present, i.e., $w_{vu} > 0$, or not, namely $w_{vu} = 0$. Disentangling these two alternatives is a key in law enforcement
 291 to unveil the effective structure of the underlying criminal network and assess which obscured ties are hiding
 292 strong interactions that are worth investigations. Information on non–zero obscured ties can be obtained under
 293 the proposed model via the conditional probability

$$\begin{aligned}\text{pr}(w_{vu} > 0 \mid y_{vu} = 0, z_v = h, z_u = k, \bar{\pi}_{hk}, \bar{\lambda}_{hk}) &= \frac{\text{pr}(w_{vu} > 0, y_{vu} = 0 \mid z_v = h, z_u = k, \bar{\pi}_{hk}, \bar{\lambda}_{hk})}{\text{pr}(y_{vu} = 0 \mid z_v = h, z_u = k, \bar{\pi}_{hk}, \bar{\lambda}_{hk})}, \\ &= \frac{(1 - e^{-\bar{\lambda}_{hk}})\bar{\pi}_{hk}}{\bar{\pi}_{hk} + (1 - \bar{\pi}_{hk})e^{-\bar{\lambda}_{hk}}},\end{aligned}\quad (7)$$

294 for every $v = 2, \dots, V$ and $u = 1, \dots, v - 1$. Note that, a very large $\bar{\lambda}_{hk}$ implies $e^{-\bar{\lambda}_{hk}} \approx 0$, and, therefore, both (6)
 295 and (7) are almost equal and coincide approximately with $\bar{\pi}_{hk}/\bar{\pi}_{hk} = 1$. This provides a reasonable result stating
 296 that, if $\bar{\lambda}_{hk}$ is very large, any observed zero tie is almost surely due to a secrecy strategy (i.e., $\text{pr}(x_{vu} = 1 \mid y_{vu} =$
 297 $0, z_v = h, z_u = k, \bar{\pi}_{hk}, \bar{\lambda}_{hk}) \approx 1$) hiding a non–zero interaction ($\text{pr}(w_{vu} > 0 \mid y_{vu} = 0, z_v = h, z_u = k, \bar{\pi}_{hk}, \bar{\lambda}_{hk}) \approx$
 298 1). This is because the probability of a zero tie under a Poisson with a very large rate is essentially 0 and, hence,
 299 any observed zero tie is almost surely the result of a secrecy strategy arising from the zero–inflation mechanism.
 300 In this context, almost all the zero ties can be used to effectively infer $\bar{\pi}_{hk}$.

301 Removing the conditioning on $y_{vu} = 0$ in (7), provides instead a measure of efficiency which quantifies the
 302 probability that v and u establish a non–zero tie, irrespectively of whether it has been obscured or not. This yields

$$\text{pr}(w_{vu} > 0 \mid z_v = h, z_u = k, \bar{\lambda}_{hk}) = 1 - e^{-\bar{\lambda}_{hk}}, \quad (8)$$

303 for each $v = 2, \dots, V$ and $u = 1, \dots, v - 1$.

304 As detailed in (6), evidence of security structures behind a zero tie $y_{vu} = 0$ among two generic criminals v and
 305 u allocated to groups h and k , respectively, does not necessarily require a high $\bar{\pi}_{hk}$. In fact, even a low $\bar{\pi}_{hk}$ can
 306 point toward an *unusual* zero tie, when $\bar{\lambda}_{hk}$ is large, since such a zero will appear as highly unlikely under the
 307 Poisson($\bar{\lambda}_{hk}$) distribution for the weighted ties. As a consequence, it is fundamental to accurately infer $\bar{\pi}_{hk}$ and
 308 $\bar{\lambda}_{hk}$, for each $h = 1, \dots, H$ and $k = 1, \dots, h$, from the observed data y_{vu} , $v = 2, \dots, V$ and $u = 1, \dots, v - 1$ in
 309 \mathbf{Y} . This is also a key to quantify evidence of efficiency via (8). As clarified in (1)–(3), the proposed ZIP–SBM
 310 achieves this goal by expressing the network \mathbf{Y} as a function of criminal allocations to underlying groups and a
 311 lower number of block–specific parameters encoding efficiency–security patterns within and across these groups.

This means that all the ties connecting criminals in group h with those in group k are conditionally independent realizations from a common zero-inflated Poisson distribution with parameters $(\bar{\pi}_{hk}, \lambda_{hk})$. Therefore, $(\bar{\pi}_{hk}, \lambda_{hk})$ can be effectively inferred, for every $h = 1, \dots, H$ and $k = 1, \dots, h$, leveraging the information from all the ties among criminals allocated to groups h and k , respectively. Recalling, for example, Li (2012) both parameters are identifiable under the zero-inflated Poisson we consider within each block.

As mentioned before, the inclusion of group structures encoded in \mathbf{z} , not only facilitates inference on efficiency-security tradeoffs measured by the zero-inflated Poisson parameters, but also allows one to obtain evidence of redundancy patterns in the criminal network, along with the corresponding sizes and composition. As a result, the proposed formulation crucially accounts for the two fundamental sources of resilience in criminal organizations, namely efficiency-security via $(\bar{\Pi}, \bar{\Lambda})$ and redundancy through \mathbf{z} . In model (1)–(3), these parameters are inferred from a Bayesian perspective. Within a criminal network context characterized by error-prone measurements and expert knowledge from criminology, such a perspective is particularly suitable to facilitate principled uncertainty quantification and formal inclusion of prior information.

The inclusion of prior distributions further allows one to introduce dependence among the covert and overt ties in the network, which is expected in the context of criminal networks. In fact, although (4) implies that variables x_{vu} and w_{vu} are conditionally independent, for each $v = 2, \dots, V$ and $u = 1, \dots, v - 1$, given the allocation vector and the block-specific parameters, by marginalizing out these latter quantities from the joint likelihood of \mathbf{X} and \mathbf{W} introduces dependence among the entries of these two matrices. More specifically, exploiting conjugacy induced by (2) together with representation (4) of model (1), it is possible to marginalize out analytically $\bar{\Pi}$ and $\bar{\Lambda}$ in $p(\mathbf{X} | \mathbf{z}, \bar{\Pi})$ and $p(\mathbf{W} | \mathbf{z}, \bar{\Lambda})$, respectively, to obtain

$$\begin{aligned} p(\mathbf{X} | \mathbf{z}) &= \prod_{h=1}^H \prod_{k=1}^h \frac{B(a + x_{hk}, b + n_{hk} - x_{hk})}{B(a, b)}, \\ p(\mathbf{W} | \mathbf{z}) &= \left(\prod_{v=2}^V \prod_{u=1}^v w_{vu}! \right)^{-1} \prod_{h=1}^H \prod_{k=1}^h \frac{a_2^{a_1} \Gamma(a_1 + w_{hk})}{(a_2 + n_{hk})^{a_1 + w_{hk}} \Gamma(a_1)}, \end{aligned} \quad (9)$$

where $B(\cdot, \cdot)$ and $\Gamma(\cdot)$ are the Beta and Gamma functions, n_{hk} is the total number of unique pairs involving a criminal in group h and one in group k , whereas x_{hk} and w_{hk} denote the sum, over such pairs, of the entries x_{vu} and w_{vu} in \mathbf{X} and \mathbf{W} , respectively. Let $\mathbf{X}_{-(v,u)}$ and $\mathbf{W}_{-(v,u)}$ be the matrices \mathbf{X} and \mathbf{W} excluding the entries x_{vu} and w_{vu} , respectively. Then, if $z_v = h$ and $z_u = k$, direct application of (9) yields the predictive probabilities

$$\begin{aligned} \text{pr}(x_{vu} = 1 | \mathbf{X}_{-(v,u)}, \mathbf{z}) &= \frac{p(x_{vu} = 1, \mathbf{X}_{-(v,u)} | \mathbf{z})}{p(\mathbf{X}_{-(v,u)} | \mathbf{z})} \\ &= \frac{B(a + x_{hk}^{-(v,u)} + 1, b + n_{hk}^{-(v,u)} - x_{hk}^{-(v,u)})}{B(a + x_{hk}^{-(v,u)}, b + n_{hk}^{-(v,u)} - x_{hk}^{-(v,u)})} = \frac{a + x_{hk}^{-(v,u)}}{a + b + n_{hk}^{-(v,u)}}, \end{aligned} \quad (10)$$

$$\begin{aligned} \text{pr}(w_{vu} = w | \mathbf{W}_{-(v,u)}, \mathbf{z}) &= \frac{p(w_{vu} = w, \mathbf{W}_{-(v,u)} | \mathbf{z})}{p(\mathbf{W}_{-(v,u)} | \mathbf{z})} \\ &= \frac{1}{w!} \frac{\Gamma(a_1 + w_{hk}^{-(v,u)} + w)}{(a_2 + n_{hk}^{-(v,u)} + 1)^{a_1 + w_{hk}^{-(v,u)} + w}} \frac{(a_2 + n_{hk}^{-(v,u)})^{a_1 + w_{hk}^{-(v,u)}}}{\Gamma(a_1 + w_{hk}^{-(v,u)})}, \end{aligned}$$

for each $v = 2, \dots, V$ and $u = 1, \dots, v - 1$, where the apex $-(v,u)$ denotes all the previously-defined quantities, evaluated disregarding the pair of nodes (v, u) . Equation (10) clarifies that the conditional distribution of each covert and overt tie between v and u is influenced by those among nodes allocated to the same pair of groups, as expected in redundant modules within criminal networks. Notice that such a form of dependence is even more general than the classical Markov one adopted in popular exponential random graph models (Frank and Strauss, 1986), since it allows two ties to be dependent also if there is no node in common, as long as these nodes are allocated to the same pair of groups. Dependence across blocks and between \mathbf{X} and \mathbf{W} is instead induced by the shared partition \mathbf{z} and its prior in (3), which in turn translates into dependencies among the ties within the observed network \mathbf{Y} . See Section 2.2 below for a detailed presentation and for a constructive motivation of the assumed supervised Gnedin process prior for \mathbf{z} .

346 2.2. Redundancy via supervised Gneden process prior

347 The definition of a prior for the group membership vector \mathbf{z} is a challenging task since it requires a carefully–
348 tailored distribution for a random partition which can incorporate realistic mechanisms of criminals affiliation
349 to redundant modules, while enabling uncertainty quantification on the unknown number of underlying groups
350 and inclusion of information from exogenous criminal attributes \mathbf{c} , when available.

351 Focusing first on the simpler case in which exogenous attributes \mathbf{c} are not available, routine implementations
352 of stochastic block models rely on Dirichlet–multinomial(α, \bar{H}) priors for \mathbf{z} (Nowicki and Snijders, 2001), where
353 $\bar{H} \geq H$ denotes the total number of groups in the whole population of criminals, including also yet unobserved
354 clusters occupied by criminals that have not been investigated. These Dirichlet–multinomial priors are obtained
355 by marginalizing out in $(z_v | \boldsymbol{\theta}) \sim \text{Multinomial}(1, \boldsymbol{\theta} = (\theta_1, \dots, \theta_{\bar{H}}))$, independently for $v = 1, \dots, V$, the vector
356 of group membership probabilities $\boldsymbol{\theta}$ distributed according to the Dirichlet($\boldsymbol{\alpha} = (\alpha, \dots, \alpha)$). Unfortunately, this
357 choice is practically and conceptually suboptimal within our context since it requires one to pre-select \bar{H} , which
358 is clearly not known in practice. Notice that \bar{H} should not be confused with H which denotes, instead, the total
359 number of groups occupied by the V observed criminals under analysis. In fact, $H \leq \min\{V, \bar{H}\}$. An alternative
360 to pre-specifying \bar{H} would be to assume that $\bar{H} \rightarrow \infty$ when $V \rightarrow \infty$, as done in the infinite relational model
361 (Kemp et al., 2006; Schmidt and Morup, 2013). However, this setting is not realistic in networks associated with
362 organized crime where, as a consequence of predefined rules and pyramidal structures (Paoli, 2007; Catino, 2014,
363 2015), it is more reasonable to expect that the total number of groups \bar{H} is finite, rather than infinite, even in the
364 whole, possibly infinite, population of criminals.

365 To overcome the above challenges, we employ a mixture-of-finite-mixtures construction (see e.g., Geng et al.,
366 2019; Legramanti et al., 2022) which leverages the Dirichlet–multinomial prior for \mathbf{z} , but crucially lets \bar{H} to be
367 finite and random, rather than finite and fixed as in the more classical specifications. In particular, for $a > 0$, let
368 $(a)_n = a(a+1) \cdots (a+n-1)$ with $(a)_0 = 1$. Then, a sensible and computationally-tractable choice in this context is
369 to consider the Gneden process prior (Gnedin, 2010; De Blasi et al., 2015), which is obtained by marginalizing out
370 in $(\mathbf{z} | \bar{H}) \sim \text{Dirichlet–multinomial}(\alpha = 1, \bar{H})$, the random number of groups \bar{H} with probability mass function
371 $p(\bar{H}) = \gamma(1 - \gamma)^{\bar{H}-1}/\bar{H}!$ for $\bar{H} \in \{1, 2, \dots\}$, where $(1 - \gamma)_{\bar{H}-1} = (1 - \gamma)(2 - \gamma)(3 - \gamma) \cdots (\bar{H} - 1 - \gamma)$, whereas
372 $\gamma \in (0, 1)$ is the prior hyper-parameter discussed in detail below. Notice that such a probability mass function for
373 \bar{H} has heavy tails and mode at 1, thus favoring parsimonious reconstructions of redundancy patterns in criminal
374 networks, while maintaining a level of flexibility to account for more complex modular architectures composed by
375 a large number of groups. Moreover, such a prior belongs to the general Gibbs-type class explored by Legramanti
376 et al. (2022) within the context of stochastic block models for binary networks, and, unlike the one considered in
377 Geng et al. (2019), crucially admits an urn-scheme representation which facilitates interpretation and inference.
378 In particular, let $n_{h,-v}$ and H_{-v} denote the cardinality of group h and the total number of non-empty groups,
379 respectively, after removing the v -th criminal. Then, leveraging such an urn scheme, the prior distribution over
380 the group assignments for criminal v , conditioned on the memberships $\mathbf{z}_{-v} = (z_1, \dots, z_{v-1}, z_{v+1}, \dots, z_V)$ of the
381 other $V - 1$ is defined as $\text{pr}(z_v = h | \mathbf{z}_{-v}) \propto (n_{h,-v} + 1)[(V - 1) - H_{-v} + \gamma]$ if h is an already-occupied group by
382 the other criminals, excluding the v -th one, i.e., $h = 1, \dots, H_{-v}$. Conversely, if h corresponds to a new group, i.e.,
383 $h = H_{-v} + 1$, then $\text{pr}(z_v = h | \mathbf{z}_{-v}) \propto H_{-v}(H_{-v} - \gamma)$. Such a representation also clarifies the role of the tuning
384 hyper-parameter $\gamma \in (0, 1)$ in controlling the formation of yet-unseen groups; the higher γ is, the lower the
385 probability of generating new groups. As highlighted in Sections 4 and 5, our empirical results are robust to the
386 choice of γ , and its impact on inference is clearly milder than pre-assuming that \bar{H} is fixed and equal to a single,
387 yet unknown, value or infinite.

388 According to the above urn scheme, when v is a generic criminal entering the network among the other $V - 1$
389 members, such a criminal can either join an already-occupied group of redundant members and inherit the
390 associated efficiency and security architectures, or can create a yet-unseen group with possibly different patterns
391 in the zero-inflation probabilities and rates of interactions with others. This affiliation process crucially depends
392 on the size of the criminal network, the current number and cardinality of non-empty groups and, finally, the
393 tuning parameter $\gamma \in (0, 1)$. All these dimensions are at the core of the structured recruiting process and growth in
394 complexity of criminal organizations (see, e.g., Catino, 2014, 2015), thereby yielding a realistic construction. For
395 example, the presence of $n_{h,-v}$ in the above urn scheme allows the inclusion of a *rich get richer* property which
396 is realistic for those groups of highly operative criminals that are more visible to law enforcement, thus requiring
397 an increased redundancy to preserve resilience.

398 Although the classical Gneden process prior provides a sensible construction, it does not account for information
399 on those exogenous criminal attributes that are often available in law-enforcement investigations. For example,
400 following Section 1.2, in the *Infinito* network we possess additional information on presumed *locale* membership

and role for each criminal. Even if it is unrealistic to believe that the redundancy patterns encoded in \mathbf{z} perfectly overlap with the exogenous partition of criminals provided by these possibly error-prone attributes, excluding this additional information in the prior for \mathbf{z} is similarly-suboptimal. In fact, following criminology theories on trust, human capital and rules (e.g., Campana and Varese, 2013; Charette and Papachristos, 2017; Bouchard and Malm, 2016; Cavallaro et al., 2020; Berlusconi, 2022) it is reasonable to expect that a given criminal is more likely to join groups mainly composed by members with the same combination of *locale*-role, rather than the opposite. To incorporate this information, we tailor the product partition model in Legramanti et al. (2022) to the Gneden process prior for obtaining the supervised version $\text{GN}(\gamma; \mathbf{c})$ in (3) which favors the formation of groups that are homogenous with respect to the attributes of the criminals. More specifically, let $\mathbf{c} = (c_1, \dots, c_V) \in \{1, \dots, C\}^V$ encode the memberships of criminals to a known exogenous partition which, within the *Infinito* network study, corresponds to a combination of *locale*-role information considered also in Legramanti et al. (2022). Then, the assumed $\text{GN}(\gamma; \mathbf{c})$ prior in (3) incorporates \mathbf{c} via the following urn scheme

$$\text{pr}(z_v = h \mid \mathbf{z}_{-v}, \mathbf{c}) \propto \begin{cases} \frac{n_{hc_v, -v} + \alpha_{c_v}}{\alpha_0} (n_{h, -v} + 1)[(V - 1) - H_{-v} + \gamma] & \text{for } h = 1, \dots, H_{-v}, \\ \frac{\alpha_{c_v}}{\alpha_0} [H_{-v}(H_{-v} - \gamma)] & \text{for } h = H_{-v} + 1, \end{cases} \quad (11)$$

where $\alpha_1, \dots, \alpha_C$ are positive cohesion parameters whose sum is α_0 , whereas $n_{hc_v, -v}$ denotes the number of criminals within group h that belong to the same exogenous partition of the v -th one. In the context of the *Infinito* network study, this means that the group-allocation probabilities of the classical, unsupervised, Gneden process prior are now inflated or deflated in (11) by a term which induces a probabilistic homophily favoring the allocation of criminal v to those groups containing a higher fraction of existing members with the same *locale*-role combination. In the motivating application, the exogenous class of each simple affiliate corresponds to the associated *locale*, whereas all bosses have a unique label indicating that such members cover a leadership position. Finally, a subgroup of affiliates belonging to the purple *locale* who are known from the judicial documents to cover a peripheral role are given a distinct label. Being probabilistic, such a reinforcement does not preclude the formation of more heterogenous groups, when necessary; see Section 5. As clarified in Section 3, this prior further facilitates the derivation of a tractable collapsed Gibbs-sampler to perform inference on the posterior distribution $p(\bar{\Pi}, \bar{\Lambda}, \mathbf{z} \mid \mathbf{Y}, \mathbf{c})$.

Notice that, as mentioned before, when \mathbf{c} is not available one can still implement the proposed ZIP-SBM by simply replacing the supervised Gneden process prior $\text{GN}(\gamma; \mathbf{c})$ with the corresponding unsupervised version $\text{GN}(\gamma)$. In this case, the resulting urn scheme representation coincides with the one in (11) after removing the factors $(n_{hc_v, -v} + \alpha_{c_v})/(n_{h, -v} + \alpha_0)$ and α_{c_v}/α_0 .

3. Bayesian Computation and Inference

We derive here a collapsed Gibbs-sampler with a data-augmentation step to draw values from the intractable posterior distribution $p(\bar{\Pi}, \bar{\Lambda}, \mathbf{z} \mid \mathbf{Y}, \mathbf{c})$, and then leverage the simulated samples to perform Bayesian inference on $(\bar{\Pi}, \bar{\Lambda}, \mathbf{z})$ under the ZIP-SBM model presented in (1)-(3). To accomplish this goal, first notice that if the augmented data in the matrices \mathbf{X} and \mathbf{W} were known, then, recalling representation (4) of model (1), $p(\bar{\Pi}, \bar{\Lambda}, \mathbf{z} \mid \mathbf{Y}, \mathbf{X}, \mathbf{W}, \mathbf{c}) = p(\bar{\Pi}, \bar{\Lambda}, \mathbf{z} \mid \mathbf{X}, \mathbf{W}, \mathbf{c}) = p(\bar{\Pi} \mid \mathbf{X}, \mathbf{z})p(\bar{\Lambda} \mid \mathbf{W}, \mathbf{z})p(\mathbf{z} \mid \mathbf{X}, \mathbf{W}, \mathbf{c})$. Leveraging Beta-Bernoulli and Gamma-Poisson conjugacy, together with representation (4), we have that

$$\begin{aligned} (\bar{\pi}_{hk} \mid \mathbf{X}, \mathbf{z}) &\sim \text{Beta}(a + x_{hk}, b + n_{hk} - x_{hk}), & \text{for every } h = 1, \dots, H \text{ and } k = 1, \dots, h, \\ (\bar{\lambda}_{hk} \mid \mathbf{W}, \mathbf{z}) &\sim \text{Gamma}(a_1 + w_{hk}, a_2 + n_{hk}), & \text{for every } h = 1, \dots, H \text{ and } k = 1, \dots, h, \end{aligned} \quad (12)$$

where n_{hk} is the total number of unique pairs involving a criminal in group h and one in group k , while x_{hk} and w_{hk} denote the sum, over such pairs, of the entries x_{vu} and w_{vu} in \mathbf{X} and \mathbf{W} , respectively. These conjugacy properties also allow one to derive closed-form expressions for $p(\mathbf{X} \mid \mathbf{z})$ and $p(\mathbf{W} \mid \mathbf{z})$ that are useful to compute $p(\mathbf{z} \mid \mathbf{X}, \mathbf{W}, \mathbf{c})$. More specifically, exploiting conjugacy in (12) and recalling Section 2.1, we can marginalize out analytically $\bar{\Pi}$ and $\bar{\Lambda}$ in $p(\mathbf{X} \mid \mathbf{z}, \bar{\Pi})$ and $p(\mathbf{W} \mid \mathbf{z}, \bar{\Lambda})$, respectively, to obtain $p(\mathbf{X} \mid \mathbf{z})$ and $p(\mathbf{W} \mid \mathbf{z})$ as in (9).

Combining (9) with the urn-scheme in (11), it is therefore possible to derive closed-form expressions for the full-conditionals of the group assignment z_v of each criminal $v = 1, \dots, V$ given those of the others \mathbf{z}_{-v} , and the matrices \mathbf{X} and \mathbf{W} . More specifically, by direct application of Bayes' rule, it follows that $(z_v \mid \mathbf{z}_{-v}, \mathbf{X}, \mathbf{W}, \mathbf{c})$ is

444 a categorical variable with full-conditional probabilities

$$\text{pr}(z_v = h \mid \mathbf{z}_{-v}, \mathbf{X}, \mathbf{W}, \mathbf{c}) \propto \text{pr}(z_v = h \mid \mathbf{z}_{-v}, \mathbf{c}) p(\mathbf{X} \mid \mathbf{z}_{-v}, z_v = h) p(\mathbf{W} \mid \mathbf{z}_{-v}, z_v = h), \quad (13)$$

445 for every $h = 1, \dots, H_{-v} + 1$, where $p(\mathbf{X} \mid \mathbf{z}_{-v}, z_v = h)$ and $p(\mathbf{W} \mid \mathbf{z}_{-v}, z_v = h)$ can be evaluated as in (9),
446 whereas $\text{pr}(z_v = h \mid \mathbf{z}_{-v}, \mathbf{c})$ admits the closed-form expression from the urn scheme in (11).

447 As a consequence of the above derivations, it is possible to devise a simple Gibbs-sampler which iteratively
448 simulates the group assignment z_v of every criminal v from its tractable full-conditional distribution. Iterating
449 over $v = 1, \dots, V$, yields a Markov chain with stationary distribution $p(\mathbf{z} \mid \mathbf{Y}, \mathbf{X}, \mathbf{W}, \mathbf{c})$. Combining this result with
450 (12) yields, therefore, a collapsed Gibbs-sampler (Van Dyk and Park, 2008) targeting $p(\bar{\Pi}, \bar{\Lambda}, \mathbf{z} \mid \mathbf{Y}, \mathbf{X}, \mathbf{W}, \mathbf{c})$.

451 Despite its tractability, the above MCMC strategy requires \mathbf{X} and \mathbf{W} which are, in fact, not fully observed.
452 Following (4) and Figure 2, the only available information is on $\mathbf{Y} = \mathbf{W} \odot (\mathbf{1} - \mathbf{X})$. Therefore, to implement
453 the previously-derived scheme, it is necessary to introduce a data-augmentation step that generates \mathbf{X} and \mathbf{W} .
454 Leveraging (5) and the representation (4) of model (1), an effective strategy to address such a goal would be to
455 sample from $p(\mathbf{X}, \mathbf{W} \mid \mathbf{Y}, \bar{\Pi}, \bar{\Lambda}, \mathbf{z}, \mathbf{c}) = p(\mathbf{W} \mid \mathbf{Y}, \mathbf{X}, \bar{\Pi}, \bar{\Lambda}, \mathbf{z}) p(\mathbf{X} \mid \mathbf{Y}, \bar{\Pi}, \bar{\Lambda}, \mathbf{z})$, where $p(\mathbf{W} \mid \mathbf{Y}, \mathbf{X}, \bar{\Pi}, \bar{\Lambda}, \mathbf{z}) =$
456 $\prod_{v=2}^V \prod_{u=1}^{v-1} p(w_{vu} \mid y_{vu}, x_{vu}, \bar{\lambda}_{z_v, z_u})$ and $p(\mathbf{X} \mid \mathbf{Y}, \bar{\Pi}, \bar{\Lambda}, \mathbf{z}) = \prod_{v=2}^V \prod_{u=1}^{v-1} p(x_{vu} \mid y_{vu}, \bar{\pi}_{z_v, z_u}, \bar{\lambda}_{z_v, z_u})$. As a direct
457 consequence of the discussion in Section 2.1, samples from $p(x_{vu} \mid y_{vu}, \bar{\pi}_{z_v, z_u}, \bar{\lambda}_{z_v, z_u})$ can be readily obtained by
458 noticing that

$$\begin{cases} (x_{vu} \mid y_{vu}, \bar{\pi}_{z_v, z_u}, \bar{\lambda}_{z_v, z_u}) \sim \delta_0 & \text{if } y_{vu} > 0, \\ (x_{vu} \mid y_{vu}, \bar{\pi}_{z_v, z_u}, \bar{\lambda}_{z_v, z_u}) \sim \text{Bern}[\text{pr}(x_{vu} = 1 \mid y_{vu} = 0, \bar{\pi}_{z_v, z_u}, \bar{\lambda}_{z_v, z_u})] & \text{if } y_{vu} = 0, \end{cases} \quad (14)$$

459 independently for each $v = 2, \dots, V$ and $u = 1, \dots, v-1$, where δ_0 is the Dirac delta at 0, whereas $\text{pr}(x_{vu} = 1 \mid$
460 $y_{vu} = 0, \bar{\pi}_{z_v, z_u}, \bar{\lambda}_{z_v, z_u}) = \text{pr}(x_{vu} = 1 \mid y_{vu} = 0, z_v = h, z_u = k, \bar{\pi}_{hk}, \bar{\lambda}_{hk})$ is the conditional probability derived in
461 closed-form in (6). Hence, if $y_{vu} > 0$, no security strategy has been implemented and, hence, $x_{vu} = 0$. Conversely,
462 if $y_{vu} = 0$, such a zero may be either the result of a hiding mechanism or simply due to an actual zero tie that
463 has not been obscured. Therefore, in this context x_{vu} is drawn from the corresponding full-conditional Bernoulli
464 variable. Given x_{vu} and recalling again the discussion in Section 2.1, samples from $p(w_{vu} \mid y_{vu}, x_{vu}, \bar{\lambda}_{z_v, z_u})$ can
465 be generated as follows

$$\begin{cases} (w_{vu} \mid y_{vu}, x_{vu}, \bar{\lambda}_{z_v, z_u}) \sim \delta_{y_{vu}} & \text{if } y_{vu} > 0, \\ (w_{vu} \mid y_{vu}, x_{vu}, \bar{\lambda}_{z_v, z_u}) \sim \delta_0 & \text{if } y_{vu} = 0 \text{ and } x_{vu} = 0, \\ (w_{vu} \mid y_{vu}, x_{vu}, \bar{\lambda}_{z_v, z_u}) \sim \text{Poisson}(\bar{\lambda}_{z_v, z_u}) & \text{if } y_{vu} = 0 \text{ and } x_{vu} = 1, \end{cases} \quad (15)$$

466 independently for $v = 2, \dots, V$ and $u = 1, \dots, v-1$, where $\delta_{y_{vu}}$ is the Dirac delta at y_{vu} . Therefore, if $y_{vu} > 0$ the
467 tie has not been hidden and, therefore, the actual interaction $w_{vu} = y_{vu}$ has been effectively observed. Conversely,
468 when $y_{vu} = 0$ and $x_{vu} = 0$, then w_{vu} must be necessarily equal to zero, since, also in this case, no security strategy
469 has been implemented. Finally, if $x_{vu} = 1$ then $y_{vu} = 0$, and, as a consequence, w_{vu} can be any value sampled
470 from a Poisson($\lambda_{vu} = \bar{\lambda}_{z_v, z_u}$). Combing (14)–(15) with the previously-derived scheme to sample from $p(\bar{\Pi}, \bar{\Lambda}, \mathbf{z} \mid$
471 $\mathbf{Y}, \mathbf{X}, \mathbf{W}, \mathbf{c})$ yields the data-augmentation collapsed Gibbs-sampler in Algorithm 1. Notice that such a routine
472 samples, iteratively, from

- 473 • **Step 1.** $p(\mathbf{X} \mid \mathbf{Y}, \bar{\Pi}, \bar{\Lambda}, \mathbf{z}, \mathbf{c}) = p(\mathbf{X} \mid \mathbf{Y}, \bar{\Pi}, \bar{\Lambda}, \mathbf{z})$,
- 474 • **Step 2.** $p(\mathbf{W} \mid \mathbf{Y}, \mathbf{X}, \bar{\Pi}, \bar{\Lambda}, \mathbf{z}, \mathbf{c}) = p(\mathbf{W} \mid \mathbf{Y}, \mathbf{X}, \bar{\Lambda}, \mathbf{z})$,
- 475 • **Step 3.** $p(\mathbf{z} \mid \mathbf{Y}, \mathbf{X}, \mathbf{W}, \mathbf{c}) = p(\mathbf{z} \mid \mathbf{X}, \mathbf{W}, \mathbf{c})$,
- 476 • **Step 4.** $p(\bar{\Pi} \mid \mathbf{Y}, \mathbf{X}, \mathbf{W}, \bar{\Lambda}, \mathbf{z}, \mathbf{c}) = p(\bar{\Pi} \mid \mathbf{X}, \mathbf{z})$,
- 477 • **Step 5.** $p(\bar{\Lambda} \mid \mathbf{Y}, \mathbf{X}, \mathbf{W}, \bar{\Pi}, \mathbf{z}, \mathbf{c}) = p(\bar{\Lambda} \mid \mathbf{W}, \mathbf{z})$.

478 Applying the results described in Van Dyk and Park (2008), it can be readily shown that Step 1.–Step 5. yield
479 a Markov chain targeting the augmented posterior $p(\bar{\Pi}, \bar{\Lambda}, \mathbf{z}, \mathbf{X}, \mathbf{W} \mid \mathbf{Y}, \mathbf{c})$. In fact, Step 1.–Step 2. generate the
480 augmented data from the joint full-conditional $p(\mathbf{X}, \mathbf{W} \mid \mathbf{Y}, \bar{\Pi}, \bar{\Lambda}, \mathbf{z}, \mathbf{c})$, whereas Step 3.–Step 5. sample from the
481 parameters' full-conditional distribution $p(\bar{\Pi}, \bar{\Lambda}, \mathbf{z} \mid \mathbf{Y}, \mathbf{X}, \mathbf{W}, \mathbf{c})$.

482 Discarding the draws for \mathbf{X} and \mathbf{W} produced by Algorithm 1, provides samples from the posterior $p(\bar{\Pi}, \bar{\Lambda}, \mathbf{z} \mid$
483 $\mathbf{Y}, \mathbf{c})$ of interest. Leveraging these samples, we then conduct posterior inference on redundancy structures encoded
484 in \mathbf{z} via the variation of information (VI) framework introduced by Wade and Ghahramani (2018) for Bayesian
485 clustering. The VI defines a proper metric among partitions which computes distances between generic grouping

Algorithm 1 Data-augmentation collapsed Gibbs-sampler for ZIP-SBM

```

— input:  $\mathbf{Y}$  and prior hyperparameters  $(a, b)$ ,  $(a_1, a_2)$ , and  $\gamma$ .
— initialize  $\mathbf{z}$ ,  $\bar{\boldsymbol{\Pi}}$  and  $\bar{\boldsymbol{\Lambda}}$ , and set the total number of MCMC iterations T.

for  $t = 1, \dots, T$  do
    for  $v = 2, \dots, V$  and  $u = 1, \dots, v - 1$  do
        1. sample  $x_{vu}$  from (14).
        2. sample  $w_{vu}$  from (15).
    end for
    for  $v = 1, \dots, V$  do
        3.1 reorder the labels in  $\mathbf{z}_{-v}$  so that only groups  $h = 1, \dots, H_{-v}$  are non-empty.
        3.2 compute the full-conditional probabilities for  $z_v$  (up to a proportionality constant) as in (13). The expressions for the quantities  $p(\mathbf{X} | \mathbf{z}_{-v}, z_v = h)$  and  $p(\mathbf{W} | \mathbf{z}_{-v}, z_v = h)$  in (13) are available analytically as in (9), whereas that for  $\text{pr}(z_v = h | \mathbf{z}_{-v}, \mathbf{c})$  can be found in (11).
        3.3 sample  $z_v$  from the categorical variable with probabilities obtained by normalizing those computed in 3.2.
    end for
    — Let  $H^{(t)}$  be the number of non-empty clusters in the previously-sampled  $\mathbf{z}$ 
    for  $h = 1, \dots, H^{(t)}$  and  $k = 1, \dots, h$  do
        4. sample  $\bar{\pi}_{hk}$  from the full-conditional Beta distribution in (12).
        5. sample  $\bar{\lambda}_{hk}$  from the full-conditional Gamma distribution in (12).
    end for
end for

```

486 structures through a comparison of individual and joint entropies (e.g., Meilă, 2007). As discussed in Wade and
487 Ghahramani (2018), such a metric yields a number of practical advantages over other popular alternatives, such
488 as the Binder's loss (Binder, 1978), and facilitates principled point estimation and uncertainty quantification
489 directly within the space of partitions. In particular, under the VI approach, a point estimate for \mathbf{z} is obtained
490 via $\hat{\mathbf{z}} = \text{argmin}_{\mathbf{z}'} \mathbb{E}_{(\mathbf{z}|\mathbf{Y})}[\text{VI}(\mathbf{z}, \mathbf{z}')]$. Similarly, a $1 - \alpha$ credible ball around $\hat{\mathbf{z}}$ can be derived by collecting all the
491 partitions with a VI distance from $\hat{\mathbf{z}}$ less than a given threshold guaranteeing that the ball has at least $1 - \alpha$
492 posterior mass, while having minimum size possible. Such inference procedures are implemented via the R library
493 `mcclust.ext` after computing the $V \times V$ *posterior similarity (or co-clustering)* matrix \mathbf{S} whose generic element
494 s_{vu} estimates $\text{pr}(z_v = z_u | \mathbf{Y})$ via the relative frequency of Gibbs samples in which $z_v^{(t)} = z_u^{(t)}$. By relying solely on
495 measures of posterior co-clustering, this approach to inference on \mathbf{z} does not suffer from possible label-switching
496 issues and does not require relabeling strategies (Stephens, 2000).

497 Concerning inference on the block-specific interaction rates and the zero-inflation probabilities, we emphasize
498 that, although this objective would be possible by leveraging the simulated values of $\bar{\boldsymbol{\Pi}}$ and $\bar{\boldsymbol{\Lambda}}$ from Algorithm 1,
499 these quantities are associated with sampled partitions of \mathbf{z} which vary throughout the Gibbs-sampling routine.
500 This makes inference less interpretable. In fact, in practice, it is more convenient to provide law enforcement with
501 a point estimate $\hat{\mathbf{z}}$ of \mathbf{z} , along with measures of uncertainty around such an estimate, and then study the plug-in
502 posterior distribution $p(\bar{\boldsymbol{\Pi}}, \bar{\boldsymbol{\Lambda}} | \mathbf{Y}, \hat{\mathbf{z}})$ for $\bar{\boldsymbol{\Pi}}$ and $\bar{\boldsymbol{\Lambda}}$, given $\hat{\mathbf{z}}$. This objective can be readily accomplished by re-
503 running Algorithm 1 without steps 3.1–3.3, and \mathbf{z} fixed at $\hat{\mathbf{z}}$. This yields samples from $p(\bar{\boldsymbol{\Pi}}, \bar{\boldsymbol{\Lambda}} | \mathbf{Y}, \hat{\mathbf{z}})$ that can
504 be used to obtain Monte Carlo estimates and uncertainty measures for the block-specific interaction rates and
505 zero-inflation probabilities associated with the specific partition $\hat{\mathbf{z}}$. Although this strategy does not propagate to
506 $\bar{\boldsymbol{\Pi}}$ and $\bar{\boldsymbol{\Lambda}}$ the uncertainty within \mathbf{z} , as illustrated in Sections 4 and 5, the posterior for \mathbf{z} is often well-concentrated
507 and, hence, the underestimation of uncertainty in $p(\bar{\boldsymbol{\Pi}}, \bar{\boldsymbol{\Lambda}} | \mathbf{Y}, \hat{\mathbf{z}})$, relative to $p(\bar{\boldsymbol{\Pi}}, \bar{\boldsymbol{\Lambda}} | \mathbf{Y})$, is typically negligible
508 when compared with the gains in interpretability. Finally, note that, due to the structured representation in (5),
509 estimation and uncertainty quantification on the tie-specific parameters matrices $(\boldsymbol{\Pi}, \boldsymbol{\Lambda})$ can be performed as a
510 byproduct of the inference on $(\bar{\boldsymbol{\Pi}}, \bar{\boldsymbol{\Lambda}}, \mathbf{z})$.

4. Simulation Studies

511 To assess the performance of the proposed ZIP-SBM and illustrate its potential in inference on efficiency-security
512 structures, we consider an in-depth analysis of three simulation scenarios based on different specifications of the
513 parameters in model (1). As illustrated within Figure 3, the networks generated under these three scenarios are
514 characterized by different sizes, number of clusters and various combinations of community, core-periphery and
515 weakly-assortative structures, along with blocks displaying different sparsity and zero-inflation patterns. Some

of these patterns reproduce those expected in criminal networks, thus providing a realistic assessment in the light of the motivating application. The three scenarios considered are described in detail below.

- **Scenario 1:** Data are simulated from the generative representation (4) of the model in (1) considering $V = 80$ criminals partitioned into $H_0 = 5$ groups of size $n_1 = 20$, $n_2 = 25$, $n_3 = n_4 = 15$ and $n_5 = 5$. The matrix $\bar{\Lambda}_0$ has diagonal entries equal to 3 and off-diagonal ones set at 0.1, whereas $\bar{\Pi}_0$ encodes within-block zero-inflation probabilities of 0.05 and across-block ones equal to 0.15. As illustrated within Figure 3, this yields a network \mathbf{Y} characterized by five well-separated community structures and relatively mild security strategies. Notice that the setting $V = 80$ is motivated by the attempt to assess the ZIP-SBM performance on a network with a size similar to the one considered in the 'Ndrangheta network application introduced within Section 1.2, where the total number of criminals under analysis is equal to 84.
- **Scenario 2:** We still simulate data from representation (4) of model (1), focusing again on $V = 80$ criminals allocated to $H_0 = 5$ different groups of size $n_1 = 20$, $n_2 = 25$, $n_3 = n_4 = 15$ and $n_5 = 5$. However, in this case we consider more heterogeneous block-architectures beyond the simple communities, combined with a higher prevalence of covert structures. This goal is accomplished by considering larger values equal to 0.3 for all the entries in $\bar{\Pi}_0$ except for $\bar{\pi}_{0;2,1} = \bar{\pi}_{0;1,2}$, $\bar{\pi}_{0;3,1} = \bar{\pi}_{0;1,3}$, $\bar{\pi}_{0;4,1} = \bar{\pi}_{0;1,4}$, $\bar{\pi}_{0;5,1} = \bar{\pi}_{0;1,5}$, $\bar{\pi}_{0;4,3} = \bar{\pi}_{0;3,4}$ and $\bar{\pi}_{0;5,3} = \bar{\pi}_{0;3,5}$ which are set to 0.6, and $\bar{\pi}_{0;1,1}$, $\bar{\pi}_{0;2,2}$ and $\bar{\pi}_{0;3,3}$ that we fix equal to 0 for assessing the ability of ZIP-SBM in learning blocks with no sparsity due to security strategies. Similarly, the diagonal elements of $\bar{\Lambda}_0$ are set equal to 4, while the off-diagonal ones coincide with 0.5, except for $\bar{\lambda}_{0;3,2} = \bar{\lambda}_{0;2,3}$, $\bar{\lambda}_{0;4,2} = \bar{\lambda}_{0;2,4}$ and $\bar{\lambda}_{0;5,4} = \bar{\lambda}_{0;4,5}$ that we set to 2, and $\bar{\lambda}_{0;5,2} = \bar{\lambda}_{0;2,5}$ which is equal to 6.
- **Scenario 3:** The third scenario provides a more challenging setting which combines the first two — still under representation (4) of the model (1) — in a larger network among $V = 120$ criminals partitioned into $H_0 = 10$ groups of size $n_1 = n_6 = 20$, $n_2 = n_7 = 15$, $n_3 = n_4 = n_8 = n_9 = 10$ and $n_5 = n_{10} = 5$. More specifically, the (5×5) -dimensional sub-matrices $\bar{\Pi}_{0[1:5,1:5]}$ and $\bar{\Lambda}_{0[1:5,1:5]}$ coincide with those specified in scenario 2, whereas $\bar{\Pi}_{0[6:10,6:10]}$ and $\bar{\Lambda}_{0[6:10,6:10]}$ have the same entries as the zero-inflation probability matrix and rate matrix, respectively, in scenario 1, except for $\bar{\pi}_{0;7,6} = \bar{\pi}_{0;6,7}$ which is set to 0.05, $\bar{\lambda}_{0;7,6} = \bar{\lambda}_{0;6,7}$ that is increased to 2, and $\bar{\lambda}_{0;6,6}$, $\bar{\lambda}_{0;7,7}$ which we fix equal to 4. In specifying $\bar{\Pi}_{0[1:5,6:10]} = \bar{\Pi}_{0[6:10,1:5]}^\top$ and $\bar{\Lambda}_{0[1:5,6:10]} = \bar{\Lambda}_{0[6:10,1:5]}^\top$ we consider instead zero-inflation probabilities and rates equal to 0.5 and 0.2, respectively, for all the entries except $\bar{\pi}_{0;10,1} = \bar{\pi}_{0;1,10}$, $\bar{\pi}_{0;8,1} = \bar{\pi}_{0;1,8}$ and $\bar{\pi}_{0;9,3} = \bar{\pi}_{0;3,9}$ which are set to 0.6, and $\bar{\lambda}_{0;10,1} = \bar{\lambda}_{0;1,10}$, $\bar{\lambda}_{0;8,1} = \bar{\lambda}_{0;1,8}$ and $\bar{\lambda}_{0;9,3} = \bar{\lambda}_{0;3,9}$ that we fix to 5.

Figure 3 provides a graphical illustration of the networks simulated under the three aforementioned scenarios. In analyzing the efficiency-security architectures underlying these three simulated networks we implement the newly-developed ZIP-SBM in (1)–(3) along with two relevant competitors which clarify the inference advantages associated with the proposed construction. These competitors include the recently-proposed supervised extended stochastic block model (ESBM) for binary networks (Legramanti et al., 2022) applied to a dichotomized version of the adjacency matrices in Figure 3, and an improved version of the Poisson SBM (P-SBM) in, e.g., McDaid

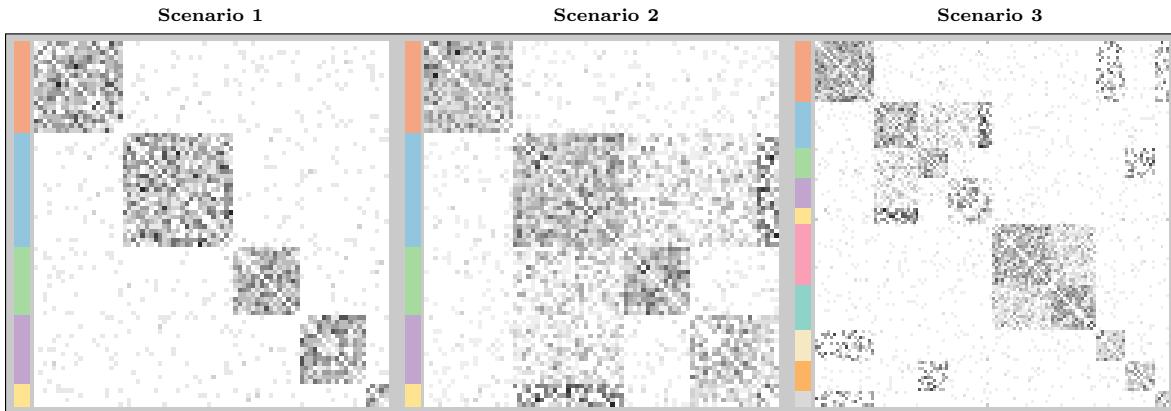


Fig. 3. For scenarios 1, 2 and 3, graphical representation of the simulated adjacency matrix \mathbf{Y} . The color annotation of the rows displays the true grouping structure encoded in the true \mathbf{z}_0 under each scenario. In the three adjacency matrices, the color of each entry ranges from white to black as the corresponding tie goes from zero to the maximum observed count interaction.

552 et al. (2013), which leverages the supervised Gneden process prior in Section 2.2 on \mathbf{z} , rather than the classical
 553 Dirichlet–multinomial. These choices guarantee a fair comparison among the three models, which rely on the same
 554 supervised prior on \mathbf{z} , thus highlighting the benefits associated with the use of zero–inflated Poisson distributions
 555 for the observed ties, rather than Bernoulli or Poisson ones.

556 Although inference on group structures can be accomplished also via alternative solutions, such as, for example,
 557 community detection algorithms (Girvan and Newman, 2002; Newman, 2006; Blondel et al., 2008) and spectral
 558 clustering (Von Luxburg, 2007), the methodological and practical superiority of stochastic block models over
 559 these competing alternatives has been already illustrated in several studies; e.g., Legramanti et al. (2022). Due
 560 to this, we focus here on state-of-the-art SBM formulations that provide competitive alternatives to the proposed
 561 ZIP–SBM model in the context of criminal networks. In particular, the ESBM with supervised Gneden process
 562 prior represents a relevant competitor aligned with the routine practice of dichotomizing the weighted ties prior to
 563 statistical modeling of criminal networks. As clarified below, this perspective yields substantial loss of information
 564 that could be avoided by analyzing the observed ties on the original count scale. The Poisson SBM provides a
 565 sensible and popular solution which is, in fact, a degenerate case of the proposed ZIP–SBM that implicitly assumes
 566 the lack of security structures. This assumption is not realistic in the context of criminal networks and, in fact, as
 567 illustrated in the following, the proposed ZIP–SBM not only expands the inference potentials of both ESBM and
 568 P–SBM, but also yields substantially more accurate reconstructions of the generative mechanisms underlying the
 569 three simulated networks.

570 Table 1 quantifies these advantages with a focus on posterior inference for the underlying grouping structures
 571 in the three simulation scenarios. Results for the ZIP–SBM, ESBM and P–SBM are based on the same supervised
 572 Gneden process prior for \mathbf{z} as in (3), with hyper–parameters $\gamma = 0.3$ and $\alpha_1 = \dots = \alpha_C = 1$, where C is equal
 573 to 5 in the first two scenarios, and to 10 in the third. Supervision in this case is with respect to a contaminated
 574 version \mathbf{c} of the true group membership labels in \mathbf{z}_0 . This is obtained by changing the true group allocation of 20
 575 randomly–selected nodes, under each scenario, thereby allowing one to assess the ability to leverage informative
 576 exogenous partitions, while preserving robustness to possible noise and contamination in these external data.
 577 Albeit sharing the same prior on \mathbf{z} , ZIP–SBM, ESBM and P–SBM differ substantially in the likelihood for the ties.
 578 More specifically, the ESBM relies on Bernoulli interactions with independent Beta(1, 1) — i.e., uniform — priors
 579 on the block probabilities (Legramanti et al., 2022), whereas the P–SBM considers Poisson ties with independent
 580 Gamma(1, 1) — i.e., Exp(1) — priors for the block–specific interaction rates (see e.g., McDaid et al., 2013). Such
 581 a prior is also considered for the entries of $\bar{\Lambda}$ in the proposed ZIP–SBM. As for the hyper–parameters of the
 582 Beta(a, b) priors on the zero–inflation probabilities in $\bar{\Pi}$, we rely instead on the more informative specification
 583 ($a = 1, b = 9$) which is useful in facilitating a more conservative identification of security architectures. Although
 584 these structures are crucial for criminal organizations, it is realistic to expect that the efficiency–security tradeoff
 585 combined with advanced investigations progressively reduce the covert portions of the criminal network. These
 586 hyper–parameter settings always led to accurate inference on the partition structure and on the block–parameters,
 587 under several different network structures both in the simulation studies and in the application. Hence, we suggest
 588 $\gamma = 0.3, \alpha_1 = \dots = \alpha_C = 1, (a = 1, b = 9)$ and $(a_1 = 1, a_2 = 1)$ as a default choice.

589 Posterior inference under the ZIP–SBM relies on 10,000 samples from Algorithm 1, after a conservative burn–in
 590 of 10,000. A plain R implementation of such a routine in a standard laptop required 2.5 minutes to draw the total
 591 of 20,000 samples in scenarios 1 and 2 ($V = 80, H_0 = 5$). Under scenario 3 ($V = 120, H_0 = 10$) such a runtime
 592 increased to 5.5 minutes. Considering the running times of general Gibbs samplers in complex models such as this
 593 one, Algorithm 1 offers a rapid and effective implementation which can efficiently track most criminal networks.
 594 Convergence and mixing is monitored via the traceplots of the quantity $VI(\mathbf{z}^{(t)}, \mathbf{z}_0)$, for $t = 1, \dots, T$, which,
 595 unlike other model–specific quantities, is available for both ZIP–SBM, ESBM and P–SBM, and informs not only
 596 on mixing and convergence of the MCMC chains for $\mathbf{z}^{(t)}, t = 1, \dots, T$ under each of the three models, but also
 597 on the concentration of these chains around the true \mathbf{z}_0 . Although graphical analysis of these traceplots suggests
 598 much rapid convergence and effective mixing under ZIP–SBM in all the three scenarios, we opt for a conservative
 599 burn–in which can be safely employed also for the competing ESBM and P–SBM. In fact, in the second and third
 600 scenario, the P–SBM experiences challenges in convergence, thus requiring the conservative burn–in employed.
 601 Posterior inference under ESBM proceeds via the Gibbs–sampler in Legramanti et al. (2022), while P–SBM can
 602 be implemented via a minor modification of Algorithm 1 removing Steps 1, 2 and 4, and setting $\mathbf{Y} = \mathbf{W}$. As
 603 in Legramanti et al. (2022), we also found results robust to moderate changes in the Gneden process hyper–
 604 parameter $\gamma \in (0, 1)$. In particular, setting γ to either 0.1 or 0.7, rather than 0.3, did not change the ZIP–SBM
 605 performance displayed in Table 1. The same robustness to the choice of the prior hyper–parameters was observed
 606 also when considering $(a = 1, b = 6)$ or $(a = 1, b = 4)$, instead of $(a = 1, b = 9)$. Similarly, setting (a_1, a_2) to
 607 either $(a_1 = 1, a_2 = 2.5)$ or $(a_1 = 2.5, a_2 = 1)$, rather than $(a_1 = 1, a_2 = 1)$, did not modify the results in Table 1

Table 1. For scenarios 1, 2 and 3, performance of ZIP–SBM and relevant competitors (ESBM and P–SBM) in recovering the true underlying partition \mathbf{z}_0 . This performance is measured via: (i) the number \hat{H} of distinct clusters in the estimated grouping structure $\hat{\mathbf{z}}$, (ii) the VI distance $VI(\hat{\mathbf{z}}, \mathbf{z}_0)$ between the estimated partition $\hat{\mathbf{z}}$ and the true one \mathbf{z}_0 , (iii) the normalized mutual information $NMI(\hat{\mathbf{z}}, \mathbf{z}_0)$ between $\hat{\mathbf{z}}$ and \mathbf{z}_0 , (iv) the posterior mean $\mathbb{E}[VI(\mathbf{z}, \mathbf{z}_0) | \mathbf{Y}]$ of the VI distance from the true \mathbf{z}_0 (which measures the overall concentration of the posterior for \mathbf{z} around the true underlying partition \mathbf{z}_0), and (v) the distance $VI(\hat{\mathbf{z}}, \mathbf{z}_b)$ between the estimated partition $\hat{\mathbf{z}}$ and the 95% credible bound \mathbf{z}_b . Bold values denote best performance in each column.

SCENARIO	\hat{H}			$VI(\hat{\mathbf{z}}, \mathbf{z}_0)$			$NMI(\hat{\mathbf{z}}, \mathbf{z}_0)$			$\mathbb{E}[VI(\mathbf{z}, \mathbf{z}_0) \mathbf{Y}]$			$VI(\hat{\mathbf{z}}, \mathbf{z}_b)$		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
ZIP–SBM	5	5	10	0.00	0.00	0.00	1.00	1.00	1.00	0.0009	0.0001	0.0001	0.00	0.00	0.00
ESBM	5	4	8	0.00	0.20	0.40	1.00	0.91	0.87	0.0027	0.2156	0.4036	0.00	0.13	0.00
P–SBM	5	8	13	0.00	0.45	0.23	1.00	0.84	0.93	0.0024	0.5880	0.2491	0.00	0.69	0.09

for ZIP–SBM, except for the third scenario where the hyper–parameter setting ($a_1 = 2.5, a_2 = 1$) led the ZIP–SBM to collapse three groups, out of the ten in total, into a single one. This slight deterioration in clustering accuracy is, however, not substantial, provided that the three collapsed groups are characterized by highly–similar block–parameters, and thus, inference on the underlying data–generating mechanism is not substantially affected. All routines proved robust also to the initialization of the partition structure \mathbf{z} . Although initializing the sampling algorithms to V singleton groups — with each node occupying its own cluster — generally led to a slightly–more–rapid convergence, other starting configurations based on randomly–generated node partitions or more extreme settings with all the nodes allocated to the same unique group, did not affect inference.

Leveraging the posterior samples for \mathbf{z} from the implementations of the ZIP–SBM, ESBM and P–SBM under the three simulation scenarios, Bayesian inference on the partition structure proceeds under the VI framework discussed in detail in Section 3. Results of these analyses are illustrated in Table 1, through several performance measures which quantify the accuracy of the estimated partition $\hat{\mathbf{z}}$ in recovering the true one \mathbf{z}_0 (see \hat{H} , $VI(\hat{\mathbf{z}}, \mathbf{z}_0)$ and $NMI(\hat{\mathbf{z}}, \mathbf{z}_0)$), along with the concentration of the entire posterior for \mathbf{z} around \mathbf{z}_0 (see $\mathbb{E}[VI(\mathbf{z}, \mathbf{z}_0) | \mathbf{Y}]$), and the size of the credible ball associated with such a posterior (see $VI(\hat{\mathbf{z}}, \mathbf{z}_b)$). Notice that, for completeness, we also compute the normalized mutual information $NMI(\hat{\mathbf{z}}, \mathbf{z}_0) \in [0, 1]$ between $\hat{\mathbf{z}}$ and \mathbf{z}_0 which provides a measure of similarity among the two partitions (e.g., Newman et al., 2020). Large values of $NMI(\hat{\mathbf{z}}, \mathbf{z}_0)$ imply low $VI(\hat{\mathbf{z}}, \mathbf{z}_0)$. As outlined in Table 1, all these measures confirm the superior performance of the ZIP–SBM in accurately learning the true underlying partition \mathbf{z}_0 in all scenarios. Despite yielding a slightly lower concentration of the posterior around \mathbf{z}_0 than the ZIP–SBM, both the ESBM and P–SBM achieve a similarly–accurate performance in point estimation within scenario 1. Conversely, in scenarios 2 and 3 the inference accuracy of both the ESBM and P–SBM drops, while the ZIP–SBM remains still able to recover exactly the true underlying partition in \mathbf{z}_0 . The failure of the ESBM is caused by the information loss arising from dichotomization, which makes the fourth and fifth groups indistinguishable in scenario 2. The same holds for clusters four–five and six–seven in scenario 3. The challenges encountered by the P–SBM are instead attributable to the inability of accounting for the sparsity in scenarios 2 and 3, which results in the creation of additional groups to cope with the lack of flexibility in the Poisson likelihood. The ZIP–SBM avoids data–dichotomization and properly incorporates both sparsity and weighted ties information, thereby achieving exact recovery of \mathbf{z}_0 and effective concentration of the posterior distribution around such a \mathbf{z}_0 also in scenarios 2 and 3.

As displayed in Table 2, the high accuracy of the ZIP–SBM in Table 1 is not specific to the three simulated networks in Figure 3, but rather holds generally in replicated simulations from the generative mechanisms behind the three scenarios. More specifically, we replicate the analyses in Table 1 — with focus on the proposed ZIP–SBM — for 100 different networks simulated under each of the three scenarios, thereby obtaining, for every scenario, 100

Table 2. Performance of the ZIP–SBM in 100 replicated studies from scenarios 1, 2 and 3. This performance is measured via the median of the quantities in Table 1, computed over the 100 replicated simulations from each of the three scenarios. The values within brackets correspond to the difference between the 90% and 10% quantiles of the different measures over the 100 replicates. These latter values quantify the variability of the performance measures under analysis across the 100 simulations.

SCENARIO	\hat{H}			$VI(\hat{\mathbf{z}}, \mathbf{z}_0)$			$NMI(\hat{\mathbf{z}}, \mathbf{z}_0)$			$\mathbb{E}[VI(\mathbf{z}, \mathbf{z}_0) \mathbf{Y}]$			$VI(\hat{\mathbf{z}}, \mathbf{z}_b)$		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
ZIP–SBM	5.0 (0.00)	5.0 (0.00)	10.0 (2.00)	0.00 (0.00)	0.00 (0.00)	0.07 (0.24)	1.00 (0.00)	1.00 (0.00)	0.99 (0.07)	0.0005 (0.002)	0.0002 (0.003)	0.0705 (0.275)	0.00 (0.00)	0.00 (0.00)	0.00 (0.20)

different values of the performance measures reported in Table 1 for the ZIP–SBM. Table 2 displays the median of these 100 values along with the difference between the 90% and 10% quantiles to quantify a range of variation across the 100 simulations. These results further strengthen the proposed ZIP–SBM which proves highly reliable and yields accurate inference on the true grouping structures among the nodes in almost all the 100 replications, under each of the three simulation scenarios.

Recalling the results in Table 1, notice that, even if the ESBM and P–SBM achieve accurate point estimation of \mathbf{z} in some scenario (e.g., scenario 1), these two models are, by construction, not able to disentangle security–efficiency architectures, and learn accurately both $\bar{\Pi}$ and $\bar{\Lambda}$. This is clear when comparing the posterior mean of $\bar{\Pi}$ and $\bar{\Lambda}$ associated with the estimated partition $\hat{\mathbf{z}}$ under the ZIP–SBM in scenario 1, with those provided by the ESBM and P–SBM, respectively. More specifically, a proxy for $\bar{\Pi}$ under the ESBM can be obtained as the posterior mean of the block–specific probabilities of a zero tie, while $\bar{\Lambda}$ can be compared against the posterior mean of the Poisson rates estimated, for each block, via the P–SBM. Under the ZIP–SBM, the mean absolute difference between the entries in $\bar{\Pi}_0$ and those in the estimated $\bar{\Pi}$ is 0.03, a value which is orders of magnitude lower than the overall error of 0.52 achieved under the ESBM. Similarly, the absolute error in recovering $\bar{\Lambda}_0$ under the ZIP–SBM is 0.03, again improving over the P–SBM error, which is 0.05. Hence, although the ESBM and P–SBM recover \mathbf{z} in scenario 1, these models are not as effective as the ZIP–SBM in estimating $\bar{\Pi}_0$ and $\bar{\Lambda}_0$. The poor performance of the ESBM in learning $\bar{\Pi}_0$ is due to the fact that, by construction, such a model cannot separate truly–zero ties from hidden ones. Similarly, the P–SBM assumes that all the zero–ties are realizations from a Poisson, and hence, yields an underestimation of the actual rates of interaction in the different blocks. The ZIP–SBM not only can learn both $\bar{\Pi}$ and $\bar{\Lambda}$, but also provides relatively accurate estimates of both matrices. These are further confirmed by the posterior standard deviations for the entries in $\bar{\Pi}$ and $\bar{\Lambda}$ whose first, second and third quartiles are [0.088, 0.102, 0.108] and [0.024, 0.030, 0.041], respectively. This high accuracy in inference for the entries of $\bar{\Pi}$ and $\bar{\Lambda}$ is preserved also for blocks displaying no sparsity patterns induced by zero inflations. For example, in scenario 2, we have $\bar{\pi}_{0;1,1} = \bar{\pi}_{0;2,2} = \bar{\pi}_{0;3,3} = 0$ and $\bar{\lambda}_{0;1,1} = \bar{\lambda}_{0;2,2} = \bar{\lambda}_{0;3,3} = 4$. The estimates provided by the ZIP–SBM for these parameters are $\hat{\pi}_{1,1} = 0.007$, $\hat{\pi}_{2,2} = 0.006$, $\hat{\pi}_{3,3} = 0.012$ and $\hat{\lambda}_{1,1} = 3.9$, $\hat{\lambda}_{2,2} = 3.9$, $\hat{\lambda}_{3,3} = 4.4$.

As clarified in Figure 4, the above advantages open avenues for a new set of inference strategies that cannot be considered under the ESBM and P–SBM, but are of key interest in criminology. A core one is the quantification of which observed zero ties $y_{vu} = 0$ are, in fact, due to a security strategy applied to a non–zero underlying interaction among the generic criminals v and u . Following Section 2.1, these events have probabilities defined in (7) for each $v = 2, \dots, V$ and $u = 1, \dots, v - 1$, which can be evaluated at the true parameters $(\mathbf{z}_0, \bar{\Pi}_0, \bar{\Lambda}_0)$, and also estimated via Monte Carlo by averaging (7) over the posterior samples from $p(\bar{\Pi}, \bar{\Lambda} | \mathbf{Y}, \hat{\mathbf{z}})$ produced by the Algorithm 1, in combination with $\hat{\mathbf{z}}$. Under the ZIP–SBM, the absolute difference between these true and estimated probabilities, averaged over the different pairs of nodes, is 0.02, 0.15 and 0.10 for scenarios 1, 2 and 3, respectively. This result supports the overall accuracy of the ZIP–SBM in the estimation of such probabilities, thereby allowing one to unveil which zero ties should be prioritized in the investigations. Figure 4 illustrates the

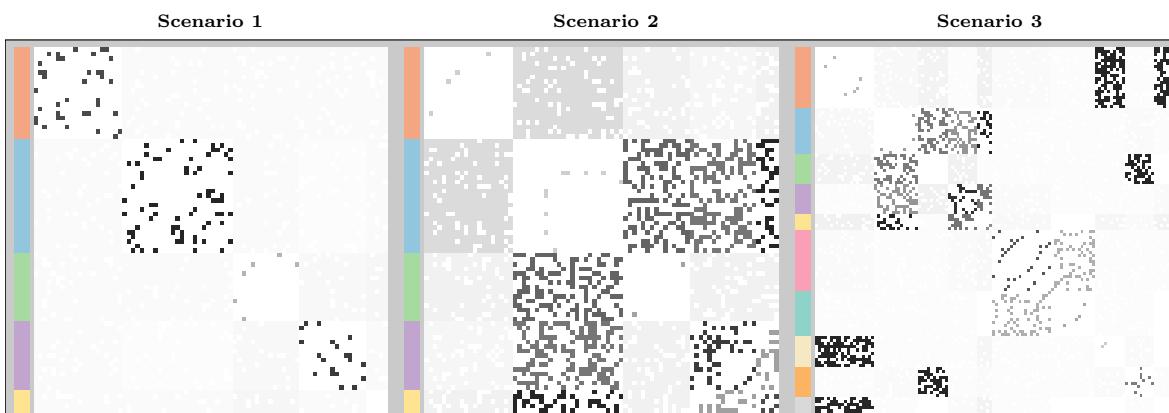


Fig. 4. For scenarios 1, 2 and 3, graphical representation of the estimated probabilities that each observed zero tie $y_{vu} = 0$ corresponds, in fact, to a strictly positive obscured interaction. The color annotation of the rows displays the grouping structure $\hat{\mathbf{z}}$ estimated under the proposed ZIP–SBM model. The color of each entry in the three matrices ranges from light gray to black as the estimated probabilities, under the proposed ZIP–SBM, range from 0 to 1. White entries denote non–zero observed ties ($y_{vu} > 0$).

output of these analyses under the ZIP–SBM for the three networks in Figure 3. According to Figure 4, among the observed zero ties in the three networks in Figure 3, the estimates provided by the ZIP–SBM correctly prioritize those belonging to blocks in which the associated rate and zero-inflation probability make such zero ties highly unusual under a standard Poisson. This effectiveness is also confirmed by the posterior standard deviations for the probabilities under analysis, whose first, second and third quartiles are [0.015, 0.017, 0.021], [0.045, 0.057, 0.075], and [0.014, 0.023, 0.053] for scenarios 1, 2 and 3, respectively, suggesting accurate concentration under the ZIP–SBM. It is important to emphasize that, in the more sparse blocks within scenarios 2–3, the estimation accuracy slightly deteriorates due to the lack of enough non-zero ties to effectively infer the rate of the count distribution. This is a general problem in zero-inflated contexts. Nonetheless, the results of the ZIP–SBM are still satisfactory even for these challenging scenarios where state-of-the-art alternatives fail.

5. Evidence of Efficiency and Security Structures in the Infinito Network

We conclude by illustrating the innovative inference potential of the proposed ZIP–SBM in the analysis of the *Infinito* 'Ndrangheta network presented in Section 1.2. State-of-the-art analyses of such a network focus either on detecting basic community architectures that suggest an overly-simplified horizontal structure underlying the 'Ndrangheta organization (e.g., Calderoni et al., 2017), or study a dichotomized version of the original weighted network (Legramanti et al., 2022) which hinders the potential to unveil those security and efficiency architectures of key interest in criminology (Morselli et al., 2007; Calderoni, 2012; Bouchard and Malm, 2016; Cavallaro et al., 2020). In fact, as illustrated in the simulation studies in Section 4, neither the ESBM, nor P–SBM, can disentangle these two fundamental structures. Conversely, the proposed ZIP–SBM is inherently motivated by the attempt to address such an endeavor via a principled model that can effectively incorporate and quantify both security and efficiency strategies, thus motivating our focus on the ZIP–SBM in the *Infinito* network study.

To support the above remark, we analyze the *Infinito* network under the proposed ZIP–SBM, considering the same hyper-parameters and MCMC settings as in the simulation studies in Section 4. This is useful in order to check the robustness of results when such default choices are considered in the analysis of substantially different networks. In fact, we still obtain satisfactory convergence and mixing (see e.g., Figure 5), along with a highly-informative reconstruction of the redundancy structures underlying the *Infinito* network. These group patterns, encoded in $\hat{\mathbf{z}}$, are shown in Figure 6 which not only yields quantitative support to a number of criminology theories, but also unveils more nuanced dynamics underlying this specific criminal organization. The former are evident in a preference to create redundancies within *locali*, rather than across these modular units, while further diversifying the affiliates from bosses in the formation of groups. This suggests a peculiar ability to guarantee resilience at different levels of the organization, while clarifying that the creation of redundancies is more challenging when moving from peripheral groups of simple affiliates to the core ones, mostly comprising bosses with highly specific human and social capital. In fact, such central groups have progressively lower size.

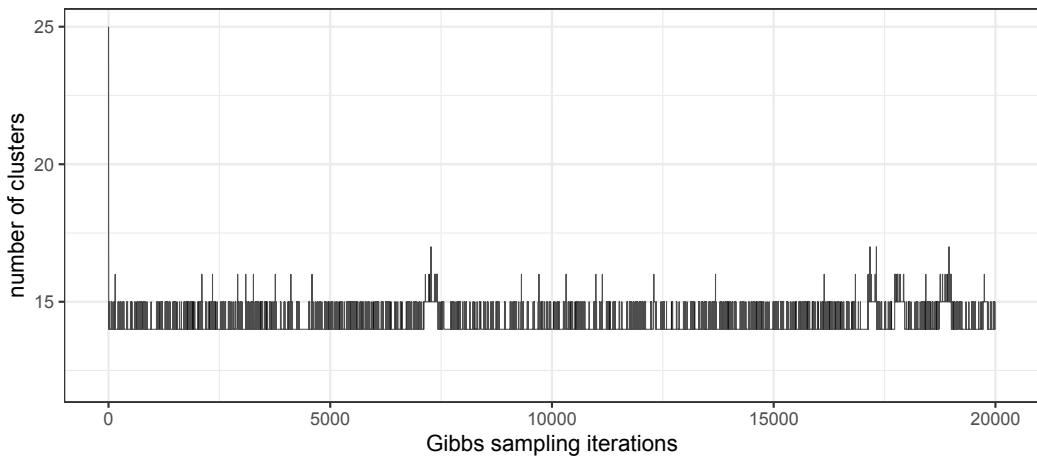


Fig. 5. For the *Infinito* 'Ndrangheta network application, traceplot of the number $H^{(t)}$ of occupied clusters in the sampled partition $\mathbf{z}^{(t)}$, for $t = 1, \dots, 20,000$, under Algorithm 1.

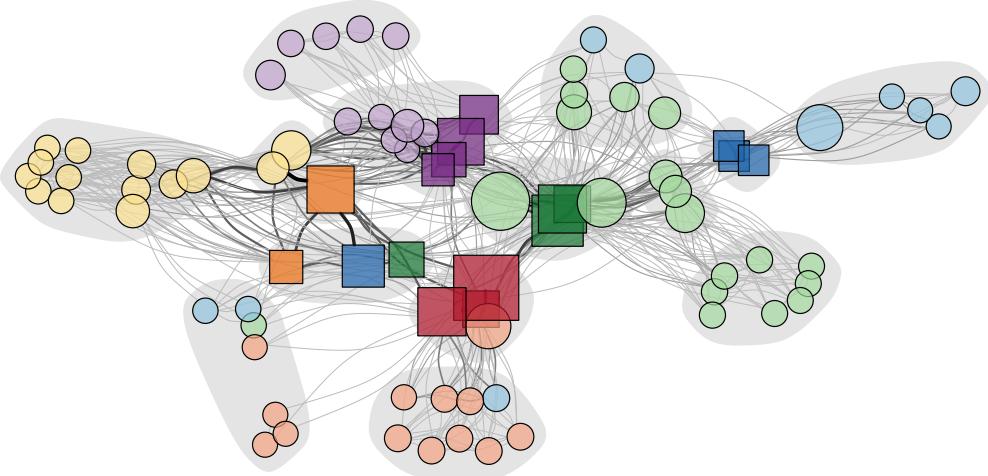


Fig. 6. Graphical representation of the *Infinito* network along with the grouping structure estimated under the proposed ZIP–SBM. The positions of the different criminals are obtained via force directed placement (Fruchterman and Reingold, 1991), whereas colors indicate the presumed *locale* membership and role (darker square nodes indicate the suspected bosses of each *locale*, while lighter circles denote simple affiliates). The size of each node is proportional to its betweenness, while the gray areas highlight the different groups encoded in $\hat{\mathbf{z}}$.

Although the above recurring patterns suggest a highly-regulated organizational structure, which is known to characterize 'Ndrangheta (e.g., Paoli, 2007; Catino, 2014; Sergi and Lavorgna, 2016), the flexibility of the proposed ZIP–SBM is also able to detect more peculiar dynamics related to the history of the specific organization reported in the judicial documents of "Operazione *Infinito*". In fact, although the supervised Gnedin process prior in (3) facilitates some overlap between the exogenous partition \mathbf{c} provided by role–*locale* information and the endogenous one $\hat{\mathbf{z}}$ induced by the redundancies within the network, Figure 6 still highlights peculiar interaction dynamics among criminals in different *locali*. For example, according to the judicial documents, the criminal belonging to the blue *locale* that has been allocated to the group of simple affiliates from the red one, is a member trying to create a new *locale*. To this end, the results in Figure 6 suggest that such a criminal might be in the process of recruiting affiliates from outside the area of its original *locale*, with a preference for those belonging to the red one. It is also interesting to notice a group composed of bosses from different *locali*, with two members relatively far, under a network perspective, from the corresponding original *locali*. This is in line with the fact that these two bosses supported an unsuccessful attempt to increase the independence of the 'Ndrangheta group in Lombardy from the leading Calabria families, causing a need for these members to move away from their original areas and strengthen relations with other *locali*. From this perspective, Figure 6 points toward a progressive movement in the direction of the yellow *locale*. It is also worth noticing that this independence attempt led to the murder of a high-rank member who had key coordinating roles for the whole organization and direct connections with the green *locale*. Figure 6 suggests that this event has resulted in a fragmentation of the green *locale* in multiple sub-groups. Finally, notice that the core groups of bosses also comprise some simple affiliates, thus suggesting that the corresponding role might be more central than reported in the judicial documents. This is the case of the highly central affiliate allocated to the group of bosses from the green *locale*, who is, in fact, a high-rank member of the organization with coordinating roles among the different *locali* while being in charge of reporting to the leading 'Ndrangheta families in Calabria.

The above results, which are based on the minimum–VI point estimate $\hat{\mathbf{z}}$ of \mathbf{z} , are further supported by the fact that the whole posterior distribution is well-concentrated around $\hat{\mathbf{z}}$. More specifically, the VI distance between $\hat{\mathbf{z}}$ and the partition \mathbf{z}_b at the edge of the 95% credible ball is 0.301, a value much lower than the maximum achievable VI distance between two generic partitions of the $V = 84$ analyzed criminals, which is $\log_2 84 = 6.392$ (Wade and Ghahramani, 2018). Similarly, the first and third quartiles of the posterior distribution on the number of non-empty groups are both equal to 14, which coincides with the total number $\hat{H} = 14$ of occupied groups in $\hat{\mathbf{z}}$. These results motivate an in-depth analysis of the covert and overt architectures behind the *Infinito* network leveraging the strategies presented in Sections 2.1 and 3, which condition on the estimated $\hat{\mathbf{z}}$.

Figure 7 illustrates the innovative inference potentials of the proposed ZIP–SBM on these key, yet-unexplored, architectures. In fact, while the analysis of Legramanti et al. (2022) on the dichotomized version of the *Infinito*

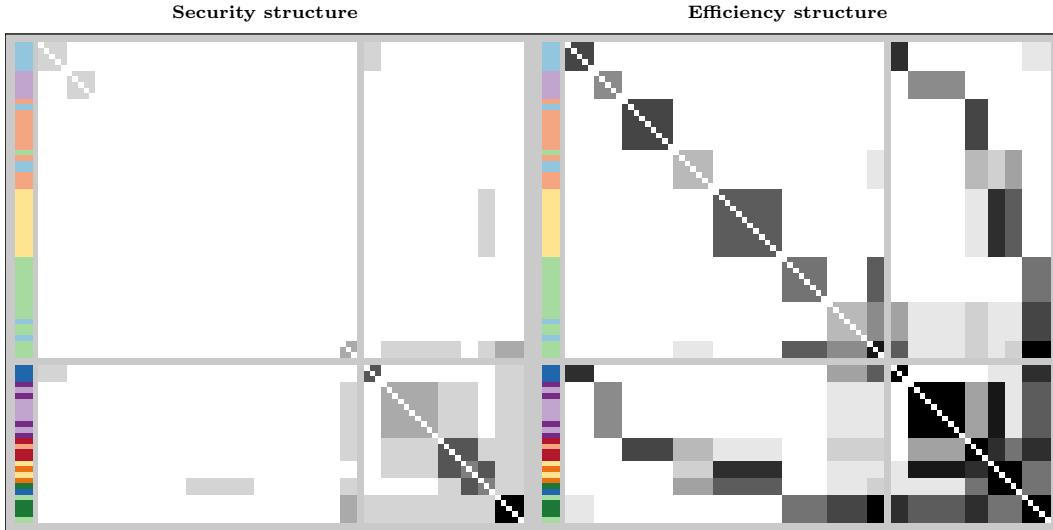


Fig. 7. Adjacency matrices representing security and efficiency block structures of the *Infinito* network, inferred under the proposed ZIP-SBM. Criminals are re-ordered and partitioned in blocks according to the estimated grouping structure $\hat{\mathbf{z}}$. Side colors correspond to the different *locali*, with darker and lighter shades denoting bosses and affiliates, respectively. The security structure displays, for each block, the posterior mean of the probability that a generic observed zero tie in that block is the result of a hiding mechanism. Conversely, the efficiency structure represents, for each block, the posterior mean of the probability that a generic tie within that block, either hidden or not, is > 0 . The color of each entry in the two matrices ranges from white to black as the corresponding probabilities go from 0 to 1. The gray lines separate groups containing only simple affiliates from those comprising also bosses.

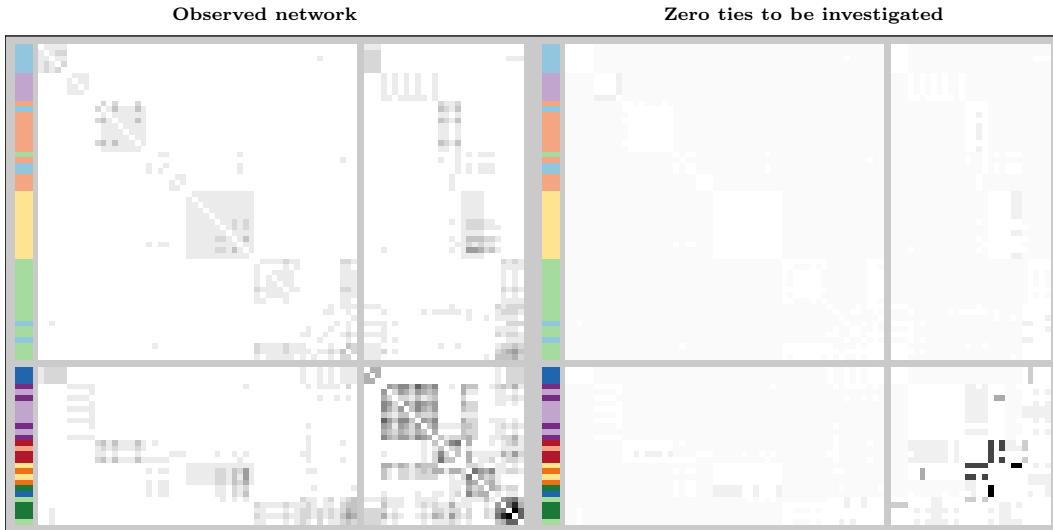


Fig. 8. Graphical representation of the adjacency matrix \mathbf{Y} associated with the *Infinito* network, along with the posterior means of the probabilities that each observed zero tie $y_{vu} = 0$ corresponds, in fact, to a strictly positive obscured interaction. Criminals are re-ordered and partitioned in blocks according to the group structure estimated under ZIP-SBM. Side colors correspond to the *locali*, with darker and lighter shades denoting bosses and affiliates, respectively. In the first matrix, the color of each entry ranges from white to black as the corresponding tie goes from zero to the maximum observed count interaction, whereas in the second the color goes from light gray to black as the estimated probabilities, under the ZIP-SBM, range from 0 to 1. White entries denote non-zero observed ties ($y_{vu} > 0$). Gray lines separate groups containing only simple affiliates from those comprising also bosses.

⁷⁴¹ network yields a similarly-refined reconstruction of the hierarchical group structures underlying the targeted
⁷⁴² 'Ndrangheta organization, the perspective considered by the authors rules out the possibility of disentangling and
⁷⁴³ quantifying security and efficiency structures. Figure 7 addresses this objective via a graphical representation of
⁷⁴⁴ the posterior mean for the probabilities in (6) and (8), respectively, which can be computed, for each $v = 2, \dots, V$

and $u = 1, \dots, v - 1$, through Monte Carlo leveraging $\hat{\mathbf{z}}$ and the posterior samples from $p(\bar{\boldsymbol{\Pi}}, \bar{\boldsymbol{\Lambda}} | \mathbf{Y}, \hat{\mathbf{z}})$; see also Section 3 for further details and recall the definition of π_{vu} and λ_{vu} in (5). As discussed in Section 2.1, the first probability quantifies the chance that a potential zero tie among criminals v and u is the result of a security strategy. Conversely, the second evaluates the probability that v and u establish a non-zero tie, irrespectively of whether this tie has been obscured or not, thus unveiling the actual efficiency structures of the organization.

As shown in Figure 7, the ZIP-SBM uncovers a hierarchical efficiency structure underlying the *Infinito* network, characterized by a peculiar combination of community architectures with across-group interactions that occur through highly-structured core-periphery patterns between the groups of affiliates and those including the bosses. The within-block interactions progressively intensify while moving from the peripheral groups of affiliates to the core ones of bosses, whereas the across-block patterns clearly suggest a tendency of simple affiliates to connect to the core only via the bosses of the corresponding *locale*. These findings provide important quantitative support to qualitative criminology theories on the hierarchical structure of 'Ndrangheta (see e.g., Paoli, 2007; Catino, 2014; Sergi and Lavorgna, 2016), while highlighting a need to strengthen ties at the higher levels of the organizational pyramid for guaranteeing efficiency in coordinating illicit activities. Noticeably, as illustrated in the first panel of Figure 7, such an intensification of the underlying ties is inherently combined with highly-structured security architectures that systematically obscure these increasingly-strong interactions while moving from the periphery of simple affiliates to the core of bosses. In fact, Figure 7 provides clear evidence of security structures mostly in the within- and across-block ties among the leading groups of bosses. To the best of our knowledge, these results provide the first quantitative illustration of how the conjectured efficiency-security tradeoff (Morselli et al., 2007) is realistically implemented in practice by organized crime to shape the structure of the observed network.

Notice that the combination of the two matrices in Figure 7 yields an accurate reconstruction of the observed network \mathbf{Y} in Figure 8. Moreover, besides incrementing knowledge on the structure and function of criminal networks, the ZIP-SBM also allows the identification of which currently-observed zero ties are more likely to hide actual non-zero interactions and, as a consequence, should be object of further investigation by law enforcement. These guidelines can be obtained from the analysis of the posterior mean for the probabilities in (7), which is displayed in the second matrix of Figure 8 and, consistent with the previous discussion, points toward the need of further investigations mainly on the across-block interactions among specific bosses from different *locali*. The reliability of these results is supported by a low posterior uncertainty. For instance, the first, second and third quartiles of the posterior standard deviations for the entries of the matrices in Figure 7 are [0.099, 0.102, 0.107] and [0.018, 0.031, 0.057], respectively.

6. Conclusions and Future Research Directions

This article develops innovative models and inference methods to answer a fundamental objective in criminology, namely the identification of redundancy, security and efficiency structures in organized crime from the analysis of weighted networks among the corresponding members. Despite the relevance of these architectures in guiding law enforcement (see e.g., Campana and Varese, 2022; Diviák, 2022), the methodological challenges underlying the attempt to disentangle such structures have hindered advancements along these lines. The proposed ZIP-SBM effectively covers this gap via a combination of stochastic block models, zero-inflated Poisson distributions and supervised Gneden process priors which allow to incorporate fundamental concepts of redundancy and homophily, while crucially leveraging structure also in the observed zero ties to ultimately learn both security and efficiency structures. This yields a unique Bayesian modeling framework for criminal networks that can be implemented via effective data-augmentation collapsed Gibbs-samplers and provides superior performance along with innovative inference potentials relative to state-of-the-art alternatives, both in simulations and in applications. All these advantages are clear in the motivating *Infinito* network application, where the newly-proposed ZIP-SBM yields an unprecedented reconstruction of the security and efficiency structures behind a complex 'Ndrangheta organization from the analysis of weighted patterns of co-attendances to summits.

The aforementioned results are expected to stimulate further research along these directions. For example, from an applied perspective it would be of interest to consider more extensive implementations of the proposed ZIP-SBM to analyze currently-available criminal network data, such as those within the UCINET repository. This would provide an important opportunity to understand how security and efficiency structures vary across different criminal organizations, including terrorists networks, or within the same organization but with respect to different forms of interaction, i.e., meetings, phone calls, family ties, and others, possibly monitored over time and/or by different investigation agencies. Such a multiplex and/or dynamic structure would also require extensions of the proposed ZIP-SBM to allow both the block-specific parameters and, potentially, the partition structure, to vary across the different network views for the same criminal organization. The contributions by e.g., Le et al. (2018);

⁷⁹⁹ Mantziou et al. (2024); Durante and Dunson (2014) and Durante et al. (2017) could provide valuable ideas and
⁸⁰⁰ building-blocks to address this goal. From a methodological perspective it would be also of interest to consider
⁸⁰¹ alternative forms of zero-inflation in the general class of zero-inflated power series models (e.g., Ghosh et al.,
⁸⁰² 2006) that may yield a more flexible characterization of weighted ties in other criminal networks, beyond those
⁸⁰³ analyzed in this article.

⁸⁰⁴ Acknowledgments

⁸⁰⁵ This research is funded by the European Union (ERC, NEMESIS, project number 101116718). Views and opinions
⁸⁰⁶ expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or
⁸⁰⁷ the European Research Council Executive Agency. Neither the European Union nor the granting authority can
⁸⁰⁸ be held responsible for them.

⁸⁰⁹ The Insight Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number
⁸¹⁰ 12/RC/2289_P2.

⁸¹¹ References

- ⁸¹² Aicher, C., Jacobs, A. Z., and Clauset, A. (2015), Learning latent block structure in weighted networks, *Journal
⁸¹³ of Complex Networks*, 3, 221–248.
- ⁸¹⁴ Berlusconi, G. (2022), Come at the king, you best not miss: criminal network adaptation after law enforcement
⁸¹⁵ targeting of key players, *Global Crime*, 23, 44–64.
- ⁸¹⁶ Berlusconi, G., Calderoni, F., Parolini, N., Verani, M., and Piccardi, C. (2016), Link prediction in criminal
⁸¹⁷ networks: A tool for criminal intelligence analysis, *PloS One*, 11, e0154244.
- ⁸¹⁸ Binder, D. A. (1978), Bayesian cluster analysis, *Biometrika*, 65, 31–38.
- ⁸¹⁹ Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008), Fast unfolding of communities in large
⁸²⁰ networks, *Journal of Statistical Mechanics: Theory and Experiment*, 2008, P10008.
- ⁸²¹ Bouchard, M., and Malm, A. (2016), Social network analysis and its contribution to research on crime and
⁸²² criminal justice, *Oxford Handbook Topics in Criminology and Criminal Justice*, 5, 1–16.
- ⁸²³ Bright, D., Brewer, R., and Morselli, C. (2021), Using social network analysis to study crime: Navigating the
⁸²⁴ challenges of criminal justice records, *Social Networks*, 66, 50–64.
- ⁸²⁵ Bright, D., Koskinen, J., and Malm, A. (2019), Illicit network dynamics: The formation and evolution of a drug
⁸²⁶ trafficking network, *Journal of Quantitative Criminology*, 35, 237–258.
- ⁸²⁷ Calderoni, F. (2012), The structure of drug trafficking mafias: the 'Ndrangheta and cocaine, *Crime, Law and
⁸²⁸ Social Change*, 58, 321–349.
- ⁸²⁹ Calderoni, F., Brunetto, D., and Piccardi, C. (2017), Communities in criminal networks: A case study, *Social
⁸³⁰ Networks*, 48, 116–125.
- ⁸³¹ Calderoni, F., Catanese, S., De Meo, P., Ficara, A., and Fiumara, G. (2020), Robust link prediction in criminal
⁸³² networks: A case study of the Sicilian Mafia, *Expert Systems with Applications*, 161, 113666.
- ⁸³³ Calderoni, F., and Superchi, E. (2019), The nature of organized crime leadership: Criminal leaders in meeting
⁸³⁴ and wiretap networks, *Crime, Law and Social Change*, 72, 419–444.
- ⁸³⁵ Campana, P. (2016), Explaining criminal networks: Strategies and potential pitfalls, *Methodological Innovations*,
⁸³⁶ 9, 1–10.
- ⁸³⁷ Campana, P., and Varese, F. (2013), Cooperation in criminal organizations: Kinship and violence as credible
⁸³⁸ commitments, *Rationality and Society*, 25, 263–289.
- ⁸³⁹ Campana, P.— (2022), Studying organized crime networks: Data sources, boundaries and the limits of structural
⁸⁴⁰ measures, *Social Networks*, 69, 149–159.
- ⁸⁴¹ Catino, M. (2014), How do Mafias organize?: Conflict and violence in three Mafia organizations, *European
⁸⁴² Journal of Sociology/Archives Européennes de Sociologie*, 55, 177–220.
- ⁸⁴³ — (2015), Mafia rules. The role of criminal codes in mafia organizations, *Scandinavian Journal of Management*,
⁸⁴⁴ 31, 536–548.
- ⁸⁴⁵ Cavallaro, L., Ficara, A., De Meo, P., Fiumara, G., Catanese, S., Bagdasar, O., Song, W., and Liotta, A. (2020),
⁸⁴⁶ Disrupting resilient criminal networks through data analysis: The case of Sicilian Mafia, *Plos one*, 15, e0236476.
- ⁸⁴⁷ Charette, Y., and Papachristos, A. V. (2017), The network dynamics of co-offending careers, *Social Networks*,
⁸⁴⁸ 51, 3–13.
- ⁸⁴⁹ Chatterjee, S. (2015), Matrix estimation by universal singular value thresholding, *The Annals of Statistics*, 43,
⁸⁵⁰ 177–214.

- 851 De Blasi, P., Favaro, S., Lijoi, A., Mena, R. H., Prünster, I., and Ruggiero, M. (2015), Are Gibbs-type priors
 852 the most natural generalization of the Dirichlet process? *IEEE Transactions on Pattern Analysis and Machine*
 853 *Intelligence*, 37, 212–229.
- 854 Diviák, T. (2022), Key aspects of covert networks data collection: Problems, challenges, and opportunities, *Social*
 855 *Networks*, 69, 160–169.
- 856 Diviák, T., Dijkstra, J. K., and Snijders, T. A. (2019), Structure, multiplexity, and centrality in a corruption
 857 network: the Czech Rath affair, *Trends in Organized Crime*, 22, 274–297.
- 858 Durante, D., and Dunson, D. B. (2014), Nonparametric Bayes dynamic modelling of relational data, *Biometrika*,
 859 101, 883–898.
- 860 Durante, D., Mukherjee, N., and Steorts, R. C. (2017), Bayesian learning of dynamic multilayer networks,
 861 *Journal of Machine Learning Research*, 18, 1–29.
- 862 Faust, K., and Tita, G. E. (2019), Social networks and crime: Pitfalls and promises for advancing the field,
 863 *Annual Review of Criminology*, 2, 99–122.
- 864 Ficara, A., Cavallaro, L., Curreri, F., Fiumara, G., De Meo, P., Bagdasar, O., Song, W., and Liotta, A. (2021),
 865 Criminal networks analysis in missing data scenarios through graph distances, *PLoS one*, 16, e0255067.
- 866 Frank, O., and Strauss, D. (1986), Markov graphs, *Journal of the American Statistical Association*, 81, 832–842.
- 867 Fruchterman, T. M., and Reingold, E. M. (1991), Graph drawing by force-directed placement, *Software: Practice*
 868 *and Experience*, 21, 1129–1164.
- 869 Geng, J., Bhattacharya, A., and Pati, D. (2019), Probabilistic community detection with unknown number of
 870 communities, *Journal of the American Statistical Association*, 114, 893–905.
- 871 Ghosh, S. K., Mukhopadhyay, P., and Lu, J.-C. J. (2006), Bayesian analysis of zero-inflated regression models,
 872 *Journal of Statistical Planning and Inference*, 136, 1360–1375.
- 873 Girvan, M., and Newman, M. E. (2002), Community structure in social and biological networks, *Proceedings of*
 874 *the National Academy of Sciences*, 99, 7821–7826.
- 875 Gnedin, A. (2010), Species sampling model with finitely many types, *Electronic Communications in Probability*,
 876 15, 79–88.
- 877 Gollini, I., Caimo, A., and Campana, P. (2020), Modelling interactions among offenders: A latent space approach
 878 for interdependent ego-networks, *Social Networks*, 63, 134–149.
- 879 Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983), Stochastic blockmodels: First steps, *Social Networks*,
 880 5, 109–137.
- 881 Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., and Ueda, N. (2006), Learning systems of concepts
 882 with an infinite relational model, in *Proceedings of the 21st National Conference on Artificial Intelligence -*
 883 *Volume 1*, pp. 381–388.
- 884 Klerks, P. (2001), The network paradigm applied to criminal organisations: Theoretical nitpicking or a relevant
 885 doctrine for investigators? Recent developments in the Netherlands, *Connections*, 24, 53–65.
- 886 Lambert, D. (1992), Zero-inflated Poisson regression, with an application to defects in manufacturing,
 887 *Technometrics*, 34, 1–14.
- 888 Le, C. M., Levin, K., and Levina, E. (2018), Estimating a network from multiple noisy realizations, *Electronic*
 889 *Journal of Statistics*, 12, 4697–4740.
- 890 Legramanti, S., Rigon, T., Durante, D., and Dunson, D. B. (2022), Extended stochastic block models with
 891 application to criminal networks, *The Annals of Applied Statistics*, 16, 2369–2395.
- 892 Li, C.-S. (2012), Identifiability of zero-inflated Poisson models, *Brazilian Journal of Probability and Statistics*,
 893 26, 306–312.
- 894 Lindquist, M. J., and Zenou, Y. (2019), Crime and networks: ten policy lessons, *Oxford Review of Economic*
 895 *Policy*, 35, 746–771.
- 896 Malm, A., Bouchard, M., Decorte, T., Vlaemynck, M., and Wouters, M. (2017), More structural holes, more
 897 risk? Network structure and risk perception among marijuana growers, *Social Networks*, 51, 127–134.
- 898 Mantziou, A., Lunagómez, S., and Mitra, R. (2024), Bayesian model-based clustering for populations of network
 899 data, *The Annals of Applied Statistics*, 18, 266–302.
- 900 Mariadassou, M., and Matias, C. (2015), Convergence of the groups posterior distribution in latent or stochastic
 901 block models, *Bernoulli*, 21, 537–573.
- 902 McDaid, A., Murphy, T., Friel, N., and Hurley, N. (2013), Improved Bayesian inference for the stochastic block
 903 model with application to large networks, *Computational Statistics & Data Analysis*, 60, 12–31.
- 904 Meilă, M. (2007), Comparing clusterings — an information based distance, *Journal of Multivariate Analysis*, 98,
 905 873–895.
- 906 Morselli, C. (2009), *Inside Criminal Networks*, Springer.

- 907 Morselli, C., Giguère, C., and Petit, K. (2007), The efficiency/security trade-off in criminal networks, *Social
908 Networks*, 29, 143–153.
- 909 Newman, M. E. (2006), Modularity and community structure in networks, *Proceedings of the National Academy
910 of Sciences*, 103, 8577–8582.
- 911 Newman, M. E., Cantwell, G. T., and Young, J. G. (2020), Improved mutual information measure for clustering,
912 classification, and community detection, *Physical Review E*, 104, 042304.
- 913 Ng, T. L. J., and Murphy, T. B. (2021), Weighted stochastic block model, *Statistical Methods & Applications*,
914 30, 1365–1398.
- 915 Nowicki, K., and Snijders, T. A. B. (2001), Estimation and prediction for stochastic blockstructures, *Journal of
916 the American Statistical Association*, 96, 1077–1087.
- 917 Paoli, L. (2007), Mafia and organised crime in Italy: the unacknowledged successes of law enforcement, *West
918 European Politics*, 30, 854–880.
- 919 Papachristos, A. V. (2014), The network structure of crime, *Sociology Compass*, 8, 347–357.
- 920 Priebe, C. E., Sussman, D. L., Tang, M., and Vogelstein, J. T. (2015), Statistical inference on errorfully observed
921 graphs, *Journal of Computational and Graphical Statistics*, 24, 930–953.
- 922 Schmidt, M. N., and Morup, M. (2013), Nonparametric Bayesian modeling of complex networks: An introduction,
923 *IEEE Signal Processing Magazine*, 30, 110–128.
- 924 Sergi, A., and Lavorgna, A. (2016), *'Ndrangheta: The Glocal Dimensions of the Most Powerful Italian Mafia*,
925 Springer.
- 926 Sparrow, M. K. (1991), The application of network analysis to criminal intelligence: An assessment of the
927 prospects, *Social Networks*, 13, 251–274.
- 928 Stephens, M. (2000), Dealing with label switching in mixture models, *Journal of the Royal Statistical Society:
929 Series B (Statistical Methodology)*, 62, 795–809.
- 930 Van Dyk, D. A., and Park, T. (2008), Partially collapsed Gibbs samplers: Theory and methods, *Journal of the
931 American Statistical Association*, 103, 790–796.
- 932 Von Lampe, K. (2006), The interdisciplinary dimensions of the study of organized crime, *Trends in Organized
933 Crime*, 9, 77–95.
- 934 Von Luxburg, U. (2007), A tutorial on spectral clustering, *Statistics and Computing*, 17, 395–416.
- 935 Wade, S., and Ghahramani, Z. (2018), Bayesian cluster analysis: Point estimation and credible balls, *Bayesian
936 Analysis*, 13, 559–626.
- 937 Young, J. G., Cantwell, G. T., and Newman, M. E. J. (2020), Bayesian inference of network structure from
938 unreliable data, *Journal of Complex Networks*, 8, cnaa046.

“Bayesian nonparametric stochastic block modeling of criminal networks” useful additional references

1. Calderoni, F., Brunetto, D., & Piccardi, C. (2017). Communities in criminal networks: A case study. *Social Networks*, 48, 116-125.
2. Camerlenghi, F., Lijoi, A., Orbanz, P., and Prünster, I. (2019), Distribution theory for hierarchical processes. *The Annals of Statistics*, 47, 67–92.
3. De Blasi, P., Favaro, S., Lijoi, A., Mena, R.H., Pruenster, I., & Ruggiero, M. (2015). Are Gibbs-type priors the most natural generalization of the Dirichlet process? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 212–229.
4. Mueller, P., Quintana, F., & Rosner, G. L. (2011). A product partition model with regression on covariates. *Journal of Computational and Graphical Statistics*, 20, 260-278.
5. Nowicki, K., & Snijders, T. A. B. (2001). Estimation and prediction for stochastic block-structures. *Journal of the American Statistical Association*, 96, 1077–1087.

Laura Anderlucci

Material list:

Julien Jaques

Material list:

Jean Steve Tamo Tchomgui, Julien Jacques, Vincent Barriac, Guillaume Fraysse, Stéphane Chrétien (2023) A Penalized Spline Estimator for Functional Linear Regression with Functional Response. hal-04120709v2

Jean Steve Tamo Tchomgui, Julien Jacques, Guillaume Fraysse, Vincent Barriac, Stéphane Chrétien (2024) A mixture of experts regression model for functional response with functional covariates. PREPRINT (Version 1) available at Research Square <https://doi.org/10.21203/rs.3.rs-4142146/v1>



A Penalized Spline Estimator for Functional Linear Regression with Functional Response

Jean Steve Tamo Tchomgui, Julien Jacques, Vincent Barriac, Guillaume Fraysse, Stéphane Chrétien

► To cite this version:

Jean Steve Tamo Tchomgui, Julien Jacques, Vincent Barriac, Guillaume Fraysse, Stéphane Chrétien. A Penalized Spline Estimator for Functional Linear Regression with Functional Response. 2023. hal-04120709v2

HAL Id: hal-04120709

<https://hal.science/hal-04120709v2>

Preprint submitted on 19 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Penalized Spline Estimator for Functional Linear Regression with Functional Response

June 1, 2023

Jean Steve Tamo Tchomgui^{1,2} & Julien Jacques¹ & Vincent Barriac² & Guillaume Fraysse²
& Stéphane Chrétien¹

¹ Univ Lyon, Univ Lyon 2, ERIC, Lyon.

{jean-steve.tamo-tchomgui, julien.jacques, stephane.chretien}@univ-lyon2.fr

² Orange Innovation, France. {vincent.barriac, guillaume.fraysse}@orange.com

Abstract

Many scientific studies in recent years have been collecting data at a high frequency, which can be considered as functional data. When both the response variable to be modelled and the covariates are functions, we provide a novel and easy-to-implement method addressing function-on-function linear modelling and obtain interpretable parameters. Two main types of models are considered: the concurrent model which explains the response curve $Y_i(t)$ at time t from the values at same time t of the covariates $X_i^l(t)$; the (feed-forward) integral model which explains $Y_i(t)$ based on the values of covariate curves $X_i^l(s)$ observed at any times $s \leq t$. A regularized inference approach is proposed, which accurately selects an appropriate set of basis functions that can be used for functional data reconstruction and at the same time provides smooth and interpretable functional parameters. Numerical studies on simulated data with different scenarios illustrate the good performance our the method to capture the relationship between covariates and response. The method is finally applied to the well-known data in order to compare it to some existed competitors. On Canadian weather data with the problem of predicting precipitations from temperature measurements and on Hawaii Ocean data for predicting ocean salinity from temperature, oxygen, chloropigments and density measurements, our method made significant improvements on prediction error.

Keywords: Functional data analysis, function-on-function regression, penalized splines, Canadian weather data.

1 Introduction

In the last few decades, data sets collected with new, fast and sometimes accurate sensors have become very common in various fields of applied sciences including economics, finance, geosciences, medicine, etc. As a result, new tools are needed to process and analyze this very fast growing resource of data. A quite natural idea that has emerged lately is to extend classical tools from data analysis to a new paradigm called Functional Data Analysis (FDA). This new paradigm has proved very successful at addressing the statistical analysis of data where at least one of the variables of interest need to be treated as a function. Extension of linear regression to the functional setting has therefore naturally become a major area of research in FDA. While the literature is too vast to cover here, the recommended references for this field are Ramsay and Silverman (2005), Ramsay et al. (2009), Horváth and Kokoszka (2012), Kokoszka and Reimherr (2017), which provide excellent introductions to FDA. Moreover Goldsmith et al. (2011) and Morris (2014) provide a broad overview of the methods of functional linear regression. In the functional setting, different types of functional linear regression have been considered, depending on the functional nature of the response and/or at least one of the covariates. Thus, using the convention that first term denotes response-type and second term denotes covariate-type, the following regression models are all the possible options to consider: function-on-scalar, scalar-on-function and function-on-function. The scalar-on-function linear regression models is the most thoroughly studied model among the three models in the current literature. Some references include Cardot et al. (1999) and Hastie and Tibshirani (1993).

Most of the inference approaches for these models rely on a basis expansion assumption. For instance Besse and Cardot (1996) and Ramsay and Silverman (2005) proposed spline-type approximations of the functional covariates and then performed the estimation step by minimizing a least squares criterion. Among other useful references, Antoch et al. (2010) uses B-spline expansions for both the functional parameters and the functional covariates. The issue of possible non-identifiability was pursued in Scheipl and Greven (2016). In these approaches, the functional regression models become equivalent to a multivariate model on the basis expansion coefficients. An alternative way is to consider Functional Principal Components Analysis (FPCA, Ramsay and Silverman (2005)), possibly using smoothness promoting penalization (Silverman, 1996; Besse et al., 1997). Possible issues in determining the number of components to account for that seem

to be still open. Indeed, it was shown in Crainiceanu et al. (2009) that the shape of the functional parameters can drastically change as one or two additional principal components are included, making the process quite unstable and relatively difficult to interpret.

In comparison with scalar-on-function problems, function-on-function models, that we address here, have been much less studied in the literature. For instance, Ivanescu et al. (2015) proposes to estimate a function-on-function regression model using a penalized mixed model. In this setting as well, the main issue faced is not only the problem of accurately selecting the number of basis functions and the location of the knots (Li and Ruppert, 2008), but also the possible interpretability of the obtained estimators (James et al., 2009). Signal compression approach (*wSigcomp*) designed by Luo et al. (2016) which is another way to address function-on-function models firstly apply wavelets transformation to covariates and with the functional response and the obtained multivariate covariates, proposed a method to estimate the functional bivariate parameter by characterize it as the solution of a generalized functional eigenvalue problem. The Optimal Penalized Function-on-Function Regression (OPFFR) proposed by (Sun et al., 2018), produce an estimator of the 2D functional parameter as optimizer of a form of penalized least squares where the penalty enforces a certain level of smoothness.

In mathematical terms, the problem considered in the present paper is the one of estimating a linear relationship between functional covariates and functional response based on the n -sample

$$\left\{ Y_i(t), X_i(t) = \left(X_i^1(t), \dots, X_i^p(t) \right)^\top, t \in [0, T] \right\}$$

$i = 1, \dots, n$, where the output variable $Y(t)$ and the p input variables $(X^l(t))_{1 \leq l \leq p}$ are assumed to belong to the separable Hilbert $L^2([0; T])$. In the sequel, we focus in particular on the following two functional linear models:

$$Y_i(t) = \beta_0(t) + \sum_{l=1}^p \beta_l(t) X_i^l(t) + \varepsilon_i(t) = (1, X_i(t))^\top \beta(t) + \varepsilon_i(t), \quad (1)$$

$$Y_i(t) = \gamma_0(t) + \sum_{l=1}^p \int_0^t \gamma_l(s, t) X_i^l(s) ds + \varepsilon_i(t) = \gamma_0(t) + \int_0^t X_i(s)^\top \gamma(s, t) ds + \varepsilon_i(t) \quad (2)$$

where $\beta(t) = (\beta_0(t), \beta_1(t), \dots, \beta_p(t))^\top$, $\gamma(s, t) = (\gamma_0(t), \gamma_1(s, t), \gamma_2(s, t), \dots, \gamma_p(s, t))^\top$ are the unknown functional parameters and are assumed to be square integrable; $\varepsilon_i(t)$ is the model error and is a sample of centered random variables with variance σ_i^2 , specific to the i^{th} individual

(Ramsay and Silverman (2005), Chapter 13); $\varepsilon_i(t)$ and $X_i(t)$ are assumed to be uncorrelated. The noise functions $\varepsilon_i(t)$ can be rigorously defined using white noise theory as presented in Hida et al. (1993). In our context, we will only use the fact that when sampled at various times from a finite set \mathcal{T} , the vector $(\varepsilon_i(t))_{t \in \mathcal{T}}$ can be expressed as a sum of a vector with i.i.d. components and a vector with prescribed covariance matrix, i.e. a vector with constant components in the simplest case.

Model (1), known as the “concurrent model”, assumes that the response function at time t , $Y_i(t)$, is explained by covariate functions $X_i^l(t)$, at exactly the same time t , the functional parameters being allowed to vary with t as well. The second model (2), called the “integral model”, represents $Y_i(t)$ using the values of the covariates curves $X_i^l(s)$ for all the observed times $s \leq t$. Clearly, Model (2) is more general and richer than Model (1). Exploring the “concurrent model” further at the first step is of great interest because, as mentioned in Hastie and Tibshirani (1993), any functional linear model can be reduced to this form.

In the present paper, we develop an efficient approach for estimating the functional parameters $\beta(t)$ of the concurrent model (1) and $\gamma(s, t)$ of the integral model (2). For this purpose, we use cubic B-spline basis expansion for both functional covariates and functional parameters. We propose penalized estimator of the corresponding functional basis coefficients. As will be shown in the sequel, our approach allows to simply choose equispaced knots and a sufficient number of basis functions to capture the main features of the covariates. Overfitting will be naturally avoided by penalizing roughness via controlling the second derivatives of the functional parameters which are being maximized.

Plan of the paper. The paper is organized as follows: Section 2 presents the two types of models we focus on and Section 3 the estimation scheme. Our estimation scheme consists of two steps. The first one addresses recovering of the functional nature of the covariates, by approximating them into a functional basis. The second step consists of penalized estimation of the functional regression coefficients, which are themselves decomposed in another functional basis. Section 4 contains a simulation based exploration of the method which confirms the efficiency of the proposed approach. Section 5 finally presents an illustration of the method on two real data sets. The first one is the well-known Canadian weather data set, in which the goal is to explain the precipitation as a function of the temperatures in different Canadian cities. The second one is

the Hawaii ocean data set in which salinity is explained as a function of four functional covariates. Finally, Section 6 concludes the paper.

2 Linear models for function-on-function regression

In this section, it is shown how the functional models (1) and (2) can be, under the basis expansion assumption of covariates and parameters, reduce to a linear mixed model onto the discrete observations of the functional response and functional covariates.

2.1 Functional concurrent model

Linear regression for a functional response involving one or more functional covariates in the concurrent model is a well-known problem. The main issue is to estimate an infinite dimensional parameter $\beta(t)$ through a finite sample of observations. As shown in Hastie and Tibshirani (1993), Model (1), also called the varying coefficient model, is interesting because any functional model can be reduced to this form. Chapter 14 in Ramsay and Silverman (2005) describes how this model can be fitted by minimizing an unweighted least squares criterion. The method proposed in this paper addresses the estimation problem using a penalized function-on-function regression as proposed in Ivanescu et al. (2015), where the problem is represented as a mixed model. Nevertheless, our work differs by the choice of the penalization criterion enforced on the functional parameter. The parameter $\beta(t)$ is expanded in functional basis using q_β basis functions to get back to a classical mixed model for which the estimations of the parameters are well known. Furthermore, we allow to choose the number of basis functions q_β to be large enough to capture any desired variations of $\beta(t)$, and we add a roughness penalty term to get a smooth solution for the parameter at the end. As a first step of our modelling, we recover the underlying functional process, by using penalized cubic B-splines expansion for all the functional covariates.

2.1.1 Functional basis expansion of covariates and model parameters

In practice, we do not properly observe a continuous curve for each realization of both the response variable $Y_i(t)$ and the covariate variables $(X_i^l(t))_{1 \leq l \leq p}$. In indeed, as opposed to the ideal observation setting, we only have access to a set of noisy observations at a finite number of points

on a grid. As a result, the functional data can be presented as a numerical vector. In order to recover the continuous form, which generally belongs to an infinite dimensional space (e.g. Hilbert separable space $L^2([0, T])$), one efficient way to proceed is by expanding the considered functions in a functional basis. The functional response, which is assumed in model (1) even in model (2) to be written as a linear combination of these predictors, is not necessary to be pre-processed. The advantage of this approach is the fact that by truncating the series at a given level q_l , we obtain an approximation of the covariate function $X_i^l(t)$ in a q_l dimensional space.

So for all the p covariates $X^l(t)$, we can therefore recover a representation in cubic B-splines functional basis. As indicated by Li and Ruppert (2008), the choice of the number of knots depends on the complexity of the variable and should be large enough to capture the patterns of the variable. It is reasonable to suppose that this number and, thus, the number of basis functions depends on the covariate. So to distinguish the basis functions of each covariate, although they just differ by their number, we will adopt in the rest of this article the system $\{B_1^l(t), B_2^l(t), \dots, B_{q_{X^l}}^l(t)\}$ as the basis function of $X^l(t)$. Then, any functional covariate can be written as:

$$X_i^l(t) = \sum_{j=1}^{q_{X^l}} x_{ij}^l B_j^l(t) = B^l(t)^\top x_i^l \quad \text{with } 1 \leq l \leq p. \quad (3)$$

The basis functions $B_j^l(t)$ being prescribed, the estimation of coefficients x_{ij}^l is done as a preliminary step (Li and Ruppert, 2008; Ruppert, 2002; Ramsay and Silverman, 2005).

Similarly as for functional covariates, we expand all the functional parameters $(\beta_l(t))_l$ of the concurrent model in functional basis. The number of basis functions q_{β^l} must be chosen as sufficiently large to capture the patterns of any $\beta_l(t)$:

$$\beta_l(t) = \sum_{j=1}^{q_{\beta^l}} b_j^l \phi_j^l(t) = \phi^l(t)^\top b^l \quad \text{with } 0 \leq l \leq p. \quad (4)$$

Using the expressions (3) and (4), the components in Model (1) become:

$$\beta(t) = \begin{pmatrix} \beta_0(t) \\ \beta_1(t) \\ \vdots \\ \beta_p(t) \end{pmatrix} = \underbrace{\begin{pmatrix} \phi^0(t)^\top b^0 \\ \phi^1(t)^\top b^1 \\ \vdots \\ \phi^p(t)^\top b^p \end{pmatrix}}_{(p+1, \sum_l q_{\beta^l}) - \text{matrix}} = \underbrace{\begin{pmatrix} \phi^0(t)^\top & 0 & \dots & 0 \\ 0 & \phi^1(t)^\top & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \phi^p(t)^\top \end{pmatrix}}_{\sum_l q_{\beta^l} - \text{vect.}} \underbrace{\begin{pmatrix} b^0 \\ b^1 \\ \vdots \\ b^p \end{pmatrix}}_{\sum_l q_{\beta^l} - \text{vect.}} = \Phi(t) b,$$

and

$$\mathbf{X}_i(t) = \begin{pmatrix} 1 \\ \mathbf{X}_i^1(t) \\ \vdots \\ \mathbf{X}_i^p(t) \end{pmatrix} = \begin{pmatrix} 1 \\ \mathbf{B}^1(t)^\top x_i^1 \\ \vdots \\ \mathbf{B}^p(t)^\top x_i^p \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \mathbf{B}^1(t)^\top & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{B}^p(t)^\top \end{pmatrix}}_{(p+1, \sum_l q_{\mathbf{X}^l}) - \text{matrix}} \underbrace{\begin{pmatrix} 1 \\ x_i^1 \\ \vdots \\ x_i^p \end{pmatrix}}_{\sum_l q_{\mathbf{X}^l} - \text{vect.}} = \mathbf{B}(t) x_i.$$

By plugging-in these expressions into Model (1), we get:

$$\mathbf{Y}_i(t) = x_i^\top \mathbf{B}(t)^\top \Phi(t) b + \varepsilon_i(t) = \mathbf{R}_i(t)^\top b + \varepsilon_i(t) \quad (5)$$

with $\mathbf{R}_i(t) = \Phi(t)^\top \mathbf{B}(t) x_i$ which is used as design matrix and b the unknown parameters to be estimated.

2.1.2 Functional concurrent model on the observations

The concurrent model implicitly assumes that the functional covariates and the functional response are observed at the same timestamps. The observation grid will consist of m points $\{t_1, \dots, t_m\}$. In mathematical terms we have:

$$\mathbf{Y}_i(t_j) = \mathbf{R}_i(t_j)^\top b + \varepsilon_i(t_j) \quad \text{with } 1 \leq i \leq n \text{ and } 1 \leq j \leq m. \quad (6)$$

One very specific issue to take care of is that the successive values of the observation noise $\varepsilon_i(t_1), \dots, \varepsilon_i(t_m)$ can not be assumed independent.

One way to address the question of dependency is to use a linear mixed model (LMM, Wood (2006)). We thus assume that the model error can be decomposed as $\varepsilon_i(t_j) = \mathbf{U}_i + \eta_{ij}$, with η_{ij} a Gaussian white noise and \mathbf{U}_i a random variable which takes into account the random effect in each individual $i = 1, \dots, n$. To summarize, our model consists of a LMM with fixed effects b and random effect \mathbf{U}_i . In matrix form we get:

$$\mathbf{Y} = \mathbf{R}^\top b + \mathbf{Z} \mathbf{U} + \boldsymbol{\eta}, \quad (7)$$

where $\mathbf{Y} = (\mathbf{Y}_1(t_1), \dots, \mathbf{Y}_1(t_m), \mathbf{Y}_2(t_1), \dots, \mathbf{Y}_n(t_m))^\top$, $\mathbf{R} = (\mathbf{R}_i(t_j))_{i,j}$ the design matrix of dimension $q_\beta \times nm$ with $q_\beta = \sum_l q_{\beta^l}$, $\mathbf{U} = (\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_n)^\top \sim \mathcal{N}(\mathbf{0}, \Gamma)$, $\boldsymbol{\eta} = (\eta_{ij})_{i,j} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_{nm})$

and

$$Z = \underbrace{\begin{pmatrix} 1_{m \times 1} & 0_{m \times 1} & \dots & 0_{m \times 1} \\ 0_{m \times 1} & 1_{m \times 1} & \dots & 0_{m \times 1} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{m \times 1} & 0_{m \times 1} & \dots & 1_{m \times 1} \end{pmatrix}}_{(nm \times n) - \text{matrix}}.$$

The specific notations we used are the matrices $0_{k \times l}$ and $1_{k \times l}$ of size $k \times l$, which are composed of zeros and ones, respectively; The notations $\mathbf{0}$ refers to the corresponding null vector and Γ the unknown covariance matrix of the random effects.

The parameters are then the fixed effects vectors b and the variance components σ^2 and Γ . We describe how to perform the inference in Section 3.

2.2 Functional integral model

The integral Model (2) assumes cumulative effects of covariates. More clearly, the model we proposes use observations of covariates up until time t to predict the response at time t . It is important to note that in most models found in the literature (Ramsay and Silverman, 2005; Horváth and Kokoszka, 2012), when both covariates and response have the same domain, consider that the response at any time t depends on the influence of the covariates on the whole domain. Such model implicitly assumes that the covariates at any time $t + s$ can influence the response variable at time t . However, in the integral model, the functional parameters are bivariate functions $\gamma_l(s, t)$, except for the constant of the model, which remains univariate. In this section, we start by expanding the parameters in a finite-dimensional functional basis and then plug this expression into the model.

2.2.1 Functional basis expansion of covariates and model parameters

The functional parameters are therefore expanded in a bivariate basis which may or may not have the same number of basis functions on each of the two dimensions. Without loss of generality and for the sake of simplicity, we assume that the number of basis functions is the same in the two

dimensions. This leads to the following expression:

$$\gamma_l(t, s) = \sum_{j,k=1}^{q_{\gamma^l}} a_{jk}^l B_{1j}^l(t) B_{2k}^l(s) \quad (8)$$

where $\{B_{1j}^l(t)\}_{1 \leq j \leq q_{\gamma^l}}$ and $\{B_{2j}^l(t)\}_{1 \leq j \leq q_{\gamma^l}}$ are the basis functions and $(a_{jk}^l)_{1 \leq j,k \leq q_{\gamma^l}}$ the unknown basis coefficients to be estimated. We can rewrite this expression in matrix form by:

$$\gamma_l(t, s) = a^{l^\top} \mathbf{B}_1^l(t) \mathbf{B}_2^l(s) \quad (9)$$

with

$$\begin{aligned} a^l &= \left(a_{11}^l \ \dots \ a_{1q_{\gamma^l}}^l \ a_{21}^l \ \dots \ \dots \ a_{q_{\gamma^l} 1}^l \ \dots \ a_{q_{\gamma^l} q_{\gamma^l}}^l \right)^\top, \\ \mathbf{B}_1^l(t) &= \text{diag} \left(B_{11}^l(t), \dots, B_{1q_{\gamma^l}}^l(t), \dots, \dots, B_{11}^l(t), \dots, B_{1q_{\gamma^l}}^l(t) \right), \\ \mathbf{B}_2^l(s) &= \left(B_{21}^l(s) \ \dots \ B_{2q_{\gamma^l}}^l(s) \ \dots \ \dots \ B_{2q_{\gamma^l}}^l(s) \ \dots \ B_{2q_{\gamma^l}}^l(s) \right)^\top. \end{aligned}$$

The functional constant being univariate, it can thus be written as in (4) in the form:

$$\gamma_0(t) = \sum_{j=1}^{q_{\gamma^0}} a_j^0 B_j^0(t) = \mathbf{B}^0(t)^\top a^0.$$

2.2.2 Functional integral model on the observations

By plugging covariates and parameters functional basis expansion in the integral Model (2), we get:

$$\begin{aligned} Y_i(t) &= \gamma_0(t) + \sum_{l=1}^p \int_0^t x_i^{l^\top} \mathbf{B}^l(s) \mathbf{B}_2^l(s)^\top \mathbf{B}_1^l(t)^\top a^l ds + \varepsilon_i(t) \\ &= \gamma_0(t) + \sum_{l=1}^p x_i^{l^\top} \underbrace{\left(\int_0^t \mathbf{B}^l(s) \mathbf{B}_2^l(s)^\top ds \right)}_{\mathbf{B}_2^l(t)} \mathbf{B}_1^l(t)^\top a^l + \varepsilon_i(t) \\ &= \gamma_0(t) + \sum_{l=1}^p x_i^{l^\top} \mathbf{B}_2^l(t) \mathbf{B}_1^l(t)^\top a^l + \varepsilon_i(t) \\ &= \mathbf{B}^0(t)^\top a^0 + \sum_{l=1}^p Q_i^l(t)^\top a^l + \varepsilon_i(t), \end{aligned}$$

with $Q_i^l(t) = \mathbf{B}_1^l(t)^\top \mathbf{B}_2^l(t)^\top x_i^l$. Finally we obtain:

$$Y_i(t) = Q_i(t)^\top a + \varepsilon_i(t) \quad (10)$$

with $a = (a^0, a^1, a^2, \dots, a^p)^\top$ and $Q_i(t) = \left(B_0(t)^\top, Q_i^1(t)^\top, Q_i^2(t)^\top, \dots, Q_i^p(t)^\top \right)^\top$ two vectors of length $q_\gamma = q_{\gamma,0} + \sum_{l=1}^p q_{\gamma,l}^2$.

Once again, we are faced with the problem of lack of independence of the different measured values for the same individual. We will proceed exactly in the same way as with the concurrent model using a linear mixed model with fixed effects given by the vector a and random effects given by the random vector $U = (U_i)_i$. The model will therefore be written as a LMM given by:

$$Y = Q^\top a + Z U + \eta, \quad (11)$$

with Z , U and η define similarly to (7). $Q = \left(Q_i(t_j) \right)_{i,j}$ the design matrix of dimension $q_\gamma \times nm$. As in the concurrent model, the parameters we need to estimate are the fixed effects vectors a and the variance components σ^2 and Γ . The inference scheme is described in Section 3.

3 B-spline-based penalized estimator

In both the concurrent and the integral models presented in Section 2.1 and Section 2.2 respectively, we have used the decomposition of the infinite-dimensional functional covariates and parameters into a truncated functional basis depending on the chosen number of basis functions. These values naturally needed to be correctly selected in order to avoid over- or under-fitting. Nevertheless, precise adjustment of these values often induces a high computational effort. In the case of the B-spline basis, even more parameters have to be properly tuned such as the choice of the spline order and the location of the knots. In order to reduce the expected cost of such a computationally demanding procedure, we made the choice of choosing a sufficiently large a priori value for q_β (or q_γ) and then apply a roughness penalty. This approach brings the benefit of reducing the overall computational cost, and of possibly improving the interpretability of the estimated functional coefficients. This last point is very interesting in the case of the linear model because as we already know, the interpretation of the predictors-response relationship becomes more difficult as the shape of the functional parameter β (or γ) does not have any simple structure.

Various approaches to regularize the parameter shape have been proposed in the literature. In our setting of interest, the main idea is oftentimes to enhance the model performance and interpretability by adding a roughness penalty. Leurgans et al. (1993) is among the first to

explore the functional penalization and show that the obtained estimator $\hat{\beta}(t)$ (resp. $\hat{\gamma}(s, t)$) becomes less sensitive to the rather subjective choice of the number of basis functions q_β (resp. q_γ). More recently, James et al. (2009) proposed a method called Functional Linear Regression That is Interpretable (FLiRTI) which addresses the issue of choosing relevant penalties. Based on variable selection ideas such as the Lasso penalty, they produce accurate, flexible and highly interpretable estimates of the functional parameters. The main idea in James et al. (2009) is, instead of enforcing sparsity on the function themselves, to enforce sparsity of the derivatives instead. Using the notation $\beta^{(l)}(t)$ for the l^{th} derivative of $\beta(t)$, we may deduce that $\beta^{(0)}(t) = 0$ guarantees $X(t)$ has no effect on $Y(t)$ at t ; $\beta^{(1)}(t) = 0$ implies that $\beta(t)$ is constant at t ; $\beta^{(2)}(t) = 0$ means that $\beta(t)$ is linear at t and so on. The FLiRTI approach also combine sparsity enforcing penalties for more than one derivative at a time, which can be useful for smooth parameters that may even vanish on some intervals.

Instead of the Lasso penalty applied in the FLiRTI method, where choosing the derivatives remains a difficult computational issue, our approach uses a Ridge penalty on the second derivative of the functional parameters. The choice of penalizing the second derivative is mainly motivated by the desire to obtain a possibly locally linear relationship if needed. Moreover, the use the Ridge penalty is motivated by the lack of exact sparsity observed in real problems and the clear benefits of getting a closed form formula for the estimators.

3.1 Penalized estimator for the concurrent model

Let us first consider the concurrent model in the classical mixed model form as in (7). In order to obtain an interpretable estimator, we will use a roughness penalty in the form of a Ridge-type penalty on the second derivatives, as advocated for in the previous paragraph. The objective function is the penalized log-likelihood function given by

$$\mathcal{L}_{\text{pen}}(b, \Gamma) = -2 \mathcal{L}(b, \Gamma | Y) + \sum_{l=0}^p \text{Pen}(\beta_l); \quad (12)$$

with,

$$\mathcal{L}(b, \Gamma | Y) = nm \log(2\pi) + \log|V| + (Y - R^\top b)^\top V^{-1} (Y - R^\top b) \quad (13)$$

using that $V = \text{Var}(ZU + \eta)$ and the penalty

$$\begin{aligned} \text{Pen}(\beta_l) &= \lambda_l \int \beta_l''(t)^2 dt = \lambda_l \int \left[\sum_{j=1}^{q_{\beta^l}} b_j^l \phi_j^{l^r}(t) \right]^2 dt = \lambda_l \sum_{s,k=1}^{q_{\beta^l}} b_s^l b_k^l \Phi_{sk}^l, \\ \text{with } \Phi_{sk}^l &= \int \phi_s^{l^r}(t) \phi_k^{l^r}(t) dt. \end{aligned}$$

When λ_l is too large, the estimation of $\beta_l(\cdot)$ will be too smooth, and we will not be able to account for the possible variations of the regression coefficients. When, instead, λ_l is too small, the estimators might become too rough and overfitting might occur.

For a given value λ_l , the estimation of $(\beta_l(t))_{0 \leq l \leq p}$ is obtained by solving:

$$\begin{aligned} \min_{b, \Gamma} \mathcal{L}_{pen}(b, \Gamma) &= \min_{b, \Gamma} -2\mathcal{L}(b, \Gamma | Y) + \sum_{l=0}^p \lambda_l \sum_{s,k=1}^{q_{\beta^l}} b_s^l b_k^l \Phi_{sk}^l \\ &= \min_{b, \Gamma} -2\mathcal{L}(b, \Gamma | Y) + b^\top (\lambda P) b, \end{aligned} \quad (14)$$

where $\lambda P \in \mathbb{R}^{q_\beta \times q_\beta}$ is given by:

$$\lambda P = \begin{pmatrix} \lambda_0 \Psi^0 & 0_{q_{\beta^0} \times q_{\beta^1}} & \dots & 0_{q_{\beta^0} \times q_{\beta^p}} \\ 0_{q_{\beta^1} \times q_{\beta^0}} & \lambda_1 \Psi^1 & \dots & 0_{q_{\beta^1} \times q_{\beta^p}} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{q_{\beta^p} \times q_{\beta^0}} & 0_{q_{\beta^p} \times q_{\beta^1}} & \dots & \lambda_p \Psi^p \end{pmatrix} \quad \text{with } \Psi^l = \begin{pmatrix} \Phi_{11}^l & \Phi_{12}^l & \dots & \Phi_{1q_{\beta^l}}^l \\ \Phi_{21}^l & \Phi_{22}^l & \dots & \Phi_{2q_{\beta^l}}^l \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{q_{\beta^l} 1}^l & \Phi_{q_{\beta^l} 2}^l & \dots & \Phi_{q_{\beta^l} q_{\beta^l}}^l \end{pmatrix}.$$

Here, $0_{q_1 \times q_2}$ is the standard notation for the null matrix of size $q_1 \times q_2$. As Ψ^l is a symmetric positive-definite matrix for any $0 \leq l \leq p$, we can easily find its Cholesky decomposition, which can be efficiently leveraged in the implementation.

We first rewrite Model (7) in the form :

$$Y = R^\top b + \varepsilon^*,$$

with $\varepsilon^* = ZU + \eta$ and $V = \text{Var}(\varepsilon^*) = Z\Gamma Z^\top + \sigma^2 I$. By setting the partial derivatives with respect to b and V to 0 and then solving the resulting linear system, we get:

$$\hat{b}_\lambda = \left(R^\top \hat{V}^{-1} R + \lambda P \right)^{-1} R^\top \hat{V}^{-1} Y. \quad (15)$$

(see Appendix 7.2 for more details).

Let us now address the problem of choosing the smoothing parameters $\lambda = (\lambda_l)_{0 \leq l \leq p}$. The correct choice will make great use of the observed accuracy of the prediction. For this purpose, for a fixed value of λ , we resort to a leave-one-out cross-validation type approach and compute $\widehat{b}_\lambda^{(-i)}$ based on the sample except for the i^{th} observation. We then compute the prediction $\widehat{Y}_\lambda^{(-i)}$ at observation i . Finally, we can compute the prediction error or cross-validation score associated with the parameter λ as

$$\mathcal{V}_n(\lambda) = \frac{1}{n} \sum_{i=1}^n \left(Y_i - \widehat{Y}_\lambda^{(-i)} \right)^2.$$

The value of λ that achieves the lowest estimated risk will be selected.

3.2 Penalized estimator for the integral model

For the integral model, we can also optimize the penalized log-likelihood function as in (12). However, the main difference lies in the specific form of the penalty. The log-likelihood of the model will thus have the expression:

$$\mathcal{L}(a, \Gamma | Y) = nm \log(2\pi) + \log |V| + (Y - Q^\top a)^\top V^{-1} (Y - Q^\top a) \quad (16)$$

using $V = \text{Var}(ZU + \eta)$, and the penalized log-likelihood:

$$\mathcal{L}_{\text{pen}}(a, \Gamma | Y) = -2\mathcal{L}(a, \Gamma | Y) + \text{Pen}(\gamma_0) + \sum_{l=1}^p \text{Pen}(\gamma_l); \quad (17)$$

For this model, the parameters γ_l will be bivariate functions except for γ_0 , which is univariate. The penalties for bivariate parameters will take the following expression:

$$\text{Pen}(\gamma_l) = \lambda_l \int \int \left\| \mathbf{H}_{\gamma_l}(t, s) \right\|^2 ds dt = \lambda_l \int \int \left\| \begin{bmatrix} \frac{\partial^2 \gamma_l(t, s)}{\partial t^2} & \frac{\partial^2 \gamma_l(t, s)}{\partial t \partial s} \\ \frac{\partial^2 \gamma_l(t, s)}{\partial s \partial t} & \frac{\partial^2 \gamma_l(t, s)}{\partial s^2} \end{bmatrix} \right\|^2 ds dt.$$

Here $\mathbf{H}_f(t, s)$ denotes the Hessian matrix of the bivariate function f and $\|\cdot\|$ is as standard the Frobenius norm. To simplify expressions we will use the notation: $\frac{\partial^2 \gamma_l(t, s)}{\partial t \partial s} \equiv \gamma_l^{ts}(t, s)$, and then we have

$$\text{Pen}(\gamma_l) = \lambda_l \int \int \left(\gamma_l^{tt}(t, s)^2 + 2\gamma_l^{ts}(t, s)^2 + \gamma_l^{ss}(t, s)^2 \right) ds dt. \quad (18)$$

We know from (9) that

$$\begin{aligned}\gamma_l(t, s)^2 &= \left(a^{l\top} \mathbf{B}_1(t) \mathbf{B}_2(s) \right)^2 = \left(\sum_{i,j=1}^{q_{\gamma^l}} a_{ij}^l \mathbf{B}_{1i}^l(t) \mathbf{B}_{2j}^l(s) \right)^2 \\ &= \sum_{i,j,k,m=1}^{q_{\gamma^l}} a_{ij}^l a_{km}^l \mathbf{B}_{1i}^l(t) \mathbf{B}_{1k}^l(t) \mathbf{B}_{2j}^l(s) \mathbf{B}_{2m}^l(s),\end{aligned}$$

so we then have the following expressions for the partial derivatives:

$$\left\{ \begin{array}{lcl} \int \int \gamma_l^{tt}(t, s)^2 ds dt &= \sum_{i,j,k,m=1}^{q_{\gamma^l}} a_{ij}^l a_{km}^l \int \mathbf{B}_{1i}^{l''}(t) \mathbf{B}_{1k}^{l''}(t) dt \int \mathbf{B}_{2j}^l(s) \mathbf{B}_{2m}^l(s) ds \\ &= \sum_{i,j,k,m=1}^{q_{\gamma^l}} a_{ij}^l a_{km}^l \Phi_{1,ik}^{l''} \Phi_{2,jm}^l; \\ \int \int \gamma_l^{ts}(t, s)^2 ds dt &= \sum_{i,j,k,m=1}^{q_{\gamma^l}} a_{ij}^l a_{km}^l \int \mathbf{B}_{1i}^{l'}(t) \mathbf{B}_{1k}^{l'}(t) dt \int \mathbf{B}_{2j}^{l'}(s) \mathbf{B}_{2m}^{l'}(s) ds \\ &= \sum_{i,j,k,m=1}^{q_{\gamma^l}} a_{ij}^l a_{km}^l \Phi_{1,ik}^{l'} \Phi_{2,jm}^{l'}; \\ \int \int \gamma_l^{ss}(t, s)^2 ds dt &= \sum_{i,j,k,m=1}^{q_{\gamma^l}} a_{ij}^l a_{km}^l \int \mathbf{B}_{1i}^l(t) \mathbf{B}_{1k}^l(t) dt \int \mathbf{B}_{2j}^{l''}(s) \mathbf{B}_{2m}^{l''}(s) ds \\ &= \sum_{i,j,k,m=1}^{q_{\gamma^l}} a_{ij}^l a_{km}^l \Phi_{1,ik}^l \Phi_{2,jm}^{l''}. \end{array} \right.$$

with the notation $\Phi_{u,sk}^{l'} = \int \mathbf{B}_{us}^{l'}(t) \mathbf{B}_{uk}^{l'}(t) dt$.

Back to the problem of minimizing the penalized log-likelihood (17), for fixed values $(\lambda_l)_{0 \leq l \leq p}$, the estimation of $(\gamma_0(t), \gamma_1(t, s), \dots, \gamma_p(t, s))$ is obtained by solving the problem:

$$\begin{aligned}\min_{a, \Gamma} \mathcal{L}_{pen}(a, \Gamma) &= \min_{a, \Gamma} -2 \mathcal{L}(a, \Gamma | Y) + \lambda_0 \sum_{s,k=1}^{q_{\gamma^0}} a_s^0 a_k^0 \Phi_{sk}^0 + \\ &\quad \sum_{l=1}^p \lambda_l \sum_{i,j,k,m=1}^{q_{\gamma^l}} a_{ij}^l a_{km}^l \left(\Phi_{1,ik}^{l''} \Phi_{2,jm}^l + 2 \Phi_{1,ik}^{l'} \Phi_{2,jm}^{l'} + \Phi_{1,ik}^l \Phi_{2,jm}^{l''} \right).\end{aligned}$$

The penalty term for any bivariate parameter $\gamma_l(t, s)$ can be seen as the tensor product between the three following terms: the $(a_{ij})_{1 \leq i,j \leq q_{\gamma^l}}$ matrix, the 4^{th} order square tensor of dimension q_{γ^l} and the matrix $(a_{km})_{1 \leq k,m \leq q_{\gamma^l}}$. We can rearrange this tensor product as a matrix product by flattening

the matrix to a vector and the 4th order tensor to a matrix. So we get a matrix product between the row vector of length $q_{\gamma^l}^2$, the square matrix of dimension $q_{\gamma^l}^2 \times q_{\gamma^l}^2$ and the column matrix of length q_{γ^l} . The minimization problem can be written in matrix form:

$$\min_{a, \Gamma} \mathcal{L}_{pen}(a, \Gamma) = \min_{a, \Gamma} -2 \mathcal{L}(a, \Gamma | Y) + a^\top (\lambda P) a, \quad (19)$$

where λP the matrix of dimension $q_\gamma \times q_\gamma$ with $q_\gamma = q_{\gamma^0} + \sum_{l=1}^p q_{\gamma^l}^2$ defined as in (14). The main difference lies in the expression of the block matrix Ψ^l for $l > 0$ given by:

$$\Psi^l = \left(\Phi_{1,ik}^{l''} \Phi_{2,jm}^l + 2 \Phi_{1,ik}^{l'} \Phi_{2,jm}^{l'} + \Phi_{1,ik}^l \Phi_{2,jm}^{l''} \right)_{1 \leq i,j,k,m \leq q_{\gamma^l}}.$$

With this expression, we proceed in the same way as in the concurrent model to obtain the penalized estimator of a and then $(\hat{\gamma}_l(t, s))_l$.

3.3 Prediction interval

We know, based on earlier works of Ruppert et al. (2003) and Wood (2006), that the variance of our estimators and joint and point wise confidence intervals in the mixed effects model can easily be obtained. The penalized estimator of fixed effects and variance components is given by:

$$\begin{cases} \hat{b}_\lambda &= (R^\top \hat{V}^{-1} R + \lambda P)^{-1} R^\top \hat{V}^{-1} Y, \\ \hat{U} &= \sigma^2 Z^\top \hat{V}^{-1} (Y - R^\top \hat{b}_\lambda); \end{cases}$$

where λ is the smoothing parameter. Since the above expressions depend on the variance components, they can only be calculated if they are known.

With a model of the form $Y = R^\top b + \varepsilon^*$ with $\varepsilon^* = ZU + \eta$, we have $\varepsilon^* \sim \mathcal{N}(0, V)$ with $V = Z\Gamma Z^\top + \sigma^2 \mathbb{I}_{nm}$ and then $Y \sim \mathcal{N}(R^\top b, V)$. Thus:

$$\begin{aligned} \text{Cov}(Y, U) &= \text{Cov}(R^\top b + ZU + \eta, U) \\ &= \text{Cov}(R^\top b, U) + Z \text{Var}(U) + \text{Cov}(\eta, U) \\ &= Z\Gamma \end{aligned}$$

We can thus deduce that $\mathbb{E}(U | Y) = \Gamma Z^\top V^{-1}(Y - R^\top b)$, and since $Y \sim \mathcal{N}(R^\top b, V)$ holds,

$$\begin{aligned}\text{Var}(\hat{b}_\lambda) &= \text{Var}\left(\left(R^\top \hat{V}^{-1} R + \lambda P\right)^{-1} R^\top \hat{V}^{-1} Y\right) \\ &= \left(R^\top \hat{V}^{-1} R + \lambda P\right)^{-1} R^\top \hat{V}^{-1} \text{Var}(Y) \hat{V}^{-1} R \left(R^\top \hat{V}^{-1} R + \lambda P\right)^{-1} \\ &= \left(R^\top \hat{V}^{-1} R + \lambda P\right)^{-1} R^\top \hat{V}^{-1} \hat{V} \hat{V}^{-1} R \left(R^\top \hat{V}^{-1} R + \lambda P\right)^{-1} \\ &= \left(R^\top \hat{V}^{-1} R + \lambda P\right)^{-1} R^\top \hat{V}^{-1} R \left(R^\top \hat{V}^{-1} R + \lambda P\right)^{-1}\end{aligned}$$

where one assumes that \hat{V} is fixed and does not depend on Y . Therefore, the diagonal elements of this matrix are considered as estimates of $\text{Var}(\hat{b}_{\lambda,j})_j$ even though it is known to often underestimate its target. With these ideas in hand, we get

$$\hat{b}_{\lambda,j} \pm z_{1-\alpha/2} \sqrt{\text{Var}(\hat{b}_\lambda)_{jj}} \text{ gives an approximate } 100(1-\alpha)\% \text{ confidence interval of } \hat{b}_{\lambda,j}.$$

Based on this confidence interval of \hat{b} we can easily build a **pointwise** confidence interval of the prediction $\hat{Y}_i(\cdot)$ at any desired time t .

4 Simulation study of functional models

The aim of this section is to illustrate and validate the estimation procedure described in Section 3 in the framework of “perfectly controlled” data, i.e. in the set-up where the assumptions about the distribution are the ones underlying our theory. The main properties of interest are accuracy, interpretability and smoothness of the estimated parameters, as well as the prediction quality. Hence, we will first conduct a simulation study without penalization of the functional parameters in order to assess the relevance of the method based on prediction accuracy. Secondly, another simulation experiment compares unpenalized parameter estimation with its penalized counterpart. Only the concurrent model is considered in this section, but similar results are expected for the integral model.

4.1 Data simulation process

The framework of the linear model we are considering in the present work can be helpful to explain the variations of a functional response variable through a set of controlled factors, which are the

covariates or explanatory variables. In order to illustrate the relevance of our model, we are going to simulate an artificial data set and evaluate if the parameters are correctly recovered. For this purpose, we will first simulate the covariates and then use them as input to our regression model in order to simulate the corresponding response.

The $p = 5$ functional covariates are simulated at $m = 50$ equidistant viewpoints $(t_j)_j$ over the domain $T = [0, 1]$ according the following procedure :

$$U_i^l(t_j) = \xi_{i,1}^l + \left(\log(10 + t_j) \right)^{\xi_{i,2}^l} + \xi_{i,3}^l \sin \left(\frac{2\pi t_j}{\xi_{i,4}^l} \right) \quad (20)$$

where $\xi_{i,r}^l$ is drawn from $\mathcal{U}([-1, 1])$ ($1 \leq r \leq 4$). This data, as we can see in Figures (1a)-(1e) inside Figure 1 for one randomly chosen individual, is generated at discrete timestamps over $T = [0, 1]$ (blue dots).

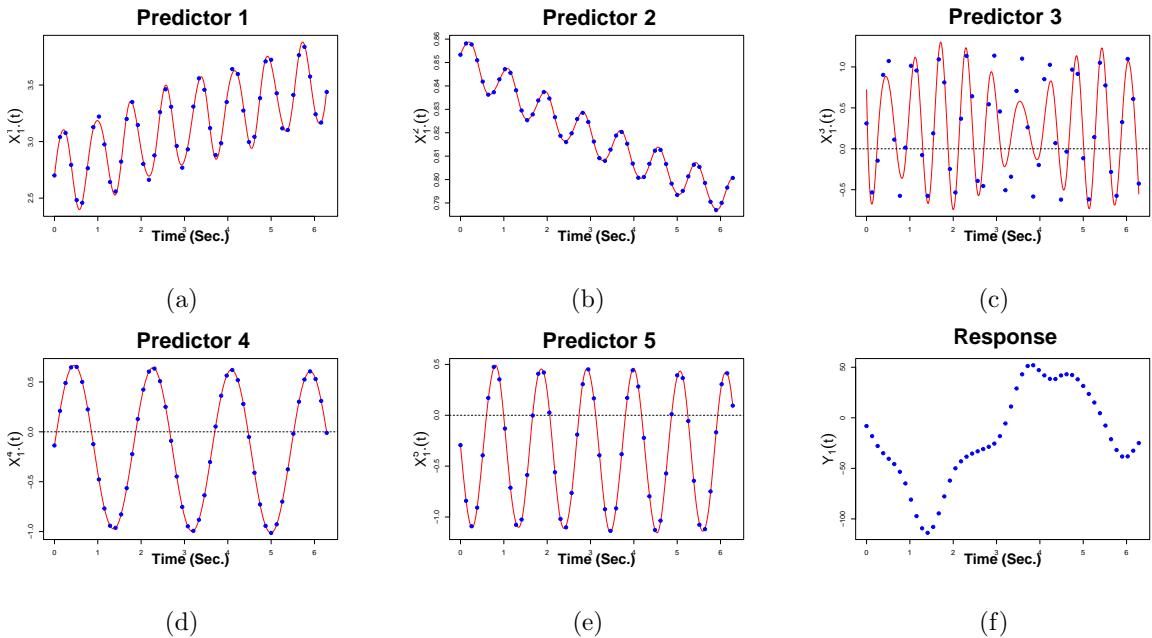


Figure 1: Simulated functional predictors and functional (concurrent) response for a randomly chosen individual.

Before estimating the model, we first compute the underlying expansion into a basis of B-Splines. We obtain the red curves in Figure 1 using $K = 25$ basis functions and equidistant distributed nodes. In other words, we observe $U_i^l(t)$ with $1 \leq l \leq p$ and $1 \leq i \leq n$, and the functional

covariates $X_i^l(t)$ are obtained as:

$$U_i^l(t) = X_i^l(t) + \delta_i^l(t) \quad \text{with} \quad X_i^l(t) = \sum_{j=1}^K x_{ij}^l b_j(t) \quad \text{and} \quad \delta_i^l(t) \sim \mathcal{N}(0, u_i^2).$$

The functional parameters $\beta(t) = (\beta_0(t), \beta_1(t), \dots, \beta_p(t))$ are chosen as follows: $\beta_0(t_j) = (\log(10+t_j))^{\rho_0}$ and $\beta_l(t_j) = \rho_1^l \sin\left(\frac{2\pi t_j}{\rho_2^l}\right)$ with $\rho_0, \rho_1^l, \rho_2^l$ some constants given in Appendix 7.1. Figure 2 shows the corresponding representations of the functional parameters.

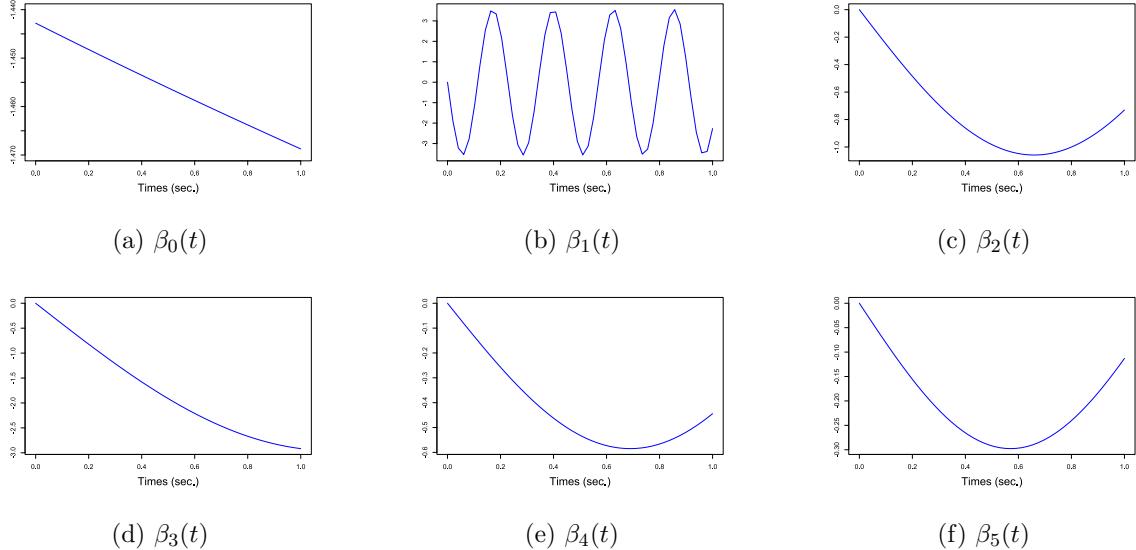


Figure 2: Functional parameters in the concurrent model.

Given the proposed functional covariates and functional parameters, we can now compute the functional response using the concurrent Model (1), for two different sampling rates $n \in \{200; 500\}$. In this experiment, $\varepsilon_i(t)$ is a Gaussian noise with mean 0 and two levels of variance $\sigma^2 \in \{1; 4\}$. For each configuration, we run $N = 50$ Monte Carlo simulations.

4.2 Assessment criteria

We assess the performance of our estimation procedure. Two criteria are considered: prediction accuracy and estimation error for the model parameters. We extend to the functional framework the well-known Mean Relative Prediction Error (MRPE), which is used to quantify the distance

between the actual and the predicted value of the functional response:

$$\text{MRPE} = \frac{1}{m} \sum_{j=1}^m \left(\frac{\sum_{i=1}^n (\bar{Y}_i(t_j) - \hat{Y}_i(t_j))^2}{\sum_{i=1}^n \bar{Y}_i(t_j)^2} \right). \quad (21)$$

We also define one extension of the determination coefficient, which consists of a simple arithmetic average of the classical determination coefficient along the time observation of the functional response. This determination coefficient noted \tilde{R}^2 is defined as follows:

$$\tilde{R}^2 = \frac{1}{m} \sum_{j=1}^m \left(1 - \frac{\sum_{i=1}^n (\bar{Y}_i(t_j) - \hat{Y}_i(t_j))^2}{\sum_{i=1}^n (\bar{Y}_i(t_j) - \bar{Y}_i(t_j))^2} \right), \quad (22)$$

where $\hat{Y}_i(t_j)$ is the predicted output of the sample i at time t_j and $\bar{Y}_i(t_j)$ the mean function of the output sample at time t_j .

To evaluate the performance of the estimation parameters, we compare the actual functional parameters with those provided by our models using the Mean Square Error (MSE) given by:

$$\text{MSE}(\beta_l(\cdot)) = \left[\sum_{l=0}^p \frac{1}{m} \sum_{j=1}^m (\beta_l(t_j) - \hat{\beta}_l(t_j))^2 \right]^{1/2}. \quad (23)$$

4.3 Simulation results

Figure 3 compares the boxplots of the Mean Square Error (23) of the estimated parameters. As expected, the MSE decreases as the number of observations increases. This is the case in the scenarios when we have either a small or a high variance of the model error. Additional information about the results, the estimated functional coefficients versus the actual ones, are available in Appendix 7.3 in Figure 13.

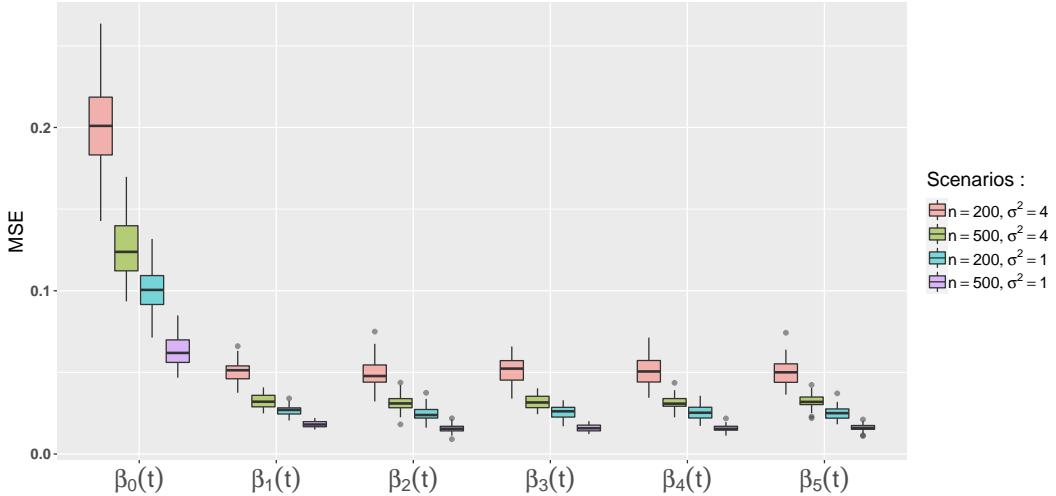


Figure 3: Boxplots of Mean Square Error of estimated parameters over the $N = 50$ Monte Carlo simulation when $n = 200$ (red) and $n = 500$ (blue).

The functional determination coefficient computed over all the scenarios is presented in Table 1. From this table, we observe that when the additive noise increases, the coefficient of determination gets smaller, and increasing the sample size improves the coefficient of determination.

Scenarios	\tilde{R}^2
$n = 200, \sigma^2 = 4$	0.868 (0.0065)
$n = 500, \sigma^2 = 4$	0.878 (0.0040)
$n = 200, \sigma^2 = 1$	0.946 (0.0022)
$n = 500, \sigma^2 = 1$	0.947 (0.0013)

Table 1: Functional determination coefficient \tilde{R}^2 over all the repetitions of any scenarios of simulation.

Let us now turn to our main objective, which is performance in prediction. We generate a test sample with $n = 2000$ observations, and we compare the difference between the actual values of the functional response and the prediction given for each model in a Monte Carlo simulation when $n = 200$ and $n = 500$. Accuracy is measured by the Mean Relative Prediction Error (MRPE). The

boxplots for our four simulation setups are given in Figure 4 while Figure 5 gives the actual values and the prediction over time. The simulations corroborate our expectations that our prediction scheme is able to cope with large variations in the functional response.

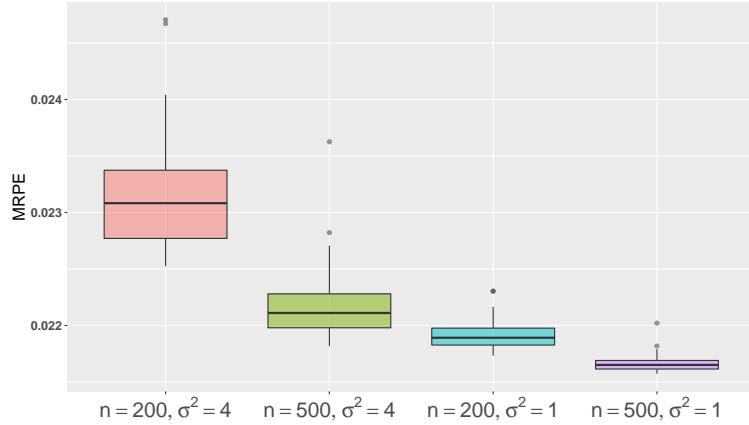


Figure 4: MRPE on a test sample of length $n = 2000$ in all the scenarios of simulation for Monte Carlo simulation

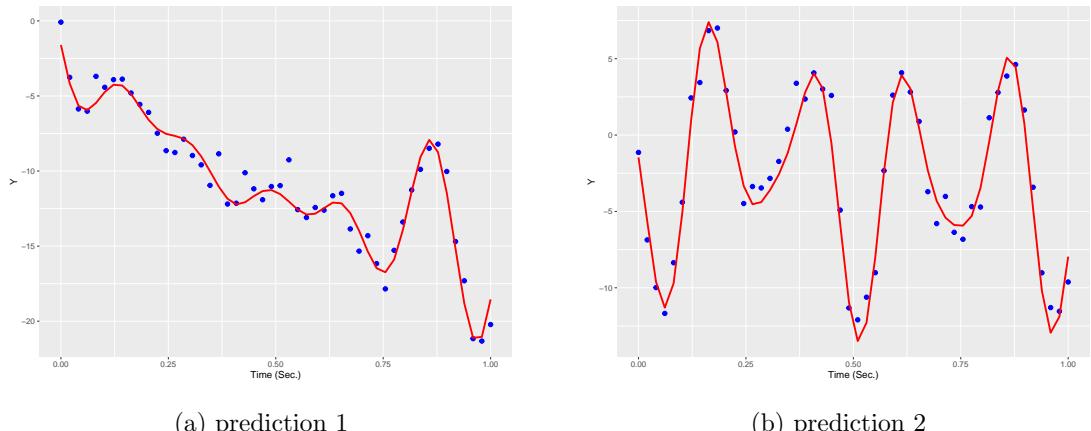


Figure 5: Prediction and actual values of the functional response for two randomly chosen individuals. The red curves is the obtained prediction and the blue dot is the actual data

4.4 Effect of regularization

A close observation of the results shows that the basis expansion of $\beta_0(t)$ requires a large number of zero coefficients, which makes the estimation problem difficult to address in the small sample setting. As a notable consequence, estimation for this parameter may become unreliable for insufficiently large samples. Since the choice of the number of basis functions L_β strongly affects the estimation of the functional parameters, our subsequent strategy will rely on introducing a regularization term. As explained in Section 3, regularization is a flexible and often robust way to adjust the number of basis functions. In order to illustrate the potential positive impact of using regularization, we propose a simulation study with $p = 3$ functional predictors whose parameters are constant (or linear) in some region and present high variability in other regions. We compare the unpenalized setting with the penalized setting when L_β is chosen arbitrarily large. More precisely, we investigate the three following scenarios: $L_\beta = 50$ without regularization, $L_\beta = 50$ with regularization and $L_\beta = 5$ without regularization.

The following functional parameters are considered: $\beta_0(t) = 8t$, $\beta_2(t) = \beta_1(1 - t)$ and

$$\beta_1(t) = \begin{cases} 0 & \text{if } t \leq 0.4, \\ 1.44 \sin\left(\frac{2\pi t}{0.38}\right) & \text{otherwise.} \end{cases}; \quad \beta_3(t) = \begin{cases} 8t & \text{if } t \leq 0.4, \\ 3.2 + 1.44 \sin\left(\frac{2\pi t}{0.38}\right) & \text{otherwise.} \end{cases}$$

For the proposed model, $L_\beta = 50$ basis functions is most certainly too large a number. Therefore, in addition to the prediction accuracy for these three models, we focus on the interpretability of the obtained functional parameters. The objective is to have a parameter that fits very well both in the regions where the function is constant as well as where the function undergoes high variability.

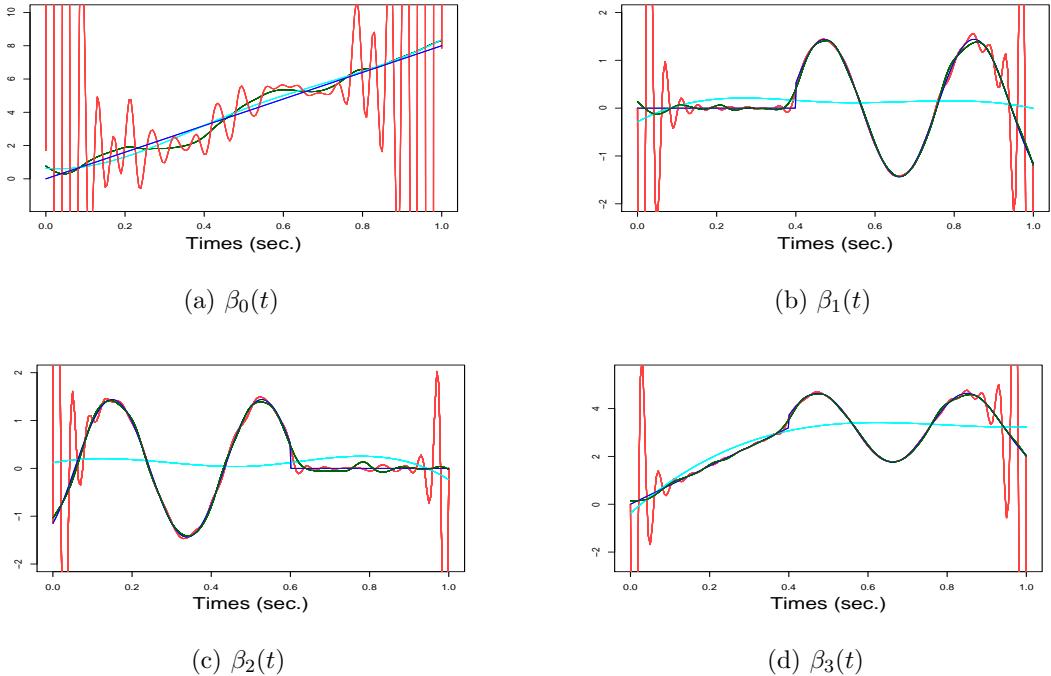


Figure 6: Estimated parameters vs actual ones in the 3 scenarios for any parameters in one of the $N = 50$ simulations. $L_\beta = 5$ without penalization in cyan ; $L_\beta = 50$ without penalization in red ; $L_\beta = 50$ with penalization in green which is completely hidden by the actual parameter in black.

Figure 6 plots one random instance, among the $N = 50$ simulations, of the functional parameters $\beta_0(t)$, $\beta_1(t)$, $\beta_2(t)$ and $\beta_3(t)$ in our three scenarios. Table 2 shows the average MSE (and standard deviation) between actual parameters and the estimated ones in the three considered scenarios. For the $\beta_0(t)$ parameter, which has a linear shape, the best fit comes from Scenario 1 where we have a small number of basis functions. The penalization process (Scenario 3) does not have the best performance on this parameter, but it has an acceptable shape as compared with the non-penalized process for the same number of basis functions (Scenario 2). The unpenalized estimator does not perform well especially at the start and at the end of the domain, while Scenario 1 does provide a sufficient number of basis functions to cope with the more complex behaviour in the very non-linear areas. Our main observation in this setup is that Scenario 3 is globally the best approach among the three since it is the only one that correctly adjusts its complexity to the

oscillations of the function to estimate.

Mean Square Error	$\beta_0(t)$	$\beta_1(t)$	$\beta_2(t)$	$\beta_3(t)$
Scenario 1: $L_\beta = 5$ without penalization	0.20_(0.04)	0.77 _(0.001)	0.77 _(0.000)	0.82 _(0.001)
Scenario 2: $L_\beta = 50$ without penalization	166.58 _(9.1)	4.09 _(3.0)	4.46 _(3.4)	3.61 _(2.77)
Scenario 3: $L_\beta = 50$ with penalization	0.38 _(0.002)	0.06_(0.002)	0.06_(0.002)	0.06_(0.002)

Table 2: Average (and standard errors) obtained over the $N = 50$ repetitions we performed of MSE between estimated parameters and actual ones in the three different scenarios.

An additional important observation is that choosing equispaced knots seems a sufficient strategy for most of the estimation problems we encountered. Sticking to this strategy allows avoiding the cumbersome task of selecting the locations of the knots using cross-validation.

5 Application to real data

In this section, we apply the proposed methodology for function-on-function regression (subsequently denoted by PenFFR which stands for “penalized function-on-function regression” and by FFR for “(unpenalized) function-on-function regression”) for concurrent and integral models. These models are applied to two well-known data sets in FDA: Canadian Weather (CW) data available in the R package `fda` and Hawai Ocean (HO) data available in the R package `FRegSigComp`. We compare the prediction accuracy obtained using our method with the accuracy obtained with other existing methods: integral and concurrent Penalized Function-on-Function Regression (PFFR, Ivanescu et al. (2015)) implemented in the R package `refund`; the signal compression approach (*wSigcomp*) designed by Luo et al. (2016) for the integral model and implemented in the R packages `FRegSigComp`; the Optimal Penalized Function-on-Function Regression (OPFFR) for the integral model (Sun et al., 2018), the Functional Principal Component Analysis (FPCA) and Functional Data Analysis method (FDA) (Ramsay and Silverman, 2005). Due to the unavailability of code for the OPFFR approach, we simply use the published results as presented in their paper (Sun et al., 2018).

Hyper-parameter tuning For our methods (FFR and PenFFR), we consider cubic B-splines basis functions for both functional predictors and regression coefficients. On the CW data set, we use 100 basis functions to address the functional and complex nature of the predictors and on HO data set, we use 40 basis functions. This choice is motivated by the fact that on the raw data, predictors on CW data set has 365 measurements while predictors in the HO data set have 200 measurements. The number of basis functions of parameters is set to 15 on CW data, both for integral and the concurrent models. For the HO data, based on the fact that we have 4 functional predictors and we know that the number of features of design matrix depends on the squared of the number of basis functions in the integral model. So for this complexity, we choose 40 basis functions for the concurrent model and only 6 for the integral model. The penalty parameters λ_l of any predictor is selected using cross-validation on a predefined grid of values (10 equispaced values between 0.1 and 2.0).

For the PFFR method we used the default settings prescribed in the software and only set the number of basis functions for both the functional parameters and predictors. To correctly compare to our proposed method, we also used a cubic splines basis for both the functional predictors and parameters for the two (CW and HO) data sets. We use as our method the same number of basis functions to recover the functional nature of the predictors and on parameters.

For the *wSigcomp* method designed for the integral model, the default settings of the software are also used. For the HO data set which is tested by authors in their package description, the number of basis functions is set to 40 for the functional parameters and 20 for predictors. For the CW data, we slightly change but in the same proportion these value and set the number of basis functions involved for the functional parameters to 80 and the predictors to 40. We have detailed the choices of the hyperparameters but it should be noted that the performance of all these methods remains slightly sensitive to a reasonable variation of these values.

Methods	Canadian Weather Data			Hawaii ocean data		
	Type of basis	$X_i^\ell(t)$	$\beta_\ell(t)$	Type of basis	$X_i^\ell(t)$	$\beta_\ell(t)$
Integral PenFFR / FFR	cubic B-splines	100	10	cubic B-splines	40	6
Concurrent PenFFR / FFR	cubic B-splines	100	40	cubic B-splines	40	20
Integral PFFR	cubic B-splines	100	10	cubic B-splines	40	6
Concurrent PFFR	cubic B-splines	100	40	cubic B-splines	40	20
<i>wSigcomp</i>	wavelets + SVD	40	80	wavelets + SVD	20	40
OPFFR	/	/	/	/	/	/
FDA	Cubic B-splines	/	10	/	/	/
FPCA	SVD	/	/	/	/	/

Table 3: Number of basis functions for the regression coefficients $\beta_\ell(t)$ and the covariates $X_i^\ell(t)$

5.1 Canadian weather data

The data set consists of $m = 365$ daily temperature measurements (average over the years 1961 to 1994) at $n = 35$ weather stations in Canada and their corresponding daily precipitation (in log scale). The weather stations are located in $K = 4$ climate zones: Atlantic, Pacific, Continental and Arctic and the aim is to use the daily temperature to predict the precipitation at each station. Figure 7 gives the daily average over the years 1961 to 1994 (temperature on the left, precipitation on the right). Note that the stations in the Pacific zone have the highest precipitation values, and stations from this zone also have the highest temperatures in the winter. The same can be said about the stations in the Arctic zone for low temperatures and precipitation. A positive relationship between temperature and precipitation can therefore be suspected.

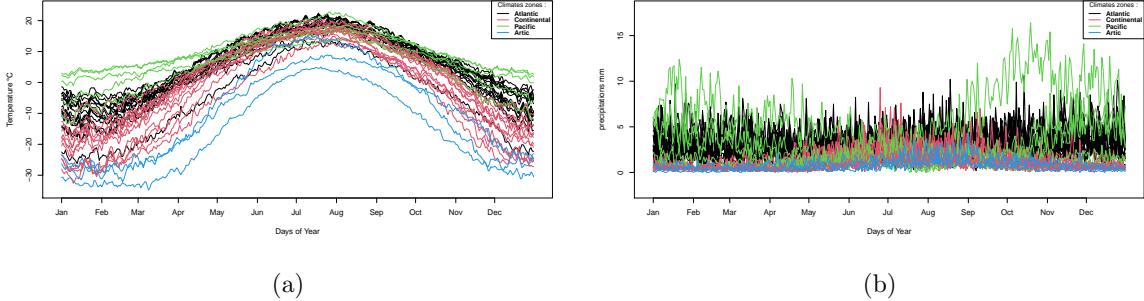


Figure 7: 35 daily mean temperature (a) and precipitation (b) measurement curves.

Our methods (FFR and PenFFR) for the concurrent model (1) and integral model (2) are compared with the PFRR, OPFRR, FPCA, FDA and *wSigcomp* methods. As previously mentioned, we use for the OPFRR, FDA and FPCA methods, the results presented in Sun et al. (2018) in terms of prediction accuracy over the 365 days of the year through the leave-one-out cross-validation integrated square error (ISE) given by:

$$\text{ISE}_i = \int_0^{365} \left(Y_i(t) - \hat{\beta}_{(-i)} X_i(t) \right)^2 dt$$

where the predictor $X_i(\cdot)$ derives from the noisy daily temperature measurements; the functional response $Y_i(\cdot)$ is the log daily precipitation and $\hat{\beta}_{(-i)}$ is the functional parameter estimated in the data set of all the observations except for the i^{th} observation.

For sake of reducing the computational burden, instead of the ISE, the L^2 -norm between the actual and prediction values on a grid of values t is used as a surrogate. It is given by:

$$\widehat{\text{ISE}}_i = \sum_{j=1}^{365} \left(Y_i(j) - \hat{\beta}_{(-i)} X_i(j) \right)^2. \quad (24)$$

The average $\widehat{\text{ISE}}_i$ values for the different models are given in Table 4. They show the numerical advantage of our proposed PenFFR method over the other methods. We also note that the variance observed in our predictions remains quite high for the different models. This is due to the quality of the input data. For recall that we are trying to predict precipitation from temperature on a dataset of 35 very different weather stations

Methods	$\widehat{\text{ISE}}$
Integral PenFFR	33.66 (22.99)
Concurrent PenFFR	36.40 (40.42)
Integral FFR	34.63 (26.03)
Concurrent FFR	36.50 (40.51))
Integral PFFR	41.37 (48.91)
Concurrent PFFR	89.31 (52.03)
<i>wSigcomp</i>	45.37 (52.45)
OPFFR	40.28 (45.76)
FDA	44.16 (56.95)
FPCA	45.51 (45.78)

Table 4: The average (and standard deviation) of $\widehat{\text{ISE}}$ for the Canadian Weather data set. The best result is in boldface.

Also shown in Figure 8 is the prediction obtained using the different methods. We restrict our attention to the integral model since it appeared to be the best model for this data set, independent of the estimation method (PenFFR, PFFR and *wSigcomp*). The prediction is given for two randomly chosen weather stations (Iqaluit and Arvida) and are compared with the actual precipitation. Similar results are illustrated by Figure 14 in the appendix for the concurrent model.

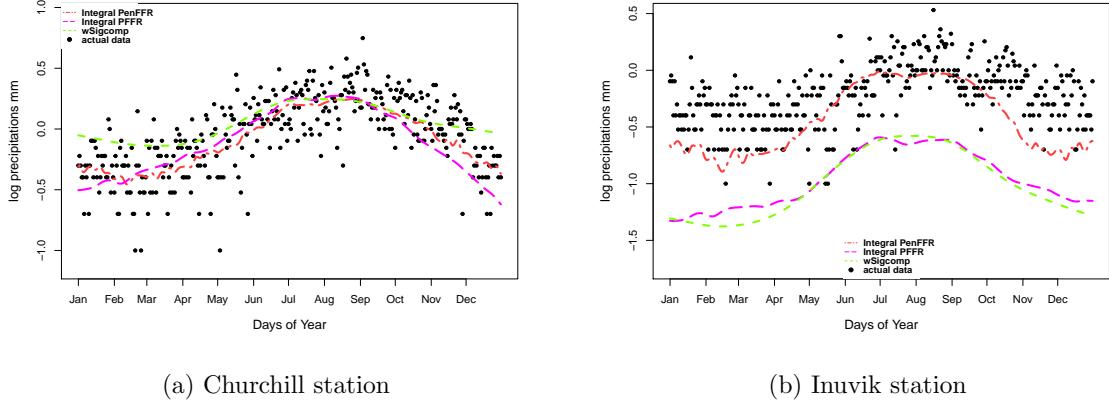


Figure 8: Prediction on two randomly chosen stations. For each figure, the black points are the actual data, the red two-dashed line represents the prediction given by our integral PenFFR, the magenta dashed line is the prediction given by the integral PFFR method, and the long-dashed green line is the prediction given by the wsigcomp method.

5.2 Hawaii ocean data

This data set is one of those used by Luo et al. (2016) to apply their *wSigcomp* approach. The data set includes physical and biochemical oceanographic observational data from the Hawaii Ocean Time-series (HOT) Program, including thermosalinograph, Conductivity, Temperature and Depth (CTD), bottle and biochemical data. The HOT program makes repeated observations of the physics, biology and chemistry at a site approximately 100 km north of Oahu, Hawaii. In the data set, five variables: Salinity, Potential Density, Temperature, Oxygen and Chloropigment, are observed every two meters between 0 and 200 meters below the sea surface on 116 different days. This data set is available from the R package "FRegSigComp", under the name Ocean data. It consists of 5 functional variables with 116 individuals, each having 101 measurement points. Here, we consider the function-on-function regression model with the salinity curves as the response variable $Y(t)$ and (Potential Density, Temperature, Oxygen, Chloropigment) curves as functional predictors $X(t) = (X^1(t), X^2(t), X^3(t), X^4(t))$. We split the full data set into two train/test sub-data sets where the training data consists of the 50 first days (observations) only.

First of all, we expand all the functions considered into a cubic B-spline basis with 40 basis

functions. Figure 9 displays the sample curves for these variables.

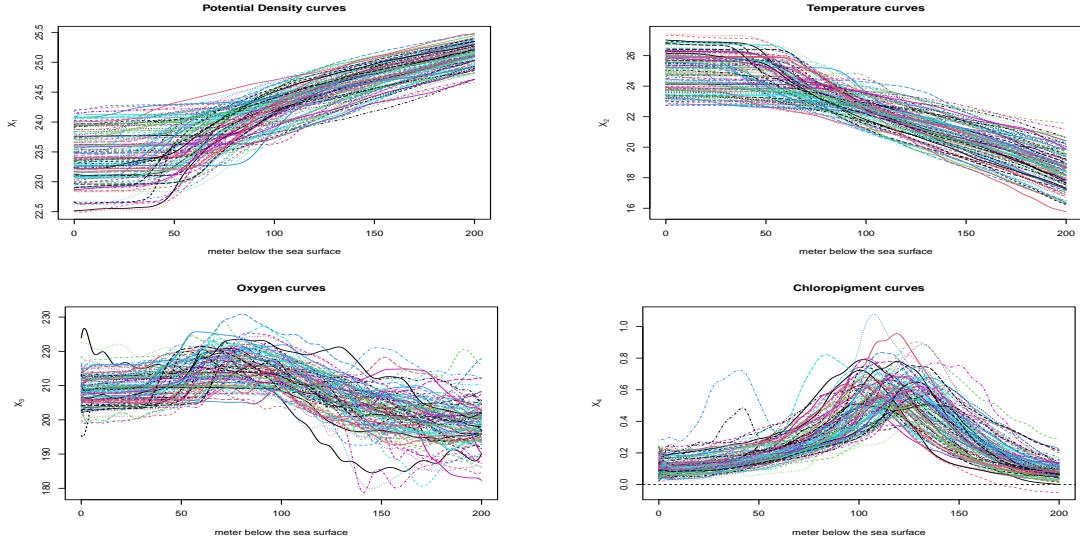


Figure 9: Original sample curves of predictors expanded by cubic B-splines basis with 40 basis functions.

Our PenFFR method is compared with PFFR and *wSigcomp* in the setting of integral models. We also consider PenFFR and PFFR for the concurrent model. Figure 10 and 11 show the estimated parameters $\hat{\gamma}_0(t)$ and $\hat{\gamma}_j(t, s)$, $1 \leq j \leq 4$ obtained for the three methods in the case of the integral model. We first notice that the shape of the estimated parameters is smooth for our method (third column). In addition, Figure 15 in the appendix shows the estimates $\hat{\beta}_j(t)$, $0 \leq j \leq 4$ of the concurrent model with the PenFFR and PFFR methods.

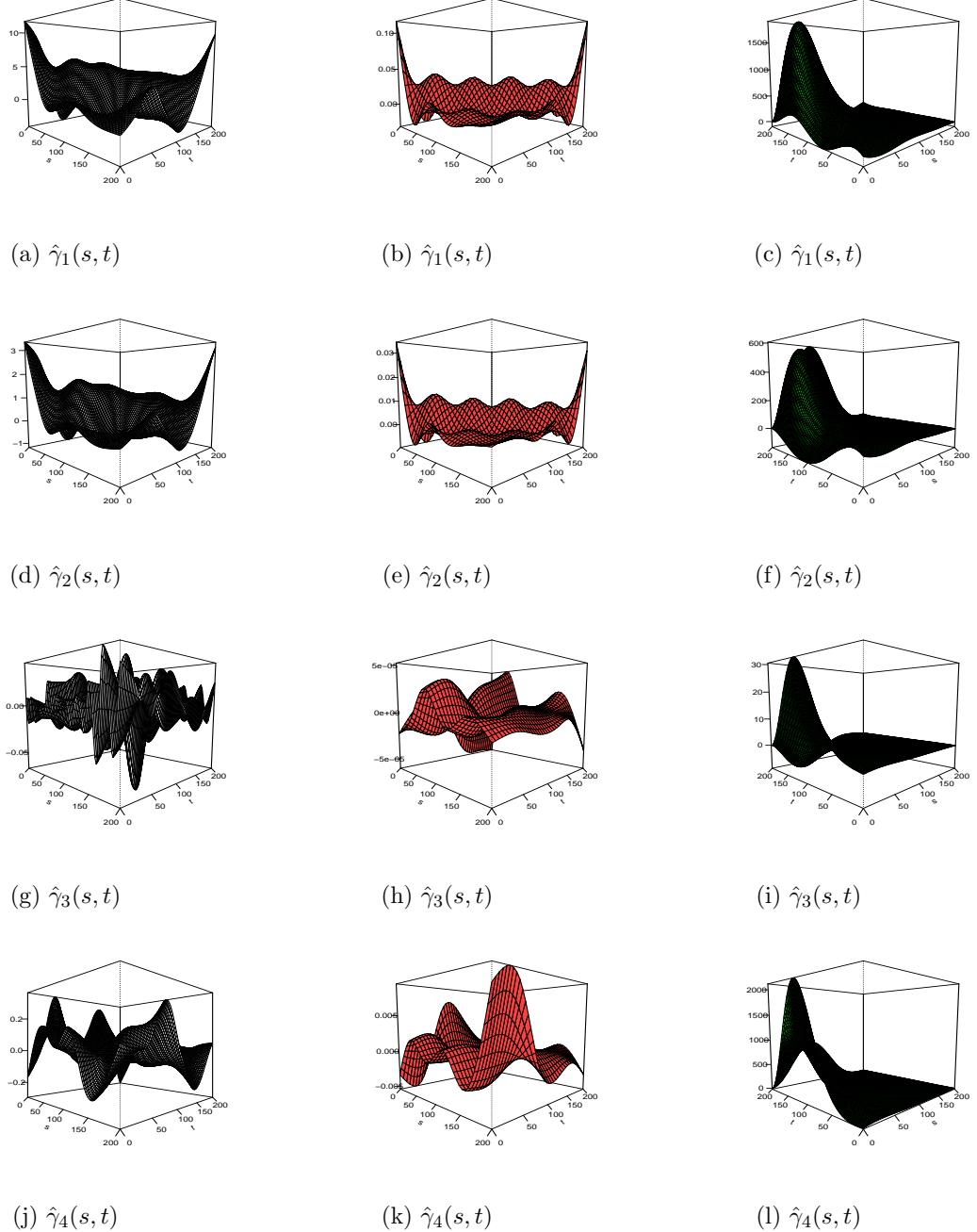


Figure 10: Estimates $\hat{\gamma}_j(s, t)$, $1 \leq j \leq 4$ for the three methods: *wSigcomp* (left column), integral PFFR (middle column) and integral PenFFR (right column).

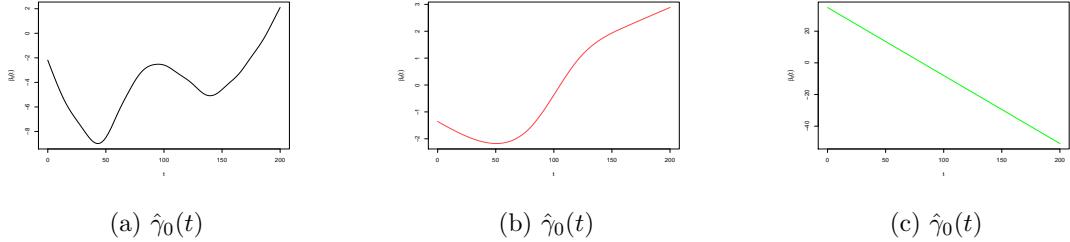


Figure 11: Estimates $\hat{\gamma}_0(t)$, $1 \leq j \leq 4$ for the three methods: *wSigcomp* (left column), integral PFFR (middle column) and integral PenFFR (right column).

Prediction accuracy using $\widehat{\text{ISE}}$ on a test set of size 66 is shown in Table 5. Since the number of individuals for this data (116) is larger than the size of the previous data set, we evaluate the performance on a single test set rather than using cross-validation in order to circumvent the potentially heavy computational burden. Our method is seen once again to outperform all other methods as illustrated in Figure 12 which shows predictions on two randomly chosen individuals.

Methods	$\widehat{\text{ISE}} (\times 10^2)$
Integral PenFFR	0.57 (0.74)
Concurrent PenFFR	4.83 (2.88)
Integral PFFR	2.49 (2.82)
Concurrent PFFR	496.68 (612.18)
<i>wSigcomp</i>	4.79 (4.46)

Table 5: The average (and standard deviation) of $\widehat{\text{ISE}}$ for the Hawaii ocean data set. The best result is in boldface.

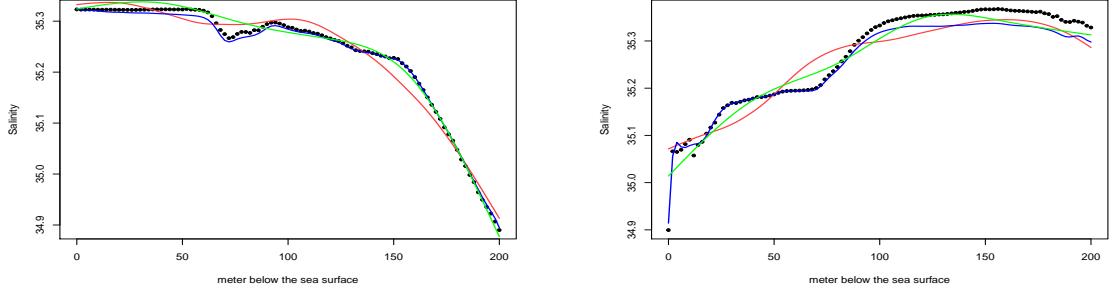


Figure 12: Prediction given by the three methods for integral model on two randomly chosen observations in the test sample: PenFFR in blue, PFFR in green and *wSigcomp* in red. Black dots are the true values.

6 Conclusion

In this article, we have presented a new estimation process for the linear regression model with functional responses and functional covariates. We approach the problem via expanding the functions onto a common B-spline basis, hence allowing the reduction of the functional model to a linear mixed model. Adaptation to unknown smoothness is performed by adding a roughness penalty on second derivatives. Unlike any estimator based on basis functions, our estimates have a smooth shape and sufficient flexibility to capture the encountered variability in various experiments with real-world data sets. We then illustrate the performance of our proposed estimation process in terms of prediction accuracy and parameter interpretability on simulated and real data sets.

Perspectives for future work on this model are manifold. First, prediction confidence bounds can be obtained using various methods such as conformal prediction (Angelopoulos and Bates, 2022), which can handle black box models and could be adapted to our setting as well. Another avenue for future investigations is to explore mixture function-on-function models. This type of mixture model can be safely expected to be extremely relevant when heterogeneous clusters are present in the population (DeSarbo and Cron (1988)). Mixture of experts can also be explored as an additional extension which could prove very efficient in predictive modelling; see Chamroukhi et al. (2022), where a new family of FME is proposed, albeit restricted to scalar responses.

References

- Angelopoulos, A. N. and Bates, S. (2022). A gentle introduction to conformal prediction and distribution-free uncertainty quantification.
- Antoch, J., Prchal, L., Rosaria De Rosa, M., and Sarda, P. (2010). Electricity consumption prediction with functional linear regression using spline estimators. *Journal of Applied Statistics*, 37(12):2027–2041.
- Besse, P. C. and Cardot, H. (1996). Approximation spline de la prevision d'un processus fonctionnel autorégressif d'ordre 1. *Canadian Journal of Statistics*, 24(4):467–487.
- Besse, P. C., Cardot, H., and Ferraty, F. (1997). Simultaneous non-parametric regressions of unbalanced longitudinal data. *Computational Statistics & Data Analysis*, 24(3):255–270.
- Cardot, H., Ferraty, F., and Sarda, P. (1999). Functional linear model. *Statistics & Probability Letters*, 45(1):11–22.
- Chamroukhi, F., Pham, N. T., Hoang, V. H., and McLachlan, G. J. (2022). Functional mixtures-of-experts. *arXiv preprint arXiv:2202.02249*.
- Crainiceanu, C. M., Staicu, A.-M., and Di, C.-Z. (2009). Generalized multilevel functional regression. *Journal of the American Statistical Association*, 104(488):1550–1561. PMID: 20625442.
- DeSarbo, W. S. and Cron, W. L. (1988). A maximum likelihood methodology for clusterwise linear regression. *Journal of classification*, 5(2):249–282.
- Goldsmith, J., Bobb, J., Crainiceanu, C. M., Caffo, B., and Reich, D. (2011). Penalized functional regression. *Journal of Computational and Graphical Statistics*, 20(4):830–851. PMID: 22368438.
- Hastie, T. and Tibshirani, R. (1993). Varying-coefficient models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 55(4):757–796.
- Hida, T., Hui-Hsiung, K., Potthoff, J., and Streit, L. (1993). *White Noise: An Infinite Dimensional Calculus*. Mathematics and its applications. Kluwer Academic Publishers.

- Horváth, L. and Kokoszka, P. (2012). *Inference for Functional Data with Applications*. Springer Series in Statistics. Springer-Verlag, New York.
- Ivanescu, A., Staicu, A.-M., Scheipl, F., and Greven, S. (2015). Penalized function-on-function regression. *Computational Statistics*, 30(2):539–568.
- James, G. M., Wang, J., and Zhu, J. (2009). Functional linear regression that's interpretable. *The Annals of Statistics*, 37(5A):2083–2108.
- Kokoszka, P. and Reimherr, M. (2017). *Introduction to Functional Data Analysis*. CRC Press.
- Leurgans, S. E., Moyeed, R. A., and Silverman, B. W. (1993). Canonical correlation analysis when the data are curves. *Journal of the Royal Statistical Society*, 55(3):725 – 740.
- Li, Y. and Ruppert, D. (2008). On the asymptotics of penalized splines. *Biometrika*, 95(2):415–436.
- Lindstrom, M. J. and Bates, D. M. (1988). Newton—raphson and em algorithms for linear mixed-effects models for repeated-measures data. *Journal of the American Statistical Association*, 83(404):1014–1022.
- Luo, R., Qi, X., and Wang, Y. (2016). Functional wavelet regression for linear function-on-function models. *Electronic Journal of Statistics*, 10(2):3179 – 3216.
- Morris, J. (2014). Functional regression. *Annual Review of Statistics and Its Application*, 2.
- Ramsay, J. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer Series in Statistics. Springer-Verlag, New York, 2 edition.
- Ramsay, J. O., Hooker, G., and Graves, S. (2009). *Functional Data Analysis with R and MATLAB*. Springer Publishing Company, Incorporated, 1st edition.
- Ruppert, D. (2002). Selecting the number of knots for penalized splines. *Journal of Computational and Graphical Statistics*, 11(4):735–757.
- Ruppert, D., Wand, M., and Carroll, R. (2003). *Semiparametric Regression*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

- Scheipl, F. and Greven, S. (2016). Identifiability in penalized function-on-function regression models. *Electronic Journal of Statistics*, 10(1):495 – 526.
- Silverman, B. W. (1996). Smoothed functional principal components analysis by choice of norm. *The Annals of Statistics*, 24(1):1 – 24.
- Sun, X., Du, P., Wang, X., and Ma, P. (2018). Optimal penalized function-on-function regression under a reproducing kernel hilbert space framework. *Journal of the American Statistical Association*, 113(524):1601–1611.
- Wood, S. N. (2006). On confidence intervals for generalized additive models based on penalized regression splines. *Australian & New Zealand Journal of Statistics*, 48(4):445–464.

7 Appendix

7.1 Simulation parameters

The values of the chosen constants is drawn from uniform law between -5 and 5. The values is given by: $\rho_0 = 0.439$, $\rho_1^1 = -3.562$, $\rho_1^2 = -1.058$, $\rho_1^3 = -2.955$, $\rho_1^4 = -0.585$, $\rho_1^5 = -0.298$, $\rho_2^1 = 0.228$, $\rho_2^2 = 2.641$, $\rho_2^3 = 4.462$, $\rho_2^4 = 2.757$ and $\rho_2^5 = 2.283$.

7.2 Mixed model estimator

We first rewrite the model in the form :

$$Y = R^\top b + \varepsilon^*, \quad (25)$$

with $\varepsilon^* = ZU + \eta$, from which we get $V = \text{Var}(\varepsilon^*) = Z\Gamma Z^\top + \sigma^2 I$. We aim to estimate the fixed effects b and the error variance V from the observed data. The most popular estimation methods for the parameters in Model (7) are maximum likelihood (ML) and restricted maximum likelihood (ReML) as described in Lindstrom and Bates (1988). The log-likelihood of the model is written as:

$$\mathcal{L}_{pen}(b, V) = nm \log(2\pi) + \log|V| + (Y - R^\top b)^\top V^{-1} (Y - R^\top b) + b^\top (\lambda P) b \quad (26)$$

First order condition: $\frac{\partial}{\partial b} (\mathcal{L}_{pen}(b, V)) = 0$.

$$\begin{aligned} \frac{\partial \mathcal{L}_{pen}}{\partial b} &= \frac{\partial}{\partial b} \left((Y^\top - (R^\top b)^\top) V^{-1} (Y - (R^\top b)) + b^\top (\lambda P) b \right) \\ &= \frac{\partial}{\partial b} \left((Y^\top V^{-1} Y - Y^\top V^{-1} R^\top b - (R^\top b)^\top V^{-1} Y + (R^\top b)^\top V^{-1} R^\top b) + b^\top (\lambda P) b \right) \\ &= -(Y^\top V^{-1} R^\top)^\top - R V^{-1} Y + 2 R V^{-1} R^\top b + 2 (\lambda P) b \\ &= -2 R V^{-1} Y + 2 (R V^{-1} R^\top + \lambda P) b. \end{aligned}$$

and by equalizing to 0, i.e. $\frac{\partial \mathcal{L}_{pen}}{\partial b} = 0$, we get:

$$\hat{b}(V) = (R V^{-1} R^\top + \lambda P)^{-1} R V^{-1} Y. \quad (27)$$

By replacing b by its estimator in the likelihood expression, we get the profiled log-likelihood given by:

$$\begin{aligned}
\mathcal{L}_p(V) &= -\frac{1}{2} \left(N \log(2\pi) + \log |V| + \left(Y - R^\top (R V^{-1} R^\top)^{-1} R V^{-1} Y \right)^\top V^{-1} \right. \\
&\quad \left. \left(Y - R^\top (R V^{-1} R^\top)^{-1} R V^{-1} Y \right) \right) \\
&= -\frac{1}{2} \left(N \log(2\pi) + \log |V| + \left(Y^\top V^{-1} - Y^\top V^{-1} R^\top (R V^{-1} R^\top)^{-1} R V^{-1} \right) \right. \\
&\quad \left. \left(Y - R^\top (R V^{-1} R^\top)^{-1} R V^{-1} Y \right) \right) \\
&= -\frac{1}{2} \left(N \log(2\pi) + \log |V| + Y^\top V^{-1} Y - Y^\top V^{-1} R^\top (R V^{-1} R^\top)^{-1} R V^{-1} Y - \right. \\
&\quad \left. Y^\top V^{-1} R^\top (R V^{-1} R^\top)^{-1} R V^{-1} Y + \right. \\
&\quad \left. Y^\top V^{-1} R^\top (R V^{-1} R^\top)^{-1} R V^{-1} R^\top (R V^{-1} R^\top)^{-1} R V^{-1} Y \right) \\
&= -\frac{1}{2} \left(N \log(2\pi) + \log |V| + Y^\top V^{-1} Y - Y^\top V^{-1} R^\top (R V^{-1} R^\top)^{-1} R V^{-1} Y \right) \\
\mathcal{L}_p(V) &= -\frac{1}{2} \left(N \log(2\pi) + \log |V| + Y^\top V^{-1} \left(I - R^\top (R V^{-1} R^\top)^{-1} R V^{-1} \right) Y \right).
\end{aligned}$$

On the other hand, there holds $V = \text{Var}(\varepsilon^*) = \sigma_U^2 Z Z^\top + \sigma^2 I$, and thus, $\mathcal{L}_p(V) = \mathcal{L}_p(\sigma_U^2, \sigma^2)$. It is obviously not easy to derive this likelihood which no longer depends on b . Moreover, maximizing this last function gives the MLE which is nevertheless biased. For these reasons, and in order to account for the degrees of freedom of the fixed effects in the model, we propose to use the Restricted Maximum Likelihood (ReML) which reads:

$$\mathcal{L}_R(V) = \mathcal{L}_p(V) - \frac{1}{2} \log |R V^{-1} R^\top| \quad (28)$$

From a numerical viewpoint, we obtain the estimator \hat{V} of the variance V by maximizing this last likelihood from which we finally deduce the value of \hat{U} given by:

$$\begin{cases} \hat{b} = (R^\top \hat{V}^{-1} R + \lambda P)^{-1} R^\top \hat{V}^{-1} Y, \\ \hat{U} = \sigma^2 Z^\top \hat{V}^{-1} (Y - R^\top \hat{b}). \end{cases} \quad (29)$$

7.3 Parameter representation on simulated data

In each scenario, we estimate the functional parameters with cubic B-splines basis, regular knots over the grid and $L_{\beta^l} = 50$ basis functions. The parameters we obtain with our model are close to the true parameters. However, we note that estimation of $\beta_0(t)$ is noised by the two large number of basis functions considered. This confirms the previously mentioned concerns about interpretability (smoothness) of the estimated parameters without regularization. Figure 13 also confirms that estimation accuracy increases with the number of observations.

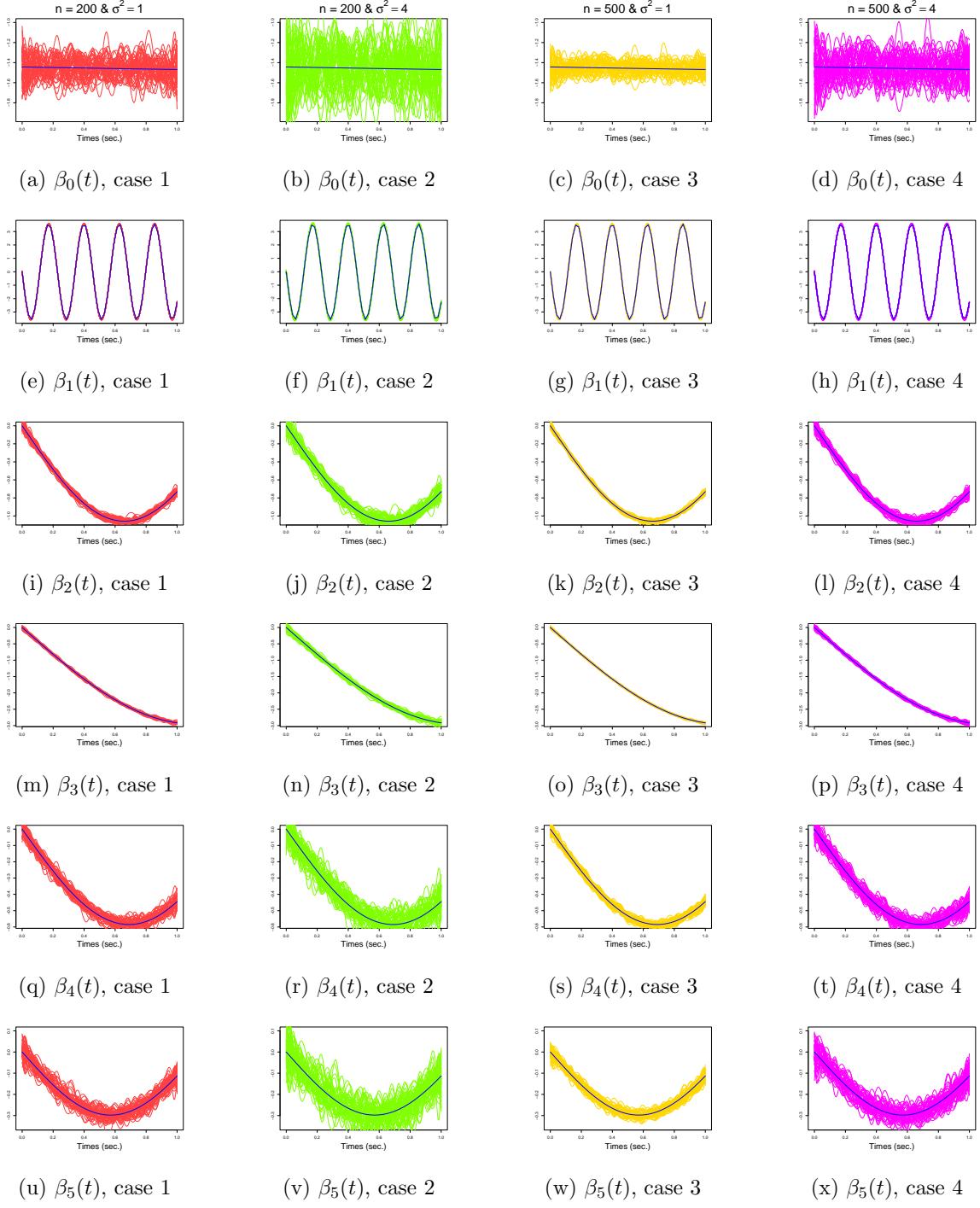


Figure 13: Estimated and actual parameters for the concurrent model over the 4 scenarios of simulation.

7.4 Prediction on concurrent models for Canadian Weather data

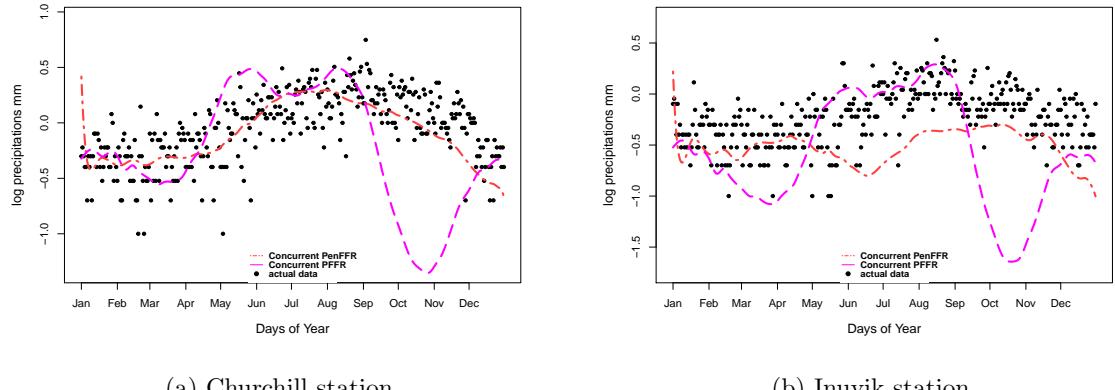


Figure 14: Prediction on two randomly chosen stations. For each figure, the black points are the actual data, the red two-dashed line is the prediction given by our concurrent PenFFR and the magenta dashed line is the prediction given by the concurrent PFFR method.

7.5 Parameters estimation for concurrent models on Hawaii Ocean Data

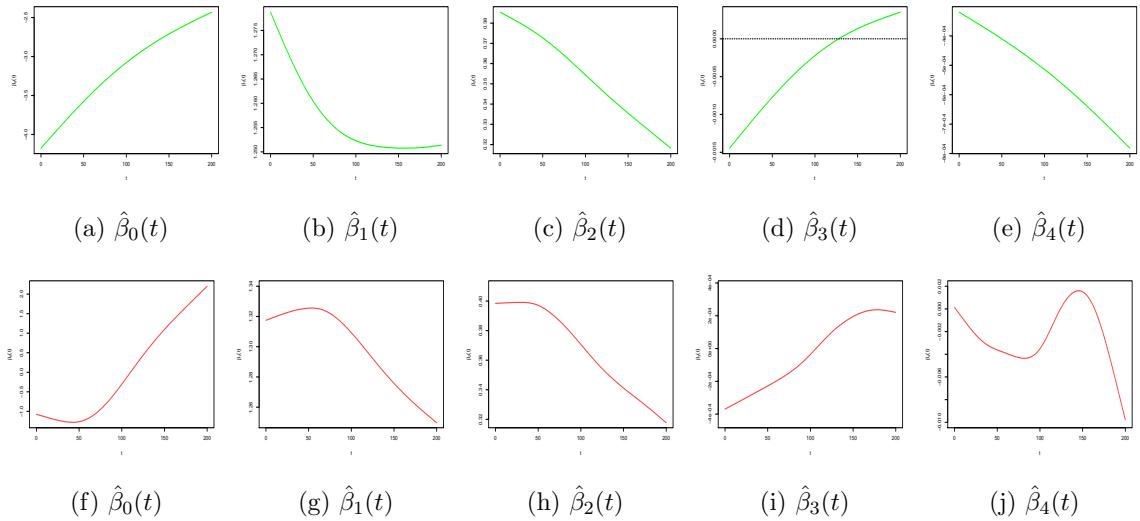


Figure 15: Estimates $\hat{\beta}_j(t)$, $0 \leq j \leq 4$ for the two methods (*pffr* and PenFFR) on concurrent model. The first row shows the estimation provided by our PenFFR method. The second row shows the estimation provided by the *pffr* method.

A mixture of experts regression model for functional response with functional covariates

Jean Steve TAMO TCHOMGUI

jean-steve.tamo-tchomgui@univ-lyon2.fr

Entrepôts, Représentation et Ingénierie des Connaissances

Julien JACQUES

Entrepôts, Représentation et Ingénierie des Connaissances

Guillaume FRAYSSE

Orange (France)

Vincent BARRIAC

Orange (France)

Stéphane CHRETIEN

Entrepôts, Représentation et Ingénierie des Connaissances

Research Article

Keywords: Mixture of Experts, Functional regression, EM algorithm, Ridge regularized estimation

Posted Date: March 26th, 2024

DOI: <https://doi.org/10.21203/rs.3.rs-4142146/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

A mixture of experts regression model for functional response with functional covariates

Jean Steve TAMO TCHOMGUI^{1,2}, Julien JACQUES²,
Guillaume FRAYSSE¹, Vincent BARRIAC¹,
Stéphane CHRETIEN²

¹Orange Innovation, France.

²Univ Lyon 2, ERIC, France.

Contributing authors: jean-steve.tamo-tchomgui@univ-lyon2.fr;
julien.jacques@univ-lyon2.fr; guillaume.fraysse@orange.com;
vincent.barriac@orange.com; stephane.chretien@univ-lyon2.fr;

Abstract

Due to the fast growth of data that are measured on a continuous scale, functional data analysis has undergone many developments in recent years. Regression models with a functional response involving functional covariates, also called "function-on-function", are thus becoming very common. Studying this type of model in the presence of heterogeneous data can be particularly useful in various practical situations. We mainly develop in this work a Function-on-Function Mixture of Experts (FFMoE) regression model. Like most of the inference approach for models on functional data, we use basis expansion (B-splines) both for covariates and parameters. A regularized inference approach is also proposed, it accurately smoothes functional parameters in order to provide interpretable estimators. Numerical studies on simulated data illustrate the good performance of FFMoE as compared with competitors. Usefullness of the proposed model is illustrated on two data sets: the reference Canadian weather data set, in which the precipitations are modeled according to the temperature, and a Cycling data set, in which the developed power is explained by the speed, the cyclist heart rate and the slope of the road.

Keywords: Mixture of Experts, Functional regression, EM algorithm, Ridge regularized estimation.

1 Introduction

During the past few decades, functional data have become a very popular type of measurement in a constantly growing number of industrial, societal and medical applications. A branch of statistics, Functional Data Analysis (FDA), was developed as a specific discipline for analysing such data. FDA's flexibility in handling complex, high-dimensional, and structured data makes it applicable to a broad range of scientific and practical problems, providing insights that traditional data analysis methods may not be able to unveil. Broadly speaking, this new paradigm concerns the statistical analysis of data where at least one of the variables of interest is treated as a curve, surface or volume (also called function for simplicity) observed over a domain set. Most notable recent applications encompass, in particular, Healthcare and Medicine (monitoring patient health over time, FMRI data), Environmental Science (temperature or precipitation trends over time), Economics and Finance (evolution of stocks or commodities, modelling consumer behaviour over time), Sports Science (Analyzing athletes' performance data over time or during an event to optimize training and performance), Meteorology (analyzing weather patterns and trends to improve forecasting models), Chemometrics (analyzing spectroscopy data to identify and quantify chemical substances), Genomics and Bioinformatics (analyzing gene expression data over time), Traffic Analysis and Urban Planning.

Our main focus in the present work is the extension of linear regression to the functional data setting, a model which has naturally become a major area of research in the field of FDA. Standard references for FDA are [1–4]. A broad overview of functional linear regression is given in [5] and [6]. Using the convention that first term denotes the type of the response and second term denotes the type of the covariate three different setups have been analyzed in the literature: Function-on-Scalar, Scalar-on-Function and Function-on-Function. In the present work, we will focus on the most challenging setup from both statistical and computational perspectives, i.e.

Function-on-Function regression problems. Function-on-Function regression problems have indeed been much less studied than the two other types of functional regression despite their relevance in many important applications. Recently, [7] proposed to estimate a Function-on-Function regression model using a penalized mixed model. A signal compression approach was also recently devised in [8], based on preprocessing the functional covariates using their wavelet transform and on proposing a method to estimate the functional parameter by characterizing them as solutions to a generalized functional eigenvalue problem. In a vast majority of current works in this area, one of the main issues is how to accurately select the most statistically relevant number of basis functions, and the location of the knots for spline models [9]. Another important issue is the interpretability of the obtained estimators [10]. In [11] proposed a Ridge-type penalization on second derivative of parameters using B-splines expansions for both functional covariates and parameters which is a first attempt at resolving the interpretability and model selection problems using convex sparsity-enforcing penalties.

Often in practice, the available data carry some heterogeneity, and the assumption that a unique relationship between the response variable and covariates holds for the full data set may not be valid. To circumvent this problem, a mixture of regression model can be proposed [12, 13]. As we know (see [14] and [15]), mixture models are very powerful at capturing subpopulation behaviour, a crucial capability in most applications. Mixture models have been studied in many different setups and specific algorithms, such as EM-type unpenalized and penalized models have been devised for the estimation of its parameters [16, 17]. Accelerated versions using space alternating schemes [18] and proximal interpretations [19, 20]. Sparsity-enforcing penalized versions were studied in [21].

Unfortunately, standard mixture models do not permit to parameterize the individual probability of each data to belong to a specific cluster. As this usually hampers

the predictive capabilities of mixture models, the framework of Mixture of Experts (MoE) models was first suggested in [22] as a powerful supervised learning procedure that can efficiently handle the potential heterogeneity often present in the data. The MoE model is based on a divide-and-conquer principle, which can be simply understood by realizing that each expert can specialize in smaller problems, and their predictive power can be combined together via a gating function in order to solve the full problem. The MoE model can also be viewed as a version of a multilayer supervised network in the sense that it is composed of K separate networks, each of which learning on a subset of the whole data data, as illustrated by Figure 1. From a more statistical learning perspective, the MoE model consists in a mixture model where both the mixture weights, a.k.a. Gating Functions, and component densities, a.k.a. Experts, depend on each data's covariate. The mixture model and its extension to MoE model has been investigated in the contexts of regression, clustering and discriminant analysis. A useful overview was proposed in [23], in which provides conditions for consistency and asymptotically normal properties are studied. Nevertheless, most MoE models only handle the scalar case. In the functional case, it would be relevant to implement efficient extensions of MoE model as well. This problem has already been tackled in [24], but for scalar response. Our contribution is to extend the MoE model to the Function-on-Function setup and provide an efficient inference algorithm.

The paper is organised as follows: Section 2 briefly presents the framework and the inference of Function-on-Function linear regression models. Section 3 presents the Function-on-Function MoE model we proposed and its inference. Section 4 describes how to implement a penalized version of the estimation scheme. Section 5 proposes extensive simulation experiments that explore the various aspects of the performance of the method. Section 6 finally presents an illustration of the method on two real-world data sets and shows the advantage, in terms of predictive quality, of considering MoE as compared with non-mixture-based approaches.

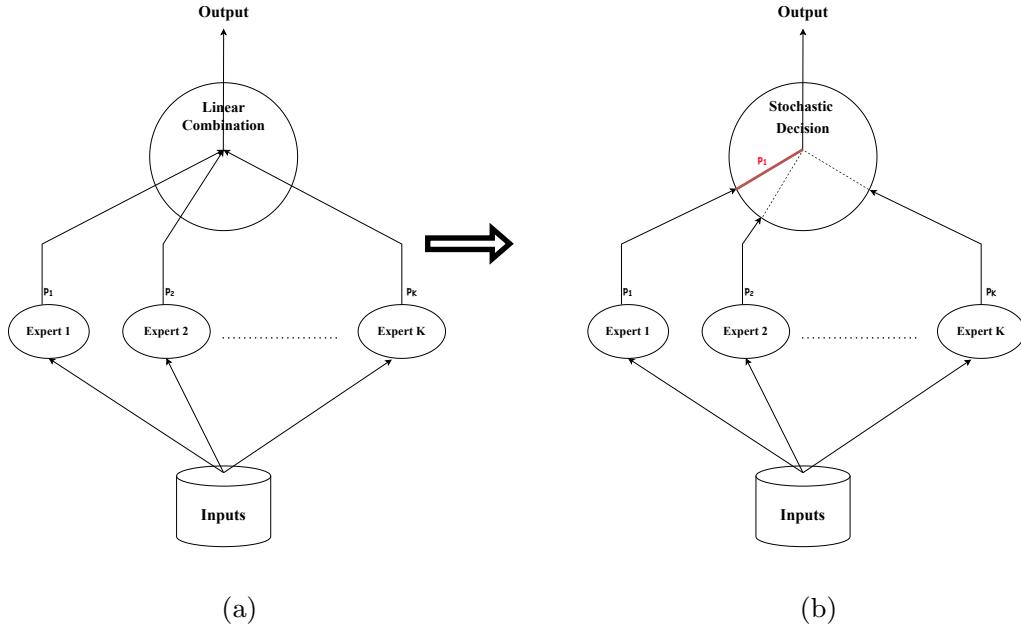


Fig. 1: System of Experts and gating networks: The case of weighted linear combination (a) and the case of stochastic decision (b) to produce output.

2 The concurrent model

2.1 The functional model

The problem under study consists in modelling the relationship between functional covariates $X^1(t), \dots, X^p(t)$ and a functional response $Y(t)$ based on a n -sample $\{Y_i(t), X_i^1(t), \dots, X_i^p(t), t \in [0, T]\}, i = 1, \dots, n$. The functional response and covariates are assumed to belong to the separable Hilbert space $L^2([0; T])$ endowed with the Lebesgue measure. In the present work, we focus on the concurrent model [1] which assumes a linear relationship between the response and covariates, where the value of the response at a particular time stamp is modelled as a linear combination of the covariates at that specific time stamp, and the coefficients of the functional covariates

are univariate smooth functions of time:

$$Y_i(t) = \beta_0(t) + \sum_{\ell=1}^p \beta_\ell(t) X_i^\ell(t) + \varepsilon_i(t) = X_i(t)^\top \beta(t) + \varepsilon_i(t), \quad (1)$$

with $X_i(t) = (1 \ X_i^1(t) \ \dots \ X_i^p(t))^\top$ and $\beta(t) = (\beta_0(t) \ \beta_1(t) \ \dots \ \beta_p(t))^\top$.

$\beta_\ell(t)$ are the unknown functional parameters, assumed to be square integrable. The residuals $\varepsilon_i(t)$ are centered random variables with variance σ_i^2 , specific to the i^{th} individual ([1], Chapter 13). Finally, $\varepsilon_i(t)$ and $X_i^\ell(t)$ are assumed to be uncorrelated. The noise functions $\varepsilon_i(t)$ can also be rigorously defined using white noise theory as presented in [25]. In the framework of the present project, we will only use the property that when sampled at various times from a finite time set \mathcal{T} , the vector $(\varepsilon_i(t))_{t \in \mathcal{T}}$ can be expressed as a sum of a vector with independent and identically distributed (i.i.d.) components and a vector with prescribed covariance matrix, which can be a prescribed to a vector with constant components in the simplest case. Considering the concurrent model is of great interest because, as mentioned in [26], any functional linear model can be reduced to this form.

2.2 From functional to multivariate models

The parameters $\beta_\ell(t)$ of Model (1) can be estimated using the method discussed in [11], where the functional problem is rewritten as a classical multivariate regression problem by expanding the functional covariates and parameters into B-spline series, i.e.:

$$X_i^\ell(t) = \sum_{j=1}^{q_{x^\ell}} x_{ij}^\ell B_j^\ell(t) = B^\ell(t)^\top x_i^\ell \quad \text{and} \quad \beta_\ell(t) = \sum_{j=1}^{q_{\beta^\ell}} b_j^\ell \phi_j^\ell(t) = \phi^\ell(t)^\top b^\ell, \quad (2)$$

where $B^\ell(t) = (B_j^1(t), \dots, B_j^{q_{x^\ell}}(t))^\top$ is the q_{x^ℓ} -dimensional vector of basis functions for the covariate $X^\ell(t)$ and $x_i^\ell = (x_{i1}^\ell, \dots, x_{iq_{x^\ell}}^\ell)$ the corresponding basis expansion

coefficients. Analogously, $\{\phi^\ell(t), b^\ell\}$ are the basis functions and basis coefficients for $\beta_\ell(t)$. Then, using the following notations :

- $\Phi(t) = (\phi^0(t)^\top \ \phi^1(t)^\top \ \cdots \ \phi^p(t)^\top)^\top$, a vector of length $\sum_\ell q_{\beta^\ell}$,
- $b = (b^0^\top \ b^1^\top \ \cdots \ b^p^\top)^\top$, a vector of length $\sum_\ell q_{\beta^\ell}$,
- $B(t) = (1 \ B^1(t)^\top \ \cdots \ B^p(t)^\top)^\top$, a vector of length $\sum_\ell q_{X^\ell}$,
- $x_i = (x_i^0(t)^\top \ x_i^1(t)^\top \ \cdots \ x_i^p(t)^\top)^\top$, a vector of length $\sum_\ell q_{X^\ell}$,

Model (1) can be written:

$$Y_i(t) = x_i^\top B(t)^\top \Phi(t) b + \varepsilon_i(t) = R_i(t)^\top b + \varepsilon_i(t). \quad (3)$$

From this viewpoint, the concurrent model can be recast as a classical linear regression model with design matrix $R_i(t) = \Phi(t)^\top B(t) x_i$ and regression parameters b . When restricted to the observation grid consisting of the m successive timestamps $\{t_1, \dots, t_m\}$, the problem reduces to:

$$Y_i(t_j) = R_i(t_j)^\top b + \varepsilon_i(t_j) \quad \text{with } 1 \leq i \leq n \text{ and } 1 \leq j \leq m. \quad (4)$$

There is nevertheless one peculiarity with this approach to underline. Indeed, in Model (4) the random variables $\varepsilon_i(t_1), \dots, \varepsilon_i(t_m)$ representing the noise can not be assumed independent. In order to circumvent this issue, one possible approach is to use a linear mixed model (LMM) as advocated in [27]. For this purpose, we will assume that the model error can be decomposed as $\varepsilon_i(t_j) = U_i + \eta_{ij}$, with η_{ij} a Gaussian white noise and U_i a random variable which takes into account the random effect in each individual $i \in \{1, \dots, n\}$. In this framework, the estimation procedure proposed in [11] consists in maximizing the ridge-type penalised likelihood, with an ℓ_2 -squared penalty on the second derivatives of $\beta_\ell(t)$. Such a penalty is recommended when smooth estimates

are sought for and provides sufficient flexibility that can still capture a substantial variety of complex shapes.

3 Mixture of experts of linear models for functional response with functional covariates

Mixture Regression (MR) models form a subset of the broad class of statistical models known as finite mixture models [13], which are designed to account for the statistical heterogeneity in a population through a finite set of empirical latent classes. MR models focus on identifying systematic differences between underlying latent groups in the population by the effect of covariates on the response. These models have to be distinguished from other mixture models that estimate the differences in levels and variance of the response variable between the groups (see [12]). MR models assumes that there are $K \in \mathbb{N}^*$ mixture components in the population. Component membership is indicated by a latent categorical variable (one-hot encoding as) $Z = (z_1, \dots, z_K)$ where z_k takes the value 1 if the observation belongs to the component k and 0 otherwise. The MR model can written

$$MR(Y|X) = \sum_{k=1}^K \pi_k \mathbb{E}_k[Y|X, z_k = 1] \quad (5)$$

where π_k is the mixture proportion of group k associated with the k -th expert $\mathbb{E}_k[Y|X]$. In the present functional case, this expert is defined by

$$\mathbb{E}_k[Y(t)|X(t), z_k = 1] = X(t)^\top \beta_k(t) \quad (6)$$

where $\beta_k(t) = (\beta_{k,0}(t), \beta_{k,1}(t), \dots, \beta_{k,p}(t))$ the functional parameters of the k^{th} expert.

Within the proposed model, there are two possible options for designing the probabilities π_k , $k = 1, \dots, K$. The first one assumes that the covariates X are not related

to latent classes Z : $\pi_k = \mathbb{P}(z_k = 1)$. The second, and more general, assumes that Z depends on X : $\pi_k = \pi_k(X) = \mathbb{P}(z_k = 1 | X)$.

The conditional density of $Y(t)$ according to the Function-on-Function Mixture of Expert (FFMoE) model is

$$f(Y(t)|X(t), \Psi(t)) = \sum_{k=1}^K \pi_k(X(t), \alpha_k(t)) \Phi(Y(t); X(t)\beta_k(t), \sigma_k^2), \quad (7)$$

with

- $\pi_k(X(t), \alpha_k(t))$ the mixture proportion of group k , also called the k^{th} gated network function;
- $\Psi_k(t) = (\beta_k(t), \alpha_k(t))$ are the functional parameters;
- $\Phi(Y(t); X(t)\beta_k(t), \sigma_k^2)$ is the Gaussian density probability function of mean $X(t)\beta_k(t)$ and variance σ_k^2 .

3.1 Modelling the gated network function

The MoE model can be seen as a submodel of the Latent class model proposed by [28] named concomitant-variable latent class model. Various models for gated network have been proposed in the past "non-functional" related literature. One instance is the version of [22] where a multinomial logistic model is introduced. Another approach presented in [29] considers non parametric models. Turning to the functional setup, various authors have proposed extensions of the logistic regression model of [22]. Most of them assume in particular that the functional terms all belong to the space of real square integrable functions $L^2([0, 1])$. See for instance [30] for an overview. In [31], it is shown that the functional nature of covariates raises important technical issues, some of them inherited from the non-functional setup but with higher complexities. Some of the more noticeable issues include the non-existence of maximum likelihood estimators

under general conditions, a remedy being working in a tailored Reproducing Kernel Hilbert Space (RKHS).

In the present work, under the realisation $\mathbf{x}_i(t)$ of $\mathbf{X}(t)$, we consider the following gating softmax function:

$$\pi_k(\mathbf{x}_i(t), \alpha_k(t)) = \frac{\exp(h_k(\mathbf{x}_i(t), \alpha_k(t)))}{1 + \sum_{k'=1}^{K-1} \exp(h_{k'}(\mathbf{x}_i(t), \alpha_{k'}(t)))}, \quad (8)$$

where

$$h_k(\mathbf{x}_i(t), \alpha_k(t)) = \int_T \alpha_k^\top(s) \mathbf{x}_i(s) ds \quad (9)$$

with $\alpha_k(t) = (\alpha_{k,0}(t), \alpha_{k,1}(t), \dots, \alpha_{k,p}(t))^\top$. Notice that, in this model, the mixture proportion is constant over time.

As for the other functional parameters, $\alpha_k(t)$ is assumed to have an expansion into a basis of functions of the form:

$$\alpha_{k,\ell}(t) = \sum_{j=1}^{L_{\alpha^\ell}} a_{k,j}^\ell \varrho_j^\ell(t) = \varrho^\ell(t)^\top a_k^\ell.$$

Similarly as for $\beta(t)$ in (2.2), we can write $\alpha_k(t) = \varrho(t)a_k$ and Equation ((9)) becomes:

$$h_k(\mathbf{x}_i(t), \alpha_k(t)) = \int_T a_k^\top \varrho(s)^\top \mathbf{B}(s) \mathbf{x}_i ds = a_k^\top \underbrace{\int_T \varrho(s)^\top \mathbf{B}(s) dt}_{r_i} \mathbf{x}_i = a_k^\top r_i,$$

Thus Model (8) can be written:

$$\pi_k(\mathbf{x}_i(t), \alpha_k(t)) = \frac{\exp(a_k^\top r_i)}{1 + \sum_{k'=1}^{K-1} \exp(a_{k'}^\top r_i)}. \quad (10)$$

To guarantee the identifiability of $\alpha_k(t) \in L^2(\mathbb{R}^{p+1})$, $k = 1, \dots, K$, $\alpha_K(t)$ is set to the null function (and hence a_K is set to null vector) [32].

3.2 Estimation of the functional MoE via the EM algorithm

In practice, as expected, we only have access to a set of (noisy) observations at the timestamps in the set $\{t_1, \dots, t_m\}$. For an observation i belonging to component k , the k^{th} expert model is given by

$$y_i(t_j) = \beta_{k,0}(t_j) + \sum_{\ell=1}^p \beta_{k,\ell}(t) x_i^\ell(t_j) + \varepsilon_i(t_j) = \beta_k(t_j)^\top x_i(t_j) + \varepsilon_i(t_j), \quad (11)$$

where $\beta_k(t) = (\beta_{0,k}(t), \beta_{1,k}(t), \dots, \beta_{p,k}(t))^\top$ for $k = 1 \dots K$, are the unknown functional experts parameters and are assumed to be square integrable.

As in the simple regression case, the successive observed values of a realisation i can not be assumed statistically independent. The mixed model approach of [27] can again be put to work after decomposing the observation error as $\varepsilon_i(t_j) = U_i + \eta_{ij}$, with η_{ij} a Gaussian white noise and U_i a random variable which accounts for the random effect in each individual observation $i = 1, \dots, n$. To sum up, model (11) consists of a LMM with fixed effects b_k and random effect U_i . In matrix form, this yields:

$$\mathbf{Y} = \mathbf{R}^\top b_k + \mathbf{WU} + \boldsymbol{\eta}, \quad (12)$$

where $\mathbf{Y} = (y_1(t_1), \dots, y_1(t_m), y_2(t_1), \dots, y_n(t_m))^\top$, $\mathbf{R} = (R_i(t_j))_{i,j}$ the design matrix of dimension $q_\beta \times nm$ with $q_\beta = \sum_\ell q_{\beta^\ell}$, $\mathbf{U} = (U_1, U_2, \dots, U_n)^\top \sim \mathcal{N}(\mathbf{0}, \Gamma)$, $\boldsymbol{\eta} =$

$(\eta_{ij})_{i,j} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_{nm})$ and

$$W = \underbrace{\begin{pmatrix} 1_{m \times 1} & 0_{m \times 1} & \dots & 0_{m \times 1} \\ 0_{m \times 1} & 1_{m \times 1} & \dots & 0_{m \times 1} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{m \times 1} & 0_{m \times 1} & \dots & 1_{m \times 1} \end{pmatrix}}_{(nm \times n) - \text{matrix}}.$$

We will make use of the notations $0_{k \times l}$ (resp. $1_{k \times l}$) of size $k \times l$ for the matrices of zeros (resp. ones) and the notation $\mathbf{0}$ for the null vector. We will also denote by Γ the unknown covariance matrix of the random effects. \mathbb{I}_{nm} refers to the $nm \times nm$ identity matrix.

The conditional density of \mathbf{Y} , given the observations is a mixture of K Gaussian distributions of mean $b_k^\top R$ and variance $V_k = WTW^\top + \sigma_k^2 \mathbb{I}_{nm}$. So we have:

$$f(\mathbf{Y}|\mathbf{X}, \Psi) = \sum_{k=1}^K \pi_k(x_i(t), \alpha_k(t)) \Phi_{nm}(\mathbf{Y}; b_k^\top R, V_k), \quad (13)$$

where \mathbf{X} is defined in the same way as \mathbf{Y} . $\Phi_\ell(x; \mu, \Sigma)$ denotes the probability density function of the L -dimensional Gaussian distribution with mean vector μ and covariance matrix Σ . $\Psi = ((a_1, b_1, \sigma_1^2), \dots, (a_K, b_K, \sigma_K^2), U, \Gamma)$ are the vector of parameters of the model to be estimated.

Inference of finite mixture model has been studied by various authors in the literature. We can mention for e.g. [22, 33] that compute Maximum Likelihood Estimators (MLE) via EM algorithm; Bayesian approaches have also been proposed as for instance in [34]; [29] present a parameter estimation approach in a semiparametric setting.

Now the FFMoE model can be defined using finite representation of functional terms. In this setting, we can easily write the observed data log-likelihood given by:

$$\mathcal{L}(\Psi; \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}) = \sum_{i=1}^n \log\left(\sum_{k=1}^K \pi_k(\mathbf{x}_i(t), \alpha_k(t)) \Phi_m(\mathbf{y}_i; b_k^\top \mathbf{R}_i, \mathbf{V}_{k,i})\right) \quad (14)$$

where \mathbf{y}_i is the vector of size m that contains all the measurements for observation i , \mathbf{R}_i and $\mathbf{V}_{k,i}$ are respectively the design matrix and block covariance matrix of \mathbf{V}_k associated with i . Then, the log-likelihood of Equation (14) becomes:

$$\sum_{i=1}^n \log\left(\sum_{k=1}^K \frac{\exp(a_k^\top r_i)}{1 + \sum_{u=1}^{K-1} \exp(a_u^\top r_i)} \cdot \frac{1}{\sqrt{(2\pi)^m |\mathbf{V}_{k,i}|}} \exp\left(-\frac{1}{2} (\mathbf{y}_i - b_k^\top \mathbf{R}_i)^\top \mathbf{V}_{k,i}^{-1} (\mathbf{y}_i - b_k^\top \mathbf{R}_i)\right)\right).$$

As is well known in Finite Mixture Models, the log-likelihood maximisation problem is cumbersome to address without introducing clever intermediate steps that form the philosophy of EM-type algorithms, as extensively discussed in the landmark paper [16]. A basic requirement for the method is to complete the data by imputing latent group membership variables z_i for each observation $i = 1 \dots n$. These latent variables are represented by K binary variables $(z_{i1}, z_{i2}, \dots, z_{iK})$. This model is called a complete model and leads to the complete data log-likelihood given by:

$$\begin{aligned} \mathcal{L}_c(\Psi; \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}) &= \sum_{i=1}^n \sum_{k=1}^K z_{ik} \log\left(\frac{\exp(a_k^\top r_i)}{1 + \sum_{u=1}^{K-1} \exp(a_u^\top r_i)} \cdot \frac{1}{\sqrt{(2\pi)^m |\mathbf{V}_{k,i}|}} \right. \\ &\quad \left. \exp\left(-\frac{1}{2} (\mathbf{y}_i - b_k^\top \mathbf{R}_i)^\top \mathbf{V}_{k,i}^{-1} (\mathbf{y}_i - b_k^\top \mathbf{R}_i)\right)\right). \end{aligned} \quad (15)$$

Let $\Psi^{(0)} = ((a_1^{(0)}, b_1^{(0)}, \sigma_1^{2(0)}), \dots, (a_K^{(0)}, b_K^{(0)}, \sigma_K^{2(0)}), \mathbf{U}^{(0)}, \Gamma^{(0)})$ be an initial estimate of Ψ . The EM algorithm is a generic process consisting of repeating two steps to updates parameters such that the log-likelihood value monotonically increases:

- **E-step:** At this step, we compute the conditional expectation of the log-likelihood given the observed data and the current parameter (at iteration l) estimation $\Psi^{(l)}$. So we define the Q function for the EM algorithm defined by:

$$Q(\Psi^{(l+1)} | \Psi^{(l)}) = \mathbb{E}(\mathcal{L}_c(\Psi^{(l+1)}; \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}) | \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)}). \quad (16)$$

This consists of computing the posterior probabilities $p_{ik}^{(l)}$ that the curves i -th sample $(y_i(t), \mathbf{x}_i(t))$ belongs to the k^{th} component of the mixture under the current model:

$$p_{ik}^{(l)} = \mathbb{E}(z_{ik} | \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)}) = \mathbb{P}(z_{ik} = 1 | \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)}).$$

Using Bayes' theorem, this conditional probability $p_{ik}^{(l)}$ can be expressed as:

$$p_{ik}^{(l)} = \frac{\pi_k(\mathbf{x}_i(t), \alpha_k^{(l)}(t)) \Phi_m(\mathbf{y}_i; b_k^{\top(l)} \mathbf{R}_i, \mathbf{V}_{k,i}^{(l)}, t \in T)}{\sum_{u=1}^K \pi_u(\mathbf{x}_i(t), \alpha_u^{(l)}(t)) \Phi_m(\mathbf{y}_i; b_u^{\top(l)} \mathbf{R}_i, \mathbf{V}_{u,i}^{(l)})}. \quad (17)$$

- **M-step:** Given the previous conditional probability and the observed data, this step updates the current parameters $\Psi^{(l)}$ by maximizing the conditional expectation of the complete data log-likelihood, that is $\Psi^{(l+1)}$:

$$\begin{aligned} Q(\Psi^{(l+1)} | \Psi^{(l)}) &= \mathbb{E}(\mathcal{L}_c(\Psi^{(l+1)}; \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}) | \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)}) \\ &= Q_1(a_k^{(l+1)} | \Psi^{(l)}) + Q_2(b_k^{(l+1)}, \mathbf{V}_k^{(l+1)} | \Psi^{(l)}). \end{aligned} \quad (18)$$

The EM algorithm was shown to be a particular case of the celebrated Proximal Point algorithm in [19, 20] using a Kullbak-Leibler type divergence for the proximity term. Another interesting interpretation in terms of alternating minimisation is given in [35]. Space alternating version of the EM algorithms where proposed in

[18, 36] and [21] for the nonsmoothly penalised case. In this paper, the maximisation of (Q) will be performed using a modified version of the **R** package [37]: in particular, the function `initFlexmix` which allows repeating the EM algorithm with different starting values and choosing the solution with the highest value of the likelihood while allowing concomitant variables, as developed in [38]. The global maximisation problem is split onto two separate maximisation problems (see Appendix A for details):

- the updating of gated network parameters via the maximisation of the function $Q_1(a_k^{(l+1)} | \Psi^{(l)})$ and
- the updating of the expert's parameters via the maximisation of the function $Q_2(b_k^{(l+1)}, V_k^{(l+1)} | \Psi^{(l)})$.

One will easily recognise in each of these two expressions, the likelihood of the multinomial logistic model $Q_1(\cdot)$ and of the linear Gaussian model $Q_2(\cdot)$ for which we know how to compute (at least numerically using e.g. Newton-Raphson iterations) the MLEs.

- The E and M steps are alternated repeatedly until numerical convergence i.e. the difference $\mathcal{L}(\Psi^{(l+1)}; \{y_i(t_j), x_i(t_j)\}_{i,j}) - \mathcal{L}(\Psi^{(l)}; \{y_i(t_j); x_i(t_j)\}_{i,j})$ changes by no more than an arbitrarily small value.

Stability and convergence properties of the method are established in the literature (see [17] for an overview and [20] for the proximal viewpoint).

With the estimates of gated network and experts parameters obtained, a hard-clustering of the link between $X(t)$ and $Y(t)$ is reached using Bayes' rule so that

$$\hat{z}_{ik} = \begin{cases} 1 & \text{if } k = \operatorname{Arg} \max_{1 \leq k \leq K} p_{ik}, \\ 0 & \text{otherwise,} \end{cases} \quad i = 1 \dots n.$$

where p_{ik} is the value of Equation (17) at convergence.

3.3 Model selection

One important challenge in statistical estimation with potentially several possible models depending on hyperparameters is the selection of the most statistically relevant one. In the present model, choosing the correct number of components K is one crucial step of the estimation problem. In the regression setting, the selection can be done using information criteria such as AIC [39] or BIC [40], or using cross validation methods. The latter being time-consuming, we will use information criterion based approaches and more specifically the BIC criterion usually defined using log-likelihood (14) as:

$$\text{BIC} = -2\mathcal{L}(\Psi; \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}) - d \log(n) \quad (19)$$

where $d = K \times (1 + \sum_{\ell=0}^p L_{\beta^\ell} + \sum_{\ell=1}^p L_{\alpha^\ell})$ is the number of free parameters of the model and n the number of observations.

3.4 Prediction

As we have already mentioned, one of the major limitations of simple mixture models is predictive modelling. Since for a new individual, its prediction will be given by the weighted sum of the predictions of each class. This is so far not ideal as this prediction is entirely driven by the prediction of the most probable class and these class probabilities will not change whatever the characteristics of the new individual. With the MoE model, we have seen that we can make this latent class probability depending on the covariates (concomitant variables). In this case, the prediction is given by expert prediction of the most probable class. To build such predictions we first need the conditional probabilities that any individual i belongs to a component

k given by:

$$\pi_k(\mathbf{X}_i(t), \hat{\alpha}_k) = \frac{\exp(\hat{\alpha}_k^\top r_i)}{1 + \sum_{v=1}^{K-1} \exp(\hat{\alpha}_v^\top r_i)}$$

where $\hat{\alpha}_k$ for $1 \leq k \leq K - 1$ are the gated parameters estimators.

We deduce, where component k_m is the most probable class for the i curve, the predictive curve by:

$$\hat{Y}_i(t) = b_{k_m}^\top \mathbf{R}_i(t).$$

As a result, estimating the group membership from covariates is essential to predict the response well.

4 Regularizing the Function-on-Function mixture of experts regression

In the FFMoE model (7) presented in Section 3, it is assumed that the functional covariates and parameters can be decomposed into a finite dimensional functional basis. This assumption allows to get the finite representation (13). The numbers of basis functions of each parameters and covariates should be correctly selected in order to avoid over- or under-fitting. Nevertheless, precise adjustment of these values often induces a high computational effort. In the case of the B-spline basis, even more parameters have to be properly tuned such as the choice of the spline order and the location of the knots. In order to reduce the expected cost of such a computationally demanding procedure, we made the choice of choosing a sufficiently large a priori value for L_β (or L_α) and then apply a penalty. This approach brings the benefit of tuning a single hyperparameter, which is the number of basis functions and improving the smoothness and then interpretability of the estimated functional coefficients. This last point is very interesting in the case of the linear model because as we already know,

the interpretation of the predictors-response relationship becomes more difficult as the shape of the functional parameter $\beta(t)$ (or $\alpha(t)$) does not have any simple structure.

Various approaches to regularize the parameter shape have been proposed in the literature. In our setting of interest, the main goal is to enhance the shape of parameters and then interpretability. [41] are among the first to explore the functional penalization and show that the obtained estimator are less sensitive to the rather subjective choice of the number of basis functions. [10] proposed a method called Functional Linear Regression That is Interpretable (FLiRTI) which address the issue of choosing relevant penalties. Based on variable selection ideas such as the Lasso penalty, FLiRTI produces accurate, flexible and highly interpretable estimates of the functional parameters. The main idea of FLiRTI method is, instead of enforcing sparsity on the function themselves, to enforce sparsity of the derivatives. Using the notation $\beta^{(l)}(t)$ for the l^{th} derivative of $\beta(t)$, we may deduce that $\beta^{(0)}(t) = 0$ guarantees $X(t)$ has no effect on $Y(t)$ at t ; $\beta^{(1)}(t) = 0$ implies that $\beta(t)$ is constant at t ; $\beta^{(2)}(t) = 0$ means that $\beta(t)$ is linear at t and so on.

4.1 Ridge-type penalty on second derivatives

Instead of the Lasso penalty, we proposed to estimated the functional MoE model (13) by maximizing a Ridge-type penalized log-likelihood. The penalty is based on the second derivative of the functional parameters (both gated and experts). This choice is mainly motivated by the desire to obtain a possibly locally constant relationship if needed. Moreover, the use the ridge penalty is motivated by the lack of exact sparsity observed in real problems and the clear benefits of getting a closed form formula for the estimators.

The corresponding penalized (data) log-likelihood function for the observed data is defined using (14) by:

$$\mathcal{L}_{pen}(\Psi; \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}) = \mathcal{L}(\Psi; \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}) + \text{Pen}(\Psi), \quad (20)$$

in which the Ridge regularization term is given by

$$\text{Pen}(\Psi) = \sum_{k=1}^K \sum_{\ell=0}^p \lambda_{k,\ell} \int \beta_{k,\ell}''(t)^2 dt + \sum_{k=1}^{K-1} \sum_{\ell=1}^p \gamma_{k,\ell} \int \alpha_{k,\ell}''(t)^2 dt$$

where

$$\int \beta_{k,\ell}''(t)^2 dt = \int \left[\sum_{j=1}^{L_{\beta^\ell}} b_{k,j}^\ell \varphi_j^{\ell''}(t) \right]^2 dt = \sum_{s,u=1}^{L_{\beta^\ell}} b_{k,s}^\ell b_{k,u}^\ell \Gamma_{su}^\ell$$

with $\Gamma_{su}^\ell = \int \varphi_s^{\ell''}(t) \varphi_u^{\ell''}(t) dt$, and

$$\int \alpha_{k,\ell}''(t)^2 dt = \int \left[\sum_{j=1}^{L_{\alpha^\ell}} a_{k,j}^\ell \varrho_j^{\ell''}(t) \right]^2 dt = \sum_{s,u=1}^{L_{\alpha^\ell}} a_{k,s}^\ell a_{k,u}^\ell \Upsilon_{su}^\ell$$

with $\Upsilon_{su}^\ell = \int \varrho_s^{\ell''}(t) \varrho_u^{\ell''}(t) dt$.

So,

$$\text{Pen}(\Psi) = \sum_{k=1}^K \sum_{\ell=0}^p \lambda_{k,\ell} \sum_{s,u=1}^{L_{\beta^\ell}} b_{k,s}^\ell b_{k,u}^\ell \Gamma_{su}^\ell + \sum_{k=1}^{K-1} \sum_{\ell=1}^p \gamma_{k,\ell} \sum_{s,u=1}^{L_{\beta^\ell}} a_{k,s}^\ell a_{k,u}^\ell \Upsilon_{su}^\ell \quad (21)$$

where $\lambda_{k,\ell}$ and $\gamma_{k,\ell}$ are the usual tuning regularization parameters which control the importance we want to place on the smoothness of estimators. As we know, selecting a good value of $\lambda_k = (\lambda_{k,\ell})_\ell$ (resp. $\gamma_k = (\gamma_{k,\ell})_\ell$) is very important to reduce the noise that less influential covariates create.

By using matrix terms, we get:

$$\mathcal{L}_{pen}(\Psi; \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}) = \mathcal{L}(\Psi; \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}) - \sum_{k=1}^K b_k^\top (\lambda_k \mathbf{P}) b_k - \sum_{k=1}^{K-1} a_k^\top (\gamma_k \mathbf{Q}) a_k$$

where $(\lambda_k \mathbf{P}) \in \mathbb{R}^{L_\beta \times L_\beta}$ is given by:

$$(\lambda_k \mathbf{P}) = \begin{pmatrix} \lambda_{k,0} \Gamma^0 & 0_{L_{\beta^0} \times L_{\beta^1}} & \dots & 0_{L_{\beta^0} \times L_{\beta^p}} \\ 0_{L_{\beta^1} \times L_{\beta^0}} & \lambda_{k,1} \Gamma^1 & \dots & 0_{L_{\beta^1} \times L_{\beta^p}} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{L_{\beta^p} \times L_{\beta^0}} & 0_{L_{\beta^p} \times L_{\beta^1}} & \dots & \lambda_{k,p} \Gamma^p \end{pmatrix} \quad \text{with } \Gamma^\ell = \begin{pmatrix} \Gamma_{11}^\ell & \Gamma_{12}^\ell & \dots & \Gamma_{1L_{\beta^\ell}}^\ell \\ \Gamma_{21}^\ell & \Gamma_{22}^\ell & \dots & \Gamma_{2L_{\beta^\ell}}^\ell \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma_{L_{\beta^\ell} 1}^\ell & \Gamma_{L_{\beta^\ell} 2}^\ell & \dots & \Gamma_{L_{\beta^\ell} L_{\beta^\ell}}^\ell \end{pmatrix};$$

and $(\gamma_k \mathbf{Q}) \in \mathbb{R}^{L_\alpha \times L_\alpha}$ by:

$$(\gamma_k \mathbf{Q}) = \begin{pmatrix} \gamma_{k,0} \Upsilon^0 & 0_{L_{\alpha^0} \times L_{\alpha^1}} & \dots & 0_{L_{\alpha^0} \times L_{\alpha^p}} \\ 0_{L_{\alpha^1} \times L_{\alpha^0}} & \gamma_{k,1} \Upsilon^1 & \dots & 0_{L_{\alpha^1} \times L_{\alpha^p}} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{L_{\alpha^p} \times L_{\alpha^0}} & 0_{L_{\alpha^p} \times L_{\alpha^1}} & \dots & \gamma_{k,p} \Upsilon^p \end{pmatrix} \quad \text{with } \Upsilon^\ell = \begin{pmatrix} \Upsilon_{11}^\ell & \Upsilon_{12}^\ell & \dots & \Upsilon_{1q_{\alpha^\ell}}^\ell \\ \Upsilon_{21}^\ell & \Upsilon_{22}^\ell & \dots & \Upsilon_{2L_{\alpha^\ell}}^\ell \\ \vdots & \vdots & \ddots & \vdots \\ \Upsilon_{L_{\alpha^\ell} 1}^\ell & \Upsilon_{L_{\alpha^\ell} 2}^\ell & \dots & \Upsilon_{L_{\alpha^\ell} L_{\alpha^\ell}}^\ell \end{pmatrix}.$$

Here, $0_{L_1 \times L_2}$ is the standard notation for the null matrix of size $L_1 \times L_2$. As Γ^ℓ (resp. Υ^ℓ) is a symmetric positive-definite matrix for any $0 \leq \ell \leq p$, we can easily find its Cholesky decomposition, which can be efficiently leveraged in the implementation.

And for the penalized complete (data) log-likelihood, we made the same process and by using (15) we get:

$$\mathcal{L}_{pen}^c(\Psi; \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}) = \mathcal{L}_c(\Psi; \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}) - \sum_{k=1}^K b_k^\top (\lambda_k \mathbf{P}) b_k - \sum_{k=1}^{K-1} a_k^\top (\gamma_k \mathbf{Q}) a_k. \quad (22)$$

4.2 Maximum Likelihood estimation via the EM algorithm

The EM algorithm for the regularized FFMoE is developed for maximizing the penalized (data) log-likelihood (22). The algorithm is simply the same as in non penalized version with small changes. The E-step is exactly the same and the M-step is done by splitting the problem into two maximize problems as (see Appendix B for details):

$$\begin{aligned} Q_{pen}(\Psi^{(l+1)} | \Psi^{(l)}) &= \mathbb{E}(\mathcal{L}_{pen}^c(\Psi^{(l+1)}) | y(t), \mathbf{x}(t); \Psi^{(l)}) \\ &= Q_{1,pen}(a_k^{(l+1)} | \Psi^{(l)}) + Q_{2,pen}(b_k^{(l+1)}, \sigma_k^{2(l+1)} | \Psi^{(l)}). \end{aligned} \quad (23)$$

5 Simulation study of mixture of experts functional models

The goal of this section if to evaluate, on the basis of simulated data, the proposed model in the case of Function-on-Function regression model. The data simulation process if derived from [24].

5.1 Data simulation process

100 data sets are simulated according to the FFMoE model with $K = 3$ components and $p = 1$ covariate, on a time domain $[0, 1]$. The covariate is simulated with $\mathbf{X}_i(t) = \mathbf{x}_i^\top \mathbf{B}(t)$, where $\mathbf{x}_i = W.v_i$ with W a 10×10 -matrix of $\mathcal{U}(0, 1)$, v_i a 10-vector of $\mathcal{N}(0, 10)$ and $\mathbf{B}(t)$ is a 10-dimensional B-splines basis. The functional parameters are $\beta_{1,0}(t) = -5t$, $\beta_{2,0}(t) = 0$ and $\beta_{3,0}(t) = 5t$, $\beta_{2,1}(t) = -\beta_{1,1}(t)$, $\beta_{3,1}(t) = 100(t-0.5)^2 -$

Scenarios	number of sampling points: m	number of observations: n
S1	20	300
S2	20	800
S3	100	300
S4	100	800

Table 1: The four scenarios of the simulation study

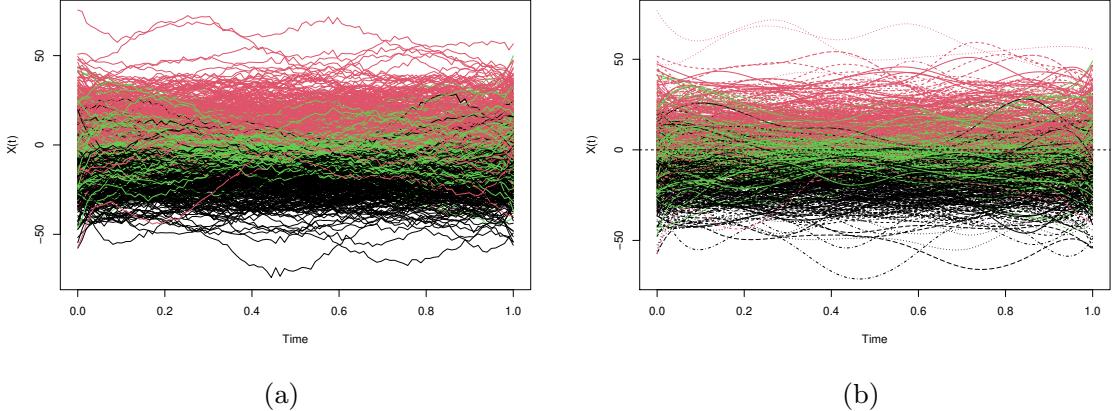


Fig. 2: Discrete observations (left) and cubic B-splines smoothing (right) of the functional covariate. Color depends on the component membership.

10 and

$$\beta_{1,1}(t) = \begin{cases} -50(t - 0.5)^2 + 2 & \text{if } 0 \leq t < 0.3 \\ 0 & \text{if } 0.3 \leq t < 0.7 \\ 50(t - 0.5)^2 - 2 & \text{if } 0.7 \leq t < 1 \end{cases}$$

The functional parameters of the gated network are $\alpha_{1,0} = \alpha_{2,0} = -10$, $\alpha_{3,0} = 0$, $\alpha_{1,1}(t) = 80(t - 0.5)^2 - 8$, $\alpha_{2,1}(t) = -\alpha_{1,1}(t)$ and $\alpha_{3,1}(t) = 0$. Finally, the residuals are simulated with $\varepsilon_i(t) \sim \mathcal{N}(0, 4)$.

The number n of observations and the number m of sampling points are given in Table 1, defining thus four scenarios S1, S2, S3, S4.

Figure 2 plots the discrete covariate observations (left panel) and their corresponding B-splines smoothing (right panel) for Scenario S3. Figure 3 displays the discrete

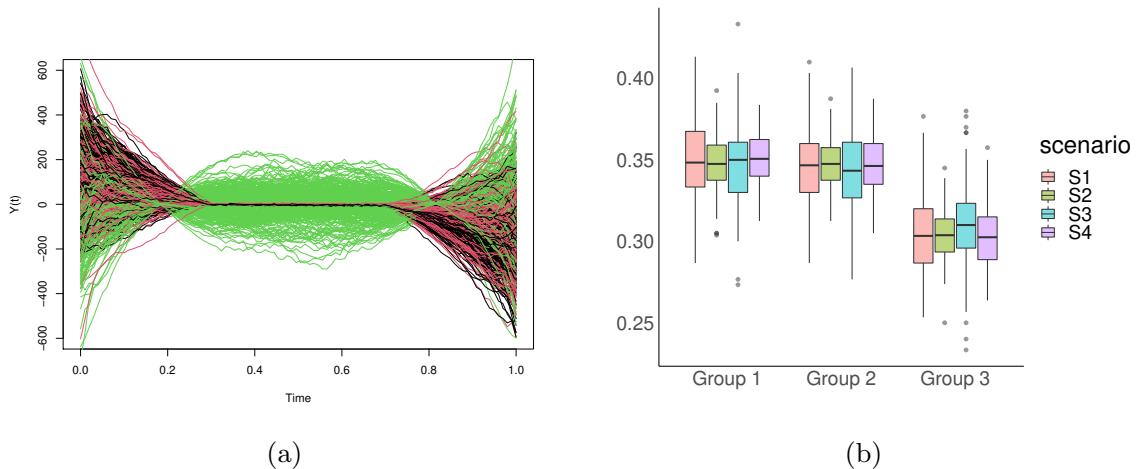


Fig. 3: Discrete observations of the functional output (left) and proportions of observations of each component on the mixture (right).

time sampled observations of the response $Y(t)$ (left) for Scenario S3, and the proportions of observations of each component on the mixture (right) for the four scenarios.

5.2 Assessment criteria of goodness of fit

The assessment of the proposed FFMoE model is performed using two specific indicators: first the estimation quality and second, the prediction quality. In addition, the efficiency of BIC for selecting the number of components is also investigated.

The quality of parameter estimation is evaluated with the Mean Square Error (MSE)

$$\text{MSE}(\beta_\ell(\cdot)) = \left[\frac{1}{m} \sum_{j=1}^m (\beta_\ell(t_j) - \hat{\beta}_\ell(t_j))^2 \right]^{1/2}. \quad (24)$$

Knowing that the label-switching problem sometimes occurs, we will take care of re-labelling the clusters using the estimated confusion matrix, a strategy which is relevant when the true number of mixture components has been guessed.

The quality of prediction is assessed using the Mean Relative Prediction Error (MRPE) on a generate test sample of length $n_{test} = 2000$ for each scenario:

$$\text{MRPE} = \frac{1}{m} \sum_{j=1}^m \left(\frac{\sum_{i=1}^{n_{test}} (\mathbf{Y}_i(t_j) - \hat{\mathbf{Y}}_i(t_j))^2}{\sum_{i=1}^{n_{test}} \mathbf{Y}_i(t_j)^2} \right). \quad (25)$$

Notice that this criterion can be highly irrelevant if the observation is associated with the wrong expert. Subsequently, two additional criteria will be defined: MRPE.good, computed only of the observations associated with the correct expert, and MRPE.bad for those associated with a wrong expert.

5.3 Competitors

The competitors are the non-mixture penalized Function-on-Function regression models PenFFR [11] and pffr [7].

The PenFFR estimation process uses basis expansion of functional covariates and parameters to transform a functional model to multivariate. Estimation scheme is achieved by maximising the penalised log-likelihood using a ridge-type penalty on the second derivatives. Cubic B-splines basis functions were employed for both for functional covariates and the functional parameters. The number of basis functions was set to 10 for both the functional parameters and covariates.

The pffr estimation process uses observed values of functional covariates. An approach that matches with densely or sparsely sampled functions. The functional parameters is estimated using restricted maximum likelihood (REML) in an associated mixed model. For the implementations of the method, we used default settings of the `pffr` function available in the R package `refund`. We only set the number of basis functions to 10 both for functional covariates and parameters.

Finally, for FFMoE and PenFFMoE, we also set the number of basis functions to 10 both for both for functional covariates and parameters.

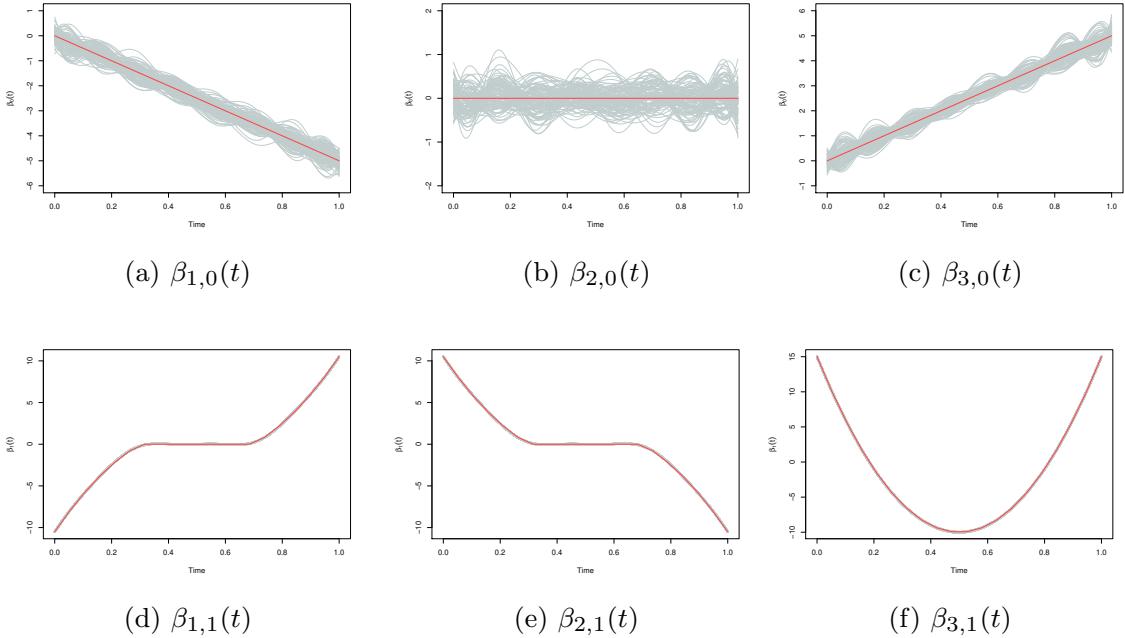


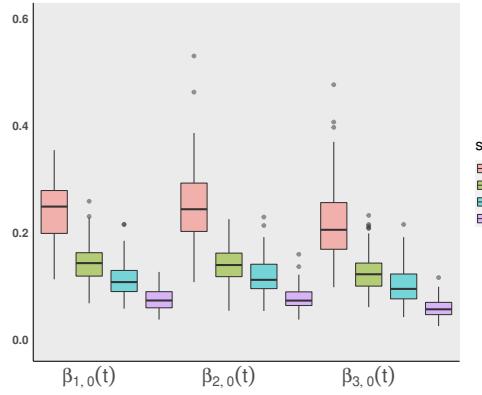
Fig. 4: Estimation of the regression coefficients for Scenario S3 with FFMoE. The red curves are the actual parameters, the gray curves are the estimation.

5.4 Simulation results

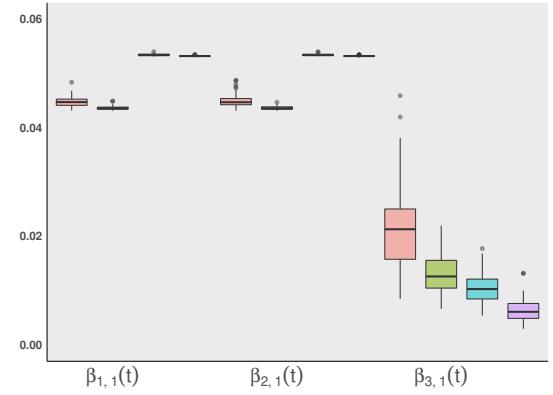
Concerning the ability of BIC to select the right number of components, BIC selects indeed the correct number $K = 3$ in 100% of the case for the four scenarios, and thus for FFMoE and PenFFMoE.

5.4.1 Parameter estimation

The relevance of our model is reflected by the parameter estimation. Figure 4 for FFMoE and Figure C1 for PenFFMoE in appendix show the estimated versus actual parameters from Scenario S3. The estimation of the covariate effect is remarkably accurate. This observation is also supported by the MSE values reported in boxplot given in Figure 6 for PenFFMoE and Figure 5 for FFMoE in all scenarios.

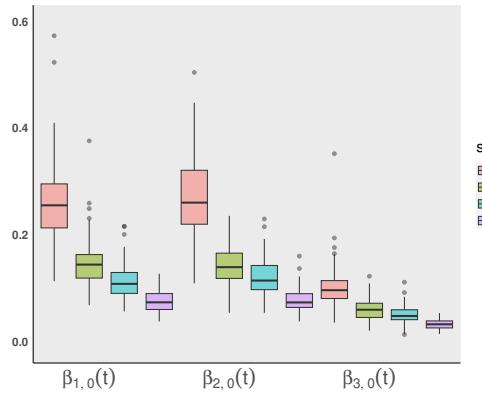


(a) Intercept

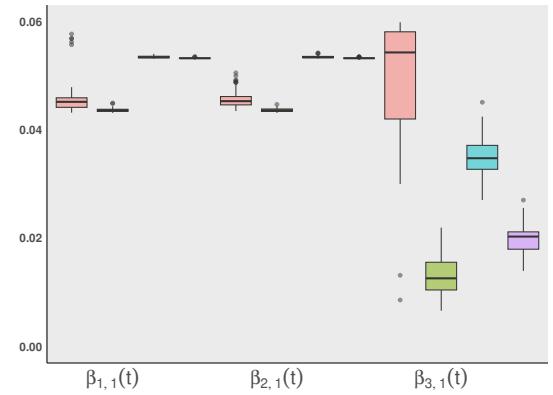


(b) Effect of X

Fig. 5: Boxplot of MSE between actual and estimated parameters for FFMoE. Functional intercept $\beta_0(t)$ (left) and functional effect $\beta_1(t)$ of $X(t)$ (right) in each of the 3 components mixture for our 4 simulated scenarios.



(a) Intercept



(b) Effect of X

Fig. 6: Boxplot of MSE between actual and estimated parameters for PenFFMoE. Functional intercept $\beta_0(t)$ (left) and functional effect $\beta_1(t)$ of $X(t)$ (right) in each of the 3 components mixture for our 4 simulated scenarios.

5.4.2 Prediction accuracy

In Table 2, we present the predictive accuracy through MRPE of the proposed method (FFMoE and PenFFMoE) and of its competitors (PenFFR and pffr) on the test sample. The results are clearly better for FFMoE and PenFFMoE. Let's remark that the

		Expert affectation accuracy	MRPE.good	MRPE.bad	MRPE
S1	FFMoE	91.3% (0.014)	0.006 ($<10^{-4}$)	2.560 (0.30)	0.230 (0.07)
	PenFFMoE	92.3% (0.018)	0.006 ($<10^{-4}$)	2.568 (0.29)	0.205 (0.07)
	PenFFR	-	-	-	1.213 (0.07)
	pffr	-	-	-	1.266 (0.08)
S2	FFMoE	93.0% (0.005)	0.006 ($<10^{-4}$)	2.500 (0.18)	0.180 (0.02)
	PenFFMoE	93.3% (0.003)	0.006 ($<10^{-4}$)	2.501 (0.18)	0.174 (0.01)
	PenFFR	-	-	-	1.192 (0.04)
	pffr	-	-	-	1.252 (0.05)
S3	FFMoE	92.0% (0.026)	0.016 ($<10^{-3}$)	2.715 (0.49)	0.280 (0.38)
	PenFFMoE	92.8% (0.040)	0.016 ($<10^{-3}$)	2.730 (0.45)	0.219 (0.16)
	PenFFR	-	-	-	1.227 (0.07)
	pffr	-	-	-	1.290 (0.09)
S4	FFMoE	93.9% (0.004)	0.015 ($<10^{-4}$)	2.628 (0.21)	0.174 (0.02)
	PenFFMoE	94.3% (0.003)	0.016 ($<10^{-4}$)	2.614 (0.20)	0.165 (0.01)
	PenFFR	-	-	-	1.237 (0.05)
	pffr	-	-	-	1.314 (0.06)

Table 2: Expert affectation accuracy and average (standard deviation) of MRPE on a test sample.

difference between MRPE.good and MRPE.bad show that it is important to correctly affect the observations to the correct expert.

Finally, Figure 7 gives the prediction for four randomly chosen observations compared to actual values. Figure (7a) and Figure (7b) correspond to situations where the observations are assigned to the correct clusters; Figure (7c) corresponds to a case where the data is assigned to the correct clusters by the penalized method but not for the non penalized method and Figure (7d) corresponds to cases where the data is assigned to a wrong cluster, for both methods.

6 Application to real-world data

In this section, we perform our proposed methodology FFMoE and PenFFMoE on two real-world data sets: Canadian Weather (CW, available in the R package [42]) and Cycling (available in the R package [43]).

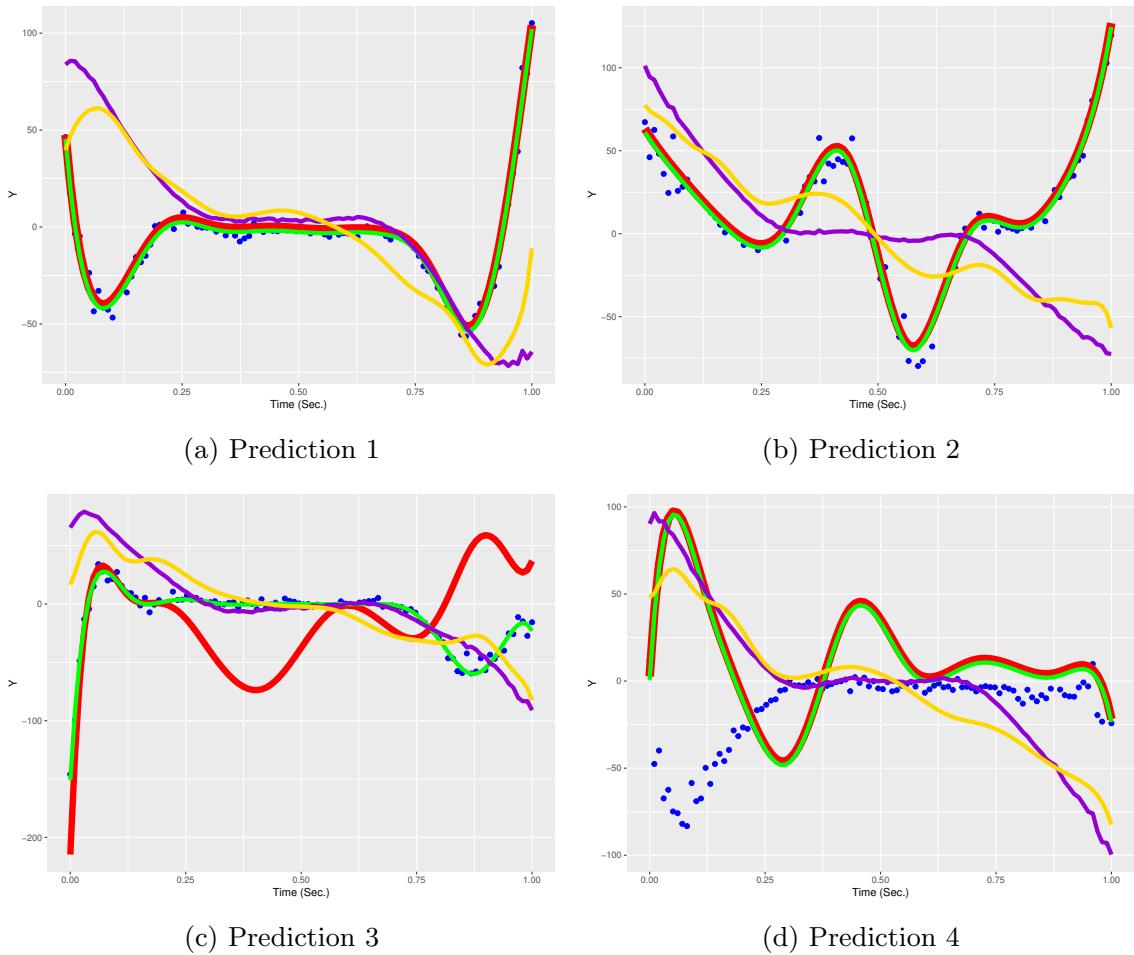


Fig. 7: observed data vs Fitted response functions for four chosen individuals on the test sample. Red and green lines match to FFMoE and PenFFMoE resp.; gold and violet lines are the prediction pffr and PenFFR resp.; the actual data is the blue dots.

In each of these data sets, the prediction accuracy of FFMoE and PenFFMoE is compared with the competitors PenFFR [11] and pffr [7]. Let us remark that PenFFR and pffr consider a single model and not a mixture as compared with FFMoE and PenFFMoE. Comparison is done by the leave-one-out cross-validation integrated square error (ISE):

$$\text{ISE}_i = \int_0^T (\mathbf{Y}_i(t) - \widehat{\mathbf{Y}}_i^{(-i)}(t))^2 dt,$$

where $\hat{Y}_i^{(-i)}(t)$ is the prediction of the i^{th} observation given by the model trained on a dataset of all the observations without the i^{th} one. Computationally, this criterion is

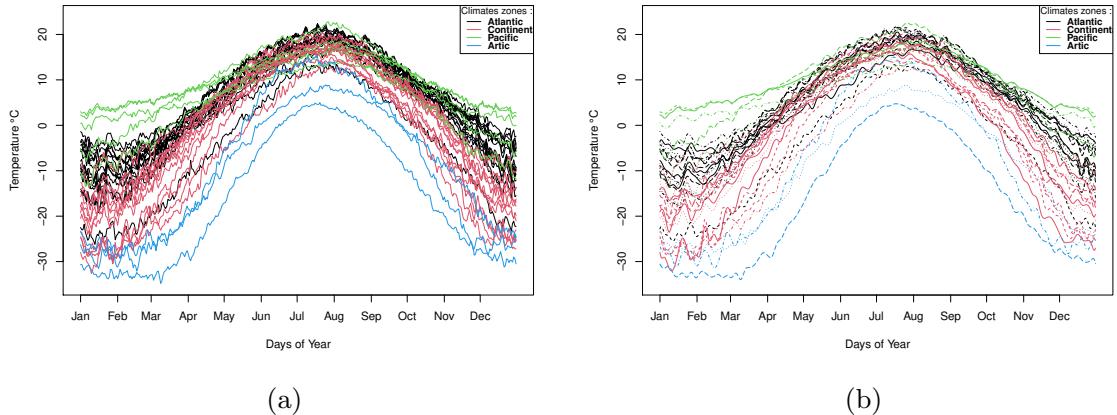


Fig. 8: 35 daily mean raw (a) and processed (b) temperature measurement curves.

approximated by the L^2 -norm between the actual and prediction values on a grid of values t is used as a surrogate. It is given by:

$$\widehat{\text{ISE}}_i = \sum_{j=1}^m (\mathbf{Y}_i(t_j) - \widehat{\mathbf{Y}}_i^{(-i)}(t_j))^2. \quad (26)$$

6.1 Canadian Weather data

The Canadian Weather data set consists of $m = 365$ daily temperature measurements (average over the year 1961 to 1994) at $n = 35$ weather stations in Canada, and their corresponding daily precipitation (in log scale). Our goal is to predict the (log) daily precipitations functions $\mathbf{Y}_i(t)$ using its corresponding temperatures $\mathbf{X}_i(t)$, for $t \in [0, 365]$.

Figure 8 displays the raw temperatures and their cubic B-splines smoothing with $L_X = 100$ basis functions and equispaced knots. Figure 9 shows the raw log precipitations profiles to predict.

Following the target to obtain smooth estimates of parameter curves (or surfaces) and accurate predictions, we must correctly choose the number of basis functions of functional parameters without forgetting that the number of parameters of the

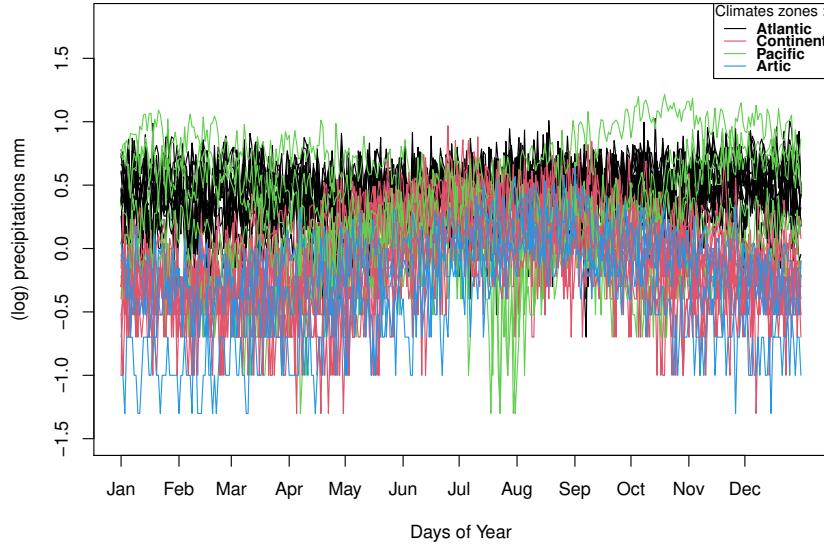


Fig. 9: raw log precipitations profiles of the 35 Canadian weather stations.

simple model is nearly multiplied by the number of components to get the number of parameters of MoE models. So we set L_β the number of basis functions to 8 both for FFMoE and PenFFMoE. And for the non-mixture models (PenFFR and pffr), we set L_β to 40. The penalty parameters λ_0 and λ_1 for the intercept and temperature effect are selected using cross-validation on a predefined grid of values (3 equispaced values between 0 and 0.5). Model selection is made using the BIC criterion for each LOO model with the number of expert components K in the set $\{1, 2, 3, 4, 5\}$. We observe in Table 3 that both for FFMoE and the PenFFMoE, the number of experts component the most often selected is $K = 4$. The same situation is observed between the two methods due to the fact that the cross-validation procedure leads to selecting mostly a null value of λ .

Table 4 shows the average value of $\widehat{\text{ISE}}_i$, the standard deviation and the median over the $n = 35$ weather stations. It is important to recall that the statistics are computed over LOO cross-validation, so on different model estimations (including the

Number of components	1	2	3	4	5
FFMoE	0%	0%	20.00%	51.43%	28.57%
PenFFMoE	0%	0%	11.43%	65.71%	22.86%

Table 3: Proportion of number of experts per model obtained by BIC selection.

Methods	average \widehat{ISE}	sd \widehat{ISE}	median \widehat{ISE}
PenFFMoE	29.91	21.07	22.83
FFMoE	30.00	30.37	21.17
PenFFR	36.40	40.42	21.04
pffr	89.51	52.06	71.22

Table 4: Average, standard deviation and median of \widehat{ISE}_i for the Canadian Weather data set. The best result is in boldface.

choice of K). We note a little enhancement in the predictive quality of the mixture models (PenFFMoE, FFMoE) compared to the models without mixture (PenFFR, pffr), and also a smaller inter-individual variance.

Another advantage of mixture models is the interpretation of the mixture component belonging. For this, new estimations of PenFFMoE and FFMoE are performed on the whole data set. The BIC criterion selects $K = 3$ components for FFMoE and $K = 4$ for PenFFMoE. Figure 10 shows the regression coefficient $\hat{\beta}_k(t)$ and gated network parameters $\hat{\alpha}_k(t)$ for the PenFFMoE version. Note that for the gated network parameters, we only have $K - 1$ curves due to the identifiability condition, which imposes that $\alpha_1(t) = 0$. We also observed that PenFFMoE parameters are slightly smoother than for FFMoE parameters (see Appendix D). This led to a better highlighting of all components of the impact of temperatures on precipitations at different times of the year.

Figure 11a plots the geographical positions of the stations. We note a high correlation with the four climate zones of Canada, which is confirmed by the confusion matrices given in Table (11b). Finally, Figure 12 gives predictions for two randomly chosen weather stations (Churchill and Edmonton) and are compared with the actual precipitation.

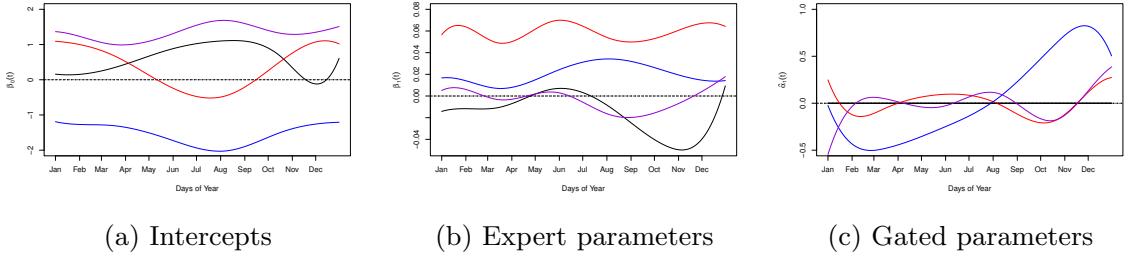


Fig. 10: Functional coefficients and gated network parameters obtained by PenFFMoE on Canadian Weather data. Color depends on group membership.

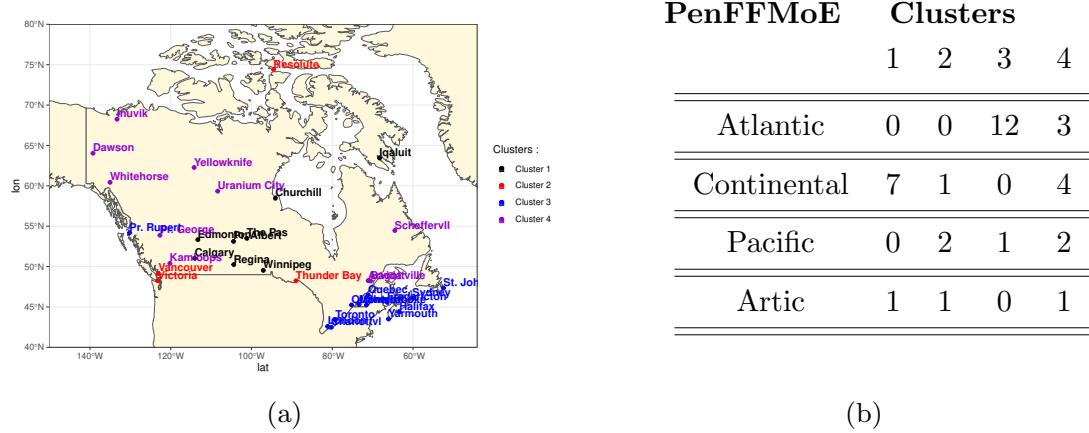


Fig. 11: Geographic visualization of the 35 weather stations clustering by PenFFMoE and confusion matrix between clusters and climates zones.

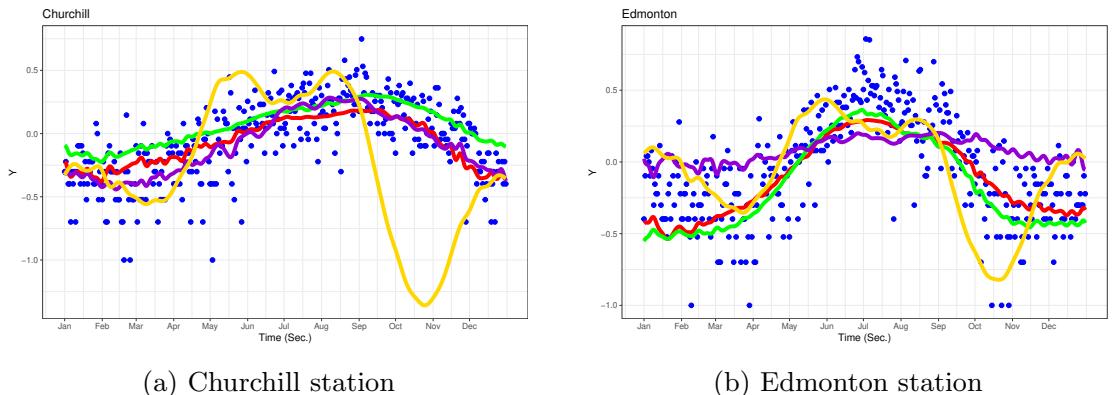


Fig. 12: Prediction for two randomly chosen stations. Blue points are the actual data, red and green lines are the predictions for FFMoE and PenFFMoE resp. The violet and gold lines are the predictions given by PenFFR and pffr resp.

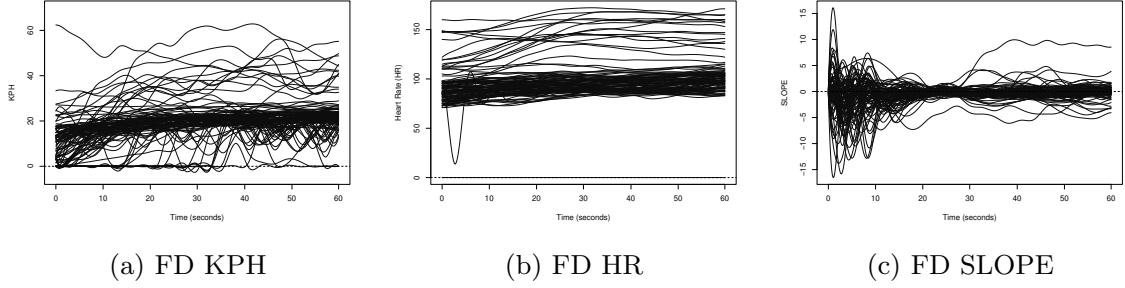


Fig. 13: Raw and functional expansion curves of speed (a,d), heart rate (b,e) and slope (c,f) for 100 cyclists.

6.2 Cycling Data

The Cycling data set, initially studied in [44], contains the measurements of several parameters during 216 cycling sessions of 30 minutes. The parameters are the power developed by the cyclist (in watts), its heart rate (in beats per minute), the pedalling cadence (in rotation per minute), the speed (in km/h), the slope (in percentage), the outdoor temperature (in Celsius degree) the altitude (in meters). The sampling rate is one measure per second. Our goal in this study is to predict the developed power according to the three parameters known to have an impact [44]: speed (KPH), heart rate (HR) and slope (SLOPE). Due to the high variability of these parameters during a period of 30 minutes, we restrict our analysis to a small portion of the curve, corresponding to the 20th-minute (chosen arbitrarily).

Figure 13 shows the functional expansion in cubic B-splines with $L_X = 50$ basis functions and equispaced knots for the three covariates. Figure 14a plots the developed power. Due to its dispersion, a logarithmic transformation is applied (Figure 14b).

We evaluate on this data set FFMoE, PenFFMoE, PenFFR and pffr. Predictive performances are evaluated through the ISE. The data set is split into train and test subsets with proportions 80% and 20%. The number of components for FFMoE, PenFFMoE is made using BIC with K in the set $\{1, 2, \dots, 15\}$. The number of basis functions of both expert parameters L_β and gated parameters L_α are set to 10. For the

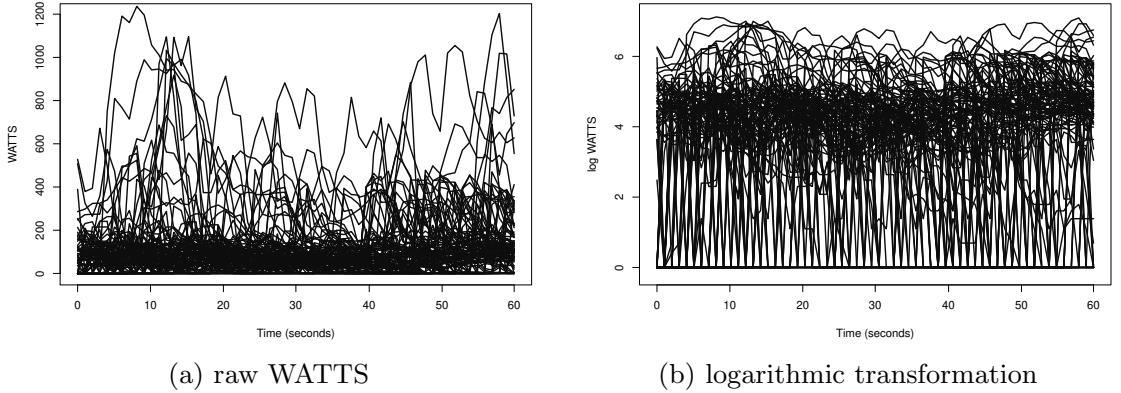


Fig. 14: Power developed by 100 cyclists and the corresponding logarithmic transformation.

non mixture models PenFFR and pffr, the number of basis functions for parameters are set to 15.

We obtained $K = 4$ for FFMoE and $K = 3$ for PenFFMoE, with a better BIC for FFMoE. Figure 15 shows the gated and expert parameters for PenFFMoE, which allows interesting interpretation. For instance, for the green cluster, the effect of the three features is almost constant, which means that the cyclist has a regular effort, with regular speed, heart rate and slope. On the contrary, for the blue cluster, the effect of KPH goes from positive to negative, whereas the effect of HR remains positive: probably that this session corresponds to an end of a climb: during the climb, the cyclist goes slowly whereas developing a high power and high HR, and then, after the summit of the climb, keeping a high power allows him to go fastly with a decreasing HR. Figure E3 in Appendix E shows the same results for FFMoE method.

Table 5 presents the average and standard deviation of ISE (over the test set) for the different models. If we consider the ISE averaged over the individuals of the test set, the best results are obtained with pffr. But looking at the median ISE, we conclude that most individuals are better predicted with the PenFFMoE method. This is in particular confirmed by Figure 16, which plots the predictions on two randomly

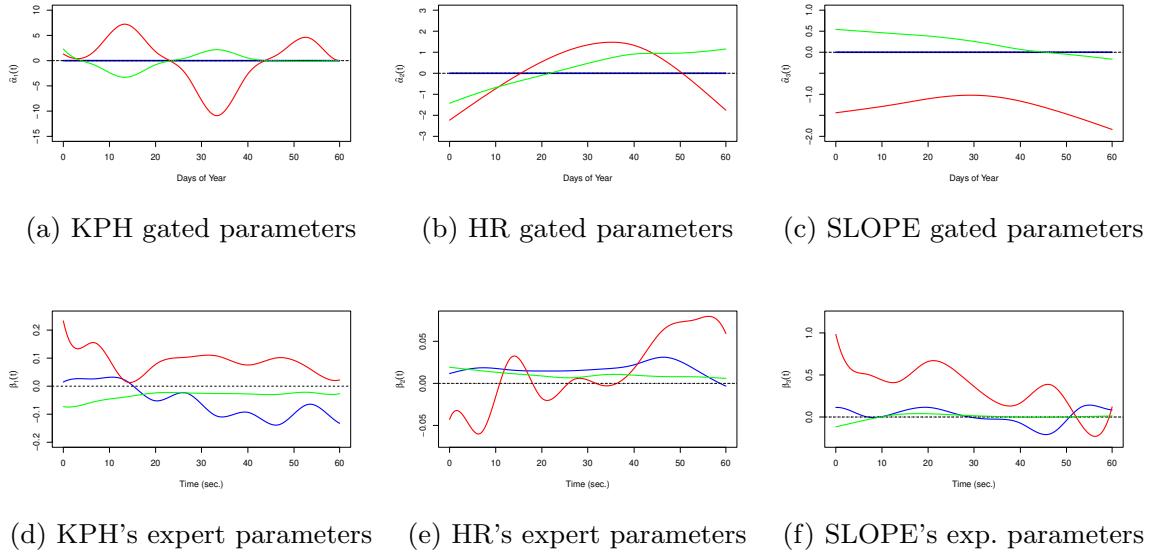


Fig. 15: Functional gated (first row) and expert (second row) parameters obtained by PenFFMoE on Cycling data. corresponding colors matched for the same cluster.

Methods	BIC	Nb clusters	Average ISE	sd \widehat{ISE}	median \widehat{ISE}
PenFFMoE	26729.9	3	160.72	225.64	34.63
FFMoE	26275.2	4	155.20	202.94	47.66
PenFFR		/	155.07	181.85	47.31
pffr		/	154.78	181.61	46.82

Table 5: The average and standard deviation of \widehat{ISE} for Cycling data set. The best result is in boldface.

chosen cycling sessions, on which we can see that the prediction with FFMoE and PenFFMoE better follow the general shape of the curves.

7 Conclusion

Functional data analysis has now reached a high level of maturity and its manifold applications span a wide range of scientific fields. In the present paper we developed a novel estimation scheme for MoE in the framework where both covariates and response are of functional type using the concurrent linear model with Gaussian error. Preliminary investigations based on plain maximum likelihood estimation, and using functional expansions in standard bases, lead us to the observation of a

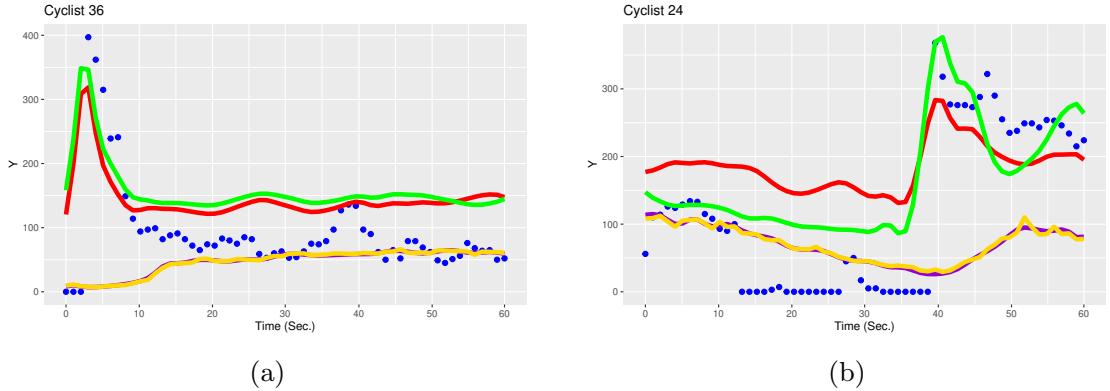


Fig. 16: Prediction on two randomly chosen cycling sessions. Blue points are the actual data, and red and green lines are the predictions for FFMoE and PenFFMoE resp. The violet and gold lines are the predictions given by PenFFR and pffr resp.

lack of smoothness of the estimators in various experiments with real-world datasets. In order to circumvent these issues, we introduced a ridge-type penalisation on the second derivatives and obtained a more stable estimator, still capable of handling substantial variability of first-order behaviours. Numerical experiments showed that the FFMoE (also PenFFMoE) has satisfactory behaviour in terms of parameter estimation (interpretability) and predictive accuracy on simulated datasets.

We then illustrate this performance on two real-world datasets. On Canadian weather dataset, PenFFMoE and FFMoE cluster the weather stations in $K = 4$ clusters that match the various climate zones. The predictive accuracy shows a definite advantage of mixture of experts over non-mixture based models. On Cycling data, the predictive quality is certainly not as good as non-mixture models, but it gives predictions that detect regime changes more easily.

Extensions of this work are potentially manifold. One possible avenue is to explore the more general exponential family on the functional response side. A second possible direction would be to investigate possible solutions for producing relevant prediction bounds, using for instance conformal prediction [45] which has attracted great interest lately in the machine learning community.

Appendix A EM for the FFMoE

Given the complete data log-likelihood and the parameters at current iteration l , we define the Q function for the EM algorithm defined by:

$$Q(\Psi^{(l+1)} | \Psi^{(l)}) = \mathbb{E}(\mathcal{L}_c(\Psi^{(l+1)}; \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}) | \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)})$$

Now we are going to describe the EM algorithm for maximizing (15):

- **E-step:**

At this step, we compute the conditional expectation of the log-likelihood given the observed data and the current parameter (at iteration l) estimation $\Psi^{(l)}$. This is equivalent to update the posterior probabilities $p_{ik}^{(l)}$ that the curves $\mathbf{x}_i(t)$ belongs to the k^{th} component of the mixture under the current model:

$$p_{ik}^{(l)} = \mathbb{E}(z_{ik} | \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)}) = \mathbb{P}(z_{ik} = 1 | \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)}).$$

Using Bayes' theorem, the conditional probability $p_{ik}^{(l)}$ can be expressed as:

$$\begin{aligned} p_{ik}^{(l)} &= \frac{\mathbb{P}(z_{ik} = 1) \mathbb{P}(\{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)} | z_{ik} = 1)}{\mathbb{P}(\{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)})} \\ &= \frac{\mathbb{P}(z_{ik} = 1) \mathbb{P}(\{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)} | z_{ik} = 1)}{\sum_{u=1}^K \mathbb{P}(z_{iu} = 1) \mathbb{P}(\{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)} | z_{iu} = 1)} \\ p_{ik}^{(l)} &= \frac{\pi_k(\mathbf{x}_i(t), \alpha_k^{(l)}(t)) \Phi_m(\mathbf{y}_i; b_k^{\top(l)} \mathbf{R}_i, V_{k,i}^{(l)})}{\sum_{u=1}^K \pi_u(\mathbf{x}_i(t), \alpha_u^{(l)}(t)) \Phi_m(\mathbf{y}_i; b_u^{\top(l)} \mathbf{R}_i, V_{u,i}^{(l)})} \end{aligned} \quad (\text{A1})$$

- **M-step:**

Given the previous posterior probability and the observed data, this step updates the current parameters $\Psi^{(l)}$ by maximizing the complete (data) log-likelihood, that

is $\Psi^{(l+1)}$:

$$\begin{aligned}
Q(\Psi^{(l+1)} | \Psi^{(l)}) &= \mathbb{E}(\mathcal{L}_c(\Psi^{(l+1)}; \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}) | \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)}) \\
&= \mathbb{E}\left(\sum_{i=1}^n \sum_{k=1}^K z_{ik} \log\left(\frac{\exp(a_k^{(l+1)^\top} r_i)}{1 + \sum_{u=1}^{K-1} \exp(a_u^{(l+1)^\top} r_i)} \cdot \frac{1}{\sqrt{(2\pi)^m |\mathbf{V}_{k,i}^{(l+1)}|}}\right.\right. \\
&\quad \left.\left. \exp(-\frac{1}{2}(\mathbf{y}_i - b_k^{(l+1)^\top} \mathbf{R}_i)^\top \mathbf{V}_{k,i}^{-1(l+1)} (\mathbf{y}_i - b_k^{(l+1)^\top} \mathbf{R}_i))\right) \right. \\
&\quad \left. \left|\{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)}\right.\right. \\
&= \sum_{i=1}^n \sum_{k=1}^K \mathbb{E}(z_{ik} | \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)}) \log\left(\frac{\exp(a_k^{(l+1)^\top} r_i)}{1 + \sum_{u=1}^{K-1} \exp(a_u^{(l+1)^\top} r_i)}\right. \\
&\quad \left.\frac{1}{\sqrt{(2\pi)^m |\mathbf{V}_{k,i}^{(l+1)}|}} \exp(-\frac{1}{2}(\mathbf{y}_i - b_k^{(l+1)^\top} \mathbf{R}_i)^\top \mathbf{V}_{k,i}^{-1(l+1)} (\mathbf{y}_i - b_k^{(l+1)^\top} \mathbf{R}_i))\right) \\
&= \underbrace{\sum_{i=1}^n \sum_{k=1}^K p_{ik}^{(l)} \log\left(\frac{\exp(a_k^{(l+1)^\top} r_i)}{1 + \sum_{u=1}^{K-1} \exp(a_u^{(l+1)^\top} r_i)}\right)}_{Q_1(a_k^{(l+1)} | \Psi^{(l)})} + \\
&\quad \underbrace{\sum_{i=1}^n \sum_{k=1}^K p_{ik}^{(l)} \log\left(\frac{1}{\sqrt{(2\pi)^m |\mathbf{V}_{k,i}^{(l+1)}|}} \exp(-\frac{1}{2}(\mathbf{y}_i - b_k^{(l+1)^\top} \mathbf{R}_i)^\top \mathbf{V}_{k,i}^{-1(l+1)} (\mathbf{y}_i - b_k^{(l+1)^\top} \mathbf{R}_i))\right)}_{Q_2(b_k^{(l+1)}, \mathbf{V}_k^{(l+1)} | \Psi^{(l)})} \\
Q(\Psi^{(l+1)} | \Psi^{(l)}) &= Q_1(a_k^{(l+1)} | \Psi^{(l)}) + Q_2(b_k^{(l+1)}, \mathbf{V}_k^{(l+1)} | \Psi^{(l)}).
\end{aligned}$$

The global maximization problem is split onto two separate maximization problems: the updating of gated network parameters via the maximization of the function $Q_1(a_k^{(l+1)} | \Psi^{(l)})$ and the updating of experts parameters via the maximization of the function $Q_2(b_k^{(l+1)}, \sigma_k^{(l+1)} | \Psi^{(l)})$. It is obvious to recognise in each of these two expressions the likelihood of the multinomial logistic model $Q_1(\cdot)$ and the linear

gaussian model $Q_2(\cdot)$ for which we know how to calculate (at least numerically through Newton-Raphson method for e.g) MLEs.

Appendix B EM for PenFFMoE

- **E-step:**

Same as in non penalize case

- **M-step:**

Given the previous posterior probability and the observed data, this step updates the current parameters $\Psi^{(l)}$ by maximizing the penalized complete (data) log-likelihood, that is $\Psi^{(l+1)}$. We define:

$$\begin{aligned}
Q_{pen}(\Psi^{(l+1)} | \Psi^{(l)}) &= \mathbb{E}(\mathcal{L}_{pen}^c(\Psi^{(l+1)}; \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}) \mid \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)}) \\
&= \mathbb{E}\left(\sum_{i=1}^n \sum_{k=1}^K z_{ik} \log\left(\frac{\exp(a_k^{(l+1)\top} r_i)}{1 + \sum_{u=1}^{K-1} \exp(a_u^{(l+1)\top} r_i)} \cdot \frac{1}{\sqrt{(2\pi)^m |\mathbf{V}_{k,i}^{(l+1)}|}}\right.\right. \\
&\quad \left.\left. \exp(-\frac{1}{2}(\mathbf{y}_i - b_k^{(l+1)\top} \mathbf{R}_i)^\top \mathbf{V}_{k,i}^{-1(l+1)} (\mathbf{y}_i - b_k^{(l+1)\top} \mathbf{R}_i))) - \right. \\
&\quad \left. \sum_{k=1}^K b_k^{(l+1)\top} (\lambda_k \mathbf{P}) b_k^{(l+1)} - \sum_{k=1}^{K-1} a_k^{(l+1)\top} (\gamma_k \mathbf{Q}) a_k^{(l+1)} \right. \\
&\quad \left. \mid \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)}\right) \\
&= \sum_{i=1}^n \sum_{k=1}^K \mathbb{E}(z_{ik} \mid \{y_i(t_j), \mathbf{x}_i(t_j)\}_{i,j}; \Psi^{(l)}) \log\left(\frac{\exp(a_k^{(l+1)\top} r_i)}{1 + \sum_{u=1}^{K-1} \exp(a_u^{(l+1)\top} r_i)}\right. \\
&\quad \left. \frac{1}{\sqrt{(2\pi)^m |\mathbf{V}_{k,i}^{(l+1)}|}} \exp(-\frac{1}{2}(\mathbf{y}_i - b_k^{(l+1)\top} \mathbf{R}_i)^\top \mathbf{V}_{k,i}^{-1(l+1)} (\mathbf{y}_i - b_k^{(l+1)\top} \mathbf{R}_i)))\right) \\
&\quad - \sum_{k=1}^K b_k^{(l+1)\top} (\lambda_k \mathbf{P}) b_k^{(l+1)} - \sum_{k=1}^{K-1} a_k^{(l+1)\top} (\gamma_k \mathbf{Q}) a_k^{(l+1)}
\end{aligned}$$

$$\begin{aligned}
&= \underbrace{\sum_{i=1}^n \sum_{k=1}^K p_{ik}^{(l)} \log \left(\frac{\exp(a_k^{(l+1)^\top} r_i)}{1 + \sum_{u=1}^{K-1} \exp(a_u^{(l+1)^\top} r_i)} \right)}_{Q_1(a_k^{(l+1)} | \Psi^{(l)})} - \sum_{k=1}^{K-1} a_k^{(l+1)^\top} (\gamma_k Q) a_k^{(l+1)} - \\
&\quad \sum_{k=1}^K b_k^{(l+1)^\top} (\lambda_k P) b_k^{(l+1)} + \\
&\underbrace{\sum_{i=1}^n \sum_{k=1}^K p_{ik}^{(l)} \log \left(\frac{1}{\sqrt{(2\pi)^m |\mathbf{V}_{k,i}^{(l+1)}|}} \exp \left(-\frac{1}{2} (\mathbf{y}_i - b_k^{(l+1)^\top} \mathbf{R}_i)^\top \mathbf{V}_{k,i}^{-1(l+1)} (\mathbf{y}_i - b_k^{(l+1)^\top} \mathbf{R}_i) \right) \right)}_{Q_2(b_k^{(l+1)}, \mathbf{V}_k^{(l+1)} | \Psi^{(l)})}
\end{aligned}$$

$$\begin{aligned}
Q(\Psi^{(l+1)} | \Psi^{(l)}) &= Q_1(a_k^{(l+1)} | \Psi^{(l)}) - \underbrace{\sum_{k=1}^{K-1} a_k^{(l+1)^\top} (\gamma_k Q) a_k^{(l+1)}}_{Q_2(b_k^{(l+1)}, \mathbf{V}_k^{(l+1)} | \Psi^{(l)})} + \\
&\quad \underbrace{\sum_{k=1}^K b_k^{(l+1)^\top} (\lambda_k P) b_k^{(l+1)}}_{Q_2(b_k^{(l+1)}, \mathbf{V}_k^{(l+1)} | \Psi^{(l)})} \\
&= Q_{1,pen}(a_k^{(l+1)} | \Psi^{(l)}) + Q_{2,pen}(b_k^{(l+1)}, \sigma_k^{2(l+1)} | \Psi^{(l)}).
\end{aligned}$$

Appendix C Parameters in simulation study

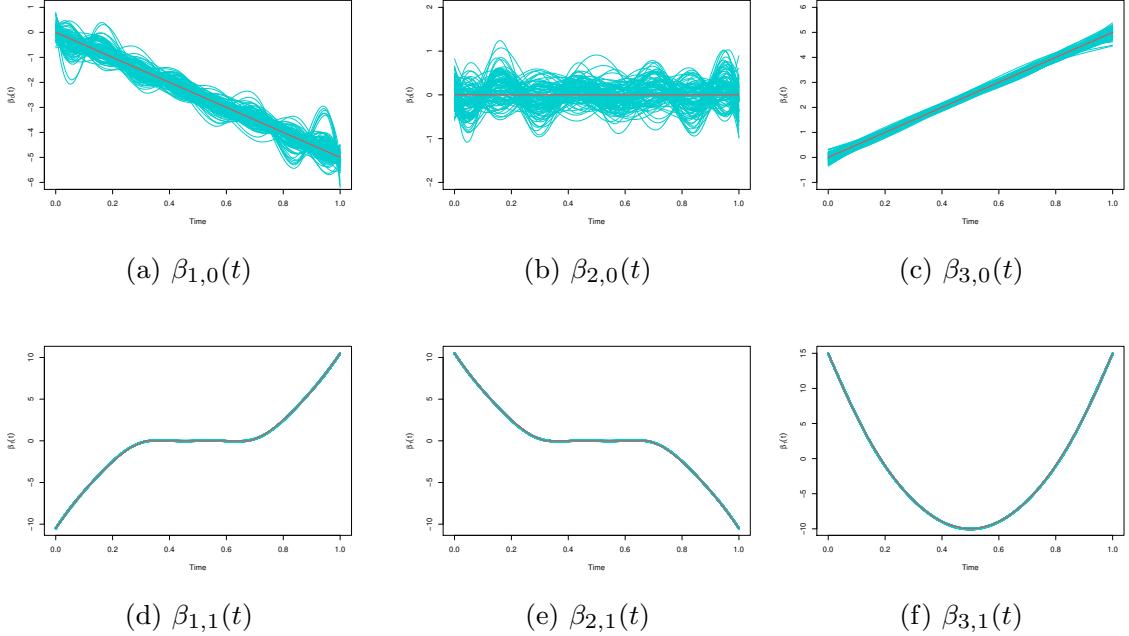


Fig. C1: Estimation of the regression coefficients for Scenario S3 with PenFFMoE. The red curves are the actual parameters, the cyan curves are the estimation.

Appendix D Estimators for Canadian weather data

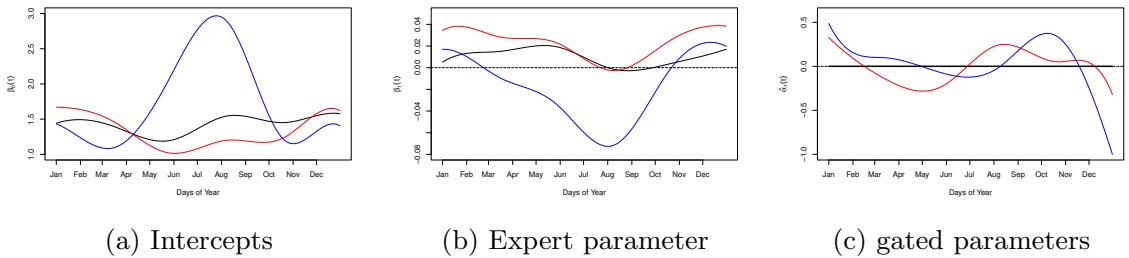


Fig. D2: Functional coefficients and gated network parameters obtained by FFMoE on Canadian Weather data. Color depends on group membership.

Appendix E Estimators for Cycling data

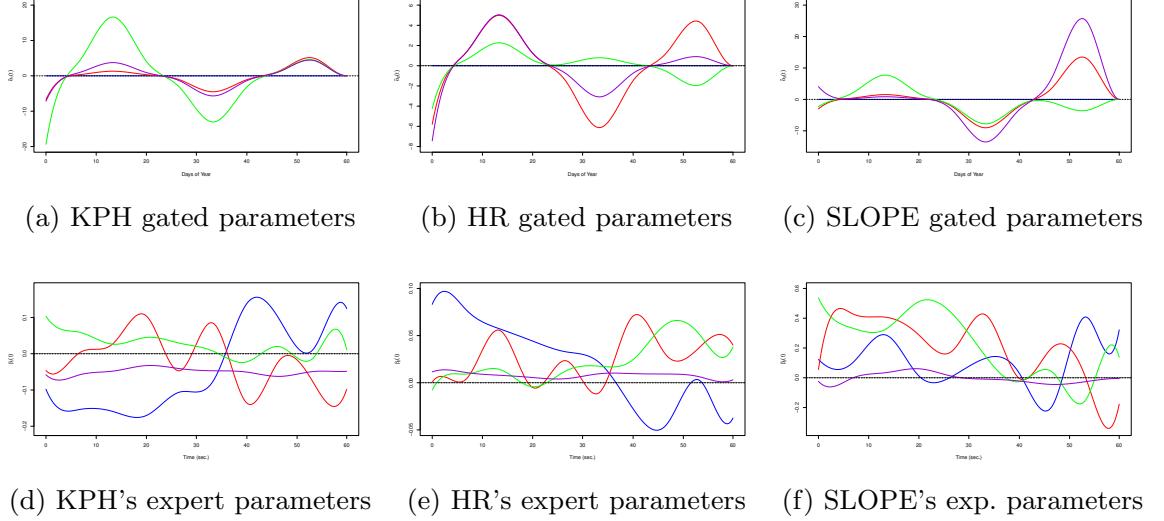


Fig. E3: Functional gated (first row) and expert (second row) parameters obtained by FFMoE on Cycling data. corresponding colors matched for the same cluster.

References

- [1] Ramsay, J., Silverman, B.W.: Functional Data Analysis, 2nd edn. Springer Series in Statistics. Springer, New York (2005). <https://doi.org/10.1007/b98888>
- [2] Ramsay, J.O., Hooker, G., Graves, S.: Functional Data Analysis with R and MATLAB, 1st edn. Springer, New York (2009)
- [3] Horváth, L., Kokoszka, P.: Inference for Functional Data with Applications. Springer Series in Statistics. Springer, New York (2012). <https://doi.org/10.1007/978-1-4614-3655-3> . <https://www.springer.com/gp/book/9781461436546>
Accessed 2021-05-27
- [4] Kokoszka, P., Reimherr, M.: Introduction to Functional Data Analysis, (2017).
<https://doi.org/10.1201/9781315117416>
- [5] Goldsmith, J., Bobb, J., Crainiceanu, C.M., Caffo, B., Reich, D.: Penalized functional regression. Journal of Computational and Graphical Statistics **20**(4), 830–851 (2011) <https://doi.org/10.1198/jcgs.2010.10007>
<https://doi.org/10.1198/jcgs.2010.10007>
- [6] Morris, J.: Functional regression. Annual Review of Statistics and Its Application **2** (2014) <https://doi.org/10.1146/annurev-statistics-010814-020413>
- [7] Ivanescu, A., Staicu, A.-M., Scheipl, F., Greven, S.: Penalized function-on-function regression. Computational Statistics **30**(2), 539–568 (2015) <https://doi.org/10.1007/s00180-014-0548-4>
- [8] Luo, R., Qi, X., Wang, Y.: Functional wavelet regression for linear function-on-function models. Electronic Journal of Statistics **10**(2), 3179–3216 (2016) <https://doi.org/10.1214/16-EJS1204>

- [9] Li, Y., Ruppert, D.: On the asymptotics of penalized splines. *Biometrika* **95**(2), 415–436 (2008)
- [10] James, G.M., Wang, J., Zhu, J.: Functional linear regression that's interpretable. *Annals of Statistics* **37**(5A), 2083–2108 (2009)
- [11] Tamo Tchomgui, J.S., Jacques, J., Barriac, V., Fraysse, G., Chrétien, S.: A Penalized Spline Estimator for Functional Linear Regression with Functional Response. working paper or preprint (2023). <https://hal.science/hal-04120709>
- [12] DeSarbo, W.S., Cron, W.L.: A maximum likelihood methodology for clusterwise linear regression. *Journal of Classification* **5**(2), 249–282 (1988)
- [13] McLachlan, G.J., Peel, D.: Finite Mixture Models. Probability and Statistics – Applied Probability and Statistics Section, vol. 299. Wiley, New York (2000)
- [14] Makov, E., Titterington, D.M., Smith, A.F.M.: Statistical Analysis of Finite Mixture Distributions, 1st edn. wiley, New-York (1985)
- [15] Lindsay, B.G.: Mixture Models: Theory, Geometry, and Applications. NSF-CBMS regional conference series in probability and statistics, Institute of Mathematical Statistics (1995). <https://books.google.fr/books?id=VFDzNhikFbQC>
- [16] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* **39**(1), 1–22 (1977)
- [17] McLachlan, G.J., Krishnan, T.: The EM Algorithm and Extensions. Wiley, Series in Probability and Statistics (2007). <https://books.google.fr/books?id=NBawzaWoWa8C>
- [18] Celeux, G., Chrétien, S., Forbes, F., Mkhadri, A.: A component-wise em algorithm

for mixtures. *Journal of Computational and Graphical Statistics* **10**(4), 697–712 (2001)

- [19] Chrétien, S., Hero, A.: Acceleration of the em algorithm via proximal point iterations. In: Proceedings. 1998 IEEE International Symposium on Information Theory (Cat. No. 98CH36252), p. 444 (1998). IEEE
- [20] Chrétien, S., Hero, A.O.: Kullback proximal algorithms for maximum-likelihood estimation. *IEEE transactions on information theory* **46**(5), 1800–1810 (2000)
- [21] Chrétien, S., Hero, A., Perdry, H.: Space alternating penalized kullback proximal point algorithms for maximizing likelihood with nondifferentiable penalty. *Annals of the Institute of Statistical Mathematics* **64**, 791–809 (2012)
- [22] Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive Mixtures of Local Experts. *Neural Computation* **3**(1), 79–87 (1991) <https://doi.org/10.1162/neco.1991.3.1.79>
- [23] Nguyen, H.D., Chamroukhi, F.: Practical and theoretical aspects of mixture-of-experts modeling: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1246 (2018) <https://doi.org/10.1002/widm.1246>
- [24] Chamroukhi, F., Pham, N.T., Hoang, V.H., McLachlan, G.J.: Functional mixtures-of-experts. *Statistics and Computing* **34**(98) (2024)
- [25] Hida, T., Hui-Hsiung, K., Potthoff, J., Streit, L.: White Noise: An Infinite Dimensional Calculus. *Mathematics and its applications*, Kluwer Academic Publishers (1993). <https://books.google.fr/books?id=-XitQgAACAAJ>
- [26] Hastie, T., Tibshirani, R.: Varying-coefficient models. *Journal of the Royal Statistical Society. Series B (Methodological)* **55**(4), 757–796 (1993). Accessed 2022-12-13

- [27] Wood, S.N.: On confidence intervals for generalized additive models based on penalized regression splines. *Australian & New Zealand Journal of Statistics* **48**(4), 445–464 (2006)
- [28] Dayton, C.M., Macready, G.B.: Concomitant-variable latent-class models. *Journal of the American Statistical Association* **83**(401), 173–178 (1988)
- [29] Young, D.S., Hunter, D.R.: Mixtures of regressions with predictor-dependent mixing proportions. *Computational Statistics & Data Analysis* **54**(10), 2253–2266 (2010) <https://doi.org/10.1016/j.csda.2010.04.002>
- [30] Mousavi, S., Sørensen, H.: Functional logistic regression: a comparison of three methods. *Journal of Statistical Computation and Simulation* **88**, 1–19 (2018) <https://doi.org/10.1080/00949655.2017.1386664>
- [31] Berrendero, J.R., Bueno-Larraz, B., Cuevas, A.: On functional logistic regression: some conceptual issues. *TEST: An Official Journal of the Spanish Society of Statistics and Operations Research* **32**(1), 321–349 (2023) <https://doi.org/10.1007/s11749-022-00836->
- [32] Jiang, W., Tanner, M.A.: Hierarchical mixtures-of-experts for exponential family regression models: Approximation and maximum likelihood estimation. *Annals of Statistics* **27**(3), 987–1011 (1999) <https://doi.org/10.1214/aos/1018031265>
- [33] Jordan, M., Jacobs, R.: Hierarchical mixtures of experts and the. *Neural computation* **6**, 181 (1994)
- [34] Peng, F., Jacobs, R.A., Tanner, M.A.: Bayesian inference in mixtures-of-experts and hierarchical mixtures-of-experts models with an application to speech recognition. *Journal of the American Statistical Association* **91**(435), 953–960 (1996). Accessed 2023-10-03

- [35] Neal, R.M., Hinton, G.E.: A view of the em algorithm that justifies incremental, sparse, and other variants. In: Learning in Graphical Models, Springer, pp. 355–368 (1998)
- [36] Fessler, J.A., Hero, A.O.: Space-alternating generalized expectation-maximization algorithm. IEEE Transactions on signal processing **42**(10), 2664–2677 (1994)
- [37] Grün, B., Leisch, F.: Flexmix version 2: Finite mixtures with concomitant variables and varying and constant parameters. Journal of Statistical Software **28**(4), 1–35 (2008) <https://doi.org/10.18637/jss.v028.i04>
- [38] Grün, B., Leisch, F.: Identifiability of Finite Mixtures of Multinomial Logit Models with Varying and Fixed Effects. Journal of Classification **25**(2), 225–247 (2008) <https://doi.org/10.1007/s00357-008-9022-8>
- [39] Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control **19**(6), 716–723 (1974) <https://doi.org/10.1109/TAC.1974.1100705>
- [40] Schwarz, G.: Estimating the Dimension of a Model. Annals of Statistics **6**(2), 461–464 (1978) <https://doi.org/10.1214/aos/1176344136>
- [41] Leurgans, S.E., Moyeed, R.A., Silverman, B.W.: Canonical correlation analysis when the data are curves. Journal of the Royal Statistical Society **55**(3), 725–740 (1993) <https://doi.org/10.1111/j.2517-6161.1993.tb01936.x>
- [42] Ramsay, J.O., Graves, S., Hooker, G.: Fda: Functional Data Analysis. (2022). R package version 6.0.5. <https://CRAN.R-project.org/package=fda>
- [43] Samardzic, S.: FREG: Functional Regression Models. (2022). R package version 1.1

- [44] Jacques, J., Samardzic, S.: Analyzing cycling sensors data through ordinal logistic regression with functional covariates. *Journal of the Royal Statistical Society* **71**(4), 969–986 (2022)
- [45] Angelopoulos, A.N., Bates, S.: A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification (2022)

Qing Mai

Material list:

Qing Mai (2024) Model-based tensor low-rank clustering. WG Slides.

Model-based Tensor Low-rank Clustering

Qing Mai

Department of Statistics
Florida State University

Working Group on Model-Based Clustering

July 23, 2024

Based on joint work with Kai Deng, Junge Li and Xin Zhang.

Outline

1 Background

2 Model

3 Estimation

4 Real Data

Tensor

We are interested in

- Data $\mathbf{X}_i, i = 1, \dots, n$, where $\mathbf{X}_i \in \mathbb{R}^{p_1 \times \dots \times p_M}$, $M \geq 2$
- Goal: Efficiently group the tensor data into K clusters

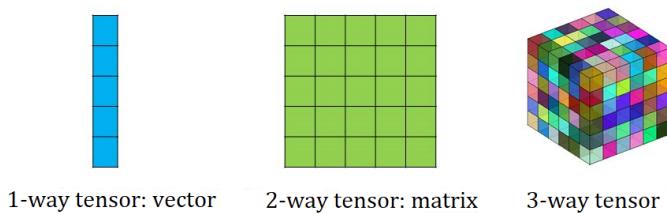
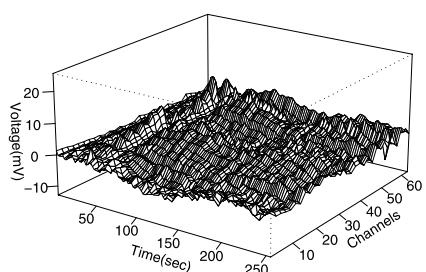
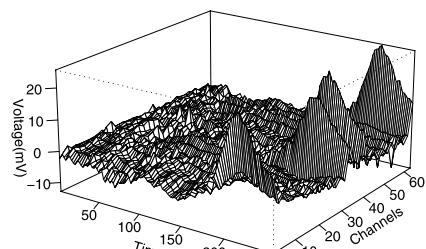
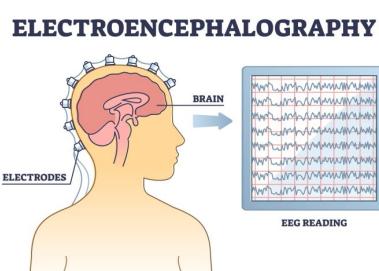


Figure: Graphical illustration of tensors. Source: Zhang & Han (2019).

A Motivating Example



- 64 Electrodes are placed on human scalps to measure voltage variation.
- Measured at 256 time points.
- 122 subjects in total:
 - 77 alcoholic subjects;
 - 45 non-alcoholic subjects.

Picture source: Left: Simple Psychology; Right: Li et al. (2010, AoS).

Background ○○○	Model ●○○○○	Estimation ○○○○	Real Data ○○○○○
-------------------	----------------	--------------------	--------------------

Outline

1 Background

2 Model

3 Estimation

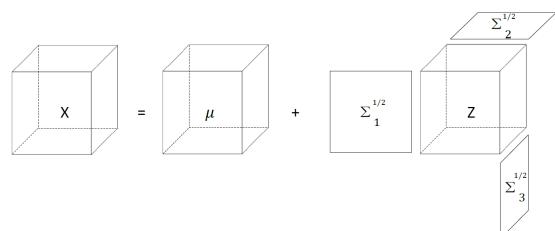
4 Real Data

Background ○○○	Model ○●○○○	Estimation ○○○○	Real Data ○○○○○
-------------------	----------------	--------------------	--------------------

Tensor Gaussian Mixture Model (TGMM)

Tensor normal distribution

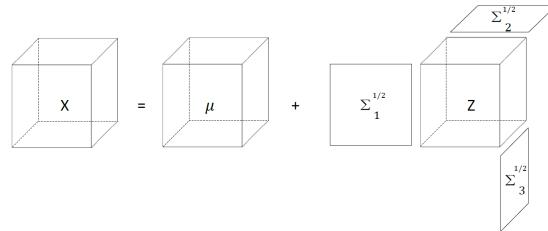
$$\mathbf{X} \sim \text{TN}(\boldsymbol{\mu}, \Sigma_1, \dots, \Sigma_M) \Leftrightarrow \\ \mathbf{X} = \boldsymbol{\mu} + [\mathbf{Z}; \Sigma_1^{1/2}, \dots, \Sigma_M^{1/2}]$$



Tensor Gaussian Mixture Model (TGMM)

Tensor normal distribution

$$\mathbf{X} \sim \text{TN}(\boldsymbol{\mu}, \Sigma_1, \dots, \Sigma_M) \Leftrightarrow \\ \mathbf{X} = \boldsymbol{\mu} + [\mathbf{Z}; \Sigma_1^{1/2}, \dots, \Sigma_M^{1/2}]$$



■ Assume

$$\mathbf{X}_i \stackrel{\text{i.i.d.}}{\sim} \sum_{k=1}^K \pi_k \text{TN}(\boldsymbol{\mu}_k, \Sigma_1, \dots, \Sigma_M).$$

- $0 < \pi_k < 1$: prior probability, $\sum_{k=1}^K \pi_k = 1$;
- $\boldsymbol{\mu}_k \in \mathbb{R}^{p_1 \times \dots \times p_M}$: mean of the k th cluster;
- $\Sigma_m \in \mathbb{R}^{p_m \times p_m}$: mode- m covariance.

■ Let $Y_i \in \{1, \dots, K\}$ be the **latent** label of \mathbf{X}_i . Then,

$$\Pr(Y_i = k) = \pi_k, \quad \mathbf{X}_i | (Y_i = k) \sim \text{TN}(\boldsymbol{\mu}_k, \Sigma_1, \dots, \Sigma_M).$$

Tensor Gaussian Mixture Model (TGMM)

■ A probabilistic model:

- Captures the correlation structure with a reduced number of parameters:

$$O(\prod_{m=1}^M p_m^2) \Rightarrow O(\sum_{m=1}^M p_m^2);$$

- Clear interpretation.

■ In the EEG data:

if two electrodes have stronger interactions at a certain time point than other electrodes, they must have stronger interactions across all time points.

Tensor Gaussian Mixture Model (TGMM)

The Optimal clustering coefficient

$$\mathbf{B}_k = [\![\boldsymbol{\mu}_k - \boldsymbol{\mu}_1; \boldsymbol{\Sigma}_1^{-1}, \dots, \boldsymbol{\Sigma}_M^{-1}]\!], \quad k = 2, \dots, K.$$

■ Optimal clustering rule

$$\psi^{\text{Opt}}(\mathbf{X}_i) = \arg \max_{k=2,\dots,K} \{\log(\pi_k/\pi_1) + \langle \mathbf{X}_i - \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_k), \mathbf{B}_{\textcolor{red}{k}} \rangle\}.$$

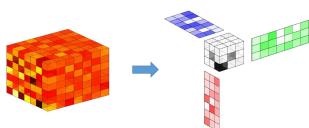
■ Conditional probability

$$\begin{aligned} \xi_{ik} &\equiv P(Y_i = k | \mathbf{X} = \mathbf{X}_i) \\ &= \begin{cases} \frac{\pi_1}{\pi_1 + \sum_{l=2}^K \pi_l \exp\left\{\langle \mathbf{X}_i - \frac{1}{2}(\mu_1 + \mu_l), \mathbf{B}_l \rangle\right\}}, & k = 1, \\ \frac{\pi_k \exp\left\{\langle \mathbf{X}_i - \frac{1}{2}(\mu_1 + \mu_k), \mathbf{B}_k \rangle\right\}}{\pi_1 + \sum_{l=2}^K \pi_l \exp\left\{\langle \mathbf{X}_i - \frac{1}{2}(\mu_1 + \mu_l), \mathbf{B}_l \rangle\right\}}, & 1 < k \leq K. \end{cases} \end{aligned}$$

Tensor Low-rank Mixture Model (TLMM)

- The mean contrasts admit Tucker low-rank decompositions

$$\mu_k - \bar{\mu} = [\![\mathcal{G}_k; \mathbf{A}_1, \dots, \mathbf{A}_M]\!], \quad \text{where } \bar{\mu} = \sum_{k=1}^K \pi_k \mu_k.$$



- ▶ The optimal clustering coefficient is

$$\mathbf{B}_k = [\Phi_k; \mathbf{D}_1, \dots, \mathbf{D}_M], \quad k = 2, \dots, K,$$
 - ▶ $\Phi_k = \mathcal{G}_k - \mathcal{G}_1, \mathbf{D}_m = \Sigma_m^{-1} \mathbf{A}_m.$
 - ▶

$$\langle \mathbf{X}_i, \mathbf{B}_k \rangle = \langle \tilde{\mathbf{X}}_i, \Phi_k \rangle$$

where

$$\tilde{\mathbf{X}}_i = [\mathbf{X}_i; \mathbf{D}_1^T, \dots, \mathbf{D}_M^T] \in \mathbb{R}^{r_1 \times \dots \times r_M}.$$
 - ▶ Further assume \mathbf{D}_m are sparse.

Figure: Tucker decomposition.

Number of free parameters:

Number of free parameters:
 $O(\prod_{m=1}^M r_m + \sum_{m=1}^M r_m(p_m - r_m))$. Source: Zhang & Han (2019)



Outline

1 Background

2 Model

3 Estimation

4 Real Data



The EM-Algorithm

Lemma

Under the TLMM model, maximizers of the Q-function satisfy

$$\begin{aligned}\tilde{\Sigma}_m^{(t+1)} &= \frac{1}{nq_m} \sum_{i=1}^n \sum_{k=1}^K \tilde{\xi}_{ik}^{(t+1)} \left(\mathbf{X}_i - \tilde{\mu}_k^{(t+1)} \right)_{(m)} \left(\tilde{\Sigma}_m^{(t+1)} \right)^{-1} \left(\mathbf{X}_i - \tilde{\mu}_k^{(t+1)} \right)_{(m)}^T, \\ \tilde{\mathbf{A}}_m^{(t+1)} &= \arg \max_{\mathbf{A}_m^T \mathbf{A}_m = \mathbf{I}_{rm}} \text{tr} \left(\tilde{\mathbf{H}}_{1m}^{(t+1)} \mathbf{A}_m \right) - \frac{1}{2} \text{tr} \left(\tilde{\mathbf{H}}_{2m}^{(t+1)} \mathbf{A}_m^T \left(\tilde{\Sigma}_m^{(t+1)} \right)^{-1} \mathbf{A}_m \right), \\ \tilde{\mathbf{g}}_k^{(t+1)} &= \frac{1}{\tilde{n}_k^{(t+1)}} \sum_{i=1}^n \tilde{\xi}_{ik}^{(t+1)} [\![\mathbf{X}_i; \tilde{\mathbf{J}}_1^{(t+1)}, \dots, \tilde{\mathbf{J}}_M^{(t+1)}]\!], \quad m = 1, \dots, M, \quad k = 1, \dots, K.\end{aligned}$$



Limitations:

- Computationally intensive due to sub-iterations
- Nonconvex optimization over Stiefel manifolds
- Lack of sparsity

Low-rankness Enhanced EM-Algorithm

The EM algorithm:

$$\begin{aligned}\tilde{\Sigma}_m^{(t+1)} &= \frac{1}{nq_m} \sum_{i=1}^n \sum_{k=1}^K \tilde{\xi}_{ik}^{(t+1)} \left(\mathbf{X}_i - \tilde{\mu}_k^{(t+1)} \right)_{(m)} \left(\tilde{\Sigma}_m^{(t+1)} \right)^{-1} \left(\mathbf{X}_i - \tilde{\mu}_k^{(t+1)} \right)_{(m)}^T, \\ \tilde{\mathbf{A}}_m^{(t+1)} &= \arg \max_{\mathbf{A}_m^T \mathbf{A}_m = \mathbf{I}_{rm}} \text{tr} \left(\tilde{\mathbf{H}}_{1m}^{(t+1)} \mathbf{A}_m \right) - \frac{1}{2} \text{tr} \left(\tilde{\mathbf{H}}_{2m}^{(t+1)} \mathbf{A}_m^T \left(\tilde{\Sigma}_m^{(t+1)} \right)^{-1} \mathbf{A}_m \right).\end{aligned}$$

Low-rankness Enhanced EM-Algorithm (LEEM)

- $\hat{\Sigma}_m^{(t+1)} = \frac{1}{nq_m} \sum_{i=1}^n \sum_{k=1}^K \tilde{\xi}_{ik}^{(t+1)} \left(\mathbf{X}_i - \hat{\mu}_k^{(t+1)} \right)_{(m)} \left(\mathbf{X}_i - \hat{\mu}_k^{(t+1)} \right)_{(m)}^T$.
- Decompose $\hat{\mu}_k^{(t)} - \hat{\mu}^{(t)}$ to obtain $\hat{\mathbf{A}}_m^{(t+1)}$.
- Update $\hat{\mathbf{D}}_m^{(t+1)}$:

$$\min_{\mathbf{D} \in \mathbb{R}^{p_m \times r_m}} \left\{ \text{tr} \left(\frac{1}{2} \mathbf{D}^T \hat{\Sigma}_m^{(t)} \mathbf{D} - (\hat{\mathbf{A}}_m^{(t+1)})^T \mathbf{D} \right) + \lambda_m^{(t+1)} \sum_{l=1}^{p_m} \sqrt{\sum_{j=1}^{r_m} \mathbf{D}_{lj}^2} \right\}.$$

Theory

- If we know that the mean contrasts are full-rank, a simpler version of LEEM is minimax optimal in terms of clustering error (Mai et al. (2022; *JASA*));
- If we know the true labels, a supervised version of LEEM is consistent if $srp \log p = o(n)$, where $p = \prod_{m=1}^M p_m$ (Li et al. (2022); *Statistica Sinica*);
- The full theory of LEEM requires further investigation (Li & Mai (2024); *JCGS*).



Outline

1 Background

2 Model

3 Estimation

4 Real Data



The EEG Dataset

- 122 subjects (Alcoholic: 77; Nonalcoholic: 45)
- $\mathbf{X}_i \in \mathbb{R}^{256 \times 64}$
 - Row: time points; Column: channels of electrodes
 - 120 trials
- Goal: Group alcoholic and nonalcoholic subjects

Preprocessing

- Average across all trials, then downsize to 64×64 matrices
- Rotation
 - Marginal covariances: $\tilde{\Sigma}_1 = \sum_{i=1}^n \mathbf{X}_i \mathbf{X}_i^T$, $\tilde{\Sigma}_2 = \sum_{i=1}^n \mathbf{X}_i^T \mathbf{X}_i$
 - Obtain $\mathbf{U}, \mathbf{V} \in \mathbb{O}^{64 \times 64}$ of which the columns are eigenvectors of $\tilde{\Sigma}_1$ and $\tilde{\Sigma}_2$
 - Rotate \mathbf{X}_i 's into $\tilde{\mathbf{X}}_i = \mathbf{U}^T \mathbf{X}_i \mathbf{V}$



Background ○○○	Model ○○○○○	Estimation ○○○○	Real Data ○○●○○
-------------------	----------------	--------------------	--------------------

Results

	LEEM	K-means	SKM	DTC	DEEM	TGMM
CE(%)	27.87	42.62	48.36	40.98	34.43	47.54

Table: Clustering error rates of the EEG dataset. Initial values, ranks, and penalty parameters of the methods are tuned by corresponding criteria. For LEEM, BIC selects the rank to be (10,10).

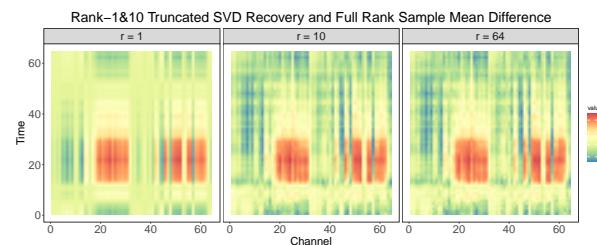


Figure: Comparison between rank- r truncated SVD recoveries of $\hat{\mu}_2 - \hat{\mu}_1$ where $r=1, 10$, and 64 .

Qing Mai LEEM	16 / 18		
Background ○○○	Model ○○○○○	Estimation ○○○○	Real Data ○○●○○

References

- Li et al. (2010). On Dimension Folding of Matrix-or Array-Valued Statistical Objects. *AoS*.
- Zhang & Han (2019). Optimal Sparse Singular Value Decomposition for High-Dimensional High-Order Data. *JASA*.

Qing Mai LEEM	17 / 18

Our papers

- Mai et al. (2022). A Doubly Enhanced EM Algorithm for Model-Based Tensor Clustering. *JASA*, **117**, 2120–2134.
- Li et al. (2024). The Tucker Low-Rank Classification Model for Tensor Data. *Statistica Sinica*, **in press**.
- Li & Mai (2024). Model-Based Tensor Low-Rank Clustering. *JCGS*, **33**, 208–218.

Thank you!

Gertraud Malsiner-Walli

Material list:

Vávra J., Komárek A., Grün B., Malsiner-Walli G. (2024) Clusterwise multivariate regression of mixed-type panel data. Statistics and Computing, 34, 46.



Clusterwise multivariate regression of mixed-type panel data

Jan Vávra¹ · Arnošt Komárek¹ · Bettina Grün² · Gertraud Malsiner-Walli²

Received: 21 July 2022 / Accepted: 26 September 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

Multivariate panel data of mixed type are routinely collected in many different areas of application, often jointly with additional covariates which complicate the statistical analysis. Moreover, it is often of interest to identify unknown groups of subjects in a study population using such data structure, i.e., to perform clustering. In the Bayesian framework, we propose a finite mixture of multivariate generalised linear mixed effects regression models to cluster numeric, binary, ordinal and categorical panel outcomes jointly. The specification of suitable priors on the model parameters allows for convenient posterior inference based on Markov chain Monte Carlo (MCMC) sampling with data augmentation. This approach allows to classify subjects in the data and new subjects as well as to characterise the cluster-specific models. Model estimation and selection of the number of data clusters are simultaneously performed when approximating the posterior for a single model using MCMC sampling without resorting to multiple model estimations. The performance of the proposed methodology is evaluated in a simulation study. Its application is illustrated on two data sets, one from a longitudinal patient study to infer prognosis groups, and a second one from the Czech part of the EU-SILC survey where households are annually interviewed to obtain insights into changes in their financial capability.

Keywords Multivariate longitudinal data · Mixed type outcome · Generalised linear mixed model (GLMM) · Model-based clustering · Classification · Sparse finite mixture · EU-SILC

1 Introduction

Multivariate panel data containing several variables of different scale types are nowadays routinely collected in many different areas of application. However, panel data require specific statistical models and estimation methods in order to be able to harvest the full potential of this data collection mode (see, e.g., Fitzmaurice et al. 2008).

The repeated measurements for the same subjects induce within-subject correlations which implies that the usual independence assumption only holds on subject level. In a regression context with a single outcome variable, one usually accounts for this using linear mixed effects models (LMMs; Laird and Ware 1982) and generalised linear mixed models (GLMMs; Breslow and Clayton 1993). In these models, fixed effects capture the overall effect aggregated over all subjects and the subject-specific random effects, which are assumed to be normally distributed across subjects, account for between-subjects variation, such that conditional on the random effects the independence assumption also holds within subjects. The availability of several outcome variables leads to multivariate GLMMs where in order to account for the within-subject correlation between different outcomes, Fieuws and Verbeke (2006) allowed these random effects to also be correlated across outcomes despite computational challenges in a maximum likelihood framework.

Specifying only random effects following a normal distribution may be insufficient to capture and also characterise the between-subjects variability. Assuming that groups of subjects exist where the effects on the outcome variables

Jan Vávra
vavraj@karlin.mff.cuni.cz

Arnošt Komárek
komarek@karlin.mff.cuni.cz

Bettina Grün
bettina.gruen@wu.ac.at

Gertraud Malsiner-Walli
gertraud.malsiner-walli@wu.ac.at

¹ Faculty of Mathematics and Physics, Charles University,
Sokolovská 49/83, 186 75 Prague, Czech Republic

² Institute for Statistics and Mathematics, Vienna University of
Economics and Business, Welthandelsplatz 1, Vienna 1020,
Austria

distinctively differ, leads to a model-based clustering (MBC) problem (Fraley and Raftery 2002). MBC of longitudinal data within the maximum likelihood framework has been considered using mixtures of LMMs (e.g., Verbeke and Lesaffre 1996; Celeux et al. 2005) and mixtures of multivariate LMMs and multivariate non-linear mixed effects model (e.g., Proust-Lima et al. 2017; Villaruel et al. 2009). Mixtures of multivariate mixed models for various outcome types have also been considered, but in general covering only special cases of specific combinations. For example, Proust-Lima et al. (2017) consider mixtures of multivariate mixed effects models for Gaussian as well as time-to-event outcomes jointly in a maximum likelihood framework. Latent variable models assuming a joint factor impacting all outcomes have also been considered to allow for inclusion of categorical outcomes using a thresholding approach in Proust-Lima et al. (2017) within a maximum likelihood framework and Vávra and Komárek (2022) using a Bayesian approach.

Extending MBC to include multivariate GLMMs as cluster-specific models provides a convenient approach to (1) include different outcome variables of varying type simultaneously and (2) account for within-subject correlation using random effects correlated across outcomes. Special cases of such a model specification have been previously considered, e.g., within a Bayesian framework Tan et al. (2022) consider mixtures of multivariate GLMMs allowing only the Gaussian, Poisson and Binomial distribution for the outcomes while also assuming that random effects are uncorrelated across outcomes, i.e., different outcomes are uncorrelated for the same subject after accounting for the clustering structure.

In this paper, we consider the general model class of finite mixtures of GLMMs and propose a modelling approach which allows to cluster subjects based on multivariate longitudinal outcomes of possibly different types by building on and combining several widely used methodologies without restricting them to a subset of special cases. We propose a model specification including multivariate GLMMs with correlated random effects across outcomes within the MBC framework and implement the estimation within the Bayesian framework exploiting a suitable prior specification which allows for simultaneous estimation of the number of clusters. An implementation of the proposed methodology is provided on GitHub at https://github.com/vavrajian/MBC_GLMMs.

The paper is organised as follows. In Sect. 2, we first outline the multivariate GLMM approach which allows the joint modelling of mixed-type (numeric, binary, ordinal and general categorical) longitudinal data. In Sect. 3, we extend this model to allow for unobserved discrete heterogeneity using a mixture approach in the spirit of MBC. Section 4 embeds the model within a Bayesian framework and outlines suitable prior specifications. In particular, a sparse finite mixture approach allows to conveniently estimate the num-

ber of data clusters or groups. Section 5 provides the details of the Markov chain Monte Carlo (MCMC) algorithm for model estimation as well as the necessary post-processing steps to obtain an identified model. The simulation study in Sect. 6 evaluates the ability of the proposed model and inference methods to identify the true number of data clusters, to induce good cluster solutions and to characterise the clusters through the coefficient estimates. Sections 7 and 8 contain the analyses of two different data sets using the proposed approach: data from a medical study, where patients are monitored over an extended time period with multiple laboratory measurements being available for each visit, and data from the EU-SILC (European Union Statistics on Income and Living Conditions) survey conducted in the Czech Republic from 2005 to 2018, where households are monitored for four consecutive years and information on their financial capabilities is collected. For both applications several variables naturally serve as outcome variables in a regression setting, and identifying groups of patients or households with similar regression patterns is of core interest. Finally, Sect. 9 concludes.

2 Multivariate regression for mixed-type panel data

We assume that the multivariate panel data are composed of n subjects with n_i observations being available for each subject i . In addition R variables of mixed-type are considered as outcome variables in the regression models. These outcomes may have the following scale types: numeric, binary, ordinal or general categorical.

We define different index sets for the outcomes depending on the scale level such that \mathcal{R}^{Num} contains the indices of the numeric outcomes, \mathcal{R}^{Bin} those of the binary outcomes, \mathcal{R}^{Ord} those of the ordinal outcomes and \mathcal{R}^{Cat} those of the general categorical outcomes. This implies that $\mathcal{R} = \{1, \dots, R\} = \mathcal{R}^{\text{Num}} \cup \mathcal{R}^{\text{Bin}} \cup \mathcal{R}^{\text{Ord}} \cup \mathcal{R}^{\text{Cat}}$.

In addition to the outcome observations $Y_{i,j}^r$ ($r = 1, \dots, R$, $j = 1, \dots, n_i$ and $i = 1, \dots, n$), for each subject i , additional observations are available which are used as covariates in the regression. We denote these additional variables, which are used as covariates in the regression for outcome variable r of subject i and its j -th observation, by $v_{i,j}^r$. For subject i , let $\mathbf{Y}_i^r = (Y_{i,1}^r, \dots, Y_{i,n_i}^r)^T$ be the complete vector of all values of outcome r and $\mathcal{C}_i^r = \{v_{i,1}^r, \dots, v_{i,n_i}^r\}$ the set of all covariates for outcome r . Combining them across the outcomes gives

$$\mathbb{Y}_i = \{\mathbf{Y}_i^r, r \in \mathcal{R}\}, \quad \mathcal{C}_i = \{\mathcal{C}_i^r, r \in \mathcal{R}\}, \quad (1)$$

which denotes all information (outcomes and covariate values) available for subject i . \mathbf{Y}^r and \mathcal{C}^r represent the complete information (outcome and covariate values) regarding one chosen outcome $r \in \mathcal{R}$ from all subjects. Finally, \mathbb{Y} and \mathcal{C} stand for all gathered information (all outcomes and covariate values) from all subjects.

The joint model for data (1) is built in the following way: For each outcome (each $r \in \mathcal{R}$) a generalised linear mixed model (GLMM) is assumed. A classical linear mixed model (LMM) is assumed for numeric outcomes and a logistic regression model with random effects is used for binary outcomes. The logistic regression model is extended for the ordinal and general categorical outcomes with more than two levels.

These individual regression models are combined into a multivariate model by assuming that the outcome variables given the regression models are independent between subjects and also within subjects conditional on the random effects of the mixed-effects models. However, the random effects are allowed to be correlated within subjects within and across the different outcomes. In this way correlation is induced between the outcomes given the regression models for each subject.

2.1 Generalised linear mixed models

A generalised linear mixed model is assumed for each outcome. This means that for each outcome a distribution from the exponential family is assumed as well as a link function which maps the linear predictor, determined by a linear combination of the fixed and random effects with their covariates, to the conditional mean of the outcome, given the covariate values. Thus, the linear predictor $\eta_{i,j}^r$ for observation j belonging to subject i specific to outcome r is given by the sum of

- a fixed part $\eta_{i,j}^{F,r} = (\mathbf{x}_{i,j}^r)^\top \boldsymbol{\beta}_r$, which is a linear combination of regressors $\mathbf{x}_{i,j}^r$ derived from the full covariate information $\mathcal{C}_{i,j}$ with the unknown vector of coefficients $\boldsymbol{\beta}_r$ of dimension d_r^F ;
- a random part $\eta_{i,j}^{R,r} = (\mathbf{z}_{i,j}^r)^\top \mathbf{b}_i^r$, which is a linear combination of regressors $\mathbf{z}_{i,j}^r$ derived from the full covariate information $\mathcal{C}_{i,j}$ with the subject-specific vector of random effects \mathbf{b}_i^r of dimension d_r^R .

The linear predictor is thus given by

$$\eta_{i,j}^r = \eta_{i,j}^{F,r} + \eta_{i,j}^{R,r} = (\mathbf{x}_{i,j}^r)^\top \boldsymbol{\beta}_r + (\mathbf{z}_{i,j}^r)^\top \mathbf{b}_i^r s.$$

The fixed-effects part $\eta_{i,j}^{F,r}$ captures the overall trend, and the random-effects part $\eta_{i,j}^{R,r}$ captures differences between subjects. While observations between different subjects are

considered independent, the individual observations $j = 1, \dots, n_i$ of subject i are assumed to be independent only given the random effects \mathbf{b}_i^r . Note that in the following notation, we may drop the three indices (i, j, r) at places where the notation would otherwise be unnecessarily complicated.

For each numeric outcome $r \in \mathcal{R}^{\text{Num}}$, we assume the classical LMM (see Laird and Ware 1982):

$$Y_{i,j}^r \mid \mathbf{b}_i^r; \mathcal{C}_{i,j}^r \sim N(\eta_{i,j}^r, \tau_r^{-1}),$$

where $\tau_r > 0$ is the precision (inverse variance) of the noise or regression model errors. The contribution of one numeric observation Y to the log-likelihood is given by:

$$\ell^N(Y|\eta, \tau) = -\frac{1}{2} \log(2\pi) + \frac{1}{2} \log \tau - \frac{\tau}{2} (Y - \eta)^2.$$

Binary outcomes are assumed to follow a logistic regression model. The success probability is linked to the linear predictor using the inverse logit function:

$$P[Y = 1|\eta] = \text{logit}^{-1}(\eta) = \frac{\exp\{\eta\}}{1 + \exp\{\eta\}}.$$

The contribution of one binary observation Y to the log-likelihood is then:

$$\ell^B(Y|\eta) = Y\eta - \log(1 + \exp\{\eta\}).$$

Logit models for ordinal outcomes with $K > 2$ levels are obtained by parameterising the *cumulative* probabilities and linking them to the linear predictors using the inverse logit function (e.g., Hartzel et al. 2001, Section 2.2):

$$p_k := P[Y > k|\eta, \mathbf{c}] = \text{logit}^{-1}(\eta - c_k) \quad \text{for any } k = 1, \dots, K,$$

where $-\infty = c_0 < c_1 < \dots < c_K = \infty$ are ordered intercepts that shift the linear predictor η and $\mathbf{c} = (c_k)_{k=0,\dots,K}$. Note that for identifiability purposes the intercept term must *not* be included among the fixed effects in the logit models for ordinal outcomes. Also note that this model formulation is based on the proportional odds assumption; the log-odds differ only in the intercepts: $\log(P[Y > k|\eta, \mathbf{c}] / P[Y \leq k|\eta, \mathbf{c}]) = \eta - c_k, k = 1, \dots, K$. For $K = 2$ this formulation is equivalent to the logistic regression model since a single free threshold c_1 is included in the model. This single threshold corresponds to the negative intercept term in logistic regression as $q_1 = 1 - p_1$ and $q_2 = p_1$. Using the notation $p_0 = P[Y > 0] = 1$ and $p_K = P[Y > K] = 0$, the probability of observing a

value k is obtained as the difference between two consecutive cumulative probabilities:

$$\begin{aligned} q_k &:= \mathbb{P}[Y = k|\eta, \mathbf{c}] = \mathbb{P}[Y > k-1|\eta, \mathbf{c}] - \mathbb{P}[Y > k|\eta, \mathbf{c}] \\ &= p_{k-1} - p_k. \end{aligned}$$

The contribution of one ordinal observation Y to the log-likelihood is given by:

$$\ell^O(Y = k|\eta, \mathbf{c}) = \log(q_k) = \log(p_{k-1} - p_k).$$

The standard logit parameterisation (Hartzel et al. 2001, Section 2.3) for a general categorical outcome with $K > 2$ levels requires a specific linear predictor η for each level $k = 1, \dots, K$. Hence, the linear predictor for this outcome is the vector $\boldsymbol{\eta} = \{\eta_1, \dots, \eta_K\}$ where each η_k is a linear combination of the same regressors with a different set of fixed effects $\beta_{r,k}$ and random effects $b_{i,k}^r$. The probability for level k is then obtained as the k -th element of the vector of probabilities obtained from transforming the linear predictor vector with the multivariate softmax function:

$$\mathbb{P}[Y = k|\boldsymbol{\eta}] = \text{softmax}_k(\boldsymbol{\eta}) = \frac{\exp\{\eta_k\}}{\sum_{k'=1}^K \exp\{\eta_{k'}\}}.$$

This yields the probability ratios $\mathbb{P}[Y = k_1|\boldsymbol{\eta}] / \mathbb{P}[Y = k_2|\boldsymbol{\eta}] = \exp\{\eta_{k_1} - \eta_{k_2}\}$. For identifiability, the predictors $\boldsymbol{\eta}$ have to be restricted. Using the usual treatment contrasts, we fix one to zero, e.g. the last one $\eta_K = 0$, by setting $\beta_{r,K} = 0$ and $b_{i,K}^r = \mathbf{0}$. This means it is sufficient to consider for $\boldsymbol{\eta}$ the $(K-1)$ -dimensional vector containing $\{\eta_1, \dots, \eta_{K-1}\}$. Imposing this restriction implies that the estimated regression coefficients capture the probability ratio between the k -th and the last category K . Hence, level K has a specific role and in general should correspond to some baseline level in applications. Note that under $K = 2$ this formulation reduces to the logistic regression assumed for binary outcomes. Then one has one actual predictor $\eta = \eta_1$ and fixes $\eta_2 = 0$. The contribution of one general categorical observation Y to the log-likelihood is given by:

$$\ell^C(Y = k|\boldsymbol{\eta}) = \eta_k - \log \left(1 + \sum_{k'=1}^{K-1} \exp\{\eta_{k'}\} \right).$$

2.2 Combining the multivariate responses

In the GLMM framework, the random effects b_i^r capture the correlation between the outcome values observed for each subject i and outcome $r \in \mathcal{R}$ conditional on the regression model. In the multivariate setting with several different outcome variables, the random effects are also used to capture correlations between different outcome variables for a subject i . To this end, we suppose a joint multivariate distribution

for all random effects similar to Fieuws and Verbeke (2004, 2006), Komárek and Komárová (2013, 2014), Vávra and Komárek (2022).

Let us denote the set of fixed effects by $\boldsymbol{\beta} = \{\beta_r, r \in \mathcal{R}\}$ and the overall vector of random effects for subject i by $\mathbf{b}_i = \{b_i^r, r \in \mathcal{R}\}$. In the following we divide the vector \mathbf{b}_i into subvectors depending on the type of outcomes to emphasise the resulting block structure. In particular, $\mathbf{b}_i^N = \{b_i^r, r \in \mathcal{R}^{\text{Num}}\}$, $\mathbf{b}_i^B = \{b_i^r, r \in \mathcal{R}^{\text{Bin}}\}$, $\mathbf{b}_i^O = \{b_i^r, r \in \mathcal{R}^{\text{Ord}}\}$ and $\mathbf{b}_i^C = \{b_i^r, r \in \mathcal{R}^{\text{Cat}}\}$. We will also use notation $\text{type}(r) \in \{N, B, O, C\}$ for the corresponding type of outcome $r \in \mathcal{R}$.

The overall random effects vector \mathbf{b}_i is now assumed to follow a centred multivariate normal distribution with a *general* covariance matrix, i.e., it is assumed

$$\mathbf{b}_i = \begin{pmatrix} \mathbf{b}_i^N \\ \mathbf{b}_i^B \\ \mathbf{b}_i^O \\ \mathbf{b}_i^C \end{pmatrix} \stackrel{\text{iid}}{\sim} N_{d^R} \left(\mathbf{0}, \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}^{NN} & \boldsymbol{\Sigma}^{NB} & \boldsymbol{\Sigma}^{NO} & \boldsymbol{\Sigma}^{NC} \\ \boldsymbol{\Sigma}^{BN} & \boldsymbol{\Sigma}^{BB} & \boldsymbol{\Sigma}^{BO} & \boldsymbol{\Sigma}^{BC} \\ \boldsymbol{\Sigma}^{ON} & \boldsymbol{\Sigma}^{OB} & \boldsymbol{\Sigma}^{OO} & \boldsymbol{\Sigma}^{OC} \\ \boldsymbol{\Sigma}^{CN} & \boldsymbol{\Sigma}^{CB} & \boldsymbol{\Sigma}^{CO} & \boldsymbol{\Sigma}^{CC} \end{pmatrix} \right),$$

where $d^R = \sum_{r \in \mathcal{R}} d_r^R$ is the total dimension of \mathbf{b}_i and $\boldsymbol{\Sigma} > 0$ is the positive definite covariance matrix of the random effects. A general structure is assumed for this matrix thus allowing to capture arbitrary within-subject dependencies between the different outcomes.

Throughout the manuscript, the notation $p(\cdot | \cdot)$ and $\ell(\cdot | \cdot)$ indicate a conditional probability distribution function and its logarithm, respectively. The unknown parameters of the model consist of the fixed effects $\boldsymbol{\beta}$, the covariance matrix $\boldsymbol{\Sigma}$, the precisions of the error terms of the LMMs for numeric outcomes $\boldsymbol{\tau} = \{\tau_r, r \in \mathcal{R}^{\text{Num}}\}$ and the ordered intercepts $\mathbf{c} = \{c_r, r \in \mathcal{R}^{\text{Ord}}\}$. The random effects \mathbf{b}_i are unknown as well. However, these are considered latent variables that are integrated out to obtain the likelihood.

The multivariate GLMM implies that the i -th subject has the following likelihood contribution

$$\begin{aligned} &p(\mathbb{Y}_i | \boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\tau}, \mathbf{c}; \mathcal{C}_i) \\ &= \int \underbrace{\prod_{r=1}^R \prod_{j=1}^{n_i} \exp \left\{ \ell^{\text{type}(r)} \left(Y_{i,j}^r | \eta_{i,j}^r, \boldsymbol{\tau}, \mathbf{c} \right) \right\}}_{p(\mathbb{Y}_i | \mathbf{b}_i, \boldsymbol{\beta}, \boldsymbol{\tau}, \mathbf{c}; \mathcal{C}_i)} \\ &\quad \times \underbrace{|2\pi \boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \mathbf{b}_i^\top \boldsymbol{\Sigma}^{-1} \mathbf{b}_i \right\}}_{p(\mathbf{b}_i | \boldsymbol{\Sigma})} d\mathbf{b}_i. \end{aligned} \tag{2}$$

The integral (2) can be expressed in closed form only if all outcomes are numeric, i.e., $\mathcal{R} = \mathcal{R}^{\text{Num}}$. Otherwise, numerical methods such as Adaptive Gaussian Quadrature (AGQ) have to be used to approximate the integral (for more details see Appendix C in the supplementary material).

3 Extending to model-based clustering

The multivariate regression for mixed-type panel data proposed so far accounts for slight subject-specific differences and has the fixed effects capturing an overall population effect. This specification assumes that all heterogeneity in the outcome variables can be essentially captured by the available covariates. However, in case of unobserved heterogeneity, i.e., if the population in fact contains several groups where different multivariate regression models apply with varying effects and conditional distributions, this model formulation is insufficient and extension to a mixture model warranted.

A mixture model enables a clusterwise regression setup where based on a model-based clustering (MBC) approach subjects are classified into groups having similar regression effects. Such a mixture model allows to classify available subjects as well as new subjects into groups. A group-specific analysis is helpful for a better understanding how the effects of the covariates differ across groups in the population.

3.1 Creating a mixture distribution

Unobserved heterogeneity refers to the fact that there exists a discrete variable $U_i \in \{1, \dots, G\}$ which represents the unobserved group-allocation indicator for subject i ($i = 1, \dots, n$). Within each group g , the model for subject i is given by $p(\mathbb{Y}_i | \boldsymbol{\beta}^{(g)}, \boldsymbol{\Sigma}^{(g)}, \boldsymbol{\tau}^{(g)}, \boldsymbol{c}^{(g)}; \mathcal{C}_i)$ of the form (2) with group-specific parameters being inserted as indicated by the superscript (g) .

This formulation assumes that all parameters vary across groups. However, in general one splits the set of all unknown parameters into a set of parameters which are common to all groups, which we will denote by $\boldsymbol{\zeta}$ in the following, and a set of parameters which are group-specific, i.e., $\boldsymbol{\zeta}^{(g)}$ for the parameters specific to group g . The combination of all group-specific parameters is denoted by $\boldsymbol{\zeta}^{1:G} = \{\boldsymbol{\zeta}^{(g)}, g = 1, \dots, G\}$.

This formulation implies that the assumed conditional probability distribution function of the i th subject's outcomes given the group allocation U_i is

$$\begin{aligned} & p(\mathbb{Y}_i | U_i = g, \boldsymbol{\zeta}, \boldsymbol{\zeta}^{(g)}; \mathcal{C}_i) \\ & \stackrel{(2)}{=} \int \underbrace{\prod_{r=1}^R \prod_{j=1}^{n_i} \exp \left\{ \ell^{\text{type}(r)} \left(Y_{i,j}^r | \boldsymbol{\eta}_{i,j}^r, \boldsymbol{\zeta}, \boldsymbol{\zeta}^{(g)} \right) \right\}}_{p(\mathbb{Y}_i | \boldsymbol{b}_i, U_i = g, \boldsymbol{\zeta}, \boldsymbol{\zeta}^{(g)}; \mathcal{C}_i)} \\ & \quad \underbrace{\left| 2\pi \boldsymbol{\Sigma}^{(g)} \right|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \boldsymbol{b}_i^\top \boldsymbol{\Sigma}^{-(g)} \boldsymbol{b}_i \right\}}_{p(\boldsymbol{b}_i | U_i = g, \boldsymbol{\Sigma}^{(g)})} d\boldsymbol{b}_i, \end{aligned}$$

where we use the notation $\boldsymbol{\Sigma}^{-(g)}$ for the inverse matrix of $\boldsymbol{\Sigma}^{(g)}$, i.e., the precision matrix.

Let $w_g = P(U_i = g | \boldsymbol{w}) \in (0, 1)$, $g = 1, \dots, G$, $\sum_{g=1}^G w_g = 1$, be the (unknown) group sizes, with $\boldsymbol{w} := (w_1, \dots, w_G)$. Integrating out the unobserved group membership U_i , the mixture distribution for the observed outcomes \mathbb{Y}_i of a single subject i given covariates and model parameters $\boldsymbol{\theta} = \{\boldsymbol{w}, \boldsymbol{\zeta}, \boldsymbol{\zeta}^{1:G}\}$ corresponds to

$$p(\mathbb{Y}_i | \boldsymbol{\theta}; \mathcal{C}_i) = \sum_{g=1}^G w_g \int p(\mathbb{Y}_i | \boldsymbol{b}_i, U_i = g, \boldsymbol{\theta}; \mathcal{C}_i) p(\boldsymbol{b}_i | U_i = g, \boldsymbol{\theta}) d\boldsymbol{b}_i,$$

i.e., a mixture distribution which consists of G components with component weights \boldsymbol{w} and component distributions resulting from the integration.

Generic identifiability in the finite mixture case means that two different parameter sets, which induce the same mixture distribution and do not include redundant clusters with duplicated parameterisations or zero component weights, only differ in the cluster labelling. Sufficient conditions for generic identifiability of finite mixtures of linear regression models are established in Hennig (2000) and extended to finite mixtures of GLMs in Grün and Leisch (2008). These results imply that generic identifiability is not expected to be an issue for multivariate outcomes where random effects are included due to repeated measurements being available for subjects with fixed cluster memberships.

3.2 Classifying (new) observations

Given the model and its parameters $\boldsymbol{\theta}$, one can assign observed subjects to groups based on their classification probabilities, i.e., the a-posteriori probabilities to be from each of the group. In this way a partition of the available subjects is obtained which usually is of core interest in clustering applications.

The classification probability or the conditional probability of subject i , which may either already have been included in the data set when fitting the model or represent a new observation, to be from group g given the data used for fitting the model is provided by the Bayes rule:

$$\begin{aligned} u_{i,g}(\boldsymbol{\theta}) &:= P[U_i = g | \mathbb{Y}_i, \boldsymbol{\theta}; \mathcal{C}_i] \\ &= \frac{w_g p(\mathbb{Y}_i | U_i = g, \boldsymbol{\zeta}, \boldsymbol{\zeta}^{(g)}; \mathcal{C}_i)}{\sum_{g'=1}^G w_{g'} p(\mathbb{Y}_i | U_i = g', \boldsymbol{\zeta}, \boldsymbol{\zeta}^{(g')}; \mathcal{C}_i)}. \quad (3) \end{aligned}$$

Expression (3) can also be written as

$$\begin{aligned} u_{i,g}(\boldsymbol{\theta}) &= \frac{w_g |\boldsymbol{\Sigma}^{(g)}|^{-\frac{1}{2}} \int \exp \{ h(\boldsymbol{b}_i; \mathbb{Y}_i, \boldsymbol{\zeta}, \boldsymbol{\zeta}^{(g)}; \mathcal{C}_i) \} d\boldsymbol{b}_i}{\sum_{g'=1}^G w_{g'} |\boldsymbol{\Sigma}^{(g')}|^{-\frac{1}{2}} \int \exp \{ h(\boldsymbol{b}_i; \mathbb{Y}_i, \boldsymbol{\zeta}, \boldsymbol{\zeta}^{(g')}; \mathcal{C}_i) \} d\boldsymbol{b}_i}, \end{aligned}$$

where

$$\begin{aligned} h_g(\mathbf{b}_i) &:= h\left(\mathbf{b}_i; \mathbb{Y}_i, \boldsymbol{\zeta}, \boldsymbol{\zeta}^{(g)}; \mathcal{C}_i\right) \\ &= \sum_{r \in \mathcal{R}} \sum_{j=1}^{n_i} \ell^{\text{type}(r)}\left(Y_{i,j}^r \mid \boldsymbol{\eta}_{i,j}^r, \boldsymbol{\zeta}, \boldsymbol{\zeta}^{(g)}\right) \\ &\quad - \frac{1}{2} \mathbf{b}_i^\top \boldsymbol{\Sigma}^{-(g)} \mathbf{b}_i. \end{aligned}$$

The integrals $\int \exp\{h_g(\mathbf{b}_i)\} d\mathbf{b}_i$ are numerically approximated via AGQ (Pinheiro and Chao 2006), see Appendix C in the supplementary material for details. Note that we avoid the evaluation of these integrals in the estimation process itself by working with the full-conditional distributions and approximate these integrals only when computing $u_{i,g}(\boldsymbol{\theta})$ a-posteriori, see Sect. 5.3.

4 Bayesian modelling with suitable prior specifications

The model parameters $\boldsymbol{\theta} = \{\mathbf{w}, \boldsymbol{\zeta}, \boldsymbol{\zeta}^{1:G}\}$ imply the likelihood

$$L(\boldsymbol{\theta}) = \prod_{i=1}^n \left\{ \sum_{g=1}^G w_g p\left(\mathbb{Y}_i \mid U_i = g, \boldsymbol{\zeta}, \boldsymbol{\zeta}^{(g)}; \mathcal{C}_i\right) \right\}. \quad (4)$$

In the following we will pursue a Bayesian approach and determine the posterior distribution of the model parameters. Compared to maximum likelihood estimation, the Bayesian framework allows to regularise the mixture likelihood through the prior specification, eases inference through data augmentation and enables convenient estimation of the number of data clusters.

The Bayesian framework and the related MCMC methodology allow for full exploitation of the hierarchical structure of the model. The integration with respect to the unobserved quantities (U_i, \mathbf{b}_i) is elegantly avoided by data augmentation and the sampling mechanism. This also applies potentially to missing outcome values, for which a predictive distribution can be obtained simultaneously with the model estimation as long as all covariates are at disposal. Such an approach allows to retain more observations compared to a complete case analysis and thus is more informative. Moreover, the likelihood (4) is regularised by setting up convenient prior distributions over model parameters. Additionally, the Bayesian framework enables the estimation of the number of clusters in the data by specifying a suitable prior distribution on the component weights \mathbf{w} .

We employ the usual prior specification used in Bayesian MBC where exchangeable priors are imposed on the components and their weights, and they are only potentially coupled through the use of hierarchical priors. Assuming that the pri-

ors on the component weights and the covariance matrices depend on hyperparameters e_0 and \mathbb{Q} , respectively, the joint prior distribution imposed decomposes into

$$\begin{aligned} p(\boldsymbol{\theta}, e_0, \mathbb{Q}) &= p(\mathbf{w}|e_0) p(e_0) p(\boldsymbol{\beta} | \boldsymbol{\tau}) \\ &\quad p(\boldsymbol{\tau}) p(\mathbf{c}) p(\boldsymbol{\Sigma} | \mathbb{Q}) p(\mathbb{Q}). \end{aligned}$$

In the following, the prior specifications suitable for this modelling approach in an MBC context are discussed in detail.

4.1 Prior setting for the component distributions

In Bayesian mixture modelling, improper priors are not feasible for the component distributions if the parameters are component-specific. Because, improper priors would induce an improper posterior as components have a positive probability of containing not a single observation (Roeder and Wasserman 1997). Thus proper priors necessarily need to be selected for these parameters. However, in general selecting slightly informative priors which impose a certain amount of regularisation is desirable in Bayesian mixture modelling because this eliminates spurious modes from the likelihood. For example, we standardise the covariates in the regression prior to the analysis and impose regularising coefficient priors gauged to the unit scale.

In the following, we only discuss the priors specified when all parameters characterising the component distributions are group-specific. Alternatively, one could also split these parameter vectors into a sub-vector containing the group-specific parameters and a sub-vector containing the parameters which are identical across groups. In this case, the priors have to be suitably modified, but the specifications are in an analogous way. Even if not strictly necessary for obtaining a proper posterior, we would in a setup with group-invariant parameters also suggest to impose the regularising priors we recommend for group-specific parameters as this usually improves the sampling efficiency.

4.1.1 Priors on the fixed-effects coefficients and the precisions

The regression coefficients for numeric outcomes $\boldsymbol{\beta}_r^{(g)} = (\beta_{r,1}^{(g)}, \dots, \beta_{r,d_r^F}^{(g)})$, $r \in \mathcal{R}^{\text{Num}}$, $g = 1, \dots, G$, are assumed to be a-priori independent and follow a conjugate normal distribution in combination with the precision parameter $\tau_r^{(g)}$, that is $N(\beta_{0,r,j}, (\tau_r^{(g)})^{-1} d_{j,j}^r)$ where $\beta_{0,r,j}$ and $d_{j,j}^r$ are fixed hyperparameters. These hyperparameters are set equal to 0 and 1, respectively, in the applications.

The regression coefficients for the binary, ordinal and general categorical outcomes, i.e., $\beta_{r,j}^{(g)}$, $r \in \mathcal{R}^{\text{Bin}} \cup \mathcal{R}^{\text{Ord}}$ and $\beta_{r,k,j}^{(g)}$, $r \in \mathcal{R}^{\text{Cat}}$ are also assumed to be a-priori independent and follow an analogous normal distribution

$N(\beta_{0,r,j}, d_{j,j}^r)$, where, however, no precision parameter τ is involved.

Regarding the ordered intercepts $c_r^{(g)}$ estimated for ordinal outcomes, i.e., $r \in \mathcal{R}^{\text{Ord}}$, the prior is not specified for them directly, but for transformed quantities. The $(K - 1)$ -dimensional ordered intercepts $(c_{r,1}^{(g)}, \dots, c_{r,K-1}^{(g)})$ are transformed into probabilities $(\pi_{r,1}^{(g)}, \dots, \pi_{r,K}^{(g)})$:

$$\begin{aligned}\pi_{r,k}^{(g)} &= P[Y_{i,j}^r = k \mid \mathbf{b}_i = \mathbf{0}, U_i = g, \mathbf{x}_{i,j}^r = \mathbf{0}] \\ &= \text{logit}^{-1}(c_{r,k}^{(g)}) - \text{logit}^{-1}(c_{r,k-1}^{(g)}), \\ c_{r,k}^{(g)} &= \log\left(\frac{\pi_{r,1}^{(g)} + \dots + \pi_{r,k}^{(g)}}{\pi_{r,k+1}^{(g)} + \dots + \pi_{r,K}^{(g)}}\right).\end{aligned}\quad (5)$$

The prior distribution is then specified for the probabilities $\pi_{r,k}^{(g)}$ for all outcomes $r \in \mathcal{R}^{\text{Ord}}$ using a product of Dirichlet distributions:

$$p(\boldsymbol{\pi}) \propto \prod_{r \in \mathcal{R}^{\text{Ord}}} \prod_{g=1}^G \prod_{k=1}^K (\pi_{r,k}^{(g)})^{\alpha_{r,k}-1}, \quad (6)$$

where the hyperparameters $\alpha_{r,k}$ are fixed. A value of 1 inducing a uniform distribution on the simplex is used in the later applications.

The precision parameters $\tau_r^{(g)}$ for numeric outcomes are assumed to follow independent Gamma priors $\tau_r^{(g)} \sim \Gamma(\alpha_1, \alpha_2)$ with shape $\alpha_1 > 0$ and rate $\alpha_2 > 0$. For calculations in the later applications, we use $\alpha_1 = \alpha_2 = 1$.

4.1.2 Priors on the random effects parameters

The covariance matrices $\Sigma^{(g)}$ of the random effects \mathbf{b}_i are general positive definite matrices. We impose a Wishart prior on the inverse covariance matrices $\Sigma^{-(g)} := (\Sigma^{(g)})^{-1}$ to preserve conjugacy. The parameters of the Wishart prior are the scale matrix \mathbb{Q} and the number of degrees of freedom $v_0 \geq d^R$. To avoid selecting a specific value for the scale matrix and aiming at obtaining a weakly informative prior for the covariance matrices, we also assume a prior for the scale matrix \mathbb{Q} while keeping the number of degrees of freedom $v_0 \geq d^R$ fixed. Again a Wishart prior is assumed for the inverse scale matrix \mathbb{Q}^{-1} . For this prior, fixed values are selected for the scale matrix and the number of degrees of freedom v_1 . In our applications we use $v_0 = v_1 = d^R + 1$ and a diagonal matrix for the scale matrix given by $\mathbb{D}^{\mathbb{Q}} = 100 \cdot \mathbb{I}_{d^R}$.

4.2 Prior setting for the component weights: sparse finite mixtures

Following the usual Bayesian mixture modelling specification, we impose a symmetric Dirichlet prior on the component weights \mathbf{w} :

$$\mathbf{w}|e_0 \sim \text{Dir}_G((e_0, \dots, e_0)) \equiv \text{Dir}_G(e_0)$$

with probability density function

$$p(\mathbf{w}|e_0) = \frac{\Gamma(G \cdot e_0)}{\Gamma(e_0)^G} \prod_{g=1}^G w_g^{e_0}.$$

The specification of e_0 is crucial depending on whether one assumes a-priori that subjects from all components are contained in the data set with a high probability. We denote by G_+ the number of components from which subjects are generated in the given data set. The choice of e_0 controls the prior probability of $G_+ < G$. This probability is high when e_0 is small because the Dirichlet prior then puts a lot of mass on the boundary regions of the simplex and many of the G weights are small a-priori. For large values of e_0 one has with high probability $G = G_+$ a-priori, i.e., the number of data clusters coincides with the number of components specified.

We follow Frühwirth-Schnatter (2011) when specifying e_0 to take into account whether the number of groups in the data set is known or should be estimated from the data. In case the number of groups in the data set are a-priori known, one would thus set G equal to this number and use a rather large value for e_0 . By contrast, in case one needs to estimate G_+ from the data set, it is convenient to pursue the *sparse finite mixture* approach proposed by Malsiner-Walli et al. (2016). This approach consists of selecting a large, fixed value for the number of components G such that G clearly exceeds the number of clusters in the data. In combination with a small value for e_0 , one achieves that a-priori $G_+ \ll G$ and one may obtain a posterior distribution for G_+ which combines the prior specification of a small number of data clusters with the information on the cluster structure contained in the data.

To attenuate the influence of a specific choice of e_0 , we assign a Gamma prior on e_0 :

$$e_0|a_e, b_e \sim \Gamma(a_e, b_e) \quad (7)$$

with probability density function

$$p(e_0|a_e, b_e) = \frac{b_e^{a_e}}{\Gamma(a_e)} e_0^{a_e-1} \exp\{-b_e e_0\}$$

and prior expected value $E(e_0) = a_e/b_e$. As recommended by Frühwirth-Schnatter and Malsiner-Walli (2019), we select the parameters a_e and b_e of the Gamma prior to have a small

mean when aiming at sparsity, i.e., $E(e_0) = a_e/b_e = 0.01$ with $a_e = 1$. In case the number of components G are assumed known and one aims at $G_+ \approx G$, we select the parameters to induce a mean of $E(e_0) = a_e/b_e = 4$ or directly fix $e_0 = 4$ to avoid sparsity.

5 Bayesian inference

For Bayesian inference, we exploit the ideas of Bayesian data augmentation (Tanner and Wong 1987) while considering all latent quantities, i.e., the component allocations $\mathbf{U} := \{\mathbf{U}_i, i = 1, \dots, n\}$, the random effect vectors $\mathbf{b} := \{\mathbf{b}_i, i = 1, \dots, n\}$ and the missing outcome values denoted by \mathbb{Y}^{mis} as additional latent variables included in the posterior distribution. The model specified in Sects. 2 and 3 results in the following joint distribution of the complete set of outcomes $\mathbb{Y} = (\mathbb{Y}^{\text{obs}}, \mathbb{Y}^{\text{mis}})$ divided into the observed data \mathbb{Y}^{obs} and the missing data \mathbb{Y}^{mis} together with the latent variables $\{\mathbf{U}, \mathbf{b}\}$, the model parameters $\boldsymbol{\theta}$ and the hyperparameters e_0 and \mathbb{Q} :

$$\begin{aligned} p(\mathbb{Y}, \mathbf{U}, \mathbf{b}, \boldsymbol{\theta}, e_0, \mathbb{Q}; \mathcal{C}) \\ = \left[\prod_{i=1}^n p(\mathbb{Y}_i \mid \mathbf{b}_i, \mathbf{U}_i, \boldsymbol{\theta}; \mathcal{C}_i) p(\mathbf{b}_i \mid \mathbf{U}_i, \boldsymbol{\theta}) \right. \\ \left. p(\mathbf{U}_i \mid e_0) \right] p(\boldsymbol{\theta} \mid \mathbb{Q}) p(\mathbb{Q}) p(e_0) \\ = \left[\prod_{i=1}^n p(\mathbb{Y}_i \mid \mathbf{b}_i, \boldsymbol{\beta}^{(U_i)}, \boldsymbol{\tau}^{(U_i)}, \mathbf{c}^{(U_i)}; \mathcal{C}_i) \right. \\ \left. p(\mathbf{b}_i \mid \boldsymbol{\Sigma}^{(U_i)}) w_{U_i} \right] p(\boldsymbol{\theta} \mid \mathbb{Q}) p(\mathbb{Q}) p(e_0), \end{aligned}$$

where $p(\boldsymbol{\theta} \mid \mathbb{Q})$ is the prior distribution of the model parameters given the scale matrix \mathbb{Q} , $p(\mathbb{Q})$ is the prior for the scale matrix and $p(e_0)$ is the prior of the Dirichlet parameter e_0 .

5.1 MCMC algorithm

The posterior distribution $p(\boldsymbol{\theta}, \mathbf{U}, \mathbf{b}, e_0, \mathbb{Q}, \mathbb{Y}^{\text{mis}} \mid \mathbb{Y}^{\text{obs}}; \mathcal{C})$ is estimated using MCMC sampling (Brooks et al. 2011). In particular, we adopt the classical Gibbs sampling scheme wherever possible. Due to the (semi)-conjugate choices of prior distributions, the full-conditioned distributions of $\boldsymbol{\beta}_r^{(g)}, r \in \mathcal{R}^{\text{Num}}, \boldsymbol{\tau}^{(g)}, \boldsymbol{\Sigma}^{(g)}, \mathbb{Q}, \mathbf{w}, \mathbf{U}$ and \mathbb{Y}^{mis} belong to well known distributional families, for which efficient and straightforward sampling mechanisms are available, requiring only updates of the parameters. This is not the case for $\boldsymbol{\beta}_r^{(g)}, r \in \mathcal{R} \setminus \mathcal{R}^{\text{Num}}, \mathbf{c}, \mathbf{b}$ and e_0 , which are sampled using a Metropolis proposal step. More details can be found in Appendices A and B in the supplementary material.

The sampling algorithm can be summarised as follows:

- (1) Choose an initial partition \mathcal{P} , values for the unknown parameters and repeat the Steps (2)–(7).
- (2) Sample the missing outcome values \mathbb{Y}^{mis} according to the data-generating process implied by the specified model.
- (3) Sample the component-specific parameters $\boldsymbol{\zeta}^{(g)}$ for $g = 1, \dots, G$ depending on the number of subjects $n^{(g)}$ assigned to component g :
 - (a) If $n^{(g)} > 0$ (non-empty component): sample the parameters from full-conditioned distributions (directly or using a Metropolis step) using the observations of the subjects currently assigned to cluster g :
 - (b) If $n^{(g)} = 0$ (empty component): sample the parameters from their prior distributions (directly or using a Metropolis step).
- (4) Sample the parameters $\boldsymbol{\zeta}$ which are identical across components and the scale matrix \mathbb{Q} from their full-conditioned distributions (directly or using a Metropolis step).
- (5) Sample the component weights \mathbf{w} from the Dirichlet distribution given by $\text{Dir}_G(\mathbf{n} + e_0 \mathbf{1})$, where $\mathbf{n} = (n^{(1)}, \dots, n^{(G)})^\top$ and $\mathbf{1}$ is a vector of ones.
- (6) Sample the allocation indicators \mathbf{U}_i independently for all subjects to create a new partition \mathcal{P} :
 - (a) Compute the full-conditioned classification probabilities $u_{i,g}(\boldsymbol{\theta}; \mathbf{b}_i)$.
 - (b) Sample new \mathbf{U}_i from the multinomial distribution with probabilities $u_{i,g}(\boldsymbol{\theta}; \mathbf{b}_i)$.
- (7) Sample e_0 using a Metropolis step from $p(e_0 \mid \mathcal{P}, G) \propto p(\mathcal{P} \mid e_0, G) \cdot p(e_0)$.

This algorithm for model estimation has been implemented in R (R Core Team 2022) with the use of the C programming language to optimise the computation time. The code is available on GitHub at https://github.com/vavrajan/MBC_GLMMs.

5.2 Post-processing

After omitting a suitable number of burn-in samples and applying thinning, the final MCMC chain contains M draws of $\boldsymbol{\theta}^m, \mathbf{U}^m$ and $\mathbf{b}_i^m, m = 1, \dots, M$. For each draw m , the cluster indicators \mathbf{U}^m induce component occupation numbers $\mathbf{n}^m = (n^{(1,m)}, \dots, n^{(G,m)})^\top$ and a specific number of non-empty components $G_+^m = G - \sum_{g=1}^G \mathbb{1}(n^{(g,m)} = 0)$. The number of non-empty components may differ among different draws m .

We estimate the number of data clusters as suggested by Malsiner-Walli et al. (2016). They use the mode \widehat{G}_+ of the posterior of the number of filled components as an estimator for the number of clusters in the data:

$$\widehat{G}_+ = \arg \max_{g \in \{1, \dots, G\}} \sum_{m=1}^M \mathbb{1}(G_+^m = g).$$

Then, for the subsequent inference only those MCMC draws are considered where the number of filled components coincides exactly with the mode \widehat{G}_+ . The MCMC draws where a different number of components is filled are discarded and omitted from the further analysis.

Before group-specific inference can be performed based on the MCMC samples, one potentially needs to resolve label switching (Redner and Walker 1984). Because the likelihood as well as the prior and thus the posterior are label invariant, the posterior is multi-modal with modes corresponding to all parameterisations obtained by permuting the labels of unique components. The component labels may be switched across different draws of the MCMC sampler and a unique labelling needs to be obtained to determine an identified model where group-specific inference is possible. We suggest to use the procedure proposed in Frühwirth-Schnatter (2011) and Malsiner-Walli et al. (2016) to resolve label switching with the later describing a method applicable when pursuing the sparse finite mixture approach.

In our simulation study and the applications, we observed that the number of filled components usually stabilises during MCMC sampling at a specific number, usually representing the lower bound of data clusters required to provide an adequate fit for the data. Initialising using a partition with all components being filled, we noted that during the first iterations of the MCMC algorithm superfluous components are emptied and only the necessary number of components required to represent the group structure in the data set remain filled. The sparse finite mixture prior imposed on the component weights induces a penalty for the inclusion of redundant filled components, hence encouraging a solution where only a few components are filled. Monitoring thus the number of filled components serves as a means to assess convergence of the MCMC chain and thus decide on a suitable number of burn-in iterations to discard.

We also noted that label switching did not occur during MCMC sampling after the burn-in samples are omitted in our simulation study and the applications. Using a multivariate regression model with repeated measurements for subjects and avoiding redundant mixture components induces rather crisp classifying probabilities. They induce well separated modes and prevent the sampler also to move between these

modes. Hence, for these analyses there was no need to apply a procedure for resolving label switching and assigning suitable labels to components such that they correspond to an identified model.

5.3 Classifying observations

After MCMC sampling there are basically two possibilities to obtain a final classification or partition of the subjects. The posterior classification probabilities $u_{i,g}$ may be estimated by conditioning not only on the observed data and parameter estimates, but also on estimates of the random effects b_i . Given the MCMC samples this approach can easily be pursued for subjects included in the data set. We use this approach in the simulation studies (Sect. 6) as well as in the application using the EU-SILC data set (Sect. 8). This approach reduces the computational time needed because costly integral approximations are avoided and allows to obtain classifications based on the MCMC draws made for posterior inference anyway.

Alternatively the posterior classification probabilities $u_{i,g}$ can also be estimated by integrating out the random effects. This approach is applied in the second application considered (Sect. 7). It is computationally more expensive, but provides more accurate estimates because the latent random effects are not conditioned on, but integrated out.

5.3.1 Conditioning on random effects

The U_i^m draws obtained during MCMC sampling are posterior draws from the multinomial distribution with success probabilities equal to the a-posteriori probabilities induced by conditioning on the observed data as well as current parameter estimates and draws of the random effects. Their empirical means obtained with $\widehat{U}_{i,g} = \frac{1}{M} \sum_{m=1}^M \mathbb{1}(U_i^m = g)$ represent suitable estimates for the classification probabilities taking into account uncertainty with respect to the parameter estimates as well as the random effects. This approach can be directly applied after the sampling procedure and post-processing and does not require any integral approximation. One only needs to store $n \cdot G \cdot M$ values. However, once the subjects are classified, these values may be discarded.

Based on these classification probabilities, subjects may be classified by assigning each subject i to the cluster g that has the highest estimate $\widehat{U}_{i,g}$ among all $g = 1, \dots, G_+$. In case these classification probabilities are not clearly indicating assignment to a specific group, one may decide to leave those subjects unclassified. Rules to decide not to assign might for example be that the second largest classification probability lies within a pre-specified tolerance (such as 0.2) below the highest one or that the highest probability itself is below a given threshold (such as 0.6). Imposing such a

rule leads to both classified subjects where classification is unambiguous, and unclassified subjects where assignment has been assessed to be not sufficiently clear.

5.3.2 Integrating out random effects

The posterior probabilities $u_{i,g}(\theta)$ are estimated for all sampled θ^m , i.e., the classifying probabilities are determined for the observed data and model parameters while integrating out the random effects. The posterior mean is then estimated by $\widehat{U}_{i,g} = \frac{1}{M} \sum_{m=1}^M u_{i,g}(\theta^m)$. This approach requires $M \cdot G$ approximations of the integral, which is in particular costly when done for each subject $i = 1, \dots, n$. Using the Laplace's approximation is in this case preferable to reduce the computational burden.

This approach approximates the posterior distribution of $u_{i,g}(\theta)$, thus allowing to construct 95% Highest Posterior Density (HPD) credible intervals. Subjects are then classified based on the highest $\widehat{U}_{i,g}$ value. Again for some subjects one may decide not to classify. In this case one can use as rule for example that the upper bound of the HPD intervals of the other groups need to lie below the lower bound of the HPD interval for group g to which one would assign based on the maximum value of $\widehat{U}_{i,g}$. This rule implies that one leaves a subject i unclassified if the classifying probability is comparable for more than one group and hence classification is not unambiguous.

Conditioning on the random effects is complicated for subjects not included in the training data set (e.g. newly observed data points) and there are no sampled indicators U_i^m readily available. This makes the latter approach an attractive choice for such observations even if it is computationally rather expensive.

6 Simulation study

We conduct a simulation study to demonstrate the performance of our proposed approach under various settings. We are particularly interested in assessing how the structure of the sampled data as well as the data generating process affect (1) the ability to estimate the number of data clusters, (2) the clustering performance measured by the misclassification rates and (3) the accuracy of the model parameter estimates. A more detailed summary of the simulation study is provided in the supplementary material.

6.1 Simulation design

A wide range of parameters are selected to specify the simulation study. Some parameters vary across the settings to study their impact on performance, while others are kept fixed. In particular, the sample size is varied with values

$n \in \{100, 250, 500, 1000\}$ and the number of true data clusters $G \in \{2, 3\}$. Regarding the panel structure, we use a rather challenging setting of only $n_i = 4$ observations per subject in order to mimic the panel structure of the applications.

For each data set we generate one outcome of each type—numeric Y^N , binary Y^B , ordinal Y^O with $K^O = 5$ levels and general categorical Y^C with $K^C = 4$ levels. With respect to the random-effects part, we only consider a random intercept term for each type of outcome $b_i = (b_i^N, b_i^B, b_i^O, b_i^C)^\top \sim N_4(\mathbf{0}, \Sigma)$. As a standard setting, we consider the covariance matrix Σ of the random effects to be the same across clusters and to be decomposed into standard deviations and correlation matrix such that

$$\Sigma = S \begin{pmatrix} 1 & -0.5 & -0.5 & -0.4 \\ -0.5 & 1 & 0.3 & 0.4 \\ -0.5 & 0.3 & 1 & 0.2 \\ -0.4 & 0.4 & 0.2 & 1 \end{pmatrix} S.$$

with $S = \text{diag}\{0.5, 0.5, 0.5, 0.5\}$. A common random-effects structure is then also imposed when fitting the model under this standard setting. In addition, several other structures for matrix Σ were considered for data generation as well as model fitting, see the supplementary material for details.

The fixed-effects part of the predictor consists of an intercept term and one other covariate $x \in (0, 1)$. This covariate represents time and is sampled in such a way that the values are close to each other for the same subject. In particular, we use the simulation parameter $\xi = \frac{1}{3}$ to define the length of the observational window for one subject, i.e., for each subject only a third of the total length of the interval is admissible for values of x . To obtain the x values for each subject i , first, the centre of the interval is sampled by $x_{c,i} \sim \frac{\xi}{2} \cdot \text{Unif}\left\{1, \dots, \frac{2}{\xi} - 1\right\}$ and then n_i values for subject i are sampled from $\text{Unif}\left(x_{c,i} - \frac{\xi}{2}, x_{c,i} + \frac{\xi}{2}\right)$ and ordered. Marginally, for $\xi < 1$ the distribution of x is not $\text{Unif}(0, 1)$ since the intervals at the boundary $(0, \frac{\xi}{2})$ and $(1 - \frac{\xi}{2}, 1)$ have lower probability. Note that this setting is selected to resemble the structure of the rotational panel in the EU-SILC data set.

We explore several different ways how the time covariate affects the outcome:

- (a) no effect of time at all (no),
- (b) a slope term common to all clusters (parallel),
- (c) different intercepts and slopes for each cluster resulting in a crossing (cross).

We follow the same scheme when specifying the models for estimation, considering models where no time effect is included, a common slope for time and a group-specific slope for time. Examples of the predictors simulated for the dif-

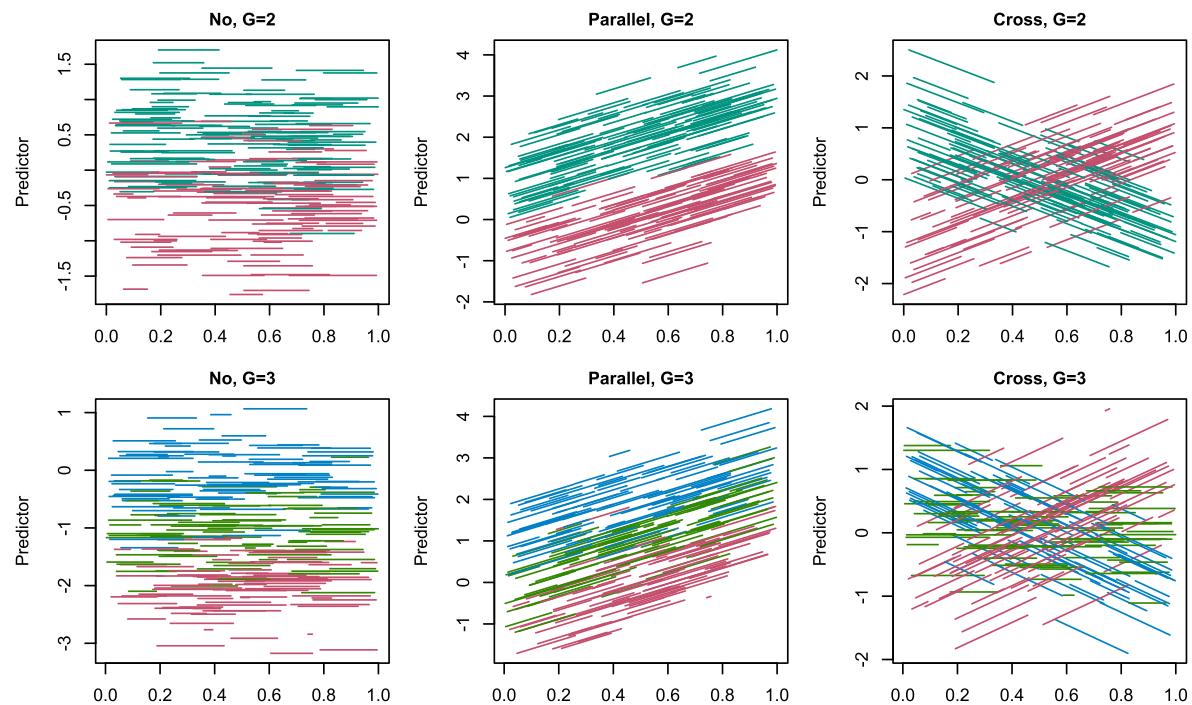


Fig. 1 Linear predictors of $n = 250$ individual subjects generated from G clusters for different types of time effects. The maximal length of the observational window is $\xi = \frac{1}{3}$

ferent time parameterisations and number of clusters G are illustrated in Fig. 1.

The intercept term is always (both when generating the data set and when estimating) considered to be group-specific. This ensures some differences between the clusters. The numerical outcome is obtained by adding an error term with group-specific standard deviation, $\{0.5, 0.8\}$ for $G = 2$ and $\{0.5, 0.75, 1\}$ for $G = 3$, to the linear predictor. This group-specificity of $\tau^{(g)}$ was not only used for generating the data but also assumed when estimating the model. For the ordinal outcome, group-specific equidistant ordered intercepts $c^{(g)}$ both generate the data and are assumed in the model specification when fitting the model (i.e., typically whole numbers shifted by a certain constant amount to have reasonable frequencies of outcome values in each cluster). Three different specifications of intercepts (e.g., using an exchange of monotonicity type) are required to obtain the predictors for the categorical outcome with $K^C = 4$ levels.

We generate 200 data sets for each considered simulation setting. For Bayesian inference, the prior distributions together with their parameter values are specified as outlined in Sect. 4. For estimating the number of data clusters or assessing the clustering abilities, we initialise the Markov chain with the maximal number of components $G_{\max} = 10$ considered for the mixture model. A burn-in period of $B = 500$ samples was enough to then use the next $M = 10\,000$

sampled parameter and latent variable values to approximate their posterior distributions. Subjects were classified using the sampled indicators U_i , leaving subjects unclassified when less than 60% of these indicators assigned the subject to the same cluster. The congruence between the true and estimated cluster allocations was assessed based on the misclassification rate obtained based on a labelling of the estimated clusters which minimises this rate as well as the *adjusted Rand index* (ARI; Rand 1971) which is label-invariant and accounts for agreement by chance.

6.2 Estimating number of data clusters and classifying subjects

Using the data generated as described in Sect. 6.1 we assess the ability of the proposed approach to estimate the number of data clusters and evaluate the classification performance, focusing in particular on the benefit incurred through joint modelling of the outcome variables. We consider the cross parameterisation of time with $\xi = \frac{1}{3}$ for data generation and also use the same model specification for estimation to be able to capture these effects. We estimate the model for each type of outcome separately as well as all four outcomes of different types jointly.

Results (see supplementary material) indicate that the performance regarding the estimation of the number of data

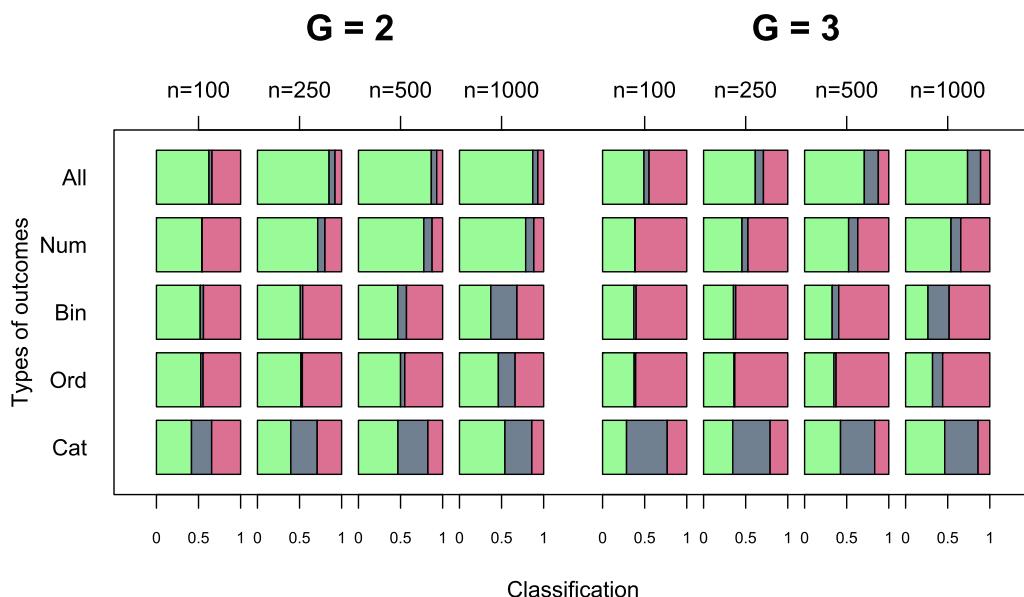


Fig. 2 Proportions of correctly classified (green), unclassified (grey) and misclassified (red) subjects in dependence of the types of outcomes used, sample size n and the true number of data clusters G . The num-

ber of data clusters used for classification are estimated based on \widehat{G}_+ , the most frequent number of non-empty components during MCMC sampling with $G_{\max} = 10$

clusters G_+ is rather comparable regardless of the type of outcome used and also when all outcomes are modelled jointly. Sample size had an effect with only one or two data clusters being selected for $n = 100$ regardless of if the true number of data clusters is 2 or 3. For $G = 2$ and $n = 250$ the number of data clusters was in general already correctly identified, whereas $n = 500$ was required for $G = 3$ to achieve a good performance.

Figure 2 provides an overview on the proportions of correctly classified, unclassified and misclassified subjects when using either only a single outcome variable or using all four outcome variables jointly. In addition the sample size n and the true number of data clusters are also varied. The results for the single outcome variables are shown in the rows labelled “Num” for numeric outcome, “Bin” for binary outcome, “Ord” for ordinal outcome and “Cat” for general categorical outcome. The results when modelling all four outcomes jointly are shown on top in the row labelled “All”. The analogous results obtained for the ARI index are visualised in the supplementary material.

Figure 2 clearly shows a general pattern of an increase in sample size n improving the classification performance. This certainly also is partly due to the underestimation of G_+ for $n \in \{100, 250\}$. In case the number of data clusters is underestimated, a high misclassification rate naturally results. Also the classification performance is in general better if the true number of data clusters is 2 instead of 3. Similar

insights can also be concluded from the results for the ARI (see supplementary material).

Figure 2 also highlights the impact of the type of outcome on the classification performance. If only a single outcome is considered, the numeric outcome performs best, while the binary and ordinal outcomes result in a classification performance barely better than a random classification. Modelling all types together clearly outperforms the single models and achieves the highest correct classification rates indicating the advantage of using a modelling approach which allows to jointly model the data. Clearly, similar insights are also drawn from the ARI results with the highest ARI values being consistently obtained if all outcomes are modelled together and the binary and ordinal outcome resulting in ARI values consistently close to 0, i.e., performing as well as using a random partition.

6.3 Estimating model parameters

Regarding the accuracy of the model parameter estimates, we focus on the estimation of the fixed effects β . In many applications these parameters will be of core interest for characterising the identified clusters and interpreting the effects. We vary the data generation setting with respect to sample size, true number of data clusters, the true and the supposed effect of the time covariate and generate 200 data sets for each setting.

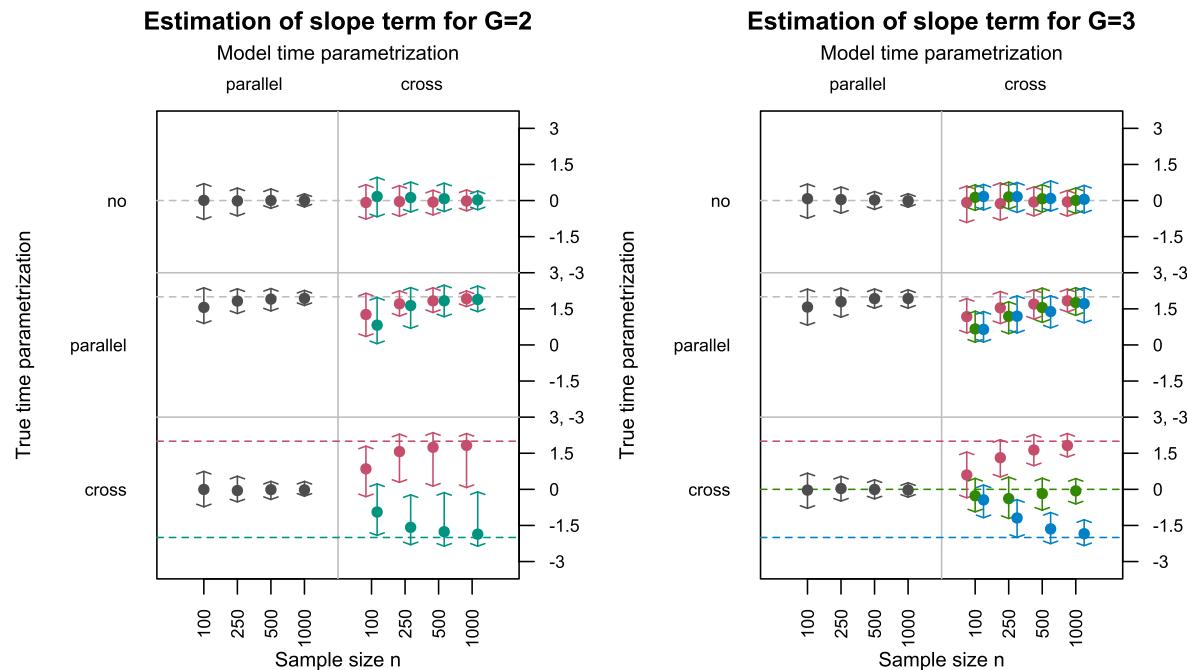


Fig. 3 Medians, 2.5 and 97.5% quantiles of estimated posterior medians of the slope term for the Bin outcome variable across 200 simulated data sets. Model estimation is performed assuming that the number of data clusters G is known. Different settings are considered for the effect

of the time covariate x for data generation (rows) and model specification (columns) and $\xi = \frac{1}{\sqrt{3}}$ is used for data generation. The dashed lines indicate the true values. These are grey in case the effects are identical across clusters and in colour otherwise

A joint model for all outcome variables is estimated assuming that the true number of data clusters is known. This is achieved by setting $G_{\max} = G$ and using $a_e = 4$ and $b_e = 1$ for the hyperparameters of the prior on the component weights to avoid sparse cluster solutions. Using this specification ensures that we estimate exactly G data clusters for each of the 200 simulated data sets. Posterior medians of the estimated group-specific intercepts are used to match the labelling of the estimates for the simulated data sets to the labels of the clusters used in data generation.

Figure 3 shows the results obtained for the slope estimates of the binary outcome. The binary outcome variable corresponds to the least informative outcome type and thus these results demonstrate that accurate estimation is achieved even under the most challenging conditions, in case the sample size is sufficiently large. Estimating a model with a common slope for all clusters leads to the correct estimation of the value 0 (in case no effect of time is present) or 2 (in case the clusters share the same slope term) for a sample size n of 250 or higher for $G = 2$ and 500 or higher for $G = 3$. However, an average effect is estimated when clusters indeed have a different slope. On the other hand, when estimating the model with different slopes across clusters, the group-specific estimates also coincide with the true common value (0 when no effect and 2 in the parallel lines), though, a small

shrinkage towards zero is visible for a low sample size n . Such a shrinkage behaviour can also be discerned in case the data generating process has group-specific slopes. However, this effect vanishes with increasing sample size and excellent results are obtained for $n = 1000$.

7 Analysis of the PBC medical study data

In the study of primary biliary cholangitis (PBC) of liver conducted by the Mayo Clinic between 1974 and 1984, 312 patients were randomly assigned to a placebo control group and to a treatment group consisting of D-penicillamine drug users. The study protocol required visits after 6 months, one year and then annually until the patients died, had a liver transplant or dropped out from the study. At each visit multiple laboratory results were obtained and combined into a longitudinal data set. At each visit not all tests were undertaken leading to missing values in the outcome variable.

This data set has in particular been studied to predict survival, see Therneau and Grambsch (2000). In the following we use the data to infer different prognosis groups based on the observed patterns of evolution of specific markers over time taking also age and gender into account as covariates. Having established an association of the groups identified

with survival, new patients may be classified based on their marker evolvement.

7.1 Data and model description

Similar to Komárek and Komárová (2013), we restrict our analysis to the patients ($n = 260$) who survived the first 910 days (2.5 years) of the study without liver transplantation. The vast majority (178) of patients have $n_i = 4$ visits recorded within this period. However, there are also patients included where only a single visit is available. Restricting the data to only the first 910 days imitates a situation, where a prognosis for a patient is desired and the aim is to establish a classification rule for patients where data from 910 days of the follow-up are available.

Komárek and Komárová (2013) modelled jointly 3 outcomes: *bili* (numeric), *platelet* (count) and *spiders* (binary). We avoid the count variable and work with other types of outcomes instead. Two numeric markers are included as outcome variables: serum bilirubin (*bili*) and *albumin* (on log-scale). Two binary outcome variables are included which indicate if the patient suffered from presence of blood vessel malformations in the skin (*spiders*) and hepatomegaly or enlarged liver (*hepato*). A single ordinal outcome variable is included which indicates the seriousness of *edema*. Missing values were augmented during MCMC sampling to keep all subjects in the analysis and obtain a posterior approximation of the unknown values.

The five markers are jointly modelled by assuming random intercepts for the patients and a group-specific linear effect of time, age at entry to the study and gender without any interaction terms. All other model parameters were also considered to be group-specific to capture differences in all possible aspects. Hence, not only a different evolution over time is expected, but also the effects of age or gender may vary across groups as well as the noise variances for the numeric outcomes and the covariance structure of the random intercepts.

A sparse finite mixture was induced by setting $a_e = 1$, $b_e = 100$ and the other hyperparameters were set to correspond to a unit scale prior distribution. With $G_{\max} = 10$ the MCMC sampling converged after few hundred steps to a $\widehat{G}_+ = 2$ solution. The burn-in period was decided as a multiple of 200 iterations based on a visual inspection of trace plots. For the results reported, $M = 10\,000$ sampled parameter and latent variable values were used without thinning to approximate their posterior distributions. Repeating this procedure for four different chains using random initialisations indicated that results are rather comparable across the chains.

7.2 Results

The $n = 260$ patients are classified based on the maximum classifying probabilities obtained by integrating out the random effects. This results in a partition of the patients into two groups. Combining this grouping with the remaining data (beyond 910 days) allows to determine the Kaplan–Meier estimates of the survival functions for each group (see Fig. 4). Even though the fitted model did not include the information on subsequent survival, the identified groups clearly exhibit different survival curves. Thus the grouping identified can be used to obtain prognosis about future survival. Note that Komárek and Komárová (2013) obtained a similar clustering of the patients into two groups using only three outcomes. This indicates that the clustering structure related to survival is quite strong in this data set allowing it to be deduced using different sets of several outcome variables with potentially different interpretations.

Table 1 provides posterior estimates of the group-specific parameters which allow to characterise the two estimated groups with respect to their covariate effects. The red cluster of Fig. 4 ($n^{(1)} = 106$, 26 men) with the drastically decreasing survival function represents about 41% of patients with high serum bilirubin increasing over time, lower serum albumin in general, increasing odds of spiders with time and increasing risk of edema over time. On the other hand, the turquoise cluster ($n^{(2)} = 154$, one man only) with much higher survival probabilities consists of 59% of the patients who have a low value of serum bilirubin only slowly increasing over time, higher values of serum albumin, stable odds in time for both spiders and hepatomegaly and increasing risk of edema with age.

8 Analysis of the EU-SILC data

The EU-SILC (Statistics on Income and Living Conditions) survey gathers data on households within member states of the European Union, Iceland, Norway and Switzerland annually since 2003. We apply our proposed approach to identify groups of households which differ in their evolvement of financial capability over time as measured by several highly correlated outcomes of mixed type.

Vávra and Komárek (2022) used a similar data set covering a shorter time frame and included less outcome variables as well as covariates in their application. Their analysis resulted in a substantially different clustering of the households also due to a different specification of cluster-specific and cluster-invariant parameters. Different model specifications in fact imply that indeed clusterings with different

Fig. 4 Kaplan–Meier survival function estimates after day 910 of the $n = 260$ patients from the PBC medical study clustered into the two estimated groups

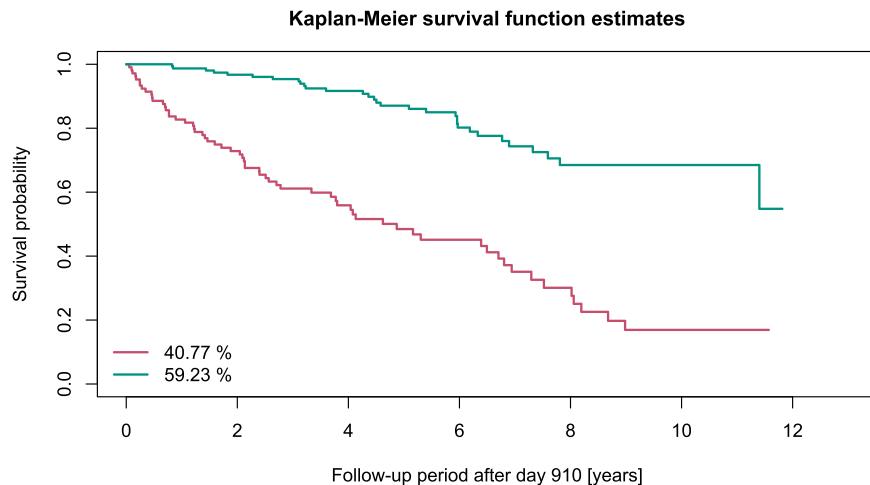


Table 1 Posterior medians of group-specific model parameters including 95% equal-tailed credible intervals

Type	Outcome	Cluster	β_0 or c_k	β_{time}	$10\beta_{\text{age}}$	β_{sex}	$\tau^{-\frac{1}{2}}$
Numeric	log(bili)	1	0.85 (0.18; 1.59)	0.21 (0.14; 0.28)	-0.02 (-0.15; 0.12)	0.10 (-0.25; 0.51)	0.50 (0.46; 0.55)
		2	0.14 (-0.24; 0.53)	0.02 (-0.02; 0.05)	-0.09 (-0.16; -0.02)	0.20 (-0.16; 0.56)	0.24 (0.23; 0.26)
	log(albumin)	1	1.31 (1.19; 1.42)	-0.02 (-0.04; 0.00)	-0.01 (-0.03; 0.01)	-0.03 (-0.08; 0.02)	0.16 (0.15; 0.18)
		2	0.80 (0.69; 0.91)	-0.01 (-0.02; 0.00)	0.01 (0.00; 0.02)	0.45 (0.35; 0.55)	0.12 (0.11; 0.13)
Binary	spiders	1	-0.45 (-2.19; 1.26)	0.50 (0.09; 0.93)	-0.30 (-0.68; 0.07)	0.55 (-0.64; 1.73)	
		2	-0.11 (-1.84; 1.63)	0.08 (-0.39; 0.54)	-0.63 (-1.15; -0.18)	-0.19 (-1.85; 1.51)	
	hepato	1	-0.17 (-1.91; 1.62)	0.32 (-0.12; 0.78)	0.30 (-0.09; 0.69)	-0.10 (-1.39; 1.14)	
		2	0.24 (-1.42; 1.92)	0.07 (-0.27; 0.41)	-0.21 (-0.59; 0.13)	-0.45 (-2.00; 1.13)	
Ordinal	edema	1	4.09 (2.78; 5.50) 7.05 (5.47; 8.80)	1.07 (0.63; 1.53)	0.56 (-0.07; 1.20)	0.62 (-0.65; 1.90)	
		2	1.63 (-0.13; 3.86) 6.75 (4.33; 10.06)	0.08 (-0.39; 0.53)	0.82 (0.07; 1.68)	-2.26 (-4.05; -0.46)	

characterisations are aimed at. This is crucial in this application where no clear data structure is present which would be consistently captured despite slightly different focus of the clustering characterisations.

8.1 Data and model description

The analysis focuses on the subset of Czech households surveyed between 2005 and 2018. This time period includes the years of the economic crisis which started in late 2008. We have $n = 23\,360$ households that were followed for exactly $n_i = 4$ consecutive years, as induced by the rotational design of the study. Starting with more than 7 000 households, each year a quarter is dropped to be replaced by a comparably sized set of new households.

Eight outcomes (two for each type) are modelled jointly using the proposed approach. All eight outcomes reflect the financial capacity of the household. Two numeric outcome variables are included which are income related: *Equivalised*

total disposable income [€/year], that sums the gross personal income components of all household members over the whole year and divides it by the *Equivalised household size* (see below), and *Lowest monthly income to make ends meet* [€/month], that reflects the minimum net monthly income required to pay for all usual necessary expenses of the household. In addition, the financial capacity is measured by the ability to afford certain luxuries. *Affordability of one week annual holiday away from home* and *Capacity to face unexpected financial expenses* are binary outcome variables ("Yes", "No"), while the possession indicators of a car or a computer are general categorical outcome variables with three levels consisting of "Yes", "No—cannot afford" and "No—other reason". Two ordinal outcome variables are also included which rather reflect subjective assessment of financial capability and are measured as the *Ability to make ends meet* (on a scale from 1="with great difficulty" to 6="very easily") and the perceived *Financial burden of the total house-*

ing cost (with levels: “a heavy burden”, “a slight burden”, “not a burden at all”).

Because the numeric income related outcomes have a heavily skewed distribution, we transformed the values to log-scale. In case the income was negative (which very rarely occurred), it was set to zero on the log-scale. The baseline levels for the general categorical and the ordinal outcomes were determined by ordering the categories with respect to their expected positive correlation with increasing financial capacity.

In the regression the time variable indicating the year when the survey was completed was included as group-specific covariate to identify how the financial capacity of the households evolves over time, in particular also during the phase of an economic crises. To capture a possible change in trend, we used a quadratic spline parameterisation with one inner knot.

Additional covariates were also included in the regression which characterise the households. These covariates were included with constant effects across the whole population. These additional variables are: *Level of urbanisation* of their location (with levels “thinly-populated area”, “intermediate area”, “densely populated area”, “capital city of Prague”), the *Highest education* level attained by at least one household member (with levels “lower than secondary”, “secondary”, “higher than secondary”) whether at least one household member is a baby (i.e., younger than 3 years) or a student (i.e., attending some educational institution) and the *Equivalised household size*. The *Equivalised household size* is obtained by summing over all household members using the following weights: a weight of 1 for the first member, a weight of 0.5 for the other household members older than 14 and a weight of 0.3 for household members who are 14 or younger.

The maximum number of components was set to $G_{\max} = 20$. To invoke sparsity we specify the parameters of the prior distribution for e_0 to be $a_e = 1$, $b_e = 100$. To regularise the effect estimates and shrink them towards zero, the standard deviations of the priors for the centred effects were set to 0.5. Ordered intercepts c and error term precisions τ are set to be group-specific, the variance matrix Σ of random effects is kept common to all households.

The burn-in period was prolonged upon a visual inspection of trace plots every 1 000 iterations until convergence was concluded. For the results reported, $M = 1 000$ sampled parameter and latent variable values were used without thinning to approximate their posterior distributions. Repeating this procedure for four different chains using random initialisations indicated again comparability of results obtained across chains.

8.2 Results

Post-processing led to the estimation of $\widehat{G}_+ = 4$ clusters. Using the sampled U_i , we classified households where the maximum classification probability was at least 0.5. Otherwise the household remained unclassified (0.55% of households). The classified households were used to create the plots in Fig. 5 describing the evolvement of the outcome variables across time for each cluster separately.

Figure 5 indicates that the cluster sizes vary strongly with the green cluster containing 78.57% of the households, followed by the yellow cluster consisting of 17.11% of the households. With respect to their cluster size, the remaining two clusters seem rather negligible with the blue cluster containing 3.65% of the households and the red cluster containing 0.12% of the households.

Assessing the financial capacity of the households as depicted by the outcome variables, one can conclude that the blue cluster (3.65%) contains households that are doing well in general, while the yellow cluster (17.11%) represents the more struggling households. In-between these two, the most common green cluster (78.57%) contains households with an intermediate financial capacity. The red cluster (0.12%) consists of the rare households faced with a bad financial situation. They are considered to be outliers as at some point in time they had negative total disposable income.

The group-specific evolvement of the log-scaled *Equivalised total disposable income* is shown in Fig. 5 on the top left. In particular the estimated posterior median curves indicate how the evolvement differs across the clusters. For all four clusters a rather strong increase is captured for the first four years which at the start of the economic crisis either levels off to a rather constant equivalised total disposal income or even to a slightly decreasing one.

The plot of the *Equivalised total disposable income* also indicates that the clusters strongly differ in the standard deviation of the error term in the linear regression model. This standard deviation captures how much the income variable differs for the same household across the four consecutive measurements and thus also reflects how volatile the income situation is for a household. Assessing the group-specific standard deviation estimates for the error term using 95% equi-tailed credible intervals indicates that the green cluster contains households with a rather constant income over time ($\widehat{\sigma} \in (0.063; 0.064)$). On the contrary, the yellow ($\widehat{\sigma} \in (0.129; 0.133)$) and, especially, the blue cluster ($\widehat{\sigma} \in (0.279; 0.300)$) contain households with dramatic changes in their income between consecutive years. The small red cluster contains the extremely low values of income

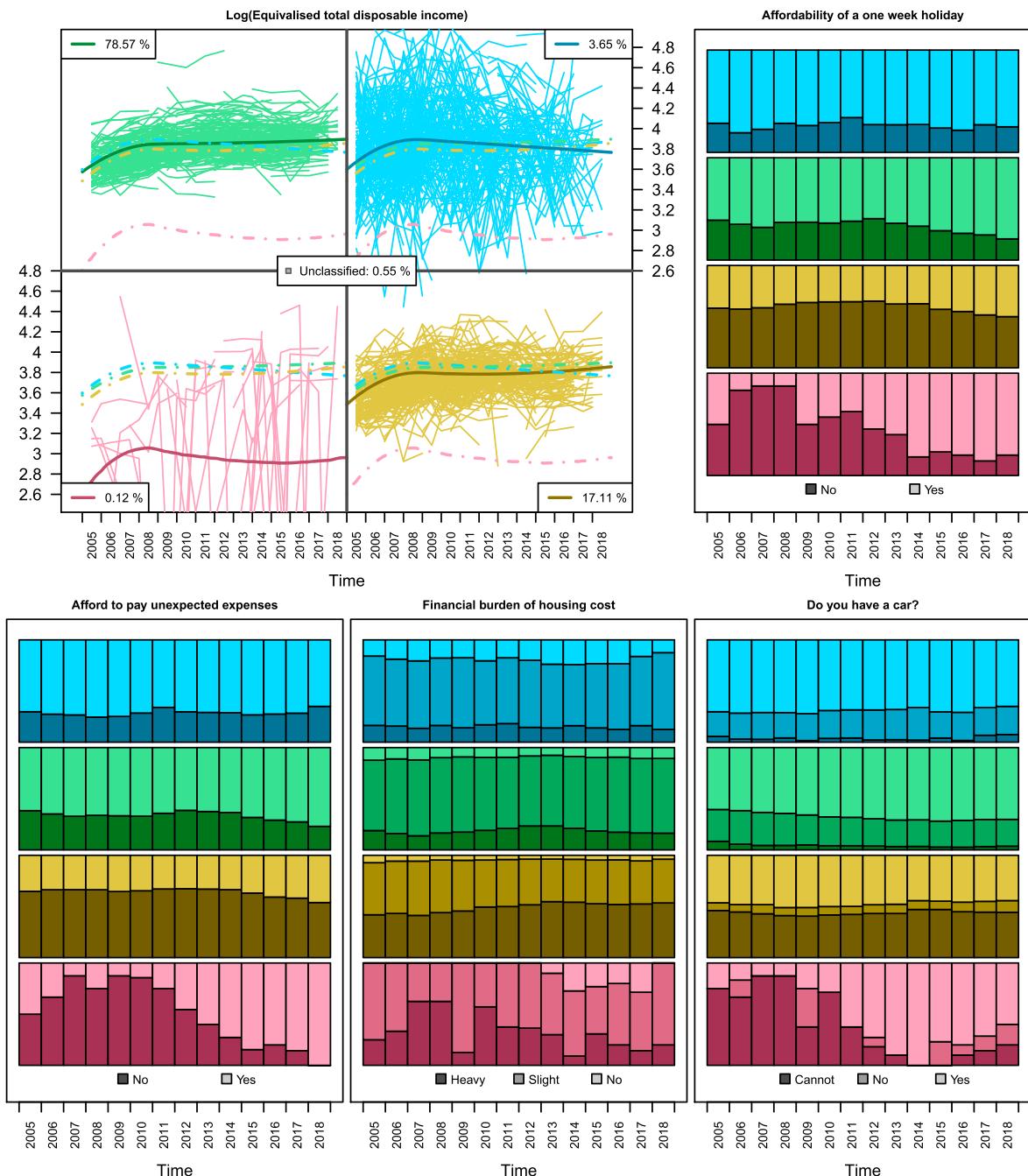


Fig. 5 Visualisation of the evolution of five (out of the eight) outcome variables across time when grouped into $\widehat{G}_+ = 4$ clusters. The upper left plot shows the observed values for the classified households together with the estimated median posterior curves for the log-transformed numeric outcome variable *Equivalised total disposable income* for a representative household of size 1.5 from a thinly-populated area, with

secondary educational level the highest achieved level of all household members and without having a baby or a student as household member. The other plots visualise the empirical frequencies of the categorical outcome variables after classifying households obtained separately for each year

(including the few negative income observations) that also heavily fluctuate from one year to another.

The posterior estimates for the other covariates included in the regression with a constant effect for the whole population indicate, based on the posterior medians and the 95% equitailed credible intervals, that living in more densely populated areas (town, city, Prague) increases the expected *Equivalised total disposable income* by 1.27% (0.93%; 1.61%), 2.12% (1.68%; 2.48%), 8.29% (7.63%; 8.98%), respectively, compared to living in a thinly populated village. Having one additional adult within a household increases the expected *Equivalised total disposable income* by 2.58% (2.41%; 2.73%). Taking care of a baby, respectively having a student, within a household decreases the expected *Equivalised total disposable income* by 5.44% (5.08%; 5.76%), respectively 3.30% (3.02%; 3.56%). Having as highest education level a secondary, respectively upper-secondary or tertiary education level within a household increases the expected *Equivalised total disposable income* by 11.02% (10.61%; 11.37%), respectively 20.44% (19.85%; 21.08%), compared to a situation when the highest education level achieved by all household members corresponds to the lower-secondary educational level.

The estimated curves for the categorical outcome variables across time were more or less flat for all four clusters, but they differed in their levels across clusters. This agrees with Fig. 5 where we barely see any evolution of the ratios in time within any of the main clusters (blue, green, yellow). These constant ratios, however, correspond to the interpretation of the clusters obtained so far. Households within the blue cluster most probably own a car, can afford a week holiday away from home, have capacity to pay for unexpected expenses and the housing cost does not seem to be a burden to them when compared to other clusters. On the other hand, the majority of households within the yellow cluster cannot afford a car, nor a week holiday, nor pay for unexpected expenses. They also agree to the highest extent with the statement that housing cost is a heavy financial burden. Households within the green cluster (the majority) are comparable to the prosperous ones in the blue cluster, but they are in general a bit worse off. We obtained an analogous interpretation for the other outcomes which are not included in Fig. 5 and these results are hence not shown.

9 Conclusion

This paper proposes an approach which allows to infer clusters from multivariate longitudinal data of possibly different types by building on and combining several different methodologies. The joint modelling by mixtures of GLMMs allows us to combine an arbitrary number of numeric, binary, ordinal or general categorical outcome variables and provides

a natural way for inclusion of other types of outcomes. The GLM-based suitable combination of distributional family and link function is used for each outcome type. This facilitates model extension to other outcome types, e.g., Poisson log-linear mixed-effects models could be considered for count data and this is already available in the current version of the software implementation.

The linear predictor may consist of group-specific as well as common fixed effects. The random effects are assumed to follow a multivariate normal distribution with a general covariance matrix and are allowed to be correlated not only within a single outcome but also across all outcome variables. This accounts for correlation between observations from the same subject even after accounting for group differences and any covariate effects in the regression. A finite mixture model is specified to embed the clusterwise regression problem into an MBC framework.

The Bayesian approach is pursued for model estimation and inference exploiting the possibility to determine the number of data clusters based on a sparse finite mixture approach, specify priors which have a regularising effect on the mixture likelihood and fully exploit the hierarchical structure and the latent variable framework using Bayesian data augmentation in MCMC inference. The sampler is implemented in the C language and wrapped in a user-friendly R interface and the implementation is available from GitHub at https://github.com/vavrajan/MBC_GLMMs together with a set of tutorials demonstrating the use.

The performance of the proposed approach is evaluated in a simulation study indicating the benefits of jointly modelling the outcome variables to improve the clustering abilities as well as highlighting the accuracy of the parameter estimates obtained from an identified mixture model. The applications demonstrate how the proposed approach helps analysing medical and economic survey data indicating the wide potential in many different areas such as health care, psychology, social sciences and many more.

The proposed model allows for group-specific as well as group-invariant parameters. In this paper, we focused on application settings where a-priori the context is informative of this specification. In particular, we required certain parameters to be group-specific or group-invariant to ease interpretation. We believe that for many applications this choice is to be made by the practitioner based on considerations of what characterises a useful clustering (Hennig 2015). Nevertheless, future research could aim at developing tools which allow to decide this specification in a data-driven way for a selected set of parameters. The usual model selection criteria such as the marginal likelihood are not very attractive to be used in this setting and we would rather expect that developing an analogue of the horseshoe prior (Carvalho et al. 2009, 2010) used for variable selection in regression analysis would seem promising which forces the posterior

to either concentrate mass on the null setting (in our case group-invariant parameters) or the alternative setting (in our case group-specific parameters). Further interest could be in a sparse specification of the variance-covariance matrix of the random effects as well as a suitable data-driven estimation.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s11222-023-10304-5>.

Acknowledgements The Czech authors acknowledge support by the GAČR Grant No. 21-13323S and by the Charles University project GA UK No. 298120. The Austrian authors acknowledge support from the Austrian Science Fund (FWF) Grant P28740.

Author Contributions JV, AK, BG and GM-W contributed to developing of the methodology presented in the manuscript. JV provided the software implementation of the method, conducted the empirical analyses, performed the simulation study and wrote the first draft of the main manuscript text. All authors edited and reviewed the manuscript.

Declarations

Competing interests The authors declare no competing interests.

References

- Breslow, N.E., Clayton, D.G.: Approximate inference in generalized linear mixed models. *J. Am. Stat. Assoc.* **88**(421), 9–25 (1993). <https://doi.org/10.2307/2290687>
- Brooks, S., Gelman, A., Jones, G., et al.: Handbook for Markov Chain Monte Carlo, 2nd edn. Taylor & Francis (2011). <https://doi.org/10.1201/b10905>
- Carvalho, C. M., Polson, N. G., Scott, J. G.: Handling sparsity via the horseshoe. In: van Dyk, D., Welling, M. (eds) Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, pp. 73–80 (2009)
- Carvalho, C.M., Polson, N.G., Scott, J.G.: The horseshoe estimator for sparse signals. *Biometrika* **97**(2), 465–480 (2010). <https://doi.org/10.1093/biomet/asq017>
- Celeux, G., Martin, O., Lavergne, C.: Mixture of linear mixed models for clustering gene expression profiles from repeated microarray experiments. *Stat. Model.* **5**(3), 243–267 (2005). <https://doi.org/10.1191/1471082x05st096oa>
- Fieuws, S., Verbeke, G.: Joint modelling of multivariate longitudinal profiles: pitfalls of the random-effects approach. *Stat. Med.* **23**, 3093–3104 (2004). <https://doi.org/10.1002/sim.1885>
- Fieuws, S., Verbeke, G.: Pairwise fitting of mixed models for the joint modeling of multivariate longitudinal profiles. *Biometrics* **62**(2), 424–431 (2006). <https://doi.org/10.1111/j.1541-0420.2006.00507.x>
- Fitzmaurice, G., Davidian, M., Verbeke, G., et al.: Longitudinal Data Analysis. CRC Press, Boca Raton (2008)
- Fralley, C., Raftery, A.E.: Model-based clustering, discriminant analysis, and density estimation. *J. Am. Stat. Assoc.* **97**(458), 611–631 (2002). <https://doi.org/10.1198/016214502760047131>
- Frühwirth-Schnatter, S.: Dealing with Label Switching under Model Uncertainty, Chap. 10. Wiley, pp. 213–239 (2011)
- Frühwirth-Schnatter, S., Malsiner-Walli, G.: From here to infinity: Sparse finite versus Dirichlet process mixtures in model-based clustering. *Adv. Data Anal. Classif.* **13**(1), 33–64 (2019). <https://doi.org/10.1007/s11634-018-0329-y>
- Grün, B., Leisch, F.: Finite mixtures of generalized linear regression models. In: Shalabh, H. C. (eds) Recent Advances in Linear Models and Related Areas, pp. 205–230. Springer (2008) https://doi.org/10.1007/978-3-7908-2064-5_11
- Hartzel, J., Agresti, A., Caffo, B.: Multinomial logit random effects models. *Stat. Model.* **1**, 81–102 (2001). <https://doi.org/10.1177/1471082x0100100201>
- Hennig, C.: Identifiability of models for clusterwise linear regression. *J. Classif.* **17**(2), 273–296 (2000). <https://doi.org/10.1007/s003570000022>
- Hennig, C.: What are the true clusters. *Pattern Recogn. Lett.* **64**, 53–62 (2015). <https://doi.org/10.1016/j.patrec.2015.04.009>
- Komárek, A., Komárková, L.: Capabilities of R package mixAK for clustering based on multivariate continuous and discrete longitudinal data. *J. Stat. Softw.* **59**(12):1–38 (2014). <https://doi.org/10.18637/jss.v059.i12>
- Komárek, A., Komárková, L.: Clustering for multivariate continuous and discrete longitudinal data. *Ann. Appl. Stat.* **7**(1), 177–200 (2013). <https://doi.org/10.1214/12-aoas580>
- Laird, N.M., Ware, J.H.: Random-effects models for longitudinal data. *Biometrics* **38**(4), 963–974 (1982). <https://doi.org/10.2307/2529876>
- Malsiner-Walli, G., Frühwirth-Schnatter, S., Grün, B.: Model-based clustering based on sparse finite Gaussian mixtures. *Stat. Comput.* **26**, 303–324 (2016). <https://doi.org/10.1007/s1122-014-9500-2>
- Pinheiro, J.C., Chao, E.C.: Efficient Laplacian and adaptive Gaussian quadrature algorithms for multilevel generalized linear mixed models. *J. Comput. Graph. Stat.* **15**(1), 58–81 (2006). <https://doi.org/10.1198/106186006x96962>
- Proust-Lima, C., Philipps, V., Diakite, A., et al.: Estimation of extended mixed models using latent classes and latent processes: The R package lcmm. *J. Stat. Softw.* **78**(2), 1–56 (2017)
- R Core Team: R: A Language and Environment for Statistical Computing. In: R Foundation for Statistical Computing, Vienna, Austria, <http://www.R-project.org> (2022)
- Rand, W.M.: Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**(336), 846–850 (1971). <https://doi.org/10.1080/01621459.1971.10482356>
- Redner, R.A., Walker, H.F.: Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev.* **26**(2), 195–239 (1984)
- Roeder, K., Wasserman, L.: Practical Bayesian density estimation using mixtures of normals. *J. Am. Stat. Assoc.* **92**(439), 894–902 (1997)
- Tan, Z., Shen, C., Subbarao, P., et al.: A joint modeling approach for clustering mixed-type multivariate longitudinal data: application to the CHILD cohort study (2022). <https://doi.org/10.48550/ARXIV.2210.08385>, arXiv:2210.08385
- Tanner, M.A., Wong, W.H.: The calculation of posterior distributions by data augmentation. *J. Am. Stat. Assoc.* **82**(398), 528–550 (1987). <https://doi.org/10.2307/2289457>
- Therneau, T.M., Grambsch, P.M.: Modeling Survival Data: Extending the Cox Model. Springer, New York (2000)
- Vávra, J., Komárek, A.: Classification based on multivariate mixed type longitudinal data: with an application to the EU-SILC database. *Adv. Data Anal. Classif.* (2022). <https://doi.org/10.1007/s11634-022-00504-8>

Verbeke, G., Lesaffre, E.: A linear mixed-effects model with heterogeneity in the random-effects population. *J. Am. Stat. Assoc.* **91**(433), 217–221 (1996). <https://doi.org/10.1080/01621459.1996.10476679>

Villarroel, L., Marshall, G., Barón, A.E.: Cluster analysis using multivariate mixed effects models. *Stat. Med.* **28**(20), 2552–2565 (2009). <https://doi.org/10.1002/sim.3632>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Angela Montanari

Material list:

Montanari A. (2024) The written Italian of university students: identification of tendencies via mixtures of GLLVM for count data. WG slides.

The written Italian of university students: identification of tendencies via mixtures of GLLVM for count data

Angela Montanari

Joint with

Laura Anderlucci, Silvia Dallari

Department of Statistical Sciences, University of Bologna

Nicola Grandi

Department of Classical Philology and Italian Studies, University of Bologna

Bertinoro - July 23rd 2024

Introduction Background

Background

- The 'state of health' of the Italian language has been recently subject to a lively debate, especially concerning the language used by young people.
- Univers-ITA, a recently closed project funded by the Italian Ministry of Education, has addressed the problem from different perspectives, including an analysis of purposely written formal texts.



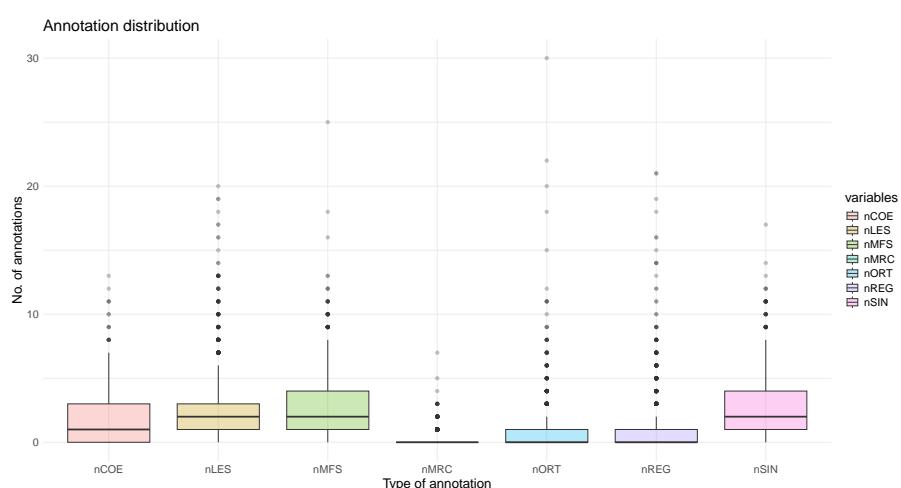
The goal of this work is to draw an exhaustive picture of the Italian written by university students, integrating the sociolinguistic and the typological perspectives.

The Data

A formal text of at most 500 words written by each student has been annotated.

- **Observations** → 2160 university students from different Italian universities.
After cleaning the data 2137 individuals have been taken into account.
- **Variables (Count)** → 7 different annotations:
 1. orthography (nORT)
 2. linguistic register (nREG)
 3. marked sentences (nMRC)
 4. lexicon (nLES)
 5. morphosyntax (nMFS)
 6. coherence (nCOE)
 7. syntax (nSIN)
- **Covariates** → variables related to the socio-economic condition of the respondents, together with other information such as the geographical area of the university and the diploma they obtained in high school.

Research Question



- The situation is really good, but...
⇒ what are the features that best characterize the written Italian? And are there groups of subjects who use different annotated variants?

Mixtures of GLLVM for count data

Starting from the model of Cagnone and Viroli (2012), we propose a mixture of GLLVM for count data where, for each individual, the Poisson distributed p -dimensional vector \mathbf{y} of observed count variables is related with a q -dimensional latent vector \mathbf{z} distributed according to a finite mixture of multivariate Gaussians:

Latent layers:

$$f(\mathbf{z}) = \sum_{i=1}^k \omega_i \phi_i^{(q)}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

$$f(\mathbf{s}) = \prod_{i=1}^k \omega_i^{s_i}$$

Observation layer:

$$f(\mathbf{y}|\mathbf{z}) = \prod_{j=1}^p f(y_j|\mathbf{z}) = \prod_{j=1}^p \frac{\pi_j(\mathbf{z})^{y_j} e^{-\pi_j(\mathbf{z})}}{y_j!}$$

- $\omega_i > 0$ and $\sum_{i=1}^k \omega_i = 1$
- $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the parameters of the q -variate normal density $\phi_i^{(q)}$
- \mathbf{s} is the allocation variable

The relation between \mathbf{y} and \mathbf{z} is modelled through the link function:

$$\log(\pi(\mathbf{z})) = \boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\mathbf{z}$$

where $\boldsymbol{\lambda}_0$ is the $p \times 1$ vector containing the intercepts, $\boldsymbol{\Lambda}$ is the $p \times q$ factor loading matrix, $\pi(\mathbf{z}) = (\pi_1(\mathbf{z}), \dots, \pi_p(\mathbf{z}))$ and $\pi_j(\mathbf{z}) = P(y_j = 1|\mathbf{z})$.

Identifiability conditions

In order to get unique and consistent estimates of the parameters, identifiability conditions need to be met. For this purpose the following restrictions are imposed:

- 1 $E(\mathbf{z}) = \sum_{i=1}^k \omega_i \boldsymbol{\mu}_i = \mathbf{0}$;
 $Var(\mathbf{z}) = \sum_{i=1}^k \omega_i (\boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i \boldsymbol{\mu}_i^T) = \mathbf{I}_q$
- 2 The upper right triangle of $\boldsymbol{\Lambda}$ is set equal to 0 (following the proposal of Jöreskog (1969) of fixing $q(q - 1)/2$ loadings equal to zero).
- 3 $\lambda_{10} = 0$

Extension with covariates

Since metadata are available, covariates can also be added to the model. Specifically, they are included in the mixture weights following a multinomial logit regression (Fokouè, 2005). Supposing m covariates are available, and calling the resulting $n \times (m + 1)$ matrix \mathbf{X} , we get:

$$\omega_i(\mathbf{X}) = \frac{\exp(\boldsymbol{\eta}_i^T \mathbf{X})}{1 + \sum_{i'=1}^{k-1} \exp(\boldsymbol{\eta}_{i'}^T \mathbf{X})},$$

with $i = 1, \dots, k$ and $\boldsymbol{\eta}_k = 0$ for identifiability reasons.

The gradient does not offer an explicit solution for $\boldsymbol{\eta}_i$, $i = 1, \dots, k \rightarrow$ a Newton-Raphson algorithm is adopted.

Covariate Selection

We suggest to choose the set of covariates by fitting a multinomial logistic model with a lasso or grouped-lasso penalty.

Model Estimation Strategy

- Model estimation is performed via a generalized EM algorithm (Dempster et al., 1977). [appendix](#)
- Some integrals cannot be solved analytically \rightarrow numerically approximated using a weighted sum over a finite number of points with weights given by Gauss-Hermite quadrature points (see Straud and Sechrest, 1966).
- $\tilde{\boldsymbol{\Lambda}} = (\lambda_0, \boldsymbol{\Lambda})$ is not in closed form \rightarrow it is derived with an iterative Newton-Raphson procedure.

Simulation Study

BIC and AIC's ability to identify the true number of clusters and factors has been assessed in a wide simulation study considering different combinations of q, k, n, p . Here the results for three scenarios are reported (100 reps):

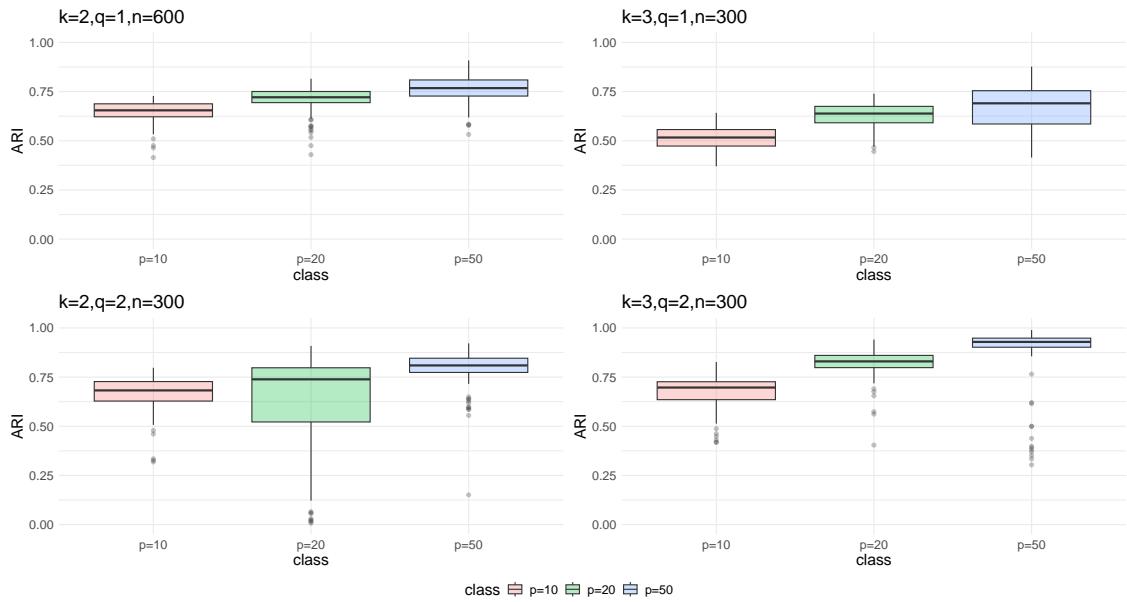
BIC				AIC				
True model specification: $k = 1, q = 2, n = 300, p = 10$								
	$k=1$	$k=2$	$k=3$	$k=4$	$k=1$	$k=2$	$k=3$	$k=4$
$q=1$	13%	0%	0%	0%	0%	0%	0 %	0%
$q=2$	81%	0%	0%	0%	71%	0%	0%	0%
$q=3$	6%	0%	0%	0%	29%	0%	0%	0%

True model specification: $k = 2, q = 2, n = 300, p = 10$								
	$k=1$	$k=2$	$k=3$	$k=4$	$k=1$	$k=2$	$k=3$	$k=4$
$q=1$	0%	0%	0%	0%	0%	0%	0%	0%
$q=2$	0%	88%	5%	1%	0%	64%	13%	2%
$q=3$	2%	4%	0%	0%	0%	17%	4%	0%

True model specification: $k = 3, q = 2, n = 300, p = 10$								
	$k=1$	$k=2$	$k=3$	$k=4$	$k=1$	$k=2$	$k=3$	$k=4$
$q=1$	0%	0%	0%	0%	0%	0%	0%	0%
$q=2$	0%	90%	10%	0%	0%	13%	83%	1%
$q=3$	0%	0%	0%	0%	0%	1%	2%	0%

Simulation Study

The recovery of the true clustering structure (k and q known) is tested via the Adjusted Rand Index (ARI) of some scenarios (100 reps):



Real Data (without covariates)

We took into account all the possible combinations given by $q = 1, 2$ and $k = 1, 2, 3, 4, 5$. The best combination according to BIC and AIC is the one with $q = 2, k = 1$. The oblimin rotated factor loading matrix is:

	$\hat{\lambda}_1$	$\hat{\lambda}_2$
nCOE		0.53
nLES		0.71
nMFS		0.56
nMRC	0.71	
nORT	0.66	0.41
nREG	1.34	
nSIN		0.45

- $\hat{\lambda}_1$ discriminates students who make inappropriate choices with respect to the context.
- $\hat{\lambda}_2$ represents difficulties in structuring a complex text as the one requested.

The correlation between the two factors is 0.37.

Real Data (with covariates)

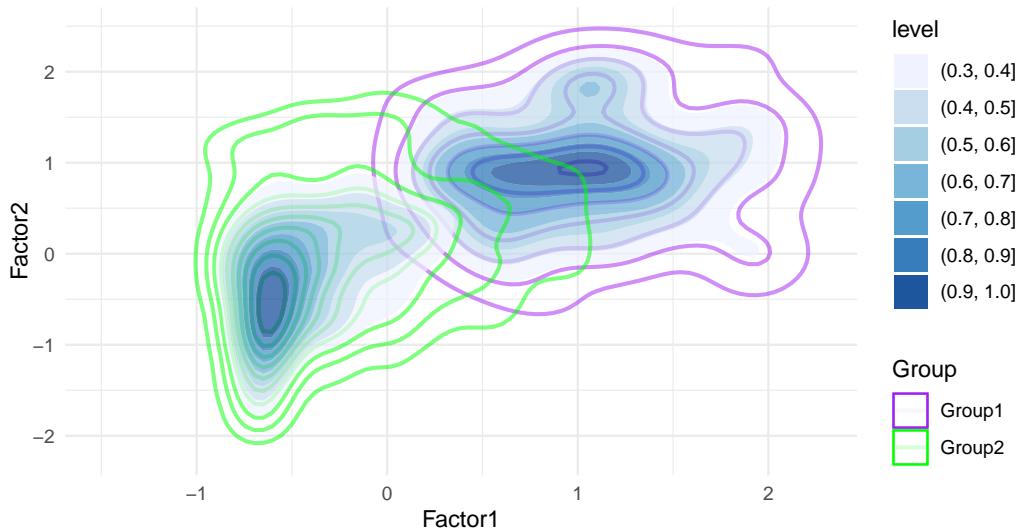
Among the original 11 initial covariates, the lasso-based model selects the following 4 predictors:

- 1) *Family origin* (Italy/Mixed/Abroad)
- 2) *High school diploma* (Lyceum/Professional/Technical)
- 3) *Place of birth* (Italy/Other Country)
- 4) *Knowledge of ancient languages* (Yes/No)

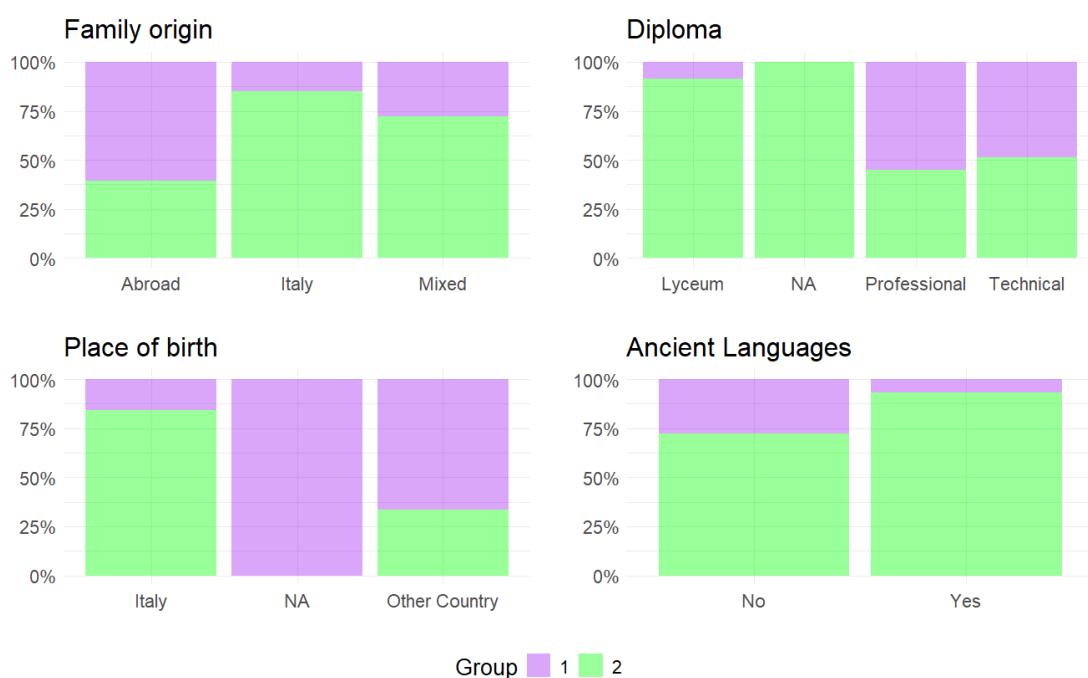
Moreover, the optimal model involves **two latent factors** and **two groups**: group 1 includes 376 students, group 2 the remaining 1761.

Real Data (with covariates)

The 2D kernel densities of the two groups obtained are:



Real Data (with covariates)



Conclusions

- The simulation results showed that mixtures of Poisson GLLVM are an interesting tool to cluster individuals, while allowing for correlations among the counts.
- The real data results suggest the presence of two types of annotation, one linked to inadequate choices and the other one connected to weaknesses in organizing a formal text, but these two are not uncorrelated.
- Students with Italian origins, a lyceum diploma and who know ancient languages tend to show less annotations than others.

References

- Cagnone, S. and Viroli, C. (2012). A factor mixture analysis model for multivariate binary data, *Statistical Modelling*, 12(3), pp.257–277.
- Dempster, A.P., Laird, N.M., Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1), 1–38.
- Fokouè, E. (2005). Mixtures of factor analyzers: an extension with covariates, *Journal of Multivariate Analysis*, 95: 370–384.
- Joëreskog, K.G. (1969). A general approach to confirmatory maximum likelihood factor analysis, *Psychometrika*, 34(2), 183–202.
- Straud, A.H. and Sechrest, D. (1966). Gaussian quadrature formulas, *Englewood Cliffs, NJ: Prentice Hall*.
- Univers-ITA: <https://site.unibo.it/univers-ita/it>

Generalized EM algorithm

Since the complete density can be written as:

$$f(\mathbf{y}, \mathbf{z}, \mathbf{s}, \mathbf{x}; \boldsymbol{\theta}) = f(\mathbf{y}|\mathbf{z}, \mathbf{x}; \boldsymbol{\theta})f(\mathbf{z}|\mathbf{s}, \mathbf{x}; \boldsymbol{\theta})f(\mathbf{s}|\mathbf{x}; \boldsymbol{\theta}),$$

where $\boldsymbol{\theta} = (\boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\omega}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, the EM algorithm consists of maximizing:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}} E_{\mathbf{z}, \mathbf{s} | \mathbf{y}, \mathbf{x}, \boldsymbol{\theta}'} \left[\log f(\mathbf{y}|\mathbf{z}, \mathbf{x}; \boldsymbol{\theta}) + \log f(\mathbf{z}|\mathbf{s}, \mathbf{x}; \boldsymbol{\theta}) + \log f(\mathbf{s}|\mathbf{x}; \boldsymbol{\theta}) \right]$$

The steps of the EM algorithm (Dempster et al., 1977) can be summarized as:

1. Choose the starting values of $\boldsymbol{\theta}' = (\tilde{\boldsymbol{\Lambda}}', \boldsymbol{\omega}', \boldsymbol{\mu}', \boldsymbol{\Sigma}')$, where $\tilde{\boldsymbol{\Lambda}} = (\boldsymbol{\lambda}_0, \boldsymbol{\Lambda})$.
2. Find:
 - $\tilde{\boldsymbol{\Lambda}}$ that maximizes $E_{\mathbf{z} | \mathbf{y}, \mathbf{x}, \boldsymbol{\theta}'} [\log f(\mathbf{y}|\mathbf{z}, \mathbf{x}; \boldsymbol{\theta})]$
 - $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ maximizing $E_{\mathbf{z}, \mathbf{s} | \mathbf{y}, \mathbf{x}, \boldsymbol{\theta}'} [\log f(\mathbf{z}|\mathbf{s}, \mathbf{x}; \boldsymbol{\theta})]$
 - $\boldsymbol{\omega}$ that maximizes $E_{\mathbf{s} | \mathbf{y}, \mathbf{x}, \boldsymbol{\theta}'} [\log f(\mathbf{s}|\mathbf{x}; \boldsymbol{\theta})]$
 - Set $\boldsymbol{\theta}' = \boldsymbol{\theta}$
3. Repeat Step 2 until convergence.

main

Adrian Raftery

Material list:

Martin Metodiev M., Perrot-Dockès M., Ouadah S., Irons N.J., Latouche P., Raftery A.E. (2024) Easily Computed Marginal Likelihoods from Posterior Simulation Using the THAMES Estimator. *Bayesian Analysis*.

Easily Computed Marginal Likelihoods from Posterior Simulation Using the THAMES Estimator

Martin Metodiev^{*,†}, Marie Perrot-Dockès[†], Sarah Ouadah[‡], Nicholas J. Irons^{§,¶,||},
 Pierre Latouche^{*,†,¶}, and Adrian E. Raftery^{§,¶,||}

Abstract. We propose an easily computed estimator of the marginal likelihood from posterior simulation output, via reciprocal importance sampling, combining earlier proposals of DiCiccio et al (1997) and Robert and Wraith (2009). This involves only the unnormalized posterior densities from the sampled parameter values, and does not involve additional simulations beyond the main posterior simulation, or additional complicated calculations, provided that the parameter space is unconstrained. Even if this is not the case, the estimator is easily adjusted by a simple Monte Carlo approximation. It is unbiased for the reciprocal of the marginal likelihood, consistent, has finite variance, and is asymptotically normal. It involves one user-specified control parameter, and we derive an optimal way of specifying this. We illustrate it with several numerical examples.

MSC2020 subject classifications: Primary 62F15, 62-04; secondary 62F12.

Keywords: marginal likelihood estimation, reciprocal importance sampling.

1 Introduction

A key quantity in Bayesian model selection is the marginal likelihood, also known as the evidence, the normalizing constant of the posterior density, or the integrated likelihood. Consider a statistical model with parameter vector θ and data \mathcal{D} . Let $L(\theta) = p(\mathcal{D}|\theta)$ be the usual likelihood, and $\pi(\theta)$ be the prior distribution of θ . Then $Z = p(\mathcal{D}) = \int L(\theta)\pi(\theta)d\theta$ is the marginal likelihood.

The marginal likelihood plays a key role in defining Bayes factors. Consider two models M_1 and M_2 with marginal likelihoods Z_1 and Z_2 . Then the Bayes factor (or ratio of posterior to prior odds) for model M_1 against M_2 is $B_{1,2} = Z_1/Z_2$.

The marginal likelihood is also a critical quantity for Bayesian model averaging (BMA). Consider K models, M_1, \dots, M_K , with prior model probabilities Π_k (which

arXiv: [2305.08952](https://arxiv.org/abs/2305.08952)

*Université Clermont Auvergne, Laboratoire de Mathématiques Blaise Pascal,
martin.metodiev@doctorant.uca.fr; Pierre.LATOUCHE@uca.fr

†Université Paris Cité, CNRS, MAP5, F-75006 Paris, France, marie.perrot-dockees@u-paris.fr

‡Université Paris-Saclay, AgroParisTech, INRAE, UMR MIA Paris-Saclay, sarah.ouadah@agroparistech.fr

§University of Washington, Department of Statistics, njirons@uw.edu

¶Equal contribution.

||Corresponding author, raftery@uw.edu.

add up to 1), and marginal likelihoods Z_k . Suppose Q is a quantity of interest, such as a parameter or a future observation to be predicted. Then the BMA posterior distribution of Q is

$$p(Q|\mathcal{D}) = \sum_{k=1}^K p(Q|\mathcal{D}, M_k) p(M_k|\mathcal{D}), \quad (1)$$

where $p(M_k|\mathcal{D})$ is the posterior model probability of M_k , which satisfies $p(M_k|\mathcal{D}) \propto \Pi_k Z_k$ and $\sum_{k=1}^K p(M_k|\mathcal{D}) = 1$. So $p(Q|\mathcal{D}) = \sum_{k=1}^K p(Q|\mathcal{D}, M_k) \Pi_k Z_k / \sum_{k=1}^K \Pi_k Z_k$.

Finally, the most likely model *a posteriori* is the one that maximizes $\Pi_k Z_k$. Choosing it minimizes the model selection error rate on average over the prior (Jeffreys, 1961). Often the prior over the model space is chosen to be uniform, in which case $\Pi_k = 1/K$, $\forall k$. In this case, Bayesian model selection by choosing the most likely model *a posteriori* boils down to choosing the model with the largest Z_k , and hence involves only the marginal likelihoods.

Bayesian models are often estimated using Monte Carlo methods in which a sample of values of θ is simulated from the posterior distribution. The most common class of such methods is Markov chain Monte Carlo (MCMC). Perhaps surprisingly, estimating the marginal likelihood from the output of MCMC and other posterior simulation methods has turned out not to be straightforward. Many different methods have been proposed, and none of them is widely considered to be generally the best. Llorente et al. (2023) provide a comprehensive review of such methods, describing 16 different methods and, remarkably, cite over 20 *other* review articles!

We seek a method that is precise, generic and simple for estimating the marginal likelihood from posterior simulation output. We take this to mean that it gives low variance estimates of the marginal likelihood, uses posterior simulation output for just the one model being analyzed, uses only likelihoods and prior densities of the sampled values of θ , and does not need additional simulations or complicated calculations.

Some well-known methods do not satisfy our desiderata. These include Chib's method (Chib, 1995), which requires complicated additional calculations, bridge sampling (Meng and Wong, 1996), which requires simulations from two models, importance sampling, which requires additional simulations, nested sampling (Skilling, 2006), which involves other simulations, and more advanced methods such as adaptive annealed importance sampling (Liu, 2014). They also include the harmonic mean of the likelihoods (Newton and Raftery, 1994), which is unbiased and consistent, but has infinite variance and is unstable, as pointed out by the original authors.

Arguably, the only methods that are precise, generic and simple for estimating the marginal likelihood from MCMC by our definition are versions of reciprocal importance sampling (RIS) (Gelfand and Dey, 1994). These are based on the identity:

$$Z^{-1} = E_\theta \left[\frac{h(\theta)}{L(\theta)\pi(\theta)} \middle| \mathcal{D} \right], \quad (2)$$

where $h(\theta)$ is a (normalized) probability density function (pdf) over the posterior sup-

port. Remarkably, (2) holds for any pdf $h(\theta)$. This leads to the estimator

$$\hat{Z}^{-1} = \frac{1}{T} \sum_{t=1}^T \frac{h(\theta^{(t)})}{L(\theta^{(t)})\pi(\theta^{(t)})}, \quad (3)$$

where $\theta^{(1)}, \dots, \theta^{(T)}$ are simulated from the posterior using MCMC or another method. This estimator has good properties in general, provided that the tails of the distribution $h(\theta)$ are thin enough in all directions. It can be hard to choose $h(\theta)$ so that it both overlaps substantially with the posterior distribution (needed for efficiency) and has thin enough tails (needed for finite variance), especially in higher dimensions. We propose a choice of $h(\theta)$ that leads to easily computed estimates and is optimal or near optimal in a certain sense.

The paper is organized as follows. In Section 2 we discuss reciprocal importance sampling and its properties. In Section 3 we describe our proposed choice of $h(\theta)$ and derive some of its properties. In Section 4 we give several numerical examples, including a multivariate Gaussian example, a Bayesian regression example, a non-Gaussian case, and a Bayesian hierarchical model. We conclude in Section 5 with a discussion. The code for this paper is made available via Github for scientific dissemination at the following [link](#). The THAMES has been implemented in an R package (Irons et al., 2023).

2 Reciprocal importance sampling

In general, the RIS estimator of the marginal likelihood is defined by Equation (3). This has several good properties. It is unbiased, in the sense that $E[\hat{Z}^{-1}] = Z^{-1}$, where the expectation is over the posterior distribution of θ . It is also strongly simulation-consistent, in the sense that $\hat{Z}^{-1} \rightarrow Z^{-1}$ almost surely as $T \rightarrow \infty$.

In addition, the RIS estimator of the reciprocal marginal likelihood, \hat{Z}^{-1} , has finite variance and is asymptotically normally distributed as $T \rightarrow \infty$ if the tails of $h(\theta)$ are thin enough. Specifically, this requires that

$$\int \frac{h(\theta)^2}{L(\theta)\pi(\theta)} d\theta < \infty. \quad (4)$$

It is hard to choose $h(\theta)$ so that it both overlaps substantially with the area of the parameter space with high posterior density, which is needed for efficiency, and so that it also has thin enough tails, which is needed for finite variance. The difficulty grows as the dimension increases.

Two choices of $h(\theta)$ in the literature deserve attention. DiCiccio et al. (1997) proposed $h(\theta) = MVN(\theta; \hat{\theta}, \hat{\Sigma})$, where $\hat{\theta}$ is the posterior mean or mode, and $\hat{\Sigma}$ is an estimate of the posterior covariance matrix. This overlaps nicely with $L(\theta)\pi(\theta)$, but its tails may not be thin enough when the posterior is asymmetric or the parameter is high-dimensional.

To remedy the problem of the tails possibly being too thick, DiCiccio et al. (1997) proposed truncating it, using instead $h(\theta) = TMVN_A(\hat{\theta}, \hat{\Sigma})$, a multivariate normal distribution truncated to the set A , where

$$A = \{\theta : (\theta - \hat{\theta})^T \hat{\Sigma}^{-1} (\theta - \hat{\theta}) < c^2\}. \quad (5)$$

Thus A is an ellipsoid with radius c and volume

$$V(A) = c^d \pi^{d/2} |\hat{\Sigma}|^{1/2} / \Gamma\left(\frac{d}{2} + 1\right). \quad (6)$$

Truncating the distribution ensures that the estimator \hat{Z}^{-1} has finite variance. The authors found that the truncation improved the performance of the RIS estimator. However, with high-dimensional parameters, the result might be sensitive to the specification of c .

Robert and Wraith (2009) proposed setting $h(\theta)$ to be a uniform distribution on the convex hull of simulated MCMC parameters values in the α -HPD region, namely the highest posterior density region containing a proportion α of the sampled parameter values. They considered the values $\alpha = 0.1$ and 0.25 . They applied it to a two-dimensional toy example where it performed well.

However, as far as we know, that method has not yet been fully developed for realistic, higher-dimensional situations. For example, we know of no simple way to compute the volume of the convex hull of a set of points in higher dimensions, which is required for the method in general. It is also not clear how best to choose α nor how sensitive the method would be to α in higher dimensions. It has been used in a higher-dimensional application by Durmus et al. (2018), but this involved comparing competing models defined on the same parameter space, thus avoiding the need to calculate the volume of A , which canceled out in Bayesian model comparisons. Calculating the volume of A may be the most difficult part of this method in general.

3 Estimating the marginal likelihood

3.1 Estimating the marginal likelihood with THAMES

We propose combining the proposals of DiCiccio et al. (1997) and Robert and Wraith (2009) to obtain a method that we believe satisfies all our desiderata. We propose specifying $h(\theta)$ to be a uniform distribution over the set A defined in Equation (5), rather than over a convex hull of points. This resolves the problem of computing the volume of A , since this is given analytically by Equation (6). If A is not a subset of the posterior support, for example if the posterior support is constrained, we adjust the volume of A by a simple Monte Carlo approximation. This yields the estimator

$$\hat{Z}^{-1} = \frac{1}{V(A)T} \sum_{\substack{t=1 \\ \theta^{(t)} \in A}}^T \frac{1}{L(\theta^{(t)})\pi(\theta^{(t)})}. \quad (7)$$

Thus \hat{Z} is a truncated harmonic mean of the unnormalized posterior densities, $L(\theta^{(t)})\pi(\theta^{(t)})$.¹ We call it the Truncated HArmonic Mean EStimator, or THAMES.

The THAMES, \hat{Z}^{-1} , has several desirable properties. It is simple to compute, involving only the prior and likelihood values of the sampled parameter values. In fact it involves only the product of the prior and likelihood values, namely the unnormalized posterior densities of the sampled parameter values. It is unbiased as an estimator of Z^{-1} , as long as A is specified independently of the sample. It is also simulation-consistent, in the sense that $\hat{Z}^{-1} \rightarrow Z^{-1}$ almost surely as $T \rightarrow \infty$, by the strong law of large numbers. Its variance (over simulation from the posterior given the data \mathcal{D}) is finite provided that

$$\int_A (L(\theta)\pi(\theta))^{-1} d\theta < \infty, \quad (8)$$

which will usually hold since A is a bounded set in \mathbb{R}^d . In fact, it suffices that the likelihood and the prior are continuous with respect to θ and strictly positive on the closure of A . If Equation (8) holds, \hat{Z}^{-1} is asymptotically normal (again as the number of parameter values simulated increases), by the Lindeberg central limit theorem. Note that asymptotic normality holds on the scale of \hat{Z}^{-1} , and not exactly on other scales such as \hat{Z} or $\log(\hat{Z})$.

If the posterior simulation method yields independent draws, then $\text{Var}(\hat{Z}^{-1})$ can be estimated directly as the empirical variance of the values of $(L(\theta^{(t)})\pi(\theta^{(t)})\mathbb{1}(\theta^{(t)} \in A))^{-1}$, divided by $T \cdot V(A)^2$. If MCMC is used, successive simulations from the posterior will in general not be independent. A central limit theorem will still hold, but the variance needs to take account of the serial dependence. This can be done approximately by computing the variance based on serial independence and multiplying it by an estimate of the spectral density of the sequence at zero. For example, if the sequence of values of $1/(L(\theta)\pi(\theta))$ can be approximated by a first-order autoregressive model with parameter ϕ , then this would be approximately $1/(1 - \phi)^2$. An alternative would be to thin the sequence enough that the resulting subsequence is approximately uncorrelated and then use the variance based on assuming independence. A different approach was taken by Frühwirth-Schnatter (2004).

Note that an approximate normal confidence interval can be obtained for \hat{Z}^{-1} , because that is the scale on which a central limit theorem holds. This could be turned into a confidence interval for \hat{Z} by taking the reciprocals of the ends of the normal confidence intervals for \hat{Z}^{-1} ; the resulting confidence interval would not be symmetric. The same could be done for $\log(\hat{Z})$ in a similar manner.

3.2 Optimal choice of control parameter, c

We now address the question of how to choose the radius c of the ellipse that specifies the THAMES in Equation (5). Ignoring serial correlation between simulated values of the

¹Recall that the unstable harmonic mean estimator described by (Newton and Raftery, 1994) was quite different, not being truncated, and being a harmonic mean of the likelihoods rather than the unnormalized posterior density values.

parameters, we suggest choosing c to minimize the estimated variance of \hat{Z}^{-1} . This could be done empirically by computing \hat{Z}^{-1} for a range of values of c , estimating $\text{Var}(\hat{Z}^{-1})$ for each value of c , and optimizing it over c by a grid search or a one-dimensional numerical optimization method.

It is possible to obtain analytic results in the case where the posterior distribution is normal. This is of considerable interest as the posterior distribution is asymptotically normal in many common situations, including some where standard regularity conditions do not hold (Heyde and Johnstone, 1979; Ghosal, 2000; Shen, 2002; Miller, 2021). In this case the THAMES has finite variance since the posterior density, and thus the product of the likelihood and the prior, is continuous with respect to θ and strictly positive everywhere.

We want to minimize the variance of the THAMES. Due to our assumption of independence of all of the successive MCMC simulations, this variance can be simplified to

$$\text{Var}(\hat{Z}^{-1} | \mathcal{D}) = \frac{1}{T} \cdot \frac{1}{\bar{Z}^2} \cdot SCV(d, c). \quad (9)$$

Here $SCV(d, c)$ denotes

$$SCV(d, c) := \frac{\text{Var}_{\theta^{(1)}} \left(\frac{\mathbb{1}_A(\theta^{(1)})/V(A)}{L(\theta^{(1)})\pi(\theta^{(1)})} \middle| \mathcal{D} \right)}{E_{\theta^{(1)}} \left(\frac{\mathbb{1}_A(\theta^{(1)})/V(A)}{L(\theta^{(1)})\pi(\theta^{(1)})} \middle| \mathcal{D} \right)^2}, \quad (10)$$

the squared coefficient of variation of the first term of the THAMES. Since the variance is a product of $\frac{1}{T}$, $\frac{1}{\bar{Z}^2}$ and $SCV(d, c)$, minimizing $SCV(d, c)$ with respect to c is equivalent to minimizing the variance of the THAMES.

We derive a statement about the optimal choice of c by assuming that the posterior covariance matrix Σ and the posterior mean m can be provided by a stochastic oracle. The THAMES can then be defined using

$$A_{or} := \{\theta : (\theta - m)^T \Sigma^{-1} (\theta - m) < c^2\}. \quad (11)$$

We will show that the radius c that minimizes the variance of the THAMES depends on the dimension d , and is equal to $c_d = \sqrt{d + L_d}$ with L_d being close to one for large d . Interestingly, in this case the SCV depends neither on the data, \mathcal{D} , nor on the number of samples from the posterior, T . Of course, this is rarely exactly the case in practice. However, plugging in consistent estimators of (m, Σ) gives approximately the same results if the number of samples from the posterior is large enough, provided that a sample splitting procedure is used. This will be a consequence of Theorem 3.3. The sample splitting procedure that we suggest is described in Section 3.3. The proofs of these results are given in Supplement A (Metodiev et al., 2024a). Additional numerical results about the behaviour of the optimal radius are given in Supplement B (Metodiev et al., 2024b).

Assumption 1. For the following theorems it is assumed that we can ignore serial correlation (i.e. we assume independence of the successive MCMC iterations) and that the posterior distribution is normal with mean $m \in \mathbb{R}^d$ and positive definite covariance matrix $\Sigma \in \mathcal{M}_{d \times d}(\mathbb{R})$. We further assume that the THAMES is defined on A_{or} .

Theorem 3.1. *There exists a unique radius $c_d \in (0, \infty)$ such that the ellipsoid A_{or} with radius c_d minimizes the variance of the THAMES. This value c_d does not depend on the posterior mean or covariance matrix. It satisfies $c_d = \sqrt{d + L_d}$, where the optimal shifting parameter $L_d \geq 0$ is a sequence for which $\frac{L_d}{d} \xrightarrow{d \rightarrow \infty} 0$.*

Remark 1. Theorem 3.1 ensures that the optimal radius c_d is asymptotically equivalent to \sqrt{d} . In fact, our calculations suggest that $c_d = \sqrt{d + L_d}$ can be approximated by $\sqrt{d+1}$ (See Supplement B (Metodiev et al., 2024b)).

Theorem 3.2. *The following statements hold for the SCV:*

1. *For any choice of the shifting parameter $L, L \in \mathbb{R}$ and for all $\varepsilon > 0$*

$$1 - \varepsilon \leq \frac{SCV(d, \sqrt{d + L_d})}{\sqrt{(d+2)\pi/4}} \leq \frac{SCV(d, \sqrt{d + L})}{\sqrt{(d+2)\pi/4}} \leq 2 + \varepsilon, \quad (12)$$

for all but finitely many d . Thus choosing the radius $\sqrt{d + L}$ results in an SCV that is both asymptotically at most twice as large as the optimal SCV and is of order \sqrt{d} .

2. *The following inequality for the SCV can be given for choosing the radius $c = \sqrt{d+1}$:*

$$0.63\sqrt{(d+2)\pi/4} - 1 \leq SCV(d, \sqrt{d + L_d}) \quad (13)$$

$$\leq SCV(d, \sqrt{d+1}) \leq 2.1\sqrt{(d+2)\pi/4} - 1 \quad (14)$$

This inequality holds for all $d \geq 1$.

Remark 2. Statement 1 of Theorem 3.2 shows that $SCV(d, c)$ is increasing with order \sqrt{d} as $d \rightarrow \infty$, both in our choice $c = \sqrt{d+1}$ and the optimal choice c_d . Further, any choice of the shifting parameter L used to define the radius $\sqrt{d + L}$ is asymptotically at most twice as bad as any optimal solution in terms of the SCV. This suggests some robustness of our estimator with respect to the choice of L . For numerical results about the behaviour of the SCV for different values of L , we refer the reader to Supplement B (Metodiev et al., 2024b).

One can also calculate the bias of the THAMES on the scale of the marginal likelihood by considering a second-order Taylor approximation and using Equation (9):

$$E[\hat{Z}] = Z + Var[\hat{Z}^{-1}]/(Z^{-1})^3 = Z(1 + SCV(d, c)/T). \quad (15)$$

Considering Equation (15), the bias can be estimated by using the plug-in estimates \widehat{SCV} and \hat{Z}^{-1} . We also observe that the bias vanishes as T increases.

Remark 3. Statement 2 of Theorem 3.2 gives a very rough theoretical guarantee: For any dimension $d \geq 1$, the SCV obtained by choosing our recommendation for the radius, $\sqrt{d+1}$, and the SCV obtained by choosing the optimal radius, c_d , can be bounded by an affine transform of $\sqrt{d+2}$. However, our calculations suggest that the SCV at the point $c = \sqrt{d+1}$ has an asymptotically optimal performance (See Supplement B (Metodiev et al., 2024b)).

So far, we have given results for the idealized situation where the posterior distribution is exactly normal. We now give a result for the more common and realistic situation where the posterior distribution is only asymptotically normal.

Theorem 3.3. *Let $p_n(\theta|\mathcal{D}_n)$ be a sequence of posterior densities with data \mathcal{D}_n , posterior covariance matrix Σ_n , posterior mean m_n and an SCV denoted by SCV_n . Then, if*

$$|\Sigma_n|^{\frac{1}{2}} p_n \left(\Sigma_n^{\frac{1}{2}} \cdot \theta + m_n \mid \mathcal{D}_n \right) \xrightarrow{n \rightarrow \infty} |\Sigma|^{\frac{1}{2}} p \left(\Sigma^{\frac{1}{2}} \cdot \theta + m \mid \mathcal{D} \right) \quad (16)$$

uniformly in θ on all compact subsets of \mathbb{R}^d , it follows that

$$SCV_n(d, c) \xrightarrow{n \rightarrow \infty} SCV(d, c) \quad (17)$$

uniformly in c on all compact subsets of $(0, \infty)$. In particular, for any $b \geq c_d \geq a > 0$,

$$(c_d)_n \in \operatorname{argmin}_{c \in [a, b]} SCV_n(d, c) \quad \forall n \Rightarrow \lim_{n \rightarrow \infty} (c_d)_n = c_d. \quad (18)$$

Remark 4. We have already stated that the normal case is important because the posterior distribution is often asymptotically normal when the size of the data, n , is large. Theorem 3.3 assures us that our results still hold in this limiting case, under some assumptions.

If the convergence of the normalized posterior pdf is uniform in θ (Equation (16)), our statements about the limit behaviour of the SCV (Theorem 3.2 and Remarks 2–3) still hold approximately when n is large (Equation (17)). If additionally any optimal radius $(c_d)_n$ does not converge to zero or infinity, any result about c_d (Theorem 3.1 and Remark 1) also holds approximately when n is large (Equation (18)).

Let H_0 denote the Fisher information matrix. Reformulating Equation (16) by replacing Σ_n by $\frac{1}{n} H_0^{-1}$, to which it is asymptotically equivalent, gives a statement that has been proven under a variety of assumptions, e.g. Miller (2021, Theorem 4), except that in these results the type of convergence is usually not uniform convergence, but a weaker type of convergence, such as convergence in distribution or convergence in total variation.

Additional assumptions can be made about the pdfs of the sequence of distributions such that convergence in distribution implies uniform convergence of the pdfs. For example, if the pdfs are asymptotically equicontinuous and we have convergence in distribution, the convergence of the pdfs is uniform (Sweeting, 1996, Theorem 1).

Note that in this case there is no problem if the parameter space is constrained. Uniform convergence of the pdfs implies that A_n is a subset of the posterior support if n is large enough. There are also no assumptions about (m_n, Σ_n) , other than that they converge to the moments of the posterior limit. In this sense, the estimators $(\hat{\mu}, \hat{\Sigma})$ take the place of these constants in practice and Theorem 3.3 holds even when we use these estimators.

Remark 5. Due to the assumption of normality it is the case that when choosing the optimal radius $c_d = \sqrt{d + L_d}$, the probability of a term of the THAMES in $\theta^{(t)}$ not being set to 0 is equal to

$$\mathbb{P}(\theta^{(t)} \in A_{or}) = \mathbb{P}((\theta^{(t)} - m)^T \Sigma^{-1} (\theta^{(t)} - m) < d + L_d) = \chi^2(d + L_d; d), \quad (19)$$

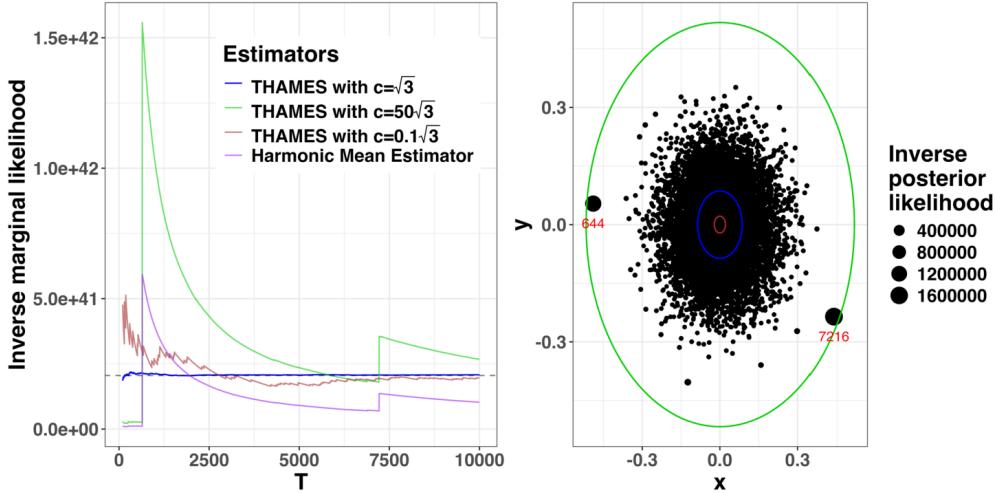


Figure 1: Left: The THAMES calculated by choosing the radii $c = \sqrt{d+1} = \sqrt{3}$, $c = 0.1\sqrt{3}$, $c = 50\sqrt{3}$ in the two-dimensional case, $d = 2$, with the true value $1/Z$ (dotted line) and the Harmonic Mean Estimator. Right: The posterior sample evaluated at the inverse of the unnormalized posterior density and the different ellipses used to define the THAMES. In this particular case the posterior covariance matrix is the scaled identity matrix, so the ellipses are spheres. The two samples occurring at points 644 and 7216 have a very low likelihood. They cause massive jumps in the Harmonic Mean Estimator when the radius of A is large and are excluded when the radius is equal to $\sqrt{d+1} = \sqrt{3}$. One can choose a smaller radius (e.g., $c = 0.1\sqrt{3}$), but then too much of the sample is excluded and convergence takes longer.

the CDF of the χ^2 -distribution with d degrees of freedom evaluated at $d + L_d$. This approaches 50% due to Theorem 3.1. Thus the algorithm sets about 50% of the highest terms in Equation (7) to 0. This means that for a large number of samples T and given the normality assumption, our algorithm is similar to the following method:

Instead of checking whether $\theta^{(t)} \in A_{or}$ directly, one can set roughly 50% of the highest terms of the THAMES, the terms not included in the Highest Posterior Density (HPD) region of size 50%, to 0.

Remark 6. It is assumed that the covariance matrix of the posterior distribution is positive definite. This assumption is necessary since otherwise a posterior density with respect to the Lebesgue-measure on \mathbb{R}^d would not exist. On the other hand this assumption is not restrictive, since the same estimation procedure can be applied to a lower dimensional subspace of \mathbb{R}^d on which a density is defined.

We can illustrate the relationship between the THAMES and the harmonic mean estimator defined by Newton and Raftery (1994) using the toy example from Figure 1. It was calculated using the same model as the one introduced in Section 4.1 with the dimensions of the parameter space $d = 2$, but by setting the data set to $\mathcal{D} \equiv 0$ to ensure stability of the estimator on the inverse likelihood scale.

The pdf of the Uniform distribution on the ellipse is essentially used as a rejection rule: Values with very low posterior density (and therefore high inverse posterior density) are rejected, while high-density values are accepted. A balance between the volume of the ellipse and the percentage of the rejected posterior sample needs to be found to ensure optimal performance. The harmonic mean estimator does not have this rejection rule, so sample points with low posterior densities can lead to massive jumps.

3.3 THAMES algorithm

Below is an algorithm for the implementation of the THAMES. Procedures for sample splitting, as well as the truncated ellipsoid correction used in the case that the parameter space is constrained have been included. These additions are described in page 11.

We recommend these additions, but we have also found that in some cases they make almost no difference. For example, sample splitting does not appear to have an impact when the dimension of the parameter space, d , is small (Section 4.1), while the truncated ellipsoid correction is negligible when the posterior mean is not close to the edge of the posterior support (Section 4.4 and Section 4.3).

Algorithm 1 \hat{Z}^{-1} calculation.

Input: Data \mathcal{D} and posterior samples $(\theta^{(i)})_{i \in [1, T]}$.

Sample splitting: Calculate the empirical mean $\hat{\theta}$ and sample covariance matrix $\hat{\Sigma}$ based on the first $T/2$ posterior samples $(\theta^{(i)})_{i \in [1, T/2]}$.

Standardization: $\tilde{\theta}^{(i)} = \hat{\Sigma}^{-1/2}(\theta^{(i)} - \hat{\theta})$ for $i \in [T/2 + 1, T]$.

Truncation subset: $\mathcal{S} = \{i : \|\tilde{\theta}^{(i)}\|_2^2 < d + 1\}$.

Calculate THAMES estimator:

$$\hat{Z}^{-1} = \frac{1}{T/2} \sum_{i=T/2+1}^T \frac{h(\theta^{(i)})}{L(\theta^{(i)})\pi(\theta^{(i)})},$$

where $h(\theta^{(i)}) = 1/V(A)$ if $i \in \mathcal{S}$ and 0 otherwise, with

$$V(A) = \sqrt{|\hat{\Sigma}|}\pi^{d/2}(d+1)^{d/2}/\Gamma(\frac{d}{2}+1) \text{ and } A = \{\theta : (\theta - \hat{\theta})^T \hat{\Sigma}^{-1}(\theta - \hat{\theta}) < d + 1\}.$$

if the posterior support $\text{supp}(\theta|\mathcal{D})$ is constrained **then**

Simulate the sample $\nu^{(1)}, \dots, \nu^{(N)}$ from the uniform distribution on A .

Approximate the volume ratio $V(A \cap \text{supp}(\theta|\mathcal{D}))/V(A)$ via the Monte Carlo estimator

$$\hat{R} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{\{\theta|\pi(\theta)L(\theta)>0\}}(\nu^{(i)}).$$

Assign $\hat{Z}^{-1} \leftarrow \hat{R}^{-1}\hat{Z}^{-1}$.

end if

Output: THAMES estimator \hat{Z}^{-1} .

Sample splitting

The theoretical guarantees established in Section 3.2 operate under the assumption of an oracle ellipsoid A_{or} . In particular, this means that the ellipsoid determining the THAMES estimator \hat{Z}^{-1} is defined independently of $(\theta^{(i)})_{i \in [1, T]}$. In practice, we find that estimating A and Z^{-1} simultaneously using the same posterior sample can induce bias in \hat{Z}^{-1} when the parameter space is high-dimensional. We therefore implement a sample splitting procedure that involves estimating A and Z^{-1} using separate posterior draws. Specifically, we first estimate the posterior mean and covariance matrix via the empirical mean $\hat{\theta}$ and sample covariance $\hat{\Sigma}$ using the first $T/2$ posterior samples $(\theta^{(i)})_{i \in [1, T/2]}$. Defining A as in Equation (5) based on $\hat{\theta}$ and $\hat{\Sigma}$, we then calculate the THAMES estimator \hat{Z}^{-1} using the last $T/2$ posterior samples $(\theta^{(i)})_{i \in [T/2+1, T]}$. The same problem was noted by Gronau et al. (2020) in their popular implementation of bridge sampling. For this reason, the bridge sampling package uses the same sample splitting procedure just described, in its default setting.

Correcting for the presence of constrained parameters

Whenever the posterior support of the parameters is not \mathbb{R}^d , for example when the parameters are variances or probabilities, it is possible that our choice of h in Equation (3), the pdf of the uniform distribution on A , is not correctly normalized. This is due to the fact that A is not necessarily a subset of the posterior support and thus h is not a pdf over this space.

In this case, the expectation of the THAMES is distorted by a multiplicative constant:

$$E_\theta[\hat{Z}^{-1}|\mathcal{D}] = E_\theta\left[\frac{h(\theta)}{L(\theta)\pi(\theta)}\middle|\mathcal{D}\right] = Z^{-1} \cdot \frac{V(A \cap \text{supp}(\theta|\mathcal{D}))}{V(A)} =: Z^{-1}R, \quad (20)$$

where $V(A \cap \text{supp}(\theta|\mathcal{D}))$ denotes the volume of the intersection between A and the posterior support. One way to deal with this problem is to transform the parameter space, e.g., by setting $\vartheta := \log(\theta)$ if θ is a variance parameter. One can then continue with marginal likelihood estimation on ϑ , using the transformed prior distribution. In this case, it is of course important to include the Jacobian of the transformation when computing the prior density. It should be noted that the default proposal used in the bridge sampling package from Gronau et al. (2020) uses this transformation, because it suffers from the same problem when the parameter space is constrained. This solution is also a viable option for the THAMES, since the transformation removes the need for any adjustments. However, this solution requires deciding on a viable transformation for each new type of constraint (e.g., simplex constraints, interval constraints, etc.) and the posterior behaviour of the transformed parameters may be hard to interpret. For this reason, we suggest a different correction.

Another way is to adjust for the bias by calculating the ratio of these volumes, R , using a simple Monte Carlo approximation: We simulate $\nu^{(1)}, \dots, \nu^{(N)} \stackrel{\text{i.i.d.}}{\sim} \text{Unif}(A)$, $N \in$

N and calculate

$$\hat{R} := \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{\text{supp}(\theta|\mathcal{D})}(\nu^{(i)}) = \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{\{\theta|\pi(\theta)L(\theta)>0\}}(\nu^{(i)}). \quad (21)$$

Given A , this is an unbiased and consistent estimator of R by the law of large numbers. The bias-adjusted THAMES is then $\hat{Z}_{adj}^{-1} = \hat{R}^{-1}\hat{Z}^{-1}$.

The problem of the parameter space being constrained is common not only for the THAMES, but for reciprocal importance sampling estimators in general. It has for example been addressed by Hajargasht and Wo'zniak (2018) and Sims et al. (2008). Hajargasht and Wo'zniak (2018) used variational Bayes techniques and showed that these ensure that the support of the chosen h is a subset of the posterior support, under mild conditions. Sims et al. (2008) used an ellipsoidal density truncated on a subset of the joint support, $\Theta_U := \{\theta|\pi(\theta)L(\theta) > U\}$, where $U > 0$. Since the support of $\pi(\theta)L(\theta)$ is equal to the posterior support, our truncation set is similar to the one chosen in Sims et al. (2008), except that we set $U = 0$.

The adjustment is usually very small. The problem arises only when the posterior mean is close enough to the edge of the parameter space. The edge of the parameter space often indicates a priori unlikely values. For this reason it is also rare that the data indicate posterior parameters being close to the edge. Thus the ratio between the volumes is close to one and the variance of \hat{R} is small. In fact, the adjustment did not have any sizeable impact on any of the examples simulated in Section 4. This may not be the case, however, if the actual data generating mechanism is very different from the model being considered. In this case, it can in practice happen that the posterior mean is indeed very close to the edge. We show one example of this in Supplement B Metodiev et al. (2024b).

In either case we have found that a small number of simulations, around $N = 100$, is usually enough. Confidence intervals obtained from the fact that \hat{R} is asymptotically normal can be used to check whether the variance of \hat{R} is large. In this case N should be increased to yield a more precise approximation. The computational cost of implementing this adjustment is typically small.

4 Examples

We now describe several simulated and real data examples to assess the THAMES estimator. In Sections 4.1, 4.2, and 4.3, three statistical models, for which exact expressions of the marginal likelihood are derived, are considered. This allows us to compare the THAMES estimated values to the exact ones for evaluation. In Section 4.4, we consider a real data example with models for which no analytical expressions for the marginal likelihood are available, to our knowledge, and where there is a need for reliable estimators. We compare our estimator to bridge sampling, which is more complicated than THAMES but is known to have performed well (Meng and Wong, 1996; Gronau et al., 2020).

4.1 Multivariate Gaussian data

We first consider the case where data $Y_i, i = 1, \dots, n$ are drawn independently from a multivariate normal distribution:

$$Y_i | \mu \stackrel{\text{iid}}{\sim} \text{MVN}_d(\mu, I_d), \quad i = 1, \dots, n,$$

along with a prior distribution on the mean vector μ :

$$p(\mu) = \text{MVN}_d(\mu; 0_d, s_0 I_d),$$

with $s_0 > 0$. As shown in Supplement A (Metodiev et al., 2024a), the posterior distribution of the mean vector μ given the data $\mathcal{D} = \{y_1, \dots, y_n\}$ is given by:

$$p(\mu | \mathcal{D}) = \text{MVN}_d(\mu; m_n, s_n I_d), \quad (22)$$

where $m_n = n\bar{y}/(n + 1/s_0)$, $\bar{y} = (1/n) \sum_{i=1}^n y_i$, and $s_n = 1/(n + 1/s_0)$.

Interestingly, while the observations $(Y_i)_i$ are independent given the vector μ , they are not independent marginally, and the marginal likelihood does not take the form of a product over marginal terms in i . Conversely, thanks to the isotropic Gaussian prior distribution which is considered for μ , where the $(\mu_j)_j$ are all iid, not only are the vectors $(Y_{.j})_j$ independent given μ , they are also independent marginally. From this property, we prove in Proposition 2 of Supplement A (Metodiev et al., 2024a) that the marginal likelihood of the model can be written analytically as

$$p(\mathcal{D}) = \prod_{j=1}^d \text{MVN}_n(y_{.j}; 0_n, s_0 1_n 1_n^\top + I_n), \quad (23)$$

where $y_{.j} \in \mathbb{R}^n$ is the vector of all observations for variable j such that $[y_{.j}]_i = y_{ij}$ and 1_n is the vector of 1 in \mathbb{R}^n .

Assessing the precision of the THAMES estimator as a function of T

We first considered the univariate case $d = 1$. We simulated a unique sample of size $n = 20$ with $\mu = 2$ and we set $s_0 = 1$, for illustration; other choices for s_0 led to similar conclusions regarding the quality of the estimation. Figure 2 shows the THAMES estimated values for the log marginal likelihood, for $T = 5, 1005, 2005, \dots, 9005$ samples of the posterior distribution (Equation (22)). Confidence intervals as well as the exact value of the log marginal likelihood computed using Equation (23) are also reported. It can be seen that the estimate converges to the correct value and that the confidence intervals contain the true value in all cases, even for $T = 5$ only.

Assessing the precision of the THAMES estimator as a function of d

For this second set of experiments, we considered different values of d , and aimed at testing the robustness of the THAMES approach on multiple data sets, with increasing

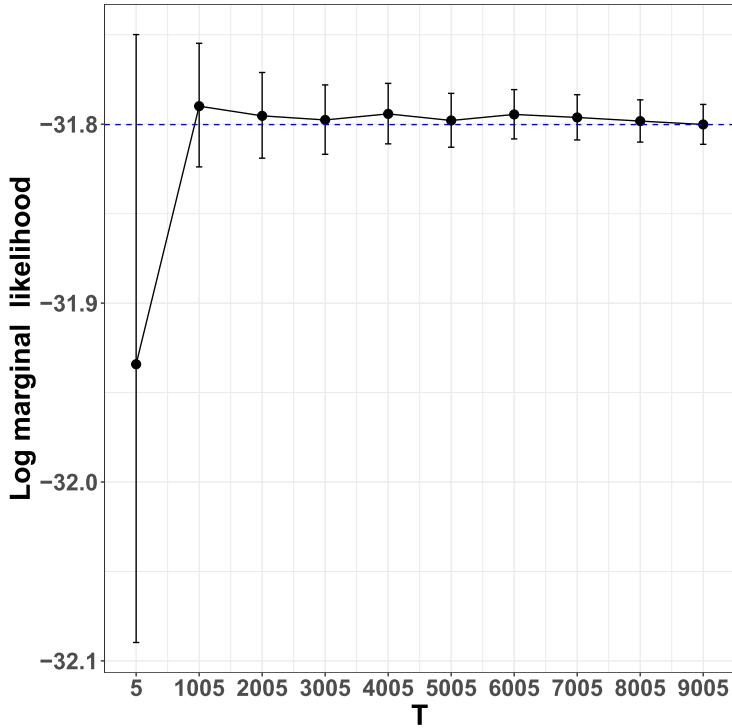


Figure 2: Estimation of the log marginal likelihood using THAMES for a unique univariate Gaussian sample with $n = 20$ as a function of T , the number of (cumulative) samples from the posterior distribution. The black dots indicate the values of the THAMES estimator of the log marginal likelihood. The vertical lines represent 95% confidence intervals, and the dashed blue line represents the exact value computed using Equation (23).

dimensionality. Thus, for each d , we generated 50 different data sets of size $n = 20$ using the multivariate Gaussian model. In practice, we set the true value of μ to 2, for all its components. Again, the prior parameter s_0 was set to 1 and similar conclusions were drawn for other values. Moreover, the value of T was set to 10,000 for all the experiments.

We also used this example to assess the sample splitting procedure for the posterior samples, as proposed in Section 3.3. The results are given in Figure 3. In the figure on the left, where *no* sample splitting of the posterior samples is used to compute THAMES, we observe that a bias appears as the dimensionality of the model considered increases, and the log marginal likelihood tends to be slightly underestimated. As illustrated by the figure on the right hand side, this bias is primarily related to the estimation of the posterior covariance matrix, and not to the THAMES estimation itself. Indeed, focusing on this figure on the right, we note that if the exact expression of the posterior covariance matrix given in Equation (22) is used to compute THAMES, then while the variance of the estimator increases with d , we do not observe any bias. Crucially, if the

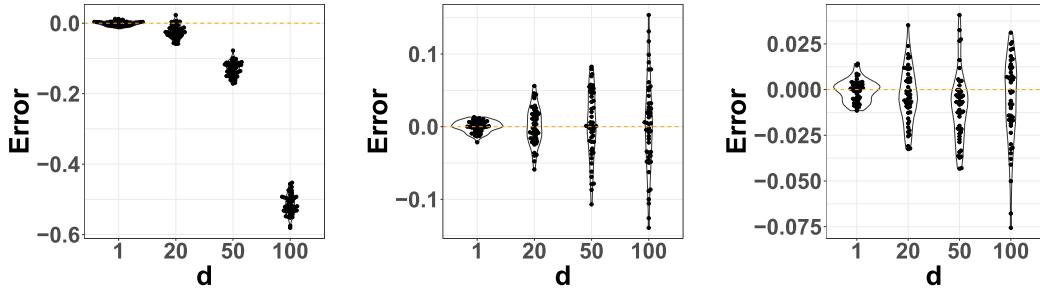


Figure 3: Difference between the estimated log marginal likelihood using THAMES and the true log marginal likelihood for a multivariate Gaussian model with $n = 20$ and $T = 10000$. The procedure is repeated on 50 different data sets for each d : Left: the THAMES approach *with no* sample splitting of the posterior samples. Middle: the THAMES approach *with* sample splitting of the posterior samples. Right: the results provided correspond to the case where the exact expression of the posterior covariance matrix given in Equation (22) is used to compute the THAMES.

sample splitting of the posterior samples is employed to compute THAMES, then again, we do not observe any bias.

Overall, we found that the sample splitting procedure of the posterior samples was not necessary to compute THAMES for low values of d . The estimated values are close to the exact ones. However, for large values of d , we recommend using the sample splitting procedure to remove the bias.

4.2 Bayesian regression

We consider a data set $(x_i, Y_i), i = 1, \dots, n$ to train a linear regression model of the form

$$Y_i|x_i, \beta, \sigma^2 \sim \mathcal{N}(x_i^\top \beta, \sigma^2), i = 1, \dots, n.$$

In this section, the goal is to assess the quality of our proposed estimator. As such, we choose a prior on (β, σ^2) for which an exact expression for the marginal likelihood, Z , exists. We compare our estimator, the THAMES, to the bridge sampling estimator implemented in Gronau et al. (2020) and a simple Monte Carlo (MC) estimator, calculated by averaging the likelihood for parameter values simulated from the prior.

Denoting $Y \in \mathbb{R}^n$, the vector of target variables Y_i , and $X \in \mathcal{M}_{n \times (d-1)}(\mathbb{R})$ the design matrix where the input vectors $x_i \in \mathbb{R}^{d-1}$ are stacked as row vectors, the linear regression model becomes:

$$Y|X, \beta, \sigma^2 \sim \text{MVN}_n(X\beta, \sigma^2 I_n).$$

We rely on a centered isotropic Gaussian prior distribution for the regression vector β

and the variance σ^2 :

$$p(\beta|\sigma^2) = \text{MVN}_{d-1}(\beta; 0_{d-1}, g\sigma^2(X^T X)^{-1}), \quad p(\sigma^2) = \text{InvGamma}\left(\sigma^2; \frac{1}{2}\nu_0, \frac{1}{2}\sigma_0^2\nu_0\right),$$

with $g, \sigma_0^2, \nu_0 > 0$. Introduced by Zellner (1971, 1986), this framework offers a conjugate prior with the attractive property of scale-invariance with respect to the regressor (Hoff, 2009). Then the posterior distribution of (β, σ^2) , given the training data set $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, is tractable:

$$\begin{aligned} p(\beta|\sigma^2, \mathcal{D}) &= \text{MVN}_{d-1}\left(\beta; \frac{g}{g+1}m_n, \frac{g}{g+1}\sigma^2(X^T X)^{-1}\right), \\ p(\sigma^2|\mathcal{D}) &= \text{InvGamma}\left(\sigma^2; \frac{1}{2}(\nu_0 + n), \frac{1}{2}(\nu_0\sigma_0^2 + s_n)\right), \end{aligned}$$

with $s_n = \mathbf{y}^T \mathbf{y} - \frac{g}{g+1} \mathbf{y}^T X (X^T X)^{-1} X^T \mathbf{y}$ and $m_n = (X^T X)^{-1} X^T \mathbf{y}$, where $\mathbf{y} \in \mathbb{R}^n$ is the observed vector of target variables associated with Y . Moreover, the marginal likelihood also has an analytical expression:

$$p(\mathbf{y}|X) = \frac{(g+1)^{-(d-1)/2}}{\pi^{n/2}} \cdot \frac{\Gamma(\frac{1}{2}(\nu_0 + n))}{\Gamma(\frac{1}{2}\nu_0)} \cdot \left(\frac{\nu_0\sigma_0^2}{\nu_0\sigma_0^2 + s_n}\right)^{\nu_0/2}.$$

Proofs for the exact expressions of the posterior and the marginal are given in Hoff (2009, Chapter 9). The data for this example are described by Hastie et al. (2009) and come from a study by Stamey et al. (1989). They examined the correlation between the level of prostate-specific antigen (lpsa) and eight clinical measures in men who were about to receive a radical prostatectomy. The variables are log cancer volume (lcavol), log prostate weight (lweight), age, log of the amount of benign prostatic hyperplasia (lbph), seminal vesicle invasion (svi), log of capsular penetration (lcp), Gleason score (gleason), and percent of Gleason scores 4 or 5 (pgg45). The target variable is the level of prostate-specific antigen (lpsa).

The choice of the hyperparameter g is a topic of much discussion (Fernández et al., 2001; Porwal and Raftery, 2022). In Porwal and Raftery (2022), $g = \sqrt{n}$ showed good performance when compared to a variety of different options, albeit in a slightly different setting where the prior on σ^2 is improper. For this reason, we chose $g = \sqrt{n}$. We chose $(\nu_0, \sigma_0^2) = (4, 1)$ for the other hyperparameters, but other choices for (g, ν_0, σ_0^2) led to similar conclusions regarding the quality of the estimation.

Seven different regression models M_2, M_3, \dots, M_8 , each with a different number of selected variables, ranging from 2 to 8, are considered for illustration. The variables are added in the order given above. Thus, M_2 includes the predictor variables lcavol and lweight, while Model M_3 considers the variables lcavol, lweight, as well as age for prediction. Finally, model M_8 takes all 8 input variables into account. Figure 4 shows the estimators of the log marginal likelihood for different number of samples from the posterior distribution in (β, σ^2) , for the different models, as well as the approximate confidence intervals.

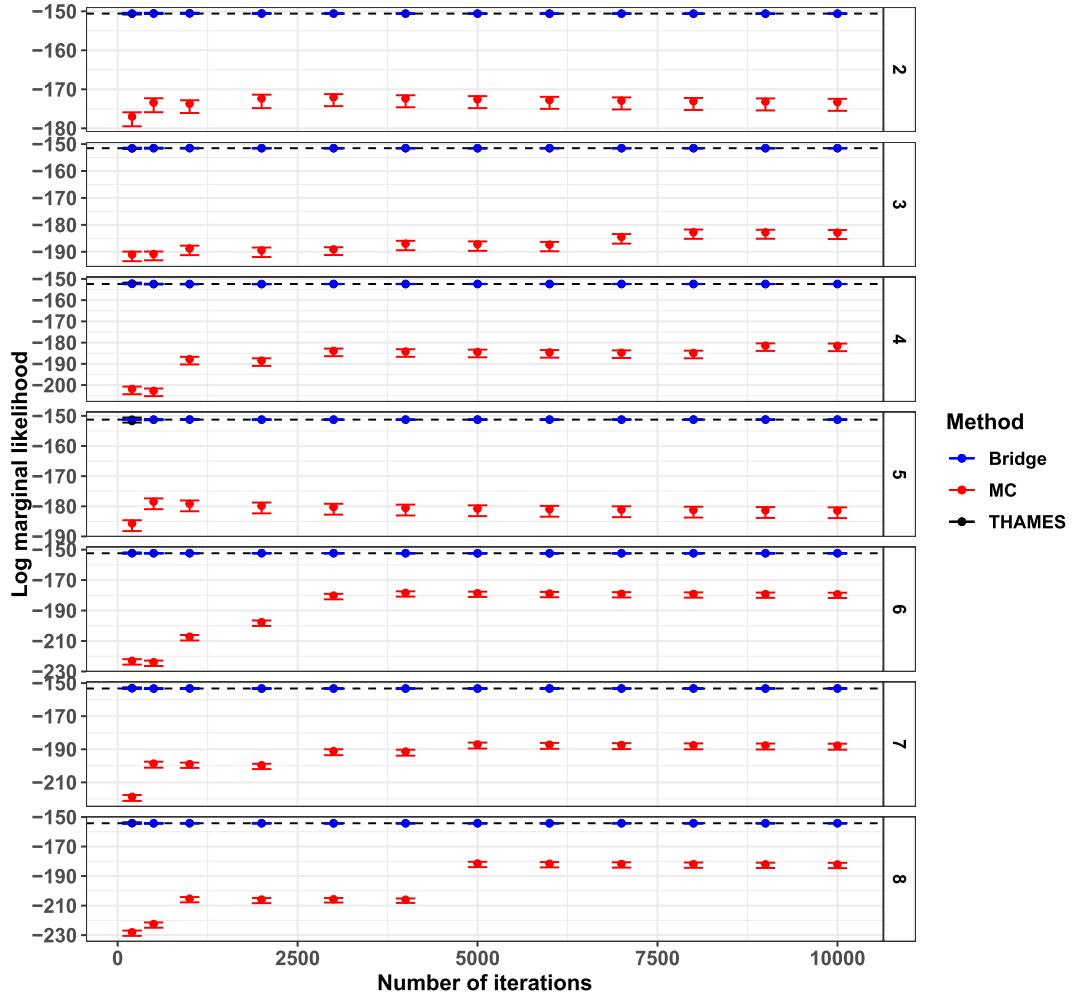


Figure 4: Log marginal likelihood (dotted line) and its estimators (dots) for the prostate data set. The approximate confidence intervals of the estimators are also indicated. Bridge sampling and THAMES are on point, while the simple Monte Carlo estimator does not seem to have converged.

Sample splitting was used and there was no noticeable bias in the results, even though we did not correct the THAMES for the bounded parameter σ^2 due to the fact that the posterior mode of this parameter is far away from 0. We also calculated the bias correction from Remark 2 which had no impact numerically, even for a posterior sample size as small as $T = 50$.

While the simple Monte Carlo estimator did not converge, the bridge sampling estimator and the THAMES behaved very similarly. Indeed, it can be seen that both these estimators converged rapidly to the correct value and that the intervals covered the correct values in most cases, even when the number of samples used was small, for

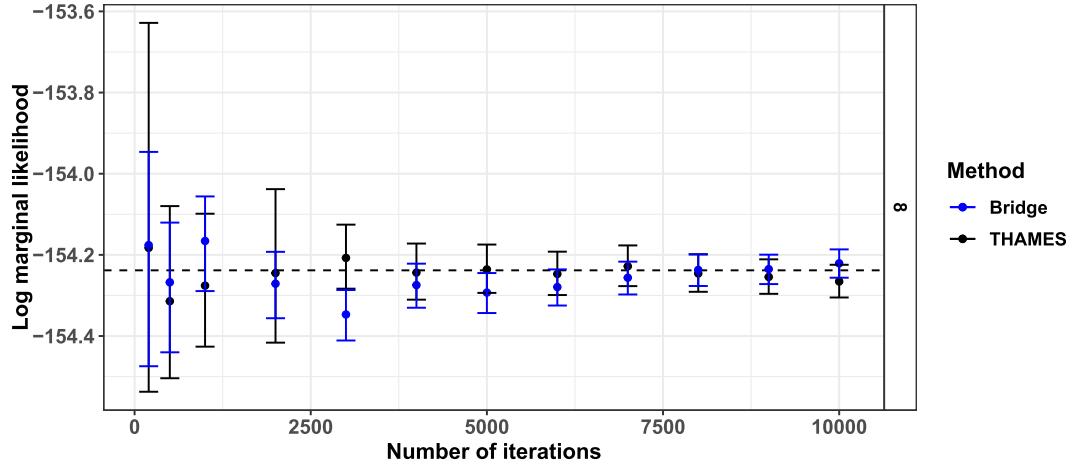


Figure 5: Log marginal likelihood (dotted line) and its estimators (dots) for the prostate data set with the full model (i.e., all eight clinical measures are selected). The approximate confidence intervals of the estimators are also indicated. The confidence intervals of the THAMES are more conservative, while the ones obtained from bridge sampling are more narrow.

all models investigated. Figure 5 shows the estimates produced by these methods when zoomed in on a finer scale in the full model, meaning that the number of clinical measures is equal to 8. Notably, the confidence intervals obtained by the THAMES were more conservative and wider than those obtained for the bridge sampling estimator, while the estimators themselves converged in a similar speed and manner.

For all models, those two estimators are particularly precise for 1000 samples of the posterior only. While the main goal of this section is to illustrate the precision of our estimation strategy for a series of models, we can also report that the model with the highest marginal likelihood, among those considered for this data set, is Model M_2 . In other words, the variables lcavol and lweight are seen as key for the prediction of the level of prostate-specific antigen.

4.3 Dirichlet-multinomial model

Extensions of the Dirichlet-multinomial model are widely used in the context of topic modelling, see, e.g., Blei et al. (2003). The expression for the marginal likelihood in this model is known, as in the previous two sections. This allows us to assess the performance of our estimator in another simulation study, in a non-Gaussian context.

A simulation study in this setting is useful for two reasons: First, this is a high-dimensional setting in which the posterior distribution of the parameters is highly non-Gaussian. In fact, the parameter space is bounded. This allows us to assess how well the THAMES performs in a very different setting, and also how much of an impact the correction for a bounded parameter space from Section 3.3 has. We check this

numerically in Supplement B (Metodiev et al., 2024b).

Second, there do exist similar models to this one for which the marginal likelihood is not tractable, e.g. Blei and Lafferty (2007). These models are therefore a possible application of the THAMES. The simulation study might give an idea of how well the THAMES would perform in these applications.

The Dirichlet-multinomial model is defined as follows: Each data point $Y_i \in \{0, \dots, l\}^K$ is drawn from a multinomial distribution given a Dirichlet-distributed random variable μ :

$$\mu \sim \text{Dirichlet}(\mu; (a_0, \dots, a_0)), \quad Y_i | \mu \stackrel{\text{i.i.d.}}{\sim} \mathcal{M}(l, \mu), \quad \forall i = 1, \dots, n.$$

Here, μ is positive and K -dimensional with components summing to 1. The covariance matrix of μ is thus necessarily singular. As noted in Remark 6, the THAMES needs to be used on posterior simulations from the subspace of \mathbb{R}^K on which a density is defined. In this case, this is $\mathbb{R}^{K-1} =: \mathbb{R}^d$. The prior density is thus

$$\pi(\mu_1, \dots, \mu_d) = \text{Dirichlet} \left(\mu_1, \dots, \mu_d, 1 - \sum_{j=1}^d \mu_j; (a_0, \dots, a_0) \right).$$

The posterior support is $\{\mu \in \mathbb{R}^d \mid \sum_{j=1}^d \mu_j \leq 1, \mu_1, \dots, \mu_d > 0\}$. The posterior distribution given the data $\mathcal{D} = \{y_1, \dots, y_n\}$ is tractable:

$$p(\mu_1, \dots, \mu_d | \mathcal{D}) = \text{Dirichlet} \left(\mu_1, \dots, \mu_d, 1 - \sum_{j=1}^d \mu_j; \alpha_1, \dots, \alpha_K \right),$$

with $\alpha_j = a_0 + \sum_{i=1}^n y_{ij}$. The marginal likelihood is thus also tractable, using Bayes' theorem.

Results

The marginal likelihood was estimated in the setting $(n, l, T, a_0) = (400, 150, 10000, 1)$ with d varying between 1, 20, 50 and 100. The quantities n and l were intentionally chosen to be large, since this model has very high-dimensional applications. For example, Blei et al. (2003) used a data set with 8000 documents, $n = 15,818$ words and used up to $K = 200$ different topics.

As mentioned, an alternative to the correction proposed in Section 3.3 is to reparameterize μ such that the support of the parameter is unconstrained. We did this by setting

$$(\mu_1^{(t)}, \dots, \mu_d^{(t)}) =: \frac{(\exp(\vartheta_1^{(t)}), \dots, \exp(\vartheta_d^{(t)}))}{\exp(-\sum_{k=1}^d \vartheta_k^{(t)}) + \sum_{k=1}^d \exp(\vartheta_k^{(t)})} = \text{softmax}(\vartheta^{(t)}), \quad t = 1, \dots, T.$$

We are using a bijective version of the softmax function (we take the first d elements of the version of the softmax which maps a sample from the Dirichlet to a parameter that

	d=1			d=20		
	MAE	SD	Time	MAE	SD	Time
Bridge	0.0001	0.0002	6.0026	0.0019	0.0024	36.9822
	MC	0.093	0.1145	0.0020	6903.4	942.444
	THAMES	0.0064	0.0087	0.0158	0.0197	0.025
d=50			d=100			
Bridge	0.0037	0.0046	58.9884	0.0086	0.0108	116.5902
	MC	13879.1	1231.7635	0.0004	18418.1	844.7528
	THAMES	0.0315	0.039	0.3094	0.0473	0.0617
THAMES			THAMES			
THAMES			THAMES			

Table 1: Average CPU times (in seconds per 10,000 posterior draws) as well as mean absolute errors and standard deviations for bridge sampling, the THAMES and the naive Monte Carlo (MC) estimator (errors of the latter were rounded to 1 decimal place). Estimates using MC are quickest to compute, but also the least precise. The THAMES is much faster than bridge sampling, although point estimates from the latter are more accurate.

sums to 0), and the induced prior $\pi_2(\vartheta) := \pi(\text{softmax}(\vartheta))|\text{softmax}_{\text{Jacobian}}(\vartheta)|$. We stress that this procedure is not necessary to calculate the THAMES, since the THAMES can be calculated on any parameter space. It is however necessary to calculate the bridge sampling estimator implemented in Gronau et al. (2020).

Table 1 shows the results when calculating the THAMES and the bridge sampling estimator on $(\vartheta^{(1)}, \dots, \vartheta^{(T)})$.² A fixed parameter μ was set to $\mu = (1/K, \dots, 1/K)$ and 50 different samples were generated using the parameters l and μ . The MC estimator was also computed for comparison.

Both bridge sampling and the THAMES outperformed the MC estimator. Additionally, while the bridge sampling estimator performed better in terms of mean absolute error, it should be noted that the THAMES is not only easier, but quicker to compute, with their differences in computation time growing as the dimension of the parameter space increases. This is likely due to the fact that the THAMES does not require additional evaluations of the likelihood, beyond the precomputed likelihood values of the posterior sample, so its computation time grows much more slowly with increasing d . Meanwhile bridge sampling does require additional evaluations, which take up an increasing amount of computation time. The average computation time for the bridge sampling estimator is about 361 times as high for $d = 1$, and 118 times as high for $d = 100$. However, we would like to emphasize that, in our opinion, the real strength of our estimator lies in the fact that it is not only quick, but also easy to implement.

²Computations were performed on an Intel(R) Core(TM) i7-7700HQ CPU at 2.80GHz with 16 GB RAM.

4.4 Mixed effects model

Netherlands schools data

To demonstrate the performance of THAMES on a random effects model, we consider the Netherlands (NL) schools dataset of Snijders and Bosker (1999). For our purposes, the data consist of language test scores of 2,287 eighth-grade pupils from 133 classes (in 131 schools) in the Netherlands. We denote by $y_{ij} \in \mathbb{R}$ the language test score of pupil i in class j , where $j \in \{1, \dots, J\}$ with $J = 133$ and $i \in \{1, \dots, n_j\}$ with n_j the size of class j . Let $n = \sum_{j=1}^J n_j = 2,287$ denote the full sample size.

We aim to determine if there is clustering of language test scores by class, with some classes performing significantly better than others on average. To do this, we fit both a simple mean model (which treats test scores of students in the same class as independent) and a random intercept model (which accounts for correlation of test scores within each class) to the data. The former (null) model H_0 posits that all classes perform the same, on average, while the latter (alternative) model H_1 allows for variation in performance at the class level. We estimate the log marginal likelihoods for the two models, Z_0 and Z_1 , respectively, using the THAMES. For comparison, we also compute estimates using bridge sampling (Gronau et al., 2020) and a simple Monte Carlo (MC) estimator that averages the likelihood against draws from the prior. With estimates of $\log(Z_0)$ and $\log(Z_1)$, we estimate the Bayes factor B_{01} to conduct a Bayesian hypothesis test of H_0 versus H_1 . Note that posterior simulation and marginal likelihood calculation are not analytically tractable for this model. As such, the use of approximate posterior sampling (e.g., via MCMC) and marginal likelihood estimation (e.g., via the THAMES) is required.

Linear model (LM)

We first consider a simple mean model (denoted LM), which posits that

$$\begin{aligned} y_{ij} &= \mu + \varepsilon_{ij}, \quad j \in \{1, \dots, J\}, i \in \{1, \dots, n_j\}, \sum_j n_j = n, \\ \varepsilon_{ij} &\stackrel{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2), \\ \mu &\sim N(\hat{\mu}, \hat{\sigma}_\mu^2), \\ \sigma_\varepsilon^2 &\sim \text{InverseGamma}(\hat{\nu}_\varepsilon, \hat{\beta}_\varepsilon). \end{aligned}$$

The fixed hyperparameters $\hat{\mu}, \hat{\sigma}_\mu^2, \hat{\nu}_\varepsilon, \hat{\beta}_\varepsilon$ are specified so as to ensure that the prior distribution is dispersed relative to the likelihood, but on the same scale, as

$$\begin{aligned} \hat{\mu} &= \text{mean}(y_{ij}) = 40.93, \quad \hat{\sigma}_\varepsilon^2 = \sqrt{2} \cdot \text{sd}(y_{ij}) = 12.73, \\ \hat{\nu}_\varepsilon &= 0.5, \quad \hat{\beta}_\varepsilon = 0.5 \cdot \text{var}(y_{ij}) = 40.53. \end{aligned}$$

The hyperparameters $(\hat{\nu}_\varepsilon, \hat{\beta}_\varepsilon)$ are chosen so that the prior mean of the precision $1/\sigma_\varepsilon^2$ equals $1/\text{var}(y_{ij})$. The set of parameters to be estimated in this model $(\mu, \sigma_\varepsilon^2)$ has dimension $d = 2$. As we are not using a conjugate prior for the linear model, the marginal likelihood does not admit an analytic expression in this case.

Full linear mixed model (full LMM)

We consider the random intercept model (denoted full LMM):

$$\begin{aligned} y_{ij} &= \mu + \alpha_j + \varepsilon_{ij}, \\ \varepsilon_{ij} &\stackrel{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2), \\ \alpha_j &\stackrel{\text{iid}}{\sim} N(0, \sigma_\alpha^2), \\ \mu &\sim N(\hat{\mu}, \hat{\sigma}_\mu^2), \\ \sigma_\varepsilon^2 &\sim \text{InverseGamma}(\hat{\nu}_\varepsilon, \hat{\beta}_\varepsilon), \\ \sigma_\alpha^2 &\sim \text{InverseGamma}(\hat{\nu}_\alpha, \hat{\beta}_\alpha). \end{aligned}$$

Here $(\hat{\mu}, \hat{\sigma}_\mu^2, \hat{\nu}_\varepsilon, \hat{\beta}_\varepsilon)$ are as above and we specify $\hat{\nu}_\alpha = 0.5, \hat{\beta}_\alpha = 0.5 \cdot \text{var}(\hat{\mu}_j) = 13.77$, where $\hat{\mu}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} y_{ij}$ is the sample mean for class $j \in \{1, \dots, J\}$. The hyperparameters $(\hat{\nu}_\alpha, \hat{\beta}_\alpha)$ are chosen so that the prior mean of the precision $1/\sigma_\alpha^2$ equals $1/\text{var}(\hat{\mu}_j)$. The set of parameters to be estimated in this model $(\mu, \sigma_\varepsilon^2, \sigma_\alpha^2, \alpha)$ has dimension $d = 136$.

Reduced linear mixed model (reduced LMM)

Note that the intercept parameters of the full LMM are not identifiable, as there is give-and-take between estimating the grand mean μ and the random intercepts α_j . By absorbing α_j into the error term structure ε_{ij} , we can specify an equivalent model (having the same marginal likelihood) with $d = 3$ identifiable parameters $(\mu, \sigma_\varepsilon^2, \sigma_\alpha^2)$. This amounts to marginalizing the α_j 's out of the model. The model (which we call reduced LMM) is given by

$$\begin{aligned} y_{ij} &= \mu + \varepsilon_{ij}, \\ \varepsilon_{ij} &\sim N(0, \sigma_\varepsilon^2 + \sigma_\alpha^2), \\ \text{Cov}(\varepsilon_{ij}, \varepsilon_{i'j'}) &= \sigma_\alpha^2, \quad i, i' \in \{1, \dots, n_j\}, i \neq i', \\ \text{Cov}(\varepsilon_{ij}, \varepsilon_{i'j'}) &= 0, \quad j \neq j', \\ \mu &\sim N(\hat{\mu}, \hat{\sigma}_\mu^2), \\ \sigma_\varepsilon^2 &\sim \text{InverseGamma}(\hat{\nu}_\varepsilon, \hat{\beta}_\varepsilon), \\ \sigma_\alpha^2 &\sim \text{InverseGamma}(\hat{\nu}_\alpha, \hat{\beta}_\alpha). \end{aligned}$$

Here $(\hat{\mu}, \hat{\sigma}_\mu^2, \hat{\nu}_\varepsilon, \hat{\beta}_\varepsilon, \hat{\nu}_\alpha, \hat{\beta}_\alpha)$ are as above.

Results

Figure 6 shows the log marginal likelihood of the NL schools data for each model computed using the THAMES, bridge sampling, and simple Monte Carlo estimators with approximate 95% confidence intervals as a function of the number of posterior MCMC or prior MC draws. Bridge sampling is a popular state-of-the-art method to

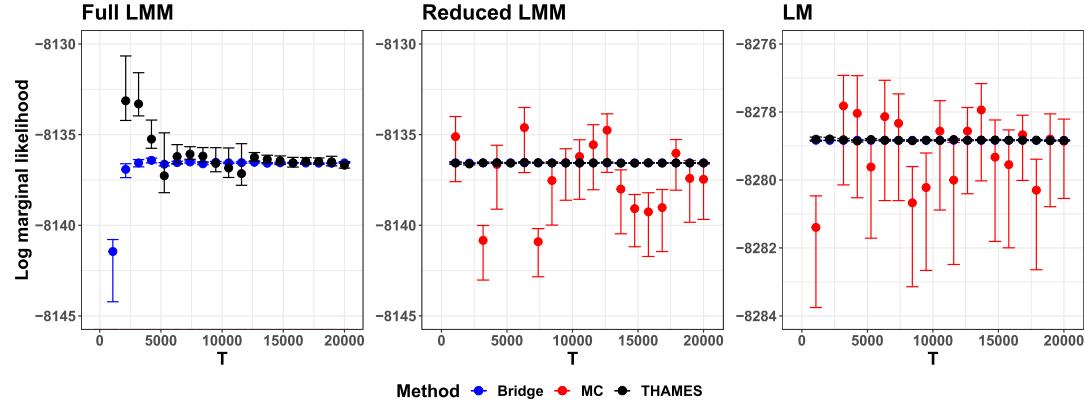


Figure 6: Log marginal likelihood estimates for models fitted to the NL schools data.

estimate log marginal likelihoods from posterior MCMC samples, which is substantially more complicated computationally than the THAMES. Posterior MCMC sampling is carried out in R using Stan (R Core Team, 2023; Stan Development Team, 2022). We use values of the sample size T evenly spaced between 1,000 and 20,000. For each T , we run 4 chains in parallel for $T/2$ iterations and remove the first $T/4$ as burn-in, yielding $T/4$ MCMC samples from each of the 4 chains, which are used to compute the THAMES and bridge sampling estimates.

THAMES provides consistent estimates of the log marginal likelihood with greater precision as the posterior sample size grows. As we would expect, THAMES converges much faster for the LM (with $d = 2$) and the reduced LMM (with $d = 3$) than for the full LMM (with $d = 136$), although the estimates of the reduced and full LMM converge to the same value. While the posterior support of this model is constrained due to the variance parameters ($\sigma_\varepsilon^2, \sigma_\alpha^2$), we found that the truncation correction defined in Section 3.3 had no impact on the results. For a given posterior sample size, we find that bridge sampling generally produces more precise estimates than the THAMES. However, the THAMES has the advantage of being much simpler to implement and more computationally efficient in practice. On average over the samples in Figure 6, bridge sampling required 6.4 times as much computation time as the THAMES for the full LMM, 556.5 times as much for the reduced LMM, and 26.8 times as much for the LM when estimated from the same number of posterior draws, as reported in Table 2.³ The MC estimator, while fast and theoretically unbiased and consistent, suffers from substantial variance. In the left panel of Figure 6, the MC estimates are not shown as they lie outside the range of the plot.

Using the THAMES estimates of the log marginal likelihoods for the LM ($\log(Z_0)$) and the (reduced) LMM ($\log(Z_1)$) with 20,000 posterior draws, the log Bayes factor ($\log(B_{01})$) is estimated as

$$\log(B_{01}) = \log(Z_0) - \log(Z_1) = -8278.842 + 8136.561 = -142.281,$$

³Computations were performed on an Apple M1 chip with 3.20GHz processor and 16 GB RAM.

	Full LMM	Reduced LMM	LM
Bridge	0.3815	1.6482	0.0723
MC	0.0002	0.0002	0.0002
THAMES	0.0581	0.0030	0.0027

Table 2: Average CPU times (in seconds per 1,000 posterior draws) to produce the estimates in Figure 6. The THAMES is faster than bridge sampling. Both take more time for the same number of iterations than the naive Monte Carlo (MC) estimator in terms of CPU time, even though the latter is far less precise (see Figure 6).

indicating decisive evidence in favor of the random intercept model (Kass and Raftery, 1995).

5 Discussion

We have proposed an estimator of the reciprocal of the marginal likelihood, called the THAMES, which is simple to compute, unbiased, consistent, has finite variance and is asymptotically normal, with available confidence intervals. It is a version of reciprocal importance sampling. The estimator has one user-specified control parameter, and we have derived an optimal value for this in the situation where the posterior distribution is normal, which is of great interest because posterior distributions are asymptotically normal in many situations. We have carried out several numerical experiments in which the estimator performs well.

A similar proposal was made independently in McEwen et al. (2022) under the name “Learned harmonic mean estimator”, where a variety of different sample models were suggested to work in conjunction. One of these models, the “Hypersphere”, corresponds to the THAMES, the difference being that no theoretical results were given for the optimal control parameter, c . Instead, c was optimized numerically as the minimum of the second harmonic moment, via the Brent hybrid root-finding algorithm. In the only high-dimensional application, which was in fact a Gaussian posterior, c was not optimized and it was noted that “alternative more effective target models can be developed that better scale to higher-dimensional settings”. We believe that with the THAMES, using the suggested optimal controlling parameter, this is the case.

The THAMES relies on estimating the posterior covariance matrix and mean. In our experience it is important that the estimator chosen for the covariance matrix be accurate for estimating each matrix entry. Elementwise accuracy appears to be important because the covariance matrix is used to precisely define a quadratic inequality. For example, using a shrinkage estimator for the covariance matrix, which can produce large errors in a small proportion of its elements, has in our experience degraded the performance of the THAMES in some situations.

One possible alternative to covariance matrix estimation would be to select a minimum-volume covering ellipse which includes a certain percentage of those points of the posterior sample which have the largest value with respect to the (unnormalized) posterior density evaluated at those points. This would ensure that an HPD-region is

well approximated, independent of the underlying posterior distribution. Determining a minimum-volume covering ellipse given a set of points can be difficult computationally, but this problem has been addressed in the literature in different settings and could possibly be adapted to the THAMES.

Acknowledgments

The authors would like to thank the anonymous referees, an Associate Editor and the Editor for their constructive comments. The authors would further like to thank Christoph Richard from the Friedrich-Alexander-Universität Erlangen-Nürnberg for his helpful comments. Their comments dramatically improved the content and quality of this paper.

Funding

Iron's research was supported by a Shanahan Endowment Fellowship and a Eunice Kennedy Shriver National Institute of Child Health and Human Development training grant, T32 HD101442-01, to the Center for Studies in Demography & Ecology at the University of Washington. Raftery's research was supported by NIH grant R01 HD070936 from the Eunice Kennedy Shriver National Institute of Child Health and Human Development (NICHD), by the Fondation des Sciences Mathématiques de Paris (FSMP), and by Université Paris-Cité.

Supplementary Material

Supplement A: Proofs and Calculations (DOI: [10.1214/24-BA1422SUPPA](https://doi.org/10.1214/24-BA1422SUPPA); .pdf). We prove the analytic results from Section 3 and derive the exact expression of the posterior and marginal density used for the multinomial likelihood in Section 4.

Supplement B: Additional Simulations (DOI: [10.1214/24-BA1422SUPPB](https://doi.org/10.1214/24-BA1422SUPPB); .pdf). We give some additional, numeric results about the approximate behaviour of the THAMES in the normal case as well as the case where the posterior support is constrained.

References

- Blei, D. M. and Lafferty, J. D. (2007). "A correlated topic model of Science." *Annals of Applied Statistics*, 1(1): 17–35. [MR2393839](#). doi: <https://doi.org/10.1214/07-AOAS114>. 19
- Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). "Latent Dirichlet Allocation." *Journal of Machine Learning Research*, 3: 993–1022. 18, 19
- Chib, S. (1995). "Marginal likelihood from the Gibbs output." *Journal of the American Statistical Association*, 90: 1313–1321. [MR1379473](#). 2
- DiCiccio, T. J., Kass, R. E., Raftery, A. E., and Wasserman, L. (1997). "Computing Bayes factors by combining simulation and asymptotic approximations." *Journal of the American Statistical Association*, 92: 903–915. [MR1482122](#). doi: <https://doi.org/10.2307/2965554>. 3, 4

- Durmus, A., Moulines, E., and Pereyra, M. (2018). “Efficient Bayesian computation by proximal Markov chain Monte Carlo: when Langevin meets Moreau.” *SIAM Journal on Imaging Sciences*, 11: 473–506. [MR3763089](#). doi: <https://doi.org/10.1137/16M1108340>. 4
- Fernández, C., Ley, E., and Steel, M. F. (2001). “Benchmark priors for Bayesian model averaging.” *Journal of Econometrics*, 100(2): 381–427. URL <https://www.sciencedirect.com/science/article/pii/S0304407600000762> [MR1820410](#). doi: [https://doi.org/10.1016/S0304-4076\(00\)00076-2](https://doi.org/10.1016/S0304-4076(00)00076-2). 16
- Frühwirth-Schnatter, S. (2004). “Estimating marginal likelihoods for mixture and Markov switching models using bridge sampling techniques.” *Econometrics Journal*, 7: 143–167. [MR2076630](#). doi: <https://doi.org/10.1111/j.1368-423X.2004.00125.x>. 5
- Gelfand, A. E. and Dey, D. K. (1994). “Bayesian model choice: asymptotics and exact calculations.” *Journal of the Royal Statistical Society: Series B (Methodological)*, 56: 501–514. [MR1278223](#). 2
- Ghosal, S. (2000). “Asymptotic normality of posterior distributions for exponential families when the number of parameters tends to infinity.” *Journal of Multivariate Analysis*, 74: 49–68. [MR1790613](#). doi: <https://doi.org/10.1006/jmva.1999.1874>. 6
- Gronau, Q. F., Singmann, H., and Wagenmakers, E.-J. (2020). “bridgesampling: An R Package for Estimating Normalizing Constants.” *Journal of Statistical Software*, 92(10): 1–29. [11](#), [12](#), [15](#), [20](#), [21](#)
- Hajargasht, G. and Wo’zniak, T. (2018). “Accurate Computation of Marginal Data Densities Using Variational Bayes.” *arXiv: Applications*. <https://arxiv.org/pdf/1805.10036.pdf>. 12
- Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition. [MR2722294](#). doi: <https://doi.org/10.1007/978-0-387-84858-7>. 16
- Heyde, C. C. and Johnstone, I. M. (1979). “On asymptotic posterior normality for stochastic processes.” *Journal of the Royal Statistical Society: Series B (Methodological)*, 41: 184–189. [MR0547243](#). 6
- Hoff, P. (2009). *A First Course in Bayesian Statistical Methods*. Springer Texts in Statistics. Springer New York. URL <https://books.google.de/books?id=DykcMwEACAAJ> [MR2648134](#). doi: <https://doi.org/10.1007/978-0-387-92407-6>. 16
- Irons, N. J., Perrot-Dockès, M., and Metodiev, M. (2023). “thames: Easily Computed Marginal Likelihoods from Posterior Simulation Using the THAMES Estimator.” R package version 0.1.0. 3
- Jeffreys, H. (1961). *Theory of Probability*. Oxford, U. K.: Oxford University Press, 3rd edition. [MR0187257](#). 2
- Kass, R. E. and Raftery, A. E. (1995). “Bayes factors.” *Journal of the American Stat-*

- tistical Association*, 90(430): 773–795. MR3363402. doi: <https://doi.org/10.1080/01621459.1995.10476572>. 24
- Liu, B. (2014). “Adaptive annealed importance sampling for multimodal posterior exploration and model selection with application to extrasolar planet detection.” *The Astrophysical Journal Supplement Series*, 213(1): 14. 2
- Llorente, F., Martino, L., Delgado, D., and Lopez-Santiago, J. (2023). “Marginal likelihood computation for model selection and hypothesis testing: An extensive review.” *SIAM Review*, 65: 3–58. MR4545927. doi: <https://doi.org/10.1137/20M1310849>. 2
- McEwen, J. D., Wallis, C. G. R., Price, M. A., and Docherty, M. M. (2022). “Machine learning assisted Bayesian model comparison: learnt harmonic mean estimator.” *arXiv*. <https://arxiv.org/pdf/2111.12720.pdf>. 24
- Meng, X.-L. and Wong, W. H. (1996). “Simulating ratios of normalizing constants via a simple identity: A theoretical exploration.” *Statistica Sinica*, 6: 831–860. MR1422406. 2, 12
- Metodiev, M., Perrot-Dockès, M., Ouadah, S., Irons, N. J., Latouche, P., and Raftery, A. E. (2024a). “Supplement A to ‘Easily Computed Marginal Likelihoods from Posterior Simulation Using the THAMES Estimator’” doi: <https://doi.org/10.1214/24-BA1422SUPPA>. 6, 13
- Metodiev, M., Perrot-Dockès, M., Ouadah, S., Irons, N. J., Latouche, P., and Raftery, A. E. (2024b). “Supplement B to ‘Easily Computed Marginal Likelihoods from Posterior Simulation Using the THAMES Estimator’” doi: <https://doi.org/10.1214/24-BA1422SUPPB>. 6, 7, 12, 19
- Miller, J. W. (2021). “Asymptotic normality, concentration, and coverage of generalized posteriors.” *The Journal of Machine Learning Research*, 22: 7598–7650. MR4318524. 6, 8
- Newton, M. A. and Raftery, A. E. (1994). “Approximate Bayesian inference with the weighted likelihood bootstrap.” *Journal of the Royal Statistical Society: Series B (Methodological)*, 56: 3–26. MR1257793. 2, 5, 9
- Porwal, A. and Raftery, A. E. (2022). “Comparing methods for statistical inference with model uncertainty.” *Proceedings of the National Academy of Sciences*, 119(16): e2120737119. URL <https://www.pnas.org/doi/abs/10.1073/pnas.2120737119>. 16
- R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>. 23
- Robert, C. P. and Wraith, D. (2009). “Computational methods for Bayesian model choice.” In *AIP conference proceedings*, volume 1193, 251–262. American Institute of Physics. 4
- Shen, X. (2002). “Asymptotic Normality of Semiparametric and Nonparametric Pos-

terior Distributions.” *Journal of the American Statistical Association*, 97: 222–235. MR1947282. doi: <https://doi.org/10.1198/016214502753479365>. 6

Sims, C. A., Waggoner, D. F., and Zha, T. (2008). “Methods for inference in large multiple-equation Markov-switching models.” *Journal of Econometrics*, 146(2): 255–274. MR2465172. doi: <https://doi.org/10.1016/j.jeconom.2008.08.023>. 12

Skilling, J. (2006). “Nested sampling for general Bayesian computation.” *Bayesian Analysis*, 1: 833–859. MR2282208. doi: <https://doi.org/10.1214/06-BA127>. 2

Snijders, T. A. B. and Bosker, R. J. (1999). *Multilevel Analysis. An Introduction to Basic and Advanced Multilevel Modelling*. Sage. MR3137621. 21

Stamey, T., Kabalin, J., McNeal, J., Johnstone, I., Freiha, F., Redwine, E., and Yang, N. (1989). “Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate II radical prostatectomy treated patients.” *Journal of Urology*, 16: 1076–1083. 16

Stan Development Team (2022). “RStan: the R interface to Stan.” R package version 2.21.7. URL <https://mc-stan.org/> 23

Sweeting, T. (1996). “On a Converse to Scheffe’s Theorem.” *The Annals of Statistics*, 14: 1252–1256. MR0856821. doi: <https://doi.org/10.1214/aos/1176350065>. 8

Zellner, A. (1971). *An Introduction to Bayesian Inference in Econometrics*. Krieger. URL <https://books.google.de/books?id=paqiswEACAAJ> MR0433791. 16

Zellner, A. (1986). “Bayesian Estimation and Prediction Using Asymmetric Loss Functions.” *Journal of the American Statistical Association*, 81(394): 446–451. URL <http://www.jstor.org/stable/2289234> MR0845882. 16

Pierre Latouche

Material list:

- Daudin J.J., Picard F. and Robin S. (2008) A mixture model for random graphs. *Statistics and Computing*, 18, 173–183.
- Daudin J.J., Pierre L. and Vacher, C. (2010) Model for heterogeneous random networks using continuous latent variables and an application to a tree-fungus network, *Biometrics*, 66, 1043–1051.
- Airoldi E.M., Blei D.M., Fienberg S.E. and Xing E.P. (2008) Mixed membership stochastic blockmodels, *Journal of Machine Learning Research*, 9:65, 1981–2014.
- Kipf T.N. and Welling M. (2017) Semi-supervised classification with graph convolutional networks, arXiv:1609.02907.

A mixture model for random graphs

J.-J. Daudin · F. Picard · S. Robin

Received: 18 December 2006 / Accepted: 14 November 2007 / Published online: 11 December 2007
© Springer Science+Business Media, LLC 2007

Abstract The Erdős–Rényi model of a network is simple and possesses many explicit expressions for average and asymptotic properties, but it does not fit well to real-world networks. The vertices of those networks are often structured in unknown classes (functionally related proteins or social communities) with different connectivity properties. The stochastic block structures model was proposed for this purpose in the context of social sciences, using a Bayesian approach. We consider the same model in a frequentist statistical framework. We give the degree distribution and the clustering coefficient associated with this model, a variational method to estimate its parameters and a model selection criterion to select the number of classes. This estimation procedure allows us to deal with large networks containing thousands of vertices. The method is used to uncover the modular structure of a network of enzymatic reactions.

Keywords Random graphs · Mixture models · Variational method

1 Introduction

The Erdős–Rényi model of a network is one of the oldest and best studied model and possesses many explicit expressions for average and asymptotic properties such as degree distribution, connectedness and clustering coefficient. However this theoretical model does not fit well to real-world, social, biological or Internet networks. For example the empirical degree distribution may be very different from the

Poisson distribution which is implied by this model. Moreover empirical clustering coefficients of real networks are generally higher than the value given by this model. Other models have been proposed to correct these shortcomings (see Nowicki and Snijders 2001; Albert and Barabási 2002; or Molloy and Reed 1995). A good review of random graph models is given in Pattison and Robins (2007).

It appears that the available methods in the literature can be divided into two categories: model-based versus algorithmic methods. In the context of social sciences and using a Bayesian approach, the stochastic block structures model (Nowicki and Snijders 2001) assumes that vertices pertain to classes with different connectivity characteristics. This model provides a proper probabilistic framework but the proposed estimation method can not deal with networks made of more than 200 vertices. However, a special attention has been recently paid to the study of biological networks which are generally much larger (see Alm and Arkin 2002; or Arita 2004). Other algorithms have been proposed: assortative mixing or mixing patterns (Newman and Girvan 2003; Newman 2004). These methods are efficient on large networks but the absence of model makes the interpretation of the results more difficult.

The key element of those methods is the mixing matrix which specifies the probability of connection between two classes. The inference of the mixing parameters is quite easy if the classes can be defined using external information such as language, race or age. However the inference is more difficult when classes and mixing parameters have to be inferred when the network topology is the only available information.

In this article we use the model-based framework proposed by Nowicki and Snijders (2001) in a frequentist setting. We derive some new theoretical properties of this

J.-J. Daudin · F. Picard · S. Robin (✉)
Mathématiques et Informatique Appliquées, AgroParisTech and
INRA UMR518, 16, rue Claude Bernard, 75005 Paris, France
e-mail: robin@inapg.fr

model. We provide an estimation algorithm using a variational approach as well as a model selection criterion to choose the number of classes. This framework allows us to deal with thousands of vertices. Our method is illustrated on a biological network.

Notation In this article, we consider an undirected graph with n vertices and define the variable X_{ij} which indicates that vertices i and j are connected:

$$X_{ij} = X_{ji} = \mathbb{I}\{i \leftrightarrow j\},$$

where $\mathbb{I}\{A\}$ equals to one if A is true, and to zero otherwise. Furthermore, we assume that no vertex is connected to itself, meaning that $X_{ii} = 0$. However, the method we present below can be generalized to directed graphs ($X_{ij} \neq X_{ji}$) with self loops ($X_{ii} \neq 0$). In the following we note K_i the degree of vertex i , i.e. the number of edges connecting it:

$$K_i = \sum_{j \neq i} X_{ij}.$$

Erdős–Rényi model This model assumes that edges are independent and occur with the same probability p :

$$\{X_{ij}\} \text{ i.i.d.}, \quad X_{ij} \sim \mathcal{B}(p).$$

In this model, the degree of each vertex has a Binomial distribution, which is approximately Poisson for large n and small p . Noting $\lambda = (n - 1)p$ we have

$$K_i \sim \mathcal{B}(n - 1, p) \approx \mathcal{P}(\lambda).$$

2 Mixture model for the degrees

In many practical situations, the Erdős–Rényi model turns out to fit the data poorly, mainly because the distribution of the degrees is far from the Poisson distribution. The scale-free (or Zipf) distribution has been intensively used as an alternative. The Zipf probability distribution function (pdf) is

$$\Pr\{K_i = k\} = c(\rho)k^{-(\rho+1)}, \quad (1)$$

where k is any positive integer, ρ is positive, $c(\rho) = \sum_{k \geq 1} k^{-(\rho+1)} = 1/\zeta(\rho + 1)$ and $\zeta(\rho + 1)$ is Riemann's zeta function. Nevertheless, we will show in Sect. 6 that this distribution may have a poor fit on real datasets. This lack of fit has been already analyzed by Stumpf et al. (2005) and Tanaka and Doyle (2005).

First of all, it is important to notice that the Zipf distribution is used to model the tail of the degree distribution. Therefore it is often best suited for the tail than for the whole distribution. In particular this distribution has a null

probability for $k = 0$ whereas some vertices may be unconnected in practice. Moreover the lack-of-fit of the Erdős–Rényi model may be simply due to some heterogeneities between vertices, some being more connected than others. A simple way to model this phenomenon is to consider that the degree distribution is a mixture of Poisson distributions.

In the mixture framework we suppose that vertices are structured into Q classes, and that there exists a sequence of independent hidden variables $\{Z_{iq}\}$ (with $\sum_q Z_{iq} = 1$) which indicate the label of vertices to classes. We note α_q the *prior* probability for vertex i to belong to class q , such that:

$$\alpha_q = \Pr\{Z_{iq} = 1\} = \Pr\{i \in q\}, \quad \text{with } \sum_q \alpha_q = 1.$$

Remark 1 In the following, we will use two equivalent notations: $\{Z_{iq} = 1\}$ or $\{i \in q\}$ to indicate that vertex i belongs to class q .

We suppose that the conditional distribution of the degrees is a Poisson distribution

$$K_i | \{i \in q\} \sim \mathcal{P}(\lambda_q).$$

Then the distribution of the degrees is a mixture of Poisson distributions such that

$$\Pr\{K_i = k\} = \sum_q \alpha_q \frac{e^{-\lambda_q} \lambda_q^k}{k!}. \quad (2)$$

For a complete review and a careful statistical analysis of the modeling of the degree distribution in networks, see (Jones and Handcock 2004).

Remark 2 Because vertices are connected, degrees are not independent. However, in the standard situation where n is large and $\lambda_q \ll n$, the dependency between degrees is weak.

In Sect. 6 we show that this model fits well to real data. Nevertheless, we claim that modeling the distribution of the degrees provides little information about the topology of the graph. Indeed, this model only deals with the degrees of vertices, but not explicitly with the probability for two given vertices to be connected. However, the observed number of connections between vertices from different classes may reveal some interesting underlying structure, such as preferential connections between classes. The mixture model for degrees is not precise enough to describe such a phenomenon. This motivates the definition of an explicit mixture model for edges.

3 Erdős–Rényi mixture for graphs

3.1 General model

We now describe the stochastic block structures model (Nowicki and Snijders 2001), a mixture model which explicitly describes the way edges connect vertices, accounting for some heterogeneity among vertices. In the following this model is called “mixture model for graphs”.

The mixture model for graphs supposes that vertices are spread into Q classes with prior probabilities $\{\alpha_1, \dots, \alpha_Q\}$. In the following, we use the indicator variables $\{Z_{iq}\}$ (with $\sum_q Z_{iq} = 1$) defined in Sect. 2.

$$\alpha_q = \Pr\{Z_{iq} = 1\} = \Pr\{i \in q\}, \quad \text{with } \sum_q \alpha_q = 1.$$

Then we denote $\pi_{q\ell}$ the probability for a vertex from class q to be connected with a vertex from class ℓ . Because the graph is undirected, these probabilities must be symmetric such that

$$\pi_{q\ell} = \pi_{\ell q}.$$

We finally suppose that edges $\{X_{ij}\}$ are conditionally independent given the classes of vertices i and j :

$$\begin{cases} X_{ij} \mid \{i \in q, j \in \ell\} \sim \mathcal{B}(\pi_{q\ell}) & \text{for } i \neq j, \\ X_{ii} = 0. \end{cases}$$

The main difference with Model (2) is that the mixture model for graphs directly deals with edges. More than describing the clustered structure of vertices, our model describes the topology of the network using the connectivity matrix $\boldsymbol{\pi} = (\pi_{q\ell})$.

3.2 Examples

In this section we aim at showing that the mixture model for graphs can be used to generalize many particular structures of random graphs. Table 1 presents some typical network configurations. The first one is the Erdős–Rényi model. We present here some more sophisticated ones.

Example 1 Random graphs with arbitrary degree distributions. The Erdős–Rényi random graph model is a poor approximation of real-world networks whose degree distribution is highly skewed. A random network having the same degree distribution as the empirical one can be built as follows: n partial edges (with only one starting vertex and no final vertex) are randomly chosen from the empirical degree distribution. These partial edges are randomly joined by pairs to form complete edges (see Molloy and Reed 1995). A permutation algorithm is also proposed in Shen-Orr et al. (2002). This model assumes that the connectivity between

two vertices is proportional to the degree of each vertex so it coincides with the independent case of the mixture model for graphs presented in Sect. 4.4.

The scale-free network proposed by Barabási and Albert (1999) is a particular case of random graphs with arbitrary distribution. To this extent, we can propose an analogous model in the mixture model for graphs framework. Suppose that the incoming vertices join the network in classes of respective size $n\alpha_q$ ($q = 1, \dots, Q$, $n\alpha_1$ being the number of original vertices). Assuming that the elements of a new class connect preferentially the elements of the oldest classes:

$$\pi_{q,1} \geq \pi_{q,2} \geq \dots \geq \pi_{q,q-1},$$

we get the same kind of structure as the scale-free model.

Example 2 Affiliation network. An affiliation network is a social network in which actors are joined by a common participation in social events, companies boards or scientists’ coauthorship of papers. All the vertices participating to the same class are connected. This model has been studied by Newman et al. (2002). This type of network may be modeled by a mixture model for graphs with ones in the diagonal of $\boldsymbol{\pi}$.

Example 3 Star pattern. Many biological networks contain star patterns, i.e. many vertices connected to the same vertex and only to it, see interaction networks of *S. Cerevisiae* in Zhang et al. (2005) for instance. This type of pattern may be modeled by a mixture model for graphs with extra-diagonal ones in $\boldsymbol{\pi}$.

4 Some properties of the mixture model for graphs

4.1 Distribution of the degrees

Proposition 1 Given the label of a vertex, the conditional distribution of the degree of this vertex is Binomial (approximately Poisson):

$$K_i \mid \{i \in q\} \sim \mathcal{B}(n - 1, \bar{\pi}_q) \approx \mathcal{P}(\lambda_q),$$

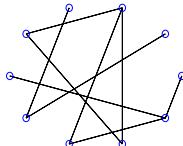
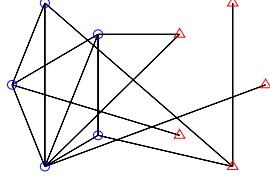
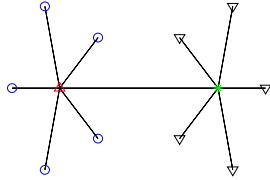
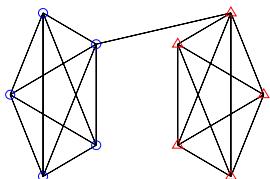
where $\bar{\pi}_q = \sum_\ell \alpha_\ell \pi_{q\ell}$ and $\lambda_q = (n - 1)\bar{\pi}_q$.

Proof Conditionally to the belonging of vertices to classes, edges connecting vertex i belonging to class q are independent. The conditional connection probability is:

$$\begin{aligned} \Pr\{X_{ij} = 1 \mid i \in q\} &= \sum_\ell \Pr\{X_{ij} = 1 \mid i \in q, j \in \ell\} \Pr\{j \in \ell\} \\ &= \sum_\ell \alpha_\ell \pi_{q\ell} = \bar{\pi}_q. \end{aligned}$$

The result follows. \square

Table 1 Some typical network configurations and their formulation in the framework of the mixture model for graphs. The node marks (\circ , \triangle , ∇ , \star) refer to their class

Description	Network	Q	π	Clustering coef.
Random		1	p	p
Product connectivity (arbitrary degree distribution)		2	$\begin{pmatrix} a^2 & ab \\ ab & b^2 \end{pmatrix}$	$\frac{(a^2 + b^2)^2}{(a+b)^2}$
Stars		4	$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	0
Clusters (affiliation networks)		2	$\begin{pmatrix} 1 & \varepsilon \\ \varepsilon & 1 \end{pmatrix}$	$\frac{1+3\varepsilon^2}{(1+\varepsilon)^2}$

4.2 Between-class connectivity

Definition 1 The connectivity between class q and ℓ is the number of edges connecting a vertex from class q to a vertex from class ℓ :

$$A_{q\ell} = \sum_{i < j} Z_{iq} Z_{j\ell} X_{ij}.$$

A_{qq} is the within-connectivity of class q .

Proposition 2 The expected connectivity between class q and ℓ is:

$$\mathbb{E}(A_{q\ell}) = n(n-1)\alpha_q \alpha_\ell \pi_{q\ell}/2.$$

Proof According to Definition 1, $A_{q\ell}$ is the sum over $n(n-1)/2$ terms. Conditionally to $\{Z_{iq} Z_{j\ell} = 1\}$, X_{ij} is a Bernoulli variable with parameter $\pi_{q\ell}$. Thus

$\mathbb{E}(Z_{iq} Z_{j\ell} X_{ij}) = \mathbb{E}(Z_{iq} Z_{j\ell}) \pi_{q\ell}$. The Z_{iq} s are independent, so we have $\mathbb{E}(Z_{iq} Z_{j\ell}) = \alpha_q \alpha_\ell$. The result follows. \square

4.3 Clustering coefficient

This coefficient is supposed to measure the aggregative trend of a graph. Since no probabilistic modeling is usually available, this coefficient is empirically defined in most cases. Albert and Barabási (2002) propose the following definition of the empirical clustering coefficient for vertex i :

$$C_i = \nabla_i / \frac{K_i(K_i - 1)}{2},$$

where ∇_i is the number of edges between the neighbors of vertex i : $\nabla_i = \sum_{j,k} X_{ij} X_{jk} X_{ik}/2$, whose minimum value is 0 and maximum value equals $K_i(K_i - 1)/2$ for a clique. A first estimator of this empirical clustering coefficient is

usually defined as the mean of the C_i 's:

$$\hat{c} = \sum_i C_i / n.$$

Denoting ∇ the “triangle” configuration ($i \leftrightarrow j \leftrightarrow k \leftrightarrow i$) and V the ‘V’ configuration ($j \leftrightarrow i \leftrightarrow k$) for any (i, j, k) uniformly chosen in $\{1, \dots, n\}$, the definition of C can be rephrased as $c = \Pr\{\nabla \mid V\}$. Because ∇ is a particular case of V , we have

$$c = \Pr\{\nabla \cap V\} / \Pr\{V\} = \Pr\{\nabla\} / \Pr\{V\}. \quad (3)$$

This property suggests another estimate of c proposed by Newman et al. (2002):

$$\tilde{c}' = 3 \sum_i \nabla_i / \sum_i V_i,$$

where V_i is the number of V s in i : $V_i = \sum_{j>k, (j,k)\neq i} X_{ij} X_{ik}$. In the following we propose a probabilistic definition of this coefficient.

Definition 2 The clustering coefficient is the probability for two vertices j and k connected to a third vertex i , to be connected, with (i, j, k) uniformly chosen in $\{1, \dots, n\}$

$$c = \Pr\{X_{ij} X_{jk} X_{ki} = 1 \mid X_{ij} X_{ik} = 1\}.$$

Proposition 3 In the mixture model for graphs, the clustering coefficient is

$$c = \sum_{q,\ell,m} \alpha_q \alpha_\ell \alpha_m \pi_{q\ell} \pi_{qm} \pi_{\ell m} / \sum_{q,\ell,m} \alpha_q \alpha_\ell \alpha_m \pi_{q\ell} \pi_{qm}$$

Proof For any triplet (i, j, k) , we have

$$\begin{aligned} \Pr\{\nabla\} &= \sum_{q,l,m} \alpha_q \alpha_l \alpha_m \\ &\times \Pr\{X_{ij} X_{jk} X_{ki} = 1 \mid i \in q, j \in l, k \in m\}, \\ &= \sum_{q,l,m} \alpha_q \alpha_l \alpha_m \pi_{q\ell} \pi_{qm} \pi_{\ell m}. \end{aligned}$$

The same reasoning can be applied to $\Pr\{V\}$ recalling that the event V in (i, j, k) means that the top of V is i . The result is then an application of (3). \square

4.4 Independent model

The model presented in Sect. 2 can be rephrased as an independent version of the mixture model for graphs. Indeed the absence of preferential connection between classes corresponds to the case where

$$\pi_{q\ell} = \eta_q \eta_\ell. \quad (4)$$

η_q is then the connection propensity of a vertex from class q , regardless the class of the other vertex. The properties of the independent model are as follows.

Distribution of degrees The conditional distribution of the degrees is Poisson with parameter λ_q such that

$$\lambda_q = (n-1)\eta_q \bar{\eta}, \quad (5)$$

where $\bar{\eta} = \sum_\ell \alpha_\ell \eta_\ell$, so λ_q is directly proportional to η_q .

Between class connectivity We get

$$\mathbb{E}(A_{q\ell}) = n(n-1)(\alpha_q \eta_q)(\alpha_\ell \eta_\ell)/2,$$

so the rows and columns of matrix $\mathbf{A} = (A_{q\ell})_{q,\ell}$ must all have the same profile. We will see in Sect. 6 that the observed number of connections between classes may be quite far from expected values.

Clustering coefficient

$$c = \frac{(\sum_q \alpha_q \eta_q^2)^2}{\bar{\eta}^2}.$$

For the standard Erdős-Rényi model ($Q = 1, \alpha_1 = 1, \bar{\eta} = \eta_1 = \sqrt{p}$), we get the known result: $c = \eta_1^4 / \eta_1^2 = p$.

Considering the independent case presented in Table 1 with $\alpha_1 = \alpha_2 = 1/2$ and $a = 0.9, b = 0.1$, we get $c = (0.9^2 + 0.1^2)^2 \simeq 0.67$. The corresponding Erdős-Rényi model with $p = (\alpha_1 a + \alpha_2 b)^2 = 1/4$ would lead to a strong underestimation of c since $c = p = 0.25$.

4.5 Likelihoods

In order to define the likelihood of the model, we use the incomplete-data framework defined by Dempster et al. (1977). Let \mathcal{X} denote the set of all edges: $\mathcal{X} = \{X_{ij}\}_{i,j=1,\dots,n}$, and \mathcal{Z} the set of all indicator variables for vertices: $\mathcal{Z} = \{Z_{iq}\}_{i=1,n}^{q=1,Q}$.

Proposition 4 The complete-data log-likelihood is

$$\begin{aligned} \log \mathcal{L}(\mathcal{X}, \mathcal{Z}) &= \sum_i \sum_q Z_{iq} \log \alpha_q \\ &+ \frac{1}{2} \sum_{i \neq j} \sum_{q,\ell} Z_{iq} Z_{j\ell} \log b(X_{ij}; \pi_{q\ell}). \end{aligned}$$

Proof We have $\log \mathcal{L}(\mathcal{X}, \mathcal{Z}) = \log \mathcal{L}(\mathcal{Z}) + \log \mathcal{L}(\mathcal{X} \mid \mathcal{Z})$ where

$$\log \mathcal{L}(\mathcal{Z}) = \sum_i \sum_q Z_{iq} \log \alpha_q,$$

$$\log \mathcal{L}(\mathcal{X} \mid \mathcal{Z}) = \frac{1}{2} \sum_{i \neq j} \sum_{q,\ell} Z_{iq} Z_{j\ell} \log b(X_{ij}; \pi_{q\ell}),$$

and $b(x; \pi) = \pi^x (1 - \pi)^{1-x}$. \square

The likelihood of the observed data $\mathcal{L}(\mathcal{X})$ is obtained by summing the complete-data likelihood over all the possible values of the unobserved variables \mathcal{Z} . Unfortunately, this sum is not tractable and it seems that no simpler form can be derived.

5 Estimation

In this section we propose a variational approach to perform an approximate maximum likelihood inference on the parameters. We follow the general strategy described in Jordan et al. (1999) or in the tutorial by Jaakkola (2000). A similar strategy is used in the bi-clustering framework by Govaert and Nadif (2005). The general statistical properties of the resulting estimators have not been investigated yet. However, this approximation allows us to deal with large networks (several thousands of nodes) whereas the Bayesian strategy adopted by Nowicki and Snijders (2001) restricts the estimation to 200 nodes.

5.1 Dependency graph

The X_{ij} s are independent conditionally to the Z_{iq} s, but are marginally dependent. For estimation purpose, it is important to know if $\Pr\{Z_{iq} = 1 \mid \mathcal{X}\}$ is equal to $\Pr\{Z_{iq} = 1 \mid \mathcal{X}_i\}$, where \mathcal{X}_i is the set of all possible edges connecting i . \mathcal{X}_i is often called the set of neighbors of vertex i . In the following, we give a counter example to show that the notion of neighborhood can not be used in the mixture model for graphs framework.

Assume that the vertices are divided in two classes, whose connectivity matrix is diagonal with $\pi_{11} = 1$ and $\pi_{22} = a$ and $0 < a < 1$. Let us consider 3 vertices i, j, k with $X_{ij} = X_{ik} = 1$. The vertices i and j are in the same class because no connection is possible between vertices pertaining to two different classes. The same is true for vertices i and k . Therefore the three vertices are in the same class and we have $\Pr\{Z_{i1} = 1 \mid \mathcal{X}_i, X_{jk}\} > 0$ if $X_{jk} = 1$ and $\Pr\{Z_{i1} = 1 \mid \mathcal{X}_i, X_{jk}\} = 0$ if $X_{jk} = 0$. Therefore $\Pr\{Z_{iq} = 1 \mid \mathcal{X}\}$ depends on all the network and not only on edges connecting to the vertex i .

This counter example clearly shows that no neighborhood can be considered in the framework of mixture model for graphs, since unconnected vertices provide as much information as connected vertices. This is why the likelihood can not be simplified for computation.

5.2 Variational approach

As often for incomplete data models, the likelihood of the observed data $\mathcal{L}(\mathcal{X})$ is not tractable. EM (Dempster et al. 1977) is the most popular algorithm for this kind of prob-

lem. Unfortunately, EM requires the computation of the conditional distribution $\Pr(\mathcal{Z} \mid \mathcal{X})$ which is itself not tractable, as explained above. Therefore, we choose a variational approach that aims at optimizing a lower bound of $\log \mathcal{L}(\mathcal{X})$, denoted by

$$\mathcal{J}(R_{\mathcal{X}}) = \log \mathcal{L}(\mathcal{X}) - \text{KL}[R_{\mathcal{X}}(\cdot), \Pr(\cdot \mid \mathcal{X})],$$

where KL denotes the Kullback–Leibler divergence, $\Pr(\mathcal{Z} \mid \mathcal{X})$ is the true conditional distribution of the indicator variables \mathcal{Z} given the data \mathcal{X} , and $R_{\mathcal{X}}$ an approximation of this conditional distribution. $\mathcal{J}(R_{\mathcal{X}})$ equals $\log \mathcal{L}(\mathcal{X})$ iff $R_{\mathcal{X}}(\cdot) = \Pr(\cdot \mid \mathcal{X})$. We emphasize that $R_{\mathcal{X}}$ depends on the data \mathcal{X} .

As shown above, we are not able to calculate $\Pr(\cdot \mid \mathcal{X})$, so we will look for the “best” (in terms of Kullback–Leibler divergence) $R_{\mathcal{X}}$ in a certain class of distributions. The estimation algorithm we propose will alternate the maximization of $\mathcal{J}(R_{\mathcal{X}})$ (i) with respect to $R_{\mathcal{X}}$ and (ii) with respect to parameters α and π . Propositions 5 and 6 give the solutions of the optimization problems (i) and (ii) respectively.

Approximate conditional distribution $R_{\mathcal{X}}$ Denoting $\mathcal{Z}_i = \{Z_{i1}, \dots, Z_{iQ}\}$, we constraint $R_{\mathcal{X}}$ to have the following form:

$$R_{\mathcal{X}}(\mathcal{Z}) = \prod_i h(\mathcal{Z}_i; \boldsymbol{\tau}_i)$$

where $\boldsymbol{\tau}_i = (\tau_{i1}, \dots, \tau_{iQ})$ and $h(\cdot; \boldsymbol{\tau})$ denotes the multinomial distribution with parameter $\boldsymbol{\tau}$. τ_{iq} can be interpreted as an approximation of $\Pr\{Z_{iq} = 1 \mid \mathcal{X}\}$. This corresponds to the mean field approximation, as presented in Jaakkola (2000).

Proposition 5 Given parameters α and π , the optimal variational parameters $\{\hat{\boldsymbol{\tau}}_i\} = \arg \max_{\{\boldsymbol{\tau}_i\}} \mathcal{J}(R_{\mathcal{X}})$ satisfy the following fixed point relation:

$$\hat{\tau}_{iq} \propto \alpha_q \prod_{j \neq i} \prod_{\ell} b(X_{ij}; \pi_{q\ell})^{\hat{\tau}_{j\ell}}.$$

Proof Based on the definition of the Kullback–Leibler divergence, we first rewrite $\mathcal{J}(R_{\mathcal{X}})$ as

$$\begin{aligned} \mathcal{J}(R_{\mathcal{X}}) &= \sum_{\mathcal{Z}} R_{\mathcal{X}}(\mathcal{Z}) \log \Pr(\mathcal{Z}, \mathcal{X}) \\ &\quad - \sum_{\mathcal{Z}} R_{\mathcal{X}}(\mathcal{Z}) \log R_{\mathcal{X}}(\mathcal{Z}) \\ &= \sum_i \sum_q \tau_{iq} \log \alpha_q \\ &\quad + \frac{1}{2} \sum_{i \neq j} \sum_{q, \ell} \tau_{iq} \tau_{j\ell} \log b(X_{ij}; \pi_{q\ell}) \\ &\quad - \sum_i \sum_q \tau_{iq} \log \tau_{iq}. \end{aligned}$$

We now have to maximize $\mathcal{J}(R_{\mathcal{X}})$ with respect to the τ_{iq} 's, subject to $\sum_q \tau_{iq} = 1$, for all i , i.e. to maximize $\mathcal{J}(R_{\mathcal{X}}) + \sum_i [\lambda_i(\sum_q \tau_{iq} - 1)]$ where λ_i is the Lagrange multiplier. The derivative with respect to τ_{iq} is

$$\log \alpha_q + \sum_{j \neq i} \sum_{\ell} \tau_{j\ell} \log b(X_{ij}; \pi_{q\ell}) - \log \tau_{iq} + 1 + \lambda_i.$$

This derivative is null iff $\widehat{\tau}_{iq}$'s satisfy the relation given in the proposition, $\exp(1 + \lambda_i)$ being the normalizing constant. \square

From a practical point of view, the $\{\widehat{\tau}_i\}$ are updated using a fixed point algorithm. At this time, we have no guaranty about the convergence toward a unique solution. In all situations we experienced, the algorithm converged rapidly.

Parameter estimates To complete the estimation procedure, we need to maximize $\mathcal{J}(R_{\mathcal{X}})$ with respect to parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$.

Proposition 6 Given the variational parameters $\{\tau_i\}$, the values of parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$ that maximize $\mathcal{J}(R_{\mathcal{X}})$ are

$$\widehat{\alpha}_q = \frac{1}{n} \sum_i \widehat{\tau}_{iq}, \quad \widehat{\pi}_{q\ell} = \sum_{i \neq j} \widehat{\tau}_{iq} \widehat{\tau}_{j\ell} X_{ij} / \sum_{i \neq j} \widehat{\tau}_{iq} \widehat{\tau}_{j\ell}.$$

Proof Due to the constraint on $\boldsymbol{\alpha}$, we have to maximize $\mathcal{J}(R_{\mathcal{X}}) + \lambda(\sum_q \alpha_q - 1)$. The calculation of the derivatives is straightforward and the result follows. \square

Estimation algorithm The algorithm we propose is the following. Starting with some initial values $\{\tau_i^{(0)}\}$ for the variational parameters, we iteratively update parameters τ_i , $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$ as follows:

$$(\boldsymbol{\alpha}^{(h+1)}, \boldsymbol{\pi}^{(h+1)}) = \arg \max_{(\boldsymbol{\alpha}, \boldsymbol{\pi})} \mathcal{J}(R_{\mathcal{X}}; \{\tau_i^{(h)}\}, \boldsymbol{\alpha}, \boldsymbol{\pi}),$$

$$\{\tau_i^{(h+1)}\} = \arg \max_{\{\tau_i\}} \mathcal{J}(R_{\mathcal{X}}; \{\tau_i\}, \boldsymbol{\alpha}^{(h+1)}, \boldsymbol{\pi}^{(h+1)}).$$

These updates are performed according to Propositions 5 and 6.

Proposition 7 For a given number of classes Q , this algorithm generates a sequence $\{\{\tau_i^{(h)}\}, \boldsymbol{\alpha}^{(h)}, \boldsymbol{\pi}^{(h)}\}_{h \geq 0}$ which increases $\mathcal{J}(R_{\mathcal{X}})$ such that

$$\begin{aligned} \mathcal{J}(R_{\mathcal{X}}; \{\tau_i^{(h+1)}\}, \boldsymbol{\alpha}^{(h+1)}, \boldsymbol{\pi}^{(h+1)}) \\ \geq \mathcal{J}(R_{\mathcal{X}}; \{\tau_i^{(h)}\}, \boldsymbol{\alpha}^{(h)}, \boldsymbol{\pi}^{(h)}). \end{aligned}$$

Proof This is a direct consequence of Propositions 5 and 6, which both guaranty that $\mathcal{J}(R_{\mathcal{X}})$ increases. \square

5.3 Choice of the number of classes

In practice the number of classes is unknown and should be estimated. We derive a Bayesian model selection criterion for this purpose which is based in the Integrated Classification Likelihood (ICL) criterion developed by Biernacki et al. (2000). We denote by $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\pi})$ the entire set of the mixture parameters which lies in $\Theta = A \times \Pi$, with A the Q -dimensional simplex and $\Pi = [0, 1]^{Q(Q+1)/2}$. Then we denote by $g_1(\boldsymbol{\alpha} | m_Q)$ and $g_2(\boldsymbol{\pi} | m_Q)$ the prior distributions of the parameters for a model m_Q with Q classes. The ICL criterion is an approximation of the complete-data integrated likelihood defined such that:

$$\mathcal{L}(\mathcal{X}, \mathcal{Z} | m_Q) = \int_{\Theta} \mathcal{L}(\mathcal{X}, \mathcal{Z} | \boldsymbol{\theta}, m_Q) g(\boldsymbol{\theta} | m_Q) d\boldsymbol{\theta},$$

where $\mathcal{L}(\mathcal{X}, \mathcal{Z} | \boldsymbol{\theta}, m_Q)$ is the complete-data likelihood of model m_Q with Q classes.

Proposition 8 For a model m_Q with Q classes, the ICL criterion is:

$$\begin{aligned} ICL(m_Q) &= \max_{\boldsymbol{\theta}} \log \mathcal{L}(\mathcal{X}, \widetilde{\mathcal{Z}} | \boldsymbol{\theta}, m_Q) \\ &\quad - \frac{1}{2} \times \frac{Q(Q+1)}{2} \log \frac{n(n-1)}{2} - \frac{Q-1}{2} \log(n). \end{aligned}$$

Proof The derivation of ICL is based on the following lemma by Biernacki et al. (2000) which can be applied to our case: if $g(\boldsymbol{\theta} | m_Q) = g_1(\boldsymbol{\alpha} | m_Q) \times g_2(\boldsymbol{\pi} | m_Q)$ then $\log \mathcal{L}(\mathcal{X}, \mathcal{Z} | m_Q) = \log \mathcal{L}(\mathcal{Z} | m_Q) + \log \mathcal{L}(\mathcal{X} | \mathcal{Z}, m_Q)$. The derivation of the first term can be done directly, using a Dirichlet prior, $\mathcal{D}(\delta)$ on proportions, which gives:

$$\begin{aligned} \log \mathcal{L}(\mathcal{Z} | m_Q) \\ = \log \int \alpha_1^{n_1} \dots \alpha_Q^{n_Q} \frac{\Gamma(Q\delta)}{\Gamma(\delta)^Q} \mathbb{I}\left(\sum_q \alpha_q = 1\right) d\boldsymbol{\alpha}, \\ = \log \Gamma(Q\delta) + \sum_q \log \Gamma(n_q + \delta) - Q \log \Gamma(\delta) \\ - \log \Gamma(n + Q\delta), \end{aligned}$$

where n_q is the number of nodes in class q . Since n_q 's are unknown, we replace the missing data \mathcal{Z} by their prediction $\widetilde{\mathcal{Z}}$. Then we consider a non informative Jeffreys prior distribution which corresponds to $\delta = 1/2$. This gives:

$$\begin{aligned} \log \mathcal{L}(\widetilde{\mathcal{Z}} | m_Q) &= \log \Gamma(Q/2) + \sum_q \log \Gamma(\tilde{n}_q + 1/2) \\ &\quad - Q \log \Gamma(1/2) - \log \Gamma(n + Q/2), \end{aligned}$$

with n the total number of nodes. Then we take the limit of this quantity for large n , and using the Stirling formula to

approximate the Gamma function we obtain

$$\begin{aligned}\log \mathcal{L}(\tilde{\mathcal{Z}} | m_Q) &= \sum_q \tilde{n}_q \log(\tilde{n}_q) - n \log(n) - \frac{Q-1}{2} \log(n) \\ &= \max_{\alpha} \log \mathcal{L}(\tilde{\mathcal{Z}} | \alpha, m_Q) - \frac{Q-1}{2} \log(n).\end{aligned}$$

As for the second term, we have $n(n-1)/2$ Bernoulli random variables with fixed labels and $\log \mathcal{L}(\mathcal{X} | \mathcal{Z}, m_Q)$ can be calculated using a BIC approximation:

$$\begin{aligned}\log \mathcal{L}(\mathcal{X} | \mathcal{Z}, m_Q) &\simeq \max_{\pi} \log \mathcal{L}(\mathcal{X} | \mathcal{Z}, \pi, m_Q) \\ &\quad - \frac{1}{2} \times \frac{Q(Q+1)}{2} \log \frac{n(n-1)}{2}.\end{aligned}$$

Finally, the sum of these two separate terms completes the proof. \square

6 Application to biological networks

We apply the methodology developed in this paper to an metabolic network of bacteria *Escherichia coli*: the small molecule interaction metabolism network. In this network, vertices are chemical reactions. Two reactions are connected if a compound produced by the first one is a part of the second one (or vice-versa). The original data are issued from <http://biocyc.org/>. They have been curated to remove some of the secondary compounds. The network we analyzed is available at <http://pbil.univ-lyon1.fr/software/motus/>; it is made up of $n = 605$ vertices and the total number of edges is 1782. We emphasize that the algorithm we propose is currently the only inferential method which can handle such a large network.

We first show that the Poisson mixture defined in Sect. 2 better fits the observed degree distribution than the scale free distribution. Then we apply the mixture model for graphs to uncover the structure of this metabolic network.

6.1 Fit of the empirical distribution of the degrees

Many papers claim that the Zipf pdf (defined in (1)) fits well the empirical degree distribution of real networks, but these claims are rarely based on statistical criteria. Moreover, the Zipf distribution is not defined for degree zero, so a threshold (minimal degree) must be defined arbitrarily. In order to assess the quality of fit of the Zipf pdf to the tail of the empirical distribution, we compute the usual chi-square statistics for different thresholds. The minimum chi-square estimate of ρ are computed for each threshold. Table 2 shows that the fit is not good even for the tail distribution with a high value of the threshold. Consequently, the Zipf pdf is only a rough approximation of the true one. It is often better suited for the tail than for the whole distribution.

The fit of the mixture of Poisson distributions is presented in Fig. 1. The BIC criterion selects three classes. Parameter estimates are given in Table 3, and Table 2 shows that the fit of the Poisson mixture is better than the fit of the Zipf distribution. The lack of fit for the two first lines is due to an unexpectedly high number of vertices with two connections: 12 vertices have no connection, 44 have one connection and 150 have two connections. This particular structure is due to a large number of chain reactions which constitute intermediates between two others.

6.2 Mixture modeling of the network

The ICL criterion selects a model with $Q = 21$ classes whose parameter estimates are given in Table 4. Figure 2 presents the graph as a dot-plot where a dot at row i and column j indicates that the edge $i \rightarrow j$ is present. To emphasize the connections between the different classes, vertices are reordered within classes. Limits between classes are obtained using a *maximum a posteriori* classification rule: vertex i is classified into the class for which $\hat{\tau}_{iq}$ is maximal.

Among the first 20 classes, eight are cliques ($\pi_{qq} = 1$) and six have within probability connectivity greater than 0.5. It turns out that all those cliques or pseudo-cliques gather

Table 2 Fit of the power law and Poisson mixture to the degree distribution: Chi-square (χ^2) statistics, degree of freedom and ratio χ^2/df for several thresholds. The same values of the parameters of the Poisson mixture have been used for all thresholds

Threshold	n	$\rho + 1$	Power law			Poisson mixture		
			χ^2 stat.	df	χ^2/df	χ^2 stat.	df	χ^2/df
0	593	—	—	—	—	67.25	29	2.32
1	549	1.79	96.22	32	3.01	58.5	28	2.09
2	399	1.93	75.83	31	2.45	32.3	27	1.20
3	315	2.08	59.7	30	1.99	30.6	26	1.18
4	252	2.19	53.07	29	1.83	27	25	1.08
5	200	2.24	52.37	28	1.87	27	24	1.13
6	172	2.37	45.44	27	1.68	25	23	1.09

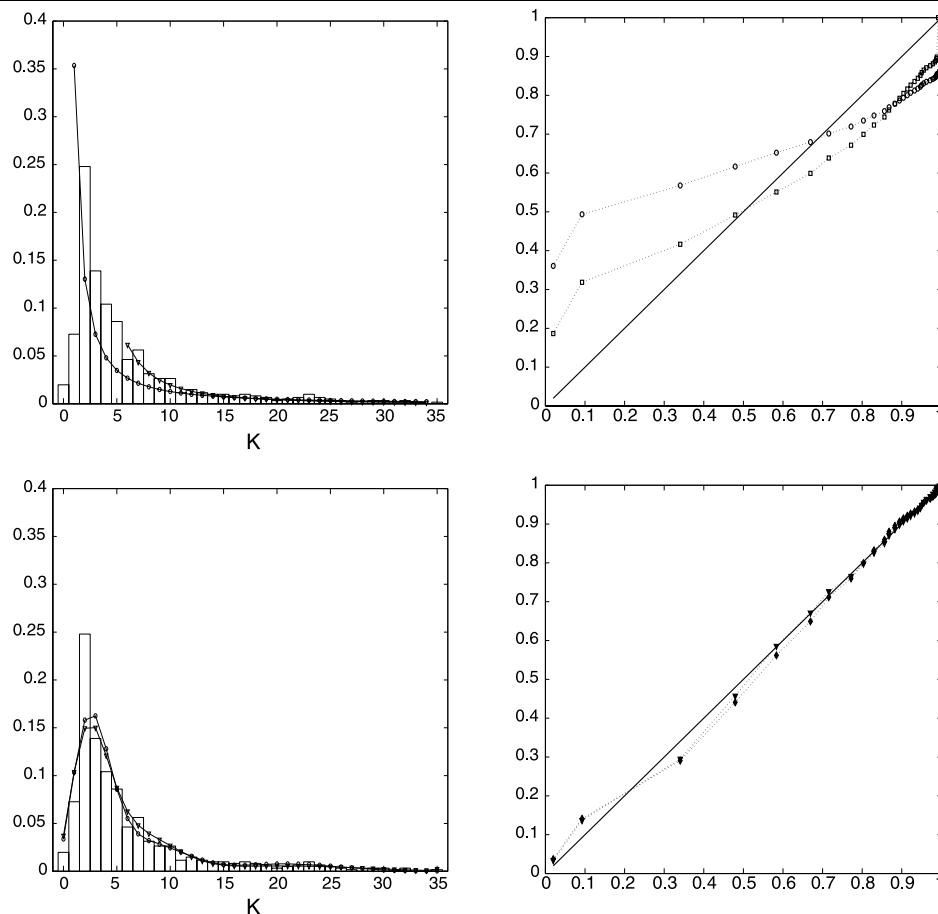


Fig. 1 Fit of the Zipf (top) and Poisson mixture (bottom) pdf on the *E. Coli* data. *Left:* histogram of degrees with adjusted distributions (Zipf: threshold 1 —○— and 6 —▽—, Poisson mixture: 3 classes —▽— and 6 classes —○—). *Right:* PP plots

Table 3 Parameter estimates for the Poisson mixture model on degrees with 3 classes

Class	1	2	3
α (%)	8.9	19.7	71.3
λ	21.5	9.1	3.0

reactions involving a same compound. Examples of compounds responsible for cliques include chorismate, pyruvate, L-aspartate, L-glutamate, D-glyceraldehyde-3-phosphate and ATP. That set of metabolites can be viewed as the backbone of the network.

Since the connection probability between classes 1 and 16 is 1, they constitute a clique which is associated with a single compound: pyruvate. However, that clique is split in two sub-cliques because of their different connectivities with reactions of classes 7 and 10. This distinction is due to the use of CO₂ in class 7 and acetylCoA in class 10, which

are secondary compounds involved in reactions of class 1 but not in those of class 16.

Remark 3 Table 4 also shows that the clique structure strongly increases the mean degree λ_q of its elements.

Remark 4 In this example, it turns out that the within connection probabilities π_{qq} are always maximal. A simulation study (not shown) prove that it is not an artifact of the method, which can detect classes without intra-connection ($\pi_{qq} = 0$).

To end, we also compare the expected clustering coefficient c given in Proposition 3 with the empirical one. The expected value for $Q = 21$ classes is 0.544, while the observed one is 0.626. The mixture model for graphs therefore slightly underestimates this coefficient. On the same dataset, the Erdős–Rényi model would give $\hat{c} = \hat{\pi} = 0.0098$.

Fig. 2 Top: Dot-plot representation of the adjacency matrix of the graph after classification of the vertices into the 21 classes

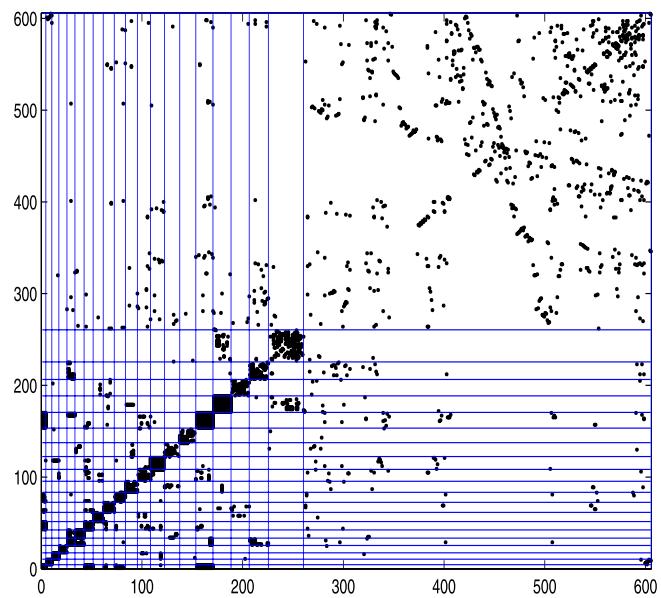


Table 4 Parameter estimates of the mixture model for graphs with $Q = 21$ classes: α , π and λ_q s. Values smaller than 0.5% are hidden for readability

α (%)	0.7	1.0	1.2	1.3	1.3	1.5	1.5	1.6	1.8	1.8	2.0	2.1	2.3	2.6	2.7	2.8	3.0	3.0	3.3	5.8	56.8
100						64		11	43			2				100					
100												4		7		1					1
100																					
71																					
	100	28										1									
	28	100																			
64						58		10	4			7		5							
						63						5									
11						10		65													
43						1		4		67				1							
π (%)											62			7							
	4							7	5			28		5							
2		7						5			1	5	100								
			6								7			25							
			1												40						
100			18			5		1			5		1			100					
						2					4					100					
1						3		2									21				
			16															19			
λ_q	33	7	9	6	17	13	12	7	10	10	10	8	17	6	7	25	21	5	6	5	3

Acknowledgements The data and the original biological problem have been provided by V. Lacroix and M.-F. Sagot (INRIA-Hélix, INRIA, Lyon). The authors also thank C. Matias, E. Birmelé (CNRS-

Statistic and Genome group, Evry univ.) and S. Schbath (INRA-MIG, Jouy-en-Josas) for all their helpful remarks and suggestions.

References

- Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**, 47–97 (2002)
- Alm, E., Arkin, A.P.: Biological networks. *Cur. Op. Struct. Biol.* **13**, 193–202 (2002)
- Arita, M.: The metabolic world of *Escherichia coli* is not small. *PNAS* **101**, 1543–1547 (2004)
- Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**, 509–512 (1999)
- Bernacki, C., Celeux, G., Govaert, G.: Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 719–725 (2000)
- Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc. B* **39**, 1–38 (1977)
- Govaert, G., Nadif, M.: An EM algorithm for the block mixture model. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 643–647 (2005)
- Jaakkola, T.: Advanced Mean Field Methods: Theory and Practice. MIT Press, Cambridge (2000). Chapter: Tutorial on variational approximation methods
- Jones, J., Handcock, M.: Likelihood-based inference for stochastic models of sexual network formation. *Theor. Pop. Biol.* **65**, 413–422 (2004)
- Jordan, M.I., Ghahramani, Z., Jaakkola, T., Saul, L.K.: An introduction to variational methods for graphical models. *Mach. Learn.* **37**, 183–233 (1999)
- Molloy, M., Reed, B.: A critical point for random graphs with a given degree sequence. *Rand. Struct. Algorithms* **6**, 161–179 (1995)
- Newman, M.E.J.: Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **69**, 066133 (2004)
- Newman, M.E.J., Girvan, M.: Statistical Mechanics of Complex Networks. Springer, Berlin (2003). Chapter: Mixing patterns and community structure in networks
- Newman, M.E.J., Watts, D.J., Strogatz, S.H.: Random graph models of social networks. *PNAS* **99**, 2566–2572 (2002)
- Nowicki, K., Snijders, T.: Estimation and prediction for stochastic block-structures. *J. Am. Stat. Assoc.* **96**, 1077–1087 (2001)
- Pattison, P.E., Robins, G.L.: Handbook of Probability Theory with Applications. Sage, Beverley Hills (2007). Chapter: Probabilistic network theory
- Shen-Orr, S.S., Milo, R., Mangan, S., Alon, U.: Networks motifs in the transcriptional regulation network of *Escherichia coli*. *Nat. Genet.* **31**, 64–68 (2002)
- Stumpf, M., Wiuf, C., May, R.: Subnets of scale-free networks are not scale-free: sampling properties of networks. *Proc. Natl. Acad. Sci. USA* **102**, 4221–4224 (2005)
- Tanaka, R., Doyle, J.: Some protein interaction data do not exhibit power law statistics. *FEBS Lett.* **579**, 5140–5144 (2005)
- Zhang, V.L., King, O.D., Wong, S.L., Goldberg, D.S., Tong, A.H.Y., Lesage, G., Andrews, B., Bussey, H., Boone, C., Roth, F.P.: Motifs, themes and thematic maps of an integrated *Saccharomyces cerevisiae* interaction network. *J. Biol.* **4**, 1–13 (2005)

Model for Heterogeneous Random Networks Using Continuous Latent Variables and an Application to a Tree–Fungus Network

Jean-Jacques Daudin,^{1,*} Laurent Pierre,² and Corinne Vacher³

¹UMR AgroParisTech/INRA518, AgroParisTech, Paris, France

²University Paris X, Nanterre, France

³UMR1202 INRA/University Bordeaux I BioGeCo, Bordeaux, France

*email: jean-jacques.daudin@agroparistech.fr

SUMMARY. The mixture model is a method of choice for modeling heterogeneous random graphs, because it contains most of the known structures of heterogeneity: hubs, hierarchical structures, or community structure. One of the weaknesses of mixture models on random graphs is that, at the present time, there is no computationally feasible estimation method that is completely satisfying from a theoretical point of view. Moreover, mixture models assume that each vertex pertains to one group, so there is no place for vertices being at intermediate positions. The model proposed in this article is a grade of membership model for heterogeneous random graphs, which assumes that each vertex is a mixture of extremal hypothetical vertices. The connectivity properties of each vertex are deduced from those of the extreme vertices. In this new model, the vector of weights of each vertex are fixed continuous parameters. A model with a vector of parameters for each vertex is tractable because the number of observations is proportional to the square of the number of vertices of the network. The estimation of the parameters is given by the maximum likelihood procedure. The model is used to elucidate some of the processes shaping the heterogeneous structure of a well-resolved network of host/parasite interactions.

KEY WORDS: Ecological network; Grade of membership model; Heterogeneous random graph; Maximum likelihood; Mixture model.

1. Introduction

Complex networks are extensively studied in different domains such as social sciences and biology. The network representation of the data is graphically attractive, but there is clearly a need for a synthetic model, giving an enlightening representation of complex networks. Statistical methods have been developed for analyzing complex data such as networks in a way that could reveal underlying data patterns through some form of classification.

There are two ways of producing a synthetic representation of such data: multidimensional scaling where a position in a metric space is assigned to each vertex, and unsupervised classification of the vertices using a mixture model.

The first approach is well described in Hoff, Raftery, and Handcock (2002). Hoff (2005) develops a more general model including additional information on vertices. The recent development of the random dot product graphs (RDPG) (Marchette and Priebe, 2008) follows the same approach, with a special focus put on the probabilistic properties of such models (degree distribution, clustering coefficient, giant component) (Young and Scheinerman, 2007).

Unsupervised classification of the vertices of networks is a rapidly developing area with many applications in social and biological sciences. The underlying idea is that common connectivity behavior shared by several vertices leads to their grouping in one *meta-vertex*, without losing too much information. Then, the initial complex network can be reduced to a simpler *meta-network*, with few *meta-vertices* connected by

few *meta-edges*. Picard et al. (2009) show applications of this idea to biological networks and Nowicki and Snijders (2001) and Handcock, Raftery, and Tantrum (2007) to social networks.

The literature about classification of vertices can be divided into two classes:

1. Usual mixture model using discrete latent variables giving the assignment of each vertex to a group, where each vertex is supposed to pertain to only one group. Nowicki and Snijders (2001) were among the first to propose what they called a stochastic block structure model because their model was on the line of an older nonstochastic block structure model largely developed in social science. Their estimation method is made through Bayesian Markov chain Monte Carlo (MCMC) algorithms for networks with less than 200 vertices. Daudin, Picard, and Robin (2008) have given more insight on the same model, the degree distribution and the clustering coefficient, and used a variational method for estimating the parameters (*Mixnet*, 2009 package). The variational method allows us to deal with several thousand vertices. Using a different approach, Handcock et al. (2007) assigned a position in a metric space to each vertex and then used a Gaussian mixture model on these positions to cluster the vertices.
2. Individual mixture model, where each vertex pertains partially to several groups, so the mixture is at the individual level and not at the population level, as is the case

for usual mixture models. This class of model has been developed in social science for usual multivariate data, under the name of grade of membership (see Manton, Woodbury, and Tolley, 1994; Erosheva, 2005). The idea is that there are hypothetical extreme profiles and that each sample unit is a mixture of these extreme profiles and inherits their properties through a weighted mean. This idea has been developed under the name of mixed-membership model, see Erosheva, Fienberg, and Lafferty (2004) and Airoldi et al. (2008) for networks.

The proposed model is on the line of the second class. It is similar to the Airoldi et al. (2008) mixed-membership model, but it is expressed in a much simpler form, leaving aside a huge number of random latent variables. A model with a vector of parameters for each vertex is tractable because the number of observations is proportional to the square of the number of vertices of the network.

The estimation step for mixture models for networks is a difficult task. Maximum likelihood procedure is generally not possible, due to the huge dimension of the space where the latent discrete variables reside. Daudin et al. (2008) and Airoldi et al. (2008) use variational methods and Nowicki and Snijders (2001) and Handcock et al. (2007) use MCMC. The statistical properties of variational estimates are not well known. They maximize a pseudolikelihood and are, by definition, inferior to maximum likelihood estimates. MCMC is highly computationally intensive and their mixing properties for high-dimensional discrete variables are doubtful for large networks. Conversely, a great advantage of the proposed model is that it allows us to obtain standard maximum likelihood with a quick and robust algorithm. We restrict our interest to the case of pure relational information between vertices, putting aside any additional information on vertices. The intensity of relation between vertices may be continuous or binary. In this article we deal with binary variables. Extension to a more general case is possible, but this is not done in this article.

The most salient characteristic of the proposed model is that it is based on extremal hypothetical vertices. Therefore we will call it extremal vertices model for random graph (EVMRG). We define the EVMRG model in Section 2. In Section 3, we give a maximum-likelihood estimation algorithm.

In Section 4, we use the EVMRG model to synthesize the heterogeneity of an ecological network, i.e., a network having species as vertices and interspecific interactions as edges. Ecological networks have long fascinated biologists because of the diversity and the complexity of the interactions between species. In the *Origin of Species* (Darwin, 1869), Charles Darwin wrote: *It is interesting to contemplate a tangled bank, clothed with many plants of many kinds, with birds singing on the bushes, with various insects flitting about, and with worms crawling through the damp earth, and to reflect that these elaborately constructed forms, so different from each other, and dependent on each other in so complex a manner, have all been produced by laws acting around us.* Comparative analyses of ecological networks have highlighted some invariant topological properties, confirming that there are common laws governing the structure of apparently diverse species assemblages. Uncovering these laws is a crucial challenge for biologists be-

cause it would allow important advances in conservation and environmental management.

Modularity is a prevalent topological feature in large ecological networks (i.e., >150 species). Modules, also named compartments in the ecological literature, are recognizable subsets of interacting species, with species more likely to be linked within than across subsets (Lewinsohn et al., 2006). On the ecological timescale, modules may arise through spatial or temporal segregation of the species. Species occurring in the same place and at the same time are more likely to fall into the same module, because they have a higher probability of interacting with each other than with species occurring elsewhere or at another time. However, modularity may also reflect more ancient events, such as phylogenetic splits.

Various methods have been used for detecting modularity in ecological networks (e.g., correspondence analysis [Lewinsohn et al., 2006]; edge betweenness algorithm [Vacher, Piou, and Desprez-Loustau, 2008a]; simulated annealing algorithm [Olesen et al., 2007]). Most often, each species is assigned to one module (only) and the network is finally represented as a set of nonoverlapping modules. Such simplification of the network structure is an issue because some species may actually not belong to any module because they are loosely linked to all the other species of the network, or may belong to several modules. For instance, species with large spatial or temporal distributions, or species with complex life-cycles going through very different habitats during their lives, are likely to interact with species belonging to different modules. The misclassified species may obscure the common or complementary features of the species belonging to a module and therefore hinder our understanding of the processes shaping species webs.

Grade of membership models present the advantage of allowing the species to have intermediate positions in the simplified representation of the network. To our knowledge, they have never been used for synthesizing the heterogeneity of ecological networks. In this study, we used the EVMRG model to synthesize the heterogeneity of a well-resolved interaction network between forest tree species and parasitic fungal species, which was shown to be modular in a previous study (Vacher, Piou, et al., 2008). Then we searched for the factors governing the position of the species in the model. A wide range of potential factors was investigated, including the phylogenetic history of the species, their life-history strategy, their introduction status, and the intensity with which they were sampled.

2. Model EVMRG

2.1 Model

Vertices. Consider a graph with n vertices, labeled in $\{1, \dots, n\}$. The model is based on Q hypothetical unobserved extreme vertices.

Each vertex i is the weighted mean of Q extreme hypothetical vertices (EHV), with weights given by $Z_i = (z_{i1}, \dots, z_{iQ})$, with $z_{iq} \geq 0$ and $\sum_q z_{iq} = 1$. Q is assumed to be a fixed constant with $Q \ll n$. Q will be determined as part of a model-selection problem in Section 3.3. The Q extreme vertices are put at the end of the canonical unit vectors $(1, 0 \dots 0)$, $(0, 1, 0 \dots 0) \dots (0 \dots 0, 1)$ in \mathbf{R}^Q in an arbitrary order. The set of vertices $\{1, \dots, n\}$ is contained in the

simplex $S_Q = \{x \in [0, 1]^Q, \sum_{q=1}^Q x_q = 1\}$, so that the EHV are extreme points of S_Q . Each EHV is supposed to be typical of the group of vertices that are near it in S_Q , with more extremal connectivity properties than its neighboring real vertices.

Edges. Each edge from a vertex i to a vertex j is associated to a binary random variable X_{ij} following a Bernoulli distribution with probability P_{ij} . The probability that there is an edge from EHV q to EHV l is equal to a_{ql} . The connectivity properties of each vertex i are a mixture of the connectivity properties of the EHV so that P_{ij} can be expressed using the weights z_{iq} and z_{jl} and the connectivity matrix A between the EHVs:

$$P_{ij} = \sum_{q,l=1,Q} z_{iq} a_{ql} z_{jl}$$

which gives the matrix relation

$$P = ZAZ'$$

with

- P the (n, n) matrix containing the p_{ij} ,
- Z the (n, Q) matrix containing the z_{iq} and Z' the transpose of Z , $Z \in S_Q^n$,
- and $A \in [0, 1]^{Q^2}$, the (Q, Q) matrix containing the a_{ql} , the connectivity matrix between the EHVs.

The random variables X_{ij} are assumed to be independent. Let X be the (n, n) matrix containing the random variables X_{ij} . Finally the model is summarized by

$$X \sim \mathbf{B}(ZAZ') \quad (1)$$

where \mathbf{B} denotes the Bernoulli distribution, $Z \in S_Q^n$, and $A \in [0, 1]^{Q^2}$.

The parameters of the model are A and Z . This model may be classified in the set of the semiparametric statistical models, for each individual (vertex) has its own set of parameters (z_{i1}, \dots, z_{iQ}) . Using statistical models, it is generally impossible to estimate as many parameters as the number of individuals. Moreover there are $Q^2 + n(Q - 1)$ parameters, so this number approaches infinity with n . However, the number of observations contained in X is not proportional to n but to n^2 , so the ratio of the number of parameters with the number of

observations approaches 0 when $n \rightarrow \infty$. In practice, for each vertex i , there are n data, (x_{i1}, \dots, x_{in}) , available to estimate the $Q - 1$ linearly independent parameters contained in the vector (z_{i1}, \dots, z_{iQ}) .

We can choose whether the graph is directed or undirected by leaving the X_{ij} loose or setting $X_{ij} = X_{ji}$ for all i, j . If the graph is directed A contains Q^2 parameters. If the graph is undirected, A is symmetric and contains $\frac{Q(Q+1)}{2}$ parameters. Note that we assume in the following that there is no self-loop ($X_{ii} = 0$, for $i = 1, n$).

2.2 Relation between EVMRG and Other Models

Several models have been proposed with a functional form similar to $X \sim \mathbf{B}(ZAZ')$: the mixture model for random graphs, the RDPG, and the mixed membership stochastic blockmodel (MMB). Table 1 summarizes the functional definition of the different models. Note that the model proposed by Hoff et al. (2002) and Handcock et al. (2007) do not have this functional form and is not included in this comparison.

2.2.1 Relation between EVMRG and Mixture Model. In a mixture model for random graphs (Nowicki and Snijders, 2001; Daudin et al., 2008), the variables Z are random and are equal to 0 or 1. In the EVMRG model the variables Z are fixed parameters, and take their values in the simplex S_Q^n . In a mixture model, each vertex is assumed to pertain to only one group. The mixture model is a mixture of populations of pure vertices. In the EVMRG model, each vertex is a compound of EHV, so the mixture is at the individual level. However, there are two practical applications of the two models:

1. The clustering of the items, i.e., the classification of each item in a group. The key element is $E(Z/X = x)$ in the mixture model and directly Z for EVMRG. Note that $E(Z/X = x)$ in the mixture model, and Z in EVMRG, take their values in the same set S_Q^n .
2. The connectivity matrix A is the key element for the description and interpretation of groups in the two models, see Daudin et al. (2008). In the mixture model, A is the mean connectivity matrix in the sense that the probability of connection is the weighted mean of the connections between the vertices. In the EVMRG, however, A represents an extreme connectivity matrix. As a result A is more contrasted in EVMRG than in the mixture model.

2.2.2 Relation between EVMRG and RDPG. The multidimensional scaling method, applied to the similarity matrix P ,

Table 1

Summary of the models $X \sim \mathbf{B}(P = f(ZAZ'))$ or $X \sim \mathbf{B}(P = f(UAV'))$, with VEM = variational EM, OLS = ordinary least squares, $S_Q = \{x \in [0, 1]^Q, \sum_{q=1}^Q x_q = 1\}$, $\mathbf{B}(\cdot)$ is the Bernoulli probability distribution function, and $\mathbf{M}(\cdot)$ is the multinomial probability distribution function with one trial and Q classes

Model	f	Z	A	P	Estimation method
EVMRG	Id	$Z \in S_Q^n$	$A \in [0, 1]^{Q^2}$	$P = f(ZAZ')$	ML
Mixture model	Id	$Z \in S_Q^n$ and binary	$A \in [0, 1]^{Q^2}$	$P = f(ZAZ')$	VEM /MCMC
RDPG	$f : \mathbf{R} \rightarrow [0, 1]$, monotone	$U, V \in \mathbf{R}_Q^n$	$A = Id$	$P = f(UAV')$	OLS
DEDICOM	Id	$Z \in \mathbf{R}_Q^n$, $Z'Z = I$	$A \in \mathbf{R}^{Q^2}$	$P = f(ZAZ')$	OLS
MMB	$f(x) = \rho x$, $\rho \in]0, 1]$	$Z \in S_Q^n$, $U_{ij} \sim \mathbf{M}(Z_i)$, $V_{ji} \sim \mathbf{M}(Z_j)$	$A \in [0, 1]^{Q^2}$	$P_{ij} = f(U'_{ij} AV_{ji})$	VEM

consists in positioning each vertex in a metric space so that the similarity between vertices is approximatively kept. The underlying model is $P = TT'$, where the (n, k) -matrix T contains the coordinates of the vertices in a k -dimensional metric space. The naive multidimensional scaling method is not well suited for modeling P, with two major drawbacks: TT' does not lie in $[0, 1]^{n^2}$ if $T \in \mathbf{R}^k$ and TT' is symmetric so it is not suited for the modeling of directed graphs.

The RDPG defined in Marchette and Priebe (2008) is

$$P_{ij} = f(t'_i t_j) \text{ with } t_i \in \mathbf{R}^k \text{ and } f(x) \in [0, 1].$$

f is a simple threshold in Marchette and Priebe (2008): $f(x) = 0$ if $x < 0$, $f(x) = x$ if $0 \leq x \leq 1$ and $f(x) = 1$ if $x > 1$. Young and Scheinerman (2007) propose to constrain T to lie in $\frac{1}{\sqrt{k}}[0, 1]^k$.

To get around the second drawback, the RDPG model is extended with two vectors for each vertex, an in-vector V and an out-vector U , so the model becomes $P_{ij} = f(u'_i \cdot v_j)$.

Another way to get around the symmetry of P , is the DEcomposition into VIrectional COMponents, called DEDICOM, which was proposed by Harshman (1978) and well described in Trendafilov (2002). This model uses only one vector for each vertex but inserts a nonsymmetric (k, k) -matrix A in the dot product. The model is

$$X = TAT' + E$$

the matrix T is constrained by $T'T = I$ and T and A are obtained by minimizing $\|X - TAT'\|^2$. Several algorithms have been proposed to achieve this task (see Kiers et al., 2002).

2.2.3 Relation between EVMRG and MMB. The MMB (see Airoldi et al., 2008) is similar to EVMRG, with a more complex setting, which is not easy to understand:

- The lines of Z (i.e., the random vectors of weights $Z_i = (z_{i1} \dots z_{iQ})$) are assumed to be identically and independently distributed along a Dirichlet distribution with parameter α
- For each pair of vertices (i, j) in this order, two multinomial random variables $U_{i \rightarrow j}$ and $V_{i \leftarrow j}$ are generated with respective probabilities Z_i and Z_j
- A is a (Q, Q) matrix $\in [0, 1]^{Q^2}$
- ρ is a sparsity parameter
- X_{ij} is a Bernoulli random variable with probability $\rho U'_{i \rightarrow j} A V_{i \leftarrow j}$

The EVMRG is essentially a marginalized version of the MMB model: the MMB model assumes a hierarchical structure: $X|U, V, A$ and $U, V|Z$, whereas the EVMRG integrates U, V from this structure to obtain $X|A, Z$. Moreover the EVMRG model does not need the ad hoc sparsity parameter ρ .

2.3 Model Identifiability

As defined so far, the model is not identifiable. Let P be a known matrix and assume that A and Z exist so that $P = ZAZ'$. It is generally possible to find other sets of parameters \tilde{A} and \tilde{Z} so that $P = \tilde{Z}\tilde{A}\tilde{Z}'$. Let H be a (Q, Q) matrix with the following properties (called H -properties):

- (1) H^{-1} exists
- (2) $H\mathbf{1}_Q = \mathbf{1}_Q$, with $\mathbf{1}_Q = (1 \dots 1)'$, made of Q ones

- (3) $\tilde{Z} = ZH \geq 0$
- (4) $\tilde{A} = H^{-1}AH'^{-1} \in [0, 1]^{Q^2}$

Then we have:

- $\tilde{Z}\tilde{A}\tilde{Z}' = ZHH^{-1}AH'^{-1}H'Z' = P$
- $\tilde{Z}\mathbf{1}_Q = ZH\mathbf{1}_Q = Z\mathbf{1}_Q = \mathbf{1}_Q$ so $\tilde{Z} \in S_Q^n$ by condition 3
- $\tilde{A} \in [0, 1]^{Q^2}$ by condition 4

so (\tilde{A}, \tilde{Z}) and (A, Z) are equivalent admissible sets of parameters.

The existence of such H -matrix is proved in Web Appendix A, with a toy example for illustration.

We propose to choose Z , which maximizes $Tr(ZZ')$ among the equivalent versions of (A, Z) . The choice is motivated by two reasons: This constraint implies unicity of (Z, A) provided that $n \gg Q$ and the n vertices are different. Moreover the EHV should not be too far from real vertices to confer upon them some reality. This closeness between EHV and some vertices is naturally provided by the maximization of $Tr(ZZ')$.

Finally the model is now:

$$X \sim \mathbf{B}(Z'AZ) \quad (2)$$

where \mathbf{B} denotes the Bernoulli distribution, $Z \in S_Q^n$, $A \in [0, 1]^{Q^2}$, and $Tr(Z'Z)$ is maximum.

3. Parameter Estimation

The log likelihood is

$$\begin{aligned} L = & \sum_{i \neq j} x_{ij} \log \left(\sum_{q,l=1,Q} z_{iq} a_{ql} z_{jl} \right) \\ & + (1 - x_{ij}) \log \left(1 - \sum_{q,l=1,Q} z_{iq} a_{ql} z_{jl} \right) \end{aligned} \quad (3)$$

and the constraints on the parameters are

$$\begin{aligned} A & \in [0, 1]^{Q^2} \\ Z & \in S_Q^n. \end{aligned}$$

Note that the set of admissible solutions, $[0, 1]^{Q^2} \times S$, is a convex polyhedron.

3.1 Log-Likelihood Derivatives

After some algebraic manipulations we obtain

$$\frac{\partial L}{\partial Z} = RZA' + R'ZA$$

with R a (n, n) matrix with $r_{ij} = \frac{x_{ij} - p_{ij}}{p_{ij}(1-p_{ij})}$, and

$$\frac{\partial L}{\partial A} = Z'RZ.$$

3.2 Estimation Algorithm

3.2.1 Algorithm. The constraints on the parameters are linear, but the log likelihood is not linear.

Let $A^{(k)}$ and $Z^{(k)}$ be the parameter estimates at step k , $P^{(k)} = Z^{(k)}A^{(k)}Z^{(k)'} = R^{(k)}$ a (n, n) matrix with $r_{ij}^{(k)} = \frac{x_{ij} - p_{ij}^{(k)}}{p_{ij}^{(k)}(1-p_{ij}^{(k)})}$.

The linear approximation of the log likelihood (3) at point $(A^{(k)}, Z^{(k)})$ is

$$\begin{aligned} L(A, Z) \approx & L(A^{(k)}, Z^{(k)}) + Tr \left[(A - A^{(k)})' \frac{\partial L}{\partial A} (A^{(k)}, Z^{(k)}) \right] \\ & + Tr \left[(Z - Z^{(k)})' \frac{\partial L}{\partial Z} (A^{(k)}, Z^{(k)}) \right]. \end{aligned}$$

The algorithm is the following:

- Find initializing values $(A^{(0)}, Z^{(0)})$
- At step $(k+1)$ use a linear programming algorithm to maximize the function in (A, Z) :

$$\begin{aligned} f_k(A, Z) = & Tr \left[A' Z^{(k)'} R^{(k)} Z^{(k)} \right] \\ & + Tr \left[Z' (R^{(k)} Z^{(k)} A^{(k)'} + R^{(k)'} Z^{(k)} A^{(k)}) \right] \end{aligned}$$

under the constraints: $A \in [0, 1]^{n^2}$ and $Z \in S_Q^n$.

Let Z^{LP_k} be the solution of the previous linear program. Compute $L(A, Z)$ on regularly spaced points along the line $(A^{(k)}, Z^{(k)}) \rightarrow (A^{LP_k}, Z^{LP_k})$ and keep the best one, $(A^{(k+1)}, Z^{(k+1)})$. A further improvement around this point and along the same line is obtained by dichotomy. Then, go to step $k+2$ if the following stopping rule is not true.

$$|L(A^{(k+1)}, Z^{(k+1)}) - L(A^{(k)}, Z^{(k)})| < \alpha.$$

3.2.2 Initialization. The algorithm is convergent because the likelihood is increased at each step. However, it may converge to a local maximum depending on the initialization. We use several random initializations based on k -means and select the best starting point. Another possibility would be to use the algorithm described by Kiers et al. (2002) for DEDICOM.

3.2.3 Assessment of the Identification of the Model. The model is not identifiable as it stands (see Section 2.3). In practice we have not seen any problem coming from the lack of identifiability when using the above algorithm. After convergence, we obtain a unique instance of the equivalent class of parameters (A, Z) by maximizing $Tr(Z'Z)$ under the constraint that $ZAZ' = Z^{(k)} A^{(k)} Z^{(k)}$, with k the iteration number at convergence.

3.3 Choice of the Number of Groups

Several criteria have been proposed for choosing the number of groups in finite mixture models, such as Akaike information criteria (AIC), Bayesian information criteria (BIC), or integrated completed likelihood (ICL), see McLachlan and Peel (2000) and Biernacki, Celeux, and Govaert (2000). From a theoretical point of view, penalized likelihood criteria are asymptotically consistent under some conditions satisfied by BIC but not by AIC, see Gassiat (2002). From a practical point of view and for moderate sample sizes, AIC has a known tendency to overestimate the number of groups for Gaussian mixtures but gives correct results for latent class models. Conversely BIC give good results for Gaussian mixtures but underestimates the number of groups for latent class models. AIC is equal to minus two times the log likelihood plus two times the number of estimated parameters, and BIC has a similar definition with the logarithm of the number of observations in place of the coefficient 2.

- For directed networks:

$$\begin{aligned} AIC(Q) = & -2L(\hat{A}_Q, \hat{Z}_Q) + 2(Q^2 + n(Q-1)) \\ BIC(Q) = & -2L(\hat{A}_Q, \hat{Z}_Q) + Q^2 \log(n(n-1)) \\ & + n(Q-1) \log(n). \end{aligned}$$

- For indirected networks:

$$\begin{aligned} AIC(Q) = & -2L(\hat{A}_Q, \hat{Z}_Q) + 2(Q(Q+1)/2 + n(Q-1)) \\ BIC(Q) = & -2L(\hat{A}_Q, \hat{Z}_Q) + Q(Q+1)/2 \log(n(n-1)/2) \\ & + n(Q-1) \log(n). \end{aligned}$$

(\hat{A}_Q, \hat{Z}_Q) are the maximum likelihood estimates of (A, Z) for Q groups. Some more theoretical work is needed to study the asymptotic properties of these criteria in the context of EVMRG, for the number of parameters tends to infinity with the number of vertices, which is a nonstandard framework. In this article, we use these criteria from a practical point of view and without any theoretical background. We have made some simulations to see if these criteria are able to recover the true number Q . For low and moderate sample sizes AIC has given good results, better than BIC which underestimated Q . Therefore we use AIC in the study of the following example.

4. Example

4.1 Data

The ecological network considered in this study consisted of 543 interactions among 51 forest tree taxa (all but 6 being true species or groups of cultivars belonging to the same genetic continuum) and 154 parasitic fungal species. The network is composed of 205 vertices and 543 edges. It is a bipartite graph because tree-fungus interactions are the only possible ones. All the observations of tree-fungus interactions originated from the database of the French governmental organization in charge of forest health monitoring (the *Département Santé des Forêts (DSF)*) for the 1972–2005 period. The methods used for data collection have been described in more detail in previous analyses of the DSF database (Vacher, Piou, et al., 2008; Vacher, Vile, et al., 2008). We have rechecked fungal species names in the Index Fungorum database (www.indexfungorum.org) since our initial analyses: 17 species names were updated, and three of the previously used species names were found to be synonymous. The fusion of synonymous species accounts for the smaller number of fungal species in this study than in our previous study (154 versus 157 in the previous study Vacher, Piou, et al., 2008) and the slightly smaller number of interactions (543 versus 547).

We characterized each tree species by phylum (Magnoliophyta or Coniferophyta) and introduction status (alien or native). An estimate of the area covered by each tree species was also available (*Inventaire Forestier National*, 2000 census report, <http://www.ifn.fr/spip>). An estimate of the total number of times each tree species had been encountered and examined by foresters during their daily work was also available from the DSF database. This variable is called “sampling intensity” and is positively correlated with area, because foresters encounter abundant tree species more frequently than rare species during their daily work (Vacher, Piou,

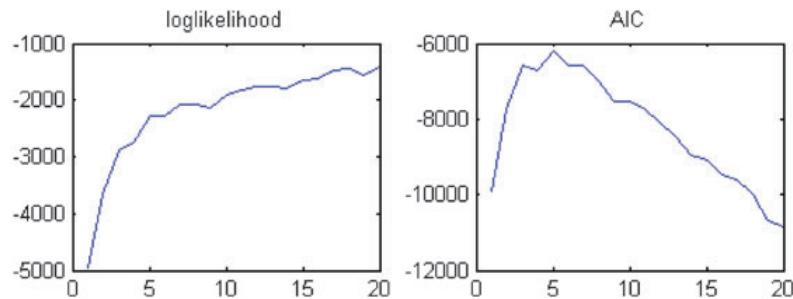


Figure 1. Evolution of the log likelihood and the AIC criteria as a function of the number of EHV, x -axis: number of extremal vertices, left side y -axis: log likelihood, right side y -axis: AIC. This figure appears in color in the electronic version of this article.

et al., 2008). The definition of tree species as aliens or species native to France was not an easy task, because the composition of European forests has been profoundly modified by human activities (Petit et al., 2004). In this study, we considered a tree species to be alien if it was introduced into France after the beginning of the modern era (which we define as the discovery of the New World by Columbus). Such recently introduced species are known as *neophytes*.

Each fungal species was characterized by phylum (Ascomycota or Basidiomycota), introduction status (alien or native), and life-history strategy. As suggested by Garcia-Guzman and Morales (2007), life-history strategies were described in terms of the parasitic lifestyle (biotrophic versus necrotrophic) and the plant organs and tissues attacked: (1) strict foliar necrotrophic parasites, (2) canker agents, (3) stem decay fungi, (4) obligate biotrophic parasites, (5) root decay fungi, (6) other foliar and twig necrotrophic parasites, (7) stem blue stain agents, (8) parasites of fine roots, (9) wilting agents, and (10) other root fungi. The first five strategies accounted for 87% of the fungal species. As for the tree species, it was not a straightforward task determining which fungal species were aliens (Desprez-Loustau, 2009). In this study, we considered a fungal species to be alien if there was documentary evidence that this species was first described in France after 1850 and good evidence that it was introduced from elsewhere.

4.2 Main Results

The AIC criteria (Figure 1) indicated that the optimal number of EHV for the tree-fungus network was 5. The connectivity matrix between the EHV (Table 2) was symmetric because the network was indirected. It indicated that one of the EHV (hereafter called FT0) had no connection with all the other EHV whereas the four remaining EHV (here-

after called F1, F2, T1, and T2) formed two pairs of highly connected vertices. The matrix of weights \hat{Z} indicated that each real vertex was a mixture of three EHV only. All the real vertices representing tree species were a mixture of FT0, T1, and T2 whereas all the real vertices representing fungal species were a mixture of FT0, F1, and F2. Therefore, T1 and T2 were two virtual tree species. They were highly connected with the virtual fungal species F1 and F2, respectively. In the data X , the zeros between trees (respectively between fungal species) are structural ones, but this information is not included a priori in the model. This bipartite network structure is recovered in the results: the EHV are tree-EHV (T1 and T2) or fungi-EHV (F1 and F2) (except the isolated EHV FT0 with no connection with any other EHV), and the probability of connection between T1 and T2 is null and the same is true for F1 and F2.

In the following the positions of the vertices in the triangular representations are “analyzed” graphically by annotating these classifications with several species descriptors (phylum, life-history strategy, introduction status, number of interactions). There are two possibilities to formalize these graphical analyses: the first one consists in using a linear model with the values of Z as (multivariate) response and the species descriptors as independent variables. The second one would be to extend the EVMRG model by including covariates giving some information about each vertex. The first type of analysis has been done and confirms the graphical results (data not shown), and we are working on the model extension approach.

The projection of phylogenetic data in the triangular representation of tree species showed that the tree species belonging to the Magnoliophyta (angiosperms) and the tree species belonging to the Coniferophyta (gymnosperms) had very different connection profiles (Figure 2). T1 was close to seven gymnosperm species which are highly represented in the French forests (*Abies alba*, *Abies grandis*, *Picea excelsa*, *Pinus laricio*, *Pinus pinaster*, *Pinus sylvestris*, and *Pseudotsuga menziesii*). All the other gymnosperm species were located on the line joining T1 and FT0, suggesting that they all had a subset of the interactions realized by the seven tree species close to T1. This result confirmed the nested pattern of interactions found in a previous study (Vacher, Piou, et al., 2008). T2 was close to six tree taxa belonging to the Magnoliophyta, which are also dominant in the French forests (large maples, cultivated poplars, beech [*Fagus*

Table 2
Connectivity matrix A between the five extremal vertices

	FT0	T1	T2	F1	F2
FT0	0	0	0	0	0
T1	0	0	0	0.996	0
T2	0	0	0	0	0.985
F1	0	0.996	0	0	0
F2	0	0	0.985	0	0

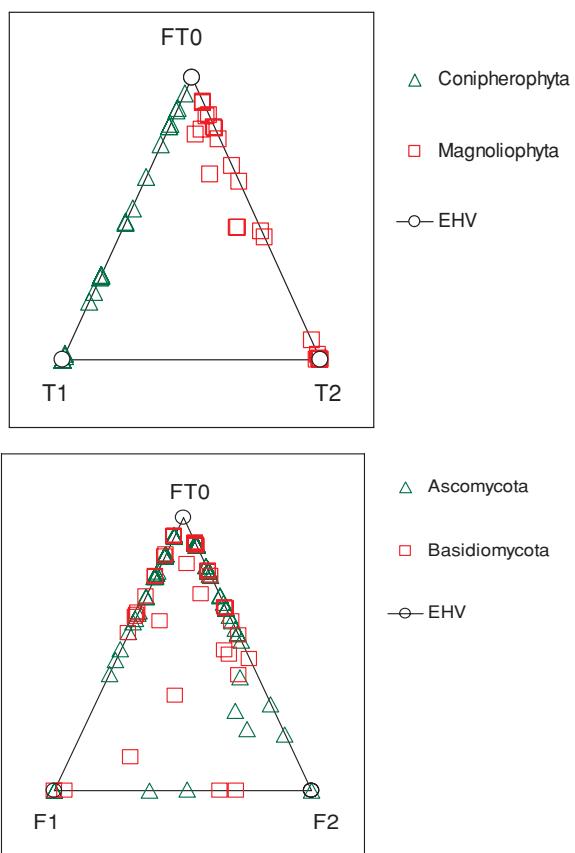


Figure 2. Triangular representations of tree species (top) and fungal species (bottom) as a function of their phylogenetic origin. This figure appears in color in the electronic version of this article.

silvatica], and three species of oaks [*Quercus petraea*, *Q. pubescens* and *Q. rubra*]. Five species belonging to the Magnoliophyta were not located on the line joining T2 and FT0, suggesting that the associations of angiosperms with parasitic fungi were slightly more diverse than those of gymnosperm species. This result is consistent with other studies (Vacher, Piou, et al., 2008) and may be accounted for by the wider distributional range of angiosperm species. Among the five species mentioned above, three belonged to the Rosaceae family (*Prunus avium*, *Sorbus aria*, *Sorbus torminalis*). It is noteworthy that these three species were classified with gymnosperm species in a previous analysis of the tree–fungus network in which each vertex was assumed to pertain to one group (Vacher, Piou, et al., 2008). The approach used here revealed that the connection profiles of these three species were mixtures between the typical profile of angiosperms (T2) and the typical profile of gymnosperms (T1), but were actually closer to the typical profile of angiosperms.

In contrast, the projection of phylogenetic data in the triangular representation of fungal species (Figure 2) showed

that the species belonging to the Ascomycota and the Basidiomycota had similar connection profiles. F1 was close to one generalist fungal species belonging to the Basidiomycota (*Armillaria ostoyae*) and two generalist fungal species belonging to the Ascomycota (*Sphaeropsis sapinea* and *Sydowia polyspora*). According to the connectivity matrix, these parasitic fungal species were highly specialized on gymnosperms. The species located on the line joining F1 and FT0 also belonged both to the Ascomycota and the Basidiomycota. F2 was close to one species only, which belonged to the Ascomycota (*Botryosphaeria stevensii*). However, the species located on the line joining F2 and FT0, which were specialized on angiosperm species according to the connectivity matrix, belonged both to the Ascomycota and the Basidiomycota. Therefore, the phylogenetic history of fungal species did not account for their specialization on gymnosperms or angiosperms.

4.3 Discussion and Conclusions about the Example

Our results confirmed that the heterogeneous structure of the network mostly results from the deep evolutionary history of seed plants (Vacher, Piou, et al., 2008). Angiosperm species and gymnosperm species had very contrasted interaction profiles, except when they covered low areas and were consequently not intensively monitored for their fungal diseases (see Web Appendix B1). In contrast, parasitic fungal species belonging to the Ascomycota and the Basidiomycota had very similar interaction profiles. A possible explanation for this asymmetric phylogenetic signal may be that, to survive, parasitic species had no other choice than to adopt an opportunistic feeding behavior, which decreased the relationship between their phylogenetic similarity and the similarity in their interaction profiles. In contrast, the relationship between the phylogenetic similarity of tree species and the similarity in their interaction profiles may have been maintained because adaptations allowing tree species to defend against or avoid parasitic fungal species were least favored by natural selection. Our results (see Web Appendix B2) also showed that the parasitic fungal species having the most opportunistic feeding behavior (i.e., able to attack both angiosperms and gymnosperms) were mainly fungal species with high saprophytic abilities, belonging to stem or root decay fungi. Therefore our results confirmed that the ability to survive well without a host may increase the opportunities for and the likelihood of host shifts (Parker and Gilbert, 2004). Finally, our results (see Web Appendix B3) showed that alien tree species and alien fungal species were well integrated into the network. This rapid integration was unexpected for a plant–pathogen network, because selection is supposed to act continually on plants, favoring the emergence of defenses against new pathogens, and impeding the development of new interactions (Parker and Gilbert, 2004; Thompson, 2006).

Our study showed that the amount of information obtained from the EVMRG model in the case of a host–parasite network was considerable. The EVMRG model therefore appears as a good approach for synthesizing the heterogeneity of ecological networks. Applying the EVMRG model to the network of interactions between tree species and parasitic fungal species of the French forests confirmed, with a single analysis, several results obtained in previous studies (Vacher,

Vile, et al., 2008) through different analyses. It also suggested that one of the results obtained previously—the classification of three angiosperm tree species belonging to the Rosaceae family in a module containing only gymnosperm species (Vacher, Piou, et al., 2008)—is likely to be false. By allowing the species to have intermediate positions in the simplified representation of the network, the EVMRG model revealed that the interaction profiles of the three species were actually closer to that of angiosperm species. Therefore, previous discussions (Vacher, Piou, et al., 2008) concerning the surprising interaction profiles of tree species belonging to the Rosaceae family should not be given too much importance.

5. Conclusion

The mixture model is a method of choice for modeling heterogeneous random graphs because it contains most of the known structures of heterogeneity: hubs, hierarchical structures, or community structure. One of the weaknesses of mixture models on random graphs is that, at the present time, there is no computationally feasible estimation method that is completely satisfying from a theoretical point of view. The discrete nature of Z implies that one has to explore a space of dimension Q^n , a task that is highly computationally intensive. The discrete values for Z are replaced by continuous ones in the EVMRG model, which leads to an easier optimization problem and allows us to obtain the maximum-likelihood estimates with an efficient algorithm. Moreover the continuous nature of Z allows us to alleviate the assumption of pure units, pertaining only to one group. The EVMRG model is more flexible than the usual mixture model for it includes the possibility for a vertex to have intermediate connectivity properties. This model, which needs a vector of parameters for each vertex, is tractable because we have n data for each vertex. A MATLAB package called CMixnet, allowing one to analyze a network using the EVMRG model, is available at <http://www.agroparistech.fr/mia/doku.php?id=productions:logiciels>. However, some additional work is necessary to understand the behavior of the maximum-likelihood estimates of n parameters and n^2 observations when $n \rightarrow \infty$.

6. Supplementary Materials

Web Appendices A and B, referenced respectively in Sections 2.3 and 4, are available under the Paper Information link at the *Biometrics* website <http://www.biometrics.tibs.org>.

ACKNOWLEDGEMENTS

We thank the *Département Santé des Forêts (DSF)* for allowing us to use their database and the French research consortium *Interactions Biotiques dans les communautés (GDR ComEvol)* for funding.

REFERENCES

- Airoldi, E. M., Blei, D. M., Fienberg, S., and Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research* **9**, 1981–2014.
- Biernacki, C., Celeux, G., and Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, 719–725.
- Darwin, C. E. (1869). *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*, 5th edition, p. 611. London: John Murray.
- Daudin, J. J., Picard, F., and Robin, S. (2008). A mixture model for random graphs. *Statistics and Computing* **18**(2), 173–183.
- Desprez-Loustau, M. L. (2009). The alien fungi of Europe. In *Handbook of Alien Species in Europe*, W. Nentwig, P. Hulme, P. Pysek, and M. Vila (eds), vol. 3, p. 400. Berlin: Springer-Verlag.
- Erosheva, E. (2005). Comparing latent structures of the grade of membership, Rasch and latent class model. *Psychometrika* **70**(4), 619–628.
- Erosheva, E., Fienberg, S., and Lafferty, J. (2004). Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences* **101**, 5220–5227.
- Garcia-Guzman, G. and Morales, E. (2007). Life-history strategies of plant pathogens: Distribution patterns and phylogenetic analysis. *Ecology* **88**, 589–596.
- Gassiat, E. (2002). Likelihood ratio inequalities with applications to various mixtures. *Annales de l'Institut Henri Poincaré* **38**, 897–906.
- Handcock, M. S., Raftery, A. E., and Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society Series A* **54**, 301–354.
- Harshman, R. A. (1978). Model for analysis of asymmetrical relationships among N objects or stimuli. *First Joint Meeting of the Psychometric Society and the Society of Mathematical Psychology*, Hamilton, Ontario, Canada.
- Hoff, D. (2005). Bilinear mixed-effects models for dyadic data. *Journal of the American Statistical Association* **100**, 286–295.
- Hoff, D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approach to social network analysis. *Journal of the American Statistical Association* **97**, 1090–1098.
- Kiers, H. A. L., ten Berge, J. M. F., Takane, Y., and De Leeuw, J. (2002). A generalization of Takane's algorithm for DEDICOM. *Psychometrika* **55**, 151–158.
- Lewinsohn, T. M., Prado, P. I., Jordano, P., Bascompte, J., and Olesen, J. M. (2006). Structure in plant-animal interaction assemblages. *Oikos* **113**, 174–184.
- Manton, K. G., Woodbury, M. A., and Tolley, H. D. (1994). *Statistical Applications Using Fuzzy Sets*. New York: Wiley Interscience.
- Marchette, D. J. and Priebe, C. E. (2008). Predicting unobserved links in incompletely observed networks. *CSDA* **52**, 1373–1386.
- McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. New York: Wiley.
- Mixnet (2009). <http://stat.genopole.cnrs.fr/software/mixnet/>, accessed November 21, 2009.
- Nowicki, K. and Snijders, T. (2001). Estimation and prediction for stochastic block-structures. *Journal of the American Statistical Association* **96**, 1077–1087.
- Olesen, J. M., Bascompte, J., Dupont, Y. L., and Jordano, P. (2007). The modularity of pollination networks. *Proceedings of the National Academy of Sciences* **104**, 19891–19896.
- Parker, I. M. and Gilbert, G. S. (2004). The evolutionary ecology of novel plant-pathogen interactions. *Annual Review of Ecology Evolution and Systematics* **35**, 675–700.
- Petit, R. J., Bialozyt, R., Garnier-Gere, P., and Hampe, A. (2004). Ecology and genetics of tree invasions: From recent introductions to quaternary migrations. *Forest Ecology and Management* **197**, 117–137.
- Picard, F., Miele, V., Daudin, J. J., Cottret, L., and Robin, S. (2009). Deciphering the connectivity structure of biological networks using MixNet. *BMC Bioinformatics* **10**,

- <http://www.biomedcentral.com/1471-2105/10/S6/S17>, accessed June 16, 2009.
- Thompson, J. N. (2006). Mutualistic webs of species. *Science* **312**, 372–373.
- Trendafilov, N. T. (2002). GIPSCAL revisited. A projected gradient approach. *Statistics and Computing* **12**, 135–145.
- Vacher, C., Piou, D., and Desprez-Loustau, M.-L. (2008). Architecture of an antagonistic tree/fungus network: The asymmetric influence of past evolutionary history. *PLoS ONE* **3**, e1740.
- Vacher, C., Vile, D., Helion, E., Piou, D., and Desprez-Loustau, M. L. (2008). Distribution of parasitic fungal species richness: Influence of climate versus host species diversity. *Diversity and Distributions* **14**, 786–798.
- Young, S. J. and Scheinerman, E. R. (2007). Random dot product graph models for social networks. *Conf WAW*, http://dx.doi.org/10.1007/978-3-540-77004-6_11.

Received May 2009. Revised October 2009.

Accepted October 2009.

Mixed Membership Stochastic Blockmodels

Edoardo M. Airoldi*

David M. Blei

*Department of Computer Science
Princeton University
Princeton, NJ 08544, USA*

EAIROLDI@PRINCETON.EDU

BLEI@CS.PRINCETON.EDU

Stephen E. Fienberg†

*Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213, USA*

FIENBERG@STAT.CMU.EDU

Eric P. Xing

*School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA*

EPXING@CS.CMU.EDU

Editor: Tommi Jaakkola

Abstract

Consider data consisting of pairwise measurements, such as presence or absence of links between pairs of objects. These data arise, for instance, in the analysis of protein interactions and gene regulatory networks, collections of author-recipient email, and social networks. Analyzing pairwise measurements with probabilistic models requires special assumptions, since the usual independence or exchangeability assumptions no longer hold. Here we introduce a class of variance allocation models for pairwise measurements: mixed membership stochastic blockmodels. These models combine global parameters that instantiate dense patches of connectivity (blockmodel) with local parameters that instantiate node-specific variability in the connections (mixed membership). We develop a general variational inference algorithm for fast approximate posterior inference. We demonstrate the advantages of mixed membership stochastic blockmodels with applications to social networks and protein interaction networks.

Keywords: hierarchical Bayes, latent variables, mean-field approximation, statistical network analysis, social networks, protein interaction networks

1. Introduction

The problem of modeling relational information among objects, such as pairwise relations represented as graphs, arises in a number of settings in machine learning. For example, scientific literature connects papers by citations, the Web connects pages by links, and protein-protein interaction data connects proteins by physical binding records. In these settings, we often wish to infer hidden attributes of the objects from the observed measurements on pairwise properties. For example, we might want to compute a clustering of the web-pages, predict the functions of a protein, or assess

*. Also in the Lewis-Sigler Institute for Integrative Genomics. Address correspondence to 228 Carl Icahn Laboratory, Princeton University.

†. Also in the School of Computer Science.

the degree of relevance of a scientific abstract to a scholar’s query. Unlike traditional data collected from individual objects, *relational data* violate the classical independence or exchangeability assumptions made in machine learning and statistics. The observations are dependent because of the way they are connected. This interdependence suggests that a different set of assumptions is more appropriate.

There is a history of research devoted to analyzing relational data. One well-studied problem is *clustering*, grouping the objects to uncover a structure based on the observed patterns of interactions. Standard model-based clustering methods, for example, mixture models, are not immediately applicable to relational data because they assume that the objects are conditionally independent given their cluster assignments. Rather, the latent stochastic blockmodel (Wang and Wong, 1987; Snijders and Nowicki, 1997) is an adaptation of mixture modeling to relational data. In that model, each object belongs to a cluster and the relationships between objects are governed by the corresponding pair of clusters. With posterior inference, one identifies a set of latent roles which govern the objects relationships with each other. A recent extension of this model relaxed the finite-cardinality assumption on the latent clusters with a nonparametric hierarchical Bayesian model based on the Dirichlet process prior (Kemp et al., 2004, 2006; Xu et al., 2006).

The latent stochastic blockmodel suffers from a limitation that each object can only belong to one cluster, or in other words, play a single latent role. However, many relational data sets are multi-facet. For example, when a protein or a social actor interacts with different partners, different functional or social contexts may apply and thus the protein or the actor may be acting according to different latent roles they can possible play. In this paper, we relax the assumption of single-latent-role for actors, and develop a *mixed membership model* for relational data. Mixed membership models, such as latent Dirichlet allocation (Blei et al., 2003), have re-emerged in recent years as a flexible modeling tool for data where the single cluster assumption is violated by the heterogeneity within of a data point. For almost two decades, these models have been successfully applied in many domains, such as surveys (Berkman et al., 1989; Erosheva, 2002), population genetics (Pritchard et al., 2000), document analysis (Minka and Lafferty, 2002; Blei et al., 2003; Buntine and Jakulin, 2006), image processing (Li and Perona, 2005), and transcriptional regulation (Airoldi et al., 2007).

The mixed membership model associates each unit of observation with multiple clusters rather than a single cluster, via a membership probability-like vector. The concurrent membership of a data in different clusters can capture its different aspects, such as different underlying topics for words constituting each document. This is also a natural idea for relational data, where the objects can bear multiple latent roles or cluster-memberships that influence their relationships to others. As we will demonstrate, a mixed membership approach to relational data lets us describe the interaction between objects playing multiple roles. For example, some of a protein’s interactions may be governed by one function; other interactions may be governed by another function.

Existing mixed membership models are not appropriate for relational data because they assume that the data are conditionally independent given their latent membership vectors. In relational data, where each object is described by its relationships to others, we would like to assume that the ensemble of mixed membership vectors help govern the relationships of each object. The conditional independence assumptions of modern mixed membership models do not apply.

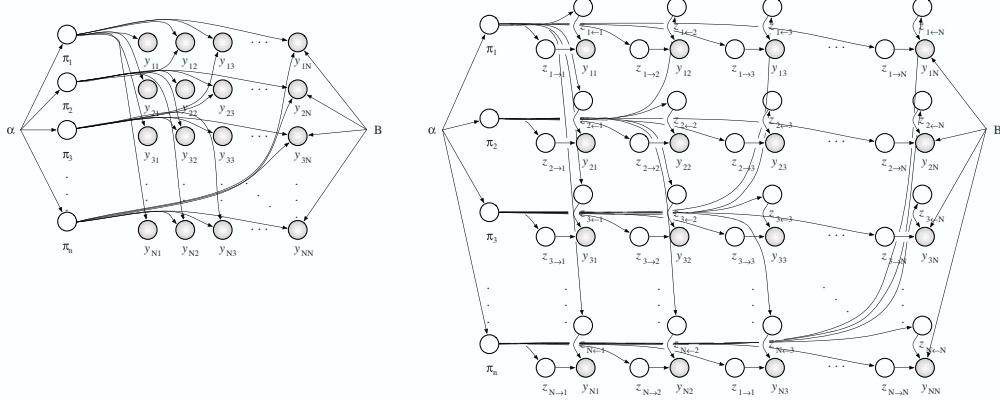


Figure 1: Two graphical model representations of the mixed membership stochastic blockmodel (MMB). Intuitively, the MMB summarized the variability of a graph with the blockmodel B and node-specific mixed membership vectors (left). In detail, a mixed membership, $\pi_n(k)$, quantifies the expected proportion of times node n instantiates the connectivity pattern of group k , according to the blockmodel. In any given interaction, $Y(n, m)$, however, node n instantiates the connectivity pattern of a single group, $z_{n \rightarrow m}(k)$. (right). We did not draw all the arrows out of the block model B for clarity; all interactions depend on it.

In this paper, we develop mixed membership models for relational data.¹ Models in this family include parameters to reduce bias due to sparsity, and can be used to analyze multiple collections of paired measurements, and collections of non-binary and multivariate paired measurements. We develop a fast nested variational inference algorithm that performs well in the relational setting and is parallelizable. We demonstrate the application of our technique to large-scale protein interaction networks and social networks. Our model captures the multiple roles that objects exhibit in interaction with others, and the relationships between those roles in determining the observed interaction matrix.

Mixed membership and the latent block structure can be recovered from relational data (Section 4.1). The application to a friendship network among students tests the model on a real data set where a well-defined latent block structure exists (Section 4.2). The application to a protein interaction network tests to what extent our model can reduce the dimensionality of the data, while revealing substantive information about the functionality of proteins that can be used to inform subsequent analyses (Section 4.3).

2. The Mixed Membership Stochastic Blockmodel

In this section, we describe the modeling assumptions if the mixed membership model of relational data. We represent observed relational data as a graph $G = (\mathcal{N}, Y)$, where $Y(p, q)$ maps pairs of nodes to values, that is, edge weights. We consider binary matrices, where $Y(p, q) \in \{0, 1\}$. The data can be thought of as a directed graph.

As a running example, we consider the monk data of Sampson (1968). Sampson measured a collection of sociometric relations among a group of monks by repeatedly asking questions such as “whom do you like?” and “whom do you dislike?” to determine asymmetric social relationships within the group. The questionnaire was repeated at four subsequent epochs. Information about these repeated, asymmetric relations was collapsed into a square binary table that encodes the directed connections between monks by Breiger et al. (1975). In analyzing this data, the goal is to determine the social structure within the monastery.

In the context of the monastery example, we assume K factors, that is, latent groups, exist in the monastery, and the observed network is generated according to distributions of group-membership for each monk and a matrix of group-group interaction strength. The per-monk distributions are specified by latent simplicial vectors. Each monk is associated with a randomly drawn vector $\vec{\pi}_i$ for monk i , where $\pi_{i,g}$ denotes the probability of monk i belonging to group g . That is, each monk can simultaneously belong to multiple groups with different degrees of affiliation strength. The probabilities of interactions between different groups are defined by a matrix of Bernoulli rates $B_{(K \times K)}$, where $B(g, h)$ represents the probability of having a link between a monk from group g and a monk from group h .

For each monk, the indicator vector $\vec{z}_{p \rightarrow q}$ denotes the group membership of monk p when he responds to survey questions about monk q and $\vec{z}_{p \leftarrow q}$ denotes the group membership of monk q when he responds to survey questions about node p .² N denotes the number of monks in the monastery, and recall that K denotes the number of distinct groups a monk can belong to.

More in general, monks can be represented by nodes in a graph, where directed (binary) edges represent positive responses to survey questions about a specific sociometric relation. In this abstract setting, the mixed membership stochastic blockmodel (MMB) posits that a graph $G = (\mathcal{N}, Y)$ is drawn from the following procedure.

- For each node $p \in \mathcal{N}$:
 - Draw a K dimensional mixed membership vector $\vec{\pi}_p \sim \text{Dirichlet}(\vec{\alpha})$.
- For each pair of nodes $(p, q) \in \mathcal{N} \times \mathcal{N}$:
 - Draw membership indicator for the initiator, $\vec{z}_{p \rightarrow q} \sim \text{Multinomial}(\vec{\pi}_p)$.
 - Draw membership indicator for the receiver, $\vec{z}_{q \rightarrow p} \sim \text{Multinomial}(\vec{\pi}_q)$.
 - Sample the value of their interaction, $Y(p, q) \sim \text{Bernoulli}(\vec{z}_{p \rightarrow q}^\top B \vec{z}_{q \rightarrow p})$.

1. In previous work we combined mixed membership and blockmodels to perform analyses of a single collection of binary, paired measurements; namely, hypothesis testing, predicting and de-noising interactions within an unsupervised learning setting (Airoldi et al., 2005).

2. An indicator vector is used to denote membership in one of the K groups. Such a membership-indicator vector is specified as a K -dimensional vector of which only one element equals to one, whose index corresponds to the group to be indicated, and all other elements equal to zero.

This process is illustrated as a graphical model in Figure 1. Note that the group membership of each node is *context dependent*. That is, each node may assume different membership when interacting to or being interacted by different peers. Statistically, each node is an admixture of group-specific interactions. The two sets of latent group indicators are denoted by $\{\vec{z}_{p \rightarrow q} : p, q \in \mathcal{N}\} =: Z_{\rightarrow}$ and $\{\vec{z}_{p \leftarrow q} : p, q \in \mathcal{N}\} =: Z_{\leftarrow}$. Also note that the pairs of group memberships that underlie interactions need not be equal; this fact is useful for characterizing asymmetric interaction networks. Equality may be enforced when modeling symmetric interactions.

Under the MMB, the joint probability of the data Y and the latent variables $\{\vec{\pi}_{1:N}, Z_{\rightarrow}, Z_{\leftarrow}\}$ can be written in the following factored form,

$$\begin{aligned} & p(Y, \vec{\pi}_{1:N}, Z_{\rightarrow}, Z_{\leftarrow} | \vec{\alpha}, B) \\ &= \prod_{p,q} P(Y(p, q) | \vec{z}_{p \rightarrow q}, \vec{z}_{p \leftarrow q}, B) P(\vec{z}_{p \rightarrow q} | \vec{\pi}_p) P(\vec{z}_{p \leftarrow q} | \vec{\pi}_q) \prod_p P(\vec{\pi}_p | \vec{\alpha}). \end{aligned} \quad (1)$$

This model generalizes to two important cases. First, multiple networks among the same actors can be generated by the same latent vectors. This may be useful, for instance, to analyze multivariate sociometric relations. Second, in the MMB the data generating distribution is a Bernoulli, but B can be a matrix that parameterizes any kind of distribution. This may be useful, for instance, to analyze collections of paired measurements, Y , that take values in an arbitrary metric space. We elaborate on this in Section 5.

2.1 Modeling Sparsity

Adjacency matrices encoding binary pairwise measurements are often sparse, that is, they contain many zeros or non-interactions. It is useful to distinguish two sources of non-interaction: they may be the result of the rarity of interactions in general, or they may be an indication that the pair of relevant blocks rarely interact. In applications to social sciences, for instance, nodes may represent people and blocks may represent social communities. It is reasonable to expect that a large portion of the non-interactions is due to limited opportunities of contact between people rather than due to deliberate choices, the structure of which the blockmodel is trying to estimate. It is useful to account for these two sources of sparsity at the model level. A good estimate of the portion of zeros that should not be explained by the blockmodel B reduces the bias of the estimates of its elements.

Thus, we introduce a sparsity parameter $\rho \in [0, 1]$ in the MMB to characterize the source of non-interaction. Instead of sampling a relation $Y(p, q)$ directly the Bernoulli with parameter specified as above, we down-weight the probability of successful interaction to $(1 - \rho) \cdot \vec{z}_{p \rightarrow q}^\top B \vec{z}_{p \leftarrow q}$. This is the result of assuming that the probability of a non-interaction comes from a mixture, $1 - \sigma_{pq} = (1 - \rho) \cdot \vec{z}_{p \rightarrow q}^\top (1 - B) \vec{z}_{p \leftarrow q} + \rho$, where the weight ρ capture the portion zeros that should not be explained by the blockmodel B . A large value of ρ will cause the interactions in the matrix to be weighted more than non-interactions, in determining plausible values for $\{\vec{\alpha}, B, \vec{\pi}_{1:N}\}$.

The sparsity parameter ρ can be estimated. Its maximum likelihood estimate provides the best data-driven guess about the proportion of zeros that the blockmodel can explain. Introducing ρ provides a strategy to rescale B , by separating zeros in the adjacency matrix into those that are likely to be due to the blockmodel and those that are not.

2.2 Summarizing and De-Noising Pairwise Measurements

It is useful to distinguish two types of data analysis that can be performed with the mixed-membership blockmodel. First, MMB can be used to summarize the data, Y , in terms of the global blockmodel, B , and the node-specific mixed memberships, Π_s . Second, MMB can be used to de-noise the data, Y , in terms of the global blockmodel, B , and interaction-specific single memberships, Z_s . In both cases the model depends on a small set of unknown constants to be estimated: α , and B . The likelihood is the same in both cases, although, the rationale for including the set of latent variables Z_s differs. When summarizing data, we could integrate out the Z_s analytically; this leads to numerical optimization of a smaller set of variational parameters, Γ_s . We choose to keep the Z_s to simplify inference. When de-noising, the Z_s are instrumental in estimating posterior expectations of each interactions individually—a network analog to the Kalman Filter. The posterior expectations of an interaction is computed as follows, in the two cases,

$$\mathbb{E} [Y(p, q) = 1] \approx \hat{\pi}_p' \hat{B} \hat{\pi}_q \quad \text{and} \quad \mathbb{E} [Y(p, q) = 1] \approx \hat{\phi}_{p \rightarrow q}' \hat{B} \hat{\phi}_{p \leftarrow q}.$$

2.3 An Illustration: Crisis in a Cloister

To illustrate the MMB, we return to an analysis of the monk data described above. Sampson (1968) surveyed 18 novice monks in a monastery and asked them to rank the other novices in terms of four sociometric relations: like/dislike, esteem, personal influence, and alignment with the monastic credo. We consider Breiger’s collation of Sampson’s data (Breiger et al., 1975). The original graph of monk-monk interaction is illustrated in Figure 2 (left).

Sampson spent several months in a monastery in New England, where novice monks were preparing to join a monastic order. Sampson’s original analysis was rooted in direct anthropological observations. He suggested the existence of tight factions among the novices: the loyal opposition (whose members joined the monastery first), the young turks (who joined later on), the outcasts (who were not accepted in the two main factions), and the waverers (who did not take sides). The events that took place during Sampson’s stay at the monastery supported his observations—members of the young turks resigned or were expelled over religious differences (John and Gregory). We shall

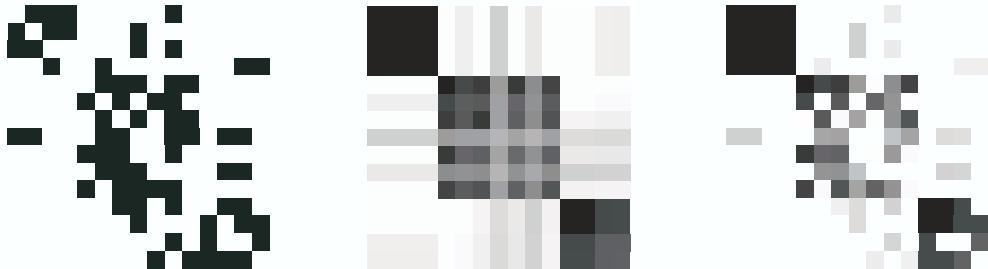


Figure 2: Original adjacency matrix of whom-do-like sociometric relations (left), relations predicted using approximate MLEs for $\hat{\pi}_{1:N}$ and B (center), and relations de-noised using the model including Z_s indicators (right).

refer to the labels assigned by Sampson to the novices in the analysis below. For more analyses, we refer to Fienberg et al. (1985), Davis and Carley (2006) and Handcock et al. (2007).

Using the algorithms presented in Section 3, we fit the monks to MMB models for different numbers of groups, providing model estimates $\{\hat{\alpha}, \hat{B}\}$ and posterior mixed membership vectors $\hat{\pi}_n$ for each monk. Here, we use the following approximation to BIC to choose the number of groups in the MMB:

$$BIC = 2 \cdot \log p(Y) \approx 2 \cdot \log p(Y|\hat{\pi}, \hat{Z}, \hat{\alpha}, \hat{B}) - |\vec{\alpha}, B| \cdot \log |Y|,$$

which selects three groups, where $|\vec{\alpha}, B|$ is the number of hyper-parameters in the model, and $|Y|$ is the number of positive relations observed (Volinsky and Raftery, 2000; Handcock et al., 2007). Note that this is the same number of groups that Sampson identified. We illustrate the fit of model fit via the predicted network in Figure 2 (Right). The three panels contrast the different resolution of the original adjacency matrix of whom-do-like sociometric relations (left panel) obtained in different uses of MMB. If the goal of the analysis if to find a parsimonious summary of the data, the amount of relational information that is captured by in $\hat{\alpha}, \hat{B}$, and $\mathbb{E}[\hat{\pi}|Y]$ leads to a coarse reconstruction of the original sociomatrix (central panel). If the goal of the analysis if to de-noising a collection of pairwise measurements, the amount of relational information that is revealed by $\hat{\alpha}, \hat{B}, \mathbb{E}[\hat{\pi}|Y]$ and $\mathbb{E}[Z_{\rightarrow}, Z_{\leftarrow}|Y]$ leads to a finer reconstruction of the original sociomatrix, Y —relations in Y are re-weighted according to how much they *make sense* to the model (right panel).

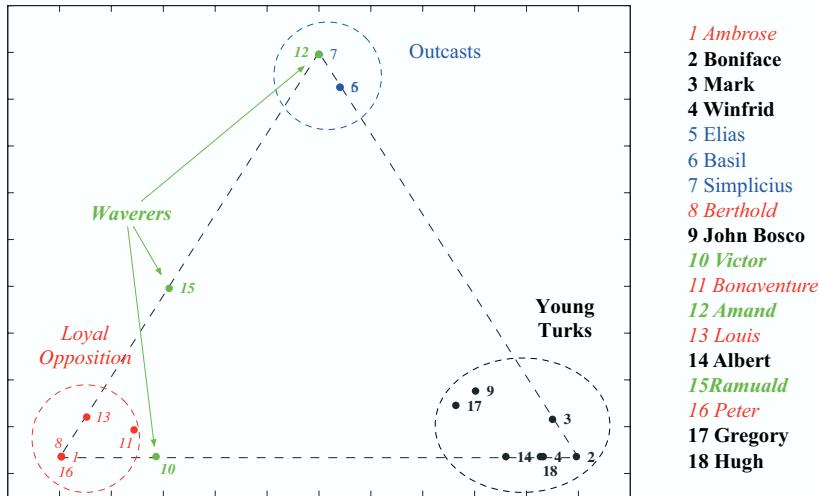
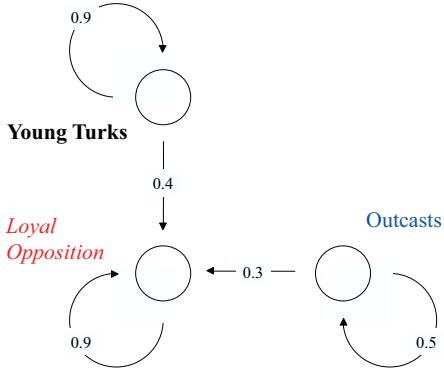


Figure 3: Posterior mixed membership vectors, $\hat{\pi}_{1:18}$, projected in the simplex. Numbered points can be mapped to monks' names using the legend on the right. The colors identify the four factions defined by Sampson's anthropological observations.

Figure 4: Estimated blockmodel in the monk data, \hat{B} .

The MMB provides interesting descriptive statistics about the actors in the observed graph. In Figure 3 we illustrate the posterior means of the mixed membership scores, $\mathbb{E}[\vec{\pi}|Y]$, for the 18 monks in the monastery. Note that the monks cluster according to Sampson's classification, with Young Turks, Loyal Opposition, and Outcasts dominating each corner respectively. We can see the central role played by John Bosco and Gregory, who exhibit relations in all three groups, as well as the uncertain affiliations of Ramuald and Victor. (Amand's uncertain affiliation, however, is not captured.) The estimated blockmodel is shown in Figure 4.

3. Parameter Estimation and Posterior Inference

Two computational problems are central to the MMB: posterior inference of the per-node mixed membership vectors and per-pair roles, and parameter estimation of the Dirichlet parameters and Bernoulli rate matrix. We derive empirical Bayes estimates of the parameters $(\vec{\alpha}, B)$, and employ a mean-field approximation scheme for posterior inference.

3.1 Posterior Inference

The posterior inference problem is to compute the posterior distribution of the latent variables given a collection of observations. The normalizing constant of the posterior distribution is the marginal probability of the data, which requires an integral over the simplicial vectors $\vec{\pi}_p$,

$$p(Y|\vec{\alpha}, B) = \int_{\Pi} \sum_{Z_s} \left(\prod_{p,q} P(Y(p,q)|\vec{z}_{p \rightarrow q}, \vec{z}_{p \leftarrow q}, B) P(\vec{z}_{p \rightarrow q}|\vec{\pi}_p) P(\vec{z}_{p \leftarrow q}|\vec{\pi}_q) \prod_p P(\vec{\pi}_p|\vec{\alpha}) \right) d\vec{\pi},$$

which is not solvable in closed form (Blei et al., 2003). A number of approximate inference algorithms for mixed membership models have appeared in recent years, including mean-field variational methods (Blei et al., 2003; Teh et al., 2007), expectation propagation (Minka and Lafferty, 2002), and Monte Carlo Markov chain sampling (MCMC) (Erosheva and Fienberg, 2005; Griffiths and Steyvers, 2004).

We appeal to variational methods (Jordan et al., 1999; Wainwright and Jordan, 2003). The main idea behind variational methods is to first posit a distribution of the latent variables with free parameters, and then fit those parameters such that the distribution is close in Kullback-Leibler divergence to the true posterior. The variational distribution is simpler than the true posterior so that the optimization problem can be approximately solved. Good reviews of variational methods can be found in Wainwright and Jordan (2003), Xing et al. (2003), Bishop et al. (2003) and Airoldi (2007).

In the MMB, we begin by bounding the log of the marginal probability of the data with Jensen's inequality,

$$\log p(Y | \alpha, B) \geq \mathbb{E}_q [\log p(Y, \vec{\pi}_{1:N}, Z_{\rightarrow}, Z_{\leftarrow} | \alpha, B)] - \mathbb{E}_q [\log q(\vec{\pi}_{1:N}, Z_{\rightarrow}, Z_{\leftarrow})].$$

We have introduced a distribution of the latent variables q that depends on a set of free parameters. We specify q as the mean-field fully-factorized family,

$$q(\vec{\pi}_{1:N}, Z_{\rightarrow}, Z_{\leftarrow} | \vec{\gamma}_{1:N}, \Phi_{\rightarrow}, \Phi_{\leftarrow}) = \prod_p q_1(\vec{\pi}_p | \vec{\gamma}_p) \prod_{p,q} \left(q_2(\vec{z}_{p \rightarrow q} | \vec{\phi}_{p \rightarrow q}) q_2(\vec{z}_{p \leftarrow q} | \vec{\phi}_{p \leftarrow q}) \right),$$

where q_1 is a Dirichlet, q_2 is a multinomial, and $\{\vec{\gamma}_{1:N}, \Phi_{\rightarrow}, \Phi_{\leftarrow}\}$ are the set of free *variational parameters* that are optimized to tighten the bound.

Tightening the bound with respect to the variational parameters is equivalent to minimizing the KL divergence between q and the true posterior. When all the nodes in the graphical model are conjugate pairs or mixtures of conjugate pairs, we can directly write down a coordinate ascent algorithm for this optimization to reach a local maximum of the bound. The updates for the variational multinomial parameters are

$$\hat{\phi}_{p \rightarrow q, g} \propto e^{\mathbb{E}_q [\log \pi_{p,g}]} \cdot \prod_h \left(B(g, h)^{Y(p,q)} \cdot (1 - B(g, h))^{1 - Y(p,q)} \right)^{\phi_{p \rightarrow q, h}} \quad (2)$$

$$\hat{\phi}_{p \leftarrow q, h} \propto e^{\mathbb{E}_q [\log \pi_{q,h}]} \cdot \prod_g \left(B(g, h)^{Y(p,q)} \cdot (1 - B(g, h))^{1 - Y(p,q)} \right)^{\phi_{p \leftarrow q, g}}, \quad (3)$$

for $g, h = 1, \dots, K$. The update for the variational Dirichlet parameters $\gamma_{p,k}$ is

$$\hat{\gamma}_{p,k} = \alpha_k + \sum_q \phi_{p \rightarrow q, k} + \sum_q \phi_{p \leftarrow q, k}, \quad (4)$$

for all nodes $p = 1, \dots, N$ and $k = 1, \dots, K$. The complete coordinate ascent algorithm is described in Figure 5.

To improve convergence, we employed a nested variational inference scheme based on an alternative schedule of updates to the traditional ordering. In a typical schedule for coordinate ascent (which we call “naïve variational inference”), one initializes the variational Dirichlet parameters $\vec{\gamma}_{1:N}$ and the variational multinomial parameters $(\vec{\phi}_{p \rightarrow q}, \vec{\phi}_{p \leftarrow q})$ to non-informative values, and then iterates the following two steps until convergence: (i) update $\vec{\phi}_{p \rightarrow q}$ and $\phi_{p \leftarrow q}$ for all edges (p, q) , and (ii) update $\vec{\gamma}_p$ for all nodes $p \in \mathcal{N}$. In such algorithm, at each variational inference cycle we need to allocate $NK + 2N^2K$ scalars.

In our experiments, the naïve variational algorithm often converged only after many iterations. We attribute this behavior to the dependence between $\vec{\gamma}_{1:N}$ and B , which is not satisfied by the naïve algorithm. Some intuition about why this may happen follows. From a purely algorithmic

```

1. initialize  $\vec{\gamma}_{pk}^0 = \frac{2N}{K}$  for all  $p, k$ 
2. repeat
3.   for  $p = 1$  to  $N$ 
4.     for  $q = 1$  to  $N$ 
5.       get variational  $\vec{\phi}_{p \rightarrow q}^{t+1}$  and  $\vec{\phi}_{p \leftarrow q}^{t+1} = f(Y(p, q), \vec{\gamma}_p^t, \vec{\gamma}_q^t, B^t)$ 
6.       partially update  $\vec{\gamma}_p^{t+1}, \vec{\gamma}_q^{t+1}$  and  $B^{t+1}$ 
7.   until convergence

```

```

5.1. initialize  $\phi_{p \rightarrow q, g}^0 = \phi_{p \leftarrow q, h}^0 = \frac{1}{K}$  for all  $g, h$ 
5.2. repeat
5.3.   for  $g = 1$  to  $K$ 
5.4.     update  $\phi_{p \rightarrow q}^{s+1} \propto f_1(\vec{\phi}_{p \leftarrow q}^s, \vec{\gamma}_p, B)$ 
5.5.     normalize  $\vec{\phi}_{p \rightarrow q}^{s+1}$  to sum to 1
5.6.   for  $h = 1$  to  $K$ 
5.7.     update  $\phi_{p \leftarrow q}^{s+1} \propto f_2(\vec{\phi}_{p \rightarrow q}^s, \vec{\gamma}_q, B)$ 
5.8.     normalize  $\vec{\phi}_{p \leftarrow q}^{s+1}$  to sum to 1
5.9.   until convergence

```

Figure 5: **Top:** The two-layered variational inference for $(\vec{\gamma}, \phi_{p \rightarrow q, g}, \phi_{p \leftarrow q, h})$ and $M = 1$. The inner algorithm consists of Step 5. The function f is described in details in the bottom panel. The partial updates in Step 6 for $\vec{\gamma}$ and B refer to Equation 4 of Section B.4 and Equation 5 of Section B.5, respectively. **Bottom:** Inference for the variational parameters $(\vec{\phi}_{p \rightarrow q}, \vec{\phi}_{p \leftarrow q})$ corresponding to the basic observation $Y(p, q)$. This nested algorithm details Step 5 in the top panel. The functions f_1 and f_2 are the updates for $\phi_{p \rightarrow q, g}$ and $\phi_{p \leftarrow q, h}$ described in Equations 2 and 3 of Section B.4.

perspective, the naïve variational EM algorithm instantiates a large coordinate ascent algorithm, where the parameters can be divided into blocks. Blocks are processed in a specific order, and the parameters within each block get all updated each time.³ At every new iteration the naïve algorithm sets all the elements of $\vec{\gamma}_{1:N}^{t+1}$ equal to the same constant. This dampens the likelihood by suddenly breaking the dependence between the estimates of parameters in $\vec{\gamma}_{1:N}^t$ and in \hat{B}^t that was being inferred from the data during the previous iteration.

Instead, the nested variational inference algorithm maintains some of this dependence that is being inferred from the data across the various iterations. This is achieved mainly through a different

3. Within a block, the order according to which (scalar) parameters get updated is not expected to affect convergence.

scheduling of the parameter updates in the various blocks. To a minor extent, the dependence is maintained by always keeping the block of free parameters, $(\vec{\phi}_{p \rightarrow q}, \vec{\phi}_{p \leftarrow q})$, optimized given the other variational parameters. Note that these parameters are involved in the updates of parameters in $\vec{\gamma}_{1:N}$ and in B , thus providing us with a channel to maintain some of the dependence among them, that is, by keeping them at their optimal value given the data.

Furthermore, the nested algorithm has the advantage that it trades time for space thus allowing us to deal with large graphs; at each variational cycle we need to allocate $NK + 2K$ scalars only. The increased running time is partially offset by the fact that the algorithm can be parallelized and leads to empirically observed faster convergence rates.

An alternative strategy to perform inference is given by Monte Carlo Markov chain (e.g., see Griffiths and Steyvers, 2004; Kemp et al., 2004). While powerful in some settings, MCMC is impractical here. There are too many variables to sample. The proposed nested variational EM algorithm outperforms MCMC variations (i.e., blocked and collapsed Gibbs samplers) in terms of memory requirements and convergence rates.

3.2 Parameter Estimation

We compute the empirical Bayes estimates of the model hyper-parameters $\{\vec{\alpha}, B\}$ with a variational expectation-maximization (EM) algorithm. Alternatives to empirical Bayes have been proposed to fix the hyper-parameters and reduce the computation. The results, however, are not always satisfactory and often times cause of concern, since the inference is sensitive to the choice of the hyper-parameters (Joutard et al., 2007). Empirical Bayes, on the other hand, guides the posterior inference towards a region of the hyper-parameter space that is supported by the data.

Variational EM uses the lower bound in Equation 5 as a surrogate for the likelihood. To find a local optimum of the bound, we iterate between fitting the variational distribution q to approximate the posterior and maximizing the corresponding bound with respect to the parameters. The latter M-step is equivalent to finding the MLE using expected sufficient statistics under the variational distribution. We consider the maximization step for each parameter in turn.

A closed form solution for the approximate maximum likelihood estimate of $\vec{\alpha}$ does not exist (Minka, 2003). We use a linear-time Newton-Raphson method, where the gradient and Hessian are

$$\begin{aligned} \frac{\partial \mathcal{L}_{\vec{\alpha}}}{\partial \alpha_k} &= N \left(\Psi \left(\sum_k \alpha_k \right) - \Psi(\alpha_k) \right) + \sum_p \left(\Psi(\gamma_{p,k}) - \Psi \left(\sum_k \gamma_{p,k} \right) \right), \\ \frac{\partial \mathcal{L}_{\vec{\alpha}}}{\partial \alpha_{k_1} \alpha_{k_2}} &= N \left(\mathbb{I}_{(k_1=k_2)} \cdot \psi'(\alpha_{k_1}) - \psi' \left(\sum_k \alpha_k \right) \right). \end{aligned}$$

The approximate MLE of B is

$$\hat{B}(g, h) = \frac{\sum_{p,q} Y(p, q) \cdot \phi_{p \rightarrow qg} \phi_{p \leftarrow qh}}{(1 - \rho) \cdot \sum_{p,q} \phi_{p \rightarrow qg} \phi_{p \leftarrow qh}},$$

for every index pair $(g, h) \in [1, K] \times [1, K]$. Finally, the approximate MLE of the sparsity parameter ρ is

$$\hat{\rho} = \frac{\sum_{p,q} (1 - Y(p, q)) \cdot (\sum_{g,h} \phi_{p \rightarrow qg} \phi_{p \leftarrow qh})}{\sum_{p,q} \sum_{g,h} \phi_{p \rightarrow qg} \phi_{p \leftarrow qh}}.$$

Alternatively, we can fix ρ prior to the analysis; the density of the interaction matrix is estimated with $\hat{d} = \sum_{p,q} Y(p,q)/N^2$, and the sparsity parameter is set to $\tilde{\rho} = (1 - \hat{d})$. This latter estimator attributes all the information in the non-interactions to the point mass, that is, to latent sources other than the block model B or the mixed membership vectors $\vec{\pi}_{1:N}$. It does however provide a quick recipe to reduce the computational burden during exploratory analyses.⁴

Several model selection strategies are available for complex hierarchical models (Joutard et al., 2007). In our setting, model selection translates into the determination of a plausible value of the number of groups K . In the various analyses presented, we selected the optimal value of K according to two strategies. On large networks, we selected K corresponding to the highest averaged held-out likelihood in a cross-validation experiment. On small networks—where cross-validation cannot be expected to work well, as we discuss in Section 5—we selected K using an approximation to BIC.

4. Experiments and Results

We present a study of simulated data and applications to social and protein interaction networks.

Simulations are performed in Section 4.1 to show that both mixed membership, $\vec{\pi}_{1:N}$, and the latent block structure, B , can be recovered from data, when they exist, and that the nested variational inference algorithm is faster than the naïve implementation while reaching the same peak in the likelihood—all other things being equal.

The application to a friendship network among students in Section 4.2 tests the model on a real data set where we expect a well-defined latent block structure to inform the observed connectivity patterns in the network. In this application, the blocks are interpretable in terms of grades. We compare our results with those that were recently obtained with a simple mixture of blocks (Doreian et al., 2007) and with a latent space model (Handcock et al., 2007) on the same data.

The application to a protein interaction network in Section 4.3 tests the model on a real data set where we expect a noisy, vague latent block structure to inform the observed connectivity patterns in the network to some degree. In this application, the blocks are interpretable in terms functional biological contexts. This application tests to what extent our model can reduce the dimensionality of the data, while revealing substantive information about the functionality of proteins that can be used to inform subsequent analyses.

4.1 Exploring Expected Model Behavior with Simulations

In developing the MMB and the corresponding computation, our hope is the the model can recover both the mixed membership of nodes to clusters and the latent block structure among clusters in situations where a block structure exists and the relations are measured with some error. To substantiate this claim, we sampled graphs of 100, 300, and 600 nodes from blockmodels with 4, 10, and 20 clusters, respectively, using the MMB. We used different values of α to simulate a range of settings in terms of membership of nodes to clusters—from unique ($\alpha = 0.05$) to mixed ($\alpha = 0.25$).

Recovering the truth. The variational EM algorithm successfully recovers both the latent block model B and the latent mixed membership vectors $\vec{\pi}_{1:N}$. In Figure 6 we show the adjacency matrices of binary interactions where rows, that is, nodes, are reordered according to their most likely membership. The estimated reordering reveals the block model that was originally used to simulate

4. Note that $\tilde{\rho} = \hat{\rho}$ in the case of single membership. In fact, that implies $\phi_{p \rightarrow qg}^m = \phi_{p \leftarrow qh}^m = 1$ for some (g, h) pair, for any (p, q) pair.

the interactions. As α increases, each node is likely to belong to more clusters. As a consequence, they express interaction patterns of clusters. This phenomenon reflects in the reordered interaction matrices as the block structure is less evident.

Nested variational inference. The nested variational algorithm drives the log-likelihood to converge faster to its peak than the naïve algorithm. In Figure 7 (left panel) we compare the running times of the nested variational-EM algorithm versus the naïve implementation. The nested algorithm, which is more efficient in terms of space, converged faster. Furthermore, the nested variational algorithm can be parallelized given that the updates for each interaction (i, j) are independent of one another.

Choosing the number of blocks. The right panel of Figure 7 shows an example where cross-validation is sufficient to perform model selection for the MMB. The example shown corresponds to a network among 300 nodes with $K = 10$ clusters. We measure the number of latent clusters

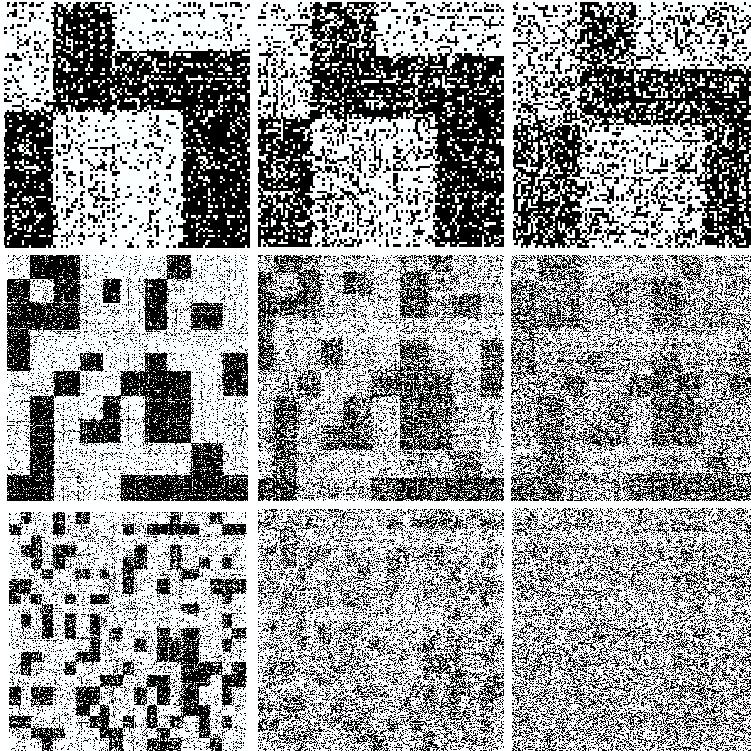


Figure 6: Adjacency matrices of corresponding to simulated interaction graphs with 100 nodes and 4 clusters, 300 nodes and 10 clusters, 600 nodes and 20 clusters (top to bottom) and α equal to 0.05, 0.1 and 0.25 (left to right). Rows, which corresponds to nodes, are reordered according to their most likely membership. The estimated reordering accurately reveals the original blockmodel.

on the X axis and the average held-out log-likelihood, corresponding to five-fold cross-validation experiments, on the Y axis. The nested variational EM algorithm was run until convergence, for each value of K we tested, with a tolerance of $\epsilon = 10^{-5}$. Our estimate for K occurs at the peak in the average held-out log-likelihood, and equals the correct number of clusters, $K^* = 10$

4.2 Application to Social Network Analysis

We considered a friendship network among a group of 69 students in grades 7–12. The analysis here directly compares clustering results obtained by MMB to published clustering results obtained by competing models, in a setting where a fair amount of social segregation is expected (Doreian et al., 2007; Handcock et al., 2007).

The National Longitudinal Study of Adolescent Health is nationally representative study that explores the how social contexts such as families, friends, peers, schools, neighborhoods, and communities influence health and risk behaviors of adolescents, and their outcomes in young adulthood (Harris et al., 2003; Udry, 2003). As part of the survey, a questionnaire was administered to a sample of students in each school, who were allowed to nominate up to 10 friends. We analyzed a friendship network among the students, at the same school that was considered by Handcock et al. (2007) and discussants. Friendship nominations were collected among 71 students in grades 7 to 12; two students did not nominate any friends. The network of binary, asymmetric friendship relations among the remaining 69 students that constitutes our data is shown in Figure 9 (left).

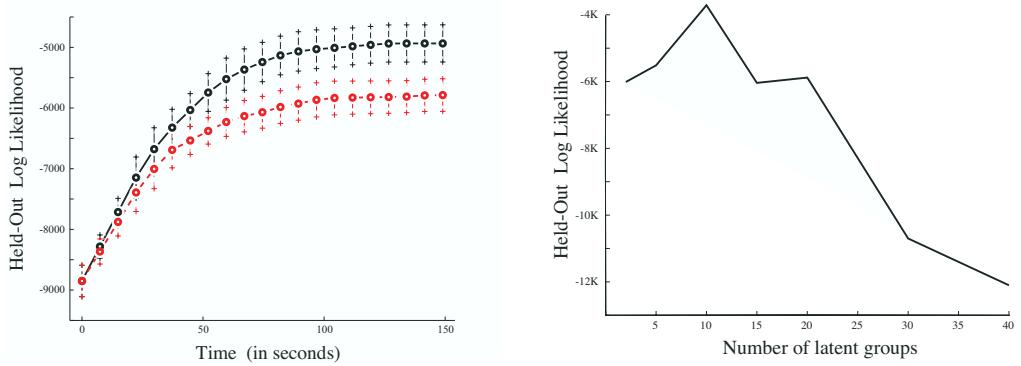


Figure 7: **Left:** The running time of the naïve variational inference (dashed, red line) against the running time of our enhanced (nested) variational inference algorithm (solid, black line), on a graph with 100 nodes and 4 clusters. We measure the number of seconds on the X axis and the log-likelihood on the Y axis. The two curves are averages over 26 experiments, and the error bars are at three standard deviations. Each of the 26 pairs of experiments was initialized with the same values for the parameters. **Right:** The held-out log-likelihood is indicative of the true number of latent clusters, on simulated data. We measure the number of latent clusters on the X axis and the log-likelihood on a test set on the Y axis. In the example shown, the peak identifies the correct number of clusters, $K^* = 10$

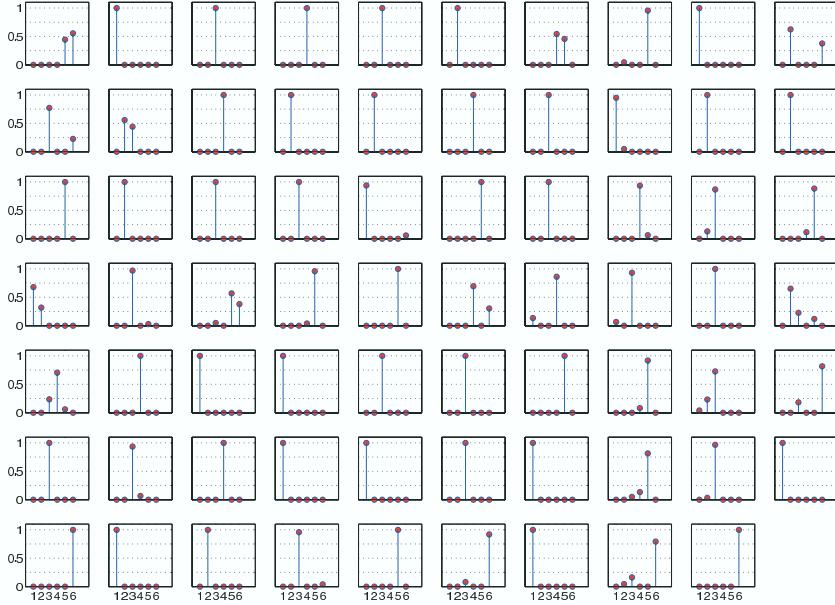


Figure 8: The posterior mixed membership scores, $\vec{\pi}$, for the 69 students. Each panel correspond to a student; we order the clusters 1 to 6 on the X axis, and we measure the student's grade of membership to these clusters on the Y axis.

Given the size of the network we used BIC to perform model selection, as in the monks example of Section 2.3. The results suggest a model with $K^* = 6$ groups. (We fix $K^* = 6$ in the analyses that follow.) The hyper-parameters estimated with the nested variational EM. They are $\hat{\alpha} = 0.0487$, $\hat{\beta} = 0.936$, and a fairly diagonal blockmodel,

$$\hat{B} = \begin{bmatrix} 0.3235 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.3614 & 0.0002 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.2607 & 0.0 & 0.0 & 0.0002 \\ 0.0 & 0.0 & 0.0 & 0.3751 & 0.0009 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0002 & 0.3795 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.3719 \end{bmatrix}.$$

Figure 8 shows the expected posterior mixed membership scores for the 69 students in the sample; few students display mixed membership. The rarity of mixed membership in this context is expected, while mixed membership may signal unexpected social situations for further investigation. For instance, it may signal a family bond such as brotherhood, or a student that is repeating a grade and is thus part of a broader social clique. In Figure 9, we contrast the friendship relation data (left) to the estimates obtained by thresholding the estimated probabilities of a relation, using the blockmodel and the node-specific latent variables (center) and the interactions-specific latent variables (right). The model provides a good summary of the social structure in the school; students

tend to befriend other students in the same grade, with a few exceptions. The low degree of mixed membership explains the absence of obvious differences between the model-based reconstructions of the friendship relations with the two model variants (center and right).

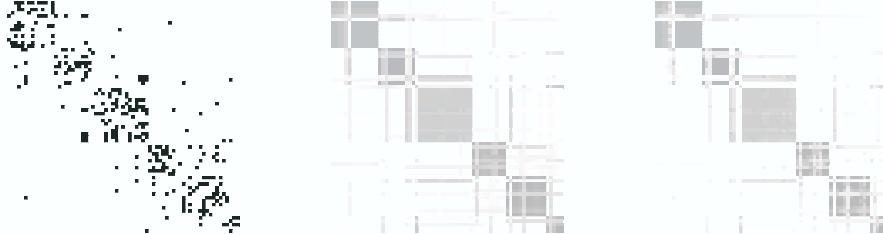


Figure 9: Original matrix of frienship relations among 69 students in grades 7 to 12 (left), and friendship estimated relations obtained by thresholding the posterior expectations $\vec{\pi}_p' B \vec{\pi}_q | Y$ (center), and $\vec{\phi}_p' B \vec{\phi}_q | Y$ (right).

Next, we attempted a quantitative evaluation of the goodness of fit. In this data, the blocks are clearly interpretable a-posteriori in terms of grades. The mixed membership vectors provide a mapping between grades and blocks. Conditionally on such a mapping, we assign students to the grade they are most associated with, according to their posterior-mean mixed membership vectors, $\mathbb{E}[\vec{\pi}_n | Y]$. To be fair in the comparison with competing models, we assign students to a unique grade—despite MMB allows for mixed membership. Table 1 computes the correspondence of grades to blocks by quoting the number of students in each grade-block pair, for MMB versus the mixture blockmodel (MB) in Doreian et al. (2007), and the latent space cluster model (LSCM) in Handcock et al. (2007). The higher the sum of counts on diagonal elements is the better is the correspondence, while the higher the sum of counts off diagonal elements is the worse is the correspondence. MMB performs best by allocating 63 students to their grades, versus 57 of MB, and 37 of LSCM. Correspondence only partially captures goodness of fit, however, it is a good metric in the setting we consider, where a fair amount of clustering is present. The results suggest that the extra-flexibility MMB offers over MB and LSCM reduces bias in the prediction of the membership of students to blocks. In other words, mixed membership does not absorb noise in this example; rather it accommodates variability in the friendship relation that is instrumental in producing better predictions.

Concluding this example, we note how the model decouples the observed friendship patterns into two complementary sources of variability. On the one hand, the connectivity matrix B is a global, unconstrained set of hyper-parameters. On the other hand, the mixed membership vectors $\vec{\pi}_{1:N}$ provide a collection of node-specific latent vectors, which inform the directed connections in the graph in a symmetric fashion.

4.3 Application to Protein Interactions in *Saccharomyces Cerevisiae*

We considered physical interactions among 871 proteins in yeast. The analysis allows us to evaluate the utility of MMB in summarizing and de-noising complex connectivity patterns quantitatively, using an independent set of functional annotations. For instance, between two models that sug-

Grade	MMB Clusters						MSB Clusters						LSCM Clusters					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
7	13	1	0	0	0	0	13	1	0	0	0	0	13	1	0	0	0	0
8	0	9	2	0	0	1	0	10	2	0	0	0	0	11	1	0	0	0
9	0	0	16	0	0	0	0	0	10	0	0	6	0	0	7	6	3	0
10	0	0	0	10	0	0	0	0	0	10	0	0	0	0	0	0	3	7
11	0	0	1	0	11	1	0	0	1	0	11	1	0	0	0	0	3	10
12	0	0	0	0	0	4	0	0	0	0	0	4	0	0	0	0	0	4

Table 1: Grade levels versus (highest) expected posterior membership for the 69 students, according to three alternative models. MMB is the proposed mixed membership stochastic blockmodel, MSB is a simpler stochastic block mixture model (Doreian et al., 2007), and LSCM is the latent space cluster model (Handcock et al., 2007).

gest different sets of interactions as reliable, we prefer the model that reveals *functionally relevant* interactions—as measured using the annotations.

Protein interactions (PPI) form the physical basis for the formation of stable protein complexes (i.e., protein clusters) and signaling pathways (i.e., cascades of protein interaction events) that carry out all major biological processes in the cell. A number of high-throughput experimental technologies have been devised to determine the set of interacting proteins on a global scale in yeast. These include two-hybrid (Y2H) screens and mass spectrometry methods (Gavin et al., 2002; Ho et al., 2002; Krogan et al., 2006). High-throughput technologies, however, often miss to identify interactions that are not present under the given conditions. Specific wet-lab methods employed by a certain technology, such as tagging, may disturb the formation of a stable protein complex, and weakly associated components may dissociate and escape detection. Statistical models that encode information about functional processes with high precision are an essential tool for carrying out probabilistic de-noising of biological signals from high-throughput experiments.

The goal of the analysis of protein interactions with MMB is to reveal the proteins’ diverse functional roles by analyzing their local and global patterns of interaction. The biochemical composition of individual proteins make them suitable for carrying out a specific set of cellular operations, or *functions*. The main intuition behind our methodology is that pairs of protein interact because they participate in the same cellular process, as part of the same stable protein complex, that is, co-location, or because they are part of interacting protein complexes, as they carry out compatible cellular operations (Alberts et al., 2002). Below, we describe the MIPS protein interactions data and the possible interpretations of the blocks in MMB in terms of biological functions, and we report results of two experiments.

4.3.1 PROTEIN INTERACTION DATA AND FUNCTIONAL ANNOTATION DATA

The Munich Institute for Protein Sequencing (MIPS) database was created in 1998 based on evidence derived from a variety of experimental techniques (Mewes et al., 2004). It includes a hand-curated collection of protein interactions that does not include interactions obtained with high-throughput technologies. The collection covers about 8000 protein complex associations in yeast.

We analyzed a subset of this collection containing 871 proteins, the interactions amongst which were hand-curated.

The MIPS institute also provides a set of functional annotations for each protein. These annotations are organized in a tree, with 15 nodes (i.e., high-level functions) at the first level, 72 nodes (i.e., the mid-level functions) at the second level, and 255 nodes (i.e., the low-level functions) at the leaf level. We mapped the 871 proteins in our collections to the high-level functions of the MIPS annotation tree. Table 2 quotes the number of proteins annotated to each of these 15 functions. Most proteins participate in more than one functional category, with an average of ≈ 2.4 functional annotations for each protein.. The relative importance of functional categories in our collection, in terms of the number of proteins involved, is similar to the relative importance of functional categories over the entire MIPS collection. We can also represent each protein in terms of its MIPS functional annotations. This leads to a 15-dimensional, binary representation for each protein, \vec{b}_p , where a component $\vec{b}_p(k) = 1$ indicates that protein p is annotated with function k in Table 2. Figure 10 shows the binary representations, $\vec{b}_{1:871}$, of the proteins in our collections; each panel corresponds to a protein; the 15 functional categories are ordered as in Table 2 on the X axis, whereas the presence or absence of the corresponding functional annotation is displayed on the Y axis. In Section 4.3.2, we fit a mixed membership blockmodel with $K = 15$, and we explore the direct correspondence between protein-specific mixed memberships to blocks, $\vec{\pi}_{1:871}$, and MIPS-derived functional annotations, $\vec{b}_{1:871}$.

An alternative source of functional annotations is the gene ontology (GO), distributed as part of the *Saccharomyces* genome database (Ashburner et al., 2000). GO provides vocabularies for describing the molecular function, biological process, and cellular component of gene products—such as proteins. Terms are organized in a directed acyclic graph. Terms at the top represent broader, more general concepts, terms lower down represent more specific concepts. There are two different relationship types between (parent-child) terms: “is a” and “part of”. Proteins are annotated to terms, and, most importantly, a protein is typically annotated to multiple terms, in different portions of the GO annotation graph. We restrict our evaluations to a collection of GO terms that is specific enough for a co-annotation (i.e., two proteins annotated to the same term) to be functionally relevant to molecular biologists (Myers et al., 2006). In Section 4.3.3, we select the mixed membership blockmodel best for predicting out-of-sample interactions, corresponding to

#	Category	Count	#	Category	Count
1	Metabolism	125	9	Interaction w/ cell. environment	18
2	Energy	56	10	Cellular regulation	37
3	Cell cycle & DNA processing	162	11	Cellular other	78
4	Transcription (tRNA)	258	12	Control of cell organization	36
5	Protein synthesis	220	13	Sub-cellular activities	789
6	Protein fate	170	14	Protein regulators	1
7	Cellular transportation	122	15	Transport facilitation	41
8	Cell rescue, defence & virulence	6			

Table 2: The 15 high-level functional categories obtained by cutting the MIPS annotation tree at the first level and how many proteins (out of 871) participate in each.

$K^* = 50$, and we explore its goodness-of-fit indirectly—rather than attempting a direct interpretation of the model’s parameters—in terms of the number of predicted interactions that are functionally relevant according to GO functional annotations.

4.3.2 DIRECT EVALUATION: THE MODEL CAPTURES SUBSTANTIVE BIOLOGY

In the first experiment, we fit a model with $K = 15$ blocks, and we attempt a direct interpretation of the blocks in terms of the 15 high-level functional categories in the MIPS annotation tree—separate from the MIPS protein interaction data, and independently conceived. We discuss results

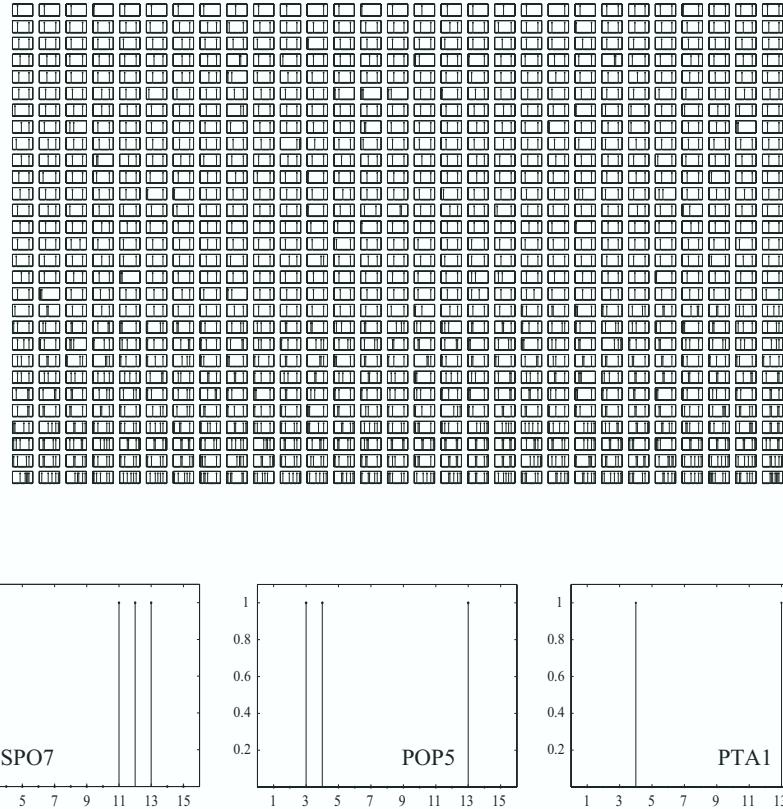


Figure 10: By mapping individual proteins to the 15 general functions in Table 2, we obtain a 15-dimensional representation for each protein. Here, each panel corresponds to a protein; the 15 functional categories are displayed on the X axis, whereas the presence or absence of the corresponding functional annotation is displayed on the Y axis. The plots at the bottom zoom into three example panels (proteins).

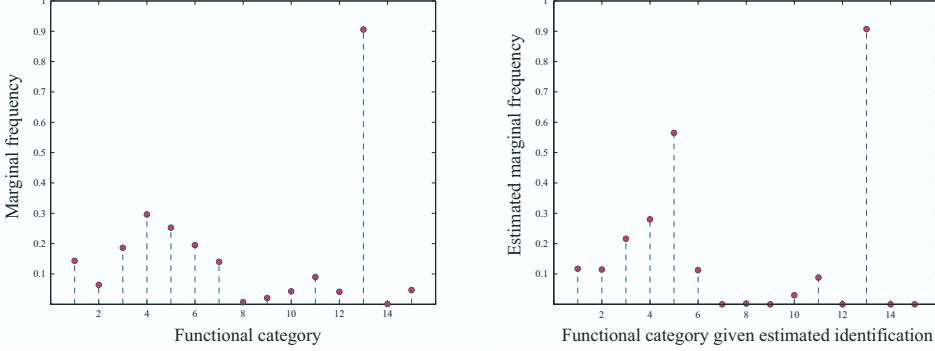


Figure 11: The mapping of blocks to functions is estimated by maximizing the accuracy of the predicted annotations of 87 proteins. We plot marginal frequencies of proteins’ membership to true functions (left) and to predicted functions (right).

that portray the relevance of mixed membership, the resolution of the identification of blocks with functional categories, and selected predictions.

We want to compute the correspondence between protein-specific mixed memberships to blocks, $\vec{\pi}_{1:87}$, and MIPS-derived functional annotations, $\vec{b}_{1:87}$. The $K = 15$ blocks in the blockmodel B are not directly identifiable in terms of functional categories. In other words, we need to estimate a permutation of the components of $\vec{\pi}_n$ in order to be able to interpret $E[\pi_n(k)|Y]$ as the expected degree of membership of protein n in function k of Table 2—rather than simply the expected degree of membership of protein n in block k , out of 15. To estimate the permutation that best identifies blocks to functions, we proceeded as follows. We sampled 87 proteins and their corresponding MIPS annotations, $\vec{b}_{1:87}$. We predicted membership of the 87 proteins by thresholding their mixed membership representations,

$$\hat{b}_n(k) = \begin{cases} 1 & \text{if } \pi_n(k) > \tau \\ 0 & \text{otherwise,} \end{cases}$$

where τ is the 95th percentile of the ensemble of elements of $\vec{\pi}_{1:87}$, corresponding to the 87 proteins in the training set. We then greedily identified the mapping that maximizing the accuracy of the predicted annotations of 87 proteins. We used this mapping to compare predicted versus known functional annotations for all proteins; in Figure 11 we plot marginal frequencies of proteins’ membership to true functions (left panel) and to predicted functions (right panel). The accuracy on the 90% testing set is about 87%. An algorithm that randomly guesses annotations, knowing the right proportions of annotations in each category, leads to a baseline accuracy of about 70%. Figure 12 shows predicted mixed memberships (dashed, red lines) versus the true annotations (solid, black lines), given the estimated mapping of blocks to functions, for six example proteins.

4.3.3 INDIRECT EVALUATION: FUNCTIONAL CONTENT OF PREDICTED INTERACTIONS

In the second experiment, we selected the mixed membership blockmodel best for predicting out-of-sample interactions, and we explored its goodness-of-fit indirectly, in terms of the number of

predicted interactions that are functionally relevant according to GO present in estimated protein interaction networks obtained with the two types of analyses that MMB supports; summarization and de-noising.

We fit models with K ranging between 2 and 255. We selected the best model ($K = 50$) using cross-validated held-out log likelihood, as in Figure 7. This finding supports the hypothesis that proteins derived from the MIPS data are interpretable in terms functional biological contexts. Alternatively, the blocks might encode signal at a finer resolution, such as that of protein complexes.

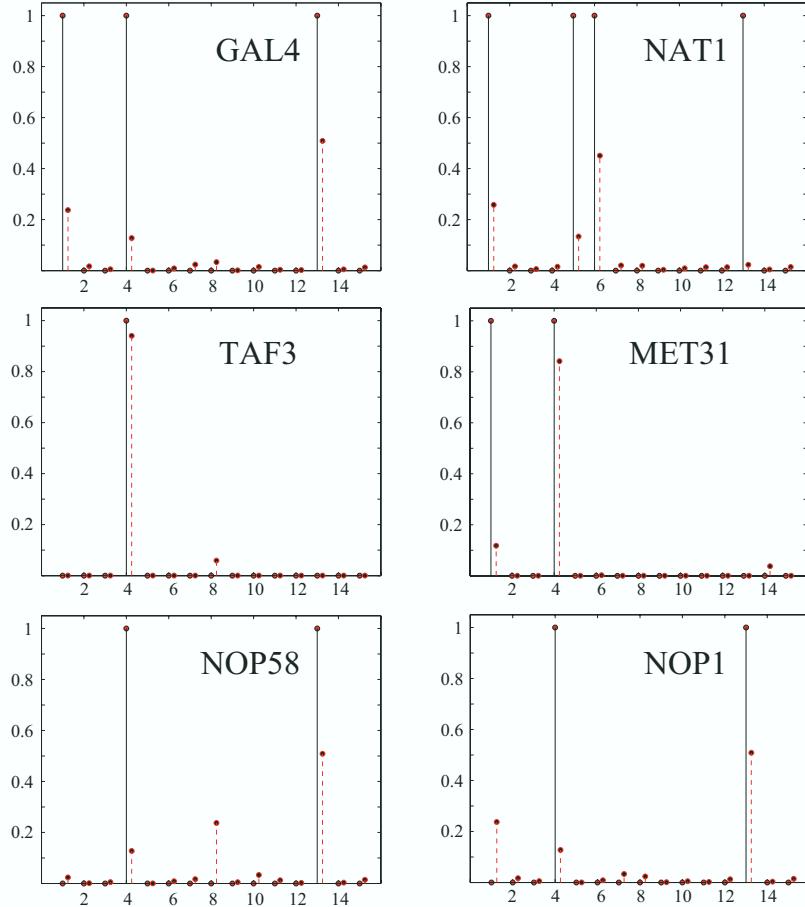


Figure 12: Predicted mixed-memberships (dashed, red lines) versus binary manually curated functional annotations (solid, black lines) for six example proteins, given the estimated mapping of blocks to functions in Figure 11.

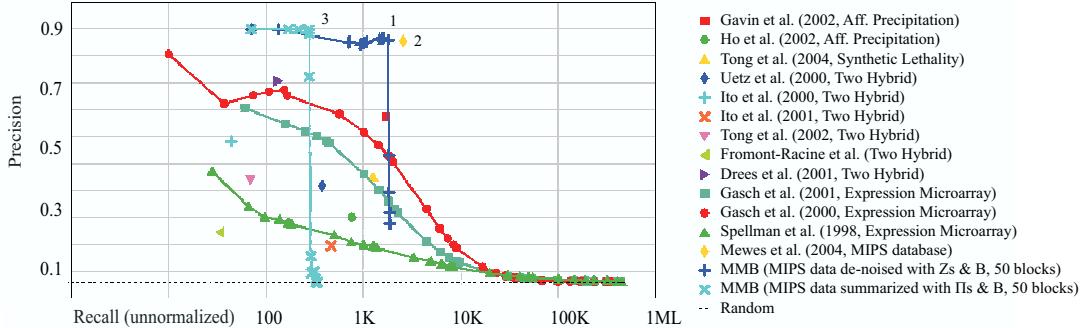


Figure 13: In the top panel we measure the functional content of the the MIPS collection of protein interactions (yellow diamond), and compare it against other published collections of interactions and microarray data, and to the posterior estimates of the MMB models—computed as described in Section 4.3.3. A breakdown of three estimated interaction networks (the points annotated 1, 2, and 3) into most represented gene ontology categories is detailed in Table 3.

If that was the case, however, we would expect the optimal number of blocks to be significantly higher; $871/5 \approx 175$, given an average size of five proteins in a complex (Krogan et al., 2006).

Using this model, we computed posterior model-based expectations of each interaction as follows,

$$\mathbb{E}[Y(p, q)] \approx \hat{\pi}_p' \hat{B} \hat{\pi}_q \quad \text{and} \quad \mathbb{E}[Y(p, q)] \approx \hat{\phi}_{p \rightarrow q}' \hat{B} \hat{\phi}_{p \leftarrow q}.$$

These computations lead to two estimated protein interaction networks with expected probabilities of interactions taking values in $[0, 1]$. We obtained binary protein interaction networks by thresholding these expected probabilities at ten different values. In terms of the two analyses described in Section 2.2, this amount to either (i) predicting physical interactions by thresholding the posterior expectations computed using blockmodel B and mixed membership map $\hat{\pi}$ s, essentially a prediction task, or (ii) we de-noise the observed interactions Y using the blockmodel B and interaction-specific membership indicators Z s, essentially a de-noising task. We use the independent set of functional annotations from the gene ontology to decide which interactions are functionally meaningful; namely those between pairs of proteins that share at least one functional annotation (Myers et al., 2006). In this sense, between two models that suggest different sets of interactions as reliable, our evaluation assigns a higher score to the model that reveals *functionally relevant* interactions. Figure 13 shows the functional content of the original MIPS collection of physical interactions (point no.2), and of the collections of interactions computed using (B, PIs) , the light blue ($-x$) line, and using (B, Zs) , the dark blue ($-+$) line, thresholded at ten different levels—precision-recall curves. The posterior means of PIs provide a parsimonious representation for the MIPS collection, and lead to precise interaction estimates, in moderate amount ($-x$ line). The posterior means of Z s provide a richer representation for the data, and describe most of the functional content of the MIPS collection with high precision ($-+$ line). Figure 13 also shows the functional content of the original MIPS collection (the yellow diamond). Most importantly, notice the estimated protein interaction

networks, that is, ex-es and crosses, corresponding to lower levels of recall feature a more precise functional content than the original. This means that the proposed latent block structure is helpful in summarizing the collection of interactions—by ranking them properly. On closer inspection, dense blocks of predicted interactions contain known functional predictions that were not in the MIPS collection, thus effectively improving the quality of the data that instantiate activity specific to few biological contexts, such as biopolymer catabolism and homeostasis. In conclusion, results suggest that MMB successfully reduces the dimensionality of the data, while revealing substantive information about the multiple functionality of proteins that can be used to inform subsequent analyses.

Table 3 provides more information about three instances of predicted interaction networks displayed in Figure 13; those corresponding the points annotated 1, 2, and 3. Specifically, the table shows a breakdown of the predicted (posterior) collections of interactions in each example network into the gene ontology categories. A count in the table corresponds to the fact that both proteins are annotated with the same GO functional category.⁵

In this application, the MMB learned information about (i) the mixed membership of objects to latent groups, and (ii) the connectivity patterns among latent groups. These estimates were useful in describing and summarizing the functional content of the MIPS collection of protein interactions. This suggests the use of MMB as a dimensionality reduction approach that may be useful for performing model-driven de-noising of new collections of interactions, such as those measured via high-throughput experiments.

5. Discussion

Modern probabilistic models for relational data analysis are rooted in the stochastic blockmodels for psychometric and sociological analysis, pioneered by Lorrain and White (1971) and by Holland and Leinhardt (1975). In statistics, this line of research has been extended in various contexts over the years (Fienberg et al., 1985; Wasserman and Pattison, 1996; Snijders, 2002; Hoff et al., 2002; Doreian et al., 2004). In machine learning, the related technique of Markov random networks (Frank and Strauss, 1986) have been used for link prediction (Taskar et al., 2003) and the traditional blockmodels have been extended to include nonparametric Bayesian priors (Kemp et al., 2004, 2006; Xu et al., 2006) and to integrate relations and text (McCallum et al., 2007).

There is a close relationship between the MMB and the latent space models (Hoff et al., 2002; Handcock et al., 2007). In the latent space models, the latent vectors are drawn from Gaussian distributions and the interaction data is drawn from a Gaussian with mean $\vec{\pi}_p' \vec{\pi}_q$. In the MMB, the marginal probability of an interaction takes a similar form, $\vec{\pi}_p' B \vec{\pi}_q$, where B is the matrix of probabilities of interactions for each pair of latent groups. Two major differences exist between these approaches. In MMB, the distribution over the latent vectors is a Dirichlet and the underlying data distribution is arbitrary—we have chosen Bernoulli. The posterior inference in latent space models (Hoff et al., 2002; Handcock et al., 2007) is carried out via MCMC sampling, while we have developed a scalable variational inference algorithm to analyze large network structures. (It would be interesting to develop a variational algorithm for the latent space models as well.) A number of well-designed numerical investigations and comparisons between variational EM and variants of MCMC have been performed in existing literature; for instance, see Buntine and Jakulin (2006),⁶

5. Note that, in GO, proteins are typically annotated to multiple functional categories.

6. See corresponding slides with additional results. (http://www.hiit.fi/~buntine/dpca_slides.pdf)

#	GO Term	Description	Pred.	Tot.
1	GO:0043285	Biopolymer catabolism	561	17020
1	GO:0006366	Transcription from RNA polymerase II promoter	341	36046
1	GO:0006412	Protein biosynthesis	281	299925
1	GO:0006260	DNA replication	196	5253
1	GO:0006461	Protein complex assembly	191	11175
1	GO:0016568	Chromatin modification	172	15400
1	GO:0006473	Protein amino acid acetylation	91	666
1	GO:0006360	Transcription from RNA polymerase I promoter	78	378
1	GO:0042592	Homeostasis	78	5778
2	GO:0043285	Biopolymer catabolism	631	17020
2	GO:0006366	Transcription from RNA polymerase II promoter	414	36046
2	GO:0016568	Chromatin modification	229	15400
2	GO:0006260	DNA replication	226	5253
2	GO:0006412	Protein biosynthesis	225	299925
2	GO:0045045	Secretory pathway	151	18915
2	GO:0006793	Phosphorus metabolism	134	17391
2	GO:0048193	Golgi vesicle transport	128	9180
2	GO:0006352	Transcription initiation	121	1540
3	GO:0006412	Protein biosynthesis	277	299925
3	GO:0006461	Protein complex assembly	190	11175
3	GO:0009889	Regulation of biosynthesis	28	990
3	GO:0051246	Regulation of protein metabolism	28	903
3	GO:0007046	Ribosome biogenesis	10	21528
3	GO:0006512	Ubiquitin cycle	3	2211

Table 3: Breakdown of three example interaction networks into most represented gene ontology categories—see text for more details. The digit in the first column indicates the example network in Figure 13 that any given line refers to. The last two columns quote the number of predicted, and possible pairs for each GO term.

and Braun and McAuliffe (2007). We refer readers interested in the comparison between variational vs. MCMC to these resources.

The model decouples the observed connectivity patterns into two sources of variability, B, Π_s , that are apparently in competition for explaining the data, possibly raising an identifiability issue. This is not the case, however, as the blockmodel B captures global/asymmetric relations, while the mixed membership vectors Π_s capture local/symmetric relations. This difference practically eliminates the issue, unless there is no signal in the data to begin with.

A recurring question, which bears relevance to mixed membership models in general, is why we do not integrate out the single membership indicators— $(\vec{z}_{p \rightarrow q}, \vec{z}_{p \leftarrow q})$. While this may lead to computational efficiencies we would often lose interpretable quantities that are useful for making predictions, for de-noising new measurements, or for performing other tasks. In fact, the posterior distributions of such quantities typically carry substantive information about elements of the appli-

cation at hand. In the application to protein interaction networks of Section 4.3, for example, they encode the interaction-specific memberships of individual proteins to protein complexes.

In the relational setting, cross-validation is feasible if the blockmodel estimated on training data can be expected to hold on test data; for this to happen the network must be of reasonable size, so that we can expect members of each block to be in both training and test sets. In this setting, scheduling of variational updates is important; nested variational scheduling leads to efficient and parallelizable inference.

A limitation of our model can be best appreciated in a simulation setting. If we consider structural properties of the network MMB is capable of generating, we count a wide array of local and global connectivity patterns. But the model does not readily generate *hubs*, that is, nodes connected with a large number of directed or undirected connections, or networks with skewed degree distributions.

From a data analysis perspective, we speculate that the value of MMB in capturing substantive information about a problem will increase in semi-supervised setting—where, for example, information about the membership of genes to functional contexts is included in the form of prior distributions. In such a setting we may be interested in looking at the change between prior and posterior membership; a sharp change may signal biological phenomena worth investigating. We need not assume that the number of groups/blocks, K , is finite. It is possible, for example, to posit that the mixed-membership vectors are sampled from a stochastic process, in the nonparametric setting. To maintain mixed membership of nodes to groups/blocks in such setting, we need to sample them from a hierarchical Dirichlet process (Teh et al., 2006), rather than from a Dirichlet Process (Escobar and West, 1995).

MMB generalizes to two important cases. First, multiple data collections $Y_{1:M}$ on the same objects can be generated by the same latent vectors. This might be useful, for example, for simultaneously analyzing the relational measurements about esteem and disesteem, liking and disliking, positive influence and negative influence, praise and blame, for example, see Sampson (1968), or those about the collection of 17 relations measured by Bradley (1987). Second, in the MMB the data generating distribution is a Bernoulli, but B can be a matrix that parameterizes any kind of distribution. For example, technologies for measuring interactions between pairs of proteins such as mass spectrometry (Ho et al., 2002) and tandem affinity purification (Gavin et al., 2002) return a probabilistic assessment about the presence of interactions, thus setting the range of $Y(p, q)$ to $[0, 1]$. This is not the case for the manually curated collection of interactions we analyze in Section 4.3.

6. Conclusions

In this paper we introduced mixed membership stochastic blockmodels, a novel class of latent variable models for relational data. These models provide exploratory tools for scientific analyses in applications where the observations can be represented as a collection of unipartite graphs. The nested variational inference algorithm is parallelizable and allows fast approximate inference on large graphs.

Acknowledgments

This work was partially supported by National Institutes of Health under Grant No. R01 AG023141-01, by the Office of Naval Research under Contracts N00014-02-1-0973 and 175-6343, by the National Science Foundation under Grants No. DMS-0240019, IIS-0218466, IIS-0745520 and DBI-0546594, by the Pennsylvania Department of Health's Health Research Program under Grant No. 2001NF-Cancer Health Research Grant ME-01-739, and by the Department of Defense, all to Carnegie Mellon University. The authors would like to thank David Banks and Jim Berger at Duke University, Alan Karr at the National Institute of Statistical Sciences for insight and advice, and acknowledge generous support from the Statistical and Applied Mathematical Sciences Institute.

Appendix A. General Model Formulation

In general, mixed membership stochastic blockmodels can be specified in terms of assumptions at four levels: population, node, latent variable, and sampling scheme level.

A.1 Population Level

Assume that there are K classes or sub-populations in the population of interest. We denote by $f(Y(p,q) | B(g,h))$ the probability distribution of the relation measured on the pair of nodes (p,q) , where the p -th node is in the h -th sub-population, the q -th node is in the h -th sub-population, and $B(g,h)$ contains the relevant parameters. The indices i,j run in $1,\dots,N$, and the indices g,h run in $1,\dots,K$.

A.2 Node Level

The components of the membership vector $\vec{\pi}_p = [\vec{\pi}_p(1), \dots, \vec{\pi}_p(k)]'$ encode the mixed membership of the n -th node to the various sub-populations. The distribution of the observed response $Y(p,q)$ given the relevant, node-specific memberships, $(\vec{\pi}_p, \vec{\pi}_q)$, is then

$$\Pr(Y(p,q) | \vec{\pi}_p, \vec{\pi}_q, B) = \sum_{g,h=1}^K \vec{\pi}_p(g) f(Y(p,q) | B(g,h)) \vec{\pi}_q(h).$$

Conditional on the mixed memberships, the response edges y_{jnum} are independent of one another, both across distinct graphs and pairs of nodes.

A.3 Latent Variable Level

Assume that the mixed membership vectors $\vec{\pi}_{1:N}$ are realizations of a latent variable with distribution $D_{\vec{\alpha}}$, with parameter vector $\vec{\alpha}$. The probability of observing $Y(p,q)$, given the parameters, is then

$$\Pr(Y(p,q) | \vec{\alpha}, B) = \int \Pr(Y(p,q) | \vec{\pi}_p, \vec{\pi}_q, B) D_{\vec{\alpha}}(d\vec{\pi}).$$

A.4 Sampling Scheme Level

Assume that the M independent replications of the relations measured on the population of nodes are independent of one another. The probability of observing the whole collection of graphs, $Y_{1:M}$, given the parameters, is then given by the following equation.

$$\Pr(Y_{1:M} | \vec{\alpha}, B) = \prod_{m=1}^M \prod_{p,q=1}^N \Pr(Y_m(p,q) | \vec{\alpha}, B).$$

Full model specifications immediately adapt to the different kinds of data, for example, multiple data types through the choice of f , or parametric or semi-parametric specifications of the prior on the number of clusters through the choice of a distribution for the π_s , D_α .

Appendix B. Details of the Variational Approximation

Here we present more details about the derivation of the variational EM algorithm presented in Section 3. Furthermore, we address a setting where M replicates are available about the paired measurements, $G_{1:M} = (N, Y_{1:M})$, and relations $Y_m(p, q)$ take values into an arbitrary metric space according to $f(Y_m(p, q) \mid \dots)$. An extension of the inference algorithm to address the case of multivariate relations, say J -dimensional, and multiple blockmodels $B_{1:J}$ each corresponding to a distinct relational response, can be derived with minor modifications of the derivations that follow.

B.1 Variational Expectation-Maximization

We begin by briefly summarizing the general strategy we intend to use. The approximate variant of EM we describe here is often referred to as *Variational EM* (Beal and Ghahramani, 2003). Recall that Y denotes the data. Rewrite $X = (\vec{\pi}_{1:N}, Z_{\rightarrow}, Z_{\leftarrow})$ for the latent variables, and $\Theta = (\vec{\alpha}, B)$ for the model's parameters. Briefly, it is possible to lower bound the likelihood, $p(Y|\Theta)$, making use of Jensen's inequality and of any distribution on the latent variables $q(X)$,

$$\begin{aligned} p(Y|\Theta) &= \log \int_X p(Y, X|\Theta) dX \\ &= \log \int_X q(X) \frac{p(Y, X|\Theta)}{q(X)} dX \quad (\text{for any } q) \\ &\geq \int_X q(X) \log \frac{p(Y, X|\Theta)}{q(X)} dX \quad (\text{Jensen's}) \\ &= \mathbb{E}_q [\log p(Y, X|\Theta) - \log q(X)] =: \mathcal{L}(q, \Theta) \end{aligned}$$

In EM, the lower bound $\mathcal{L}(q, \Theta)$ is then iteratively maximized with respect to Θ , in the M step, and q in the E step (Dempster et al., 1977). In particular, at the t -th iteration of the E step we set

$$q^{(t)} = p(X|Y, \Theta^{(t-1)}), \quad (5)$$

that is, equal to the posterior distribution of the latent variables given the data and the estimates of the parameters at the previous iteration.

Unfortunately, we cannot compute the posterior in Equation 5 for the admixture of latent blocks model. Rather, we define a direct parametric approximation to it, $\tilde{q} = q_\Delta(X)$, which involves an extra set of *variational parameters*, Δ , and entails an approximate lower bound for the likelihood $\mathcal{L}_\Delta(q, \Theta)$. At the t -th iteration of the E step, we then minimize the Kullback-Leibler divergence between $q^{(t)}$ and $q_\Delta^{(t)}$, with respect to Δ , using the data.⁷ The optimal parametric approximation is, in fact, a proper posterior as it depends on the data Y , although indirectly, $q^{(t)} \approx q_{\Delta^*(Y)}^{(t)}(X) = p(X|Y)$.

B.2 Lower Bound for the Likelihood

According to the mean-field theory (Jordan et al., 1999), one can approximate an intractable distribution such as the one defined by Equation (1) by a fully factored distribution $q(\vec{\pi}_{1:N}, Z_{1:M}^\rightarrow, Z_{1:M}^\leftarrow)$

7. This is equivalent to maximizing the approximate lower bound for the likelihood, $\mathcal{L}_\Delta(q, \Theta)$, with respect to Δ .

defined as follows:

$$\begin{aligned} & q(\vec{\pi}_{1:N}, Z_{1:M}^{\rightarrow}, Z_{1:M}^{\leftarrow} | \vec{\gamma}_{1:N}, \Phi_{1:M}^{\rightarrow}, \Phi_{1:M}^{\leftarrow}) \\ &= \prod_p q_1(\vec{\pi}_p | \vec{\gamma}_p) \prod_m \prod_{p,q} \left(q_2(\vec{z}_{p \rightarrow q}^m | \vec{\phi}_{p \rightarrow q}^m, 1) q_2(\vec{z}_{p \leftarrow q}^m | \vec{\phi}_{p \leftarrow q}^m, 1) \right), \end{aligned}$$

where q_1 is a Dirichlet, q_2 is a multinomial, and $\Delta = (\vec{\gamma}_{1:N}, \Phi_{1:M}^{\rightarrow}, \Phi_{1:M}^{\leftarrow})$ represent the set of free variational parameters need to be estimated in the approximate distribution.

Minimizing the Kulback-Leibler divergence between this $q(\vec{\pi}_{1:N}, Z_{1:M}^{\rightarrow}, Z_{1:M}^{\leftarrow} | \Delta)$ and the original $p(\vec{\pi}_{1:N}, Z_{1:M}^{\rightarrow}, Z_{1:M}^{\leftarrow})$ defined by Equation (1) leads to the following approximate lower bound for the likelihood.

$$\begin{aligned} \mathcal{L}_{\Delta}(q, \Theta) &= \mathbb{E}_q \left[\log \prod_m \prod_{p,q} p_1(Y_m(p, q) | \vec{z}_{p \rightarrow q}^m, \vec{z}_{p \leftarrow q}^m, B) \right] \\ &+ \mathbb{E}_q \left[\log \prod_m \prod_{p,q} p_2(\vec{z}_{p \rightarrow q}^m | \vec{\pi}_p, 1) \right] + \mathbb{E}_q \left[\log \prod_m \prod_{p,q} p_2(\vec{z}_{p \leftarrow q}^m | \vec{\pi}_q, 1) \right] \\ &+ \mathbb{E}_q \left[\log \prod_p p_3(\vec{\pi}_p | \vec{\alpha}) \right] - \mathbb{E}_q \left[\prod_p q_1(\vec{\pi}_p | \vec{\gamma}_p) \right] \\ &- \mathbb{E}_q \left[\log \prod_m \prod_{p,q} q_2(\vec{z}_{p \rightarrow q}^m | \vec{\phi}_{p \rightarrow q}^m, 1) \right] - \mathbb{E}_q \left[\log \prod_m \prod_{p,q} q_2(\vec{z}_{p \leftarrow q}^m | \vec{\phi}_{p \leftarrow q}^m, 1) \right]. \end{aligned}$$

Working on the single expectations leads to

$$\begin{aligned} \mathcal{L}_{\Delta}(q, \Theta) &= \sum_m \sum_{p,q,g,h} \phi_{p \rightarrow q, g}^m \phi_{p \leftarrow q, h}^m \cdot f(Y_m(p, q), B(g, h)) \\ &+ \sum_m \sum_{p,q,g} \phi_{p \rightarrow q, g}^m \left[\psi(\gamma_{p,g}) - \psi(\sum_g \gamma_{p,g}) \right] \\ &+ \sum_m \sum_{p,q,h} \phi_{p \leftarrow q, h}^m \left[\psi(\gamma_{p,h}) - \psi(\sum_h \gamma_{p,h}) \right] \\ &+ \sum_p \log \Gamma(\sum_k \alpha_k) - \sum_{p,k} \log \Gamma(\alpha_k) + \sum_{p,k} (\alpha_k - 1) \left[\psi(\gamma_{p,k}) - \psi(\sum_k \gamma_{p,k}) \right] \\ &- \sum_p \log \Gamma(\sum_k \gamma_{p,k}) + \sum_{p,k} \log \Gamma(\gamma_{p,k}) - \sum_{p,k} (\gamma_{p,k} - 1) \left[\psi(\gamma_{p,k}) - \psi(\sum_k \gamma_{p,k}) \right] \\ &- \sum_m \sum_{p,q,g} \phi_{p \rightarrow q, g}^m \log \phi_{p \rightarrow q, g}^m - \sum_m \sum_{p,q,h} \phi_{p \leftarrow q, h}^m \log \phi_{p \leftarrow q, h}^m \end{aligned}$$

where

$$f(Y_m(p, q), B(g, h)) = Y_m(p, q) \log B(g, h) + (1 - Y_m(p, q)) \log(1 - B(g, h));$$

m runs over $1, \dots, M$; p, q run over $1, \dots, N$; g, h, k run over $1, \dots, K$; and $\psi(x)$ is the derivative of the log-gamma function, $\frac{d \log \Gamma(x)}{dx}$.

B.3 The Expected Value of the Log of a Dirichlet Random Vector

The computation of the lower bound for the likelihood requires us to evaluate $\mathbb{E}_q [\log \vec{\pi}_p]$ for $p = 1, \dots, N$. Recall that the density of an exponential family distribution with natural parameter $\vec{\theta}$ can be written as

$$\begin{aligned} p(x|\alpha) &= h(x) \cdot c(\alpha) \cdot \exp \left\{ \sum_k \theta_k(\alpha) \cdot t_k(x) \right\} \\ &= h(x) \cdot \exp \left\{ \sum_k \theta_k(\alpha) \cdot t_k(x) - \log c(\alpha) \right\}. \end{aligned}$$

Omitting the node index p for convenience, we can rewrite the density of the Dirichlet distribution p_3 as an exponential family distribution,

$$p_3(\vec{\pi}|\vec{\alpha}) = \exp \left\{ \sum_k (\alpha_k - 1) \log(\pi_k) - \log \frac{\prod_k \Gamma(\alpha_k)}{\Gamma(\sum_k \alpha_k)} \right\},$$

with natural parameters $\Theta_k(\vec{\alpha}) = (\alpha_k - 1)$ and natural sufficient statistics $t_k(\vec{\pi}) = \log(\pi_k)$. Let $c'(\vec{\theta}) = c(\alpha_1(\vec{\theta}), \dots, \alpha_K(\vec{\theta}))$; using a well known property of the exponential family distributions (Schervish, 1995) we find that

$$\mathbb{E}_q [\log \pi_k] = \mathbb{E}_{\vec{\theta}} [\log t_k(x)] = \psi(\alpha_k) - \psi(\sum_k \alpha_k),$$

where $\psi(x)$ is the derivative of the log-gamma function, $\frac{d \log \Gamma(x)}{dx}$.

B.4 Variational E Step

The approximate lower bound for the likelihood $\mathcal{L}_\Delta(q, \Theta)$ can be maximized using exponential family arguments and coordinate ascent (Wainwright and Jordan, 2003).

Isolating terms containing $\phi_{p \rightarrow q, g}^m$ and $\phi_{p \leftarrow q, h}^m$ we obtain $\mathcal{L}_{\phi_{p \rightarrow q, g}^m}(q, \Theta)$ and $\mathcal{L}_{\phi_{p \leftarrow q, h}^m}(q, \Theta)$. The natural parameters $\vec{g}_{p \rightarrow q}^m$ and $\vec{g}_{p \leftarrow q}^m$ corresponding to the natural sufficient statistics $\log(z_{p \rightarrow q}^m)$ and $\log(z_{p \leftarrow q}^m)$ are functions of the other latent variables and the observations. We find that

$$\begin{aligned} g_{p \rightarrow q, g}^m &= \log \pi_{p, g} + \sum_h z_{p \leftarrow q, h}^m \cdot f(Y_m(p, q), B(g, h)), \\ g_{p \leftarrow q, h}^m &= \log \pi_{q, h} + \sum_g z_{p \rightarrow q, g}^m \cdot f(Y_m(p, q), B(g, h)), \end{aligned}$$

for all pairs of nodes (p, q) in the m -th network; where $g, h = 1, \dots, K$, and

$$f(Y_m(p, q), B(g, h)) = Y_m(p, q) \log B(g, h) + (1 - Y_m(p, q)) \log(1 - B(g, h)).$$

This leads to the following updates for the variational parameters $(\vec{\phi}_{p \rightarrow q}^m, \vec{\phi}_{p \leftarrow q}^m)$, for a pair of nodes (p, q) in the m -th network:

$$\begin{aligned} \hat{\phi}_{p \rightarrow q, g}^m &\propto e^{\mathbb{E}_q[g_{p \rightarrow q, g}^m]} \\ &= e^{\mathbb{E}_q[\log \pi_{p, g}]} \cdot e^{\sum_h \phi_{p \rightarrow q, h}^m \cdot \mathbb{E}_q[f(Y_m(p, q), B(g, h))]} \\ &= e^{\mathbb{E}_q[\log \pi_{p, g}]} \cdot \prod_h \left(B(g, h)^{Y_m(p, q)} \cdot (1 - B(g, h))^{1 - Y_m(p, q)} \right)^{\phi_{p \rightarrow q, h}^m}, \\ \hat{\phi}_{p \leftarrow q, h}^m &\propto e^{\mathbb{E}_q[g_{p \leftarrow q, h}^m]} \\ &= e^{\mathbb{E}_q[\log \pi_{q, h}]} \cdot e^{\sum_g \phi_{p \leftarrow q, g}^m \cdot \mathbb{E}_q[f(Y_m(p, q), B(g, h))]} \\ &= e^{\mathbb{E}_q[\log \pi_{q, h}]} \cdot \prod_g \left(B(g, h)^{Y_m(p, q)} \cdot (1 - B(g, h))^{1 - Y_m(p, q)} \right)^{\phi_{p \leftarrow q, g}^m}, \end{aligned}$$

for $g, h = 1, \dots, K$. These estimates of the parameters underlying the distribution of the nodes' group indicators $\vec{\phi}_{p \rightarrow q}^m$ and $\vec{\phi}_{p \leftarrow q}^m$ need be normalized, to make sure $\sum_k \hat{\phi}_{p \rightarrow q, k}^m = \sum_k \hat{\phi}_{p \leftarrow q, k}^m = 1$.

Isolating terms containing $\gamma_{p,k}$ we obtain $\mathcal{L}_{\gamma_{p,k}}(q, \Theta)$. Setting $\frac{\partial \mathcal{L}_{\gamma_{p,k}}}{\partial \gamma_{p,k}}$ equal to zero and solving for $\gamma_{p,k}$ yields:

$$\hat{\gamma}_{p,k} = \alpha_k + \sum_m \sum_q \phi_{p \rightarrow q, k}^m + \sum_m \sum_q \phi_{p \leftarrow q, k}^m,$$

for all nodes $p \in \mathcal{P}$ and $k = 1, \dots, K$.

The t -th iteration of the variational E step is carried out for fixed values of $\Theta^{(t-1)} = (\vec{\alpha}^{(t-1)}, B^{(t-1)})$, and finds the optimal approximate lower bound for the likelihood $\mathcal{L}_{\Delta^*}(q, \Theta^{(t-1)})$.

B.5 Variational M Step

The optimal lower bound $\mathcal{L}_{\Delta^*}(q^{(t-1)}, \Theta)$ provides a tractable surrogate for the likelihood at the t -th iteration of the variational M step. We derive empirical Bayes estimates for the hyper-parameters Θ that are based upon it.⁸ That is, we maximize $\mathcal{L}_{\Delta^*}(q^{(t-1)}, \Theta)$ with respect to Θ , given expected sufficient statistics computed using $\mathcal{L}_{\Delta^*}(q^{(t-1)}, \Theta^{(t-1)})$.

Isolating terms containing $\vec{\alpha}$ we obtain $\mathcal{L}_{\vec{\alpha}}(q, \Theta)$. Unfortunately, a closed form solution for the approximate maximum likelihood estimate of $\vec{\alpha}$ does not exist (Blei et al., 2003). We can produce a Newton-Raphson method that is linear in time, where the gradient and Hessian for the bound $\mathcal{L}_{\vec{\alpha}}$ are

$$\begin{aligned} \frac{\partial \mathcal{L}_{\vec{\alpha}}}{\partial \alpha_k} &= N \left(\psi \left(\sum_k \alpha_k \right) - \psi(\alpha_k) \right) + \sum_p \left(\psi(\gamma_{p,k}) - \psi \left(\sum_k \gamma_{p,k} \right) \right), \\ \frac{\partial \mathcal{L}_{\vec{\alpha}}}{\partial \alpha_{k_1} \alpha_{k_2}} &= N \left(\mathbb{I}_{(k_1=k_2)} \cdot \psi'(\alpha_{k_1}) - \psi' \left(\sum_k \alpha_k \right) \right). \end{aligned}$$

Isolating terms containing B we obtain \mathcal{L}_B , whose approximate maximum is

$$\hat{B}(g, h) = \frac{1}{M} \sum_m \left(\frac{\sum_{p,q} Y_m(p, q) \cdot \phi_{p \rightarrow qg}^m \phi_{p \leftarrow qh}^m}{(1-\rho) \cdot \sum_{p,q} \phi_{p \rightarrow qg}^m \phi_{p \leftarrow qh}^m} \right),$$

for every index pair $(g, h) \in [1, K] \times [1, K]$.

In Section 2.1 we introduced an extra parameter, ρ , to control the relative importance of presence and absence of interactions in likelihood, that is, the score that informs inference and estimation. Isolating terms containing ρ we obtain \mathcal{L}_ρ . We may then estimate the sparsity parameter ρ by

$$\hat{\rho} = \frac{1}{M} \sum_m \left(\frac{\sum_{p,q} (1 - Y_m(p, q)) \cdot (\sum_{g,h} \phi_{p \rightarrow qg}^m \phi_{p \leftarrow qh}^m)}{\sum_{p,q} \sum_{g,h} \phi_{p \rightarrow qg}^m \phi_{p \leftarrow qh}^m} \right).$$

Alternatively, we can fix ρ prior to the analysis; the density of the interaction matrix is estimated with $\hat{d} = \sum_{m,p,q} Y_m(p, q) / (N^2 M)$, and the sparsity parameter is set to $\tilde{\rho} = (1 - \hat{d})$. This latter estimator attributes all the information in the non-interactions to the point mass, that is, to latent sources other than the block model B or the mixed membership vectors $\vec{\pi}_{1:N}$. It does, however, provide a quick recipe to reduce the computational burden during exploratory analyses.⁹

-
- 8. We could term these estimates *pseudo* empirical Bayes estimates, since they maximize an approximate lower bound for the likelihood, \mathcal{L}_{Δ^*} .
 - 9. Note that $\tilde{\rho} = \hat{\rho}$ in the case of single membership. In fact, that implies $\phi_{p \rightarrow qg}^m = \phi_{p \leftarrow qh}^m = 1$ for some (g, h) pair, for any (p, q) pair.

References

- E. M. Airoldi. Getting started in probabilistic graphical models. *PLoS Computational Biology*, 3(12):e252, 2007.
- E. M. Airoldi, D. M. Blei, E. P. Xing, and S. E. Fienberg. A latent mixed-membership model for relational data. In *ACM SIGKDD Workshop on Link Discovery: Issues, Approaches and Applications*, 2005.
- E. M. Airoldi, S. E. Fienberg, and E. P. Xing. Mixed membership analysis of expression studies—attribute data. Manuscript, 2007. URL <http://arxiv.org/abs/0711.2520/>.
- B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. Garland, 4th edition, 2002.
- M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubinand, and G. Sherlock. Gene ontology: Tool for the unification of biology. The gene ontology consortium. *Nature Genetics*, 25(1):25–29, 2000.
- M. J. Beal and Z. Ghahramani. The variational Bayesian EM algorithm for incomplete data: With application to scoring graphical model structures. In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, editors, *Bayesian Statistics*, volume 7, pages 453–464. Oxford University Press, 2003.
- L. Berkman, B. H. Singer, and K. Manton. Black/white differences in health status and mortality among the elderly. *Demography*, 26(4):661–678, 1989.
- C. Bishop, D. Spiegelhalter, and J. Winn. VIBES: A variational inference engine for Bayesian networks. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 777–784. MIT Press, Cambridge, MA, 2003.
- D. M. Blei, A. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- R. T. Bradley. *Charisma and Social Structure*. Paragon House, 1987.
- M. Braun and J. McAuliffe. Variational inference for large-scale models of discrete choice. Manuscript, 2007. URL <http://arxiv.org/abs/0712.2526/>.
- R. L. Breiger, S. A. Boorman, and P. Arabie. An algorithm for clustering relational data with applications to social network analysis and comparison to multidimensional scaling. *Journal of Mathematical Psychology*, 12:328–383, 1975.
- W. L. Buntine and A. Jakulin. Discrete components analysis. In C. Saunders, M. Grobelnik, S. Gunn, and J. Shawe-Taylor, editors, *Subspace, Latent Structure and Feature Selection Techniques*. Springer-Verlag, 2006. URL <http://arxiv.org/abs/math.ST/0604410/>.
- G. B. Davis and K. M. Carley. Clearing the FOG: Fuzzy, overlapping groups for social networks. Manuscript, 2006.

- A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.
- P. Doreian, V. Batagelj, and A. Ferligoj. *Generalized Blockmodeling*. Cambridge University Press, 2004.
- P. Doreian, V. Batagelj, and A. Ferligoj. Discussion of “Model-based clustering for social networks”. *Journal of the Royal Statistical Society, Series A*, 170, 2007.
- E. A. Erosheva. *Grade of Membership and Latent Structure Models with Application to Disability Survey Data*. PhD thesis, Carnegie Mellon University, Department of Statistics, 2002.
- E. A. Erosheva and S. E. Fienberg. Bayesian mixed membership models for soft clustering and classification. In C. Weihs and W. Gaul, editors, *Classification—The Ubiquitous Challenge*, pages 11–26. Springer-Verlag, 2005.
- M. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.
- S. E. Fienberg, M. M. Meyer, and S. Wasserman. Statistical analysis of multiple sociometric relations. *Journal of the American Statistical Association*, 80:51–67, 1985.
- O. Frank and D. Strauss. Markov graphs. *Journal of the American Statistical Association*, 81:832–842, 1986.
- A. C. Gavin, M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, and et. al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415:141–147, 2002.
- T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, 2004.
- M. S. Handcock, A. E. Raftery, and J. M. Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society, Series A*, 170:1–22, 2007.
- K. M. Harris, F. Florey, J. Tabor, P. S. Bearman, J. Jones, and R. J. Udry. The national longitudinal study of adolescent health: research design. Technical report, Carolina Population Center, University of North Carolina, Chapel Hill, 2003.
- Y. Ho, A. Gruhler, A. Heilbut, G. D. Bader, L. Moore, S. L. Adams, A. Millar, P. Taylor, K. Bennett, and K. Boutilier et. al. Systematic identification of protein complexes in *saccharomyces cerevisiae* by mass spectrometry. *Nature*, 415:180–183, 2002.
- P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090–1098, 2002.
- P. W. Holland and S. Leinhardt. Local structure in social networks. In D. Heise, editor, *Sociological Methodology*, pages 1–45. Jossey-Bass, 1975.
- M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. Introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

MIXED MEMBERSHIP STOCHASTIC BLOCKMODELS

- C. Joutard, E. M. Airoldi, S. E. Fienberg, and T. M. Love. Discovery of latent patterns with hierarchical bayesian mixed-membership models and the issue of model choice. In *Data Mining Patterns, New Methods and Applications*, 2007. Forthcoming.
- C. Kemp, T. L. Griffiths, and J. B. Tenenbaum. Discovering latent classes in relational data. Technical Report AI Memo 2004-019, MIT, 2004.
- C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.
- N. J. Krogan, G. Cagney, H. Yu, G. Zhong, X. Guo, A. Ignatchenko, J. Li, S. Pu, N. Datta, A. P. Tikuisis, T. Punna, J. M. Peregrin-Alvarez, M. Shales, X. Zhang, M. Davey, M. D. Robinson, A. Paccanaro, J. E. Bray, A. Sheung, B. Beattie, D. P. Richards, V. Canadien, A. Lalev, F. Mena, P. Wong, A. Starostine, M. M. Canete, J. Vlasblom, S. Wu, C. Orsi, S. R. Collins, S. Chandran, R. Haw, J. J. Rilstone, K. Gandi, N. J. Thompson, G. Musso, P. St Onge, S. Ghanny, M. H. Y. Lam, G. Butland, A. M. Altaf-Ul, S. Kanaya, A. Shilatifard, E. O'Shea, J. S. Weissman, C. J. Ingles, T. R. Hughes, J. Parkinson, M. Gerstein, S. J. Wodak, A. Emili, and J. F. Greenblatt. Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. *Nature*, 440(7084):637–643, 2006.
- F.-F. Li and P. Perona. A Bayesian hierarchical model for learning natural scene categories. *IEEE Computer Vision and Pattern Recognition*, 2005.
- F. Lorrain and H. C. White. Structural equivalence of individuals in social networks. *Journal of Mathematical Sociology*, 1:49–80, 1971.
- A. McCallum, X. Wang, and N. Mohanty. Joint group and topic discovery from relations and text. In *Statistical Network Analysis: Models, Issues and New Directions*, Lecture Notes in Computer Science. Springer-Verlag, 2007.
- H. W. Mewes, C. Amid, R. Arnold, D. Frishman, U. Guldener, and et. al. Mips: analysis and annotation of proteins from whole genomes. *Nucleic Acids Research*, 32:D41–44, 2004.
- T. Minka. Estimating a Dirichlet distribution. Manuscript, 2003.
- T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Uncertainty in Artificial Intelligence*, 2002.
- C. L. Myers, D. A. Barret, M. A. Hibbs, C. Huttenhower, and O. G. Troyanskaya. Finding function: An evaluation framework for functional genomics. *BMC Genomics*, 7(187), 2006.
- J. K. Pritchard, M. Stephens, N. A. Rosenberg, and P. Donnelly. Association mapping in structured populations. *American Journal of Human Genetics*, 67:170–181, 2000.
- F. S. Sampson. *A Novitiate in a Period of Change: An Experimental and Case Study of Social Relationships*. PhD thesis, Cornell University, 1968.
- Mark J. Schervish. *Theory of Statistics*. Springer, 1995.

- T. A. B. Snijders. Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 2002.
- T. A. B. Snijders and K. Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14:75–100, 1997.
- B. Taskar, M. F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Neural Information Processing Systems 15*, 2003.
- Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.
- Y. W. Teh, D. Newman, and M. Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, volume 19, 2007.
- R. J. Udry. The national longitudinal study of adolescent health: (add health) waves i and ii, 1994–1996; wave iii 2001–2002. Technical report, Caolina Population Center, University of North Carolina, Chapel Hill, 2003.
- C. T. Volinsky and A. E. Raftery. Bayesian information criterion for censored survival models. *Biometrics*, 56:256–262, 2000.
- M. J. Wainwright and M. I. Jordan. Graphical models, exponential families and variational inference. Technical Report 649, Department of Statistics, University of California, Berkeley, 2003.
- Y. J. Wang and G. Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82:8–19, 1987.
- S. Wasserman and P. Pattison. Logit models and logistic regression for social networks: I. an introduction to markov graphs and p^* . *Psychometrika*, 61:401–425, 1996.
- E. P. Xing, M. I. Jordan, and S. Russell. A generalized mean field algorithm for variational inference in exponential families. In *Uncertainty in Artificial Intelligence*, volume 19, 2003.
- Z. Xu, V. Tresp, K. Yu, and H.-P. Kriegel. Infinite hidden relational models. In *Uncertainty in Artificial Intelligence*, 2006.

SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS

Thomas N. Kipf
 University of Amsterdam
 T.N.Kipf@uva.nl

Max Welling
 University of Amsterdam
 Canadian Institute for Advanced Research (CIFAR)
 M.Welling@uva.nl

ABSTRACT

We present a scalable approach for semi-supervised learning on graph-structured data that is based on an efficient variant of convolutional neural networks which operate directly on graphs. We motivate the choice of our convolutional architecture via a localized first-order approximation of spectral graph convolutions. Our model scales linearly in the number of graph edges and learns hidden layer representations that encode both local graph structure and features of nodes. In a number of experiments on citation networks and on a knowledge graph dataset we demonstrate that our approach outperforms related methods by a significant margin.

1 INTRODUCTION

We consider the problem of classifying nodes (such as documents) in a graph (such as a citation network), where labels are only available for a small subset of nodes. This problem can be framed as graph-based semi-supervised learning, where label information is smoothed over the graph via some form of explicit graph-based regularization (Zhu et al., 2003; Zhou et al., 2004; Belkin et al., 2006; Weston et al., 2012), e.g. by using a graph Laplacian regularization term in the loss function:

$$\mathcal{L} = \mathcal{L}_0 + \lambda \mathcal{L}_{\text{reg}}, \quad \text{with} \quad \mathcal{L}_{\text{reg}} = \sum_{i,j} A_{ij} \|f(X_i) - f(X_j)\|^2 = f(X)^\top \Delta f(X). \quad (1)$$

Here, \mathcal{L}_0 denotes the supervised loss w.r.t. the labeled part of the graph, $f(\cdot)$ can be a neural network-like differentiable function, λ is a weighing factor and X is a matrix of node feature vectors X_i . $\Delta = D - A$ denotes the unnormalized graph Laplacian of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes $v_i \in \mathcal{V}$, edges $(v_i, v_j) \in \mathcal{E}$, an adjacency matrix $A \in \mathbb{R}^{N \times N}$ (binary or weighted) and a degree matrix $D_{ii} = \sum_j A_{ij}$. The formulation of Eq. 1 relies on the assumption that connected nodes in the graph are likely to share the same label. This assumption, however, might restrict modeling capacity, as graph edges need not necessarily encode node similarity, but could contain additional information.

In this work, we encode the graph structure directly using a neural network model $f(X, A)$ and train on a supervised target \mathcal{L}_0 for all nodes with labels, thereby avoiding explicit graph-based regularization in the loss function. Conditioning $f(\cdot)$ on the adjacency matrix of the graph will allow the model to distribute gradient information from the supervised loss \mathcal{L}_0 and will enable it to learn representations of nodes both with and without labels.

Our contributions are two-fold. Firstly, we introduce a simple and well-behaved layer-wise propagation rule for neural network models which operate directly on graphs and show how it can be motivated from a first-order approximation of spectral graph convolutions (Hammond et al., 2011). Secondly, we demonstrate how this form of a graph-based neural network model can be used for fast and scalable semi-supervised classification of nodes in a graph. Experiments on a number of datasets demonstrate that our model compares favorably both in classification accuracy and efficiency (measured in wall-clock time) against state-of-the-art methods for semi-supervised learning.

2 FAST APPROXIMATE CONVOLUTIONS ON GRAPHS

In this section, we provide theoretical motivation for a specific graph-based neural network model $f(X, A)$ that we will use in the rest of this paper. We consider a multi-layer Graph Convolutional Network (GCN) with the following layer-wise propagation rule:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right). \quad (2)$$

Here, $\tilde{A} = A + I_N$ is the adjacency matrix of the undirected graph \mathcal{G} with added self-connections. I_N is the identity matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and $W^{(l)}$ is a layer-specific trainable weight matrix. $\sigma(\cdot)$ denotes an activation function, such as the ReLU(\cdot) = $\max(0, \cdot)$. $H^{(l)} \in \mathbb{R}^{N \times D}$ is the matrix of activations in the l^{th} layer; $H^{(0)} = X$. In the following, we show that the form of this propagation rule can be motivated¹ via a first-order approximation of localized spectral filters on graphs (Hammond et al., 2011; Defferrard et al., 2016).

2.1 SPECTRAL GRAPH CONVOLUTIONS

We consider spectral convolutions on graphs defined as the multiplication of a signal $x \in \mathbb{R}^N$ (a scalar for every node) with a filter $g_\theta = \text{diag}(\theta)$ parameterized by $\theta \in \mathbb{R}^N$ in the Fourier domain, i.e.:

$$g_\theta \star x = Ug_\theta U^\top x, \quad (3)$$

where U is the matrix of eigenvectors of the normalized graph Laplacian $L = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = U\Lambda U^\top$, with a diagonal matrix of its eigenvalues Λ and $U^\top x$ being the graph Fourier transform of x . We can understand g_θ as a function of the eigenvalues of L , i.e. $g_\theta(\Lambda)$. Evaluating Eq. 3 is computationally expensive, as multiplication with the eigenvector matrix U is $\mathcal{O}(N^2)$. Furthermore, computing the eigendecomposition of L in the first place might be prohibitively expensive for large graphs. To circumvent this problem, it was suggested in Hammond et al. (2011) that $g_\theta(\Lambda)$ can be well-approximated by a truncated expansion in terms of Chebyshev polynomials $T_k(x)$ up to K^{th} order:

$$g_{\theta'}(\Lambda) \approx \sum_{k=0}^K \theta'_k T_k(\tilde{\Lambda}), \quad (4)$$

with a rescaled $\tilde{\Lambda} = \frac{2}{\lambda_{\max}}\Lambda - I_N$. λ_{\max} denotes the largest eigenvalue of L . $\theta' \in \mathbb{R}^K$ is now a vector of Chebyshev coefficients. The Chebyshev polynomials are recursively defined as $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, with $T_0(x) = 1$ and $T_1(x) = x$. The reader is referred to Hammond et al. (2011) for an in-depth discussion of this approximation.

Going back to our definition of a convolution of a signal x with a filter $g_{\theta'}$, we now have:

$$g_{\theta'} \star x \approx \sum_{k=0}^K \theta'_k T_k(\tilde{L})x, \quad (5)$$

with $\tilde{L} = \frac{2}{\lambda_{\max}}L - I_N$; as can easily be verified by noticing that $(U\Lambda U^\top)^k = U\Lambda^k U^\top$. Note that this expression is now K -localized since it is a K^{th} -order polynomial in the Laplacian, i.e. it depends only on nodes that are at maximum K steps away from the central node (K^{th} -order neighborhood). The complexity of evaluating Eq. 5 is $\mathcal{O}(|\mathcal{E}|)$, i.e. linear in the number of edges. Defferrard et al. (2016) use this K -localized convolution to define a convolutional neural network on graphs.

2.2 LAYER-WISE LINEAR MODEL

A neural network model based on graph convolutions can therefore be built by stacking multiple convolutional layers of the form of Eq. 5, each layer followed by a point-wise non-linearity. Now, imagine we limited the layer-wise convolution operation to $K = 1$ (see Eq. 5), i.e. a function that is linear w.r.t. L and therefore a linear function on the graph Laplacian spectrum.

¹We provide an alternative interpretation of this propagation rule based on the Weisfeiler-Lehman algorithm (Weisfeiler & Lehmann, 1968) in Appendix A.

In this way, we can still recover a rich class of convolutional filter functions by stacking multiple such layers, but we are not limited to the explicit parameterization given by, e.g., the Chebyshev polynomials. We intuitively expect that such a model can alleviate the problem of overfitting on local neighborhood structures for graphs with very wide node degree distributions, such as social networks, citation networks, knowledge graphs and many other real-world graph datasets. Additionally, for a fixed computational budget, this layer-wise linear formulation allows us to build deeper models, a practice that is known to improve modeling capacity on a number of domains (He et al., 2016).

In this linear formulation of a GCN we further approximate $\lambda_{\max} \approx 2$, as we can expect that neural network parameters will adapt to this change in scale during training. Under these approximations Eq. 5 simplifies to:

$$g_{\theta'} \star x \approx \theta'_0 x + \theta'_1 (L - I_N) x = \theta'_0 x - \theta'_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x, \quad (6)$$

with two free parameters θ'_0 and θ'_1 . The filter parameters can be shared over the whole graph. Successive application of filters of this form then effectively convolve the k^{th} -order neighborhood of a node, where k is the number of successive filtering operations or convolutional layers in the neural network model.

In practice, it can be beneficial to constrain the number of parameters further to address overfitting and to minimize the number of operations (such as matrix multiplications) per layer. This leaves us with the following expression:

$$g_{\theta} \star x \approx \theta \left(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) x, \quad (7)$$

with a single parameter $\theta = \theta'_0 = -\theta'_1$. Note that $I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ now has eigenvalues in the range $[0, 2]$. Repeated application of this operator can therefore lead to numerical instabilities and exploding/vanishing gradients when used in a deep neural network model. To alleviate this problem, we introduce the following *renormalization trick*: $I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \rightarrow \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$, with $\tilde{A} = A + I_N$ and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$.

We can generalize this definition to a signal $X \in \mathbb{R}^{N \times C}$ with C input channels (i.e. a C -dimensional feature vector for every node) and F filters or feature maps as follows:

$$Z = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta, \quad (8)$$

where $\Theta \in \mathbb{R}^{C \times F}$ is now a matrix of filter parameters and $Z \in \mathbb{R}^{N \times F}$ is the convolved signal matrix. This filtering operation has complexity $\mathcal{O}(|\mathcal{E}|FC)$, as $\tilde{A}X$ can be efficiently implemented as a product of a sparse matrix with a dense matrix.

3 SEMI-SUPERVISED NODE CLASSIFICATION

Having introduced a simple, yet flexible model $f(X, A)$ for efficient information propagation on graphs, we can return to the problem of semi-supervised node classification. As outlined in the introduction, we can relax certain assumptions typically made in graph-based semi-supervised learning by conditioning our model $f(X, A)$ both on the data X and on the adjacency matrix A of the underlying graph structure. We expect this setting to be especially powerful in scenarios where the adjacency matrix contains information not present in the data X , such as citation links between documents in a citation network or relations in a knowledge graph. The overall model, a multi-layer GCN for semi-supervised learning, is schematically depicted in Figure 1.

3.1 EXAMPLE

In the following, we consider a two-layer GCN for semi-supervised node classification on a graph with a symmetric adjacency matrix A (binary or weighted). We first calculate $\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ in a pre-processing step. Our forward model then takes the simple form:

$$Z = f(X, A) = \text{softmax}\left(\hat{A} \text{ReLU}\left(\hat{A} X W^{(0)}\right) W^{(1)}\right). \quad (9)$$

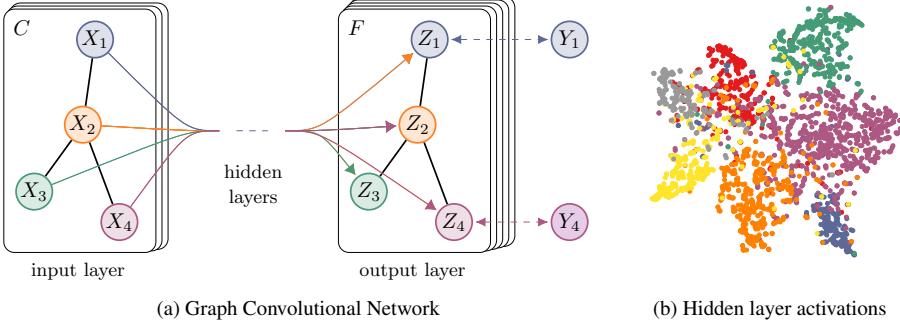


Figure 1: *Left:* Schematic depiction of multi-layer Graph Convolutional Network (GCN) for semi-supervised learning with C input channels and F feature maps in the output layer. The graph structure (edges shown as black lines) is shared over layers, labels are denoted by Y_i . *Right:* t-SNE (Maaten & Hinton, 2008) visualization of hidden layer activations of a two-layer GCN trained on the Cora dataset (Sen et al., 2008) using 5% of labels. Colors denote document class.

Here, $W^{(0)} \in \mathbb{R}^{C \times H}$ is an input-to-hidden weight matrix for a hidden layer with H feature maps. $W^{(1)} \in \mathbb{R}^{H \times F}$ is a hidden-to-output weight matrix. The softmax activation function, defined as $\text{softmax}(x_i) = \frac{1}{Z} \exp(x_i)$ with $Z = \sum_i \exp(x_i)$, is applied row-wise. For semi-supervised multi-class classification, we then evaluate the cross-entropy error over all labeled examples:

$$\mathcal{L} = - \sum_{l \in \mathcal{Y}_L} \sum_{f=1}^F Y_{lf} \ln Z_{lf}, \quad (10)$$

where \mathcal{Y}_L is the set of node indices that have labels.

The neural network weights $W^{(0)}$ and $W^{(1)}$ are trained using gradient descent. In this work, we perform batch gradient descent using the full dataset for every training iteration, which is a viable option as long as datasets fit in memory. Using a sparse representation for A , memory requirement is $\mathcal{O}(|\mathcal{E}|)$, i.e. linear in the number of edges. Stochasticity in the training process is introduced via dropout (Srivastava et al., 2014). We leave memory-efficient extensions with mini-batch stochastic gradient descent for future work.

3.2 IMPLEMENTATION

In practice, we make use of TensorFlow (Abadi et al., 2015) for an efficient GPU-based implementation² of Eq. 9 using sparse-dense matrix multiplications. The computational complexity of evaluating Eq. 9 is then $\mathcal{O}(|\mathcal{E}|CHF)$, i.e. linear in the number of graph edges.

4 RELATED WORK

Our model draws inspiration both from the field of graph-based semi-supervised learning and from recent work on neural networks that operate on graphs. In what follows, we provide a brief overview on related work in both fields.

4.1 GRAPH-BASED SEMI-SUPERVISED LEARNING

A large number of approaches for semi-supervised learning using graph representations have been proposed in recent years, most of which fall into two broad categories: methods that use some form of explicit graph Laplacian regularization and graph embedding-based approaches. Prominent examples for graph Laplacian regularization include label propagation (Zhu et al., 2003), manifold regularization (Belkin et al., 2006) and deep semi-supervised embedding (Weston et al., 2012).

²Code to reproduce our experiments is available at <https://github.com/tkipf/gcn>.

Recently, attention has shifted to models that learn graph embeddings with methods inspired by the skip-gram model (Mikolov et al., 2013). DeepWalk (Perozzi et al., 2014) learns embeddings via the prediction of the local neighborhood of nodes, sampled from random walks on the graph. LINE (Tang et al., 2015) and node2vec (Grover & Leskovec, 2016) extend DeepWalk with more sophisticated random walk or breadth-first search schemes. For all these methods, however, a multi-step pipeline including random walk generation and semi-supervised training is required where each step has to be optimized separately. Planetoid (Yang et al., 2016) alleviates this by injecting label information in the process of learning embeddings.

4.2 NEURAL NETWORKS ON GRAPHS

Neural networks that operate on graphs have previously been introduced in Gori et al. (2005); Scarselli et al. (2009) as a form of recurrent neural network. Their framework requires the repeated application of contraction maps as propagation functions until node representations reach a stable fixed point. This restriction was later alleviated in Li et al. (2016) by introducing modern practices for recurrent neural network training to the original graph neural network framework. Duvenaud et al. (2015) introduced a convolution-like propagation rule on graphs and methods for graph-level classification. Their approach requires to learn node degree-specific weight matrices which does not scale to large graphs with wide node degree distributions. Our model instead uses a single weight matrix per layer and deals with varying node degrees through an appropriate normalization of the adjacency matrix (see Section 3.1).

A related approach to node classification with a graph-based neural network was recently introduced in Atwood & Towsley (2016). They report $\mathcal{O}(N^2)$ complexity, limiting the range of possible applications. In a different yet related model, Niepert et al. (2016) convert graphs locally into sequences that are fed into a conventional 1D convolutional neural network, which requires the definition of a node ordering in a pre-processing step.

Our method is based on spectral graph convolutional neural networks, introduced in Bruna et al. (2014) and later extended by Defferrard et al. (2016) with fast localized convolutions. In contrast to these works, we consider here the task of transductive node classification within networks of significantly larger scale. We show that in this setting, a number of simplifications (see Section 2.2) can be introduced to the original frameworks of Bruna et al. (2014) and Defferrard et al. (2016) that improve scalability and classification performance in large-scale networks.

5 EXPERIMENTS

We test our model in a number of experiments: semi-supervised document classification in citation networks, semi-supervised entity classification in a bipartite graph extracted from a knowledge graph, an evaluation of various graph propagation models and a run-time analysis on random graphs.

5.1 DATASETS

We closely follow the experimental setup in Yang et al. (2016). Dataset statistics are summarized in Table 1. In the citation network datasets—Citeseer, Cora and Pubmed (Sen et al., 2008)—nodes are documents and edges are citation links. Label rate denotes the number of labeled nodes that are used for training divided by the total number of nodes in each dataset. NELL (Carlson et al., 2010; Yang et al., 2016) is a bipartite graph dataset extracted from a knowledge graph with 55,864 relation nodes and 9,891 entity nodes.

Table 1: Dataset statistics, as reported in Yang et al. (2016).

Dataset	Type	Nodes	Edges	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003
NELL	Knowledge graph	65,755	266,144	210	5,414	0.001

Citation networks We consider three citation network datasets: Citeseer, Cora and Pubmed (Sen et al., 2008). The datasets contain sparse bag-of-words feature vectors for each document and a list of citation links between documents. We treat the citation links as (undirected) edges and construct a binary, symmetric adjacency matrix A . Each document has a class label. For training, we only use 20 labels per class, but all feature vectors.

NELL NELL is a dataset extracted from the knowledge graph introduced in (Carlson et al., 2010). A knowledge graph is a set of entities connected with directed, labeled edges (relations). We follow the pre-processing scheme as described in Yang et al. (2016). We assign separate relation nodes r_1 and r_2 for each entity pair (e_1, r, e_2) as (e_1, r_1) and (e_2, r_2) . Entity nodes are described by sparse feature vectors. We extend the number of features in NELL by assigning a unique one-hot representation for every relation node, effectively resulting in a 61,278-dim sparse feature vector per node. The semi-supervised task here considers the extreme case of only a single labeled example per class in the training set. We construct a binary, symmetric adjacency matrix from this graph by setting entries $A_{ij} = 1$, if one or more edges are present between nodes i and j .

Random graphs We simulate random graph datasets of various sizes for experiments where we measure training time per epoch. For a dataset with N nodes we create a random graph assigning $2N$ edges uniformly at random. We take the identity matrix I_N as input feature matrix X , thereby implicitly taking a featureless approach where the model is only informed about the identity of each node, specified by a unique one-hot vector. We add dummy labels $Y_i = 1$ for every node.

5.2 EXPERIMENTAL SET-UP

Unless otherwise noted, we train a two-layer GCN as described in Section 3.1 and evaluate prediction accuracy on a test set of 1,000 labeled examples. We provide additional experiments using deeper models with up to 10 layers in Appendix B. We choose the same dataset splits as in Yang et al. (2016) with an additional validation set of 500 labeled examples for hyperparameter optimization (dropout rate for all layers, L2 regularization factor for the first GCN layer and number of hidden units). We do not use the validation set labels for training.

For the citation network datasets, we optimize hyperparameters on Cora only and use the same set of parameters for Citeseer and Pubmed. We train all models for a maximum of 200 epochs (training iterations) using Adam (Kingma & Ba, 2015) with a learning rate of 0.01 and early stopping with a window size of 10, i.e. we stop training if the validation loss does not decrease for 10 consecutive epochs. We initialize weights using the initialization described in Glorot & Bengio (2010) and accordingly (row-)normalize input feature vectors. On the random graph datasets, we use a hidden layer size of 32 units and omit regularization (i.e. neither dropout nor L2 regularization).

5.3 BASELINES

We compare against the same baseline methods as in Yang et al. (2016), i.e. label propagation (LP) (Zhu et al., 2003), semi-supervised embedding (SemiEmb) (Weston et al., 2012), manifold regularization (ManiReg) (Belkin et al., 2006) and skip-gram based graph embeddings (DeepWalk) (Perozzi et al., 2014). We omit TSVM (Joachims, 1999), as it does not scale to the large number of classes in one of our datasets.

We further compare against the iterative classification algorithm (ICA) proposed in Lu & Getoor (2003) in conjunction with two logistic regression classifiers, one for local node features alone and one for relational classification using local features and an aggregation operator as described in Sen et al. (2008). We first train the local classifier using all labeled training set nodes and use it to bootstrap class labels of unlabeled nodes for relational classifier training. We run iterative classification (relational classifier) with a random node ordering for 10 iterations on all unlabeled nodes (bootstrapped using the local classifier). L2 regularization parameter and aggregation operator (*count* vs. *prop*, see Sen et al. (2008)) are chosen based on validation set performance for each dataset separately.

Lastly, we compare against Planetoid (Yang et al., 2016), where we always choose their best-performing model variant (transductive vs. inductive) as a baseline.

6 RESULTS

6.1 SEMI-SUPERVISED NODE CLASSIFICATION

Results are summarized in Table 2. Reported numbers denote classification accuracy in percent. For ICA, we report the mean accuracy of 100 runs with random node orderings. Results for all other baseline methods are taken from the Planetoid paper (Yang et al., 2016). Planetoid* denotes the best model for the respective dataset out of the variants presented in their paper.

Table 2: Summary of results in terms of classification accuracy (in percent).

Method	Citeseer	Cora	Pubmed	NELL
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
LP [32]	45.3	68.0	63.0	26.5
DeepWalk [22]	43.2	67.2	65.3	58.1
ICA [18]	69.1	75.1	73.9	23.1
Planetoid* [29]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	70.3 (7s)	81.5 (4s)	79.0 (38s)	66.0 (48s)
GCN (rand. splits)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

We further report wall-clock training time in seconds until convergence (in brackets) for our method (incl. evaluation of validation error) and for Planetoid. For the latter, we used an implementation provided by the authors³ and trained on the same hardware (with GPU) as our GCN model. We trained and tested our model on the same dataset splits as in Yang et al. (2016) and report mean accuracy of 100 runs with random weight initializations. We used the following sets of hyperparameters for Citeseer, Cora and Pubmed: 0.5 (dropout rate), $5 \cdot 10^{-4}$ (L2 regularization) and 16 (number of hidden units); and for NELL: 0.1 (dropout rate), $1 \cdot 10^{-5}$ (L2 regularization) and 64 (number of hidden units).

In addition, we report performance of our model on 10 randomly drawn dataset splits of the same size as in Yang et al. (2016), denoted by GCN (rand. splits). Here, we report mean and standard error of prediction accuracy on the test set split in percent.

6.2 EVALUATION OF PROPAGATION MODEL

We compare different variants of our proposed per-layer propagation model on the citation network datasets. We follow the experimental set-up described in the previous section. Results are summarized in Table 3. The propagation model of our original GCN model is denoted by *renormalization trick* (in bold). In all other cases, the propagation model of both neural network layers is replaced with the model specified under *propagation model*. Reported numbers denote mean classification accuracy for 100 repeated runs with random weight matrix initializations. In case of multiple variables Θ_i per layer, we impose L2 regularization on all weight matrices of the first layer.

Table 3: Comparison of propagation models.

Description	Propagation model	Citeseer	Cora	Pubmed
Chebyshev filter (Eq. 5) $K = 3$	$\sum_{k=0}^K T_k(\tilde{L})X\Theta_k$	69.8	79.5	74.4
Chebyshev filter (Eq. 5) $K = 2$	$\sum_{k=0}^K T_k(\tilde{L})X\Theta_k$	69.6	81.2	73.8
1 st -order model (Eq. 6)	$X\Theta_0 + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}X\Theta_1$	68.3	80.0	77.5
Single parameter (Eq. 7)	$(I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}})X\Theta$	69.3	79.2	77.4
Renormalization trick (Eq. 8)	$\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X\Theta$	70.3	81.5	79.0
1 st -order term only	$D^{-\frac{1}{2}}AD^{-\frac{1}{2}}X\Theta$	68.7	80.5	77.8
Multi-layer perceptron	$X\Theta$	46.5	55.1	71.4

³<https://github.com/kimiyoung/planetoid>

6.3 TRAINING TIME PER EPOCH

Here, we report results for the mean training time per epoch (forward pass, cross-entropy calculation, backward pass) for 100 epochs on simulated random graphs, measured in seconds wall-clock time. See Section 5.1 for a detailed description of the random graph dataset used in these experiments. We compare results on a GPU and on a CPU-only implementation⁴ in TensorFlow (Abadi et al., 2015). Figure 2 summarizes the results.

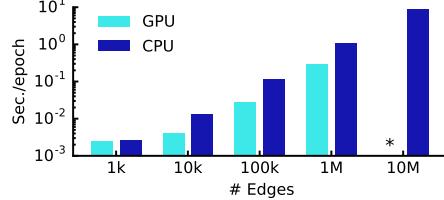


Figure 2: Wall-clock time per epoch for random graphs. (*) indicates out-of-memory error.

7 DISCUSSION

7.1 SEMI-SUPERVISED MODEL

In the experiments demonstrated here, our method for semi-supervised node classification outperforms recent related methods by a significant margin. Methods based on graph-Laplacian regularization (Zhu et al., 2003; Belkin et al., 2006; Weston et al., 2012) are most likely limited due to their assumption that edges encode mere similarity of nodes. Skip-gram based methods on the other hand are limited by the fact that they are based on a multi-step pipeline which is difficult to optimize. Our proposed model can overcome both limitations, while still comparing favorably in terms of efficiency (measured in wall-clock time) to related methods. Propagation of feature information from neighboring nodes in every layer improves classification performance in comparison to methods like ICA (Lu & Getoor, 2003), where only label information is aggregated.

We have further demonstrated that the proposed renormalized propagation model (Eq. 8) offers both improved efficiency (fewer parameters and operations, such as multiplication or addition) and better predictive performance on a number of datasets compared to a naïve 1st-order model (Eq. 6) or higher-order graph convolutional models using Chebyshev polynomials (Eq. 5).

7.2 LIMITATIONS AND FUTURE WORK

Here, we describe several limitations of our current model and outline how these might be overcome in future work.

Memory requirement In the current setup with full-batch gradient descent, memory requirement grows linearly in the size of the dataset. We have shown that for large graphs that do not fit in GPU memory, training on CPU can still be a viable option. Mini-batch stochastic gradient descent can alleviate this issue. The procedure of generating mini-batches, however, should take into account the number of layers in the GCN model, as the K^{th} -order neighborhood for a GCN with K layers has to be stored in memory for an exact procedure. For very large and densely connected graph datasets, further approximations might be necessary.

Directed edges and edge features Our framework currently does not naturally support edge features and is limited to undirected graphs (weighted or unweighted). Results on NELL however show that it is possible to handle both directed edges and edge features by representing the original directed graph as an undirected bipartite graph with additional nodes that represent edges in the original graph (see Section 5.1 for details).

Limiting assumptions Through the approximations introduced in Section 2, we implicitly assume locality (dependence on the K^{th} -order neighborhood for a GCN with K layers) and equal importance of self-connections vs. edges to neighboring nodes. For some datasets, however, it might be beneficial to introduce a trade-off parameter λ in the definition of \tilde{A} :

$$\tilde{A} = A + \lambda I_N. \quad (11)$$

⁴Hardware used: 16-core Intel® Xeon® CPU E5-2640 v3 @ 2.60GHz, GeForce® GTX TITAN X

This parameter now plays a similar role as the trade-off parameter between supervised and unsupervised loss in the typical semi-supervised setting (see Eq. 1). Here, however, it can be learned via gradient descent.

8 CONCLUSION

We have introduced a novel approach for semi-supervised classification on graph-structured data. Our GCN model uses an efficient layer-wise propagation rule that is based on a first-order approximation of spectral convolutions on graphs. Experiments on a number of network datasets suggest that the proposed GCN model is capable of encoding both graph structure and node features in a way useful for semi-supervised classification. In this setting, our model outperforms several recently proposed methods by a significant margin, while being computationally efficient.

ACKNOWLEDGMENTS

We would like to thank Christos Louizos, Taco Cohen, Joan Bruna, Zhilin Yang, Dave Herman, Pramod Sinha and Abdul-Saboor Sheikh for helpful discussions. This research was funded by SAP.

REFERENCES

- Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.
- James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, 2016.
- Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research (JMLR)*, 7(Nov):2399–2434, 2006.
- Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2008.
- Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR)*, 2014.
- Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr, and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, pp. 3, 2010.
- Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems (NIPS)*, 2016.
- Brendan L. Douglas. The Weisfeiler-Lehman method and graph isomorphism testing. *arXiv preprint arXiv:1101.5211*, 2011.
- David K. Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems (NIPS)*, pp. 2224–2232, 2015.
- Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, volume 9, pp. 249–256, 2010.
- Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks.*, volume 2, pp. 729–734. IEEE, 2005.
- Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.

- David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- Thorsten Joachims. Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning (ICML)*, volume 99, pp. 200–209, 1999.
- Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
- Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.
- Qing Lu and Lise Getoor. Link-based classification. In *International Conference on Machine Learning (ICML)*, volume 3, pp. 496–503, 2003.
- Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, 9(Nov):2579–2605, 2008.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems (NIPS)*, pp. 3111–3119, 2013.
- Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International Conference on Machine Learning (ICML)*, 2016.
- Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710. ACM, 2014.
- Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.
- Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.
- Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958, 2014.
- Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 1067–1077. ACM, 2015.
- Boris Weisfeiler and A. A. Lehmann. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2(9):12–16, 1968.
- Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pp. 639–655. Springer, 2012.
- Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning (ICML)*, 2016.
- Wayne W. Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pp. 452–473, 1977.
- Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems (NIPS)*, volume 16, pp. 321–328, 2004.
- Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *International Conference on Machine Learning (ICML)*, volume 3, pp. 912–919, 2003.

A RELATION TO WEISFEILER-LEHMAN ALGORITHM

A neural network model for graph-structured data should ideally be able to learn representations of nodes in a graph, taking both the graph structure and feature description of nodes into account. A well-studied framework for the unique assignment of node labels given a graph and (optionally) discrete initial node labels is provided by the 1-dim Weisfeiler-Lehman (WL-1) algorithm (Weisfeiler & Lehmann, 1968):

Algorithm 1: WL-1 algorithm (Weisfeiler & Lehmann, 1968)

Input: Initial node coloring $(h_1^{(0)}, h_2^{(0)}, \dots, h_N^{(0)})$
Output: Final node coloring $(h_1^{(T)}, h_2^{(T)}, \dots, h_N^{(T)})$
 $t \leftarrow 0;$
repeat
 | **for** $v_i \in \mathcal{V}$ **do**
 | | $h_i^{(t+1)} \leftarrow \text{hash} \left(\sum_{j \in \mathcal{N}_i} h_j^{(t)} \right);$
 | | $t \leftarrow t + 1;$
until stable node coloring is reached;

Here, $h_i^{(t)}$ denotes the coloring (label assignment) of node v_i (at iteration t) and \mathcal{N}_i is its set of neighboring node indices (irrespective of whether the graph includes self-connections for every node or not). $\text{hash}(\cdot)$ is a hash function. For an in-depth mathematical discussion of the WL-1 algorithm see, e.g., Douglas (2011).

We can replace the hash function in Algorithm 1 with a neural network layer-like differentiable function with trainable parameters as follows:

$$h_i^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}_i} \frac{1}{c_{ij}} h_j^{(l)} W^{(l)} \right), \quad (12)$$

where c_{ij} is an appropriately chosen normalization constant for the edge (v_i, v_j) . Further, we can take $h_i^{(l)}$ now to be a vector of *activations* of node i in the l^{th} neural network layer. $W^{(l)}$ is a layer-specific weight matrix and $\sigma(\cdot)$ denotes a differentiable, non-linear activation function.

By choosing $c_{ij} = \sqrt{d_i d_j}$, where $d_i = |\mathcal{N}_i|$ denotes the degree of node v_i , we recover the propagation rule of our Graph Convolutional Network (GCN) model in vector form (see Eq. 2)⁵.

This—loosely speaking—allows us to interpret our GCN model as a differentiable and parameterized generalization of the 1-dim Weisfeiler-Lehman algorithm on graphs.

A.1 NODE EMBEDDINGS WITH RANDOM WEIGHTS

From the analogy with the Weisfeiler-Lehman algorithm, we can understand that even an untrained GCN model with random weights can serve as a powerful feature extractor for nodes in a graph. As an example, consider the following 3-layer GCN model:

$$Z = \tanh \left(\hat{A} \tanh \left(\hat{A} \tanh \left(\hat{A} X W^{(0)} \right) W^{(1)} \right) W^{(2)} \right), \quad (13)$$

with weight matrices $W^{(l)}$ initialized at random using the initialization described in Glorot & Bengio (2010). \hat{A} , X and Z are defined as in Section 3.1.

We apply this model on Zachary’s karate club network (Zachary, 1977). This graph contains 34 nodes, connected by 154 (undirected and unweighted) edges. Every node is labeled by one of four classes, obtained via modularity-based clustering (Brandes et al., 2008). See Figure 3a for an illustration.

⁵Note that we here implicitly assume that self-connections have already been added to every node in the graph (for a clutter-free notation).

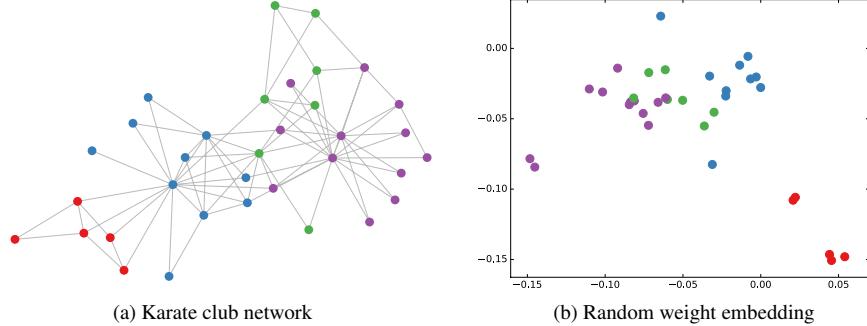


Figure 3: *Left:* Zachary’s karate club network (Zachary, 1977), colors denote communities obtained via modularity-based clustering (Brandes et al., 2008). *Right:* Embeddings obtained from an untrained 3-layer GCN model (Eq. 13) with random weights applied to the karate club network. Best viewed on a computer screen.

We take a featureless approach by setting $X = I_N$, where I_N is the N by N identity matrix. N is the number of nodes in the graph. Note that nodes are randomly ordered (i.e. ordering contains no information). Furthermore, we choose a hidden layer dimensionality⁶ of 4 and a two-dimensional output (so that the output can immediately be visualized in a 2-dim plot).

Figure 3b shows a representative example of node embeddings (outputs Z) obtained from an untrained GCN model applied to the karate club network. These results are comparable to embeddings obtained from DeepWalk (Perozzi et al., 2014), which uses a more expensive unsupervised training procedure.

A.2 SEMI-SUPERVISED NODE EMBEDDINGS

On this simple example of a GCN applied to the karate club network it is interesting to observe how embeddings react during training on a semi-supervised classification task. Such a visualization (see Figure 4) provides insights into how the GCN model can make use of the graph structure (and of features extracted from the graph structure at later layers) to learn embeddings that are useful for a classification task.

We consider the following semi-supervised learning setup: we add a softmax layer on top of our model (Eq. 13) and train using only a single labeled example per class (i.e. a total number of 4 labeled nodes). We train for 300 training iterations using Adam (Kingma & Ba, 2015) with a learning rate of 0.01 on a cross-entropy loss.

Figure 4 shows the evolution of node embeddings over a number of training iterations. The model succeeds in linearly separating the communities based on minimal supervision and the graph structure alone. A video of the full training process can be found on our website⁷.

⁶We originally experimented with a hidden layer dimensionality of 2 (i.e. same as output layer), but observed that a dimensionality of 4 resulted in less frequent saturation of $\tanh(\cdot)$ units and therefore visually more pleasing results.

⁷<http://tkipf.github.io/graph-convolutional-networks/>

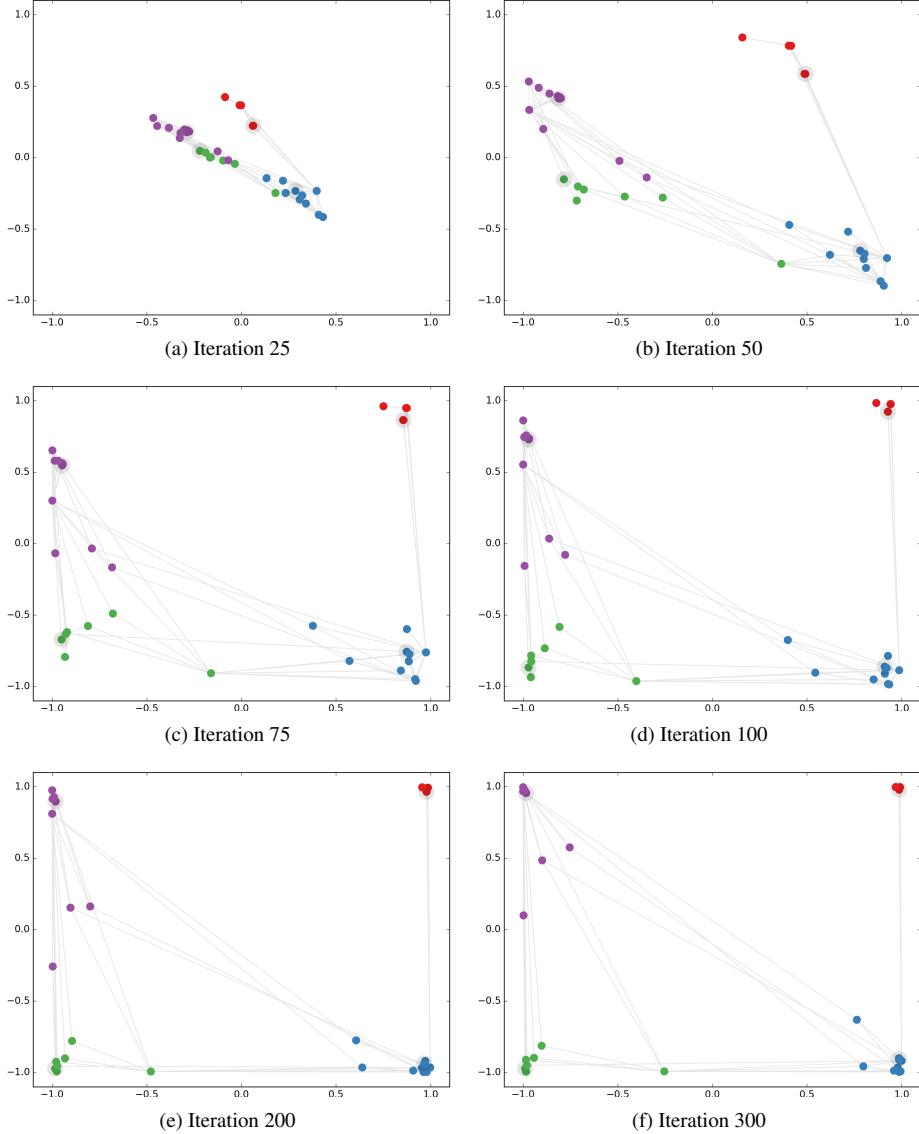


Figure 4: Evolution of karate club network node embeddings obtained from a GCN model after a number of semi-supervised training iterations. Colors denote class. Nodes of which labels were provided during training (one per class) are highlighted (grey outline). Grey links between nodes denote graph edges. Best viewed on a computer screen.

B EXPERIMENTS ON MODEL DEPTH

In these experiments, we investigate the influence of model depth (number of layers) on classification performance. We report results on a 5-fold cross-validation experiment on the Cora, Citeseer and Pubmed datasets (Sen et al., 2008) using all labels. In addition to the standard GCN model (Eq. 2), we report results on a model variant where we use residual connections (He et al., 2016) between hidden layers to facilitate training of deeper models by enabling the model to carry over information from the previous layer’s input:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) + H^{(l)}. \quad (14)$$

On each cross-validation split, we train for 400 epochs (without early stopping) using the Adam optimizer (Kingma & Ba, 2015) with a learning rate of 0.01. Other hyperparameters are chosen as follows: 0.5 (dropout rate, first and last layer), $5 \cdot 10^{-4}$ (L2 regularization, first layer), 16 (number of units for each hidden layer) and 0.01 (learning rate). Results are summarized in Figure 5.

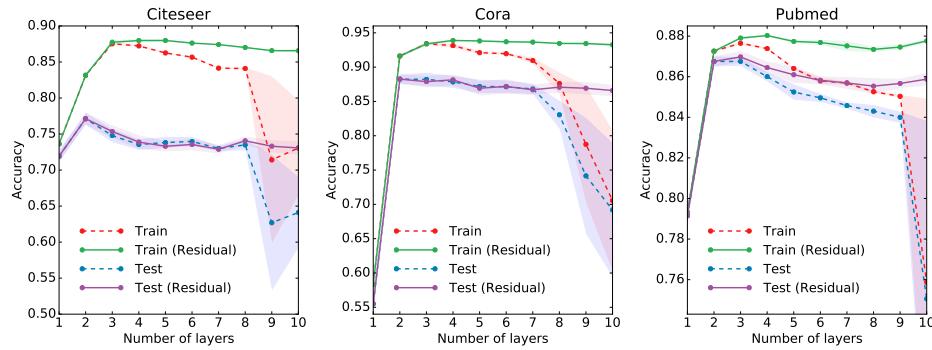


Figure 5: Influence of model depth (number of layers) on classification performance. Markers denote mean classification accuracy (training vs. testing) for 5-fold cross-validation. Shaded areas denote standard error. We show results both for a standard GCN model (dashed lines) and a model with added residual connections (He et al., 2016) between hidden layers (solid lines).

For the datasets considered here, best results are obtained with a 2- or 3-layer model. We observe that for models deeper than 7 layers, training without the use of residual connections can become difficult, as the effective context size for each node increases by the size of its K^{th} -order neighborhood (for a model with K layers) with each additional layer. Furthermore, overfitting can become an issue as the number of parameters increases with model depth.

Christian Hennig

Material list:

P. Coretto and C. Hennig (2024) Nonparametric consistency for maximum likelihood estimation and clustering based on mixtures of elliptically-symmetric distributions. arXiv:2311.06108.

NONPARAMETRIC CONSISTENCY FOR MAXIMUM LIKELIHOOD ESTIMATION AND CLUSTERING BASED ON MIXTURES OF ELLIPTICALLY-SYMMETRIC DISTRIBUTIONS

Pietro Coretto* Christian Hennig†

Monday 29th April, 2024

Abstract

The consistency of the maximum likelihood estimator for mixtures of elliptically-symmetric distributions for estimating its population version is shown, where the underlying distribution P is nonparametric and does not necessarily belong to the class of mixtures on which the estimator is based. In a situation where P is a mixture of well enough separated but nonparametric distributions it is shown that the components of the population version of the estimator correspond to the well separated components of P . This provides some theoretical justification for the use of such estimators for cluster analysis in case that P has well separated subpopulations even if these subpopulations differ from what the mixture model assumes.

Keywords. Asymptotic analysis, finite mixture models, model-based clustering, canonical functional.

1. Introduction

Clustering methods are widely used in statistics, computer science, and many applied fields of science to discover group structures and to deal with the intrinsic inhomogeneity of complex data sets. While there is abundant work on methodology, algorithms, and applications, a smaller body of literature has investigated the relationship between the clusters found by a method and the underlying data-generating mechanism. Assuming that the observed data set is generated by independent and identical observations from a probability law P , consistency concerns the relationship between P and the outcome of a method for random samples of a size converging to infinity. In cluster analysis, the clustering itself and/or distributional parameters characterising the clustering may be of interest.

Here we will derive consistency results for model-based clustering, i.e., clustering based on probability mixture models. More precisely, the results will concern maximum likelihood (ML) estimators (MLE) of finite mixtures of distributions from elliptically symmetrical distribution (ESD) families such as the Gaussian distribution. Finite mixture models (FMM) are convex combinations of probability distributions suitable

*Department of Economics and Statistics; University of Salerno (Italy) – E-mail: pcoretto@unisa.it.
For this research the author was supported by Ministero dell'Università e della Ricerca (MUR, Italy), grant number 20223725WE.

†Department of Statistics; University of Bologna (Italy) – E-mail: christian.hennig@unibo.it

to represent inhomogeneous populations. FMMs are a versatile tool for modelling complex distributions, and are at the basis of a variety of data analysis, prediction, and inference tasks: density estimation, regression, clustering, and classification (see [Frühwirth-Schnatter et al., 2019](#); [Hennig et al., 2016](#)).

Given K , the number of mixture components, FMMs are represented by parameters characterising the individual mixture components (such as Gaussian mean and variance) and the component proportions. MLEs can be obtained in practice using the Expectation-Maximisation (EM) algorithm of [Dempster et al. \(1977\)](#). Given the fitted parameters, a partition of the data can be obtained by use of the maximum posterior assignment rule, see Section [2.1](#).

In clustering, clusters are usually interpreted as corresponding to the estimated mixture components, although this is not always appropriate if different mixture components are not well separated, see [Hennig \(2010\)](#).

FMMs are parametric, and therefore, as a standard, statisticians are interested in whether the parameters can be consistently estimated if the true underlying P is indeed such a mixture. Consistency theory for ML estimation of mixture parameters is not as easy to obtain as for standard ML estimation, and standard conditions such as required by [Wald \(1949\)](#) are not fulfilled. [Kiefer and Wolfowitz \(1956\)](#) noted that the likelihood function of univariate Gaussian mixtures is unbounded if one of the mixture components' variances is allowed to converge to zero from above. This issue occurs for many classes of FMMs including those of ESDs as treated here. There have been several proposals to solve it. A popular strategy, originally due to [Dennis \(1981\)](#), is to impose an "eigen-ratio constraint" (ERC). This is a constant that bounds the ratio between the largest and smallest eigenvalue across all components' scatter matrices. The ERC allows for a non-compact parameter space; it is easier but also more restrictive to constrain the parameter space to be compact as done, e.g., by [Redner \(1981\)](#), who proved consistency for the MLE of a general class of FMMs. [Hathaway \(1985\)](#) expanded the work of [Redner \(1981\)](#) including the ERC in the case of a univariate Gaussian mixture. There is a long history of ML-theory for mixtures. For reviews see [Redner and Walker \(1984\)](#); [Chen \(2017\)](#).

Here, however, we focus on nonparametric consistency, i.e., we ask what happens if P is general and not necessarily of the assumed mixture type. Note that we will treat the number of estimated mixture components K as fixed, so that the fitted mixture cannot approximate almost any P by selecting a sufficiently large K . The results will apply to a P that is indeed a mixture of the assumed family with K mixture components, however the underlying P could also have fewer or more than K mixture components of this kind (in clustering it may sometimes be desirable to fit a mixture with more than K not well separated components by a K -component mixture where the K components are better separated and justified to be interpreted as clusters, see, e.g., [Biernacki et al. \(2000\)](#)), or it may not be possible to represent it as a mixture of the assumed type at all.

In practice, arguably, parametric statistical model assumptions are never precisely fulfilled, yet often it is claimed that the application of data analytic methods based on parametric models require the parametric model assumptions to be fulfilled. In fact often standard theory relies on these model assumptions, and it may be suspected that if assumptions are violated, the method may not achieve the performance that is theoretically guaranteed assuming the model. Nonparametric theory as derived here, even if only asymptotic, will apply to and justify the use of such methods more generally.

Nonparametric consistency results for cluster analysis go back to the seminal works

of [Pollard \(1981\)](#) (for K -means clustering) and [Hartigan \(1981\)](#) (for single linkage). Further such results have been provided by [Sriperumbudur and Steinwart \(2012\)](#) for the DBSCAN algorithm and [von Luxburg et al. \(2008\)](#) for spectral clustering, among others. Multivariate Gaussian FMMs have been studied by [García-Escudero et al. \(2015\)](#) and [Coretto and Hennig \(2017\)](#). The latter paper also has a nonparametric consistency result regarding a robust ML-type method accommodating outliers by incorporating an improper uniform mixture component covering the whole data space, and an Expectation-Conditional Maximization (ECM) algorithm implementing the ERC.

There are two aspects of nonparametric consistency relevant in clustering (cp. [von Luxburg et al. \(2008\)](#)).

- (i) Is the clustering method consistent at all, i.e., is there a limit clustering (or limit parameters characterising it) if the sample size grows larger and larger? Many clustering methods including MLE of FMMs are defined by parameters optimising an objective function. The asymptotic limit is then usually a functional defined on P by generalizing the objective function to general distributions (“population version” or “canonical functional”) such as in [Pollard \(1981\)](#); [García-Escudero et al. \(2015\)](#); [Coretto and Hennig \(2017\)](#) and also in the present work.
- (ii) Provided that a limit clustering exists, does this correspond to a reasonable clustering structure given P that is of interest when doing cluster analysis?

Addressing these aspects, the present paper provides two main contributions:

- (i) We establish the existence and the consistency of the MLE for a general class of FMMs based on ESDs for a general nonparametric P , generalizing the results for Gaussian FMMs in [García-Escudero et al. \(2015\)](#); [Coretto and Hennig \(2017\)](#). This implies finite sample existence of the MLE under mild conditions.
- (ii) Under the additional assumption that P is a mixture of K nonparametric components that put most of their probability mass on sufficiently well separated subsets of the data space, we show that the components of the canonical functional resulting from the MLE of FMMs of ESDs correspond to the well separated mixture components of P . This can be interpreted as follows. If P is a distribution generating well enough separated clusters of a very flexible kind, particularly not necessarily corresponding to the ESDs assumed to be the components of the FMM estimated by the MLE, the MLE will anyway for large enough data recover these clusters. We are not aware of any result of this kind in the literature regarding nonparametric consistency based on canonical functionals.

The paper is organized as follows. In Section 2, we define the class of ESDs and the corresponding FMMs. We review the connection between model-based clustering and FMMs. We also define the MLE and give an account of the issue of unbounded likelihood. Finally we introduce and motivate the regularization of the ML optimization problem based on the ERC. Section 3 is devoted to the finite sample analysis of the ML procedure. In Section 4, we establish the MLE’s existence and consistency, assuming that P is some general nonparametric distribution. Section 5 investigates the canonical functional corresponding to the MLE for FMMs of ESDs for distributions P that can be written as mixtures of sufficiently well separated nonparametric components. In Section 6 we present numerical experiments that illustrate the results. Section 7 concludes the paper.

2. Finite mixture modeling with elliptically symmetric distributions

Elliptically symmetric distributions can be obtained applying an affine transformation to a spherical distribution. Let the random vector $Y \in \mathbb{R}^p$ have a spherical distribution. With fixed $\boldsymbol{\mu} \in \mathbb{R}^p$, and $\boldsymbol{\Omega} \in \mathbb{R}^{p \times p}$, the random vector $X = \boldsymbol{\mu} + \boldsymbol{\Omega}Y$ is said to have an elliptically-symmetric distribution denoted by $X \sim \text{ESD}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} := \boldsymbol{\Omega}\boldsymbol{\Omega}^\top$. With a fixed distribution of the generating Y , $\text{ESD}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ forms a symmetric location-scatter class of models with location parameter $\boldsymbol{\mu}$ and a symmetric positive semi-definite scatter matrix parameter $\boldsymbol{\Sigma}$. If $E[Y] = \mathbf{0}$ and $\text{Var}[Y] \propto \mathbf{I}_p$, the $p \times p$ -unit matrix, then $\boldsymbol{\mu} = E[X]$ and $\boldsymbol{\Sigma} \propto \text{Var}[X]$. If Y is absolutely continuous and $\boldsymbol{\Sigma}$ is non-singular, than X is absolutely continuous with density function

$$f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) := \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} g\left((\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right), \quad (1)$$

where $g : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is the so-called radial or density generator function. Here we assume that $g(\cdot)$ is monotonically non-increasing, which implies that $f(\cdot)$ is unimodal. $\lim_{t \rightarrow +\infty} g(t) = 0$ is required so that $f(\cdot)$ is a proper density.

The choice of $g(\cdot)$ (or, equivalently, the distribution of Y) defines a specific family of ESDs. A number of popular models can be obtained in this way, for example multivariate Gaussian distributions, multivariate Student-t distributions and multivariate logistic distributions. Some families of ESDs have parameters in addition to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, e.g. the degrees of freedom of the Student-t. We do not involve such additional parameters, so the results given here apply to MLEs for a family of multivariate Student-t distribution with fixed degrees of freedom, and generally to any location-scale family as defined in eq. (1) with g fulfilling the conditions above.

[Kelker \(1970\)](#) originally introduced elliptical distributions to generalize the multivariate Gaussian model. ESDs have elliptically shaped density contours. They can model joint linear dependence between the p components of a random vector and possibly heavy tails. For these reasons, ESDs occur frequently in the theory of statistics and applications (see [Genton, 2004](#), for a comprehensive overview). Finite mixture densities based on eq. (1) are the convex combinations

$$\psi(\mathbf{x}; \boldsymbol{\theta}) := \sum_{k=1}^K \pi_k f(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad (2)$$

where $\pi_k \in [0, 1]$ so that $\sum_{k=1}^K \pi_k = 1$, and $\boldsymbol{\theta}$ is a parameter vector collecting all elements of $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$, $k = 1, 2, \dots, K$. Given a family of ESDs, an FMM is defined by the set of distributions for all possible choices of $\boldsymbol{\theta}$. FMMs are popular tools for modeling multimodality, skewness and heterogenous populations, and for performing several supervised and unsupervised tasks such as semi-parametric density estimation, classification and clustering ([McLachlan and Peel, 2000](#); [Frühwirth-Schnatter et al., 2019](#)).

2.1. Clustering and classification. FMMs used for model-based clustering and classification normally identify the classes or clusters with the mixture components. FMMs can be equivalently formulated involving component membership indicators for the observations.

Assume that there are K mixture components. Let X_1, \dots, X_n model a sequence of i.i.d. observations. These are accompanied by a sequence of 0-1 valued i.i.d. random

variables $\zeta_i = (Z_{i1}, Z_{i2}, \dots, Z_{iK})^\top$, $i = 1, \dots, n$, with $\sum_{k=1}^K Z_{ik} = 1$, i.e., for given i one of the Z_{ik} is 1, all the others are 0. Assume that ζ_i has a categorical distribution with $\Pr[Z_{ik} = 1] = \text{E}[Z_{ik} = 1] = \pi_k$ for $k = 1, 2, \dots, K$. Let eq. (1) be the density function of the conditional distribution of $X_i | Z_{ik} = 1$, then eq. (2) is the density function of the unconditional distribution of X_i . Z_{ik} is the component (cluster) membership indicator for observation i and the k th mixture component. eq. (2) can be seen as a model generating an expected fraction π_k of points from its k -th component $f(\cdot; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. If the K mixture components are reasonably separated, sampling from eq. (2) will generate clustered regions of data points. Cluster analysis is unsupervised, i.e., ζ_1, \dots, ζ_n are unobserved.

ζ_1, \dots, ζ_n represent a partition $\mathcal{G}_K := \{G_k; k = 1, 2, \dots, K\}$ of $\{1, \dots, n\}$, where $Z_{ik} := \mathbb{I}\{i \in G_k\}$, $\mathbb{I}\{\cdot\}$ being the usual indicator function, meaning that the mixture component (cluster) having generated X_i is k .

In cluster analysis one wants to assign an object to one of the K groups of \mathcal{G}_K , which for $i = 1, \dots, n$ amounts to predicting ζ_i from X_i .

This can be done based on the posterior probability

$$\tau_k(\mathbf{x}; \boldsymbol{\theta}) := \Pr[Z_{ik} = 1 | X_i = \mathbf{x}] = \frac{\pi_k f(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\psi(\mathbf{x}; \boldsymbol{\theta})}. \quad (3)$$

Given a parameter vector $\boldsymbol{\theta}$, an object \mathbf{x} can be assigned to the cluster $\text{cl}(\mathbf{x}, \boldsymbol{\theta}) \in \{1, 2, \dots, K\}$ according to the “*maximum a posteriori probability*” (MAP) rule:

$$\text{cl}(\mathbf{x}, \boldsymbol{\theta}) := \arg \max_{k \in \{1, 2, \dots, K\}} \tau_k(\mathbf{x}; \boldsymbol{\theta}). \quad (4)$$

The MAP rule implements the optimal Bayes classifier, i.e. the assignment that minimizes the expected 0-1 loss, also known as misclassification rate. Since $\boldsymbol{\theta}$ is not known, it has to be estimated from the data, and eq. (4) is then computed based on the estimator $\hat{\boldsymbol{\theta}}$.

Remark 1. The optimality of the MAP rule for the clustering problem requires that (i) data are generated from eq. (2), (ii) “true” clusters are defined in terms of $Z_{ik} = \mathbb{I}\{i \in G_k\}$, that is, $f(\cdot; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the “true” underlying k -th class-conditional density, and (iii) the posterior ratios in eq. (3) are computed at the “true” generating parameter $\boldsymbol{\theta}$ (the optimality would hold asymptotically with a consistent estimator for the “true” $\boldsymbol{\theta}$). In practice, these conditions are arguably never fulfilled ((ii) is a matter of definition, but relies on (i)).

Applying this approach to data generated from a general distribution P that does not necessarily have a density of type eq. (2) means that the method imposes a partition on P that is governed by “cluster prototype densities” of the form eq. (1). Here we investigate what happens then, at least asymptotically.

2.2. Maximum likelihood estimator. The unknown mixture parameter vector $\boldsymbol{\theta}$ is typically estimated by ML. Often, computations are performed based on the Expectation-Maximization (EM) algorithm (McLachlan and Peel, 2000). Let $\mathbb{X}_n = \{\mathbf{x}_i, i = 1, 2, \dots, n\}$ be the observed sample. Define the sample likelihood and log-likelihood functions

$$\mathcal{L}_n(\boldsymbol{\theta}) := \prod_{i=1}^n \psi(\mathbf{x}_i; \boldsymbol{\theta}), \quad \ell_n(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^n \log(\psi(\mathbf{x}_i; \boldsymbol{\theta})). \quad (5)$$

Finding the maximum of eq. (5), the sample MLE, is not straightforward. In fact, eq. (5) does not have a maximum. [Kiefer and Wolfowitz \(1956\)](#) discovered the unboundedness of ℓ_n in the context of univariate Gaussian FMM. The issue extends to many classes of FMMs, including those studied here. To see why, let us fix additional notations also used throughout the rest of the paper. Let $\|\cdot\|$ be the Euclidean norm. If the parameter vectors $\boldsymbol{\theta}$ come with indexes and accents, its members $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ have the same indexes and accents, i.e., $\tilde{\boldsymbol{\theta}}_m$ contains $(\tilde{\pi}_{mk}, \tilde{\boldsymbol{\mu}}_{mk}, \tilde{\boldsymbol{\Sigma}}_{mk})$ for all $m \in \mathbb{N}$. For given $\boldsymbol{\theta}$, let $\boldsymbol{\kappa}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ (indexes and accents are applied to $\boldsymbol{\kappa}$ as above). Scalar parameters contained in $\boldsymbol{\theta}$'s sub-vectors are indexed so that the first two subscripts always denote the mixture component and the dimension in the feature space respectively, e.g. $\mu_{k,j} \in \mathbb{R}$ is the j -th coordinate of $\boldsymbol{\mu}_k$.

Consider a sequence $(\boldsymbol{\theta}_m)_{m \in \mathbb{N}}$ where $\boldsymbol{\mu}_{1,m} = \mathbf{x}_1 \in \mathbb{X}_n$ and the smallest eigenvalue of $\boldsymbol{\Sigma}_{1,m}$ converges to 0 as $m \rightarrow +\infty$, then $\ell_n(\boldsymbol{\theta}_m) \rightarrow +\infty$. The likelihood degeneracy is caused by the fact that the density peak of $f(\cdot)$ is controlled by the smallest eigenvalue of the scatter matrix. In fact, eq. (1) can be parameterized in terms of the eigenvalue decomposition of $\boldsymbol{\Sigma}$. That is

$$f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \left(\prod_{j=1}^p \lambda_j(\boldsymbol{\Sigma}) \right)^{-\frac{1}{2}} g \left(\sum_{j=1}^p \lambda_j(\boldsymbol{\Sigma})^{-1} (\mathbf{x} - \boldsymbol{\mu})^\top V_j(\boldsymbol{\Sigma}) V_j(\boldsymbol{\Sigma})^\top (\mathbf{x} - \boldsymbol{\mu}) \right), \quad (6)$$

where $\lambda_j(\boldsymbol{\Sigma})$ is the j -th eigenvalue of $\boldsymbol{\Sigma}$, and $V_j(\boldsymbol{\Sigma})$ is its corresponding normalized eigenvector, i.e. $\|V_j(\boldsymbol{\Sigma})\| = 1$ for all $j = 1, 2, \dots, p$. Define $\lambda_{\min}^*(\boldsymbol{\Sigma}) = \min \{\lambda_j(\boldsymbol{\Sigma}); j = 1, 2, \dots, p\}$, and $\lambda_{\max}^*(\boldsymbol{\Sigma}) = \max \{\lambda_j(\boldsymbol{\Sigma}); j = 1, 2, \dots, p\}$. The density $f(\cdot)$ can be bounded as

$$f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \leq (\lambda_{\min}^*(\boldsymbol{\Sigma}))^{-\frac{p}{2}} g \left(\lambda_{\max}^*(\boldsymbol{\Sigma})^{-1} \|\mathbf{x} - \boldsymbol{\mu}\|^2 \right) \leq g(0) (\lambda_{\min}^*(\boldsymbol{\Sigma}))^{-\frac{p}{2}}. \quad (7)$$

Furthermore, $f(\cdot) \in O(\lambda_{\min}^*(\boldsymbol{\Sigma})^{-\frac{p}{2}})$, meaning that $f(\boldsymbol{\mu}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \rightarrow +\infty$ as $\lambda_{\min}^*(\boldsymbol{\Sigma}) \searrow 0$ unless also $\lambda_{\max}^*(\boldsymbol{\Sigma}) \searrow 0$. The ML problem can therefore not be solved in plain unconstrained form, and requires either constraints on the parameter space or a penalty. Let $\Lambda(\boldsymbol{\theta}) = \{\lambda_j(\boldsymbol{\Sigma}_k); j = 1, \dots, p, k = 1, \dots, k\}$, $\lambda_{\min}(\boldsymbol{\theta}) = \min\{\Lambda(\boldsymbol{\theta})\}$, $\lambda_{\max}(\boldsymbol{\theta}) = \max\{\Lambda(\boldsymbol{\theta})\}$.

A possible constraint to define a proper ML problem is the eigenratio constraint (ERC)

$$\frac{\lambda_{\max}(\boldsymbol{\theta})}{\lambda_{\min}(\boldsymbol{\theta})} \leq \gamma, \quad (8)$$

with a constant $\gamma \geq 1$. $\gamma = 1$ constrains all component scatter matrices to be spherical and equal. Increasing γ continuously allows for more flexible scatter shapes. The ERC has been introduced by [Dennis \(1981\)](#) and [Hathaway \(1985\)](#) for the Gaussian case and brought back to the attention of the literature by [Ingrassia \(2004\)](#). For a recent review see [García-Escudero et al. \(2018\)](#). Although the resulting MLE will not be affine equivariant on the original feature space, affine equivariance can be achieved by spherizing the data. The ERC captures the discrepancy between scatter shapes across components. Therefore, at least in clustering applications, these constraints have a data-analytic interpretation. Other proposals of fully affine equivariant constraints exist in the literature (see [Gallegos and Ritter, 2009; Ritter, 2014](#)); however, there are no algorithms for their exact implementation. Another approach to solving the MLE existence problem is to rely on penalized ML methods. [Ciuperca et al. \(2003\)](#) treated the case of the univariate

Gaussian mixture. A recent comprehensive review is found in [Chen \(2017\)](#).

In the following sections we study the sequence of constrained MLEs

$$\boldsymbol{\theta}_n \in \arg \max_{\boldsymbol{\theta} \in \tilde{\Theta}_K} \ell_n(\boldsymbol{\theta}), \quad (9)$$

where the constrained parameter space is

$$\tilde{\Theta}_K := \left\{ \boldsymbol{\theta} : \pi_k \geq 0 \forall k \geq 1, \sum_{k=1}^K \pi_k = 1; \frac{\lambda_{\max}(\boldsymbol{\theta})}{\lambda_{\min}(\boldsymbol{\theta})} \leq \gamma \right\}. \quad (10)$$

The ERC implies that $\lambda_{\max}(\boldsymbol{\theta}) \in O(\lambda_{\min}(\boldsymbol{\theta}))$ for all $\boldsymbol{\theta} \in \tilde{\Theta}_K$. For a sequence $(\boldsymbol{\theta}_m)_{m \in \mathbb{N}}$ such that $\boldsymbol{\theta}_m \in \tilde{\Theta}_K$ and $\lambda_j(\boldsymbol{\Sigma}_{k,m}) \searrow 0$ as $m \rightarrow +\infty$ for some $j \in \{1, 2, \dots, p\}$ and $k \in \{1, 2, \dots, K\}$ the ERC enforces $\lambda_{\max}(\boldsymbol{\theta}_m) \searrow 0$. The ERC comes with two major technical challenges: (i) the parameter space $\tilde{\Theta}_K$ is not compact; (ii) the resulting ML problem cannot be solved by the standard constrained optimization methods.

3. Finite sample existence

In this section we give precise non-asymptotic conditions involving n, p, K and $g(\cdot)$ that guarantee the existence of $\boldsymbol{\theta}_n$ given the input data set \mathbb{X}_n . The existence of the sequence $(\boldsymbol{\theta}_n)_{n \in \mathbb{N}}$ is the prerequisite for the study of its asymptotic behavior investigated in Section 4 and 5.

Assumption 1. For fixed p, n, K, \mathbb{X}_n :

- (a) $n > K$, and \mathbb{X}_n contains at least $K + 1$ distinct points;
- (b) for all $\beta \in \mathbb{R}_{>0}$ and $\alpha \in \{0, 1, \dots, K\}$, $g(\beta y^{-1})^{n-\alpha} \in o(y^{\frac{p}{2}n})$ as $y \searrow 0$.

Assumption 1-(b) plays the central role in dealing with the sample likelihood function's unboundedness. It guarantees that for $\boldsymbol{\theta} \in \tilde{\Theta}_K$, such that $\lambda_{\min}(\boldsymbol{\theta}) \searrow 0$, the densities in the product $\mathcal{L}_n(\boldsymbol{\theta})$ vanish sufficiently fast at all data points $\mathbf{x}_i \in \mathbb{X}_n$ that do not coincide with mixture components' centers $\boldsymbol{\mu}_k$ for all $k \in \{1, 2, \dots, K\}$. The following Lemma is the key result to obtain the compactification of the parameter space.

Lemma 1. *Let Assumption 1 hold. Consider a sequence $(\boldsymbol{\theta}_m)_{m \in \mathbb{N}} \in \tilde{\Theta}_K$ such that $\lambda_j(\boldsymbol{\Sigma}_{k,m}) \searrow 0$ as $m \rightarrow \infty$ for some $k \in \{1, 2, \dots, K\}$ and $j \in \{1, 2, \dots, p\}$. Then, $\sup_{\tilde{\Theta}_K} \ell_n(\boldsymbol{\theta}_m) \rightarrow -\infty$ as $m \rightarrow \infty$.*

Proof. First, rearrange $\mathcal{L}_n(\cdot)$. Consider a vector of indexes $\mathbf{k} := (k_1, k_2, \dots, k_n)^\top$ where $k_r \in \{1, 2, \dots, K\}$ for $r \in \{1, 2, \dots, n\}$. For a fixed such \mathbf{k} define the products

$$p_{\mathbf{k}}^f(\mathbf{x}_1, \dots, \mathbf{x}_n; \boldsymbol{\theta}) := \prod_{r=1}^n f(\mathbf{x}_r; \boldsymbol{\mu}_{k_r}, \boldsymbol{\Sigma}_{k_r}), \quad \text{and} \quad p_{\mathbf{k}}^\pi(\boldsymbol{\theta}) := \prod_{r=1}^n \pi_{k_r}. \quad (11)$$

There exists K^n of such possible vectors of indexes \mathbf{k} , say $\mathbf{k}_1, \mathbf{k}_2, \dots, \mathbf{k}_{K^n}$. Based on these vectors it is possible to write the sample likelihood function as a mixture of K^n components like eq. (11), that is

$$\mathcal{L}_n(\boldsymbol{\theta}) = \sum_{h=1}^{K^n} p_{\mathbf{k}_h}^\pi(\boldsymbol{\theta}) p_{\mathbf{k}_h}^f(\mathbf{x}_1, \dots, \mathbf{x}_n; \boldsymbol{\theta}). \quad (12)$$

Since $\boldsymbol{\theta}_m \in \tilde{\Theta}_K$, $\lambda_j(\boldsymbol{\Sigma}_{km}) \searrow 0$ implies that $\lambda_{\max}(\boldsymbol{\theta}_m) \searrow 0$ as $m \rightarrow +\infty$. Assume w.l.o.g. (otherwise consider a suitable subsequence) that for sufficiently large m , the sequence $(\boldsymbol{\theta}_m)_{m \in \mathbb{N}}$ is such that, for $j \in \{1, 2, \dots, p\}$, and $k \in \{1, 2, \dots, K\}$, the centrality parameter μ_{kjm} either converges, or it leaves any compact set. If the limits of $(\boldsymbol{\mu}_{km})_{m \in N}$ belong to \mathbb{X}_n , there are at most K of them. Therefore, according to Assumption 1-(a), there exists $i \in \{1, 2, \dots, n\}$ and $\varepsilon > 0$ such that $\mathbf{x}_i \in \mathbb{X}_n \setminus \mathbb{X}'$ and $\|\mathbf{x}_i - \boldsymbol{\mu}_{km}\| > \varepsilon$ for large enough m . Consider a factorization such as the one in eq. (11) for some vector of indexes \mathbf{k} . Define $\mathbb{X}'_{\mathbf{k}} \subseteq \mathbb{X}'$ where

$$\mathbb{X}'_{\mathbf{k}} := \left\{ \mathbf{x}_r \in \mathbb{X}' : \lim_{m \rightarrow \infty} \boldsymbol{\mu}_{k_r m} = \mathbf{x}_r, \text{ for any } k_r \in \{k_1, k_2, \dots, k_n\} \right\}.$$

Let $\#(\mathbb{X}'_{\mathbf{k}}) = q_{\mathbf{k}} \leq K$. Therefore, $p_{\mathbf{k}}^f(\cdot)$ in eq. (11) can be factorized as

$$p_{\mathbf{k}}^f(\mathbf{x}_1, \dots, \mathbf{x}_n; \boldsymbol{\theta}_m) = \prod_{r: \mathbf{x}_r \in \mathbb{X}'_{\mathbf{k}}} f(\mathbf{x}_r; \boldsymbol{\mu}_{k_r m}, \boldsymbol{\Sigma}_{k_r m}) \prod_{r: \mathbf{x}_r \in \mathbb{X} \setminus \mathbb{X}'_{\mathbf{k}}} f(\mathbf{x}_r; \boldsymbol{\mu}_{k_r m}, \boldsymbol{\Sigma}_{k_r m}). \quad (13)$$

As $m \rightarrow +\infty$, $f(\mathbf{x}_r; \boldsymbol{\mu}_{k_r m}, \boldsymbol{\Sigma}_{k_r m}) \in O(\lambda_{\min}(\boldsymbol{\theta}_m)^{-\frac{p}{2}})$ for all r such that $\mathbf{x}_r \in \mathbb{X}'_{\mathbf{k}}$, therefore

$$\sup_{\tilde{\Theta}_K} \prod_{r: \mathbf{x}_r \in \mathbb{X}'_{\mathbf{k}}} f(\mathbf{x}_r; \boldsymbol{\mu}_{k_r m}, \boldsymbol{\Sigma}_{k_r m}) \in O\left(\lambda_{\min}(\boldsymbol{\theta}_m)^{-\frac{p}{2} q_{\mathbf{k}}}\right).$$

On the other hand, for all r such that $\mathbf{x}_r \in \mathbb{X} \setminus \mathbb{X}'_{\mathbf{k}}$ and a positive constant c_{k_r}

$$f(\mathbf{x}_r; \boldsymbol{\mu}_{k_r m}, \boldsymbol{\Sigma}_{k_r m}) \leq O(\lambda_{\min}(\boldsymbol{\theta}_m))^{-\frac{p}{2}} g(\lambda_{\min}(\boldsymbol{\theta}_m)^{-1} c_{k_r}).$$

Assumption 1-(b) and $q_{\mathbf{k}} \leq K$ ensure that $g(\lambda_{\min}(\boldsymbol{\theta}_m)^{-1} c_{k_r})^{n-q_{\mathbf{k}}} \in o(\lambda_{\min}(\boldsymbol{\theta}_m)^{\frac{p}{2} n})$, therefore

$$\sup_{\tilde{\Theta}_K} \prod_{r: \mathbf{x}_r \in \mathbb{X} \setminus \mathbb{X}'_{\mathbf{k}}} f(\mathbf{x}_r; \boldsymbol{\mu}_{k_r m}, \boldsymbol{\Sigma}_{k_r m}) \in O\left(\lambda_{\min}(\boldsymbol{\theta}_m)^{-\frac{p}{2}(n-q_{\mathbf{k}})}\right) o\left(\lambda_{\min}(\boldsymbol{\theta}_m)^{\frac{p}{2} n}\right).$$

The latter implies that

$$\sup_{\tilde{\Theta}_K} p_{\mathbf{k}}^f(\mathbf{x}_1, \dots, \mathbf{x}_n; \boldsymbol{\theta}_m) \in O\left(\lambda_{\min}(\boldsymbol{\theta}_m)^{-\frac{p}{2} q_{\mathbf{k}}}\right) O\left(\lambda_{\min}(\boldsymbol{\theta}_m)^{-\frac{p}{2}(n-q_{\mathbf{k}})}\right) o\left(\lambda_{\min}(\boldsymbol{\theta}_m)^{\frac{p}{2} n}\right)$$

That is

$$\sup_{\tilde{\Theta}_K} p_{\mathbf{k}}^f(\mathbf{x}_1, \dots, \mathbf{x}_n; \boldsymbol{\theta}_m) \in o(1).$$

We conclude that $\sup_{\tilde{\Theta}_K} p_{\mathbf{k}}^f(\mathbf{x}_1, \dots, \mathbf{x}_n; \boldsymbol{\theta}_m) \rightarrow 0$ for large enough m , whatever the vector of indexes \mathbf{k} . The latter implies that all factors of $\mathcal{L}_n(\boldsymbol{\theta}_m)$ in (12) vanish and therefore $\sup_{\tilde{\Theta}_K} \ell_n(\boldsymbol{\theta}_m) \rightarrow -\infty$. \square

Theorem 1 (finite sample existence). *Under Assumption 1, $\boldsymbol{\theta}_n$ exists.*

Proof. First note that $\tilde{\Theta}_K$ is not empty because for any $\gamma \geq 1$ and any choice $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \dots, \boldsymbol{\Sigma}_K$ such that the ERC is not fulfilled, for all $k = 1, 2, \dots, K$ one can always replace $\boldsymbol{\Sigma}_k$ with $\dot{\boldsymbol{\Sigma}}_k = V(\boldsymbol{\Sigma}_k)\hat{\Lambda}(\boldsymbol{\Sigma}_k)V(\boldsymbol{\Sigma}_k)^T$, where $V(\boldsymbol{\Sigma}_k) \in \mathbb{R}^{p \times p}$ is an orthogonal matrix whose columns are eigenvectors of $\boldsymbol{\Sigma}_k$, and $\hat{\Lambda}(\boldsymbol{\Sigma}_k) \in \mathbb{R}^{p \times p}$ is a diagonal matrix where $\hat{\Lambda}(\boldsymbol{\Sigma}_k)[j, j] = \min\{\lambda_j(\boldsymbol{\Sigma}_k), \gamma \lambda_{\min}(\boldsymbol{\theta})\}$. By construction any such choice $\dot{\boldsymbol{\Sigma}}_1, \dot{\boldsymbol{\Sigma}}_2, \dots, \dot{\boldsymbol{\Sigma}}_K$ will always satisfy the eigen-ratio constraint. With the following step we show that there

exists a compact set $T_K \subset \tilde{\Theta}_K$ such that $\sup_{\theta \in T_K} \ell_n(\theta) = \sup_{\theta \in \tilde{\Theta}_K} \ell_n(\theta)$.

Step (a): Consider θ such that $\pi_1 = 1$, $\mu_1 = \mathbf{x}_1$, $\Sigma_j = \mathbf{I}_p$ for all $k \in \{1, 2, \dots, K\}$, arbitrary μ_k and $\pi_k = 0$ for all $k \neq 1$. For such a choice of θ , $\ell_n(\theta) = n^{-1} \sum_{i=1}^n \log f(\mathbf{x}_i; \mathbf{x}_1, \mathbf{I}_p) > -\infty$, thus $\sup_{\theta \in \tilde{\Theta}_K} \ell_n(\theta) > -\infty$.

Step (b): Consider a sequence $(\dot{\theta}_m)_{m \in \mathbb{N}}$. Lemma 1 rules out the possibility that $\dot{\lambda}_{kjm} \searrow 0$ for some index $k \in \{1, 2, \dots, K\}$ and $j \in \{1, 2, \dots, p\}$, because this would imply that $\sup_{\tilde{\Theta}_K} \ell_n(\dot{\theta}_m) \rightarrow -\infty$. Using eq. (7),

$$\ell_n(\theta) = \frac{1}{n} \sum_{i=1}^n \log \left(\sum_{k=1}^K \pi_k f(\mathbf{x}_i; \mu_k, \Sigma_k) \right) \leq K \log (2\pi \lambda_{\min}(\theta))^{-\frac{p}{2}}. \quad (14)$$

Assume that $\dot{\theta}_m \rightarrow \dot{\theta}$ where $\dot{\theta} \in \tilde{\Theta}_K$ and $\dot{\lambda}_{kjm} \rightarrow +\infty$ for some indexes $k \in \{1, 2, \dots, K\}$ and $j \in \{1, 2, \dots, p\}$. Because of the ERC, $\lambda_{\min}(\dot{\theta}_m) \rightarrow +\infty$, and eq. (14) implies that $\sup_{\dot{\theta}_m} \ell_n(\dot{\theta}_m) \rightarrow -\infty$. Therefore, the case that $\dot{\lambda}_{kjm} \rightarrow +\infty$ is also ruled out.

Step (c): now suppose that $\|\dot{\mu}_{km}\| \rightarrow +\infty$ for some $k \in \{1, 2, \dots, K\}$. W.l.o.g take $k = 1$. Choose an alternative sequence $(\ddot{\theta}_m)_{m \in \mathbb{N}}$ that is equal to $(\dot{\theta}_m)_{m \in \mathbb{N}}$ except now $\ddot{\mu}_{1m} = \mathbf{0}$ for all $m \in \mathbb{N}$. Note that $f(\mathbf{x}_i; \ddot{\mu}_{1m}, \Sigma_1) \rightarrow 0$ for all $i \in \{1, 2, \dots, n\}$, which implies that $\psi(\mathbf{x}_i, \dot{\theta}_m) < \psi(\mathbf{x}_i, \ddot{\theta}_m)$ for large enough m for all $i \in \{1, 2, \dots, n\}$. Hence, $\ell_n(\dot{\theta}_m) < \ell_n(\ddot{\theta}_m)$ for large m .

Steps (a)–(c) plus the fact that $\pi_k \in [0, 1]$ for all $k \in \{1, 2, \dots, K\}$ imply that the vector θ maximizing $\ell_n(\cdot)$ must lie in a compact subset $T_n \subset \tilde{\Theta}_K$, and the continuity of $\ell_n(\cdot)$) guarantees existence of θ_n . \square

Remark 2 (Finite sample existence for specific models). The key Assumption 1-(b) holds depending on the density generator function $g(\cdot)$. For the Gaussian distribution, $g(t) = c_p \exp(-t/2)$, with c_p being a constant dependent on p . It is easy to see that the condition is fulfilled for all $n > K$. For the Student-t distribution with ν degrees of freedom, $g(t) = c_{p,\nu} (1+t/\nu)^{-(\nu+p)/2}$ where $c_{p,\nu}$ is a constant that depends on both p and ν . In the latter case Assumption 1-(b) holds if $n > K(1+p/\nu)$, requiring more data points for larger p and smaller ν . Not surprisingly, if $\nu \rightarrow +\infty$, the condition is fulfilled with $n > K$ as for the Gaussian case. Assumption 1-(b) can be easily checked for other ESDs as well.

Remark 3 (Computing). The previous statement is also relevant to ensure that applying computing algorithms searching for θ_n on a given input data set makes sense. In the FMM context, the usual approach to compute the MLE is to run the Expectation-Maximization (EM) algorithm of Dempster et al. (1977) or some of its variants (see McLachlan and Krishnan, 1997). García-Escudero et al. (2015) and Coretto and Hennig (2017) developed EM-type algorithms implementing the ERC for the case of a Gaussian FMM; Coretto and Hennig (2017) also proved convergence results. The general structure of the EM applied to the FMM will apply to finite mixtures of ESDs; however, the generality of $g(\cdot)$ in eq. (1) does not allow to write down the M-step explicitly. Furthermore, the implementation of the ERC for the Gaussian case can be extended to those ESD that have a Gaussian representation. For instance, for mixtures of Student-t distributions, one can take the EM algorithm presented in Chapter 7 of Peel and McLachlan (2000) and add the ERC via a conditional M-step similar to the CM1-step of Algorithm 2 in Coretto and Hennig (2017). In general, an appropriate constrained M-step needs to be developed depending on the specific $g(\cdot)$, but this is outside the scope of this paper.

4. Asymptotic analysis for general P

Now consider a general P as generator of p -dimensional data to be analysed by an MLE derived from an FMM of ESDs. Treating K in the definition of ψ as fixed, define the following log-likelihood-type functionals:

$$L(\boldsymbol{\theta}, P) := \int \log \psi(\mathbf{x}, \boldsymbol{\theta}) dP(\mathbf{x}),$$

and

$$L_K := L_K(P) := \sup_{\boldsymbol{\theta} \in \bar{\Theta}_K} L(\boldsymbol{\theta}), \quad \boldsymbol{\theta}^* := \boldsymbol{\theta}^*(P) := \arg \max_{\boldsymbol{\theta} \in \bar{\Theta}} L(\boldsymbol{\theta}, P).$$

$\boldsymbol{\theta}^*(P)$ is not normally unique (in particular, there is the “label switching” issue, i.e., the numbering of the mixture components is arbitrary), and is taken to be any of the maximising parameter vectors. Without demanding that P is of type eq. (2), the usual interpretation of these quantities is that $L(\boldsymbol{\theta}; P)$ is proportional, up to a term that only depends on P , to the Kullback-Liebler risk (KLR) of approximating the density of P by $\psi(\cdot, \boldsymbol{\theta})$. Therefore, $\boldsymbol{\theta}^*(P)$ provides the best approximation to P in terms of KLR obtainable from a K -components ESD mixture model. The following analysis establishes the existence of $\boldsymbol{\theta}^*(P)$ and the convergence of $(\boldsymbol{\theta}_n)_{n \in \mathbb{N}}$ to $\boldsymbol{\theta}^*(P)$. The notation $E_P[\cdot]$ denotes the expectation with respect to the distribution P .

Assumption 2. For every set $A = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\} \subset \mathbb{R}^p$ with at most K points: $P(A) < 1$.

Assumption 3. $E_P[\log g(\|\mathbf{X}\|^2)] > -\infty$.

Assumption 4. For fixed $\beta \in \mathbb{R}_{>0}$, $g(\beta y^{-1}) \in o(y)$ as $y \searrow 0$.

Assumption 5. $L_{K-1}(P) < L_K(P)$.

Assumption 2 is fulfilled if P does not concentrate its mass on K points. Note that this assumption is fulfilled by the empirical distribution P_n for sufficiently large n when P is absolutely continuous. Assumption 3 is needed to ensure that under P it makes sense to maximize the log-likelihood function. When $f(\cdot)$ is the Gaussian density, Assumption 3 is fulfilled if $E_P \|\mathbf{X}\|^2$ exists. Assumption 4 is the analog of Assumption 1-(b) and deals with the degeneration of the scatter matrices. It implies that, far from the centers $\boldsymbol{\mu}_k$, $f(\cdot)$ vanishes at a speed that is sufficiently fast compared to the speed at which it becomes unbounded in the center when $\lambda_{\min}^*(\boldsymbol{\Sigma}_k) \searrow 0$. Assumption 4 is fulfilled by the Gaussian model. When $f(\cdot)$ is the Student-t distribution with ν degrees of freedom, it holds if $\nu + p > 2$.

Regarding Assumption 5, note that generally $r < s \Rightarrow L_r(P) \leq L_s(P)$, because any parameter $\boldsymbol{\theta}$ with r mixture components can be reproduced with more mixture components setting some component proportions to 0. If $L_{K-1}(P) = L_K(P)$, then maxima of the likelihood exist with a component proportion of 0, and the corresponding location and scatter parameters can take any value. Particularly then the maxima of the likelihood cannot be forced into any compact set.

To prove the consistency Theorem 3, we first establish the existence of the ML functional $\boldsymbol{\theta}^*$ (or equivalently $\boldsymbol{\theta}^*(P) \neq \emptyset$) in Theorem 2. The existence Theorem 2 is obtained via the compactification of the parameter space based on the following Lemmas 2-4.

Lemma 2. Under Assumption 3, for all $K \geq 1$, $L_K(P) > -\infty$.

Proof. Choose $\boldsymbol{\theta} \in \tilde{\Theta}_K$ such that $\boldsymbol{\mu}_1 = \mathbf{0}$, $\boldsymbol{\Sigma}_1 = \mathbf{I}_p$, $\pi_1 = 1$, $\pi_2 = \dots = \pi_K = 0$. Note that $\boldsymbol{\Sigma}_1$ fulfills the ERC. All remaining parameters of $\boldsymbol{\theta}$ are chosen arbitrarily. The statement is proven by using Assumption 3:

$$\sup_{\boldsymbol{\theta} \in \tilde{\Theta}_K} \int \log \psi(\mathbf{x}; \boldsymbol{\theta}) dP(\mathbf{x}) \geq \int \log \pi_1 f(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) dP(\mathbf{x}) \geq \int \log g(\|\mathbf{x}\|^2) dP(\mathbf{x}) > -\infty.$$

□

Lemma 3. Under Assumptions 2 to 4 there exist $\lambda_{\min}^* > 0$, $\lambda_{\max}^* < \infty$, $\epsilon > 0$, so that

- (a) $L(\boldsymbol{\theta}, P) \leq L_K - \epsilon$ for every $\boldsymbol{\theta}$ with $\lambda_{\min}(\boldsymbol{\theta}) < \lambda_{\min}^*$ or $\lambda_{\max}(\boldsymbol{\theta}) > \lambda_{\max}^*$,
- (b) for iid samples X_1, X_2, \dots from P , for sequences $(\dot{\boldsymbol{\theta}}_n)_{n \in \mathbb{N}}$ with $\lambda_{\min}(\dot{\boldsymbol{\theta}}_n) < \lambda_{\min}^*$ or $\lambda_{\max}(\dot{\boldsymbol{\theta}}_n) > \lambda_{\max}^*$ for large enough n : $\ell_n(\dot{\boldsymbol{\theta}}_n) \leq \ell_n(\boldsymbol{\theta}_n) - \epsilon$ P -a.s.

Proof. Consider a sequence $(\boldsymbol{\theta}_m)_{m \in \mathbb{N}}$ where $\boldsymbol{\theta}_m \in \tilde{\Theta}_K$ with $\lambda_{\max}(\boldsymbol{\theta}_m) \rightarrow +\infty$ for all $m \in \mathbb{N}$. The ERC enforces $\lambda_{\min}(\boldsymbol{\theta}_m) \rightarrow +\infty$ and therefore $\sup_{\mathbf{x}} f(\mathbf{x}; \boldsymbol{\mu}_{km}, \boldsymbol{\Sigma}_{km}) \searrow 0$ for all $k = 1, 2, \dots, K$, and $\sup_{\mathbf{x}} \psi(\mathbf{x}; \boldsymbol{\theta}_m) \searrow 0$. The dominated convergence theorem implies that $E_P[\psi(\mathbf{x}; \boldsymbol{\theta}_m)] \searrow 0$, and therefore $L(\boldsymbol{\theta}_m, P) \leq \log(E_P[\psi(\mathbf{x}; \boldsymbol{\theta}_m)]) \rightarrow -\infty$. $L(\boldsymbol{\theta}_m, P) \searrow -\infty$ according to Lemma 2 makes it impossible that $L(\boldsymbol{\theta}_m, P)$ is close to $L_K(P)$ for m large enough. The latter proves the existence of the upper bound $\lambda_{\max}^* < \infty$ as required in part (a).

Now consider a sequence $(\boldsymbol{\theta}_m)_{m \in \mathbb{N}}$ where $\boldsymbol{\theta}_m \in \tilde{\Theta}_K$ with $\lambda_{\min}(\boldsymbol{\theta}_m) \searrow 0$. Define

$$A_{m,\epsilon} = \left\{ \mathbf{x} : \min_{1 \leq k \leq K} \|\mathbf{x} - \boldsymbol{\mu}_{km}\| > \epsilon \right\}.$$

Assumption 2 ensures that for $\delta > 0$ there exists $\epsilon > 0$ so that $P(A_{m,\epsilon}) \geq \delta$ for all $m \in \mathbb{N}$.

$$\begin{aligned} L(\boldsymbol{\theta}_m, P) &\leq \int_{A_{m,\epsilon}} \log \psi(\mathbf{x}; \boldsymbol{\theta}_m) dP(\mathbf{x}) + \int_{A_{m,\epsilon}^c} \log \psi(\mathbf{x}; \boldsymbol{\theta}_m) dP(\mathbf{x}) \\ &\leq \int_{A_{m,\epsilon}} \log \left(K(2\pi)^{-\frac{p}{2}} \lambda_{\min}(\boldsymbol{\theta}_m)^{-\frac{p}{2}} g(\gamma \lambda_{\min}(\boldsymbol{\theta}_m)^{-1} \epsilon^2) \right) dP(\mathbf{x}) + \\ &\quad + \int_{A_{m,\epsilon}^c} \log \left(K(2\pi)^{-\frac{p}{2}} \lambda_{\min}(\boldsymbol{\theta}_m)^{-\frac{p}{2}} g(0) \right) dP(\mathbf{x}) \\ &\leq P(A_{m,\epsilon}) \left(\log K - \frac{p}{2} \log(2\pi) - \frac{p}{2} \log \lambda_{\min}(\boldsymbol{\theta}_m) + \log(g(\gamma \lambda_{\min}(\boldsymbol{\theta}_m)^{-1} \epsilon^2)) \right) + \\ &\quad + P(A_{m,\epsilon}^c) \left(\log K - \frac{p}{2} \log(2\pi) - \frac{p}{2} \log \lambda_{\min}(\boldsymbol{\theta}_m) + \log g(0) \right) \\ &\leq c_1 + c_2 \log \lambda_{\min}(\boldsymbol{\theta}_m)^{-1} + c_3 \log g(c_4 \lambda_{\min}(\boldsymbol{\theta}_m)^{-1}) \end{aligned}$$

for positive constants c_1, c_2, c_3 and c_4 all independent of $\boldsymbol{\theta}_m$. The previous inequality can be rearranged as follows

$$L(\boldsymbol{\theta}_m, P) \leq \log g(c_4 \lambda_{\min}(\boldsymbol{\theta}_m)^{-1}) \left(\frac{c_1}{\log g(c_4 \lambda_{\min}(\boldsymbol{\theta}_m)^{-1})} - c_2 \frac{\log \lambda_{\min}(\boldsymbol{\theta}_m)}{\log g(c_4 \lambda_{\min}(\boldsymbol{\theta}_m)^{-1})} + c_3 \right).$$

By Assumption 4, as $m \rightarrow +\infty$, $g(c_4 \lambda_{\min}(\boldsymbol{\theta}_m)^{-1}) \searrow 0$ faster than $\lambda_{\min}(\boldsymbol{\theta}_m)$ and $L_K(P) \rightarrow -\infty$, which contradicts $L_K(P) > -\infty$ established by Lemma 2. Therefore,

there exists a lower bound $\lambda_{\min}^* > 0$ as stated in part (a) of the statement.

Regarding part (b), let the sequence $(\dot{\boldsymbol{\theta}}_n)_{n \in \mathbb{N}}$ be chosen as the sequence above taking $m = n \rightarrow \infty$. Let P_n be the empirical distribution. The class of all $A_{n,\epsilon}$ is a subset of the class of intersections of the complements of all closed balls, and therefore a Vapnik-Chervonenkis class (see [van der Vaart and Wellner, 1996](#)). Glivenko-Cantelli enforces $P_n(A_{n,\epsilon}) - P(A_{n,\epsilon}) \rightarrow 0$ a.s. Note that $L(\dot{\boldsymbol{\theta}}_n, P_n) = \ell_n(\dot{\boldsymbol{\theta}}_n)$, and $L_K(P_n) = \ell_n(\boldsymbol{\theta}_n)$. For large enough n Lemma 2 holds with P_n replacing P . Therefore part (b) of the statement is proved by replacing P with P_n in the proof of part (a). \square

Lemma 4. *Under Assumptions 2 to 5, there is a compact set $T \subset \mathbb{R}^p$, $\epsilon_1 > 0$ and $\epsilon_2 > 0$ so that*

- (a) $L(\boldsymbol{\theta}, P) \leq L_K - \epsilon_1$ whenever $\boldsymbol{\mu}_k \notin T$ for some $k \in \{1, 2, \dots, K\}$;
- (b) $\sup_{\boldsymbol{\theta} \in \tilde{\Theta}_K: \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K \in T} L(\boldsymbol{\theta}, P) = L_K(P) < +\infty$;
- (c) for iid samples X_1, X_2, \dots from P , for sequences $(\dot{\boldsymbol{\theta}}_n)_{n \in \mathbb{N}}$ with $\dot{\boldsymbol{\theta}}_n \in \tilde{\Theta}_K$ and $\dot{\boldsymbol{\mu}}_{kn} \in T$ for all $k \in \{1, 2, \dots, K\}$ for large enough n : $\sup_{\dot{\boldsymbol{\theta}}_n} \ell_n(\dot{\boldsymbol{\theta}}_n) \leq \ell_n(\boldsymbol{\theta}_n) - \epsilon_2$ P -a.s whenever $\dot{\boldsymbol{\mu}}_{kn} \notin T$ for some $k \in \{1, 2, \dots, K\}$, and $\sup_{\dot{\boldsymbol{\theta}}_n} \ell_n(\dot{\boldsymbol{\theta}}_n) = \ell_n(\boldsymbol{\theta}_n)$ P -a.s.

Proof. Consider a sequence $(\boldsymbol{\theta}_m)_{m \in \mathbb{N}}$ where $\boldsymbol{\theta}_m \in \tilde{\Theta}_K$ for all $m \in \mathbb{N}$. Assume $\|\boldsymbol{\mu}_{km}\| \rightarrow \infty$ for $k \in \{1, 2, \dots, r\}$, and $\boldsymbol{\mu}_{km} \in T$ for all $k > r$. Note that $r = K$ would imply $L(\boldsymbol{\theta}_m, P) \rightarrow -\infty$ for large enough m , therefore, we only consider the case when $r < K$. Take a sequence $(\epsilon_m)_{m \in \mathbb{N}}$ and construct the sets

$$A_m := \left\{ \mathbf{x} : \forall k \in \{1, \dots, r\} f(\mathbf{x}, \boldsymbol{\mu}_{km}, \boldsymbol{\Sigma}_{km}) \leq \epsilon_m \sum_{j=r+1}^K \pi_{jm} f(\mathbf{x}, \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}) \right\},$$

where $\epsilon_m \searrow 0$ slowly enough that $A_1 \supset A_2 \supset \dots$ and $P(A_m) \nearrow 1$. Define another sequence $(\bar{\boldsymbol{\theta}}_m)_{m \in \mathbb{N}}$ such that $\bar{\boldsymbol{\theta}}_m \in \tilde{\Theta}_{K-r}$ with $\bar{\boldsymbol{\mu}}_{km} = \boldsymbol{\mu}_{(k+r)m}$, $\bar{\boldsymbol{\Sigma}}_{km} = \boldsymbol{\Sigma}_{(k+r)m}$, and $\bar{\pi}_{km} = \pi_{(k+r)m} (\sum_{j=r+1}^K \pi_{jm})^{-1}$ for all $k \in \{r+1, \dots, K\}$. By construction $\bar{\boldsymbol{\mu}}_{km} \in T$ for all $k \in \{r+1, \dots, K\}$. Lemma 3 implies that for all $\boldsymbol{\theta} \in \tilde{\Theta}_K$ that are sufficiently close to L_L (that is $L(\boldsymbol{\theta}, P) > L_k - \epsilon$ according to Lemma 3) the mixture component densities are bounded above: $f(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \leq f_{\max} = (2\pi\lambda_{\min}^*)^{-\frac{p}{2}}$ for all k and some suitably defined $\lambda_{\min}^* > 0$. The latter also implies that $L_K < +\infty$. $\psi(\mathbf{x}, \bar{\boldsymbol{\theta}}_m) \geq \psi(\mathbf{x}, \boldsymbol{\theta}_m)$ for all $\mathbf{x} \in A_m$, therefore

$$\begin{aligned} L(\boldsymbol{\theta}_m, P) &= \int_{A_m} \log(\psi(\mathbf{x}, \boldsymbol{\theta}_m)) dP(\mathbf{x}) + \int_{A_m^c} \log(\psi(\mathbf{x}, \boldsymbol{\theta}_m)) dP(\mathbf{x}) \\ &\leq \int_{A_m} \log((1 + \epsilon_m)\psi(\mathbf{x}, \bar{\boldsymbol{\theta}}_m)) dP(\mathbf{x}) + P(A_m^c) \log(f_{\max}). \\ &\leq \int \mathbb{I}_{A_m}(\mathbf{x}) \log(\psi(\mathbf{x}, \bar{\boldsymbol{\theta}}_m)) dP(\mathbf{x}) + a_m \end{aligned}$$

with $a_m \searrow 0$. $\log(f(\mathbf{x}, \bar{\boldsymbol{\mu}}_{km}, \bar{\boldsymbol{\Sigma}}_{km}))$ can be bounded by $\log(f_{\max})$, and by applying the dominated convergence theorem we obtain that $\int \mathbb{I}_{A_m}(\mathbf{x}) \log(\psi(\mathbf{x}, \bar{\boldsymbol{\theta}}_m)) dP(\mathbf{x}) \rightarrow L(\bar{\boldsymbol{\theta}}_m, P)$. Therefore, $L(\boldsymbol{\theta}_m, P) \leq L(\bar{\boldsymbol{\theta}}_m, P) + o(1)$ for large enough m . Because of Assumption 5, $L(\bar{\boldsymbol{\theta}}_m, P) \leq L_{K-r} < L_K$ and therefore $L(\boldsymbol{\theta}_m, P) < L_K$ for sufficiently large m , which proves part (a) of the statement.

Observe that Lemma 3 implies that for all $\boldsymbol{\theta} \in \tilde{\Theta}_K$ for which $L(\boldsymbol{\theta})$ is close to L_K , each eigenvalue in $\boldsymbol{\theta}$ is contained in $[\lambda_{\min}^*, \lambda_{\max}^*]$ with $0 < \lambda_{\min}^* \leq \lambda_{\max}^* < +\infty$, $\psi(\mathbf{x}, \boldsymbol{\theta}) \leq f_{\max}$, and therefore $L_K < +\infty$. Consider $(\boldsymbol{\theta}_m)_{m \in \mathbb{N}}$ so that $L(\boldsymbol{\theta}_m) \rightarrow L_k$ with $\mu_{1m}, \dots, \mu_{Km} \in T$, and $0 < \lambda_{\min}^* \leq \lambda_{kjm} \leq \lambda_{\max}^* < +\infty$ for all $k \in \{1, \dots, K\}$, $j \in \{1, \dots, p\}$. Because of compactness, there exists $\boldsymbol{\theta}_0$ such that $\boldsymbol{\theta}_m \rightarrow \boldsymbol{\theta}_0$ and, using Fatou's Lemma, $L_K = \lim_{m \rightarrow \infty} L(\boldsymbol{\theta}_m, P) \leq E_P[\limsup_m \psi(\mathbf{x}, \boldsymbol{\theta}_m)] = L(\boldsymbol{\theta}_0) \leq L_K < +\infty$. The latter proves the existence of a maximum stated in part (b).

As for the proof of Lemma 3-(b), part (c) of the statement is shown by setting $n = m$, replacing the previous sequences $(\boldsymbol{\theta}_m)_{m \in \mathbb{N}}$ and $(\bar{\boldsymbol{\theta}}_m)_{m \in \mathbb{N}}$ with $(\dot{\boldsymbol{\theta}}_n)_{n \in \mathbb{N}}$ and $(\ddot{\boldsymbol{\theta}}_n)_{n \in \mathbb{N}}$, and replacing P with the empirical distribution P_n . The same steps as before can be applied taking into account the following observations. Glivenko-Cantelli enforces $P_n(A_n) - P(A_n) \rightarrow 0$ a.s. because we can construct a sequence of closed balls $(B_n)_{n \in \mathbb{N}}$ so that $B_n \subseteq A_n$ and $P(B_n) \rightarrow 1$ a.s. The closed balls are a Vapnik-Chervonenkis class, and $P_n(A_n) \geq P_n(B_n)$. Note that for all $\boldsymbol{\theta} \in \tilde{\Theta}_K$ with $L(\boldsymbol{\theta}, P) > L_{K-1}$: $\ell_n(\boldsymbol{\theta}) \rightarrow L(\boldsymbol{\theta}, P)$ a.s. by the strong law of large numbers and therefore for large enough n : $\sup_{\boldsymbol{\theta} \in \tilde{\Theta}_K} \ell_n(\boldsymbol{\theta}) > L_{K-1}$ a.s. Furthermore, if $\dot{\boldsymbol{\theta}}_n$ is chosen optimally in a compact set as in the proof of part (b), $\ell_n(\cdot)$ converges uniformly over $\tilde{\Theta}_{K-r}$ (see Theorem 2 in Jennrich (1969)) and therefore $\limsup_{n \rightarrow \infty} \ell_n(\dot{\boldsymbol{\theta}}_n) \leq L_{K-1} < L_K$. \square

Theorem 2 (Existence of the ML functional). *Under Assumptions 2 to 5 there is a compact subset $T_{\tilde{\Theta}_K} \subset \tilde{\Theta}_K$ so that there exists $\boldsymbol{\theta} \in T_{\tilde{\Theta}_K}$ such that $-\infty < L(\boldsymbol{\theta}) = L_K < +\infty$, and for all $\boldsymbol{\theta} \notin T_{\tilde{\Theta}_K}$, $L(\boldsymbol{\theta})$ is bounded away from L_K .*

Proof. The statement is shown by putting together Lemmas 2–4. \square

Holzmann et al. (2006) found sufficient conditions for the identifiability of certain classes of finite mixtures of ESDs. If $P = P_0$, and $\psi(x; \boldsymbol{\theta}_0)$ is its density whose mixture components also fulfill the sufficient conditions of Holzmann et al. (2006), the previous theorem guarantees that the argmax functional $\boldsymbol{\theta}^*(P)$ exists and is unique up to label switching. This is a consequence of Lemma 1 of Wald (1949). But here we are interested in the more realistic case when ψ is not necessarily a density of P . In this case neither $L(\boldsymbol{\theta})$ nor $\ell_n(\boldsymbol{\theta})$ can be expected to have a unique maximum, not even up to label switching. Based on a technique inspired by Redner (1981), we show that the sequence of constrained ML estimators is asymptotically close to one of the parameters $\boldsymbol{\theta}^*$ giving the best approximation of P in terms of the KLR. This technique provides consistency on the quotient space topology identifying all population log-likelihood maxima. Define the sets

$$S(\dot{\boldsymbol{\theta}}) := \left\{ \boldsymbol{\theta} \in \Theta_K(P) : \int \log \psi(\mathbf{x}; \boldsymbol{\theta}) dP(\mathbf{x}) = \int \log \psi(\mathbf{x}; \dot{\boldsymbol{\theta}}) dP(\mathbf{x}) \right\},$$

$$\mathcal{T}(\dot{\boldsymbol{\theta}}, \varepsilon) := \left\{ \boldsymbol{\theta} \in \Theta_K(P) : \|\boldsymbol{\theta} - \ddot{\boldsymbol{\theta}}\| < \varepsilon \quad \forall \ddot{\boldsymbol{\theta}} \in S(\dot{\boldsymbol{\theta}}) \right\}, \quad \text{for any } \varepsilon > 0.$$

Note that $S(\dot{\boldsymbol{\theta}})$ and $\mathcal{T}(\dot{\boldsymbol{\theta}}, \varepsilon)$ do not depend on the specific likelihood maximiser if $\dot{\boldsymbol{\theta}} = \boldsymbol{\theta}^*$.

Theorem 3 (Consistency). *Under Assumptions 2 to 5, for every $\varepsilon > 0$ and every sequence of maximizers $\boldsymbol{\theta}_n$ of $\ell_n(\cdot)$: $\lim_{n \rightarrow \infty} \Pr[\boldsymbol{\theta}_n \in \mathcal{T}(\boldsymbol{\theta}^*, \varepsilon)] = 1$.*

Proof. Because of Lemmas 3-(b) and 4-(c) there is a compact set $T_{\tilde{\Theta}_K} \subset \tilde{\Theta}_K$ so that all $\boldsymbol{\theta}_n \in T_{\tilde{\Theta}_K}$ for large enough n a.s. Using Lemma 3, $|\log \psi(\mathbf{x}, \boldsymbol{\theta})| \leq C$ for some finite constant C for all $\boldsymbol{\theta} \in T_{\tilde{\Theta}_K}$. Sufficient conditions for Theorem 2 in Jennrich (1969) are

satisfied, and therefore $\sup_{\boldsymbol{\theta} \in T_{\tilde{\Theta}_K}} |\ell_n(\boldsymbol{\theta}) - L(\boldsymbol{\theta})| \rightarrow 0$ P -a.s. Applying the same argument as in the proof of Theorem 5.7 in van der Vaart and Wellner (1996), we obtain $L(\boldsymbol{\theta}_n) \rightarrow L(\boldsymbol{\theta}^*)$ P -a.s. By continuity of $L(\cdot)$ and Theorem 2 we have that for every $\varepsilon > 0$ there exists a $\delta > 0$ such that $L(\boldsymbol{\theta}^*) - L(\boldsymbol{\theta}) > \delta$ for all $\boldsymbol{\theta} \in T_{\tilde{\Theta}_K} \setminus \mathcal{T}(\boldsymbol{\theta}^*, \varepsilon)$. Denote (Ω, \mathcal{A}, P) the probability space where the sample random variables are defined, and consider the following events

$$A_n := \left\{ \omega \in \Omega : \boldsymbol{\theta}_n \in T_{\tilde{\Theta}_K} \setminus \mathcal{T}(\boldsymbol{\theta}^*, \varepsilon) \right\},$$

and

$$B_n := \left\{ \omega \in \Omega : L(\boldsymbol{\theta}^*) - L(\boldsymbol{\theta}_n) > \delta \right\}.$$

Clearly $A_n \subseteq B_n$ for all n . $P(B_n) \rightarrow 0$ for $n \rightarrow \infty$ implies $P(A_n) \rightarrow 0$. The latter proves the result. \square

5. The MLE functional in the mixture case

After having showed the nonparametric consistency of the MLE for its own canonical functional, here we investigate the canonical functional for a P that is a mixture of K well enough separated nonparametric components Q_1, \dots, Q_K .

Such a P can be interpreted as generating K well separated clusters, even though the Q_k , $k = 1, \dots, K$, can be chosen so flexibly that one or more of them themselves could be multimodal or even a mixture of well separated components generating homogeneous data. As the MLE functional of the mixture of ESDs enforces K components to be fitted to P , it seems reasonable, interpreting these as corresponding to K clusters, to expect that they align with Q_1, \dots, Q_K also in the latter case if the separation between Q_1, \dots, Q_K is stronger than the separation between any “subcomponents” of any Q_k .

For P of this type, the clusters generated by the K ESD mixture components of the MLE functional (ESDC) will indeed approximately correspond to the “central regions” of Q_1, \dots, Q_K , and the parameters of the ESDC will approximate the parameters of the MLE canonical functionals of separate ESDs evaluated at each of Q_1, \dots, Q_K alone. For example, if the ESD family is chosen as Gaussian with flexible means and covariance matrices, the corresponding parameters of the ESDC will approximate the mean and covariance matrix functionals of Q_1, \dots, Q_K .

Here are some definitions and assumptions. Let Q_1, \dots, Q_K be distributions on \mathbb{R}^p (generally the same notation refers to distributions and their cumulative distribution functions) parameterized in such a way that 0 is their “center” in some sense; it could be the mode, the mean, the multivariate median or quantile; important is only that Q_k is defined relative to 0. Let $\xi_1, \dots, \xi_K > 0$ mixture proportions with $\sum_{k=1}^K \xi_k = 1$. For $m \in \mathbb{N}$, $k \in \{1, \dots, K\}$ let $\boldsymbol{\rho}_{mk} \in \mathbb{R}^p$ sequences so that

$$\lim_{m \rightarrow \infty} \min_{k_1 \neq k_2 \in \{1, \dots, K\}} \|\boldsymbol{\rho}_{mk_1} - \boldsymbol{\rho}_{mk_2}\| = \infty.$$

Define a sequence of distributions P_m on \mathbb{R}^p by $P_m(\mathbf{x}) := \sum_{k=1}^K \xi_k Q_k(\mathbf{x} - \boldsymbol{\rho}_{mk})$. The mixture P_m is constructed in such a way that its mixture components for increasing m become better and better separated, although they are nonparametric and may have non-vanishing densities, so there may be overlap between them even for arbitrarily large m .

Consider, for $\epsilon > 0$, the “central set” $\{\mathbf{x} : \|\mathbf{x}\| < \epsilon\}$ about 0. ϵ can be chosen large enough

that for arbitrarily small $\eta > 0$:

$$\forall k \in \{1, \dots, K\} : Q_k\{\|\boldsymbol{x}\| < \epsilon\} \geq 1 - \eta, \quad (15)$$

which in particular implies that

$$\exists \delta > 0 : \forall k \in \{1, \dots, K\} : \xi_k Q_k\{\|\boldsymbol{x}\| < \epsilon\} \geq \delta. \quad (16)$$

The following theorem states that in this setup, when evaluating $\boldsymbol{\theta}^*(P_m)$, eventually the different clusters include the full (arbitrarily large) central sets of the different mixture components, and in this sense the clustering corresponds to the mixture structure. We require:

Assumption 6. For any sequence $(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)_{n \in \mathbb{N}}$, Q_k , $k = 1, \dots, K$ for which $\frac{\lambda_{\max}^*(\boldsymbol{\Sigma}_n)}{\lambda_{\min}^*(\boldsymbol{\Sigma}_n)} \leq \gamma$:

$$\lim_{n \rightarrow \infty} \lambda_{\min}^*(\boldsymbol{\Sigma}_n) = 0 \Rightarrow \lim_{n \rightarrow \infty} \int \log f(\boldsymbol{x}; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n) dQ_k(\boldsymbol{x}) = -\infty.$$

Assumption 7. $\exists c_0 > -\infty$ so that for all $k \in \{1, \dots, K\}$: $\int \log g(\|\boldsymbol{x}\|) dQ_k(\boldsymbol{x}) \geq c_0$.

Remark 4. Assumption 6 is e.g. fulfilled for Q_k if Q_k is not concentrated on a single point and $g(x) \in o(x^{-\delta})$, where $\delta > \frac{p}{2\epsilon}$, $\epsilon > 0$ so that there are two disjoint sets A and B : $Q_k(A) \geq \epsilon$, $Q_k(B) \geq \epsilon$, $\inf_{x \in A, y \in B} \frac{\|x-y\|}{2} =: \eta > 0$. This is because in that case, using eq. (6),

$$\begin{aligned} \int \log f(\boldsymbol{x}; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n) dQ_k(\boldsymbol{x}) &\leq \\ \epsilon [\log g(\lambda_{\max}^*(\boldsymbol{\Sigma}_n)^{-1} \eta^2) - \frac{p}{2} \log \lambda_{\min}^*(\boldsymbol{\Sigma}_n)] + (1-\epsilon) [\log g(0) - \frac{p}{2} \log \lambda_{\min}^*(\boldsymbol{\Sigma}_n)] \\ &\in o((\epsilon \delta - \frac{p}{2}) \log \lambda_{\max}^*(\boldsymbol{\Sigma}_n)). \end{aligned}$$

With similar reasoning, Q_k continuous and $g(x) \in o(x^{-\delta})$ with $\delta > \frac{p}{2}$ will fulfill Assumption 6.

Assumption 7 may be violated if Q_k has far heavier tails than f ; e.g., if f is Gaussian, it amounts to Q_k having an existing covariance matrix.

Theorem 4. *With the above definitions, under Assumption 6, for large enough m , the clusters of $\boldsymbol{\theta}^*(P_m)$ can be numbered in such a way that for $k \in \{1, \dots, K\}$:*

$$B_\epsilon(\boldsymbol{\rho}_{mk}) := \{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{\rho}_{mk}\| < \epsilon\} \subseteq C_{mk} = \{\boldsymbol{x} : \text{cl}(\boldsymbol{x}, \boldsymbol{\theta}^*(P_m)) = k\}.$$

Proof. Show the following statements:

S1 For $\tilde{\boldsymbol{\theta}}_m$ defined by $\tilde{\pi}_{mk} = \xi_k$, $\tilde{\boldsymbol{\mu}}_{mk} = \boldsymbol{\rho}_{mk}$, $\tilde{\boldsymbol{\Sigma}}_{mk} = \mathbf{I}_p$, $k = 1, \dots, K$:

$$\exists m^- > -\infty \forall m : L(\tilde{\boldsymbol{\theta}}_m, P_m) \geq m^-. \quad (17)$$

S2 For a sequence $(\boldsymbol{\theta}_m)_{m \in \mathbb{N}}$ let

$$D_m(\boldsymbol{\theta}_m) := \max_{1 \leq k \leq K} \min_{1 \leq j \leq K} \|\boldsymbol{\rho}_{mk} - \boldsymbol{\mu}_{mj}\|.$$

If $\limsup_{m \rightarrow \infty} D_m(\boldsymbol{\theta}_m) = \infty$, then $\liminf_{m \rightarrow \infty} L(\boldsymbol{\theta}_m, P_m) = -\infty$.

S3 The following hold for $\boldsymbol{\theta}_m := \boldsymbol{\theta}^*(P_m)$: There are constants $0 < c_1 < c_2 < \infty$ independent of m so that $c_1 < \lambda_{\min}(\boldsymbol{\theta}_m) < c_2$, and there is a constant $c_3 > 0$ so that for large enough m , $k = 1, \dots, K$: $\pi_{mk} \geq c_3$.

S4 If $\exists m_D < \infty$ so that $\forall m : D_m(\boldsymbol{\theta}_m) \leq m_D$, and S3 holds for $(\boldsymbol{\theta}_m)_{m \in \mathbb{N}}$, then for large enough m the components of $\boldsymbol{\theta}_m$ can be numbered so that for $k = 1, \dots, K$:

$$B_\epsilon(\boldsymbol{\rho}_{mk}) \subseteq C_{mk}. \quad (18)$$

S1 together with S2 imply that $\limsup_{m \rightarrow \infty} D_m(\boldsymbol{\theta}^*(P_m)) < \infty$, so that, together with S3, $(\boldsymbol{\theta}^*(P_m))_{m \in \mathbb{N}}$ fulfills the condition for S4, from which the theorem follows.

Proof of S1:

$$\begin{aligned} L(\tilde{\boldsymbol{\theta}}_m, P_m) &= \int \log \left(\sum_{j=1}^K \tilde{\pi}_{mj} f(\mathbf{x}; \tilde{\boldsymbol{\mu}}_{mj}, \tilde{\boldsymbol{\Sigma}}_{mj}) \right) d \sum_{k=1}^K \xi_k Q_k(\mathbf{x} - \boldsymbol{\rho}_{mk}) \\ &= \sum_{k=1}^K \xi_k \int \log \left(\sum_{j=1}^K \xi_j f(\mathbf{x}; \boldsymbol{\rho}_{mj}, \mathbf{I}_p) \right) d Q_k(\mathbf{x} - \boldsymbol{\rho}_{mk}) \\ &\geq \sum_{k=1}^K \xi_k \int \log (\xi_k f(\mathbf{x}; \boldsymbol{\rho}_{mk}, \mathbf{I}_p)) d Q_k(\mathbf{x} - \boldsymbol{\rho}_{mk}) \\ &= \sum_{k=1}^K \xi_k \int \log (\xi_k g(\mathbf{x}^\top \mathbf{x})) d Q_k(\mathbf{x}) = m^-, \end{aligned}$$

independently of m . $m^- > -\infty$ follows from Assumption 7.

Proof of S2:

It suffices to consider $\lim_{m \rightarrow \infty} D_m(\boldsymbol{\theta}_m) = \infty$ because in case this holds for the \limsup , there is a subsequence diverging to ∞ . W.l.o.g., number mixture components so that $D_m(\boldsymbol{\theta}_m) = \|\boldsymbol{\rho}_{mk^*} - \boldsymbol{\mu}_{mk^*}\|$ for a fixed $k^* \in \{1, \dots, K\}$.

The following is required for showing that $\lim_{m \rightarrow \infty} L(\boldsymbol{\theta}_m, P_m) = -\infty$: Because of the properties of f and g , for $k = 1, \dots, K$ and any sequence $(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)_{n \in \mathbb{N}}$ for which $\frac{\lambda_{\max}^*(\boldsymbol{\Sigma}_n)}{\lambda_{\min}^*(\boldsymbol{\Sigma}_n)} \leq \gamma$:

$$\lim_{n \rightarrow \infty} \lambda_{\min}^*(\boldsymbol{\Sigma}_n) = \infty \Rightarrow \int \log f(\mathbf{x}; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n) d Q_k(\mathbf{x}) = -\infty. \quad (19)$$

Furthermore,

$$\lim_{n \rightarrow \infty} |\boldsymbol{\mu}_n| = \infty \Rightarrow \int \log f(\mathbf{x}; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n) d Q_k(\mathbf{x}) = -\infty, \quad (20)$$

regardless of whether $\lambda_{\min}^*(\boldsymbol{\Sigma}_n)$ is bounded (in which case $\log f(\mathbf{x}; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n) \rightarrow -\infty$ for $\mathbf{x} \in S$ with $Q_k(S)$ arbitrarily close to 1), diverges to ∞ (in which case eq. (19) obtains), or converges to zero (Assumption 6).

Observe

$$L(\boldsymbol{\theta}_m, P_m) = \xi_{k^*} \int \log \left(\sum_{j=1}^K \pi_{mj} f(\mathbf{x}; \boldsymbol{\mu}_{mj}, \boldsymbol{\Sigma}_{mj}) \right) d Q_{k^*}(\mathbf{x} - \boldsymbol{\rho}_{mk^*}) + L_m^*, \quad (21)$$

where

$$L_m^* := \sum_{k \neq k^*} \xi_k \int \log \left(\sum_{j=1}^K \pi_{mj} f(\mathbf{x}; \boldsymbol{\mu}_{mj}, \boldsymbol{\Sigma}_{mj}) \right) dQ_k(\mathbf{x} - \boldsymbol{\rho}_{mk}).$$

Because of eq. (20), the first term of eq. (21) diverges to $-\infty$, and so does $L(\boldsymbol{\theta}_m, P_m)$ if L_m^* is bounded from above. Assumption 6 implies that $\lambda_{\min}(\boldsymbol{\theta}_m) \geq c > 0$, therefore $f(\mathbf{x}; \boldsymbol{\mu}_{mk}, \boldsymbol{\Sigma}_{mk}) \leq c^{-p/2} g(0) < \infty$, and $L_m^* \leq c^{-p/2} g(0)$. This proves S2.

Proof of S3:

$c_1 < \lambda_{\min}(\boldsymbol{\theta}^*(P_m)) < c_2$ holds by S1, Assumption 6, and eq. (19).

Because of S2, $\exists m_D < \infty$ so that $\forall m : D_m(\boldsymbol{\theta}^*(P_m)) \leq m_D$. Number the components of $\boldsymbol{\theta}_m$ so that for $k = 1, \dots, K$:

$$\|\boldsymbol{\rho}_{mk} - \boldsymbol{\mu}_{mk}\| = \min_{1 \leq j \leq K} \|\boldsymbol{\rho}_{mk} - \boldsymbol{\mu}_{mj}\|. \quad (22)$$

This is possible because of $D_m(\boldsymbol{\theta}_m) \leq m_D$.

Assume w.l.o.g. that $\pi_{m1} \rightarrow 0$. Then, eq. (21) holds with $k^* = 1$. L_m^* is once more bounded from above, and

$$\lim_{m \rightarrow \infty} \int \log \left(\sum_{j=1}^K \pi_{mj} f(\mathbf{x}; \boldsymbol{\mu}_{mj}, \boldsymbol{\Sigma}_{mj}) \right) dQ_1(\mathbf{x} - \boldsymbol{\rho}_{m1}) = -\infty, \quad (23)$$

because $\pi_{m1} \rightarrow 0$, $f(\mathbf{x}; \boldsymbol{\mu}_{m1}, \boldsymbol{\Sigma}_{m1}) \leq c_1^{-p/2} g(0) < \infty$ as in the proof of S2, and for $j \neq 1 : f(\mathbf{x}; \boldsymbol{\mu}_{mj}, \boldsymbol{\Sigma}_{mj}) \rightarrow 0$ for $\mathbf{x} \in S$ with $\int_S dQ_1(\mathbf{x} - \boldsymbol{\rho}_{m1})$ arbitrarily close to 1. But for $\boldsymbol{\theta}_m = \boldsymbol{\theta}^*(P_m)$, eq. (23) is in contradiction to S1.

Proof of S4:

For large enough m number the components of $\boldsymbol{\theta}_m$ according to eq. (22). For $\tilde{\mathbf{x}} \in B_\epsilon(\mathbf{0})$ so that $\mathbf{x} = \tilde{\mathbf{x}} + \boldsymbol{\rho}_{mk} \in B_\epsilon(\boldsymbol{\rho}_{mk})$ consider

$$\tau_k(\mathbf{x}; \boldsymbol{\theta}_m) = \frac{\pi_{mk} f(\mathbf{x}; \boldsymbol{\mu}_{mk}, \boldsymbol{\Sigma}_{mk})}{\sum_{j=1}^K \pi_{mj} f(\mathbf{x}; \boldsymbol{\mu}_{mj}, \boldsymbol{\Sigma}_{mj})}.$$

Because of S3, $\pi_{mk} \geq c_3$, and $f(\mathbf{x}; \boldsymbol{\mu}_{mk}, \boldsymbol{\Sigma}_{mk}) \geq c_4 > 0$, whereas for $j \neq k : f(\mathbf{x}; \boldsymbol{\mu}_{mj}, \boldsymbol{\Sigma}_{mj}) \rightarrow 0$ uniformly for $\tilde{\mathbf{x}} \in B_\epsilon(\mathbf{0})$, so that $\tau_k(\mathbf{x}; \boldsymbol{\theta}_m) \rightarrow 1$, and for large enough $m : \forall \mathbf{x} : \text{cl}(\mathbf{x}, \boldsymbol{\theta}_m) = k$. \square

Remark 5. It is not essential that $B_\epsilon(\boldsymbol{\rho}_{mk})$ is defined based on the Euclidean distance; other L_p -distances will work as well as Mahalanobis distances with any fixed covariance matrix.

Remark 6. For m large, the different components Q_k , $k = 1, \dots, K$, are shifted by $\boldsymbol{\rho}_{mk}$ and their central sets $B_\epsilon(\boldsymbol{\rho}_{mk})$ are therefore very far away from each other. It may therefore not seem surprising that these ultimately belong to different clusters. But the statement is not trivial. The Q_k can have densities that are nowhere zero and even have heavy tails (Assumption 7 will then require f to have heavy tails, too), so that there will always be overlap between the Q_k . There are clustering methods with fixed K that for $n \rightarrow \infty$ will not match the K different clusters to $B_\epsilon(\boldsymbol{\rho}_{m1}), \dots, B_\epsilon(\boldsymbol{\rho}_{mK})$:

- If the Q_k (or even at least one of them) are chosen so that for $n \rightarrow \infty$, the

largest distance $D_{max,n}$ of an observation to its nearest neighbour goes to ∞ in probability (which holds for distributions with heavy enough tails, see Theorem 5.3 in [Jammalamadaka and Janson \(2015\)](#)), then for any given but arbitrarily large m, n can be chosen large enough so that $D_{max,n}$ is arbitrarily much larger than $\max_{1 \leq j, k \leq K} \|\rho_{mj} - \rho_{mk}\|$ with arbitrarily large probability. Then, the Single Linkage dendrogram based on the Euclidean distance will merge different central sets earlier than connecting the observation that is farthest from its nearest neighbour with anything else.

- A trimmed clustering method trimming a proportion of α observations can trim a complete central set if $\alpha \geq \xi_k$ for some k . Similarly, RIMLE ([Coretto and Hennig \(2017\)](#)) may classify a complete central set as noise.

Arguably there are situations in which the behaviour of trimmed clustering and RIMLE as explained above may be seen as desirable, namely if one of the Q_k looks like modelling more than one cluster; e.g., it can be bimodal with two clearly separated modes. It may then be seen as appropriate if this attracts more than one of the K clusters, whereas another Q_k with either low probability or low (smoothed) density may be classified as generating outliers.

The following theorem states that for the setup of Theorem 4, the estimators of μ_{mk} and Σ_{mk} converge to the estimators for the individual mixture components Q_{mk} , so that the increasing separation of mixture components for $m \rightarrow \infty$ implies that ultimately the characteristics of Q_{mk} defined in terms of the ESD densities f can be estimated without influence of the other mixture components. For example, if f defines a Gaussian location-scale family, μ_{mk}^* will converge toward the mean of Q_{mk} , and Σ_{mk}^* will converge to the covariance matrix of Q_{mk} , see Corollary 1, which shows that the required Assumption 8 below holds at least in this case.

For $k = 1, \dots, K$, let

$$\tilde{\kappa}_k := (\tilde{\mu}_k, \tilde{\Sigma}_k) = \arg \max_{\kappa} \tilde{L}(\kappa, Q_k), \quad \tilde{L}(\kappa, Q) := \int \log f(\mathbf{x}; \kappa) dQ_k(\mathbf{x}).$$

The corresponding functionals for $Q_{mk} = Q_k(\bullet - \rho_{mk})$ are

$$\tilde{\mu}_{mk} := \tilde{\mu}_k + \rho_{mk}, \quad \tilde{\Sigma}_{mk} := \tilde{\Sigma}_k. \quad (24)$$

Assumption 8. For given Q_k ,

$$\forall \varepsilon > 0 \exists \beta > 0 : \|\kappa - \tilde{\kappa}_k\| > \varepsilon \Rightarrow L(\tilde{\kappa}_k, Q_k) - L(\kappa, Q_k) > \beta.$$

Assumption 9. For $\tilde{\theta} = (\xi_1, \dots, \xi_K, \tilde{\kappa}_1, \dots, \tilde{\kappa}_K)$: $\frac{\lambda_{\min}(\tilde{\theta})}{\lambda_{\max}(\tilde{\theta})} \leq \gamma$.

Theorem 5. *With the above definitions, under Assumptions 6 and 7, for large enough m , the clusters of $\theta^*(P_m)$ can be numbered in such a way that for $k \in \{1, \dots, K\}$, with $\theta_m^* := \theta^*(P_m)$.*

$$\lim_{m \rightarrow \infty} \|\pi_{mk}^* - \xi_k\| = 0, \quad (25)$$

and if Q_1, \dots, Q_K fulfill Assumption 9, then for those Q_k that fulfill Assumption 8:

$$\lim_{m \rightarrow \infty} \|\kappa_{mk}^* - \tilde{\kappa}_{mk}\| = 0. \quad (26)$$

Proof. Assume throughout that mixture components are numbered according to eq. (22). Show the following statements:

S1 For $\boldsymbol{\theta}_m^* = \boldsymbol{\theta}^*(P_m)$, using the notation of eq. (3), and $\boldsymbol{\theta}_{mk}^*$ denoting the parameters in $\boldsymbol{\theta}_m^*$ belonging to mixture component k , $k = 1, \dots, K$,

$$\boldsymbol{\theta}_{mk}^* = \arg \max_{\boldsymbol{\theta}} \int \tau_k(\mathbf{x}; \boldsymbol{\theta}_{mk}^*)(\log \pi_k + \log f(\mathbf{x}; \boldsymbol{\kappa}_k)) dP_m(\mathbf{x}).$$

S2 eq. (25) follows from S1.

S3 From S1,

$$\boldsymbol{\kappa}_{mk}^* = \arg \max_{\boldsymbol{\kappa}} q_{mk}(\boldsymbol{\kappa}),$$

where $q_{mk}(\boldsymbol{\kappa})$ is a function so that $q_{mk}(\boldsymbol{\kappa}) - \int \log f(\mathbf{x}; \boldsymbol{\kappa}) dQ_k(\mathbf{x} - \boldsymbol{\rho}_{mk})$ converges uniformly to 0.

S4 eq. (26) follows from S3 and Assumption 8.

Proof of S1:

The proof is based on the two-step form of a mixture model involving for each observed X an unobserved random variable ζ indicating the mixture component that has generated X , see Section 2.1. S1 is the population version of the fixed point equation on which the EM-algorithm is based, see Section 3 and in particular Corollary 2 of Dempster et al. (1977).

Regarding the mixture eq. (2), on which the ML-estimation is based, let $\tilde{\psi}(\bullet; \boldsymbol{\theta})$ be the joint density of $Y = (X, \zeta)$, and $\bar{\psi}(\bullet | \mathbf{x}; \boldsymbol{\theta})$ be the conditional density of Y given $X = \mathbf{x}$ so that for all \mathbf{y}, \mathbf{x} :

$$\bar{\psi}(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta}) = \frac{\tilde{\psi}(\mathbf{y}; \boldsymbol{\theta})}{\psi(\mathbf{x}; \boldsymbol{\theta})}. \quad (27)$$

Define

$$\begin{aligned} G(\boldsymbol{\theta}' | \boldsymbol{\theta}) &:= \int \int \log(\tilde{\psi}(\mathbf{y}; \boldsymbol{\theta}')) \bar{\psi}(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta}) d\mathbf{y} dP(\mathbf{x}) \\ H(\boldsymbol{\theta}' | \boldsymbol{\theta}) &:= \int \int \log(\bar{\psi}(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta}')) \bar{\psi}(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta}) d\mathbf{y} dP(\mathbf{x}). \end{aligned}$$

Then, following Dempster et al. (1977), eq. (27) still holds averaged over the $y | \mathbf{x}$ assumed distributed according to $\bar{\psi}(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta})$, and then averaging over $P_m(\mathbf{x})$ yields

$$L(\boldsymbol{\theta}', P) = G(\boldsymbol{\theta}' | \boldsymbol{\theta}) - H(\boldsymbol{\theta}' | \boldsymbol{\theta}).$$

For given \mathbf{x} , formula (1e6.6) in Rao (1965) implies

$$\begin{aligned} \int (\log \bar{\psi}(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta}) - \log \bar{\psi}(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta}')) \bar{\psi}(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta}) dy &\geq 0 \\ \Rightarrow H(\boldsymbol{\theta} | \boldsymbol{\theta}) &\geq H(\boldsymbol{\theta}' | \boldsymbol{\theta}). \end{aligned}$$

Let $M(\boldsymbol{\theta}_m^*) := \arg \max_{\boldsymbol{\theta}} G(\boldsymbol{\theta} | \boldsymbol{\theta}_m^*)$. Then

$$L(M(\boldsymbol{\theta}_m^*)) = Q(M(\boldsymbol{\theta}_m^*) | \boldsymbol{\theta}_m^*) - H(M(\boldsymbol{\theta}_m^*) | \boldsymbol{\theta}_m^*) \geq G(\boldsymbol{\theta}_m^* | \boldsymbol{\theta}_m^*) - H(\boldsymbol{\theta}_m^* | \boldsymbol{\theta}_m^*) = L(\boldsymbol{\theta}_m^*),$$

implying $\theta_m^* = \arg \max_{\theta} G(\theta | \theta_m^*)$ (otherwise the “ \geq ” above would be “ $>$ ” and $M(\theta_m^*)$ would improve L). Note that

$$G(\theta | \theta_m^*) = \int \sum_{k=1}^K \tau_k(\mathbf{x}; \theta_{mk}^*) (\log \pi_k + \log f(\mathbf{x}; \boldsymbol{\kappa}_k)) dP_m(\mathbf{x}),$$

which can be maximised for each k separately, leading to S1.

Proof of S2:

$\int \tau_k(\mathbf{x}; \theta_{mk}^*) (\log \pi_k + \log f(\mathbf{x}; \boldsymbol{\kappa}_k)) dP_m(\mathbf{x})$ can be maximised separately over π_1, \dots, π_K and $\boldsymbol{\kappa}_1, \dots, \boldsymbol{\kappa}_K$. Maximising $\int \tau_k(\mathbf{x}; \theta_{mk}^*) \log \pi_k dP_m(\mathbf{x})$ yields

$$\pi_{mk}^* = \int \tau_k(\mathbf{x}; \theta_{mk}^*) dP_m(\mathbf{x}) = \int_{B_\varepsilon(\boldsymbol{\rho}_{mk})} \tau_k(\mathbf{x}; \theta_{mk}^*) dP_m(\mathbf{x}) + \int_{B_\varepsilon(\boldsymbol{\rho}_{mk})^c} \tau_k(\mathbf{x}; \theta_{mk}^*) dP_m(\mathbf{x}), \quad (28)$$

$B_\varepsilon(\boldsymbol{\rho}_{mk})$ chosen as in the proof of S4 of Theorem 4. From there, $\tau_k(\mathbf{x}; \theta_{mk}^*) \rightarrow 1$ for $\mathbf{x} \in B_\varepsilon(\mathbf{0})$, i.e., $\mathbf{x} \in B_\varepsilon(\boldsymbol{\rho}_{mk})$, therefore

$$\lim_{m \rightarrow \infty} \left| \int_{B_\varepsilon(\boldsymbol{\rho}_{mk})} \tau_k(\mathbf{x}; \theta_{mk}^*) dP_m(\mathbf{x}) - \int_{B_\varepsilon(\boldsymbol{\rho}_{mk})} d \sum_{k=1}^K \xi_k Q_{mk}(\mathbf{x}) \right| = 0, \quad (29)$$

and furthermore, for $j \neq k$, $m \rightarrow \infty$, because of eq. (15),

$$Q_{mj}(B_\varepsilon(\boldsymbol{\rho}_{mk})) \rightarrow 0 \Rightarrow \int_{B_\varepsilon(\boldsymbol{\rho}_{mk})} d \xi_j Q_{mj}(\mathbf{x}) \rightarrow 0. \quad (30)$$

ε can be chosen so large that $Q_{mk}(B_\varepsilon(\boldsymbol{\rho}_{mk}))$ is arbitrarily close to 1, and due to eq. (30), $\int_{B_\varepsilon(\boldsymbol{\rho}_{mk})} d \sum_{k=1}^K \xi_k Q_{mk}(\mathbf{x})$ is arbitrarily close to ξ_k . Furthermore, assuming m large enough that $B_\varepsilon(\boldsymbol{\rho}_{mj})$, $j = 1, \dots, K$, do not intersect, with $\tilde{B} = \left(\bigcup_{j=1}^K B_\varepsilon(\boldsymbol{\rho}_{mj}) \right)^c$,

$$\int_{B_\varepsilon(\boldsymbol{\rho}_{mk})^c} \tau_k(\mathbf{x}; \theta_{mk}^*) dP_m(\mathbf{x}) = \sum_{j \neq k} \int_{B_\varepsilon(\boldsymbol{\rho}_{mj})} \tau_k(\mathbf{x}; \theta_{mk}^*) dP_m(\mathbf{x}) + \int_{\tilde{B}} \tau_k(\mathbf{x}; \theta_{mk}^*) dP_m(\mathbf{x}).$$

For the same reason as eq. (29) and eq. (30) (with roles of j and k inverted),

$$\sum_{j \neq k} \int_{B_\varepsilon(\boldsymbol{\rho}_{mj})} \tau_k(\mathbf{x}; \theta_{mk}^*) dP_m(\mathbf{x}) \rightarrow 0.$$

Because of eq. (15), $P_m(\tilde{B})$ is arbitrarily small for ε large enough, as is $\int_{\tilde{B}} \tau_k(\mathbf{x}; \theta_{mk}^*) dP_m(\mathbf{x})$. Putting everything together, eq. (28) implies eq. (25).

Proof of S3:

From S1, $\boldsymbol{\kappa}_{mk}^* = \arg \max_{\boldsymbol{\kappa}} \tilde{q}_{mk}(\boldsymbol{\kappa})$ with

$$\tilde{q}_{mk}(\boldsymbol{\kappa}) = \int \tau_k(\mathbf{x}; \theta_{mk}^*) \log f(\mathbf{x}; \boldsymbol{\kappa}_k) dP_m(\mathbf{x}) = \int \tau_k(\mathbf{x}; \theta_{mk}^*) \log f(\mathbf{x}; \boldsymbol{\kappa}_k) d \sum_{k=1}^K \xi_k Q_{mk}(\mathbf{x}).$$

As in eq. (28) and the proof of S2,

$$\tilde{q}_{mk}(\boldsymbol{\kappa}) = \int_{B_\varepsilon(\boldsymbol{\rho}_{mk})} \tau_k(\mathbf{x}; \theta_{mk}^*) \log f(\mathbf{x}; \boldsymbol{\kappa}_k) dP_m(\mathbf{x}) + \int_{B_\varepsilon(\boldsymbol{\rho}_{mk})^c} \tau_k(\mathbf{x}; \theta_{mk}^*) \log f(\mathbf{x}; \boldsymbol{\kappa}_k) dP_m(\mathbf{x}),$$

and

$$\begin{aligned} \int_{B_\varepsilon(\boldsymbol{\rho}_{mk})^c} \tau_k(\mathbf{x}; \boldsymbol{\theta}_{mk}^*) \log f(\mathbf{x}; \boldsymbol{\kappa}_k) dP_m(\mathbf{x}) &= \sum_{j \neq k} \int_{B_\varepsilon(\boldsymbol{\rho}_{mj})} \tau_k(\mathbf{x}; \boldsymbol{\theta}_{mk}^*) \log f(\mathbf{x}; \boldsymbol{\kappa}_k) dP_m(\mathbf{x}) \\ &\quad + \int_{\tilde{B}} \tau_k(\mathbf{x}; \boldsymbol{\theta}_{mk}^*) \log f(\mathbf{x}; \boldsymbol{\kappa}_k) dP_m(\mathbf{x}). \end{aligned}$$

Because of Assumption 6, regarding $\arg \max_{\boldsymbol{\kappa}} q_{mk}(\boldsymbol{\kappa})$, it suffices to consider $\boldsymbol{\kappa}$ with $\lambda_{\min}^*(\boldsymbol{\Sigma}) > c_1$ for suitable $c_1 > 0$, $c_2 = \log(c_1^{-p/2} g(0))$, so that $\log f(\mathbf{x}; \boldsymbol{\kappa}_k) < c_2$. On $B_\varepsilon(\boldsymbol{\rho}_{mk})$, according to the proof of Theorem 4, $\tau_k(\mathbf{x}; \boldsymbol{\theta}_{mk}^*) \rightarrow 1$ and, for $j \neq k$, $\tau_j(\mathbf{x}; \boldsymbol{\theta}_{mj}^*) \rightarrow 0$, and furthermore, for $m \rightarrow \infty$,

$$\int_{B_\varepsilon(\boldsymbol{\rho}_{mk})} \tau_k(\mathbf{x}; \boldsymbol{\theta}_{mk}^*) \log f(\mathbf{x}; \boldsymbol{\kappa}_k) dQ_{mj}(\mathbf{x}) < c_2 \int_{B_\varepsilon(\boldsymbol{\rho}_{mk})} \tau_k(\mathbf{x}; \boldsymbol{\theta}_{mk}^*) dQ_{mj}(\mathbf{x}) \rightarrow 0.$$

Therefore,

$$\lim_{m \rightarrow \infty} \left| \int_{B_\varepsilon(\boldsymbol{\rho}_{mk})} \tau_k(\mathbf{x}; \boldsymbol{\theta}_{mk}^*) \log f(\mathbf{x}; \boldsymbol{\kappa}_k) dP_m(\mathbf{x}) - \int_{B_\varepsilon(\boldsymbol{\rho}_{mk})} \log f(\mathbf{x}; \boldsymbol{\kappa}_k) d\xi_k Q_{mk}(\mathbf{x}) \right| = 0. \quad (31)$$

For $j \neq k$, on $B_\varepsilon(\boldsymbol{\rho}_{mj})$: $\tau_k(\mathbf{x}; \boldsymbol{\theta}_{mk}^*) \rightarrow 0$, therefore

$$\int_{B_\varepsilon(\boldsymbol{\rho}_{mj})} \tau_k(\mathbf{x}; \boldsymbol{\theta}_{mk}^*) \log f(\mathbf{x}; \boldsymbol{\kappa}_k) dP_m(\mathbf{x}) < c_2 \int_{B_\varepsilon(\boldsymbol{\rho}_{mj})} \tau_k(\mathbf{x}; \boldsymbol{\theta}_{mk}^*) dP_m(\mathbf{x}) \rightarrow 0, \quad (32)$$

and

$$\int_{\tilde{B}} \tau_k(\mathbf{x}; \boldsymbol{\theta}_{mk}^*) \log f(\mathbf{x}; \boldsymbol{\kappa}_k) dP_m(\mathbf{x}) < c_2 P_m(\tilde{B}). \quad (33)$$

Consider $q_{mk}(\boldsymbol{\kappa}) = \frac{\tilde{q}_{mk}(\boldsymbol{\kappa})}{\xi_k}$ in order to drop ξ_k from eq. (31). Once more ε can be chosen large enough that $P_m(\tilde{B})$ becomes arbitrarily small, and $Q_{mk}(B_\varepsilon(\boldsymbol{\rho}_{mk}))$ becomes arbitrarily large, so that eq. (31), eq. (32), and eq. (33) together imply that

$$q_{mk}(\boldsymbol{\kappa}) - \int \log f(\mathbf{x}; \boldsymbol{\kappa}) dQ_{mk}(\mathbf{x}) \rightarrow 0,$$

and the convergence is uniform over $\boldsymbol{\kappa}$ as neither $\tau_j(\mathbf{x}; \boldsymbol{\theta}_{mj}^*)$ for any j nor ε nor c_2 depend on $\boldsymbol{\kappa}$.

Proof of S4:

Given arbitrarily small ε and the corresponding β from Assumption 8, according to S3, for m large enough, uniformly

$$|q_{mk}(\boldsymbol{\kappa}) - \tilde{L}(\boldsymbol{\kappa}, Q_k)| < \frac{\beta}{2}.$$

This means that $\boldsymbol{\kappa}$ with $\|\boldsymbol{\kappa} - \tilde{\boldsymbol{\kappa}}_{mk}\| > \epsilon$ cannot maximise $q_{mk}(\boldsymbol{\kappa})$, and therefore $\boldsymbol{\kappa}_{mk}^* = \arg \max_{\boldsymbol{\kappa}} q_{mk}(\boldsymbol{\kappa})$ will become arbitrarily close to $\tilde{\boldsymbol{\kappa}}_{mk}$, because Assumption 9 enforces $\frac{\lambda_{\min}(\boldsymbol{\theta})}{\lambda_{\max}(\boldsymbol{\theta})} < \gamma$, which is also required to hold for $\boldsymbol{\theta}^*(P^m)$. This proves eq. (26). \square

Corollary 1. *In the situation of Theorem 5, requiring Assumptions 6 and 7, if g is chosen so that $f(\bullet; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the density of a p -variate Gaussian distribution with mean $\boldsymbol{\mu}$*

and covariance matrix Σ , then

$$\lim_{m \rightarrow \infty} \left\| \boldsymbol{\mu}_{mk}^* - \int \mathbf{x} dQ_k(\mathbf{x}) - \boldsymbol{\rho}_{mk} \right\| = 0. \quad (34)$$

If additionally Assumption 9 holds, then

$$\lim_{m \rightarrow \infty} \left\| \boldsymbol{\Sigma}_{mk}^* - \int (\mathbf{x} - \tilde{\boldsymbol{\mu}}_k)(\mathbf{x} - \tilde{\boldsymbol{\mu}}_k)^T dQ_k(\mathbf{x}) \right\| = 0.$$

In this case, Assumption 6 will hold if Q_k is not concentrated on a single point, see Remark 4. Assumption 7 amounts to $\int \|\mathbf{x}^\top \mathbf{x}\| dQ_k(\mathbf{x}) < \infty$.

Proof. Due to Assumption 6, consider κ with $\lambda_{\min}^*(\boldsymbol{\Sigma}_k) > c_1$ for $k = 1, \dots, K$, $c_1 > 0$. For the multivariate Gaussian (see, e.g., Anderson and Olkin (1985))

$$\begin{aligned} \log f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= c - \frac{1}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}), \\ \frac{\partial}{\partial \boldsymbol{\mu}} \log f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}), \\ \frac{\partial}{\partial \boldsymbol{\Sigma}^{-1}} \log f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \frac{1}{2} \boldsymbol{\Sigma} - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top. \end{aligned}$$

For $\boldsymbol{\mu} \in B_\varepsilon(\boldsymbol{\mu}_0)$ for any $\boldsymbol{\mu}_0$,

$$\left\| \frac{\partial}{\partial \boldsymbol{\mu}} \log f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \right\| \leq c_1^{-p} (\|\boldsymbol{\mu}_0 - \mathbf{x}\| + \varepsilon).$$

For $\boldsymbol{\Sigma}^{-1} \in B_\varepsilon(\boldsymbol{\Sigma}_0^{-1})$ for any $\boldsymbol{\Sigma}_0$,

$$\left\| \frac{\partial}{\partial \boldsymbol{\Sigma}^{-1}} \log f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \right\| \leq \frac{1}{2} \left(\boldsymbol{\Sigma}_0 + \varepsilon - (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top \right).$$

Both these bounding functions are integrable w.r.t. P_m because of Assumption 7. This means that $\log f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ can be differentiated under the integral sign. Standard algebra yields

$$\tilde{\boldsymbol{\mu}}_{mk} = \int \mathbf{x} dQ_{mk}(\mathbf{x}), \quad \tilde{\boldsymbol{\Sigma}}_{mk} = \int (\mathbf{x} - \tilde{\boldsymbol{\mu}}_{mk})(\mathbf{x} - \tilde{\boldsymbol{\mu}}_{mk})^\top dQ_{mk}(\mathbf{x}).$$

Assumption 8 follows because $\log f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is strictly concave in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}^{-1}$ (Anderson and Olkin (1985)), and the Corollary follows from Theorem 5. Assumption 9 is not required for eq. (34), because $\int \mathbf{x} dQ_{mk}(\mathbf{x})$ maximises the Gaussian likelihood regardless of $\boldsymbol{\Sigma}$, cp. S3 and S4 in the proof of Theorem 5. \square

6. Numerical experiments

In order to illustrate the results in Section 5, we present a small simulation study. We computed MLEs for mixtures of two different families of ESDs, namely Gaussian (MLE-N) and multivariate t_5 (MLE-T5), and applied them to mixtures with components that deviate from those assumed. We generated data from three different mixture models, each with $K = 3$, namely a mixture of three bivariate t_3 -distributions, a mixture of three skew normal distributions (Lee and McLachlan (2013)), and a mixture of one skew

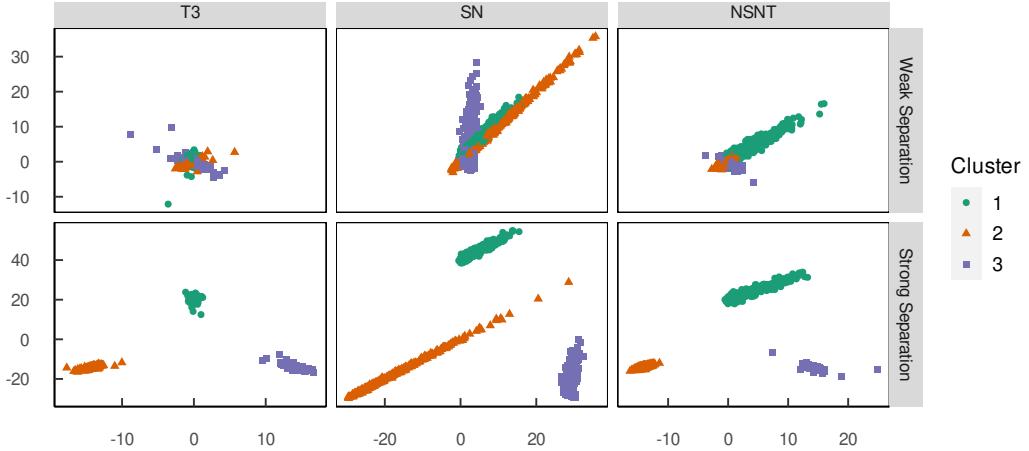


Figure 1: Data sets generated from mixture models T3, SN, and NSNT. “Weak Separation” is obtained setting $s = 0.5$ for T3 and NSNT, and $s = 2$ for SN. “Strong Separation” is obtained setting $s = 20$ for T3 and NSNT, and $s = 40$ for SN. The sample size is set to $n = 1000$ in all the six cases.

normal, one Gaussian and one t_3 -distribution. The previous models are labeled T3, SN, and NSNT respectively. For each model we varied the amount of separation between mixture components. Parameters were chosen as follows. For all models $\pi_1 = \pi_2 = \pi_3 = 1/3$. Given a separation parameter $s > 0$, let

$$\begin{aligned} \boldsymbol{\mu}_1 &= (0, s)^T, \quad \boldsymbol{\mu}_2 = (-s/\sqrt{2}, -s/\sqrt{2})^T, \quad \boldsymbol{\mu}_3 = (s/\sqrt{2}, -s/\sqrt{2})^T, \\ \boldsymbol{\Sigma}_1 &= \begin{pmatrix} 0.1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \boldsymbol{\Sigma}_2 = \begin{pmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{pmatrix}, \quad \boldsymbol{\Sigma}_3 = \begin{pmatrix} 0.55 & -0.45 \\ -0.45 & 0.55 \end{pmatrix}, \\ \boldsymbol{\delta}_1 &= (5, 5)^T, \quad \boldsymbol{\delta}_2 = (15, 15)^T, \quad \boldsymbol{\delta}_3 = (1, 10)^T. \end{aligned}$$

In the T3 model, the three mixture components had means $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3$ and covariance matrices $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \boldsymbol{\Sigma}_3$. s was chosen to range from 0.5 to 20. Skew normal distributions in the SN model were generated according to $\boldsymbol{\mu}_k + \boldsymbol{\delta}_k | Z_0 | + Z_1$, where $Z_0 \sim \mathcal{N}_1(0, 1)$, $Z_1 \sim \mathcal{N}_2(\mathbf{0}, \boldsymbol{\Sigma}_k)$, $k = 1, 2, 3$, where \mathcal{N}_p denotes the p -dimensional Gaussian distribution. If the k -th component of the mixture has a skew normal distribution, its expectation is $\boldsymbol{\mu}_k + \boldsymbol{\delta}_k \sqrt{2/\pi}$, and its covariance matrix is $\boldsymbol{\Sigma}_k + \boldsymbol{\delta}_k \boldsymbol{\delta}_k^T (1 - 2/\pi)$. For this model, the separation parameter s was chosen from the range $s = 2$ to $s = 40$. For the mixed mixture model NSNT, the skew normal component was generated using the parameters $\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \boldsymbol{\delta}_1$, the Gaussian component was generated using the mean $\boldsymbol{\mu}_2$ and the covariance matrix $\boldsymbol{\Sigma}_2$, and the t_3 -component was generated using the mean $\boldsymbol{\mu}_3$ and the covariance matrix $\boldsymbol{\Sigma}_3$. For this model, s was chosen in the interval [1,20]. For each of the three models, we considered a grid of five (logarithmically equispaced) values of s for a total of 15 data generating processes. We also considered two sample sizes, $n = 1000, 10000$, for a total of 30 sample designs. Figure 1 shows examples of datasets with $n = 1000$ generated by the three models, with two different choices of the separation parameter s .

We ran 1000 Monte Carlo replicates with i.i.d. sampling. For each sample, we computed MLE-N and MLE-T5 with $\gamma = 100$. After computing the MLEs, the observations were assigned to clusters using the MAP rule. These were compared with the true mixture components using the Adjusted Rand Index (ARI; [Hubert and Arabie \(1985\)](#)). Results

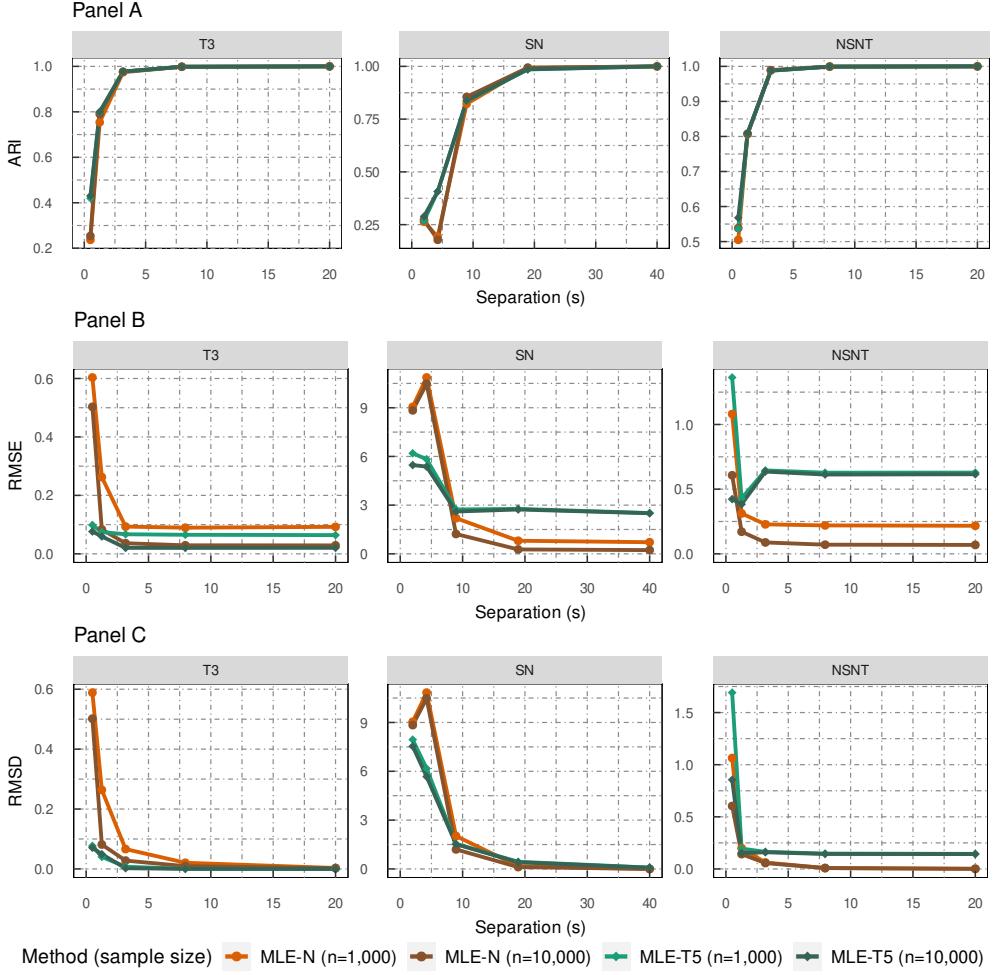


Figure 2: Monte Carlo averages for ARI (Panel A), RMSE for mean parameters (Panel B) and RMSD for ML mean functionals (Panel C).

are shown in Figure 2. Panel A shows the ARI averages over the Monte Carlo runs. Panel B shows the root mean square error (RMSE) of the mean parameters fitted by MLN-N and MLE-T5 compared to the mean of the generating mixture components. Panel C shows the root mean square deviation (RMSD) between the mean parameters estimated by the fit of MLE-N and MLE-T5, respectively, and the corresponding estimators computed on only the observations from the true mixture components, i.e., the sample mean (as to be compared to the MLE-N component mean estimators), and the ML estimator of location of a t_5 -distribution (as to be compared to the MLE-T5 component location estimators). The latter would indicate the best performance one could expect, using true component information, estimating $\tilde{\kappa}_{mk}$ from eq. (26) (Theorem 5) by ML.

The results show that with low separation s estimation and clustering do not work well. For the larger values of s , the ARI becomes almost perfect, which means that the clustering from the misspecified mixture model matches the true component memberships. This illustrates Theorem 4, although with $n = 1000$ already such a good classification is achieved that any improvement with $n = 10000$ is not visible.

The influence of growing n becomes clear looking at Panel B, where the improvement of the RMSE between $n = 1000$ and $n = 10000$ is obvious. For MLE-N the RMSE for $n = 10000$ is close to zero for all three setups. We are interested in particular in examining the RMSE for estimating the mean with the aim of illustrating Corollary 1. For t_3 -distributions, the mean is the center of symmetry, and therefore the ML-estimator that treats the data as t_5 estimates the t_3 -mean as well, and does this better (with lower variance) than the arithmetic mean. Correspondingly, MLE-T5 achieves a lower RMSE than MLE-N for the model T3, but both will converge to zero for $n \rightarrow \infty$. The models SN and NSNT involve asymmetric mixture components for which the ML location functional based on the t_5 -distribution is not the same as the mean. For this reason, the RMSE for MLE-T5 comparing it to the mean will not converge to zero for $n \rightarrow \infty$, and consequently it seems to hit a nonzero floor in Panel B.

Panel C shows that for strong enough separation the location estimators from MLE-N and MLE-T5 become indistinguishable from the corresponding estimators computed based on the true component information for models T3 and SN, as should be expected from Theorem 5 and Panel A. For model NSNT, however, the RMSD of MLE-T5 does not seem to converge to zero. Surprised by this, we found numerically that the ML functional based on the t_5 -distribution computed for the components separately violates $\frac{\lambda_{\min}(\tilde{\theta})}{\lambda_{\max}(\tilde{\theta})} \leq \gamma = 100$, i.e., Assumption 9, and therefore it cannot be recovered even asymptotically by MLE-T5. In fact, also the Gaussian ML functional violates $\frac{\lambda_{\min}(\tilde{\theta})}{\lambda_{\max}(\tilde{\theta})} \leq 100$ here, but this does not affect the consistency of the means from MLE-N, as Assumption 9 is not required for eq. (34).

7. Conclusion

Statistical model assumptions are not normally fulfilled in real life situations, and it is of interest how statistical methodology behaves in cases in which the model assumptions are not fulfilled. Consistency for the canonical functional means that for large data sets the estimator will stabilize even if its distributional assumptions are not fulfilled, although still assuming i.i.d. data. Results of this kind can be found in many places. Additionally here we look at specific underlying distributions P that are of such a kind that using the mixture MLE could be of interest for clustering, namely a mixture with K well separated components that can have a different distributional shape from what is assumed. The components of the canonical functional will then correspond in a well defined sense to the components of P , although potentially very strong separation is required. Without requiring strong enough separation, such a result cannot be had, and actually it would not be desirable, as the components Q_k of P can themselves be heterogeneous. If different Q_k are not well separated, from the point of view of clustering it may be more sensible to split up a heterogeneous, potentially multimodal, Q_k into subgroups rather than separate it from a $Q_l \neq Q_k$ from which it is not separated. That said, investigating the canonical functional in more general situations is certainly of interest. Furthermore, performance guarantees for fixed n would be welcome, although these will require more restrictive assumptions.

The presented results concern the global optimum of the likelihood function whereas existing algorithms are not guaranteed to find it. Similar results for local optima as found by the EM-algorithm would also be a desirable extension.

References

- Anderson, T. and I. Olkin (1985). Maximum-likelihood estimation of the parameters of a multivariate normal distribution. *Linear Algebra and its Applications* 70, 147–171.
- 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, 719–725.
- Chen, J. (2017). Consistency of the MLE under mixture models. *Statistical Science* 32(1), 47–63.
- Ciuperca, G., A. Ridolfi, and J. Idier (2003). Penalized maximum likelihood estimator for normal mixtures. *Scandinavian Journal of Statistics* 30, 45–59.
- Coretto, P. and C. Hennig (2017). Consistency, breakdown robustness, and algorithms for robust improper maximum likelihood clustering. *Journal of Machine Learning Research* 18(142), 1–39.
- 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 (Methodological)* 39(1), 1–38.
- Dennis, J. E. J. (1981). Algorithms for nonlinear fitting. In M. J. D. Powell (Ed.), *Proceedings of the NATO Advanced Research Institute on Nonlinear Optimization held at Trinity Hall, Cambridge*, NATO Conference Series. Series II: Systems Science, London. Academic Press in cooperation with NATO Scientific Affairs Division.
- Frühwirth-Schnatter, S., G. Celeux, and C. P. Robert (Eds.) (2019). *Handbook of Mixture Analysis*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. Boca Raton, FL: CRC Press.
- Gallegos, M. T. and G. Ritter (2009). Trimmed ML estimation of contaminated mixtures. *Sankhyā. The Indian Journal of Statistics* 71(2, Ser.A), 164–220.
- 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.
- García-Escudero, L. A., A. Gordaliza, C. Matrán, and A. Mayo-Iscar (2015). Avoiding spurious local maximizers in mixture modeling. *Statistics and Computing* 25, 1–15.
- Genton, M. G. (Ed.) (2004). *Skew-elliptical distributions and their applications*. Boca Raton, FL: Chapman & Hall CRC.
- Hartigan, J. A. (1981). Consistency of single linkage for high-density clusters. *Journal of the American Statistical Association* 76(374), 388–394.
- Hathaway, R. J. (1985). A constrained formulation of maximum-likelihood estimation for normal mixture distributions. *The Annals of Statistics* 13(2), 795–800.
- Hennig, C. (2010). Methods for merging gaussian mixture components. *Advances in Data Analysis and Classification* 4, 3–34.

- Hennig, C., M. Meila, F. Murtagh, and R. Rocci (Eds.) (2016). *Handbook of cluster analysis*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. Boca Raton, FL: CRC Press.
- Holzmann, H., A. Munk, and T. Gneiting (2006). Identifiability of finite mixtures of elliptical distributions. *Scandinavian Journal of Statistics. Theory and Applications* 33(4), 753–763.
- Hubert, L. and P. Arabie (1985). Comparing partitions. *Journal of Classification* 2(2), 193–218.
- Ingrassia, S. (2004). A likelihood-based constrained algorithm for multivariate normal mixture models. *Statistical Methods & Applications* 13(2), 151–166.
- Jammalamadaka, S. R. and S. Janson (2015). Asymptotic distribution of the maximum interpoint distance in a sample of random vectors with a spherically symmetric distribution. *The Annals of Applied Probability* 25(6), 3571–3591.
- Jennrich, R. I. (1969). Asymptotic properties of non-linear least squares estimators. *The Annals of Mathematical Statistics* 40(2), 633–643.
- Kelker, D. (1970). Distribution theory of spherical distributions and a location-scale parameter generalization. *Sankhyā (Statistics). The Indian Journal of Statistics. Series A* 32, 419–438.
- Kiefer, J. and J. Wolfowitz (1956). Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters. *The Annals of Mathematical Statistics* 27(4), 887–906.
- Lee, S. X. and G. J. McLachlan (2013). Model-based clustering and classification with non-normal mixture distributions (with discussion). *Statistical Methods and Applications* 22, 427–454.
- McLachlan, G. and D. Peel (2000). *Finite mixture models*. Wiley Series in Probability and Statistics: Applied Probability and Statistics. New York: John Wiley & Sons, Inc.
- McLachlan, G. J. and T. Krishnan (1997). *The EM Algorithm and Extensions*. New York: Wiley.
- Peel, D. and G. J. McLachlan (2000). Robust mixture modelling using the t distribution. *Statistics and Computing* 10(4), 339–348.
- Pollard, D. (1981). Strong consistency of k -means clustering. *The Annals of Statistics* 9(1), 135–140.
- Rao, C. R. (1965). *Linear Statistical Inference and its Applications*. New York: Wiley-Interscience.
- Redner, R. (1981). Note on the consistency of the maximum likelihood estimate for nonidentifiable distributions. *The Annals of Statistics* 9(1), 225–228.
- Redner, R. A. and H. F. Walker (1984). Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review* 26, 195–239.
- Ritter, G. (2014). *Robust Cluster Analysis and Variable Selection*. Monographs on Statistics and Applied Probability. Chapman and Hall/CRC.

- Sriperumbudur, B. and I. Steinwart (2012). Consistency and rates for clustering with dbscan. In N. D. Lawrence and M. Girolami (Eds.), *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, Volume 22 of *Proceedings of Machine Learning Research*, La Palma, Canary Islands, pp. 1090–1098. PMLR.
- van der Vaart, A. and J. A. Wellner (1996). *Weak convergence and empirical processes*. Springer Series in Statistics. New York: Springer-Verlag.
- von Luxburg, U., M. Belkin, and O. Bousquet (2008). Consistency of spectral clustering. *The Annals of Statistics* 36(2), 555–586.
- Wald, A. (1949). Note on the consistency of the maximum likelihood estimate. *The Annals of Mathematical Statistics* 20, 595–601.

Keefe Murphy

Material list:

Murphy, K. and Perrakis, K. (2024) AMoRE - Automatic Mixtures of Regularised Experts. WG slides.

Murphy, K. and T. B. Murphy (2020). "Gaussian parsimonious clustering models with covariates and a noise component", Advances in Data Analysis and Classification 14(2): 293–325.

Perrakis, K., T. Lartigue, F. Dondelinger, and S. Mukherjee (2023). "Regularized joint mixture models", Journal of Machine Learning Research 24(1): 677–723.

AMoRE

Automatic Mixtures of Regularised Experts

Keefe Murphy ¹ Konstantinos Perrakis ²

¹Department of Mathematics and Statistics,
Hamilton Institute, Maynooth University

keefe.murphy@mu.ie

²Department of Mathematical Sciences, Durham University

WGMBC: July 24th 2024



MoEClust
oooo

AMoRE
ooooo

Preliminary Results
ooo

Open Issues
oo

Background

MoEClust

Murphy, K. and T. B. Murphy (2020). “Gaussian parsimonious clustering models with covariates and a noise component”, *Advances in Data Analysis and Classification* 14(2): 293–325.



regjmix¹

Perrakis, K., T. Lartigue, F. Dondelinger, and S. Mukherjee (2023). “Regularized joint mixture models”, *Journal of Machine Learning Research* 24(1): 677–723.

¹<https://github.com/k-perrakis/regjmix/>

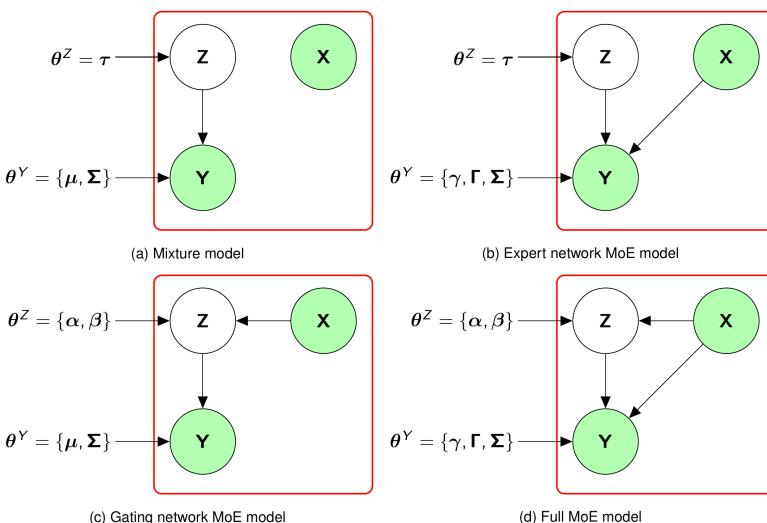
MoEClust

- With responses $\mathbf{Y}_{n \times d}$ and a predictor matrix $\mathbf{X}_{n \times p}$,
MoEClust models address two equivalent aims:
 - 1 incorporating covariates in Gaussian parsimonious clustering models
 - 2 introducing parsimony to Gaussian mixtures of experts models
- Many analyses using finite Gaussian mixtures cluster outcome variables only and typically relate covariates to the uncovered clusters thereafter
- Typical use of unconstrained covariance matrix parameterisation discourages model-selection criteria from favouring models with covariates
- By allowing either, neither, or both the component mixing proportions and component means to depend on different subsets of fixed covariates of mixed type, improvements are demonstrated over both `mclust` models without covariates² and non-parsimonious Gaussian MoE models³

²Banfield & Raftery (1993); Celeux & Govaert (1995)

³Jacobs et al. (1991)

MoE Family



$$f(\mathbf{y}_i | \mathbf{x}_i) = \sum_{g=1}^G \tau_g (\mathbf{x}_i | \boldsymbol{\theta}_g^Z) \phi_d (\mathbf{y}_i | \boldsymbol{\theta}_g^Y (\mathbf{x}_i)) = \{\gamma_{g0} + \boldsymbol{\Gamma}_g \mathbf{x}_i^\top, \boldsymbol{\Sigma}_g\},$$

$$\boldsymbol{\Sigma}_g = \lambda_g \mathbf{D}_g \mathbf{A}_g \mathbf{D}_g^\top, \quad \tau_g (\mathbf{x}_i | \boldsymbol{\theta}_g^Z) = \text{softmax} (\alpha_{g0} + \mathbf{x}_i^\top \boldsymbol{\beta}_g)$$

MoE Fitting

- E-step is standard; for M-step, expected complete data log-likelihood is composed of two terms that can be maximised separately:

$$\mathbb{E}[\ell_c(\cdot)] = \sum_{i=1}^n \sum_{g=1}^G \hat{z}_{ig}^{(t+1)} \log \tau_g(\mathbf{x}_i | \boldsymbol{\theta}_g^Z) + \sum_{i=1}^n \sum_{g=1}^G \hat{z}_{ig}^{(t+1)} \log \phi_d(\mathbf{y}_i | \boldsymbol{\theta}_g^Y(\mathbf{x}_i))$$

- 1st term is of the same form as a *soft* multinomial logistic regression model:

$$\log \frac{\Pr(\hat{z}_{ig}^{(t+1)} = 1)}{\Pr(\hat{z}_{i1}^{(t+1)} = 1)} = \alpha_{g0} + \mathbf{x}_i^\top \boldsymbol{\beta}_g \quad \forall g \geq 2, \text{ where } (\alpha_{10}, \boldsymbol{\beta}_1) = \{0, 0, \dots, 0\}^\top$$

thus methods for fitting such models are used to estimate gating parameters

- Fitting G separate multivariate regressions (weighted by \hat{z}_{ig}), yields G sets of $n \times p$ residuals $\hat{r}_{ig} = \mathbf{y}_i - \gamma_{g0} - \boldsymbol{\Gamma}_g \mathbf{x}_i^\top$, which satisfy $\sum_{i=1}^n \hat{z}_{ig} \hat{r}_{ig} = 0$
- 2nd term can be written in the same form as criterion used in M-step of a standard finite Gaussian mixture model

$$\sum_{i=1}^n \sum_{g=1}^G \hat{z}_{ig} \log |\boldsymbol{\Sigma}_g| + \sum_{i=1}^n \sum_{g=1}^G \hat{z}_{ig} \hat{r}_{ig}^\top \boldsymbol{\Sigma}_g^{-1} \hat{r}_{ig}$$

with component means equal to $\mathbf{0}_d$ to obtain $\hat{\boldsymbol{\Sigma}}_g$ estimates for any GPCM type

Model Selection

- Natural to question which covariates should be included in MoE models and in which of the gating/expert networks
- Model comparison for MoEClust models is even more challenging, as there are potentially 14 different covariance parameterisations to consider
- Ideal situation:
 - 1 Select & fix covariates with application-specific context in mind
 - 2 Fit model(s) and find optimal G and GPCM constraints
 - 3 Repeat both steps & compare MoEClust models with different covariates in different parts of the model in terms of BIC
- As exhaustive searches are infeasible when # covariates is large, we consider a greedy stepwise *forward* search algorithm where each “step” can consist of
 - 1 Adding or removing a component
 - 2 Adding or removing a covariate in the gating network (when $G \geq 2$)
 - 3 Adding or removing a covariate in the expert network
- Each action is evaluated for each set of GPCM constraints as selected variables may only be optimal for a given G & given GPCM type
- The action which yields the best improvement in BIC is accepted and the algorithm proceeds until no further improvement can be found

Issues with Automating MoEClust

- 1 Penalised weighted linear regression in the expert network *and* penalised MLR would be challenging; it is cumbersome to tune penalty parameters in settings with latent groups which require iterative procedures for parameter estimation
- 2 This is exacerbated by the need to choose the optimal G and select among the 14 GPCM types, but would still be challenging with sparsifying penalties⁴ on the component precision matrices $\Omega = \Sigma_g^{-1}$
- 3 The likes of the LASSO — and MLR in general — do not have closed-form solutions, which complicates the EM machinery as the likelihood is only guaranteed to increase without necessarily being maximised at each iteration

We address these issues using prior distributions⁵ to impose regularisation s.t. $f(\mathbf{y}_i | \mathbf{x}_i, \boldsymbol{\theta}) \pi(\boldsymbol{\theta})$ is maximised rather than $f(\mathbf{y}_i | \mathbf{x}_i, \boldsymbol{\theta})$ to find *posterior modes* under a semi-Bayesian multi-cycle ECM strategy⁶ which leaves the usual E-step unaltered but replaces the M-step with a series of conditional maximisations

⁴e.g. Fop, Murphy, & Scrucca (2019)

⁵as per Fraley & Raftery (2007)

⁶Meng & Rubin (1993)

Graphical LASSO

- We henceforth assume the covariates $\mathbf{X}_{n \times p}$ have been normalised
- We assume a graphical LASSO⁷ prior $\pi(\Omega_g) \propto \exp\left(0.5\tilde{\psi}\|\Omega_g\|_1\right)$ on the precision matrices, where $\tilde{\psi} = 0.5\sqrt{2n \log d}$ is the ‘universal’ graphical LASSO penalty⁸
- With $\mathbf{r}_{ig}^{(t)} = (\mathbf{y}_i - \boldsymbol{\gamma}_{g0}^{(t)} - \boldsymbol{\Gamma}_g^{(t)\top} \mathbf{x}_i)$, the solution can be obtained using `glasso`:

$$\Omega_g^{(t+1)} = \arg \max_{\Omega_g^+} \left\{ \log |\Omega_g| - \text{tr}(\Omega_g \mathbf{R}_g^{(t)}) - \omega_g^{(t)} \|\Omega_g\|_1 \right\}.$$

- This is a graphical LASSO objective with penalty $\omega_g^{(t)} = \frac{\tilde{\psi}}{\sum_{i=1}^n \hat{z}_{ig}^{(t)}} = \frac{\sqrt{2n \log d}}{2n_g^{(t)}}$ and $d \times d$ covariance matrix $\mathbf{R}_g^{(t)} = \frac{\sum_{i=1}^n \hat{z}_{ig}^{(t)} \mathbf{r}_{ig}^{(t)\top} \mathbf{r}_{ig}^{(t)}}{n_g^{(t)}}$, under which $\hat{\Omega}_g$ is *sparse*
- For now, our rationale for the universal penalty is that learning Ω_g is not the main goal of the analysis, but regularisation is required to attain workable, non-spurious solutions while preserving parsimony without running 14 models

⁷Friedman, Hastie, & Tibshirani 2008)

⁸Städler & Mukherjee (2013)

Normal Jeffreys Expert Network

- For the un-penalised intercepts, we derive the update $\gamma_{g0}^{(t+1)} = \frac{(\mathbf{Y} - \mathbf{X}\boldsymbol{\Gamma}_g^{(t)})^\top \hat{\mathbf{z}}_g^{(t)}}{n_g^{(t)}}$
- We also derive sparse, closed-form, tuning-free slope updates using the hierarchical form of the normal Jeffreys prior⁹ $\boldsymbol{\gamma}_{g\ell} | \mathbf{S}_{g\ell} \sim \phi_p(\mathbf{0}_p, \mathbf{S}_{g\ell} = \text{diag}(s_{g\ell 1}, \dots, s_{g\ell p}))$, assuming latent $s_{g\ell j}$ with $\pi(s_{g\ell j}) \propto s_{g\ell j}^{-1}$ and $\mathbb{E}_{s|\gamma}[s_{g\ell j}^{-1}] = \gamma_{g\ell j}^{-2}$
- Letting $\mathbf{D} = \mathbf{I}_d \otimes \mathbf{X}$, $\boldsymbol{\delta}_g = (\gamma_{g01}\mathbf{1}_n, \dots, \gamma_{g0d}\mathbf{1}_n)^\top$, and $\boldsymbol{\Phi}_g = \boldsymbol{\Omega}_g \otimes \mathbf{I}_n$, with

$$\mathbf{V}_g^{(t)} = \begin{bmatrix} \mathbf{V}_{g1}^{(t)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{g2}^{(t)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{V}_{gd}^{(t)} \end{bmatrix} \text{ and } \mathbf{Z}_g^{(t)} = \begin{bmatrix} \text{diag}(\hat{\mathbf{z}}_g^{(t)}) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \text{diag}(\hat{\mathbf{z}}_g^{(t)}) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \text{diag}(\hat{\mathbf{z}}_g^{(t)}) \end{bmatrix},$$

where $\mathbf{V}_{g\ell}^{(t)} = \text{diag}(\gamma_{g\ell 1}^{-2(t)}, \dots, \gamma_{g\ell p}^{-2(t)})$, yields the solution

$$\boldsymbol{\Gamma}_g^{(t+1)} = \left(\mathbf{D}^\top \hat{\mathbf{Z}}_g^{\frac{1}{2}(t)} \boldsymbol{\Psi}_g^{(t+1)} \hat{\mathbf{Z}}_g^{\frac{1}{2}(t)} \mathbf{D} + \mathbf{V}_g^{(t)} \right)^{-1} \mathbf{D}^\top \hat{\mathbf{Z}}_g^{\frac{1}{2}(t)} \boldsymbol{\Psi}_g^{(t+1)} \hat{\mathbf{Z}}_g^{\frac{1}{2}(t)} (\text{vec}(\mathbf{Y}) - \boldsymbol{\delta}_g^{(t+1)})$$

⁹Figueiredo (2001)

Pólya-gamma Gating Network

- The NJ prior substantially shrinks small coefficients to exactly 0 while leaving larger coefficients relatively unaffected, but we need an analytic expression for the parameter updates in order to be able to use it for the gating network
- We distinguish between the intercepts α_{g0} and penalised slopes β_g and use

$$\tau_g(\mathbf{x}_i | \boldsymbol{\theta}_g^Z) = \frac{\exp(\alpha_{g0} + \mathbf{x}_i^\top \boldsymbol{\beta}_g)}{\sum_{h=1}^G \exp(\alpha_{h0} + \mathbf{x}_i^\top \boldsymbol{\beta}_h)}, \quad \mathcal{L}(\boldsymbol{\theta}_g^Z | \boldsymbol{\theta}_{-g}^Z; \mathbf{z}) = \prod_{i=1}^n \frac{\exp(\eta_{ig})^{\mathbb{1}(z_{ig}=1)}}{1 + \exp(\eta_{ig})},$$

$$\eta_{ig} = \alpha_{g0} + \mathbf{x}_i^\top \boldsymbol{\beta}_g - c_{ig}, \quad c_{ig} = \log \sum_{h=1, h \neq g}^G \exp(\alpha_{h0} + \mathbf{x}_i^\top \boldsymbol{\beta}_h)$$

to apply a Pólya-gamma data augmentation scheme¹⁰, leveraging the following identity for binary logistic regression, in MLR settings where $G > 2$

$$f(\eta) = \frac{\exp(\eta)^a}{(1 + \exp(\eta))^b} = 2^{-b} \exp(\eta(a - b/2)) \int_0^\infty \exp(-\delta\eta^2/2) \pi(\delta) d\delta,$$

where $d \sim \text{PG}(b, 0)$, with $a = \mathbb{1}(z_{ig} = 1)$ and $b = 1$

¹⁰Polson et al. (2013)

Normal Jeffreys Gating Network

- Expressing the softmax as a Bernoulli distribution allows writing the conditional log-likelihood for $\theta_g^Z | \theta_{-g}^Z$ as a logistic log-likelihood
- Ultimately, we update $\theta_2^Z, \dots, \theta_G^Z$ by a sequential set of conditional EM steps¹¹ using an augmented logistic likelihood which is recognisable as a Gaussian kernel

$$\mathcal{L}(\boldsymbol{\theta}_g^Z | \boldsymbol{\theta}_{-g}^Z; \mathbf{z}, \boldsymbol{\delta}) = \prod_{i=1}^n \exp \left(\eta_{ig} (\mathbb{1}(z_{ig} = 1) - 1/2) - \delta_{ig} \eta_{ig}^2 / 2 \right),$$

- The expectations $\hat{z}_{ig}^{(t)}$ and $\hat{\delta}_{ig}^{(t)} = \frac{\tanh(\eta_{ig}^{(t)} / 2)}{2\eta_{ig}^{(t)}}$ are updated before each CM 'cycle'
 - With the conjugate NJ prior $\beta_g | \mathbf{Q}_g \sim \phi_p(\mathbf{0}_p, \mathbf{Q}_g)$, $\pi(q_{gj}) \propto q_{gj}^{-1}$, and $\kappa_{ig}^{(t)} = c_{ig}^{(t)} - \mathbf{x}_i^\top \boldsymbol{\beta}_g^{(t)} + (\hat{z}_{ig}^{(t)} - 0.5) / \hat{\delta}_{ig}^{(t)}$, $\mathbf{B}_g^{(t)} = \text{diag}(\beta_{g1}^{-2(t)}, \dots, \beta_{gp}^{-2(t)})$, $\zeta_{ig}^{(t+1)} = c_{ig}^{(t)} - \alpha_{g0}^{(t+1)} + (\hat{z}_{ig}^{(t)} - 0.5) / \hat{\delta}_{ig}^{(t)}$, $\hat{\Delta}_g^{(t)} = \text{diag}(\hat{\delta}_{ig}^{(t)}, \dots, \hat{\delta}_{ng}^{(t)})$, we obtain
- $$\alpha_{g0}^{(t+1)} = \frac{\sum_{i=1}^n \hat{\delta}_{ig}^{(t)} \kappa_{ig}^{(t)}}{\sum_{i=1}^n \hat{\delta}_{ig}^{(t)}} \quad \text{and} \quad \boldsymbol{\beta}_g^{(t+1)} = (\mathbf{X}^\top \hat{\Delta}_g^{(t)} \mathbf{X} + \mathbf{B}_g^{(t)})^{-1} \mathbf{X}^\top \hat{\Delta}_g^{(t)} \zeta_g^{(t+1)}$$

- Regularisation via the sparsity-inducing NJ prior can also alleviate separability issues

¹¹Durante et al. (2019)

AIS Data¹²: MoEClust

- Using $p = 5$ hematological variables and only the covariates 'sex' & (normalised) 'BMI', an exhaustive search with $G = 1, \dots, 9$ still requires fitting 1,804 models
- Greedy stepwise search identifies the same solution, yet still visits 138 models

Step	Optimal Action	BIC	G	GPCM	Gating	Expert	# Models
1	—	-4202.8	1	EEE	—	—	3
2	Add expert covariate	-4050.6	1	EEE	—	sex	20
3	Add component	-4015.4	2	EVE	—	sex	17
4	Add gating covariate	-4013.4	2	EVE	BMI	sex	56
5	STOP	-4013.4	2	EVE	BMI	sex	42

- The gating network:

Cluster	(Intercept)	sex(male)	BMI
1	—	—	—
2	-0.26	—	0.60

- The expert network:

	Cluster 1					Cluster 2				
	RCC	WCC	Hc	Hg	Fe	RCC	WCC	Hc	Hg	Fe
(Intercept)	4.47	6.71	40.82	13.61	45.90	4.30	7.46	39.94	13.47	74.80
sex(male)	0.64	0.01	5.03	1.94	15.90	0.64	0.29	5.51	2.08	57.20
BMI	—	—	—	—	—	—	—	—	—	—

¹²Cook & Weisberg (1994)

AIS Data: AMoRE

- Running AMoRE once per $G = 1, \dots, 9$ also identifies a $G = 2$ model as optimal
- The gating network:

Cluster	(Intercept)	sex(male)	BMI
1	—	—	—
2	-0.97	—	1.22

- The expert network:

	Cluster 1					Cluster 2				
	RCC	WCC	Hc	Hg	Fe	RCC	WCC	Hc	Hg	Fe
(Intercept)	4.38	6.55	40.49	13.51	51.87	4.54	8.36	40.80	13.89	78.71
sex(male)	0.57	—	4.65	1.79	19.01	0.57	—	5.44	2.00	54.74
BMI	—	—	—	—	—	—	—	—	—	—

- GLASSO precision matrices $\hat{\Omega}_g$:

	Cluster 1					Cluster 2				
	RCC	WCC	Hc	Hg	Fe	RCC	WCC	Hc	Hg	Fe
RCC	7.080	—	-0.644	—	0.001	3.711	—	-0.310	—	0.003
WCC		0.617	-0.075	—	0.005		0.218	—	0.035	0.001
Hc			0.602	-1.151	0.004			0.268	-0.477	—
Hg				3.889	-0.007				1.682	0.003
Fe					0.002					0.001

- The ARI for this vs. MoEClust is only 0.20 but the AMoRE BIC (-4006.3) is superior to MoEClust (-4013.4)

AIS Data: Hybrid AMoRE-GPCM

- Running AMoRE for $G = 1, \dots, 9$ with all 14 GPCM parameterisations instead of glasso also identifies a $G = 2$ (EVE) model as optimal
- The gating network:

Cluster	(Intercept)	BMI	sex(male)
1	—	—	—
2	-0.33	0.50	—

- The expert network:

	Cluster 1					Cluster 2				
	RCC	WCC	Hc	Hg	Fe	RCC	WCC	Hc	Hg	Fe
(Intercept)	4.47	6.71	40.76	13.59	46.76	4.32	7.66	40.21	13.58	76.59
BMI	—	—	—	—	—	—	—	—	0.09	—
sex(male)	0.62	—	4.96	1.92	14.63	0.61	—	5.21	1.91	56.62

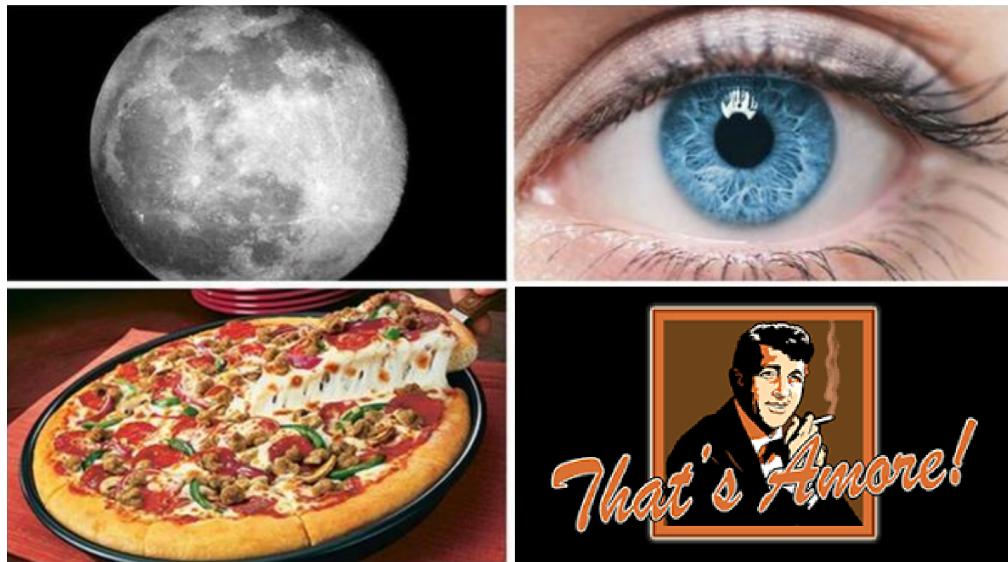
- EVE covariance matrices $\hat{\Sigma}_g$:

	Cluster 1					Cluster 2				
	RCC	WCC	Hc	Hg	Fe	RCC	WCC	Hc	Hg	Fe
RCC	0.133	0.142	0.902	0.310	0.616	0.073	0.051	0.497	0.171	2.477
WCC		1.815	1.149	0.339	-2.063		4.550	0.133	-0.105	-8.307
Hc			8.179	2.693	-0.082			4.544	1.526	-0.346
Hg				1.103	0.278				0.594	1.15
Fe					482.459					1940.832

- The ARI for this vs. MoEClust is 0.96 and the BIC (-4000.4) is superior to AMoRE with the graphical LASSO (-4006.3)

Future Work

- Early results are encouraging as AMoRE allows component-specific sparsity: covariates can enter the model without necessarily affecting each part, each component, and/or each element of the response
- A notable issue is that the $\beta_g^{(t+1)}$ & $\Gamma_g^{(t+1)}$ updates depend on $\beta_g^{(t)}$ & $\Gamma_g^{(t)}$ from previous iterations, so exploring sensitivity to initialisation will be crucial
- We currently use model-based agglomerative hierarchical clustering on (\mathbf{Y}, \mathbf{X}) and employ an iterative reallocation scheme from MoEClust, with initial expert and gating network coefficients obtained using *un-penalised* weighted linear and weighted multinomial logistic regressions
- All that remains to be chosen is G , the number of mixture components; we use the BIC with the number of *non-zero* parameters in θ^Z and θ^Y , as choosing G automatically would require a fully-Bayesian approach
- We also need to
 - Conduct more thorough simulation studies and find interesting applications
 - Evaluate the model in $n \ll p$ settings & further explore $d \ll p$ cases
 - Investigate alternatives to the universal graphical LASSO penalty for component precision matrices
 - Integrate NJ priors on expert and gating networks (or simply un-penalised Pólya-gamma gating) into MoEClust with existing 14 GPCM models



Thank you!

Appendix

Additional AIS Results

Appendix
•••

AIS: AMoRE — All Covariates

- Running AMoRE using all covariates also selects a $G = 2$ (EVE) model, again with sparse $\hat{\Omega}_g$ matrices
- The gating network:

Cluster	(Intercept)	sex(male)	BMI	field	gym	netball	row	swim	400m	sprint	tennis	wpolo	SSF	Bfat	LBM	Ht	Wt
1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
2	—0.75	1.59	—	—	—	4.91	—	—	—	—	—	—	—	—	—	—	—

- The expert network:

	Cluster 1					Cluster 2				
	RCC	WCC	Hc	Hg	Fe	RCC	WCC	Hc	Hg	Fe
(Intercept)	4.56	6.95	41.99	14.08	47.43	4.22	6.19	38.71	12.92	66.17
sex(male)	0.58	—	4.22	1.63	—	0.78	0.77	6.74	2.60	48.52
BMI	—	—	—	—	—	—	—	—	—	—
field	—	—	—	—	—	—	—	—	—	—
gym	—	—	—	—	—	—	—	—	—	—
netball	—	—	—	—	—	—	—	1.59	—	—
row	—	—	—	—	—	—	—	—	—	—
swim	—	—	—	—	—	—	—	—	—	—
400m	—	—	—	—	—	—	—	—	—	—
sprint	—	—	—	—	—	—	—	—	—	—
tennis	—	—	—	—	—	—	—	—	—	39.12
wpolo	—	—	—	—	—	—	—	2.11	—	—
SSF	—	—	—	—	—	—	—	—	—	—
Bfat	—	—	—	—	—	—	—	—	—	—
LBM	—	—	—	—	—	—	—	—	—	—
Ht	—	—	—	—	—	—	—	—	—	—
Wt	—	—	—	—	—	—	—	—	—	—

- The ARI for this vs. MoEClust is just 0.23 but the AMoRE BIC (-3991.5) outperforms AMoRE with only 'sex', 'BMI' and the graphical LASSO (-4006.3)

AIS: Hybrid AMoRE-GPCM — All Covariates

- Running the hybrid AMoRE using all covariates also selects a $G = 2$ (EVE) model
- The gating network:

Cluster	(Intercept)	sex(male)	BMI	field	gym	netball	row	sport	swim	400m	sprint	tennis	wpolo	SSF	Bfat	LBM	Ht	Wt
1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
2	-1.97	2.32	—	—	—	4.82	2.35	4.87	—	3.13	1.73	—	—	—	—	—	-0.95	0.73

- The expert network:

	Cluster 1					Cluster 2				
	RCC	WCC	Hc	Hg	Fe	RCC	WCC	Hc	Hg	Fe
(Intercept)	4.49	7.03	41.58	13.79	44.74	4.29	6.68	40.60	13.78	69.86
sex(male)	0.49	—	3.74	1.74	—	0.70	—	4.92	1.80	42.96
BMI	—	—	—	—	—	—	—	—	—	—
field	0.22	—	—	—	—	—	—	—	—	—
gym	—	—	—	—	—	—	—	—	—	—
netball	—	6.03	—	—	—	—	0.91	-1.77	-0.80	—
row	—	—	—	—	—	—	—	—	—	—
swim	—	—	—	—	—	—	—	—	-0.16	—
400m	—	—	—	—	—	—	—	—	—	—
sprint	0.22	—	—	—	—	—	—	—	—	—
tennis	—	—	—	—	—	0.24	—	—	-0.24	—
wpolo	—	—	—	—	—	—	2.30	—	-0.19	—
SSF	—	—	-0.31	—	—	—	—	—	—	—
Bfat	—	—	—	—	—	—	—	—	—	—
LBM	—	—	—	—	—	—	—	—	—	—
Ht	—	—	—	-0.10	—	—	—	—	—	—
Wt	—	—	—	—	—	—	—	—	—	—

- The ARI for this vs. MoEClust is just 0.20 but the BIC (-3957.1) is now superior to AMoRE with all covariates and the graphical LASSO (-3991.5)

AIS: AMoRE with Constrained GLASSO

- Running AMoRE for $G = 1, \dots, 9$ with a common Ω matrix with $\omega = \sqrt{\frac{\log d}{2n}}$ and $\mathbf{R}^{(t)} = n^{-1} \sum_{i=1}^n \sum_{g=1}^G \widehat{z}_{ig}^{(t)} \mathbf{r}_{ig}^{(t)\top} \mathbf{r}_{ig}^{(t)}$ also selects a $G = 2$ model
- The gating network:

Cluster	(Intercept)	sex(male)	BMI
1	—	—	—
2	-1.81	—	0.60

- The expert network:

	Cluster 1					Cluster 2				
	RCC	WCC	Hc	Hg	Fe	RCC	WCC	Hc	Hg	Fe
(Intercept)	4.42	6.87	40.58	13.58	51.14	4.39	8.39	40.33	13.65	112.58
sex(male)	0.60	—	4.97	1.92	23.59	0.63	—	5.36	1.97	63.68
BMI	—	—	—	—	—	—	—	—	0.08	—

- GLASSO precision matrix $\widehat{\Omega}$:

	RCC	WCC	Hc	Hg	Fe
RCC	9.343	—	-0.939	—	—
WCC	0.350	—	-0.031	—	0.003
Hc			0.604	-1.150	0.002
Hg				3.698	-0.005
Fe					0.001

- The ARI for this vs. MoEClust is only 0.20 and the AMoRE BIC (-4161.4) is inferior to MoEClust (-4013.4)

AIS: AMoRE with Constrained GLASSO — All Covariates

- Running AMoRE with a common Ω using all covariates also selects a $G = 2$ model
- The gating network:

Cluster	sport													SSF	Bfat	LBM	Ht	Wt
	(Intercept)	sex(male)	BMI	field	gym	netball	row	swim	400m	sprint	tennis	wpolo						
1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
2	—1.31	—	—	—	—	—	—	—	—	—	—	2.34	0.77	—	—	—	-0.73 1.14	

- The expert network:

	Cluster 1					Cluster 2					SSF	Bfat	LBM	Ht	Wt
	RCC	WCC	Hc	Hg	Fe	RCC	WCC	Hc	Hg	Fe					
(Intercept)	4.38	4.93	40.29	13.49	49.25	4.48	5.04	40.95	13.75	52.39					
sex(male)	0.60	—	5.03	1.92	19.71	0.57	2.31	4.83	1.92	133.88					
BMI	—	0.42	—	—	—	—	—	—	—	—					
field	0.22	—	—	—	—	—	2.25	—	—	—	58.30				
gym	—	—	—	—	—	—	—	—	—	—					
netball	—	—	—	—	—	—	4.72	—	—	—					
row	—	—	—	—	—	—	1.49	—	—	—					
swim	—	—	—	—	—	—	—	—	—	—					
400m	—	—	—	—	—	—	—	—	—	—					
sprint	—	—	—	—	—	—	—	—	—	—					
tennis	—	—	—	—	—	—	—	—	—	—					
wpolo	—	—	—	—	—	—	3.72	—	—	—					
SSF	—	—	—	—	—	—	—	—	—	—					
Bfat	—	—	—	—	—	—	—	—	—	—					
LBM	—	—	—	—	—	—	1.10	—	—	—	-38.13				
Ht	—	—	—	—	—	—	—	—	—	—					
Wt	—	—	—	—	—	—	—	—	—	—					

- The ARI for this vs. MoEClust is only 0.30 but the AMoRE BIC (-4137.0) is superior to AMoRE with a common Ω using only 'sex' and 'BMI' (-4161.4)

This is a preprint. The revised version of this paper is published as

K. Murphy and T. B. Murphy (2020) “Gaussian parsimonious clustering models with covariates and a noise component”. *Advances in Data Analysis and Classification*, 14(2): 293–325. [doi: [10.1007/s11634-019-00373-8](https://doi.org/10.1007/s11634-019-00373-8)].

Gaussian Parsimonious Clustering Models with Covariates and a Noise Component

Keefe Murphy¹ Thomas Brendan Murphy^{2,3}

keefe.murphy@mu.ie brendan.murphy@ucd.ie

¹ Department of Mathematics and Statistics, Maynooth University

² School of Mathematics and Statistics, University College Dublin

³ Insight Centre for Data Analytics, University College Dublin

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

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 $f(\mathbf{y}_i | \mathbf{x}_i, z_{ig} = 1) \Pr(z_{ig} = 1)$ with $f(\mathbf{y}_i | \mathbf{x}_i, z_{ig} = 1) f(\mathbf{x}_i | z_{ig} = 1) \Pr(z_{ig} = 1)$. 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 & 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 $f(\mathbf{y}_i | \mathbf{x}_i, z_{ig} = 1) \Pr(z_{ig} = 1 | \mathbf{x}_i)$.

This is achieved using the mixture of experts (MoE) paradigm ([Dayton & 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 framework 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 & Raftery, 1993](#); [Celeux & Govaert, 1995](#)). This has, to date, been lacking for all but mixtures of regressions and mixtures of regressions with concomitant variables where the same covariates enter both parts of the model ([Dang & McNicholas, 2015](#)).

Parsimony is obtained in GPCMs by imposing 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, proposed solutions to initialising the EM algorithm sensibly, addressing issues of model selection, 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 of the full suite of MoEClust models is provided by the associated R package `MoEClust` ([Murphy & Murphy, 2021](#)), with which all results were obtained, which is available from <https://www.r-project.org> ([R Core Team, 2021](#)). 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 and Section 3. Section 3.3 describes the addition of a noise component for capturing outliers. Section 4 discusses proposals for addressing some practical issues affecting performance, namely the initialisation of the EM algorithm used to fit the models (Section 4.1), and issues around model selection (Section 4.2). The performance of the proposed models is illustrated in Section 5 with applications to univariate response CO₂ emissions data (Section 5.1) and multivariate response data from the Australian Institute of Sports (Section 5.2). Finally, the paper concludes with a brief discussion in Section 6, with some additional results deferred to the Appendices.

2 Modelling

This section builds up the MoEClust models by first describing the mixture of experts (MoE) modelling framework in Section 2.1 — elaborating on the special cases of the MoE model in Section 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 and 2.3. Finally, a brief review of existing models and software is given in Section 2.4.

2.1 Mixtures of Experts

The mixture of experts model (Dayton & 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 $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$, is modelled by a G -component finite mixture model where the model parameters depend on the associated covariate inputs $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ 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:

$$f(\mathbf{y}_i | \mathbf{x}_i) = \sum_{g=1}^G \tau_g(\mathbf{x}_i | \boldsymbol{\beta}_g) f(\mathbf{y}_i | \boldsymbol{\theta}_g(\mathbf{x}_i)).$$

Each component is modelled by a probability density function $f(\mathbf{y}_i | \boldsymbol{\theta}_g(\mathbf{x}_i))$ with component-specific parameters $\boldsymbol{\theta}_g(\mathbf{x}_i)$ and mixing proportions $\tau_g(\mathbf{x}_i | \boldsymbol{\beta}_g)$ (henceforth $\tau_g(\mathbf{x}_i)$, for simplicity); 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 \mathbf{x}_i . The terminology used to describe MoE models in the machine learning literature often refers to the component densities $f(\mathbf{y}_i | \boldsymbol{\theta}_g(\mathbf{x}_i))$ 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 & Keane (2007) impose a multinomial probit structure here instead, while Xu et al. (1994) effectively obtain a CWM (albeit with strictly continuous covariates) by assuming Gaussian gating functions. The mixture components (expert networks) are generalised linear models (GLM; McCullagh & Nelder, 1983). Thus,

$$\hat{\tau}_g(\mathbf{x}_i) = \frac{\exp(\tilde{\mathbf{x}}_i \hat{\boldsymbol{\beta}}_g)}{\sum_{h=1}^G \exp(\tilde{\mathbf{x}}_i \hat{\boldsymbol{\beta}}_h)}, \quad (1)$$

and

$$\hat{\boldsymbol{\theta}}_g(\mathbf{x}_i) = \left\{ \psi(\tilde{\mathbf{x}}_i \hat{\boldsymbol{\gamma}}_g), \hat{\boldsymbol{\Sigma}}_g \right\}, \quad (2)$$

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 $\hat{\boldsymbol{\gamma}}_g$ and the $p \times p$ component covariance matrix $\hat{\boldsymbol{\Sigma}}_g$), a $(d+1)$ -dimensional vector of regression parameters $\hat{\boldsymbol{\beta}}_g$ in the gates in (1), and $\tilde{\mathbf{x}}_i = (1, \mathbf{x}_i)$. 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) is simply the identity, such that covariates are linearly related to the response variables, i.e.

$$f(\mathbf{y}_i | \mathbf{x}_i) = \sum_{g=1}^G \tau_g(\mathbf{x}_i) \phi(\mathbf{y}_i | \boldsymbol{\theta}_g(\mathbf{x}_i) = \{\tilde{\mathbf{x}}_i \boldsymbol{\gamma}_g, \boldsymbol{\Sigma}_g\}). \quad (3)$$

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 1, with the following interpretations, due to Gormley & 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 : $f(\mathbf{y}_i) = \sum_{g=1}^G \tau_g \phi(\mathbf{y}_i | \boldsymbol{\theta}_g = \{\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\})$.
- (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 : $f(\mathbf{y}_i | \mathbf{x}_i) = \sum_{g=1}^G \tau_g \phi(\mathbf{y}_i | \boldsymbol{\theta}_g(\mathbf{x}_i) = \{\tilde{\mathbf{x}}_i \boldsymbol{\gamma}_g, \boldsymbol{\Sigma}_g\})$.
- (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 : $f(\mathbf{y}_i | \mathbf{x}_i) = \sum_{g=1}^G \tau_g(\mathbf{x}_i) \phi(\mathbf{y}_i | \boldsymbol{\theta}_g = \{\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\})$.
- (d) in the *full MoE model*, given by (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: $(\mathbf{y}_i | \mathbf{x}_i, z_{ig} = 1) \sim \phi(\mathbf{y}_i | \boldsymbol{\theta}_g(\mathbf{x}_i) = \{\tilde{\mathbf{x}}_i \boldsymbol{\gamma}_g, \boldsymbol{\Sigma}_g\})$, $\Pr(z_{ig} = 1 | \mathbf{x}_i) = \tau_g(\mathbf{x}_i)$.

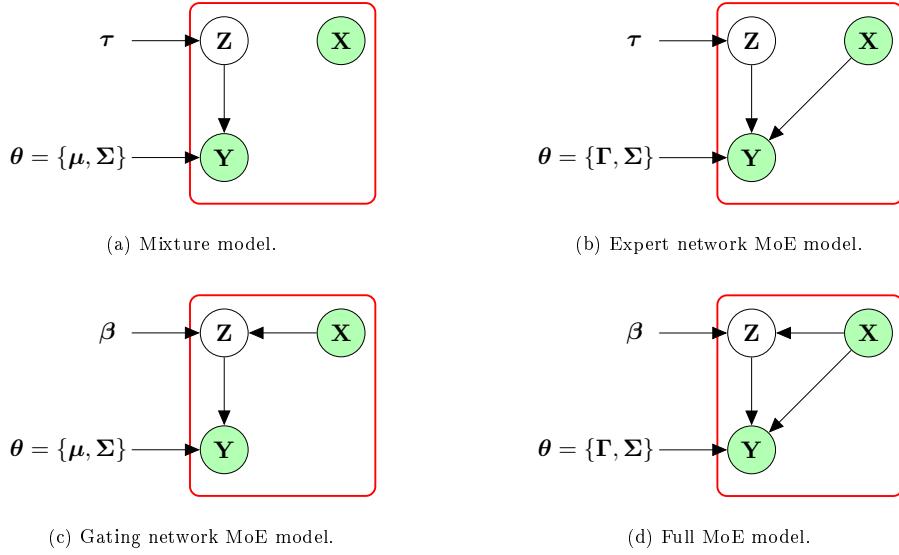


Figure 1: The graphical model representation of the mixture of experts models. The differences between the special cases are due to the presence or absence of edges between the covariates \mathbf{X} and the latent variables \mathbf{Z} and/or response variables \mathbf{Y} . Note that different subsets of the covariates in \mathbf{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 τ , 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 & 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 $\Pr(z_{ig} = 1 | \mathbf{x}_i) = \tau_g(\mathbf{x}_i)$ (MoE) or $f(\mathbf{x}_i | z_{ig} = 1) \Pr(z_{ig} = 1)$ (CWM). In other words, the direction of the edge between \mathbf{X} and \mathbf{Z} in the full MoE model in Figure 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 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 & Raftery (1993) and Celeux & 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 & Murphy, 2008, 2010), the multivariate t -distribution and associated t EIGEN family (Andrews & McNicholas, 2012), and the multivariate contaminated normal distribution (Punzo & 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 $\Sigma_g = \lambda_g \mathbf{D}_g \mathbf{A}_g \mathbf{D}_g^\top$, where λ_g is a scalar controlling the volume, \mathbf{A}_g is a diagonal matrix, with entries proportional to the eigenvalues of Σ_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 Σ_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 1 summarises the geometric characteristics of the GPCM constraints, which are then shown in Figure 2.

Note for models with names ending with \dagger 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.

Table 1: Nomenclature, descriptions, and parameter counts of the parameterisations of the component covariance matrices Σ_g available under GPCMs — all of which are available when there is no dependency in any way on covariates — in increasing order of complexity. \dagger indicates availability in the first four special cases of the Gaussian MoE framework shown in Figure 1 and the MoEClust family; \bullet 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	\dagger	\bullet		(univariate)	equal			1
V	σ_g		\dagger		(univariate)	variable			G
EII	$\lambda \mathcal{I}$	\dagger	\bullet	\bullet	spherical	equal	equal	—	1
VII	$\lambda_g \mathcal{I}$		\bullet	\bullet	spherical	variable	equal	—	G
EEI	$\lambda \mathbf{A}$	\bullet	\bullet	\bullet	diagonal	equal	equal	axis-aligned	p
VEI	$\lambda_g \mathbf{A}$		\bullet	\bullet	diagonal	variable	equal	axis-aligned	$G + (p-1)$
EVI	$\lambda \mathbf{A}_g$		\bullet	\bullet	diagonal	equal	variable	axis-aligned	$1 + G(p-1)$
VVI	$\lambda_g \mathbf{A}_g$		\dagger	\dagger	diagonal	variable	variable	axis-aligned	Gp
EEE	$\lambda \mathbf{D} \mathbf{A} \mathbf{D}^\top$	\bullet	\bullet		ellipsoidal	equal	equal	equal	$p(p+1)/2$
VEE	$\lambda_g \mathbf{D} \mathbf{A} \mathbf{D}^\top$		\bullet		ellipsoidal	variable	equal	equal	$G + p(p-1)/2 + (p-1)$
EVE	$\lambda \mathbf{D}_g \mathbf{A}_g \mathbf{D}^\top$		\bullet		ellipsoidal	equal	variable	equal	$1 + p(p-1)/2 + G(p-1)$
VVE	$\lambda_g \mathbf{D}_g \mathbf{A}_g \mathbf{D}^\top$		\bullet		ellipsoidal	variable	variable	equal	$G + p(p-1)/2 + G(p-1)$
EEV	$\lambda \mathbf{D}_g \mathbf{A} \mathbf{D}^\top$		\bullet		ellipsoidal	equal	equal	variable	$1 + Gp(p-1)/2 + (p-1)$
VEV	$\lambda_g \mathbf{D}_g \mathbf{A} \mathbf{D}^\top$		\bullet		ellipsoidal	variable	equal	variable	$G + Gp(p-1)/2 + (p-1)$
EVV	$\lambda \mathbf{D}_g \mathbf{A}_g \mathbf{D}^\top$		\bullet		ellipsoidal	equal	variable	variable	$Gp(p+1)/2 - (G-1)$
VVV	$\lambda_g \mathbf{D}_g \mathbf{A}_g \mathbf{D}^\top$		\dagger		ellipsoidal	variable	variable	variable	$Gp(p+1)/2$

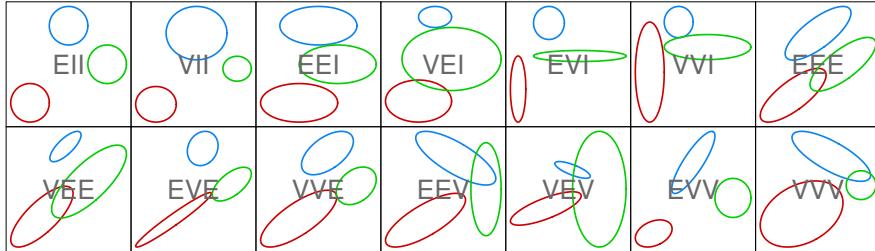


Figure 2: Ellipses of isodensity for each of the 14 parsimonious eigen-decomposition covariance parameterisations for multivariate data in GPCMs, with three components in two dimensions.

2.3 The MoEClust Family of Models

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

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

where $\boldsymbol{\Sigma}_g$ can follow any of the GPCM constraints outlined in Table 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 2.1.1, 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 4.2).

2.4 Existing Models and Software

A number of tools for fitting MoE models are available in the R programming environment (R Core Team, 2021). These include **flexmix** (Grün & 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 & Leisch, 2007, 2008) can accommodate the full range of MoE models outlined in Section 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 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 1, and are thus examples of existing software which can fit GPCMs, but only using the standard finite mixture model (model (a) in Figure 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 & 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 & 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 & 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.

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 3.1 and then extended to MoEClust models in Section 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 3.3. When gating concomitants are present, the noise component is treated in two different ways.

3.1 Fitting MoE Models

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

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

where $\tau_g(\mathbf{x}_i^{(G)})$ and $\boldsymbol{\theta}_g(\mathbf{x}_i^{(E)})$ are as defined by (1). The data are augmented by imputing the latent cluster membership indicator $\mathbf{z}_i = (z_{i1}, \dots, z_{iG})^\top$. Thus, the conditional distribution of $(\mathbf{y}_i, \mathbf{z}_i | \mathbf{x}_i)$ is of the form

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

Hence, the complete data likelihood is of the form

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

and the complete data log-likelihood has the form

$$\begin{aligned} \ell_c(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Sigma} | \mathbf{Y}, \mathbf{X}, \mathbf{Z}) &= \sum_{i=1}^n \sum_{g=1}^G z_{ig} \left[\log \tau_g(\mathbf{x}_i^{(G)}) + \log \phi(\mathbf{y}_i | \boldsymbol{\theta}_g(\mathbf{x}_i^{(E)})) \right] \\ &= \sum_{i=1}^n \sum_{g=1}^G z_{ig} \log \tau_g(\mathbf{x}_i^{(G)}) + \sum_{i=1}^n \sum_{g=1}^G z_{ig} \log \phi(\mathbf{y}_i | \boldsymbol{\theta}_g(\mathbf{x}_i^{(E)})). \end{aligned} \tag{4}$$

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 $\hat{\mathbf{z}}_i$, followed by a M-step (maximisation) which maximises the expected complete data log-likelihood, computed with the estimates $\hat{\mathbf{Z}} = (\hat{\mathbf{z}}_1, \dots, \hat{\mathbf{z}}_n)$, to provide estimates of the component weight parameters $\hat{\tau}_g(\mathbf{x}_i^{(G)})$ and the component parameters $\hat{\boldsymbol{\theta}}_g(\mathbf{x}_i^{(E)})$. 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

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

where $\{\hat{\boldsymbol{\beta}}^{(t)}, \hat{\boldsymbol{\gamma}}^{(t)}, \hat{\boldsymbol{\Sigma}}^{(t)}\}$ 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 (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 (5) can be maximised separately under the EM framework:

$$\begin{aligned} \mathbb{E}\left[\ell_c(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Sigma} | \mathbf{Y}, \mathbf{X}, \mathbf{Z}, \widehat{\boldsymbol{\beta}}^{(t)}, \widehat{\boldsymbol{\gamma}}^{(t)}, \widehat{\boldsymbol{\Sigma}}^{(t)})\right] &= \sum_{i=1}^n \sum_{g=1}^G \widehat{z}_{ig}^{(t+1)} \log \tau_g(\mathbf{x}_i^{(G)}) \\ &\quad + \sum_{i=1}^n \sum_{g=1}^G \widehat{z}_{ig}^{(t+1)} \log \phi(\mathbf{y}_i | \boldsymbol{\theta}_g(\mathbf{x}_i^{(E)})). \end{aligned} \quad (5)$$

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:

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

Thus, methods for fitting such models can be used to estimate the parameters of the gating network. However, due to the nonlinear numerical optimisation involved in estimating MLR coefficients, for which no closed-form updates are available, we caution that this step merely increases the expectation of this portion of (5) at each iteration, without explicitly maximising it. Nevertheless, the monotonicity of the EM is preserved and the procedure is still guaranteed to converge (Dempster et al., 1977).

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.

3.2 Fitting MoEClust Models

Maximising the second term in (5), corresponding to the expert network, gives rise to the following expression

$$\begin{aligned} -\frac{1}{2} \left(p \log 2\pi + \sum_{i=1}^n \sum_{g=1}^G \widehat{z}_{ig}^{(t+1)} \log |\boldsymbol{\Sigma}_g| + \right. \\ \left. \sum_{i=1}^n \sum_{g=1}^G \widehat{z}_{ig}^{(t+1)} (\mathbf{y}_i - \tilde{\mathbf{x}}_i^{(E)} \boldsymbol{\gamma}_g)^\top \boldsymbol{\Sigma}_g^{-1} (\mathbf{y}_i - \tilde{\mathbf{x}}_i^{(E)} \boldsymbol{\gamma}_g) \right). \end{aligned} \quad (6)$$

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 $\widehat{\mathbf{r}}_{ig} = \mathbf{y}_i - \tilde{\mathbf{x}}_i^{(E)} \widehat{\boldsymbol{\gamma}}_g$ which, crucially, satisfy $\sum_{i=1}^n \widehat{z}_{ig} \widehat{\mathbf{r}}_{ig} = 0$. Thus, maximising (6) is equivalent to minimising

$$\sum_{i=1}^n \sum_{g=1}^G \widehat{z}_{ig}^{(t+1)} \log |\boldsymbol{\Sigma}_g| + \sum_{i=1}^n \sum_{g=1}^G \widehat{z}_{ig}^{(t+1)} \widehat{\mathbf{r}}_{ig}^\top \boldsymbol{\Sigma}_g^{-1} \widehat{\mathbf{r}}_{ig}, \quad (7)$$

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 (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 $\widehat{\mathbf{R}}$:

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

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

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

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 $\widehat{\tau}_g(\mathbf{x}_i^{(G)})$ via MLR, ii) estimating the expert network parameters $\widehat{\gamma}_g$ and hence the component-specific means via weighted multivariate multiple linear regression, and iii) estimating the constrained component covariance matrices $\widehat{\Sigma}_g$ 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 2.1.1, their respective contribution to (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 (1). As described in Section 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 1, where gating covariates 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 $\mathbb{E}(\mathbf{Y} | \mathbf{X}^{(E)} = \mathbf{0}) = \mathbf{0}$ (Eisenhauer, 2003).

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 Σ_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 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 & Raftery \(1993\)](#). Such a noise component can be included regardless of where covariates (if any) enter, and regardless of the GPCM constraints employed. 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 & Coretto \(2008\)](#), especially if $n \leq p$. For univariate responses V is given by the range of y_1, \dots, 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 (8) or not affect (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) settings, respectively. The NGN setting 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 setting, relative to models without a noise component, though it is only defined for $G \geq 2$.

$$\text{GN: } f(\mathbf{y}_i | \mathbf{x}_i) = \sum_{g=1}^G \tau_g(\mathbf{x}_i^{(G)}) \phi\left(\mathbf{y}_i \mid \boldsymbol{\theta}_g(\mathbf{x}_i^{(E)}) = \left\{ \tilde{\mathbf{x}}_i^{(E)} \boldsymbol{\gamma}_g, \boldsymbol{\Sigma}_g \right\}\right) + \frac{\tau_0(\mathbf{x}_i^{(G)})}{V}. \quad (8)$$

$$\text{NGN: } f(\mathbf{y}_i | \mathbf{x}_i) = \sum_{g=1}^G \tau_g(\mathbf{x}_i^{(G)}) \phi\left(\mathbf{y}_i \mid \boldsymbol{\theta}_g(\mathbf{x}_i^{(E)}) = \left\{ \tilde{\mathbf{x}}_i^{(E)} \boldsymbol{\gamma}_g, \boldsymbol{\Sigma}_g \right\}\right) + \frac{\tau_0}{V}. \quad (9)$$

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 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 4.2). Novel strategies for dealing with both issues are proposed.

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, which is similar to that of Ning et al. (2008) but modified to account for the multivariate nature of the response. 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.

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 $(\mathbf{y}_i, \mathbf{x}_i^{(E)})$. When $\mathbf{x}_i^{(E)}$ includes categorical or ordinal covariates, the model-based approach to clustering mixed-type data of McParland & Gormley (2016) can be employed at this stage, though this is not considered further here.

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 $\Omega_1, \Omega_2, \dots, \Omega_G$.
 - 2 Estimate the expert network regression $\eta_g(\gamma_g, \cdot)$ on every subset $\{\Omega_g\}_{g=1}^G$.
 - 3 Compute the fitted values $\hat{\mathbf{y}}_{ig} = \eta_g(\hat{\gamma}_g, \mathbf{x}_i^{(E)}) \forall (i, g)$ and hence the residuals $\hat{\mathbf{r}}_{ig} = \mathbf{y}_i - \hat{\mathbf{y}}_{ig}$.
 - 4 Compute $\hat{\Psi}_g = \text{Cov}(\hat{\mathbf{R}}_g) = \frac{1}{n-d_E-1} \hat{\mathbf{R}}_g^\top \hat{\mathbf{R}}_g \forall g$.
 - 5 Compute the squared Mahalanobis distance $\widehat{M}_{ig} = d_M^2(\mathbf{y}_i, \hat{\mathbf{y}}_{ig}) = \hat{\mathbf{r}}_{ig}^\top \hat{\Psi}_g^{-1} \hat{\mathbf{r}}_{ig} \forall (i, g)$.
 - 6 Let $k_i = \arg \min_{h \in \{1, \dots, G\}} (\widehat{M}_{ih})$ and reassign observation i to subset Ω_{k_i} .
 - 7 Repeat Steps 2–6 until convergence is achieved, i.e. until the partition ceases to change.
-

If at any stage a level is dropped from a categorical variable in subset Ω_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 $\hat{\gamma}_g$ 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

$$\sum_{g=1}^G \min_{\{\eta_g, \gamma_g\}} \left(\sum_{i \in \Omega_g} d_M^2(\mathbf{y}_i, \eta_g(\gamma_g, \mathbf{x}_i^{(E)})) \right). \quad (10)$$

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 $\widehat{\Psi}_g^{-1}$ 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 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 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 (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.

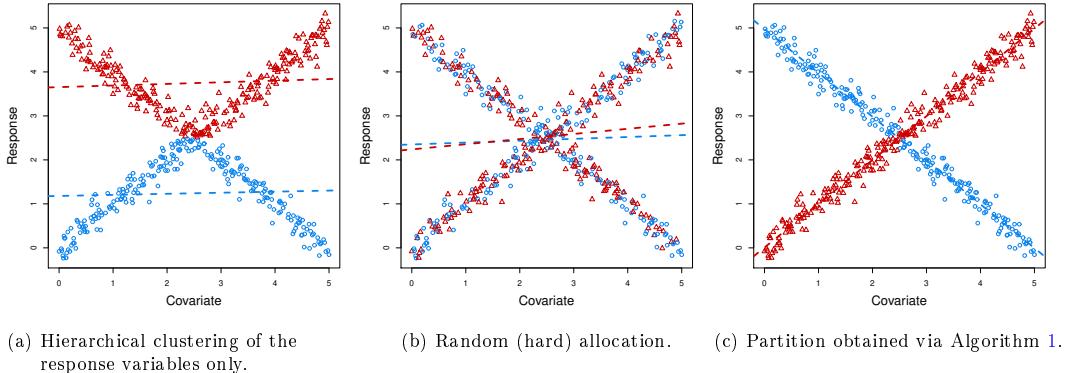


Figure 3: Initial 2-component hard partitions on univariate data clearly arising from a mixture of two linear regressions, obtained using (a) agglomerative hierarchical clustering, (b) random (hard) allocation, and (c) Algorithm 1 applied to the initialisation in (b) upon convergence after 6 iterations, demonstrating the improvement achieved by incorporating expert network covariates into the initialisation strategy. Allocations are distinguished by blue circles and red triangles. Corresponding fitted lines are also shown.

4.2 Model Selection

Whether a variable should be considered as a covariate or part of the response is usually clear from the context of the data being clustered. However, within the suite of MoE models outlined in Section 2.1.1, it is natural to question which 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 (and, in doing, affect the means of all p response variables) and/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, as is often the case in unsupervised clustering tasks.

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. Notably, avoiding the duplication of the EM initialisation routine for certain steps involving only the addition of a gating network covariate speeds up the algorithm somewhat. See Appendix A for an example of how to conduct such a stepwise search using code from the MoEClust R package (Murphy & Murphy, 2021).

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 models under the GN and NGN settings, given by (8) and (9) respectively, when evaluating every action involving models with gating concomitants.

When r is not so prohibitively large as to render an exhaustive search infeasible, Gormley & 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 2.1.1. 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.

5 Results

The clustering performance of the MoEClust models is illustrated by application to two well-known data sets: univariate CO₂ data (Section 5.1) and multivariate data from the Australian Institute of Sports (Section 5.2). Additional results are provided for each data set in the Appendices. In particular, code examples (Appendix A) and details of the initialisation (Appendix B) for the CO₂ data and results of the stepwise search (Appendix C) for the AIS data are given.

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 & Murphy, 2021).

5.1 CO₂ Data

As a univariate example of an application of MoEClust, data on the CO₂ emissions of $n = 28$ countries in the year 1996 (Hurn et al., 2003) are clustered, with Gaussian component densities. Studying the relationship between CO₂ 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 ($G \in \{1, \dots, 5\}$) are fitted, with either the equal (E) or unequal variance (V) models from Table 1. Quantile-based clustering of the CO₂ values is used to initialise Algorithm 1 when the expert network excludes GNP, otherwise agglomerative model-based hierarchical clustering of both CO₂ and GNP is used.

Table 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 A.

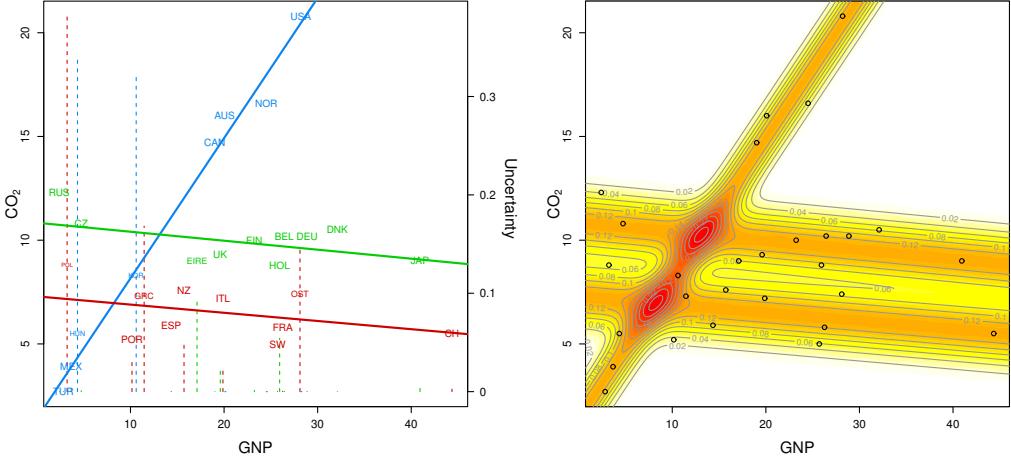
Table 2: The MoEClust BIC and ICL values of the top models under the six MoE special cases for the CO₂ data. Each row is optimal with respect to G and GPCM type, given the included covariates.

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 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 3. Its fit is exhibited in Figure 4, which shows that the relationship between CO₂ 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 $\widehat{U}_i = \min_{g \in \{1, \dots, \widehat{G}\}} (1 - \widehat{z}_{ig})$, are also shown.

Table 3: Estimated parameters of the optimal MoEClust model fit to the CO₂ 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_g^2	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 CO₂, accounting for the gating and expert networks, the latter of which includes GNP as a covariate. The densities are calculated by using predictions from the expert network as the means in evaluating $\sum_{g=1}^G \tau_g \phi(y_h^* | \tilde{\mathbf{x}}_h^* \tilde{y}_g, \hat{\sigma}_g^2)$ over an equally-spaced two-dimensional $H \times H$ grid $\{y_h^*, \mathbf{x}_h^*\}_{h=1}^H$ spanning the CO₂ and GNP variables' respective ranges.

Figure 4: Scatterplots of GNP against CO₂ 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, 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 using 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 CO₂ data, is provided in Appendix B.

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 & Weisberg, 1994). The thirteen variables recorded in the study are detailed in Table 4.

Table 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 mostly 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 concentration	Ht	height (cm)
		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 Table C.1 in Appendix C. The optimal model ($\text{BIC} = -4010.14$) 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 ($G = 2$, VVE, $\text{BIC} = -4563.12$), the best model with all covariates in the expert network only ($G = 1$, EEE, $\text{BIC} = -4234.79$), regardless of τ being constrained or not, and the best model with all covariates in the gating network only ($G = 2$, VEE, $\text{BIC} = -4092.77$), 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 extra 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 (i.e. those without gating network covariates) and 14 otherwise, 16 possible combinations of gating and/or expert network covariate settings, and consideration also being given to $G > 1$ models with equal mixing proportions, this still requires fitting 2,252 MoEClust models. However, 13 models either represented spurious solutions — 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 3.3) — or otherwise failed to converge, and were thus ultimately discarded. Table 5 gives the BIC and ICL values of a selection of the retained fitted models, representing the optimal models for certain special cases of interest.

Table 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, rows 3 and 7 roughly correspond to the optimal models according to `mclust`, with unequal and equal mixing proportions, respectively. For each model, its number of estimated parameters and its rank (according to BIC) among the full set of fitted models are also given.

Rank (BIC)	Gating	Expert	G	GPCM	BIC	ICL	No. Parameters
198		sex	2	VVV	-4113.31	-4121.32	42
890		sex	5	VVI	-4319.85	-4345.55	58
294			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
269	Equal		2	EVE	-4140.98	-4192.21	29
1	Equal	sex	2	EVE	-4010.14	-4057.87	39
26	BMI, sex		3	EEE	-4038.55	-4042.86	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 & 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), with 41 parameters, 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 Table C.2 in Appendix C). Consideration was also given to both the GN and NGN settings, 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 6). Both ‘SSF’ and ‘Ht’ enter the gating network, though not for the noise component, which has a constant mixing proportion ($\hat{\tau}_0 \approx 0.08$), as per the NGN setting in (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 5, adding a noise component to a model otherwise identical to the optimal model from Table 5 (BIC=-3992.81), and to models with a noise component but no stepwise selection of covariates (or no covariates at all).

Table 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 6), the means are summarised in Table 7 by the posterior mean of the fitted values of the model according to (11). The noise component is accounted for by $\bar{\mathcal{V}}$, the p -dimensional centroid of the region used to estimate V :

$$\hat{\boldsymbol{\mu}}_g = \frac{\sum_{i=1}^n \hat{z}_{ig} \hat{\mathbf{y}}_i}{\sum_{i=1}^n \hat{z}_{ig}} = \frac{\sum_{i=1}^n \hat{z}_{ig} \left(\sum_{g=1}^G \hat{z}_{ig} (\tilde{\mathbf{x}}_i^{(E)} \hat{\gamma}_g) + \hat{z}_{i0} \bar{\mathcal{V}} \right)}{\sum_{i=1}^n \hat{z}_{ig}}. \quad (11)$$

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 7 gives the means and average gates in terms of both components and clusters, as well as the common $\hat{\Sigma}$ matrix.

Table 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), 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 6, the means are summarised by the posterior mean of the model’s fitted values, given by (11).

	Cluster 1			Cluster 2			$\hat{\Sigma}$ (EEE)				
	All	Female	Male	All	Female	Male	RCC	WCC	Hc	Hg	Fe
$\hat{\tau}_g(\mathbf{x}_i)$	0.60	0.21	0.39	0.33	0.25	0.08					
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	15.41	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 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 model fit to the AIS data, the data and clustering results are shown using a generalised pairs plot (Emerson et al., 2013) in Figure 5, which depicts the pairwise relationships between

the hematological response variables and the included gating and expert network covariates, coloured according to the MAP classification. Different plot types are used as appropriate on the off-diagonals (e.g. scatterplots, boxplots, mosaic plots), depending on whether the given panel depicts two responses, two covariates, or one of each, and the nature of the variables.

In Figure 5, the marginal distributions of each variable are shown along the diagonal via histograms and barplots as appropriate, along with overlaid parametric marginal density estimates for the responses. For each component, the density estimates at each point $y^{(j)*}$ in an evenly-spaced grid spanning the range of the j -th response are obtained by averaging over univariate densities evaluated for each of the n observed sets of covariates $\mathbf{x}_1, \dots, \mathbf{x}_n$; i.e., by calculating

$$\frac{1}{n} \sum_{i=1}^n \hat{\tau}_g\left(\mathbf{x}_i^{(G)}\right) \phi\left(y^{(j)*} \mid \tilde{\mathbf{x}}_i^{(E)} \hat{\gamma}_{jg}, \hat{\sigma}_{jg}^2\right), \quad (12)$$

where $\hat{\gamma}_{jg}$ and $\hat{\sigma}_{jg}^2$ — with $\text{diag}(\hat{\Sigma}_g) = (\hat{\sigma}_{1g}^2, \dots, \hat{\sigma}_{pg}^2)$ — are the component-specific expert network coefficients and variance parameters of the j -th response, respectively. This expression simplifies greatly when covariates do not enter both parts of the model, particularly if they are absent from the expert network. In any case, the overall mixture density estimate is given by summing the G component-specific Gaussian densities from (12), and further adding $\frac{1}{nV} \sum_{i=1}^n \hat{\tau}_0(\mathbf{x}_i^{(G)})$ or simply $\hat{\tau}_0/V$, respectively, if the model also includes a GN or NGN noise component.

In panels of Figure 5 showing two responses, bivariate Gaussian ellipses with axes related to the within-cluster covariances are drawn. Owing to the presence of an expert network covariate in the fitted model, these ellipses are centred on the posterior mean of the fitted values, as described in (11). Their volume, shape, and orientation are also modified for the same reason: they are derived by adding the extra variability in the fitted values (weighted by $\hat{\mathbf{Z}}$) to $\hat{\Sigma}_g$. Thus, the depicted ellipses do not conform to the EEE covariance constraints of the optimal model.

It is clear from Figure 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 (Hubert & Arabie, 1985) of just 0.11. This index 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 5). 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.

Indeed, Table 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 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).

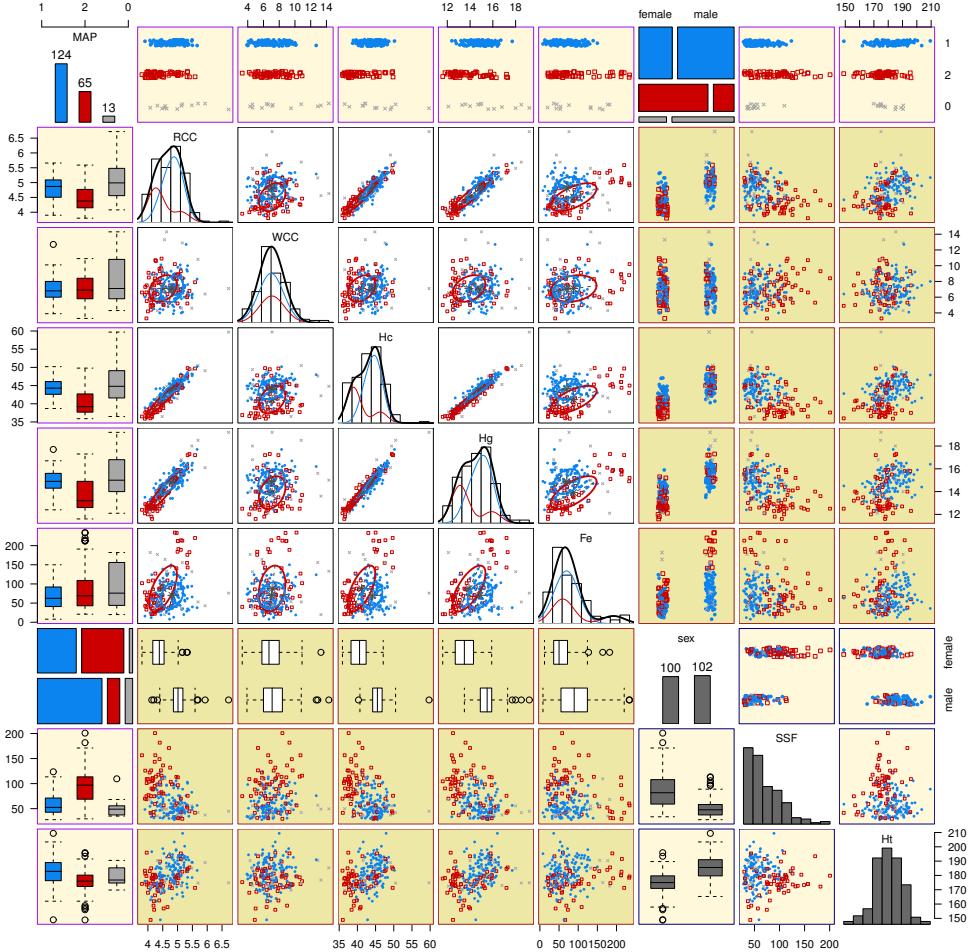


Figure 5: Generalised pairs plot for the optimal ‘full’ MoEClust model fit to the AIS data, depicting pairwise relationships between the hematological response variables, the expert network covariate sex, the gating concomitants SSF and Ht, and the MAP classification. Colours and plotting symbols relate 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, scatterplots are used for panels involving two continuous variables, a mix of boxplots and jittered strip plots are used for mixed pairs, and marginal distributions are shown along the diagonal via histograms (with overlaid density estimates) and barplots as appropriate. The axis labels relate to the *range* of the depicted variable always, and thus the *y*-axis labels correspond neither to counts nor densities in the plots along the diagonal.

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 & Leisch, 2007, 2008), biology (Wang et al., 1996), econometrics (Wang et al., 1998), marketing (Wedel & 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 the 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 & 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. 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. To this end, a novel greedy forward stepwise search algorithm has been employed for model/variable selection purposes. This strategy has the added advantages of introducing additional parsimony, by potentially reducing the number of regression parameters in the gating and/or expert networks, and speeding up the EM algorithm itself. However, 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 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 (G_p) 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 gating network concomitant, 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 & 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. By shrinking some coefficient estimates to zero, this sparsity could help with relaxing the assumptions that covariates affect all components and, in the case of the expert network, the assumption that covariates affect the means of all p dependent variables. Regularisation in another, Bayesian sense, by specifying a prior on the component variances/covariances in the spirit of Fraley & Raftery (2007), and/or component regression parameters, could also prove useful for avoiding spurious solutions due to computational singularities, as described in Section 3.3. MoEClust models could also be developed in the context of hierarchical mixtures of experts (Jordan & 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 & Murphy, 2008, 2010). These could be incorporated into Gaussian mixture of experts models using residuals in an equivalent fashion to Section 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 t EIGEN family of covariance constraints (Andrews & 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.

Acknowledgements

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.

7 References

- J. L. Andrews & P. D. McNicholas. Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t -distributions: the t EIGEN family. *Statistics and Computing*, 22(5): 1021–1029, 2012. [6](#), [25](#)
- J. Banfield & A. E. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49(3): 803–821, 1993. [2](#), [6](#), [12](#)
- T. Benaglia, D. Chauveau, D. R. Hunter, & D. Young. mixtools: an R package for analyzing finite mixture models. *Journal of Statistical Software*, 32(6): 1–29, 2009. [7](#), [8](#)
- C. Biernacki, G. Celeux, & G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7): 719–725, 2000. [15](#)

- C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, NY, USA, 2006. [3](#)
- D. Böhning, E. Dietz, R. Schaub, P. Schlattmann, & B. G. Lindsay. The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family. *Annals of the Institute of Statistical Mathematics*, 46(2): 373–388, 1994. [9](#)
- G. Celeux & G. Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28 (5): 781–793, 1995. [2](#), [5](#), [6](#)
- R. D. Cook & S. Weisberg. *An Introduction to Regression Graphics*, volume 405 of *Wiley Series in Probability and Statistics*. John Wiley & Sons, New York, NY, USA, 1994. [18](#)
- U. J. Dang & P. D. McNicholas. Families of parsimonious finite mixtures of regression models. In I. Morlini, T. Minerva, & M. Vichi, editors, *Advances in Statistical Models for Data Analysis: the 9th biannual meeting of the Classification and Data Analysis Group (CLADAG) of the Italian Statistical Society*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 73–84, September 18–20, 2013, University of Modena and Reggio Emilia, Italy, 2015. Cham, Switzerland: Springer. [2](#), [8](#), [20](#), [24](#)
- U. J Dang, A. Punzo, P. D. McNicholas, S. Ingrassia, & R. P. Browne. Multivariate response and parsimony for Gaussian cluster-weighted models. *Journal of Classification*, 34(1): 4–34, 2017. [2](#), [8](#)
- C. M. Dayton & G. B. Macready. Concomitant-variable latent-class models. *Journal of the American Statistical Association*, 83(401): 173–178, 1988. [2](#), [3](#)
- A. P. Dempster, N. M. Laird, & D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39(1): 1–38, 1977. [8](#), [10](#)
- J. G. Eisenhauer. Regression through the origin. *Teaching Statistics*, 25(3): 76–80, 2003. [11](#)
- J. W. Emerson, W. A. Green, B. Schloerke, J. Crowley, D. Cook, H. Hofmann, & H. Wickham. The generalized pairs plot. *Journal of Computational and Graphical Statistics*, 22(1): 79–91, 2013. [21](#)
- C. Fraley & A. E. Raftery. Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of Classification*, 24(2): 155–181, 2007. [25](#)
- L. A. García-Escudero, A. Gordaliza, F. Greselin, S. Ingrassia, & A. Mayo-Iscar. Eigenvalues and constraints in mixture modeling: geometric and computational issues. *Advances in Data Analysis and Classification*, 12(2): 203–233, 2018. [12](#)
- J. Geweke & M. Keane. Smoothly mixing regressions. *Journal of Econometrics*, 138(1): 252–290, 2007. [4](#)
- I. C. Gormley & T. B. Murphy. Clustering ranked preference data using sociodemographic covariates. In S. Hess & A. Daly, editors, *Choice Modelling: The State-of-the-art and The State-of-practice – Proceedings from the Inaugural International Choice Modelling Conference*, pages 543–569, March 30–April 1, 2009, University of Leeds, Harrogate, UK, 2010. Bingley, UK: Emerald Group Publishing Limited. [16](#)
- I. C. Gormley & T. B. Murphy. Mixture of experts modelling with social science applications. In K. Mengeszen, C. P. Robert, & D. M. Titterington, editors, *Mixtures: Estimation and Applications*, volume 887 of *Wiley Series in Probability and Statistics*, chapter 9, pages 101–121. John Wiley & Sons, New York, NY, USA, 2011. [4](#)
- B. Grün & F. Leisch. Fitting finite mixtures of generalized linear regressions in R. *Computational Statistics & Data Analysis*, 51(11): 5247–5252, 2007. [7](#), [24](#)

- B. Grün & F. Leisch. FlexMix version 2: finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, 28(4): 1–35, 2008. [7](#), [24](#)
- C. Hennig. Identifiability of models for clusterwise linear regression. *Journal of Classification*, 17(2): 273–296, 2000. [2](#)
- C. Hennig & P. Coretto. The noise component in model-based cluster analysis. In C. Preisach, H. Burkhardt, L. Schmidt-Thieme, & R. Decker, editors, *Data Analysis, Machine Learning and Applications: Proceedings of the 31st Annual Conference of the Gesellschaft für Klassifikation e.V.*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 127–138, March 7–9, 2007, Albert-Ludwigs-Universität Freiburg, Germany, 2008. Berlin, Germany: Springer. [12](#)
- L. Hubert & P. Arabie. Comparing partitions. *Journal of Classification*, 2(1): 193–218, 1985. [22](#)
- M. Hurn, A. Justel, & C. P. Robert. Estimating mixtures of regressions. *Journal of Computational and Graphical Statistics*, 12(1): 55–79, 2003. [17](#)
- S. Ingrassia, S. C. Minotti, & G. Vittadini. Local statistical modeling via the cluster-weighted approach with elliptical distributions. *Journal of Classification*, 29(3): 363–401, 2012. [1](#), [5](#)
- S. Ingrassia, A. Punzo, G. Vittadini, & S. C. Minotti. The generalized linear mixed cluster-weighted model. *Journal of Classification*, 32(1): 85–113, 2015. [2](#)
- R. A. Jacobs, M. I. Jordan, S. J. Nowlan, & G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1): 79–87, 1991. [2](#), [3](#), [4](#)
- M. I. Jordan & R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2): 181–214, 1994. [25](#)
- A. E. Lamont, J. K. Vermunt, & M. L. Van Horn. Regression mixture models: does modeling the covariance between independent variables and latent classes improve the results? *Multivariate Behavioural Research*, 51(1): 35–52, 2016. [1](#)
- R. Lebret, S. Iovleff, F. Langrognet, C. Biernacki, G. Celeux, & G. Govaert. Rmixmod: the R package of the model-based unsupervised, supervised, and semi-supervised classification mixmod library. *Journal of Statistical Software*, 67(6): 1–29, 2015. [7](#), [8](#)
- P. C. Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences, India*, 2(1): 49–55, 1936. [13](#)
- A. Mazza, A. Punzo, & S. Ingrassia. flexCWM: a flexible framework for cluster-weighted models. *Journal of Statistical Software*, 86: 1–27, 2018. [8](#)
- P. McCullagh & J. Nelder. *Generalized Linear Models*, volume 37 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, London, UK, 1983. [4](#)
- P. D. McNicholas & T. B. Murphy. Parsimonious Gaussian mixture models. *Statistics and Computing*, 18(3): 285–296, 2008. [6](#), [25](#)
- P. D. McNicholas & T. B. Murphy. Model-based clustering of microarray expression data via latent Gaussian mixture models. *Bioinformatics*, 26(21): 2705–2712, 2010. [6](#), [25](#)
- D. McParland & I. C. Gormley. Model based clustering for mixed data: clustMD. *Advances in Data Analysis and Classification*, 10(2): 155–169, 2016. [13](#)
- E. H. Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society*, 26(9): 394–395, 1920. [14](#)

- K. Murphy & T. B. Murphy. *MoEClust: Gaussian parsimonious clustering models with covariates and a noise component*, 2021. URL <https://cran.r-project.org/package=MoEClust>. R package version 1.4.0. [2](#), [16](#), [29](#)
- H. Ning, Y. Hu, & T. S. Huang. Efficient initialization of mixtures of experts for human pose estimation. In *Proceedings of the International Conference on Image Processing (ICIP 2008)*, pages 2164–2167, October 12–15, 2008, San Diego, CA, USA, 2008. New York, NY, USA: Institute for Electrical and Electronics Engineers (IEEE). [13](#)
- A. Punzo & S. Ingrassia. Parsimonious generalized linear Gaussian cluster-weighted models. In I. Morlini, T. Minerva, & M. Vichi, editors, *Advances in Statistical Models for Data Analysis: the 9th biannual meeting of the Classification and Data Analysis Group (CLADAG) of the Italian Statistical Society*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 201–209, September 18–20, 2013, University of Modena and Reggio Emilia, Italy, 2015. Cham, Switzerland: Springer. [8](#)
- A. Punzo & S. Ingrassia. Clustering bivariate mixed-type data via the cluster-weighted model. *Computational Statistics*, 31(3): 989–103, 2016. [2](#)
- A. Punzo & P. D. McNicholas. Parsimonious mixtures of multivariate contaminated normal distributions. *Biometrical Journal*, 58(6): 1506–1537, 2016. [6](#)
- R Core Team. *R: a language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021. [2](#), [7](#)
- G. E. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2): 461–464, 1978. [15](#)
- L. Scrucca, M. Fop, T. B. Murphy, & A. E. Raftery. mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1): 289–317, 2016. [2](#), [6](#), [7](#), [8](#), [11](#)
- T. J. Thompson, P. J. Smith, & J. P. Boyle. 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, 1998. [24](#)
- P. Wang, M. L. Puterman, I. Cockburn, & N. Le. Mixed Poisson regression models with covariate dependent rates. *Biometrics*, 52(2): 381–400, 1996. [24](#)
- P. Wang, M. L. Puterman, & I. Cockburn. Analysis of patent data – a mixed-Poisson regression-model approach. *Journal of Business & Economic Statistics*, 16(1): 27–41, 1998. [24](#)
- M. Wedel & W. A. Kamakura. *Market Segmentation: Conceptual and Methodological Foundations*, volume 8 of *International Series in Quantitative Marketing*. Springer, New York, NY, US, 2012. [24](#)
- L. Xu, M. Jordan, & G. E. Hinton. An alternative model for mixtures of experts. In G. Tesauro, D. Touretzky, & T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 633–640, Cambridge, MA, USA, 1994. MIT Press. [4](#)
- D. S. Young & D. R. Hunter. Mixtures of regressions with predictor-dependent mixing proportions. *Computational Statistics & Data Analysis*, 54(10): 2253–2266, 2010. [25](#)
- A. Zellner. An efficient method of estimating seemingly unrelated regression equations and tests for aggregation bias. *Journal of the American Statistical Association*, 57(298): 348–368, 1962. [10](#)

Appendices

Appendix A CO₂ Data: Code Examples

Code to reproduce both the exhaustive (Listing A.1) and greedy forward stepwise (Listing A.2) searches for the CO₂ data described in Section 5.1, using the MoEClust R package (Murphy & Murphy, 2021), is provided below. The code in Listing A.1 can be used to reproduce the results in Table 2.

Listing A.1: Exhaustive search R code for the CO₂ 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, equalPro=TRUE, expert= ~ GNP)

# 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 A.2: Stepwise search R code for the CO₂ 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)
```

Appendix B CO₂ 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 CO₂ data, with and without the initial partition obtained by model-based agglomerative hierarchical clustering being passed through Algorithm 1, are shown in Figure B.1. A BIC value of -155.20 is achieved after 21 EM iterations (starting, after 6 iterations of our proposed initialisation strategy, from a log-likelihood of -66.01) compared to a value of -161.06 after 28 EM iterations without Algorithm 1 (starting from a log-likelihood of -76.39). While the models differ only in terms of the initialisation strategy employed, Table 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 B.1a has one cluster with a steep slope and two clusters with near-zero slopes but different intercepts. Notably, supplying 100 random starts to Algorithm 1 did not yield an improved BIC in any instance. Similarly, the optimal BIC value obtained by supplying 100 random starts to the EM algorithm directly was also greater than -161.06 but still less than -155.20 .

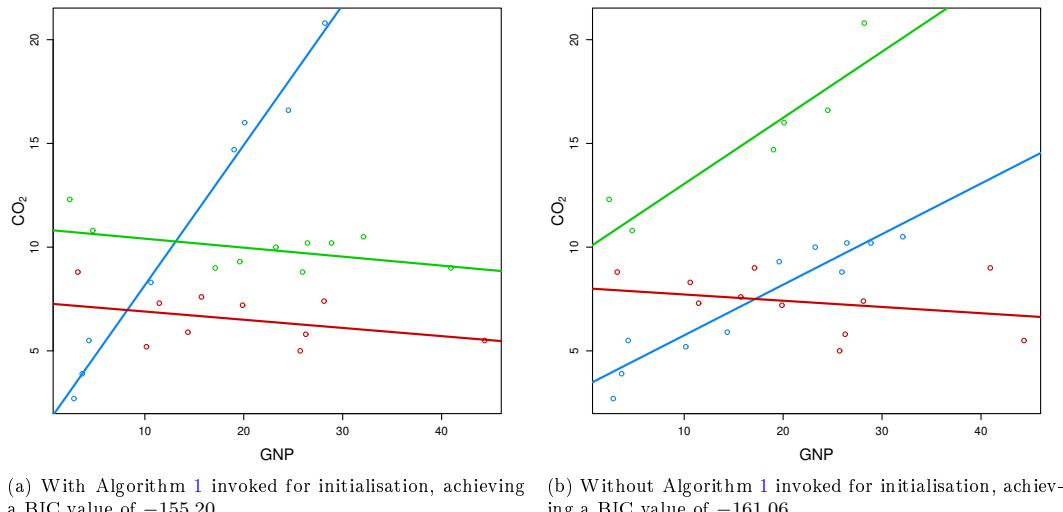


Figure B.1: Scatterplots of GNP against CO₂ emissions for $n = 28$ countries showing $G = 3$ coloured linear regression components from MoEClust models with equal variances and mixing proportions, with (a) and without (b) the initialisation strategy described in Algorithm 1 invoked.

Appendix C AIS Data: Stepwise Model Search

For the AIS data, Table 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 5). Note, however, that the remaining covariates in Table 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 C.1 with $G = 2$ EVE Gaussian components, and the model found by the second stepwise search described in Table C.2 with three, of which two are EEE Gaussian and one is uniform. Ultimately, the model with the noise component identified in Table 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 (9).

Table 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 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 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 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 setting in (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

Regularized Joint Mixture Models

Konstantinos Perrakis

*Department of Mathematical Sciences
Durham University, UK*

KONSTANTINOS.PERRAKIS@DURHAM.AC.UK

Thomas Lartigue

*Aramis Project Team, Inria &
Center of Applied Mathematics, CNRS, École Polytechnique, IP Paris, France*

THOMAS.LARTIGUE@DZNE.DE

Frank Dondelinger

*Lancaster Medical School
Lancaster UK*

FDONDELINGER.WORK@GMAIL.COM

Sach Mukherjee

*German Center for Neurodegenerative Diseases, Bonn, Germany
& MRC Biostatistics Unit, University of Cambridge, UK*

SACH.MUKHERJEE@DZNE.DE

Editor: Samuel Kaski

Abstract

Regularized regression models are well studied and, under appropriate conditions, offer fast and statistically interpretable results. However, large data in many applications are heterogeneous in the sense of harboring distributional differences between latent groups. Then, the assumption that the conditional distribution of response Y given features X is the same for all samples may not hold. Furthermore, in scientific applications, the covariance structure of the features may contain important signals and its learning is also affected by latent group structure. We propose a class of mixture models for paired data (X, Y) that couples together the distribution of X (using sparse graphical models) and the conditional $Y | X$ (using sparse regression models). The regression and graphical models are specific to the latent groups and model parameters are estimated jointly. This allows signals in either or both of the feature distribution and regression model to inform learning of latent structure and provides automatic control of confounding by such structure. Estimation is handled via an expectation-maximization algorithm, whose convergence is established theoretically. We illustrate the key ideas via empirical examples. An R package is available at <https://github.com/k-perrakis/regjmix>.

Keywords: distribution shifts, heterogeneous data, joint learning, latent groups, mixture models, sparse regression

1. Introduction

Regularized regression models usually assume homogeneity in the sense that the same conditional distribution of a response Y given features X is taken to hold for all samples. In the presence of latent groups that might have different underlying conditional distributions, regression modeling may be confounded, possibly severely. Similarly, covariance structure among features can be an important signal in scientific applications but its learning may be strongly affected by latent group structure.

©2023 Konstantinos Perrakis, Thomas Lartigue, Frank Dondelinger and Sach Mukherjee.

License: CC-BY 4.0, see <https://creativecommons.org/licenses/by/4.0/>. Attribution requirements are provided at <http://jmlr.org/papers/v24/21-0796.html>.

These issues are a concern whenever data might harbour unrecognized distributional shifts or group structure, an issue that is increasingly prominent in an era of large and often heterogeneous data. Furthermore, for heterogeneous data the two aspects – the distribution of features X and the conditional $Y | X$ – are related in practice, since either or both may contain signals relevant to detecting and modeling group structure, which in turn is essential to overall estimation. Motivated by such heterogeneous data settings with paired data of the form (X, Y) , in this paper we study a class of joint mixture models that couple together both aspects – sparse graphical models for X and parsimonious regression models for $Y | X$ – in one framework. Specifically, in high-level notation, we consider models of the form

$$\begin{aligned} Z &\sim \tau_Z \\ X | Z=k &\sim p_X(\mu_k, \Sigma_k) \\ Y | X, Z=k &\sim p_Y(g(X^T \beta_k), \sigma_k^2) \end{aligned} \tag{1}$$

where $Z \in \{1, \dots, K\}$ is a latent indicator of group membership with distribution τ_Z , $p_W(m, s)$ denotes the probability distribution of a random variable W with location m and scale s , and $g(\cdot)$ is a link function. This work focuses on the familiar and important case where both X and Y are normally distributed and g is the identity function. However, the key ideas apply to any model of the general form in Eq. (1).

In this context the presence of group structure has the following consequences:

- *Confounding due to latent groups.* Associations between components of X and Y may be entirely different “globally” (with Z marginalized out) vs. “locally” (conditionally on Z) with regression coefficients differing even in signs and sparsity patterns.
- *Ambiguous group structure in feature space.* Clustering the X ’s alone may lead to cluster labels which do not capture the relevant structure, as instances of group signal in X may be unrelated to Y (e.g. clustering gene expression data may yield well-defined clusters; however, these may not relate to a specific biological/medical response).
- *Group-specific signal in regression coefficients.* Nonidentical coefficients or feature importance across groups provide a potentially useful discriminant signal for identifying the group structure itself. This signal cannot be detected by clustering the X ’s.

Two common strategies given paired data (X, Y) are: (S1) ignore any potential grouping and fit one regression model using the entire data and (S2) cluster the X ’s and then fit separate regression models to the group-specific data. Strategy (S1) is risky, since the resulting regression coefficients may be entirely incorrect if latent group structure is present (e.g. due to Simpson’s paradox and related phenomena). Also, when the modeling aspect is of importance, under (S1) evaluation of predictive loss is not a satisfactory guide for model assessment, since prediction error may be apparently small despite severe model misspecification. Strategy (S2), although in some ways safer, is also not guaranteed to protect from such effects, unless the resulting group structure obtained from clustering the X ’s is correct with reference to the overall problem; this may not hold in general. Furthermore, since (S2) models the X data alone, it cannot exploit any signal in the conditionals $Y | X$ to guide the clustering. A common variant of (S2) is to perform a dimension reduction on X , such as PCA, and cluster on the reduced space. This, however, does not resolve the problem as the major principal components may not be predictive of Y ; see e.g. Jolliffe (1982).

1.1 Related work and contribution

For discrete latent variables Z a standard way of approaching such heterogeneous problems is via mixture models. The literature on mixtures is vast; below we summarize related work according to model structure, covering the most popular approaches for the case of continuous responses.

Mixtures for $X|Z$. The most commonly used and extensively studied approach within this category is the Gaussian mixture model (GMM); see e.g., McLachlan and Peel (2000). GMMs have undergone a series of novel developments focusing on parsimonious modeling of the covariance matrices such as parametrizations based on eigenvalue decomposition (Banfield and Raftery, 1993; Celeux and Govaert, 1995; Fraley and Raftery, 2002), factorizations based on factor analysis models (McNicholas and Murphy, 2008), and extensions to sparse graphical model estimation (Anandkumar et al., 2012; Städler et al., 2017; Fop et al., 2019), among others. These approaches differ from ours in that they consider only the X signal and do not include a regression component, thus, inheriting the potential drawbacks of strategy (S2).

Mixtures for $Y|X, Z$. Finite mixtures of regression (FMR) models belong in this category. Similarly to GMMs, Gaussian FMRs have been studied and developed extensively, allowing for flexible modeling designs (see e.g. Früwirth-Schnatter, 2005) and regularized estimation (Khalili and Jiahua, 2007; Städler et al., 2010; Khalili and Lin, 2013). FMRs focus on the relationship between Y and X without including a generative probability model for X . Our approach is motivated by settings in which the X distribution itself is of interest and may be confounded by latent group structure. Furthermore, under FMRs a *new* X' cannot be allocated to *one specific* group and thereby used to obtain a group-specific prediction.

Mixtures for $Y, Z|X$. Mixtures of experts (MoE; Dayton and Macready, 1988; Jacobs et al., 1991; Jordan and Jacobs, 1994; Jacobs, 1997) jointly model the response and latent allocations. MoEs consist of expert networks (these are models that predict Y from X) and a discriminative model (the gating network) that chooses among the experts. The parsimonious covariance parametrizations for GMMs (Banfield and Raftery, 1993; Celeux and Govaert, 1995; Fraley and Raftery, 2002) have been introduced within the MoE framework initially in Dang and McNicholas (2015) for the special case where the same set of predictors enter the expert and gating networks, and, more recently, in Murphy and Murphy (2020) for the general case where different predictors are allowed to enter in the two networks; the latter work also introduces an additional noise component for outlier detection. Regularized MoE approaches include those of Khalili (2010) and Chamroukhi and Huynh (2018), among others. MoEs include FMRs as a special case in the absence of a gating network. Also, similarly to FMRs, MoEs condition upon features and, thus, lack a generative model for X . However, unlike with FMRs, group-specific prediction of the response is possible under MoEs, as the learned gating network can be used to allocate new feature observations.

Mixtures for $Y, X|Z$. A first approach within this category is profile regression (Molitor et al., 2010; Liverani et al., 2015). Under profile regression X and Y are conditionally independent given the latent group indicator Z . Specifically, the component $Y|Z$ involves a regression model including a “profile” parameter (capturing the effect of X) plus additional co-variates, while the component $X|Z$ is some multivariate distribution (e.g. Gaussian). A second approach, more relevant to our work, is the cluster weighted model (CWM) mixture

introduced by Ingrassia et al. (2012). In this case, we have a linear model component for $Y|X, Z$ and a multivariate distribution component for $X|Z$, following the hierarchical structure of Eq. (1). As illustrated, in Ingrassia et al. (2012) Gaussian CWMs lead to the same family of probability distributions generated by GMMs (see also below) and under specific conditions include FMRs and MoEs as special cases. Extensions of CWMs accommodate the use of GLMs and mixed-type data (Ingrassia et al., 2015; Punzo and Ingassia, 2016), parsimonious parametrizations (similarly to GMMs and MoEs) of covariances (Dang et al., 2017) and latent factor structures for the feature matrix (Subedi et al., 2013), among others.

The class of models proposed here – henceforth, referred to as *regularized joint mixture* (RJM) models – belong to the latter category of mixtures. Specifically, the model specification (the likelihood part of the model) is of a CWM type, but the resulting clustering and parameter learning process under RJMs is different due to regularization. CWMs rely on maximum-likelihood (ML) estimation and in this case under the normal-normal setting with identity link (as considered here), Eq. (1) is equivalent to a GMM on the concatenated matrix $[X, Y]$ as shown in Ingrassia et al. (2012). However, under RJMs, the equivalence to the GMM no longer holds, because the regression and graphical model parts are treated differently (below we include comparisons with direct Gaussian mixture modeling of the concatenated matrix $[X, Y]$). We view regularization as essential for delivering a usable solution to the problem. In many cases the number of features p may be on the same order as sample size n , or larger, and at the group level the sample sizes are of course smaller; hence without suitable regularization both the regression and graphical models will typically be ill-behaved. In the specific implementation we propose, we use the graphical lasso (Friedman et al., 2008a) for graphical model estimation, while for the regression part we consider: (i) the Bayesian lasso (Park and Casella, 2008) and (ii) the normal-Jeffreys prior (Figueiredo, 2001). We note that other choices would be possible within the general framework, subject to computational considerations and appropriate handling of tuning parameters. In summary, the merits of RJMs are the following:

- (i) Learns latent group structure by combining information from the distribution over X and the regression of Y on X within a principled framework;
- (ii) Provides group-specific feature importance and graphical models with explicit sparsity patterns;
- (iii) Applicable in $p > n$ settings;
- (iv) Allows group-specific prediction for the response given a new feature vector X' .

1.2 Motivation

Given the large literature on mixture models, it is important to clarify at the outset why the models we study are needed. We are motivated by applications in which latent group structure may be important and where aspects such as (potentially group specific) feature importance and covariance structure among the X 's play a role.

To take one example, in biomedicine there is much interest in latent disease subtypes. These will often have subtype-specific covariance patterns, due to differences in underlying

regulatory networks, and the analyst may want to understand subtype-specific disease biology and feature importance. Focusing on only the X 's may be insufficient, because this does not account for the response Y (and there may be many ways of clustering X not relevant to the Y of interest). For example, if X are data on human subjects and Y a cancer phenotype, many instances of cluster structure in X may be unrelated to the cancer setting. In addition, focusing solely on differences in regression models $Y|X$ means that subgroup recovery is difficult or impossible if these differences are not large enough. Similarly, in data-driven marketing, latent customer subgroups may have different covariance structure among features and at the same time manifest differences in regression models linking such features to responses (such as revenue per customer). The formulation we propose includes both sources of information in one model and thereby allows for subgroup identification and parameter estimation that accounts for both aspects.

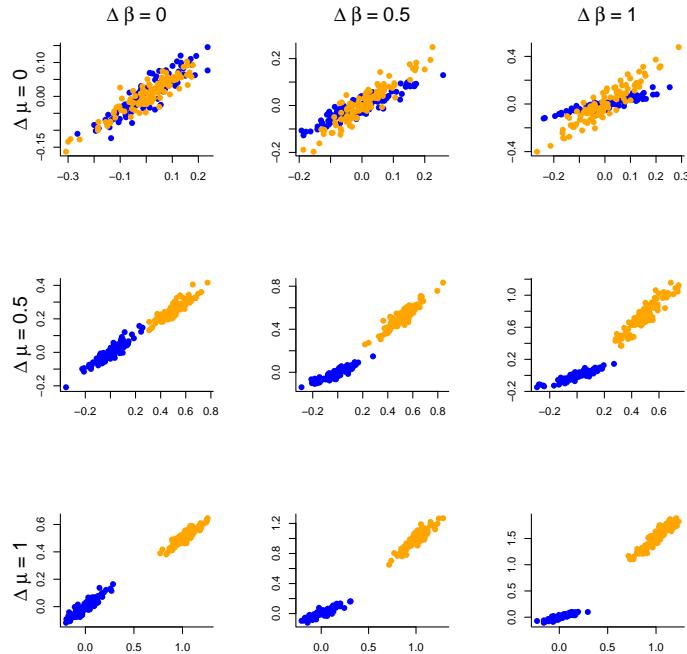


Figure 1: Examples for two subgroups. Each panel shows a specific level of difference in the regression models, quantified by the difference $\Delta\beta$ in regression coefficients. The difference in the feature distributions is controlled via a simple mean shift $\Delta\mu$.

Figure 1 shows simple illustrative examples to bring out some of these points. We emphasize that this is for illustrative purposes only, intended to highlight some interesting contrasts (full empirical results appear in Section 5 below). In these examples, we consider settings in which there may be either or both of an X signal difference (in the figure this is a simple mean shift $\Delta\mu$, but the models we propose are general and for multivariate X allow

also for differences in covariance structure) and a difference in subgroup-specific regression coefficients ($\Delta\beta$). For this initial illustration we consider two latent groups, each with sample size equal to 100, and ten potential predictors, but with only one predictor having an effect (a non-zero regression coefficient) on the response; full details of the simulations can be found in Appendix A. Results in terms of subgroup identification, as quantified by Rand Index, are summarized in Figure 2. When there is no difference in regression models ($\Delta\beta = 0$), MoEs cannot detect any structure (since the X distribution is not modelled). On the other hand, with a stronger difference in regression models ($\Delta\beta = 1$), MoE outperforms a Gaussian mixture (on the X 's), since the latter does not model the regression part. The approach we propose models both aspects in a unified framework, hence works well regardless of where the signal lies. Furthermore, and as shown in detail via empirical examples below, by accounting for the latent structure, RJM is able to detect subgroup-specific sparsity patterns whilst avoiding Simpson's paradox-like effects that could otherwise arise. Later, we show detailed empirical results, including an example, based on cancer data, that highlights some of these points, and in particular how subgroup identification benefits from joint modeling, relative to simply clustering X (or clustering the stacked vector (X, Y)) or using MoE.

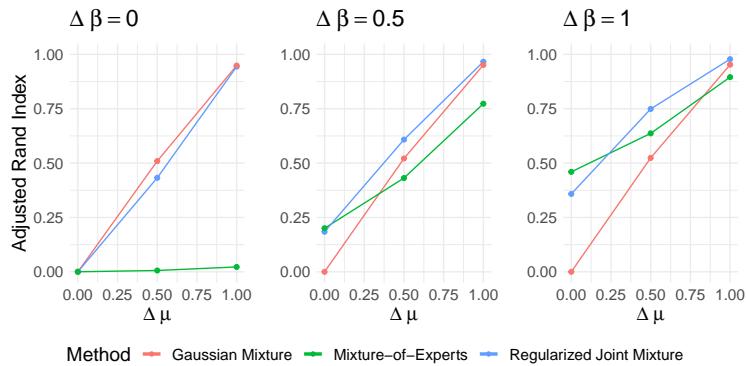


Figure 2: Role of signal location in subgroup identification. The difference $\Delta\beta$ in regression coefficients represents the signal from the regression models, from no signal (left panel), to a strong signal (right panel). In each panel the signal in the feature distribution increases from left to right via a simple mean shift $\Delta\mu$.

The remainder of the paper is structured as follows. In Section 2 we lay out the model specification and discuss the regularization methods under consideration as well as efficient tuning strategies. Computational and theoretical details of the expectation-maximization (EM) optimization are covered in Section 3. In Section 4 we discuss prediction using RJMs and discuss how predictive measures can potentially be used for cluster selection. In Section 5 we present empirical examples, focusing initially on small-scale simulations and then proceeding to larger scale semi-synthetic experiments and applications to real data. The paper concludes with a discussion in Section 6.

2. The RJM model

2.1 Model specification

Let \mathbf{y} denote an n -dimensional vector of outputs or responses and \mathbf{X} an $n \times p$ feature matrix. Samples are indexed by $i = 1, \dots, n$. Let K denote the number of groups and $z_i \in \{1 \dots K\}$ represent the true (latent) group indicator for the sample point (y_i, \mathbf{x}_i) with $\Pr(z_i = k) = \tau_k$. The group-specific parameters are $\boldsymbol{\theta}_k = (\boldsymbol{\theta}_k^X, \boldsymbol{\theta}_k^Y)^T$ with $\boldsymbol{\theta}_k^X$ and $\boldsymbol{\theta}_k^Y$ being the parameters governing respectively the marginal distribution of X and the regression model of Y on X .

We allow for group-specific parameters, but assume that samples are independent and identically distributed within groups. The joint distribution of (y_i, \mathbf{x}_i) in group k is

$$p(y_i, \mathbf{x}_i | \boldsymbol{\theta}_k, z_i = k) = p(y_i | \boldsymbol{\theta}_k^Y, \mathbf{x}_i, z_i = k)p(\mathbf{x}_i | \boldsymbol{\theta}_k^X, z_i = k). \quad (2)$$

The features are modeled as p -dimensional multivariate normal so that $\boldsymbol{\theta}_k^X = (\boldsymbol{\mu}_k, \text{vec}(\boldsymbol{\Sigma}_k))^T$, where $\boldsymbol{\mu}_k$ is the mean and $\boldsymbol{\Sigma}_k$ the $p \times p$ covariance matrix. For the responses, we specify a normal linear regression model, with parameters $\boldsymbol{\theta}_k^Y = (\alpha_k, \boldsymbol{\beta}_k, \sigma_k^2)^T$, where α_k is the intercept, $\boldsymbol{\beta}_k$ the vector of regression coefficients and σ_k^2 the error variance. Inclusion of the intercept is necessary in the present setting, because it is not possible to center the response appropriately when the group labels are unknown. Thus, we have that

$$p(\mathbf{x}_i | \boldsymbol{\theta}_k^X, z_i = k) \equiv p(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, z_i = k) = N_p(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (3)$$

and

$$p(y_i | \boldsymbol{\theta}_k^Y, \mathbf{x}_i, z_i = k) \equiv p(y_i | \alpha_k, \boldsymbol{\beta}_k, \sigma_k^2, \mathbf{x}_i, z_i = k) = N(y_i | \alpha_k + \mathbf{x}_i^T \boldsymbol{\beta}_k, \sigma_k^2). \quad (4)$$

Marginalizing out the latent variables leads to a mixture representation of the form

$$p(\mathbf{y}, \mathbf{X} | \boldsymbol{\theta}, \boldsymbol{\tau}) = \prod_{i=1}^n \sum_{k=1}^K p(y_i | \boldsymbol{\theta}_k^Y, \mathbf{x}_i, z_i = k)p(\mathbf{x}_i | \boldsymbol{\theta}_k^X, z_i = k)\tau_k, \quad (5)$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K)^T$ and $\boldsymbol{\tau} = (\tau_1, \dots, \tau_K)^T$.

2.2 Regularization and priors

Given the likelihood function of $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$ in (5) we consider general solutions of the form

$$\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\tau}} = \arg \max_{\boldsymbol{\theta}, \boldsymbol{\tau}} \left\{ \log p(\mathbf{y}, \mathbf{X} | \boldsymbol{\theta}, \boldsymbol{\tau}) + \sum_{k=1}^K \text{pen}(\boldsymbol{\theta}_k^{*X}) + \sum_{k=1}^K \text{pen}(\boldsymbol{\theta}_k^{*Y}) \right\}, \quad (6)$$

where $\text{pen}(\cdot)$ denotes a penalty function, and $\boldsymbol{\theta}_k^{*X}$ and $\boldsymbol{\theta}_k^{*Y}$ are parameter subsets that we wish to penalize. The particular parameters we penalize are the group-specific covariances and regression vectors; hence, $\boldsymbol{\theta}_k^{*X} \equiv \text{vec}(\boldsymbol{\Sigma}_k)$ and $\boldsymbol{\theta}_k^{*Y} \equiv \boldsymbol{\beta}_k$ in (6) (although under one approach below we consider $\boldsymbol{\theta}_k^{*Y} \equiv (\boldsymbol{\beta}_k, \sigma_k^2)^T$). Penalization is required because the corresponding ML estimates may be ill-behaved or ill-defined and we are interested in group-specific feature importance and conditional independence structure.

In general, tuning of penalty parameters is challenging in the latent group setting. The common approach, based on cross-validation (CV), needs to be handled with care when the estimation of parameters requires iterative procedures converging to (local) maxima, such as the EM algorithm. Specifically, performing CV at each iteration of the algorithm would

change the penalty and, thus, also change the objective function at each iteration. A brute-force solution to the problem would be to pre-specify a grid of values for the penalty and select the value that optimizes a specific criterion. However, apart from open issues related to the range and length of the grid, this would also be a computationally burdensome task, requiring multiple EM processes (each with multiple starts) for each point at the grid.

Given these considerations, we argue that more viable strategies are the following; (i) using “universal penalties” from existing literature, (ii) using CV in a stepwise manner, and (iii) considering the penalties as free parameters under estimation. The first strategy is advisable to use for parameters whose learning is not the main goal of the analysis, but for which regularization is required in order to attain workable, non-spurious solutions. Universal penalties which satisfy certain theoretical requirements (see e.g., Donoho and Johnstone, 1994; Städler and Mukherjee, 2013) typically work well to that end. On the other hand, for the main parameters of interest, whose learning (e.g. estimation, sparsity patterns) is of importance, the second and third strategies seem to be more appropriate as they offer more specific, application-driven solutions. In short, the stepwise CV approach entails adjusting the penalties a few times during the EM and running the algorithm sufficiently long in order to reach local maxima after the last adjustment. The last strategy, requires maximizing the objective with respect to the penalties and, thus, requires the introduction of a prior distribution. Within this framework one could ideally (yet not necessarily) consider properties of the prior; for instance, its behavior under small and/or large samples and the consequent effect on shrinkage, among others. From a pragmatic perspective, the prior choice is linked to more realistic considerations relating to the maximization required in the EM algorithm. For instance, the half-Cauchy prior, which is a standard choice for penalties (Polson and Scott, 2012) in fully Bayesian implementations based on posterior sampling, may not be necessarily convenient to work with within an EM framework.

We proceed with a description of the penalty functions, discussing our choices from both penalized likelihood and Bayesian viewpoints, and explaining our reasoning for the way that penalty parameters are tuned based on the aforementioned strategies. For the remainder of this Section, it is convenient to discuss the corresponding solutions under known group labels. For the subset of datapoints where $z_i = k$, let us denote the $n_k \times 1$ response by \mathbf{y}_k and the $n_k \times p$ predictor matrix by \mathbf{X}_k for $k = 1, \dots, K$. We emphasize that this is for expositional clarity only; the actual solutions under latent group labels are, of course, obtained iteratively via the EM algorithm presented in Section 3.

2.2.1 REGULARIZATION OF Σ_k

For the regularization of Σ_k we use the graphical lasso (Meinshausen and Bühlmann, 2006; Yuan and Lin, 2007; Friedman et al., 2008a). The graphical lasso induces sparsity in the inverse covariance matrix $\Omega_k = \Sigma_k^{-1}$ for group k . In this case we have $\text{pen}(\boldsymbol{\theta}_k^{*x}) \equiv \text{pen}(\Omega_k) = -\zeta \|\Omega_k\|_1$ in (6), where $\zeta > 0$ controls the strength of regularization and $\|\cdot\|_q$ is the L_q norm. For known group labels the graphical lasso estimate would be

$$\hat{\Omega}_k = \arg \max_{\Omega_k \in M^+} \left\{ \log |\Omega_k| - \text{tr}(\Omega_k \hat{\mathbf{S}}_k) - \zeta \|\Omega_k\|_1 \right\}, \quad (7)$$

where M^+ is the space of positive definite matrices and $\hat{\mathbf{S}}_k$ is the ML covariance estimate of \mathbf{X}_k . The solution in (7) is equivalent to the posterior mode under a likelihood as in (3)

and a prior distribution of the following form

$$p(\boldsymbol{\Omega}_k | \psi) \propto \left[\prod_{j=1}^p \text{Exp}(\omega_{kj|j} | \psi/2) \prod_{j < l, l=2}^p \text{DE}(\omega_{kjl} | 0, \psi^{-1}) \right] \mathbb{1}_{\{\boldsymbol{\Omega}_k \in M^+\}}, \quad (8)$$

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function, $\text{Exp}(\cdot | r)$ is the exponential distribution with rate $r > 0$ and $\text{DE}(\cdot | \mu, b)$ is the double exponential distribution with location $\mu \in \mathbb{R}$ and scale $b > 0$. The connection between (7) and the corresponding posterior mode under the likelihood in (3) and the prior in (8) is that for any given value of ζ we have that $\psi = n_k \zeta$ (Wang, 2012).

For the graphical lasso penalty, we use the “universal” threshold $\tilde{\psi} = \sqrt{2n \log p}/2$ derived from the logical arguments developed in Städler and Mukherjee (2013). Note that here we use n instead of n_k as the group labels will be unknown. The reason for choosing the universal-threshold approach for the graphical lasso is that we are less interested in the sparsity pattern of $\boldsymbol{\Omega}_k$ itself. Rather, we mainly want a well-behaved estimate that allows group structure to be effectively accounted for.

2.2.2 REGULARIZATION OF $\boldsymbol{\beta}_k$

The lasso approach. We consider a scaled version of the lasso (Tibshirani, 1996), where the penalty in (6) is given by $\text{pen}(\boldsymbol{\theta}_k^{*Y}) = \text{pen}(\boldsymbol{\beta}_k, \sigma_k^2) = \lambda_k \|\boldsymbol{\beta}_k\|_1 / \sigma_k$ with $\lambda_k > 0$. Here we introduce group specific penalty parameters λ_k , unlike previously, where parameter ψ is common across groups. This type of lasso regularization has been studied by Städler et al. (2010) in the context of FMR; a setting where the role of the scaling of variance parameters is much more important than in standard homogeneous regression, as the σ_k ’s may differ and the grouping is not fixed. If the groups were known, the lasso estimate would be

$$\hat{\alpha}_k, \hat{\boldsymbol{\beta}}_k, \hat{\sigma}_k^2 = \arg \min_{\alpha_k, \boldsymbol{\beta}_k, \sigma_k^2} \left\{ \frac{\|\mathbf{y}_k - \alpha_k \mathbf{1}_{n_k} - \mathbf{X}_k \boldsymbol{\beta}_k\|_2^2}{2\sigma_k^2} + \lambda_k \frac{\|\boldsymbol{\beta}_k\|_1}{\sigma_k} + (n_k + p + 2) \log \sigma_k \right\}, \quad (9)$$

where $\mathbf{1}_q$ denotes a q -dimensional vector of ones. The solution in (9), which is slightly different than the one in Städler et al. (2010), corresponds to the posterior mode under the Bayesian lasso formulation (Park and Casella, 2008), that specifies independent double exponential priors for the regression coefficients conditional on the error variance which is assigned the scale-invariant Jeffreys prior. Namely,

$$p(\boldsymbol{\beta}_k | \sigma_k^2, \lambda_k) = \prod_{j=1}^p \frac{\lambda_k}{2\sigma_k} \exp \left(-\lambda_k \frac{|\beta_{kj}|}{\sigma_k} \right) \text{ and } p(\sigma_k^2) \propto \frac{1}{\sigma_k^2}, \quad (10)$$

with the correspondence to (9) completed when $p(\alpha_k) \propto 1$. We propose two methods for handling λ_k ; the *fixed-penalty lasso* (FLasso) based on a plug-in estimate and the *random-penalty lasso* (RLasso) based on the construction of a suitable prior.

FLasso. This approach is essentially a two-step tuning procedure. We start with initial estimates, $\hat{\lambda}_k^{(0)}$ obtained by minimizing the CV mean squared error based on some prior clustering of the data. Then, at a certain iteration we re-calculate the CV estimates and fix each group penalty to the new estimate $\hat{\lambda}_k^{(1)}$ for all further EM iterations. Specifically, we fix the parameter after the first iteration where the group assignments do not change. From

a Bayesian perspective the FLasso approach can be viewed as an empirical Bayes method as we use the data in order to plug-in $\hat{\lambda}_k^{(0)}$ and $\hat{\lambda}_k^{(1)}$ in the prior of β_k appearing in (10). The monotonic behaviour of the EM may be disrupted at the re-estimation iteration, but after that point it will hold.

RLasso. In this approach we propose placing a prior distribution on λ_k so that this parameter will be automatically updated during the EM. We construct the prior so that it satisfies the requirement of supporting no penalization asymptotically ($\lambda_k \rightarrow 0$ as $n \rightarrow \infty$). A suitable prior for our purposes is the Pareto distribution whose scale parameter is also the lower bound of its support. Specifically, we have a prior distribution with scale $a_n > 0$ and shape $b_n > 0$ (parameters are defined to depend on n) of the following form

$$p(\lambda_k) = b_n a_n^{b_n} \lambda_k^{-(b_n+1)}, \quad (11)$$

where $\lambda_k \in [a_n, \infty)$. In our setting parameter a_n does need to be specified explicitly and is regarded to be decreasing in n , while the shape parameter is specified as $b_n = (p - 1) - c\sqrt{2K \log p/n}$ for some $c \in (0, 1]$. The rationale for these choices and further details are discussed in Appendix B. As shown next, RLasso will lead to a reasonable update for λ_k during the M-step.

The normal-Jeffreys approach. The normal-Jeffreys (NJ) prior (Figueiredo, 2001) consists of independent improper priors; in our context the prior is given by

$$p(\beta_k) = \prod_{j=1}^p p(\beta_{kj}) \propto \prod_{j=1}^p |\beta_{kj}|^{-1}. \quad (12)$$

For known group labels with $p(\alpha_k, \sigma_k^2) \propto 1/\sigma_k^2$ the corresponding penalized estimate is

$$\hat{\alpha}_k, \hat{\beta}_k, \hat{\sigma}_k^2 = \arg \min_{\alpha_k, \beta_k, \sigma_k^2} \left\{ \frac{\|\mathbf{y}_k - \alpha_k \mathbf{1}_{n_k} - \mathbf{X}_k \beta_k\|_2^2}{2\sigma_k^2} + \sum_{j=1}^p \log |\beta_{kj}| + (n_k + 2) \log \sigma_k \right\}. \quad (13)$$

Here $\text{pen}(\theta_k^{*Y}) \equiv \text{pen}(\beta_k) = \sum_{j=1}^p \log |\beta_{kj}|$. The NJ is well known in the shrinkage-prior literature (Griffin and Brown, 2005; Carvalho et al., 2010; Polson and Scott, 2010). As with most shrinkage priors, (12) can be expressed as a scale-mixture of normals; namely, $p(\beta_{kj}|s_{kj}) = (2\pi s_{kj})^{-1/2} \exp(-\beta_{kj}^2/2s_{kj})$ with $\pi(s_{kj}) \propto s_{kj}^{-1}$ and, therefore, we have that $\int p(\beta_{kj}|s_{kj})\pi(s_{kj})ds_{kj} \propto |\beta_{kj}|^{-1}$. As the mixing distribution lacks a hyper-parameter the prior is characterized by the absence of a “global” scale parameter. Also, due to heavy tails small coefficients are shrunk a lot, while large signals remain relatively unaffected; similarly to other heavy-tailed priors (Carvalho et al., 2010; Griffin and Brown, 2005). Figueiredo (2003) and Bae and Mallick (2004) show that the NJ prior strongly induces sparsity and yields good performance in terms of selection.

The use of the NJ prior is appealing for the RJM framework. Handling penalties is cumbersome in our setting and the NJ prior provides an attractive “tuning-free” alternative. In general, shrinkage priors which lack a global scale parameter fail to capture the average signal density of the data (Carvalho et al., 2010); however, despite this potential shortcoming of the NJ prior the potential benefits are worth exploring. Also, the posterior mode under

(12) is easy to find through the use of an EM algorithm where the scaling parameters s_{kj} are considered latent (Figueiredo, 2003). This additional latent structure can be easily incorporated in our EM without additional computational costs. In fact, the corresponding NJ-EM update is in closed-form, which is not the case in the lasso approach.

3. The RJM-EM algorithm

In this Section we present first the expectation and maximization steps of the proposed EM algorithm. We then prove that under certain conditions on the regularization the proposed algorithm converges towards a critical point of the likelihood function.

3.1 The EM steps

The E-Step. Irrespective of regularization approach, the group-membership probabilities of the mixture model in (5) at iteration t of the algorithm are calculated as

$$m_{ki}^{(t)} \equiv \widehat{\Pr}(z_i = k | y_i, \mathbf{x}_i, \boldsymbol{\theta}_k^{(t)}) = \frac{p(y_i | \boldsymbol{\theta}_k^{Y(t)}, \mathbf{x}_i, z_i = k) p(\mathbf{x}_i | \boldsymbol{\theta}_k^{X(t)}, z_i = k) \tau_k^{(t)}}{\sum_k p(y_i | \boldsymbol{\theta}_k^{Y(t)}, \mathbf{x}_i, z_i = k) p(\mathbf{x}_i | \boldsymbol{\theta}_k^{X(t)}, z_i = k) \tau_k^{(t)}}, \quad (14)$$

for $i = 1, \dots, n$, with the distributions appearing in the right-hand side of (14) defined in (3) and (4). Let us define some quantities that will be used throughout; namely, $n_k^{(t)} = \sum_{i=1}^n m_{ki}^{(t)}$, $\mathbf{m}_k^{(t)} = (m_{k1}^{(t)}, \dots, m_{kn}^{(t)})^T$ and $\mathbf{M}_k^{(t)} = \text{diag}(\mathbf{m}_k^{(t)})$.

A convenient feature of the RJM design is that due to the hierarchical structure of the model the objective function can be split into separate simple parts; specifically,

$$Q(\boldsymbol{\theta}, \boldsymbol{\tau}, \boldsymbol{\lambda} | \boldsymbol{\theta}^{(t)}, \boldsymbol{\tau}^{(t)}, \boldsymbol{\lambda}^{(t)}) = Q^Y(\boldsymbol{\theta}^Y, \boldsymbol{\lambda} | \boldsymbol{\theta}^{Y(t)}, \boldsymbol{\lambda}^{(t)}) + Q^X(\boldsymbol{\theta}^X | \boldsymbol{\theta}^{X(t)}) + Q^Z(\boldsymbol{\tau} | \boldsymbol{\tau}^{(t)}), \quad (15)$$

where $\boldsymbol{\theta}^Y = (\boldsymbol{\theta}_1^Y, \dots, \boldsymbol{\theta}_K^Y)^T$ and $\boldsymbol{\theta}^X = (\boldsymbol{\theta}_1^X, \dots, \boldsymbol{\theta}_K^X)^T$. Here, by $\boldsymbol{\lambda}$ we denote the vector of group penalty parameters of the regression component. Depending upon regularization approach, the elements of vector $\boldsymbol{\lambda}$ at iteration t are fixed in FLasso, free and under estimation in RLasso and absent in NJ; respectively having $\boldsymbol{\lambda}^{(t)} = (\hat{\lambda}_1^{(t*)}, \dots, \hat{\lambda}_K^{(t*)})^T$ (where $t^* = \{0, 1\}$ with zero and one corresponding to the initial CV estimate and the re-estimated CV value; see FLasso in Section 2.2.2), $\boldsymbol{\lambda}^{(t)} = (\lambda_1^{(t)}, \dots, \lambda_K^{(t)})^T$ and $\boldsymbol{\lambda}^{(t)} = \emptyset$.

Starting in reverse order from the right-hand side of (15) we have that

$$Q^Z(\boldsymbol{\tau} | \boldsymbol{\tau}^{(t)}) = \sum_{k=1}^K n_k^{(t)} \log \tau_k, \quad (16)$$

while the second component of the objective function is given by

$$Q^X(\boldsymbol{\theta}^X | \boldsymbol{\theta}^{X(t)}) = \frac{1}{2} \sum_{k=1}^K \left[\sum_{i=1}^n m_{ki}^{(t)} \left[\log |\boldsymbol{\Omega}_k| - (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Omega}_k (\mathbf{x}_i - \boldsymbol{\mu}_k) \right] - \tilde{\psi} \|\boldsymbol{\Omega}_k\|_1 \right], \quad (17)$$

where $\tilde{\psi} = \sqrt{2n \log p}/2$. The last component Q^Y in (15) depends on regularization method. We define two distinct functions Q_{lasso}^Y and Q_{NJ}^Y . For lasso we use the re-parametrization

$\chi_k = \alpha_k/\sigma_k$, $\phi_k = \beta_k/\sigma_k$ and $\rho_k = \sigma_k^{-1}$ (Städler et al., 2010), resulting in

$$Q_{\text{lasso}}^Y(\boldsymbol{\theta}^Y, \boldsymbol{\lambda} | \boldsymbol{\theta}^{Y(t)}, \boldsymbol{\lambda}^{(t)}) = \sum_{k=1}^K \left[-\frac{(\rho_k \mathbf{y} - \chi_k \mathbf{1}_n - \mathbf{X}\boldsymbol{\phi}_k)^T \mathbf{M}_k^{(t)} (\rho_k \mathbf{y} - \chi_k \mathbf{1}_n - \mathbf{X}\boldsymbol{\phi}_k)}{2} \right. \\ \left. - \lambda_k \|\boldsymbol{\phi}_k\|_1 + (n_k^{(t)} + p + 2) \log \rho_k + f(\lambda_k) \right], \quad (18)$$

which is convex. Here $f^{(t)}(\lambda_k) = 0$ for FLasso and $f(\lambda_k) = c\sqrt{2K \log p/n} \log \lambda_k$ for RLasso. Under the NJ approach the corresponding objective is given by

$$Q_{\text{NJ}}^Y(\boldsymbol{\theta}^Y | \boldsymbol{\theta}^{Y(t)}) = -\frac{1}{2} \sum_{k=1}^K \left[\frac{(\mathbf{y} - \alpha_k \mathbf{1}_n - \mathbf{X}\boldsymbol{\beta}_k)^T \mathbf{M}_k^{(t)} (\mathbf{y} - \alpha_k \mathbf{1}_n - \mathbf{X}\boldsymbol{\beta}_k)}{\sigma_k^2} + \boldsymbol{\beta}_k^T \mathbf{V}_k^{(t)} \boldsymbol{\beta}_k \right. \\ \left. + (n_k^{(t)} + 2) \log \sigma_k^2 \right], \quad (19)$$

where $\mathbf{V}_k^{(t)} = \text{diag}(1/\beta_{k1}^{2(t)}, \dots, 1/\beta_{kp}^{2(t)})$. This matrix arises from the second underlying latent structure (the latent scale parameters) in the EM; details are provided in Appendix C. As we will see below, matrix \mathbf{V}_k will in fact provide the final sparse estimate of $\boldsymbol{\beta}_k$, as some of its diagonal entries go to infinity during the EM; consequently, the diagonal entries of $\mathbf{U}_k = \mathbf{V}_k^{-1}$ that go to zero correspond to the coefficients that are set equal to zero.

The M-Step. From (16) we have that the group probabilities are updated as

$$\tau_k^{(t+1)} = n_k^{(t)} / n. \quad (20)$$

Concerning parameter block $\boldsymbol{\theta}_k^X$, from (17) we obtain the following updating equations

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_{i=1}^n m_{ki}^{(t)} \mathbf{x}_i}{n_k^{(t)}}, \quad (21)$$

$$\boldsymbol{\Omega}_k^{(t+1)} = \arg \max_{\boldsymbol{\Omega}_k} \left\{ \log |\boldsymbol{\Omega}_k| - \text{tr}(\boldsymbol{\Omega}_k \mathbf{S}_k^{(t)}) - \zeta_k^{(t)} \|\boldsymbol{\Omega}_k\|_1 \right\}, \quad (22)$$

where in (22) we have that $\mathbf{S}_k^{(t)} = n_k^{-(t)} \sum_{i=1}^n m_{ki}^{(t)} (\mathbf{x}_i - \boldsymbol{\mu}_k^{(t+1)}) (\mathbf{x}_i - \boldsymbol{\mu}_k^{(t+1)})^T$ and penalty given by $\zeta_k^{(t)} = \tilde{\psi}/n_k^{(t)} = \sqrt{2n \log p}/(2n_k^{(t)})$. For the lasso objective in (18) the FLasso group penalties are simply $\lambda_k^{(t+1)} = \hat{\lambda}_k^{(t*)}$, while the RLasso update is

$$\lambda_k^{(t+1)} = \frac{cK^{1/2}}{\|\boldsymbol{\phi}_k^{(t)}\|_1} \sqrt{\frac{2 \log p}{n}} = \frac{cK^{1/2}}{\|\boldsymbol{\beta}_k^{(t)}\|_1} \left(\sigma_k^{(t)} \sqrt{\frac{2 \log p}{n}} \right). \quad (23)$$

Note that the RLasso update is a scaled version of the optimal universal penalty under orthonormal predictors (quantity inside the parenthesis in (23)) and that the scaling depends

on the sparsity of the coefficients and on c . For the components of $\boldsymbol{\theta}_k^Y$, the updates are as follows

$$\rho_k^{(t+1)} = \frac{\mathbf{y}^T \mathbf{M}_k^{(t)} (\chi_k^{(t)} \mathbf{1}_n + \mathbf{X} \boldsymbol{\phi}_k^{(t)}) + \sqrt{\left(\mathbf{y}^T \mathbf{M}_k^{(t)} (\chi_k^{(t)} \mathbf{1}_n + \mathbf{X} \boldsymbol{\phi}_k^{(t)})\right)^2 + 4\mathbf{y}^T \mathbf{M}_k^{(t)} \mathbf{y} (n_k^{(t)} + p + 2)}}{2\mathbf{y}^T \mathbf{M}_k^{(t)} \mathbf{y}}, \quad (24)$$

$$\chi_k^{(t+1)} = \frac{(\rho_k^{(t+1)} \mathbf{y} - \mathbf{X} \boldsymbol{\phi}_k^{(t)})^T \mathbf{m}_k^{(t)}}{n_k^{(t)}}, \quad (25)$$

$$\boldsymbol{\phi}_k^{(t+1)} = \arg \min_{\boldsymbol{\phi}_k} \frac{1}{2} \|\mathbf{M}_k^{1/2(t)} (\rho_k^{(t+1)} \mathbf{y} - \chi_k^{(t+1)} \mathbf{1}_n - \mathbf{X} \boldsymbol{\phi}_k)\|_2^2 + \lambda_k^{(t+1)} \|\boldsymbol{\phi}_k\|_1. \quad (26)$$

Finally, the EM updates of $\boldsymbol{\theta}_k^Y$ under the NJ prior are the following

$$\sigma_k^{2(t+1)} = \frac{(\mathbf{y} - \alpha_k^{(t)} \mathbf{1}_n - \mathbf{X} \boldsymbol{\beta}_k^{(t)})^T \mathbf{M}_k^{(t)} (\mathbf{y} - \alpha_k^{(t)} \mathbf{1}_n - \mathbf{X} \boldsymbol{\beta}_k^{(t)})}{n_k^{(t)} + 2}, \quad (27)$$

$$\alpha_k^{(t+1)} = \frac{(\mathbf{y} - \mathbf{X} \boldsymbol{\beta}_k^{(t)})^T \mathbf{m}_k^{(t)}}{n_k^{(t)}}, \quad (28)$$

$$\boldsymbol{\beta}_k^{(t+1)} = \left(\mathbf{X}^T \mathbf{M}_k^{(t)} \mathbf{X} + \sigma_k^{2(t+1)} \mathbf{V}_k^{(t)} \right)^{-1} \mathbf{X}^T \mathbf{M}_k^{(t)} (\mathbf{y} - \alpha_k^{(t+1)} \mathbf{1}_n). \quad (29)$$

As remarked previously, in practice we work with $\mathbf{U}_k^{(t)} = \mathbf{V}_k^{-1(t)}$. Specifically, we have two available options for $\boldsymbol{\beta}_k$; the first is suited for the $n > p$ case and is given by

$$\boldsymbol{\beta}_k^{(t+1)} = \mathbf{U}_k^{\frac{1}{2}(t)} \left(\sigma_k^{2(t+1)} \mathbf{I}_p + \mathbf{U}_k^{\frac{1}{2}(t)} \mathbf{X}^T \mathbf{M}_k^{(t)} \mathbf{X} \mathbf{U}_k^{\frac{1}{2}(t)} \right)^{-1} \mathbf{U}_k^{\frac{1}{2}(t)} \mathbf{X}^T \mathbf{M}_k^{(t)} (\mathbf{y} - \alpha_k^{(t+1)} \mathbf{1}_n), \quad (30)$$

while the second, which is faster to compute when $n < p$, is given by

$$\boldsymbol{\beta}_k^{(t+1)} = \sigma_k^{-2(t+1)} \mathbf{U}_k^{(t)} \left[\mathbf{I}_p - \mathbf{X}^T \left(\sigma_k^{2(t+1)} \mathbf{M}_k^{-1(t)} + \mathbf{X} \mathbf{U}_k^{(t)} \mathbf{X}^T \right)^{-1} \mathbf{X} \mathbf{U}_k^{(t)} \right] \mathbf{X}^T \mathbf{M}_k^{(t)} (\mathbf{y} - \alpha_k^{(t+1)} \mathbf{1}_n) \quad (31)$$

Additional details on practical implementation appear in Appendix D.

3.2 Convergence guarantees

3.2.1 PRELIMINARIES

The proposed EM algorithm is an expectation/conditional-maximization (ECM) as introduced by Meng and Rubin (1993). Let us recall some elements of their formalism. We call $\boldsymbol{\xi} \in \Xi$ the variable optimised in the M step. The corresponding optimised function is $Q(\boldsymbol{\xi} | \boldsymbol{\xi}^{(t)})$, where $\boldsymbol{\xi}^{(t)}$ is the value of the parameter after t ECM steps. Then, the exact M step is defined as

$$\boldsymbol{\xi}^{(t+1)} := \arg \max_{\boldsymbol{\xi} \in \Xi} Q(\boldsymbol{\xi} | \boldsymbol{\xi}^{(t)}). \quad (32)$$

When the optimisation in (32) is inconvenient, Meng and Rubin (1993) proposed to replace it by $S \in \mathbb{N}^*$ successive block-wise updates (“conditional maximization”; CM). Given S constraint functions $\{g_s(\boldsymbol{\xi})\}_{s=1}^S$, the CM step is decomposed into the S intermediary steps:

$$\boldsymbol{\xi}^{(t+s/S)} := \arg \max_{\boldsymbol{\xi} \in \Xi : g_s(\boldsymbol{\xi}) = g_s(\boldsymbol{\xi}^{(t+(s-1)/S)})} Q(\boldsymbol{\xi} | \boldsymbol{\xi}^{(t)}), \quad (33)$$

for $s = 1, \dots, S$. In their theorems 2 and 3, Meng and Rubin (1993) provide conditions under which all limit points of any ECM sequence are critical points of the observed likelihood. We propose a reformulation of their theorem 3, where we list explicitly all the required conditions.

Theorem 1 (Theorem 3 of Meng and Rubin (1993)) *With $r \in \mathbb{N}^*$, let Ξ be a subset of the Euclidean space \mathbb{R}^r . Let $\{\boldsymbol{\xi}^{(t)}\}_{t \in \mathbb{N}} \in \Xi^{\mathbb{N}}$ be an ECM sequence that has an observed log-likelihood called $L(\boldsymbol{\xi})$ as its objective function. The initial term $\boldsymbol{\xi}^{(0)}$ is such that $L(\boldsymbol{\xi}^{(0)}) > -\infty$. Let $Q(\boldsymbol{\xi} | \boldsymbol{\xi}^{(t)})$ be the corresponding expected complete likelihood that is conditionally maximised at each CM step, with constraints functions $\{g_s(\boldsymbol{\xi})\}_{s=1}^S$. Finally, call Ξ° the interior of Ξ , and assume that:*

- Each conditional maximisation in the CM step (33) has a unique optimum
- $\forall s, g_s(\boldsymbol{\xi})$ is differentiable and the gradient $\nabla g_s(\boldsymbol{\xi}) \in \mathbb{R}^{r \times d_s}$ is of full rank on Ξ°
- $\bigcap_{s=1}^S \{\nabla g_s(\theta) u | u \in \mathbb{R}^{d_s}\} = \{0\}$
- The condition (6)-(10) of Wu (1983):

(6) $\Xi_{\boldsymbol{\xi}^{(0)}} := \left\{ \boldsymbol{\xi} \in \Xi | L(\boldsymbol{\xi}) \geq L(\boldsymbol{\xi}^{(0)}) \right\}$ is compact for any $L(\boldsymbol{\xi}^{(0)}) > -\infty$

(7) L is continuous on Ξ and differentiable on Ξ°

(9) $\Xi_{\boldsymbol{\xi}^{(0)}} \subseteq \Xi^\circ$

(10) $Q(\boldsymbol{\xi}_1 | \boldsymbol{\xi}_2)$ is continuous in both $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$.

Then all limit points of $\{\boldsymbol{\xi}^{(t)}\}_{t \in \mathbb{N}}$ are stationary points of the objective $L(\boldsymbol{\xi})$.

Note that condition (8) of Wu (1983):

(8) The sequence $\{L(\boldsymbol{\xi}^{(t)})\}_t$ is upper bounded for any $\boldsymbol{\xi}^{(0)} \in \Xi$

is verified as a direct consequence of (6) and (7) and is actually not an additional condition.

3.2.2 MAIN RESULTS

Here, we apply the convergence Theorem 1 to the proposed ECM algorithm for the RJM model. First we show that without modification, our ECM verifies almost all the hypotheses of Theorem 1. In particular the ones specific to the ECM procedure, as laid out by Meng and Rubin (1993). Then, we provide conditions on the ECM penalization under which the remaining, more restrictive, regularity hypotheses in Wu (1983) are also verified.

In our case, the optimization variable is $\boldsymbol{\xi} := (\boldsymbol{\mu}, \boldsymbol{\Omega}, \boldsymbol{\phi}, \boldsymbol{\chi}, \boldsymbol{\rho}, \boldsymbol{\lambda}, \boldsymbol{\tau}) \in \Xi$. Where the closure of the parameter set Ξ is $\bar{\Xi} = \mathbb{R}^{Kp} \times S_p(\mathbb{R})^+ \times \mathbb{R}^{Kp} \times \mathbb{R}^K \times \mathbb{R}_+^K \times \mathbb{R}_+^K \times$

$S_K \subset \mathbb{R}^{K(p^2+2p+3)}$, with $S_p(\mathbb{R})^+$ the cone of positive semi-definite matrices of size p and $S_K := \{\boldsymbol{\tau} \in [0, 1]^K \mid \sum_k \tau_k = 1\}$. Its interior is $\Xi^\circ = \mathbb{R}^{Kp} \times S_p(\mathbb{R})^{++} \times \mathbb{R}^{Kp} \times \mathbb{R}^K \times \mathbb{R}_+^{*K} \times \mathbb{R}_+^{*K} \times S_K^\circ$, with $S_p(\mathbb{R})^{++}$ the open cone of positive definite matrices of size p and $S_K^\circ := \{\boldsymbol{\tau} \in]0, 1[^K \mid \sum_k \tau_k = 1\}$. The ECM sequence takes its values in Ξ . With the proper priors, the parameters $\boldsymbol{\Omega}, \boldsymbol{\rho}, \boldsymbol{\lambda}$, and $\boldsymbol{\tau}$ cannot take values on the border of their respective sets during an ECM sequence. In such a scenario, the ECM sequence lives in Ξ° and we can simply consider that $\Xi = \Xi^\circ$, which helps with several of the hypotheses. To ensure this property, it is sufficient to set the regularization such that the objective $L(\boldsymbol{\xi})$ is infinite everywhere on the border. This objective is the penalized observed log-likelihood function:

$$\begin{aligned} L(\boldsymbol{\xi}) &= \sum_{i=1}^n \log \left(\sum_{k=1}^K p(y_i | \boldsymbol{\theta}_k^Y, \mathbf{x}_i, z_i = k) p(\mathbf{x}_i | \boldsymbol{\theta}_k^X, z_i = k) \tau_k \right) - \text{pen}(\boldsymbol{\xi}) \\ &= \sum_{i=1}^n \log \sum_{k=1}^K \exp \left(-\frac{1}{2} \left((y_i \rho_k - \chi_k - \mathbf{x}_i^T \boldsymbol{\phi}_k)^2 - 2 \log \rho_k \right. \right. \\ &\quad \left. \left. + \|\mathbf{x}_i - \boldsymbol{\mu}_k\|_{\boldsymbol{\Omega}_k}^2 - \log |\boldsymbol{\Omega}_k| \right. \right. \\ &\quad \left. \left. - 2 \log \tau_k + (p+1) \log 2\pi + \frac{2}{n} \text{pen}(\boldsymbol{\xi}) \right) \right), \end{aligned} \quad (34)$$

where $\|\mathbf{x}_i - \boldsymbol{\mu}_k\|_{\boldsymbol{\Omega}_k}^2 := (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Omega}_k (\mathbf{x}_i - \boldsymbol{\mu}_k)$. The form $\text{pen}(\boldsymbol{\xi})$ for the penalty term is a generalization of the separable penalty $\sum_k \text{pen}(\boldsymbol{\theta}_k^{*X}) + \sum_k \text{pen}(\boldsymbol{\theta}_k^{*Y})$ proposed in Eq. (6). With the posterior weights $m_{ki}^{(t)} = \widehat{\Pr}(z_i = k | y_i, \mathbf{x}_i, \boldsymbol{\xi}^{(t)})$ as defined in the E-step (14), we can define the function $Q(\boldsymbol{\xi} | \boldsymbol{\xi}^{(t)})$ to maximise in the CM step:

$$\begin{aligned} Q(\boldsymbol{\xi} | \boldsymbol{\xi}^{(t)}) &= \sum_{i=1}^n \sum_{k=1}^K -\frac{1}{2} m_{ki}^{(t)} \left((y_i \rho_k - \chi_k - \mathbf{x}_i^T \boldsymbol{\phi}_k)^2 - 2 \log \rho_k \right. \\ &\quad \left. + \|\mathbf{x}_i - \boldsymbol{\mu}_k\|_{\boldsymbol{\Omega}_k}^2 - \log |\boldsymbol{\Omega}_k| \right. \\ &\quad \left. - 2 \log \tau_k + (p+1) \log 2\pi + \frac{2}{n} \text{pen}(\boldsymbol{\xi}) \right). \end{aligned} \quad (35)$$

The conditional maximization of this function is carried out in Eq. (20) to (26), for one specific version of the penalty $\text{pen}(\boldsymbol{\xi})$. As required by Theorem 1, each of these optimizations is uniquely defined. This property is penalty dependent. Hence, in general it is required to use penalties/priors on the parameters that lead to uni-modal posterior distributions.

On the other hand, the general structure of the conditional updates is independent of the penalty. Indeed, we propose a block-type update where each block is updated conditionally to all other being fixed. The order of the updates is: $\boldsymbol{\tau} \rightarrow \boldsymbol{\mu} \rightarrow \boldsymbol{\Omega} \rightarrow \boldsymbol{\lambda} \rightarrow \boldsymbol{\rho} \rightarrow \boldsymbol{\chi} \rightarrow \boldsymbol{\phi}$.

This correspond to constraint functions of the form:

$$\begin{aligned}
 g_1(\xi) &= \xi \setminus \tau := (\mu, \Omega, \lambda, \rho, \chi, \phi), \\
 g_2(\xi) &= \xi \setminus \mu := (\tau, \Omega, \lambda, \rho, \chi, \phi), \\
 g_3(\xi) &= \xi \setminus \Omega := (\tau, \mu, \lambda, \rho, \chi, \phi), \\
 g_4(\xi) &= \xi \setminus \lambda := (\tau, \mu, \Omega, \rho, \chi, \phi), \\
 g_5(\xi) &= \xi \setminus \rho := (\tau, \mu, \Omega, \lambda, \chi, \phi), \\
 g_6(\xi) &= \xi \setminus \chi := (\tau, \mu, \Omega, \lambda, \rho, \phi), \\
 g_7(\xi) &= \xi \setminus \phi := (\tau, \mu, \Omega, \lambda, \rho, \chi).
 \end{aligned} \tag{36}$$

When the joint optimization in several consecutive blocks is possible, usually because they are separate in the objective, this approach can be simplified by “fusing” the corresponding blocks. For instance, in Eq (20) to (24), we perform a joint optimization in $\tau, \mu, \Omega, \lambda, \rho$ under the constraint that ϕ, χ is fixed. Then, in Eq (25), an optimization on χ with λ, ρ, ϕ fixed. Finally, in Eq (26), an optimization on ϕ with λ, ρ, χ fixed. Note that in this case, the optimization in τ, μ, Ω is a true M step and is independent on the other parameters. Hence, once this block has been updated in the first step, constraining it to remain fixed in the subsequent steps is unnecessary.

The functions $g_s(\xi)$ defined in (36) are obviously differentiable on Ξ with gradients of the form:

$$\nabla g_s(\xi) = [0_{d_s \times d_1} \quad \dots \quad 0_{d_s \times d_{s-1}} \quad I_{d_s} \quad 0_{d_s \times d_{s+1}} \quad \dots \quad 0_{d_s \times d_S}]^T \in \mathbb{R}^{r \times d_s},$$

which are of rank d_s (full rank), with $r := \sum_s d_s = K(p^2 + 2p + 3)$. We can also see that there is no “overlap” between their non-zero components, which results in the desired property that $\bigcap_{s=1}^S \{\nabla g_s(\theta) u | u \in \mathbb{R}^{d_s}\} = \{0\}$. As a consequence, as long as the penalty is chosen such that the posterior distribution in each parameter is unimodal, the algorithm verifies the three “ECM-specific” conditions for convergence introduced by Meng and Rubin (1993).

Among the basic conditions identified by Wu (1983), some are also systematically verified by our algorithm with little assumption on the penalty; namely:

- (7) The model part of $L(\xi)$ is always continuous and differentiable in $\Xi = \Xi^\circ$. Hence, this property is guaranteed for $L(\xi)$ as long as the penalty term is also continuous and differentiable.
- (9) $\Xi_{\xi^{(0)}} := \left\{ \xi \in \Xi | L(\xi) \geq L(\xi^{(0)}) \right\} \subseteq \Xi = \Xi^\circ$.
- (10) $Q(\xi_1 | \xi_2)$ is continuous in ξ_1 for the same reason that $L(\xi)$ is continuous in ξ . The dependency of $Q(\xi_1 | \xi_2)$ on ξ_2 is entirely through the terms $p(z_i = k | y_i, \mathbf{x}_i, \xi_2) = p(y_i, \mathbf{x}_i, z_i = k | \xi_2) / \sum_l p(y_i, \mathbf{x}_i, z_i = l | \xi_2)$ which are continuous in ξ_2 for the same reason that the likelihood is.

The final missing hypothesis is (6), the compacity of the level lines of the likelihood function. This property is much more restrictive and requires specific hypotheses on the regularization. The following theorem synthesizes every observation made so far and provides sufficient conditions to verify the final hypothesis.

Theorem 2 (Convergence of the ECM algorithm for RJMs) Consider an ECM sequence $\{\boldsymbol{\xi}^{(t)}\}_{t \in \mathbb{N}} \in \Xi^{\mathbb{N}}$ with objective function the observed log-likelihood $L(\boldsymbol{\xi})$ of the RJM model (34). The initial term $\boldsymbol{\xi}^{(0)}$ is such that $L(\boldsymbol{\xi}^{(0)}) > -\infty$ and the conditional maximization step of the expected complete likelihood $Q(\boldsymbol{\xi}|\boldsymbol{\xi}^{(t)})$ in Eq. (35) is conducted using the block-wise scheme defined by the constraint function $g_s(\boldsymbol{\xi})$ in Eq. (36). Assume that the regularization term $\text{pen}(\boldsymbol{\xi})$ is continuous, differentiable and such that each of the block maximizations is unique (uni-modal posterior). Assume additionally that there exists a positive constant $\delta > 0$ such that

$$\text{pen}(\boldsymbol{\xi}) \geq \delta \sum_{k=1}^K (\log \tau_k^{-1} + \|\mu_k\| + \|\Omega_k\| + \log |\Omega_k^{-1}| + f_\lambda(\lambda_k) + \rho_k + \log \rho_k^{-1} + |\chi_k| + \|\phi_k\|), \quad (37)$$

where f_λ is a lower bounded function on \mathbb{R}_+^* such that $f_\lambda(x) \xrightarrow{x \rightarrow 0} +\infty$ and $f_\lambda(x) \xrightarrow{x \rightarrow +\infty} +\infty$.

Then, Theorem 1 applies and all limit points of $\{\boldsymbol{\xi}^{(t)}\}_{t \in \mathbb{N}}$ are stationary points of the objective $L(\boldsymbol{\xi})$.

Remark 3

- The norms $\|\cdot\|$ on each parameters in Eq. (45) (in Appendix E) are unspecified since all norms are equivalent in finite dimension.
- For f_λ , the lower bound on the penalty in λ_k , a function such as $f_\lambda(x) = x - \log x$ is suitable.

Sketch of proof: The full details of the proof can be found in Appendix E, providing here a brief proof sketch. We prove that under the assumptions of Theorem 2, all the hypotheses of Theorem 1 are met, which yields the desired convergence result.

As previously discussed, all the hypotheses of Theorem 1, save for hypothesis (6), of Wu (1983) are organically verified within the RJM model. Provided that $L(\boldsymbol{\xi})$ is infinite on the border of Ξ , we can safely take $\Xi = \Xi^\circ$, and then the few required assumptions on the penalty are verified: it is continuous, differentiable and such that each of the block maximizations is unique. Hence, all the efforts of the proof are spent on proving that $L(\boldsymbol{\xi})$ is infinite on the border of Ξ and that hypothesis (6) is verified. This is done all at once; thanks to the control (45), we are able to define an increasing family Ξ_m of compacts of Ξ° such that the log-likelihood $L(\boldsymbol{\xi})$ on any point of $\Xi \setminus \Xi_m$ is as low as desired with a well chosen compact Ξ_m . With this result, we have that (i) $L(\boldsymbol{\xi})$ is $-\infty$ outside of Ξ° and (ii) $\Xi_{\boldsymbol{\xi}^{(0)}} := \{\boldsymbol{\xi} \in \Xi | L(\boldsymbol{\xi}) \geq L(\boldsymbol{\xi}^{(0)})\}$ is compact for any $L(\boldsymbol{\xi}^{(0)}) > -\infty$ (hypothesis (6)), allowing us to conclude.

4. Prediction and cluster selection

An interesting feature of RJMs relates to prediction. Specifically, a new observation $\mathbf{x}^* \in \mathbb{R}^p$ can be allocated to a cluster via the quantities $\hat{\pi}_k^* \propto \hat{\tau}_k \varphi_p(\mathbf{x}^* | \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)$, where $\hat{\tau}_k$, $\hat{\boldsymbol{\mu}}_k$ and $\hat{\boldsymbol{\Sigma}}_k$ are EM estimates for $k = 1, \dots, K$. A simple prediction of y^* then follows via $\hat{y}^* = \hat{\alpha}_{\tilde{k}} + \mathbf{x}^{*T} \hat{\boldsymbol{\beta}}_{\tilde{k}}$, where $\tilde{k} = \arg \max_k \hat{\pi}_k$ and $\hat{\alpha}_{\tilde{k}}$ and $\hat{\boldsymbol{\beta}}_{\tilde{k}}$ are the corresponding EM estimates.

Although not our primary focus in this paper, it is interesting to briefly consider the idea of setting K based on predictive loss. In more detail, let \mathcal{G} denote the set of the number of clusters under consideration, so that the cluster indicator under model $g \in \mathcal{G}$ is $k_g = 1, \dots, K_g$. Under each model g we can obtain cluster allocations for a subset of held-out test data \mathbf{y}^* and \mathbf{X}^* . Denote by $\mathbf{y}_{k_g}^*$ the $n_{k_g}^* \times 1$ test-response vector and by $\mathbf{X}_{k_g}^*$ the $n_{k_g}^* \times p$ test-feature matrix assigned to group $k_g = 1, \dots, K_g$ conditional on g . Then, the solution for the “best” group-wise predictive model (in an ℓ_2 sense) is given by

$$\hat{g}^{\text{pred}} = \arg \min_{g \in \mathcal{G}} \left\{ \frac{1}{K_g} \sum_{k_g=1}^{K_g} \frac{\|\mathbf{y}_{k_g}^* - \hat{\alpha}_{k_g} \mathbf{1}_{n_{k_g}^*} - \mathbf{X}_{k_g}^{*T} \hat{\beta}_{k_g}\|_2^2}{n_{k_g}^*} \right\}. \quad (38)$$

This effectively sets K to minimize predictive loss and connects in a way supervised and unsupervised learning, providing a simple and quite natural way of determining the number of clusters based on a “guided” search that aims to optimize prediction of \mathbf{y} .

Of course, standard approaches for inferring the number of clusters, such as information criteria, can be used within the RJM framework. For instance, using BIC (Schwarz, 1978) in our case translates in selecting the number of clusters as

$$\hat{g}^{\text{BIC}} = \arg \max_{g \in \mathcal{G}} \left\{ 2 \log p(\mathbf{y}, \mathbf{X} | \boldsymbol{\theta}_g, \boldsymbol{\tau}_g) - \log(n) \nu_g \right\}, \quad (39)$$

where ν_g is the number of elements in $\boldsymbol{\theta}_g$ that are not set equal to zero. Similarly, for AIC (Akaike, 1974) we have

$$\hat{g}^{\text{AIC}} = \arg \max_{g \in \mathcal{G}} \left\{ 2 \log p(\mathbf{y}, \mathbf{X} | \boldsymbol{\theta}_g, \boldsymbol{\tau}_g) - 2\nu_g \right\}. \quad (40)$$

5. Empirical examples

In this Section we present results from simulation experiments, starting with a small-scale simulation in Section 5.1 which allows us to evaluate and visualize easily the various learning aspects of RJMs. In Section 5.2 we use data from The Cancer Genome Atlas in semi-synthetic examples of much larger scale, providing detailed comparisons with baseline and various oracle-type approaches. All simulations are based on data-generating mechanisms which are multivariate generalizations of three elementary problems which are depicted in Figure 3, the purpose of which is to facilitate understanding of more complex multivariate problems as the ones in Section 5.2 via illustration of simpler univariate analogues. Finally, in Section 5.3 we show results using fully empirical data.

5.1 Small-scale simulations

Set-up. We consider two groups ($K = 2$) with total $n = 100$ and balanced groups, i.e. $n_k = 50$ for $k \in \{1, 2\}$. The number of predictors is $p = 10$, where in each group only the first predictor (\mathbf{x}_{k1}) has a non-zero coefficient, i.e. only $\beta_{k1} \neq 0$ for $k \in \{1, 2\}$. The covariates are generated as $\mathbf{X}_k \sim N_p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, with $\boldsymbol{\mu}_1 = (0, \dots, 0)^T$ and $\boldsymbol{\mu}_2 = (1, \dots, 1)^T$ being of dimensionality $p \times 1$. For the covariances we consider two scenarios: an *uncorrelated-scenario* with diagonal covariances of the form $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \mathbf{I}_p$ and a *correlated-scenario* with

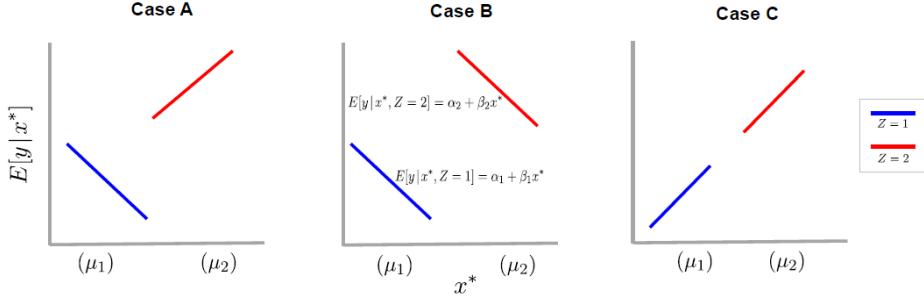


Figure 3: Some interesting cases of group structure. Univariate analogues of problems considered in the empirical examples (all of which are multivariate) in order to illustrate the key ideas. Shown are two latent groups with some separation between the group-wise means of a single feature x^* . Three specific cases, that differ with respect to the regression model linking y and x^* , are shown: equal-intercept/unequal-slope (Case A); unequal-intercept/equal-slope (Case B); and equal intercept and slope (Case C).

non-diagonal covariances, where each variable \mathbf{x}_{kj} , for $j = 2, \dots, p$, is again Gaussian noise, but the signal variable \mathbf{x}_{k1} is generated as $\mathbf{x}_{k1} \sim N_{n_k}(1.5\mathbf{x}_{k3} + 0.5\mathbf{x}_{k5} - 0.7\mathbf{x}_{k7}, 0.5\mathbf{I}_{n_k})$. The response of each group is generated as $\mathbf{y}_k \sim N_{n_k}(\mathbf{1}_{n_k}\alpha_k + \mathbf{x}_{k1}\beta_{k1}, \sigma_k^2\mathbf{I}_{n_k})$. Specification of the slopes and intercepts is based on the three cases of Figure 3; see Table 3 (Appendix F). Finally, the error variance σ_k^2 of each group is set to fix signal strength in a label-oracle sense, namely so that the correlation between test data (for test group sample sizes of 250) and predictions from a lasso model is approximately 0.8 when group labels are known. The results that follow are from 50 repetitions of each simulation. We focus on regression signal detection, estimation of coefficients and group assignment performance.

Signal detection and estimation. The variable inclusion frequencies over 50 repetitions of the simulations for the uncorrelated scenario are presented in Figure 4. RJM-NJ performs better overall as it detects influential effects almost all of the times, while the inclusion rates of non-influential effects are much lower than 50%. RJM-FLasso is effective in detecting the signals but produces much denser solutions. RJM-RLasso solutions are sparser in comparison to FLasso; we note, however, that RLasso tends to over-shrink the coefficients of the influential predictors as well. The inclusion frequencies for the correlated scenario are similar (Appendix F, Figure 12). Violin plots of slope estimates are presented in Appendix F (Figure 13, uncorrelated scenario; Figure 14, correlated scenario), while the corresponding plots for intercepts are presented in Figures 15 and 16. The NJ estimates are overall more accurate. In all comparisons we include results from mixtures of experts obtained from R package MoEClust (Murphy and Murphy, 2020), which performs simultaneous selection for experts, gates and covariance structures using forward search model selection based on BIC. In terms of variable selection MoE performs exceptionally well under case A and yields similar results to RJM-NJ under cases B and C (see Figures 4 and 12). As for estimation, MoE leads to overall accurate slope and intercept estimates in case A, however,

the corresponding estimates in cases B and C have a higher variance in comparison to RJM estimates (see Figures 13 to 16, Appendix F).

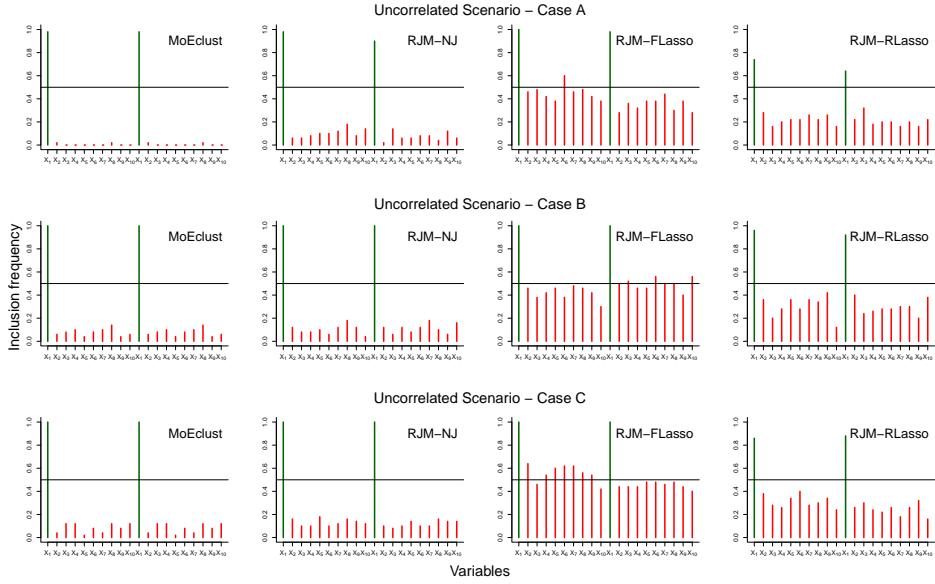


Figure 4: First simulation, uncorrelated scenario. Variable inclusion frequencies (over 50 repetitions) for signal variables (in green) and non-signal variables (in red) for regression cases A, B and C. Horizontal black lines correspond to a frequency of 0.5.

Latent group assignment. A natural question is whether including the regression part of the model within a unified framework provides any gains with respect to simply clustering the X matrix. In practice, between-group differences in means may be subtle. Hence, we consider in particular performance as the magnitude of the mean difference varies; i.e., we set $\mu_1 = \mathbf{0}_p$ and $\mu_2 = \mathbf{0}_p + d\mathbf{1}_p$ with $d = h \cdot U$, where h defines a grid ranging from 0.1 to 1 with a step size of 0.05 and $U \in \{-1, +1\}$ is a uniformly random sign. Hence, $|d|$ is a measure of the strength of the mean signal. We compare to k -means, hierarchical clustering, GMMs and MoEs as implemented in R using the default options of `kmeans`, `hclust`, `mclust` (Scrucca et al., 2016) and `MoEclust` respectively; for the latter two using the BIC-optimal model. In addition, for the clustering approaches we use as input: (i) only \mathbf{X} and (ii) \mathbf{X} together with \mathbf{y} stacked in one data matrix. For these simulations we use 20 repetitions.

One standard-error plots of adjusted Rand index averages as functions of $|d|$ are shown in Figure 5. As seen, in case A, RJMs generally outperform all methods except of MoE; the latter performs better for lower values of $|d|$, while RJMs perform better for higher values. In cases B and C (where RJM is over-parameterized) our methods remain competitive in the uncorrelated scenario, while lead to better overall results in the correlated scenario. On the other hand, MoE is not effective under cases B and C, which as argued in Section 1 (recall Figure 2) is to be expected when there are no differences in regression coefficients.

REGULARIZED JOINT MIXTURES

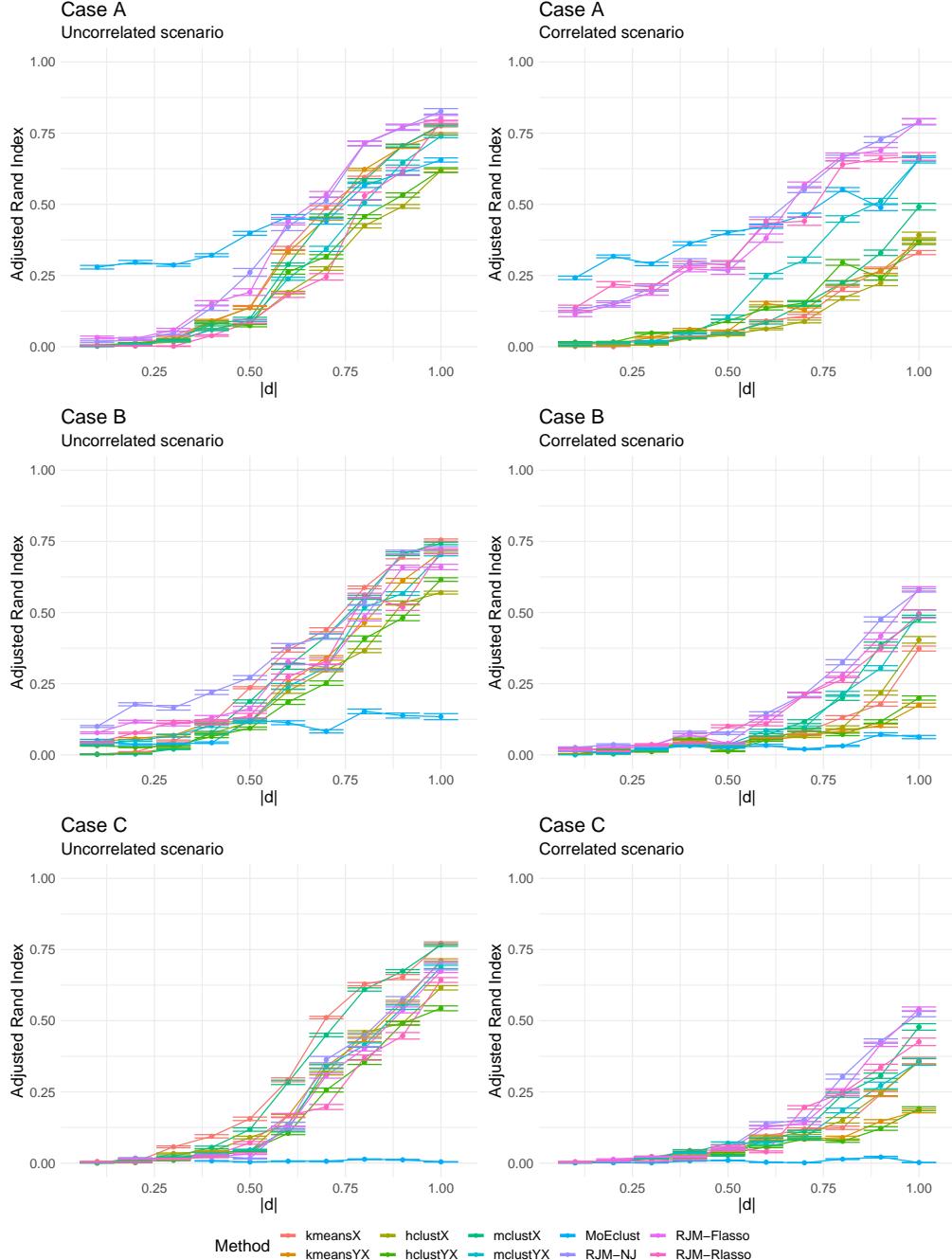


Figure 5: First simulation, cases A (top), B (middle) and C (bottom). One standard-error plots of adjusted Rand index averages (from 20 repetitions) vs absolute distance ($|d|$) of the group-wise covariate means under the uncorrelated scenario (left) and correlated scenario (right).

5.2 Semi-synthetic simulations based on real cancer data

The simulations presented below are based on data from the The Cancer Genome Atlas (TCGA, <https://cancergenome.nih.gov>). The rationale is to anchor the simulation in real covariance structures. Specifically, we use data previously used in Taschler et al. (2019), consisting of gene expression values from four cancer types; breast (BRCA), kidney renal clear cell (KIRC), lung adenocarcinoma (LUAD) and thyroid (THCA). Our strategy is to treat the cancer type as hidden: this allows us to test our approaches in the context of differential covariance structure as seen in a real group-structured problem whilst having access to true gold-standard labels.

Table 1: Second simulation. Intercept values and slope-generating mechanisms for the two groups under the three cases illustrated in Figure 3. $[\text{TN}(\mu, \sigma^2, l, u)]$ denotes a truncated normal distribution, where $\mu \in \mathbb{R}$, $\sigma^2 > 0$ and l, u are the respective lower and upper truncation bounds, while $\text{mTN}(\mu, \sigma^2, a, b)$ denotes the specific mixture of $\text{TN}(\mu, \sigma^2, -\infty, a)$ and $\text{TN}(\mu, \sigma^2, b, \infty)$ with $a < b$ and mixing parameter equal to 0.5; i.e., a truncated normal with support everywhere except in (a, b) .

Case	Group	Intercept	Slopes	
			Common locations	Disjoint locations
A	1	$\alpha_1 = \alpha_2 = 0$	$\beta_1^* \sim \text{TN}(0, \tilde{\sigma}^2, -\infty, -0.1)$	$\beta_1^* \sim \text{mTN}(0, \tilde{\sigma}^2, -0.1, 0.1)$
	2		$\beta_2^* \sim \text{TN}(0, \tilde{\sigma}^2, 0.1, \infty)$	$\beta_2^* \sim \text{mTN}(0, \tilde{\sigma}^2, -0.1, 0.1)$
B	1	$\alpha_1 = 0$	$\beta_1^* = \beta_2^* \sim \text{mTN}(0, \tilde{\sigma}^2, -0.1, 0.1)$	$\beta_1^* \sim \text{mTN}(0, \tilde{\sigma}^2, -0.1, 0.1)$
	2			$\beta_2^* \sim \text{mTN}(0, \tilde{\sigma}^2, -0.1, 0.1)$
C	1	$\alpha_1 = \alpha_2 = 0$	$\beta_1^* = \beta_2^* \sim \text{mTN}(0, \tilde{\sigma}^2, -0.1, 0.1)$	$\beta_1^* \sim \text{mTN}(0, \tilde{\sigma}^2, -0.1, 0.1)$
	2			$\beta_2^* \sim \text{mTN}(0, \tilde{\sigma}^2, -0.1, 0.1)$

Set-up. In these experiments we use covariates from two cancer types; namely, the BRCA and KIRC groups. For all simulations we use $n = 250$, balanced group sample sizes, i.e. $n_k = 125$ for $k = 1, 2$, and varying dimensionality for the features; namely, i) $p = 100$ ($n > p$ problem), ii) $p = 250$ ($n = p$ problem) and iii) $p = 500$ ($n < p$ problem). We consider sparse problems where the percentage of non-zero coefficients (β^*) is $s = 4\%$ and the setting in which some of the non-zero coefficients are at common locations and others are at disjoint locations across the two groups (placing half the non-zero coefficients at common locations). Specification of the common-location β_k^* 's will determine the three general cases depicted initially in Figure 3. To rule out very small coefficients, we draw from a truncated normal distribution, with support excluding the interval $(-0.1, 0.1)$. Group specific intercept values and slope-generating mechanisms (based on Figure 3), are summarized in Table 1. Given the matrices \mathbf{X}_k , the intercepts α_k and the sparse vectors β_k the response is generated as $\mathbf{y}_k \sim N_{n_k}(\mathbf{m}_k, \mathbf{I}_{n_k}\sigma_y^2)$, where $\mathbf{m}_k = \mathbf{I}_{n_k}\alpha_k + \mathbf{X}_k\beta_k$ for $k = 1, 2$ and $\sigma_y^2 = 1$. The scale parameter $\tilde{\sigma}^2$ in Table 1 is tuned so that the overall signal-to-noise under each case is approximately equal to three; i.e. $\text{Var}(\mathbf{m})/\sigma_y^2 \approx 3$ and $\mathbf{m} = (\mathbf{m}_1, \mathbf{m}_2)^T$.

Performance is evaluated as a function of the absolute distance $|d|$ between the group-wise feature means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. We initially normalize the features, so that $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \mathbf{0}$, and consider again the case where each element of $\boldsymbol{\mu}_2$ is shifted by $d = h \cdot u$, where $h \in$

$\{0.1, 0.2, \dots, 0.8, 0.9\}$ and u is a random uniform sign. Each simulation is repeated 20 times using random subsamples of features from the original data. Here we present results for the $n < p$ setting ($p = 500$); results for $p \in \{100, 250\}$ can be found in Appendix G. For the regression questions addressed below our aim is to compare RJM with the “clustering-then-regression” approach. Obviously, a range of regression methods could be used in the second step. As the simulations are sparse and linear by design, to ensure that the simple “clustering-then-regression” approach is not disadvantaged we use lasso in the second step.

Group assignment. We compare to the same methods considered in Section 5.1, except of MoEClust as the dimensionalities are too large for a forward model search for optimal selection of expert and gating functions. Figure 6 presents error plots of adjusted Rand index averages. In general, we observe a phase-transition type of behaviour as all methods improve as $|d|$ increases. However, the transition is faster with RJM which outperforms the other methods and stabilizes relatively quickly to correct assignment. For $p = 500$ lasso-based RJM outperforms the NJ variant for cases B and C, however, for $p = \{100, 250\}$ all RJM methods perform equally well more or less; see Figures 17 and 18 in Appendix G.

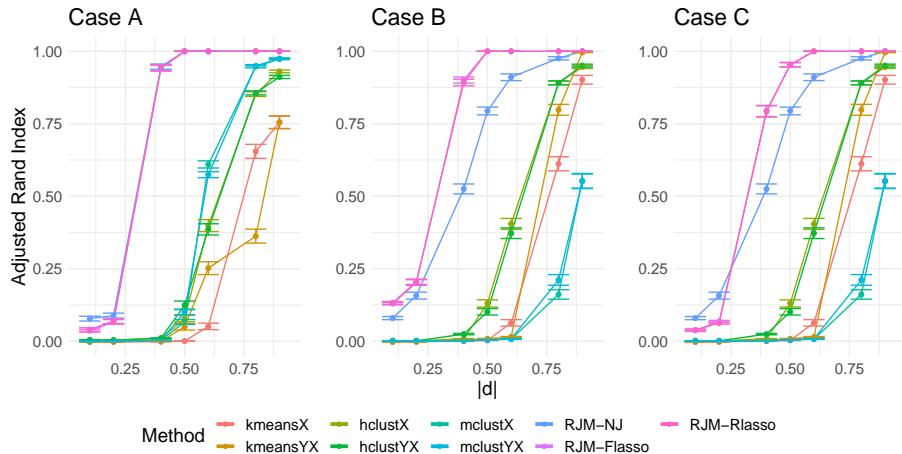


Figure 6: Second simulation, $p = 500$, group assignment. Average adjusted Rand Index as a function of the absolute distance ($|d|$) of the group-wise covariate means, for cases A (left), B (center) and C (right). [Error bars indicate standard errors from 20 repetitions.]

Variable selection. For this comparison we initially set a benchmark model called the *label-oracle-lasso*, under which the true group labels are assumed known and we fit separate lasso regressions via `glmnet` (Friedman et al., 2008b). We also consider a *cluster-lasso* model which involves separate regressions based on estimated group labels. This approach involves an initial clustering step: we give an advantage to the cluster-lasso by using, for each dimension considered, the clustering approach that performs best (`hclust` for $p = 100$ and `Mclust` for $p \in \{250, 500\}$). Naturally, the cluster-lasso will be equivalent to the oracle-lasso when group assignment is perfect.

We summarize results via the area under the ROC curve (AUC) based on the ranking of the absolute values of the coefficients. In particular, we consider the difference between

the AUC from oracle-lasso and AUC from competing methods (cluster-lasso and RJM approaches). One standard-error plots for $p = 500$ are presented in Figure 7. As expected cluster-lasso yields smaller selection loss (approaching oracle-lasso) as the separation of group-wise means increases, but so do the RJM methods. RJM-FLasso is overall better and even seems to result in slightly improved selection in comparison to the oracle-lasso as $|d|$ increases, possibly due to the fact that RJM uses weighted estimation based on the entire sample. Importantly, RJM methods overall outperform the common cluster-lasso approach in low and/or medium magnitude regions of $|d|$. These results illustrate the nontrivial gains possible from a unified treatment of the various aspects of the model vs. the simple approach of clustering followed by sparse regression. For the simulations with $p = 100$ and $p = 250$ see Figures 19 and 20 (Appendix G), respectively.

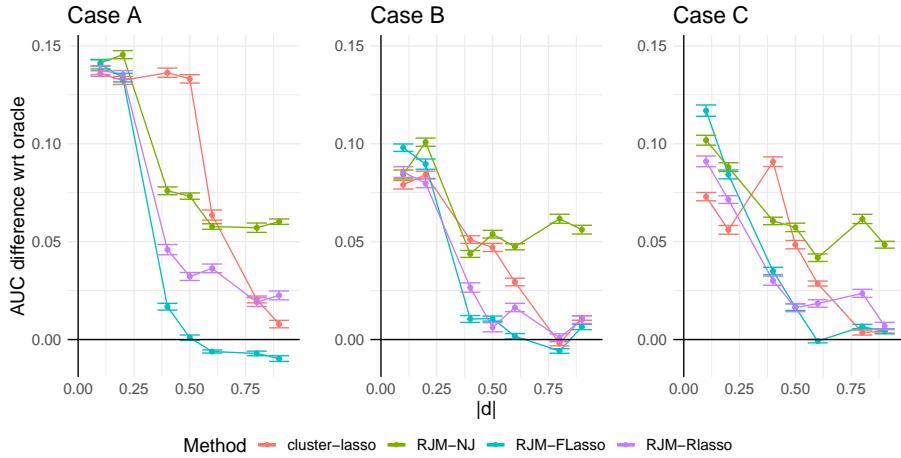


Figure 7: Second simulation, $p = 500$, variable selection. AUC loss from oracle-lasso as a function of the absolute distance ($|d|$) of the group-wise covariate means, for cases A (left), B (center) and C (right). [Error bars indicate standard errors from 20 repetitions.]

Estimation. Comparisons are made again with respect to the label-oracle-lasso; this time we consider the increase in root mean squared error (RMSE) resulting from the fact that the group labels are unknown. Here we also consider the *pooled-lasso*; a “naive” model which does not take into account group structure. This allows us to investigate the effect of ignoring group structure under each case in Table 1; this is of particular interest in Case A where the common-location coefficients have opposite signs.

Results for the $p = 500$ are summarized in Figure 8. We use standardized coefficients for the calculation of RMSE in order to have a common scale across simulations and cases. As expected, under Case A the pooled-lasso model performs poorly, while RJM methods provide overall better estimates than cluster-lasso. Under cases B and C, cluster-lasso and RJM which are over-parameterized (common-location effects are equal) perform more or less the same and are in general comparable to the pooled-lasso which is under-parameterized (due to the disjoint-location effects). The $p = 100$ and $p = 250$ cases are shown in Appendix G (Figures 21 and 22); results are in general similar with the difference that cluster-lasso

performs better when $|d| \approx 1$ and RJM-RLasso performs overall worse. Overall, our illustrations suggest that the RJM-FLasso is the most stable method, followed by RJM-NJ.

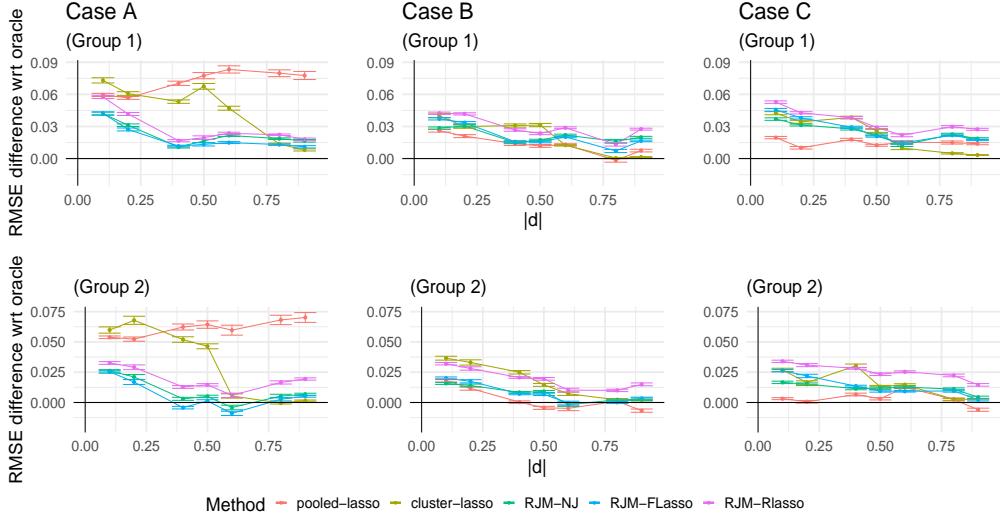


Figure 8: Second simulation, $p = 500$, estimation. Increase in RMSE relative to the oracle-lasso as a function of the absolute distance ($|d|$) of the group-wise covariate means, under group one (top) and group two (bottom), and for cases A (left), B (center) and C (right). [Error bars indicate standard errors from 20 repetitions.]

Selection of number of clusters. In Appendix H we further consider the case where the number of clusters is not known *a-priori* in simulation experiments which take into account all four cancer types. Results on cluster selection using the predictive approach described in Section 4 are shown in Figure 23.

5.3 Real cancer data example

In this Section we consider a fully non-synthetic example, where both features and response are real empirical data. The general strategy is as follows. We use the TCGA data as introduced above, with gene expression levels treated as features. Specifically, we include all four cancer types used in the previous section, selecting via stratified random sampling $n = 250$ samples in total. Stratified sampling ensures that the cancer-type proportions are preserved; specifically the dataset under consideration consists of 102 BRCA, 51 KIRC, 49 LUAD and 48 THCA observations (abbreviations as previously introduced). A total of $p = 100$ gene expression levels (selected at random from all genes) are used in these experiments. As before, in the applications that follow the true labels (i.e., the cancer type indicator) are treated as latent and hence not used in analysis, but only to evaluate performance. As responses, we use one of the $p = 100$ gene expression levels, with the remaining forming the feature set. This procedure has the advantage of allowing us to consider many different responses (genes) whilst entirely eschewing synthetic data generation. We first show a

illustrative example using one particular gene as response and then show results from all responses considered.

Illustrative analysis for a particular response gene (NAPSA). We consider the gene NAPSA (napsin A aspartic peptidase) (gene ID 9476) to illustrate the set-up. For this illustrative analysis we assume that the cancer types are given so that it is known *a priori* that there are four classes. This particular gene is used to illustrate the approach as it is informative with respect to hidden group structure, as shown in Figure 9 (left panel), but not to the extent of fully revealing the class structure. Heatmaps of the sample covariance matrices of the remaining 99 genes under each cancer type are presented in Figure 9 (panels in the right); these generally indicate slight differences in covariance structures.

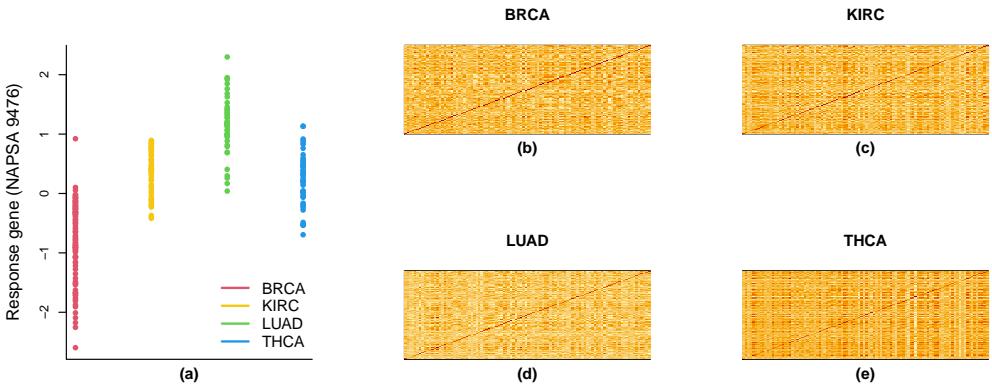


Figure 9: Real data example, data visualization for single, illustrative response. Plot (a) of response gene NAPSA in the four cancer types, and heatmaps of feature covariances in (b) BRCA, (c) KIRC, (d) LUAD and (e) THCA cancer types.

For evaluation of clustering performance we compare to the same methods as in Section 5.1, including again MoE as implemented in package `MoEclust`; however, for computational convenience, we use the option of a classification-EM algorithm, which is a faster but generally sub-optimal algorithm (although we note that initial tests suggested a gain in clustering performance for this dataset). We further consider k -medoids, fuzzy c -means (implemented via `par` and `fanny` in package `cluster`, respectively) and clustered support vector machines (`clustSVM`, package `SwarmSVM`; Gu and Han, 2013). Finally, for the purely cluster-oriented approaches (k -means/medoids, fuzzy c -means, `hclust` and `mclust`) we use as input the concatenated matrix containing the response and the predictor genes. Table 2 shows the resulting adjusted Rand index under each method (for k -means and `clustSVM`, which are highly sensitive to initialization, the values are averages from 100 runs). As seen, RJMs clearly outperform the other approaches.

Figure 10 shows the resulting regression coefficient estimates (396 in total given the four cancer types), from the three RJM variants, ranked in absolute value from highest to lowest (non-zero values in green; zero values in red). Consistent with the results presented in

Table 2: Real data application, clustering performance. Adjusted Rand index values for the ten methods under consideration using gene NAPSA as response variable.

Methods and clustering performance					
Method	k -means	k -medoids	fuzzy c -means	hclust	clustSVM
Adj. Rand Index	0.43	0.44	0.59	0.51	0.42
Method	mclust	MoEclust	RJM-NJ	RJM-FLasso	RJM-RLasso
Adj. Rand Index	0.29	0.55	0.72	0.68	0.75

Section 5.1, we observe again that RJM-NJ results in the most parsimonious model (fewer than 100 predictors), followed by RJM-RLasso (around 100 predictors), while RJM-FLasso includes the most predictors (more than 100). We also observe that the lasso variants tend to shrink the coefficients of influential predictors more than RJM-NJ; this is also generally anticipated as the NJ prior has heavier tails in comparison to the Bayesian lasso prior. Finally, the forward search of MoEclust included a few predictors in the gating networks, but resulted in entirely sparse expert networks as all regression coefficients were set to zero.

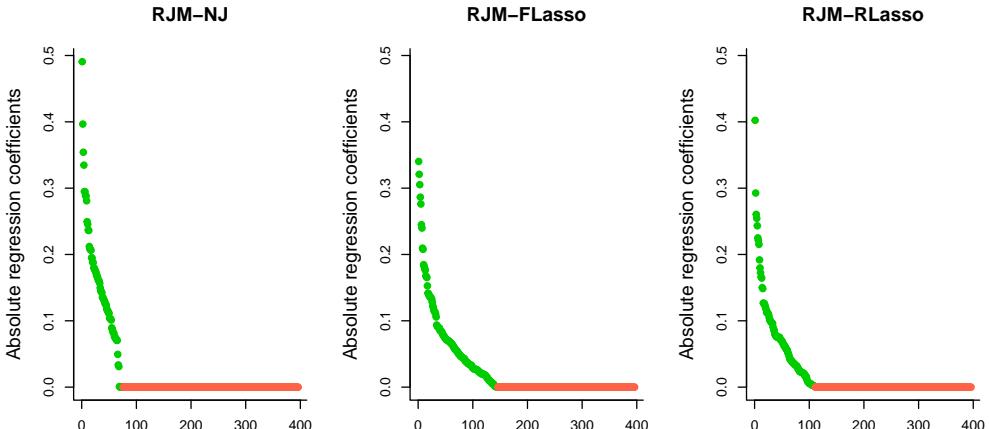


Figure 10: Real data application, regression performance. Absolute values of RJM regression coefficients ranked in decreasing order, using gene NAPSA as response variable. Green points indicate non-zero coefficients; red points, coefficients that are set equal to zero.

Performance over all responses. Above we considered a specific gene to illustrate the key ideas; here we show results from all responses. That is, we consider in turn each of the genes as response, treating all others as features. Thus, there are 100 problems considered in total (each with the same four latent subgroups). In this case, the input for the purely clustering methods is the data matrix of the predictor variables.

Violin plots of the resulting adjusted Rand index values from the ten methods are presented in Figure 11. These results, spanning one hundred different responses, support the results seen above, as the three RJM variants consistently perform relatively well over most of the responses and the results are broadly in line with some of the results in Sections 5.1 and 5.2. In Appendix I, we further consider BIC-based model selection; as shown there, RJMs select more frequently the correct number of groups in comparison to GMMs and MoEs.

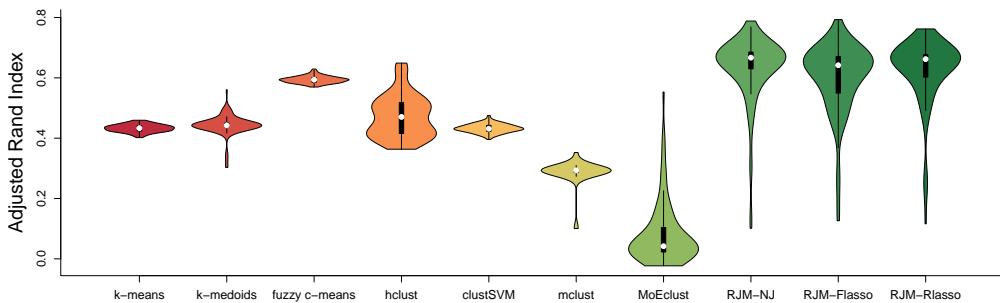


Figure 11: Real data application, clustering performance. Violin plots of adjusted Rand index values from TCGA-based experiments spanning a hundred different responses (see text for details).

5.4 Summary

Broadly speaking the RJM variants performed similarly to one another in terms of clustering/group assignment. In several situations they outperformed the other methods compared with, while, at the same time, they tended to remain competitive across the range of scenarios tested. Also, RJM can improve selection for the number of groups when this is not known at the outset. There were some differences between the RJM variants with respect to regression modelling. The NJ approach tended to perform well, in terms of variable selection and estimation, in sparse settings characterized by moderate to strong signals, while the lasso approaches yielded relatively denser models overall.

6. Discussion

We introduced a class of regularized mixture models that jointly deal with sparse covariance structure and sparse regression in the context of latent groups. We showed that principled joint modeling of these two aspects leads to gains with respect to simpler decoupled or pooled strategies and that exploiting established ℓ_1 -penalized tools and related Bayesian approaches leads to practically applicable solutions. The RJM methods presented in this paper are implemented as an R package `regjmmix`, available at <https://github.com/k-perrakis/regjmmix>. Future research directions include extensions to generalized linear models and

mixed models. Below we discuss some additional aspects and point to specific directions for future work.

Distribution shifts and shift-robust learning. By accounting for data heterogeneity, RJMs help to guard against (potentially severe) confounding of multivariate regression models by hidden group structure. This has interesting connections to distribution shifts in machine learning and shift-robust learning; see e.g. Recht et al. (2018); Heinze-Deml and Meinshausen (2021). In particular, we think RJM would be a useful tool for shift-robust learning, since it could be used to block paired data (X, Y) into distributionally non-identical groups which could in turn be used to train and test predictors in a shift-robust fashion, facilitating shift-robust learning under unknown distributional regimes.

Choice of regularization. In this work we used the graphical lasso approach for covariance estimation, mainly motivated by certain biomedical applications where network models are of interest. However, the general RJM strategy could be used with other kinds of multivariate models (e.g. factor models). For the regression coefficients we considered: (i) the Bayesian lasso prior under two strategies (FLasso/RLasso) and (ii) the NJ prior. For practitioners seeking the closest analogue to the popular lasso approach based on cross-validation in the non-latent regression setting, we recommend FLasso. When it is preferable to use a shrinkage prior with heavier tails we recommend using NJ, which can be very effective in detecting sparsity patterns without over-shrinking large coefficients. In general, our main goal was to explore some of the available regularization options, but we note that the RJM model is fairly modular in the sense that other methods from the penalized likelihood or the Bayesian literatures – recent reviews provided by Desboulets (2018) and van Erp et al. (2019), respectively – can be used within the same framework. Of course, specifics will depend upon approach; for instance, the elastic-net (Zou and Hastie, 2005) and the adaptive lasso (Zou, 2006) are fairly easy to incorporate at present, while other methods such as the horseshoe estimator (Carvalho et al., 2010) require further investigation.

High-dimensional issues. RJM remains effective when $p > n$, but a general issue when jointly modeling (Y, X) is that for relatively large p cluster allocation will be mainly guided by X . In the empirical examples RJM outperformed `mclust` (recall that without regularization RJM is equivalent to a GMM); to provide some intuition about that let us consider the two regularization steps. The first on the covariance matrix of X can be viewed as p lasso regressions (Meinshausen and Bühlmann, 2006) that essentially discard non-influential relationships among features. The second discards non-influential predictor effects on the response. Overall this sparsification may be viewed as a dimensionality reduction, which mitigates over-emphasis on X . One idea for handling this issue as p grows larger is to consider explicit weighting of the effect of X , e.g. by replacing the multivariate normal in (3) with a density of the form $N_p(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{1/\delta}$, where the “power-parameter” ($\delta > 1$) would inflate the covariance. While from a computational standpoint the proposed framework is scalable and can also handle the $p > n$ case, in very high dimensions it would become computationally burdensome. This can be potentially addressed via high-dimensional projections. Finally, although our EM convergence result is general, there remain open theoretical questions concerning rates of convergence and optimality of the estimators themselves.

Acknowledgments

We would like to thank Keefe Murphy for interesting discussions and for updating R package `MoEClust` in order to make it possible to implement the comparisons presented in this paper. We acknowledge support via the German Bundesministerium für Bildung und Forschung (BMBF) project “MechML”, the Medical Research Council (programme number MC_UU_00002/17) and the National Institute for Health Research (NIHR) Cambridge Biomedical Research Centre.

Appendix A. Simulations for Section 1.2

The simulation results presented in Section 1.2 are based on 20 repetitions where we consider two groups ($K = 2$) with total $n = 200$ and a balanced design, i.e. $n_k = 100$ for $k \in \{1, 2\}$. The number of predictors is $p = 10$. In each group only one predictor (\mathbf{x}_k^*) has a non-zero coefficient (β_k^*), and this predictor is chosen randomly over the 20 repetitions. The three cases in Figures 1 and 2 correspond to: (i) $\beta_1^* = \beta_2^* = 0.5$ (plots on the left), (ii) $\beta_1^* = 0.5, \beta_2^* = 1$ (plots on the middle), and (iii) $\beta_1^* = 0.5, \beta_2^* = 1.5$ (plots on the right). The features are generated as $\mathbf{X}_k \sim N_{10}(\boldsymbol{\mu}_k, 0.5\mathbf{I}_p)$. That is, the two feature groups share the same diagonal covariance structure, but we let the mean vectors to vary. Specifically, the first mean vector is always $\boldsymbol{\mu}_1 = (0, \dots, 0)^T$, while for the second mean vector we consider again three cases (which lead to the variability with respect to the x -axis of the plots in Figure 2); namely, (i) $\boldsymbol{\mu}_2 = (0, \dots, 0)^T$, (ii) $\boldsymbol{\mu}_2 = (0.5, \dots, 0.5)^T$ and (iii) $\boldsymbol{\mu}_2 = (1, \dots, 1)^T$. The response of each group is generated as $\mathbf{y}_k \sim N_{n_k}(\mathbf{x}_k^* \beta_k^*, \sigma_k^2 \mathbf{I}_{n_k})$, where $\sigma_k^2 = \text{Var}(\mathbf{x}_k^* \beta_k^*)/5$ for $k \in \{1, 2\}$. The regularized joint mixture model is based on the normal-Jeffreys prior discussed in Section 2.2.2. For the implementation of the Gaussian mixture and mixture-of-experts models we used R packages `mclust` (Scrucca et al., 2016) and `MoEClust` (Murphy and Murphy, 2020), respectively, using the default model-search options, selecting the BIC-optimal model.

Appendix B. Justification for the RLasso Pareto prior

We generally want the Pareto prior to be such that it will not penalize the regression coefficients asymptotically. Under the prior in (11) the mode of λ_k is a_n , while the prior mean is given by

$$\mathbb{E}(\lambda_k) = a_n \frac{b_n}{b_n - 1},$$

for $b_n > 1$ and the prior variance by

$$\text{Var}(\lambda_k) = a_n^2 \frac{b_n}{(b_n - 1)^2(b_n - 2)}$$

for $b_n > 2$. Given that $a_n \rightarrow 0$ as $n \rightarrow \infty$, in order to meet our requirement we want $b_n \rightarrow C > 2$ as $n \rightarrow \infty$; to that end, we specify $b_n = (p - 1) - c\sqrt{2K \log p/n}$, for $c \in (0, 1]$. Explicit specification of a_n is not required as it does not affect the posterior mode; any decreasing function of n (subject to $a_n > 0$) will satisfy the requirement. As for c , we recommend setting it equal to $\min\{\sqrt{2p/3n}, 1\}$ as a default option.

Appendix C. Objective function under the NJ prior

The hierarchical form of the NJ prior is $\beta_k | \mathbf{S}_k \sim N_p(\mathbf{0}, \mathbf{S}_k)$, where $\mathbf{S}_k = \text{diag}(s_{k1}, \dots, s_{kp})$ assuming latent s_{kj} with $\pi(s_{kj}) \propto s_{kj}^{-1}$ for $k = 1, \dots, K$ and $j = 1, \dots, p$. The conditional distribution of any s (dropping momentarily subscripts k, j for simplicity) is

$$p(s|\beta) = \frac{q(s)}{\int q(s)ds}, \text{ with } q(s) = p(\beta|s)s^{-1} \text{ and } \int q(s)ds = |\beta|^{-1}.$$

Given this, it follows that

$$\mathbb{E}_{s|\beta}[s^{-1}] = \int s^{-1}p(s|\beta)ds = \beta^{-2} \quad (41)$$

This result will be needed in the derivation of the E-step below. The joint prior of β_k and σ_k^2 is $p(\beta, \sigma^2 | \mathbf{S}) = p(\beta | \mathbf{S})p(\sigma^2) \propto \prod_{k=1}^K \exp\left(-\frac{1}{2}\beta_k^T \mathbf{S}_k^{-1} \beta_k\right) \frac{1}{\sigma_k^2}$. The objective under the NJ prior presented in Eq. (19) in the main paper, is derived as follows.

$$\begin{aligned} Q_{\text{NJ}}^Y(\boldsymbol{\theta}^Y | \boldsymbol{\theta}^{Y(t)}) &= \mathbb{E}_{\mathbf{z}, \mathbf{S} | \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}^{Y(t)}} [\log f(\mathbf{y} | \mathbf{X}, \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\sigma}^2) + \log \pi(\boldsymbol{\beta} | \mathbf{S}) + \log \pi(\boldsymbol{\sigma}^2)] \\ &= \mathbb{E}_{\mathbf{z} | \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}^{Y(t)}} [\log f(\mathbf{y} | \mathbf{X}, \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\sigma}^2)] + \mathbb{E}_{\mathbf{S} | \boldsymbol{\beta}^{(t)}} [\log \pi(\boldsymbol{\beta} | \mathbf{S})] + \log \pi(\boldsymbol{\sigma}^2) \\ &= \sum_i \mathbb{E}_{\mathbf{z} | \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}^{Y(t)}} [\log f(y_i | \mathbf{x}_i, z_i, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\sigma}^2)] + \mathbb{E}_{\mathbf{S} | \boldsymbol{\beta}^{(t)}} [\log \pi(\boldsymbol{\beta} | \mathbf{S})] + \log \pi(\boldsymbol{\sigma}^2) \\ &= \sum_{i=1}^n \sum_{k=1}^K m_{ki}^{(t)} \left\{ -\frac{1}{2\sigma_k^2} (y_i - \alpha_k - \mathbf{x}_i^T \boldsymbol{\beta}_k)^2 - \frac{1}{2} \log \sigma_k^2 \right\} \\ &\quad + \sum_{k=1}^K \left\{ \mathbb{E}_{\mathbf{S}_k | \boldsymbol{\beta}_k^{(t)}} \left[-\frac{1}{2} \boldsymbol{\beta}_k^T \mathbf{S}_k^{-1} \boldsymbol{\beta}_k \right] \right\} - \sum_{k=1}^K \left\{ \log \sigma_k^2 \right\} \end{aligned} \quad (42)$$

$$\begin{aligned} &= \sum_{k=1}^K \left\{ -\frac{1}{2\sigma_k^2} (\mathbf{y} - \alpha_k \mathbf{1}_n - \mathbf{X} \boldsymbol{\beta}_k)^T \mathbf{M}_k^{(t)} (\mathbf{y} - \alpha_k \mathbf{1}_n - \mathbf{X} \boldsymbol{\beta}_k) - \frac{n_k^{(t)}}{2} \log \sigma_k^2 \right\} \\ &\quad + \sum_{k=1}^K \left\{ -\frac{1}{2} \boldsymbol{\beta}_k^T \mathbb{E}_{\mathbf{S}_k | \boldsymbol{\beta}_k^{(t)}} [\mathbf{S}_k^{-1}] \boldsymbol{\beta}_k \right\} - \sum_{k=1}^K \left\{ \log \sigma_k^2 \right\} \end{aligned} \quad (43)$$

$$\begin{aligned} &= \sum_{k=1}^K \left\{ -\frac{1}{2\sigma_k^2} (\mathbf{y} - \alpha_k \mathbf{1}_n - \mathbf{X} \boldsymbol{\beta}_k)^T \mathbf{M}_k^{(t)} (\mathbf{y} - \alpha_k \mathbf{1}_n - \mathbf{X} \boldsymbol{\beta}_k) - \frac{n_k^{(t)} + 2}{2} \log \sigma_k^2 \right\} \\ &\quad + \sum_{k=1}^K \left\{ -\frac{1}{2} \boldsymbol{\beta}_k^T \mathbf{V}_k^{(t)} \boldsymbol{\beta}_k \right\} \\ &= -\frac{1}{2} \sum_{k=1}^K \left\{ \frac{(\mathbf{y} - \alpha_k \mathbf{1}_n - \mathbf{X} \boldsymbol{\beta}_k)^T \mathbf{M}_k^{(t)} (\mathbf{y} - \alpha_k \mathbf{1}_n - \mathbf{X} \boldsymbol{\beta}_k)}{\sigma_k^2} + \boldsymbol{\beta}_k^T \mathbf{V}_k^{(t)} \boldsymbol{\beta}_k \right. \\ &\quad \left. + (n_k^{(t)} + 2) \log \sigma_k^2 \right\}, \end{aligned} \quad (44)$$

where $m_{ki}^{(t)}$ appearing in (42) is given in (14) in the main paper, $\mathbf{M}_k^{(t)} = \text{diag}(\mathbf{m}_k^{(t)})$ with $\mathbf{m}_k^{(t)} = (m_{k1}^{(t)}, \dots, m_{kn}^{(t)})^T$ and $\mathbf{V}_k^{(t)} = \text{diag}(1/\beta_{k1}^{2(t)}, \dots, 1/\beta_{kp}^{2(t)})$. The transition from (43) to (44) is due to (41).

Appendix D. Details and implementation of the EM

For the graphical lasso optimization in (22) in the main paper we use the efficient R package `glassoFast` (Sustik and Calderhead, 2012). For the lasso optimizations in (26) we use `glmnet` (Friedman et al., 2008b) with penalty equal to $\lambda_k^{(t+1)}/n_k^{(t)}$.

We initialize the algorithm via a simple clustering of the data. For this we use R package `mclust`. Through the resulting group assignments we obtain initial estimates $\boldsymbol{\theta}_k^{X(0)}$ and $\boldsymbol{\theta}_k^{Y(0)}$. In order to initiate EMs from different starting points we add random perturbations to $\boldsymbol{\mu}_k^{(0)}$, $\boldsymbol{\beta}_k^{(0)}$ and $\sigma_k^{2(0)}$ and positive random perturbations to the diagonal elements of $\boldsymbol{\Sigma}_k^{(0)}$. The multiple EMs can be easily run in parallel. As a default option we use ten EM starts.

For the termination of the algorithm we use a combination of two criteria that are commonly used in practice. The first is to simply set a maximum number (T) of EM iterations. Empirical results suggest that the option $T = 20$ is sufficient. The second criterion takes into account the relative change in the objective function in (15); namely, the algorithm is stopped when

$$\left| \frac{Q(\boldsymbol{\theta}, \boldsymbol{\tau}, \boldsymbol{\lambda} | \boldsymbol{\theta}^{(t)}, \boldsymbol{\tau}^{(t)}, \boldsymbol{\lambda}^{(t)})}{Q(\boldsymbol{\theta}, \boldsymbol{\tau}, \boldsymbol{\lambda} | \boldsymbol{\theta}^{(t-1)}, \boldsymbol{\tau}^{(t-1)}, \boldsymbol{\lambda}^{(t-1)})} - 1 \right| \leq \epsilon$$

using as default option $\epsilon = 10^{-6}$. Moreover, the algorithm is stopped, and results are discarded, when the sample size of a certain group becomes prohibitively small for estimation. We define this criterion as a function of total sample size and the number of groups. Specifically, we terminate if $\min_k n_k^{(t)} \leq n/(10K)$.

Appendix E. Proof of Theorem 2

We need to prove that with our model and under the assumptions of Theorem 2, all the hypotheses of Theorem 3 of Meng and Rubin (1993) (Theorem 1) are met.

As discussed throughout section 3.2.2 in the main paper, under the RJM model, the following conditions are sufficient to verify all the hypotheses of Theorem 4.1 except for hypothesis 6: the penalty is continuous, differentiable, such that each of the block maximisation is unique and $L(\boldsymbol{\xi})$ infinite on the border of Ξ . The first three conditions are already assumptions of our Theorem. Hence, it remains only to be shown that $L(\boldsymbol{\xi})$ is infinite on the border of Ξ and that hypothesis (6) is met. Both are very similar properties, we prove both of them together. For this task, we make use of the “penalty lower bound assumption” of our Theorem, recalled in Eq. (45).

$$\text{pen}(\boldsymbol{\xi}) \geq \delta \sum_{k=1}^K (\log \tau_k^{-1} + \|\boldsymbol{\mu}_k\| + \|\boldsymbol{\Omega}_k\| + \log |\boldsymbol{\Omega}_k^{-1}| + f_\lambda(\lambda_k) + \rho_k + \log \rho_k^{-1} + |\chi_k| + \|\boldsymbol{\phi}_k\|) . \quad (45)$$

To begin with, we have:

$$\begin{aligned} L(\boldsymbol{\xi}) = \sum_{i=1}^n \log \sum_{k=1}^K \exp \left(-\frac{1}{2} \left((y_i \rho_k - \chi_k - \mathbf{x}_i^T \boldsymbol{\phi}_k)^2 - 2 \log \rho_k \right. \right. \\ \left. \left. + \|\mathbf{x}_i - \boldsymbol{\mu}_k\|_{\Omega_k}^2 - \log |\Omega_k| \right. \right. \\ \left. \left. - 2 \log \tau_k + (p+1) \log 2\pi + \frac{2}{n} \text{pen}(\boldsymbol{\xi}) \right) \right). \end{aligned}$$

Let

$$f_{i,k}(\boldsymbol{\xi}) = (y_i \rho_k - \chi_k - \mathbf{x}_i^T \boldsymbol{\phi}_k)^2 - 2 \log \rho_k + \|\mathbf{x}_i - \boldsymbol{\mu}_k\|_{\Omega_k}^2 - \log |\Omega_k| - 2 \log \tau_k + \frac{2}{n} \text{pen}(\boldsymbol{\xi}).$$

Such that

$$L(\boldsymbol{\xi}) = -\frac{n(p+1) \log 2\pi}{2} + \sum_{i=1}^n \log \sum_{k=1}^K \exp \left(-\frac{1}{2} f_{i,k}(\boldsymbol{\xi}) \right). \quad (46)$$

From Eq. (45) and the fact that $(y_i \rho_k - \chi_k - \mathbf{x}_i^T \boldsymbol{\phi}_k)^2 \geq 0$ and $\|\mathbf{x}_i - \boldsymbol{\mu}_k\|_{\Omega_k}^2 \geq 0$, we have:

$$\begin{aligned} f_{i,k}(\boldsymbol{\xi}) &\geq \frac{2}{n} \delta \sum_{l=1}^K \left(-(1 + \frac{n}{\delta} \mathbb{1}_{l=k}) \log \tau_l + \|\boldsymbol{\mu}_l\| + \|\Omega_l\| - (1 + \frac{n}{2\delta} \mathbb{1}_{l=k}) \log |\Omega_l| \right. \\ &\quad \left. + f_\lambda(\lambda_l) + \rho_l - (1 + \frac{n}{\delta} \mathbb{1}_{l=k}) \log \rho_l + |\chi_l| + \|\boldsymbol{\phi}_l\| \right) \\ &= \frac{2}{n} \delta \sum_{l=1}^K \left(f_{k,l}^\tau(\tau_l) + f_{k,l}^\mu(\boldsymbol{\mu}_l) + f_{k,l}^\Omega(\Omega_l) + f_{k,l}^\lambda(\lambda_l) + f_{k,l}^\rho(\rho_l) + f_{k,l}^\chi(\chi_l) + f_{k,l}^\phi(\boldsymbol{\phi}_l) \right). \end{aligned} \quad (47)$$

Where:

$$\begin{aligned} f_{k,l}^\tau(\tau_l) &:= -(1 + \frac{n}{\delta} \mathbb{1}_{l=k}) \log \tau_l, \\ f_{k,l}^\mu(\boldsymbol{\mu}_l) &:= \|\boldsymbol{\mu}_l\|, \\ f_{k,l}^\Omega(\Omega_l) &:= \|\Omega_l\| - (1 + \frac{n}{2\delta} \mathbb{1}_{l=k}) \log |\Omega_l|, \\ f_{k,l}^\lambda(\lambda_l) &:= f_\lambda(\lambda_l), \\ f_{k,l}^\rho(\rho_l) &:= \rho_l - (1 + \frac{n}{\delta} \mathbb{1}_{l=k}) \log \rho_l, \\ f_{k,l}^\chi(\chi_l) &:= |\chi_l|, \\ f_{k,l}^\phi(\boldsymbol{\phi}_l) &:= \|\boldsymbol{\phi}_l\|. \end{aligned} \quad (48)$$

The dependency on k, l is denoted in the indices of all these functions for the sake of uniformity, although only $f_{k,l}^\tau$, $f_{k,l}^\Omega$ and $f_{k,l}^\rho$ actually depend on k and l . We recall that, with $a > 0$, the function $x \mapsto x - a \log x$, is lower bounded on \mathbb{R}_+^* , and converges towards $+\infty$ both when $x \rightarrow 0$ and when $x \rightarrow +\infty$. To analyse $f_{k,l}^\Omega$, it is convenient to consider the nuclear norm for $\|\Omega_k\|$ and rewrite the whole as: $f_{k,l}^\Omega(\Omega_l) = \sum_j \psi_{l,j} - (1 + \frac{n}{2\delta} \mathbb{1}_{l=k}) \log \psi_{l,j}$, with $\{\psi_{l,j}\}_{j=1}^p$ the eigenvalues of Ω_l .

With these observations at hand, note that all the functions in (48) can be lower bounded by the same constant $c > -\infty$, valid for all values of k and l . They also all converge towards $+\infty$ on the boundary of their respective sets of definition.

For $m > 0$, we define Ξ_m as the compact subset of Ξ such that $\xi \in \Xi_m \iff \xi \in \Xi$ and $\forall k = 1, \dots, K$:

$$\begin{aligned} \tau_k &\geq \frac{1}{m}, \\ \|\boldsymbol{\mu}_k\| &\leq m, \\ \psi_{\max}(\boldsymbol{\Omega}_k) &\leq m, \\ \psi_{\min}(\boldsymbol{\Omega}_k) &\geq \frac{1}{m}, \\ \frac{1}{m} &\leq \lambda_k \leq m, \\ \frac{1}{m} &\leq \rho_k \leq m, \\ |\chi_k| &\leq m, \\ \|\boldsymbol{\phi}_k\| &\leq m. \end{aligned} \tag{49}$$

It is clear that $\forall m > 0$, $\Xi_m \subseteq \Xi^\circ$.

With all these objects defined, we can finish the proof. For any real number $A > -\infty$, let us show that there exists $M > 0$ such that $\forall \xi \in \Xi \setminus \Xi_M$, $L(\xi) < A$. First consider the following: for any real number $B > -\infty$, there exists a $m_B > 0$ such that, for all k and l :

$$\begin{aligned} &\text{if } \tau_l < \frac{1}{m_B}, \text{ then } f_{k,l}^{\boldsymbol{\tau}}(\tau_l) > B, \\ &\text{if } \|\boldsymbol{\mu}_l\| > m_B, \text{ then } f_{k,l}^{\boldsymbol{\mu}}(\boldsymbol{\mu}_l) > B, \\ &\text{if } \psi_{\max}(\boldsymbol{\Omega}_l) > m_B, \text{ then } f_{k,l}^{\boldsymbol{\Omega}}(\boldsymbol{\Omega}_l) > B, \\ &\text{if } \psi_{\min}(\boldsymbol{\Omega}_l) < \frac{1}{m_B}, \text{ then } f_{k,l}^{\boldsymbol{\Omega}}(\boldsymbol{\Omega}_l) > B, \\ &\text{if } \lambda_l < \frac{1}{m_B} \text{ or } \lambda_l > m_B, \text{ then } f_{k,l}^{\boldsymbol{\lambda}}(\lambda_l) > B, \\ &\text{if } \rho_l < \frac{1}{m_B} \text{ or } \rho_l > m_B, \text{ then } f_{k,l}^{\boldsymbol{\rho}}(\rho_l) > B, \\ &\text{if } |\chi_l| > m_B, \text{ then } f_{k,l}^{\boldsymbol{\chi}}(\chi_l) > B, \\ &\text{if } \|\boldsymbol{\phi}_l\| > m_B, \text{ then } f_{k,l}^{\boldsymbol{\phi}}(\boldsymbol{\phi}_l) > B. \end{aligned} \tag{50}$$

If $\xi \in \Xi \setminus \Xi_{m_B}$, then by definition of the sets Ξ_m (49), there exist at least one $l \in \{1, \dots, K\}$ such that at least one of the above scenarios is realised. By injecting the resulting lower bound into the inequality (47), we get:

$$\forall k, \quad f_{i,k}(\boldsymbol{\xi}) > \frac{2}{n} \delta (B + (7K - 1)c).$$

Then, from Eq. (46):

$$\begin{aligned}
L(\boldsymbol{\xi}) &= -\frac{n(p+1)\log 2\pi}{2} + \sum_{i=1}^n \log \sum_{k=1}^K \exp\left(-\frac{1}{2}f_{i,k}(\boldsymbol{\xi})\right) \\
&< -\frac{n(p+1)\log 2\pi}{2} + \sum_{i=1}^n \log \sum_{k=1}^K \exp\left(-\frac{1}{n}\delta(B+(7K-1)c)\right) \\
&= -\frac{n(p+1)\log 2\pi}{2} + n \log K - \delta(B+(7K-1)c).
\end{aligned}$$

Since $\delta > 0$, then there exists $B_A > 0$ such that for all $B \geq B_A$:

$$-\frac{n(p+1)\log 2\pi}{2} + n \log K - \delta(B+(7K-1)c) < A.$$

As a consequence, $M := m_{B_A}$ is such that $\forall \boldsymbol{\xi} \in \Xi \setminus \Xi_M$, $L(\boldsymbol{\xi}) < A$. In other words $\{\boldsymbol{\xi} \in \Xi | L(\boldsymbol{\xi}) \geq A\} \subseteq \Xi_M$.

We have proven that for any $A > -\infty$, there exists $M > 0$ such that $\{\boldsymbol{\xi} \in \Xi | L(\boldsymbol{\xi}) \geq A\} \subseteq \Xi_M$. Since the Ξ_m are compacts, this means that the closed set $\{\boldsymbol{\xi} \in \Xi | L(\boldsymbol{\xi}) \geq A\}$ is also a compact, hence hypothesis (6) of Wu (1983) is verified. Moreover, since $\Xi_m \subseteq \Xi^\circ$, this means that the log-likelihood goes to $-\infty$ on the border of Ξ . Hence no EM sequence will take values on the border, hence we can safely consider that $\Xi = \Xi^\circ$. With these last two hypotheses verified, we can apply Theorem 3 of Meng and Rubin (1993) and benefit from the convergence guarantees.

Appendix F. Further results from Section 5.1

Table 3: First simulation. Intercept and slope parameter values for the two groups under the three cases illustrated in Figure 3 in the main paper.

Case	Group	Intercept	Slope
A	1	0	1
	2	0	-1
B	1	0	1
	2	1	1
C	1	0	1
	2	0	1

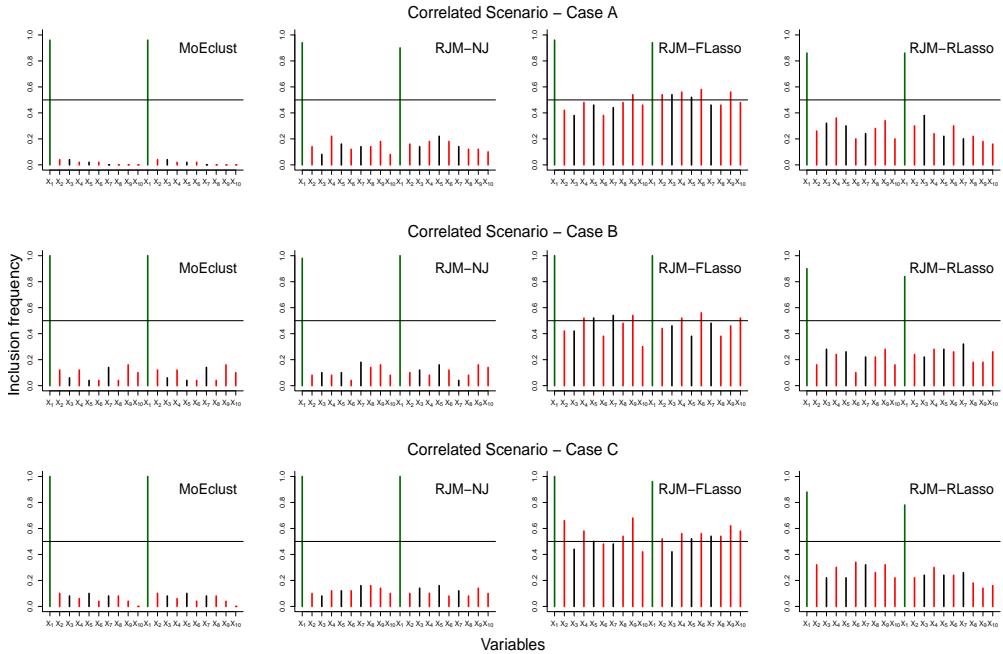


Figure 12: First simulation, correlated scenario. Variable inclusion frequencies (under 50 repetitions) for signal variables (in green), correlated noise variables (in black) and uncorrelated noise variables (in red) for regression cases A, B and C. Horizontal black lines correspond to a frequency of 0.5.

REGULARIZED JOINT MIXTURES

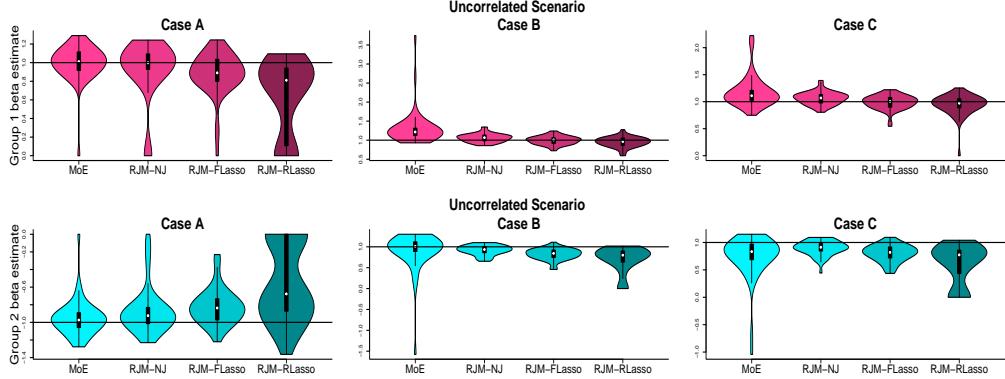


Figure 13: First simulation, uncorrelated scenario. Violin plots of MoEclust and RJM slope estimates (from 50 repetitions) for cases A, B and C. Horizontal black lines correspond to the true slopes.

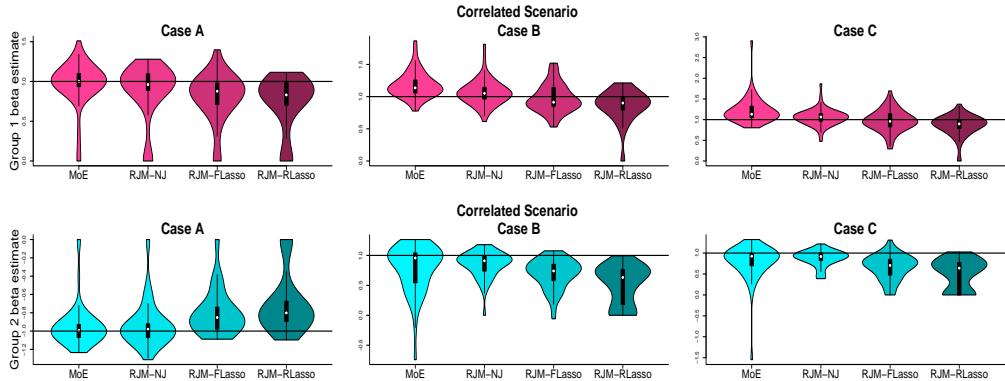


Figure 14: First simulation, correlated scenario. Violin plots of MoEclust and RJM slope estimates (from 50 repetitions) for cases A, B and C. Horizontal black lines correspond to the true slopes.

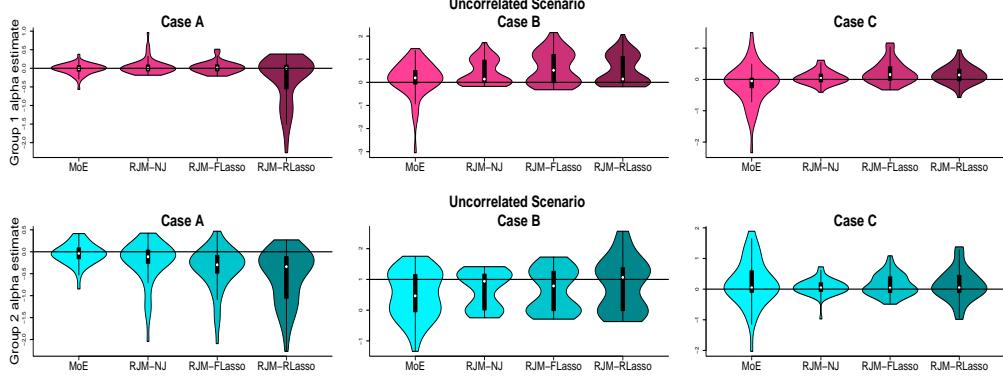


Figure 15: First simulation, uncorrelated scenario. Violin plots of MoEclust and RJM intercept estimates (from 50 repetitions) for cases A, B and C. Horizontal black lines correspond to the real intercepts.

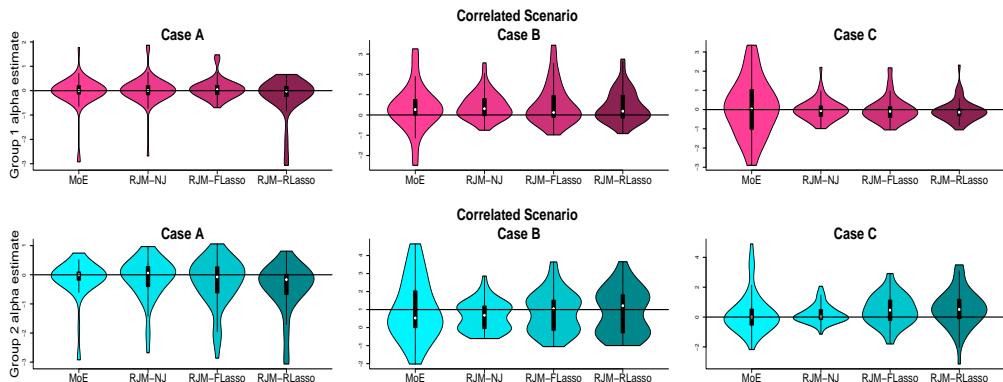


Figure 16: First simulation, correlated scenario. Violin plots of MoEclust and RJM intercept estimates (from 50 repetitions) for cases A, B and C. Horizontal black lines correspond to the real intercepts.

Appendix G. Further results from Section 5.2

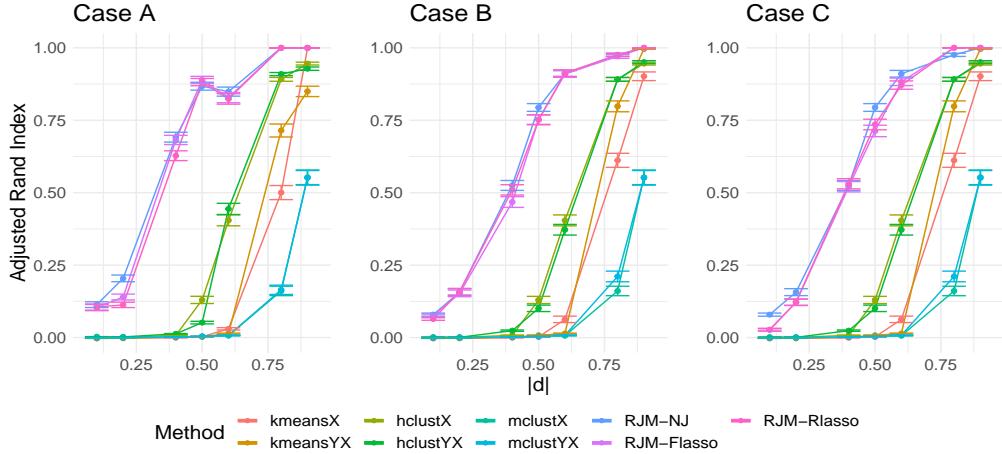


Figure 17: Second simulation, $p = 100$, group assignment. Average adjusted Rand Index as a function of the absolute distance ($|d|$) of the group-wise covariate means, for cases A (left), B (center) and C (right). [Error bars indicate standard errors from 20 repetitions.]

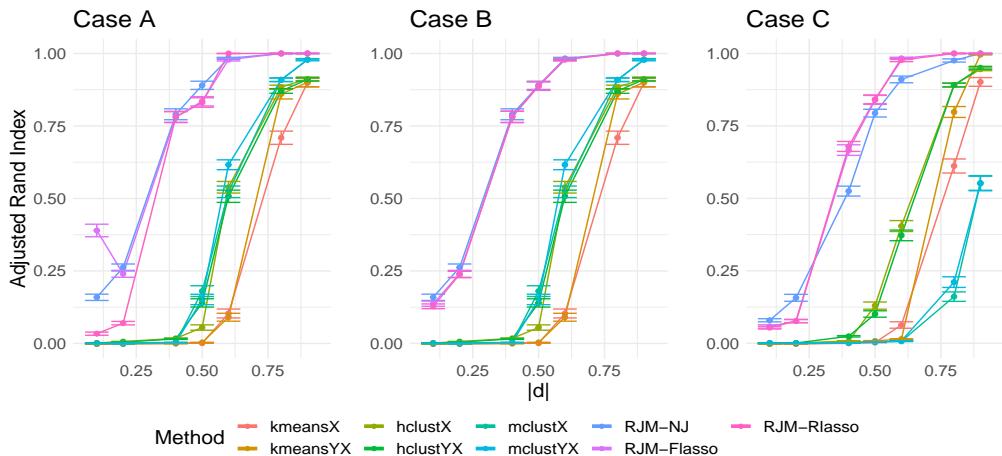


Figure 18: Second simulation, $p = 250$, group assignment. Average adjusted Rand Index as a function of the absolute distance ($|d|$) of the group-wise covariate means, for cases A (left), B (center) and C (right). [Error bars indicate standard errors from 20 repetitions.]

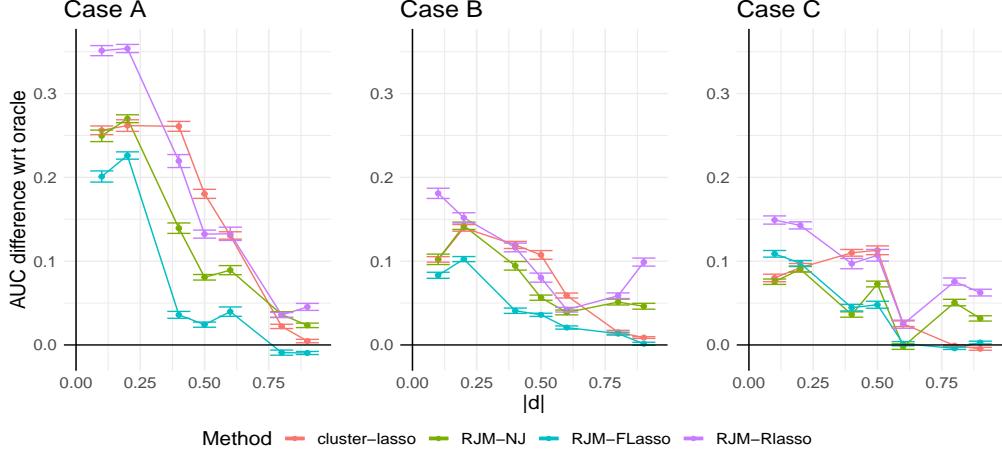


Figure 19: Second simulation, $p = 100$, variable selection. AUC loss from oracle-lasso as a function of the absolute distance ($|d|$) of the group-wise covariate means, for cases A (left), B (center) and C (right). [Error bars indicate standard errors from 20 repetitions.]

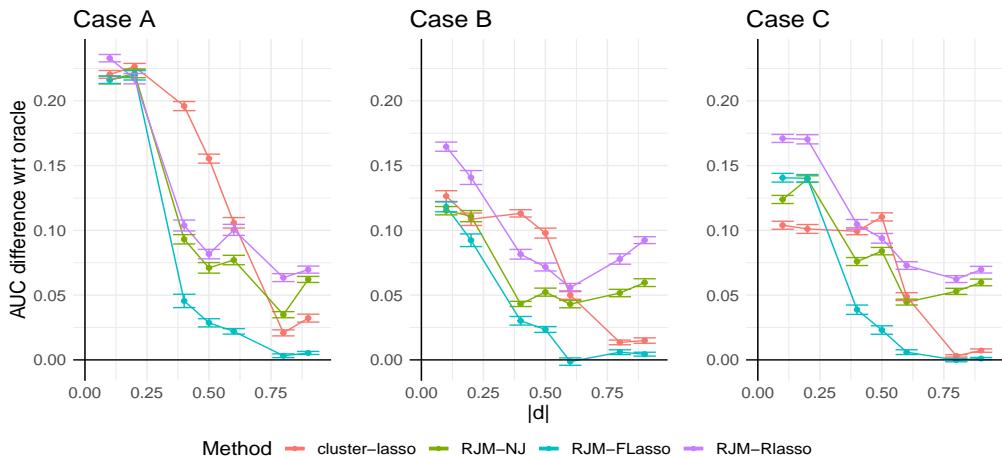


Figure 20: Second simulation, $p = 250$, variable selection. AUC loss from oracle-lasso as a function of the absolute distance ($|d|$) of the group-wise covariate means, for cases A (left), B (center) and C (right). [Error bars indicate standard errors from 20 repetitions.]

REGULARIZED JOINT MIXTURES

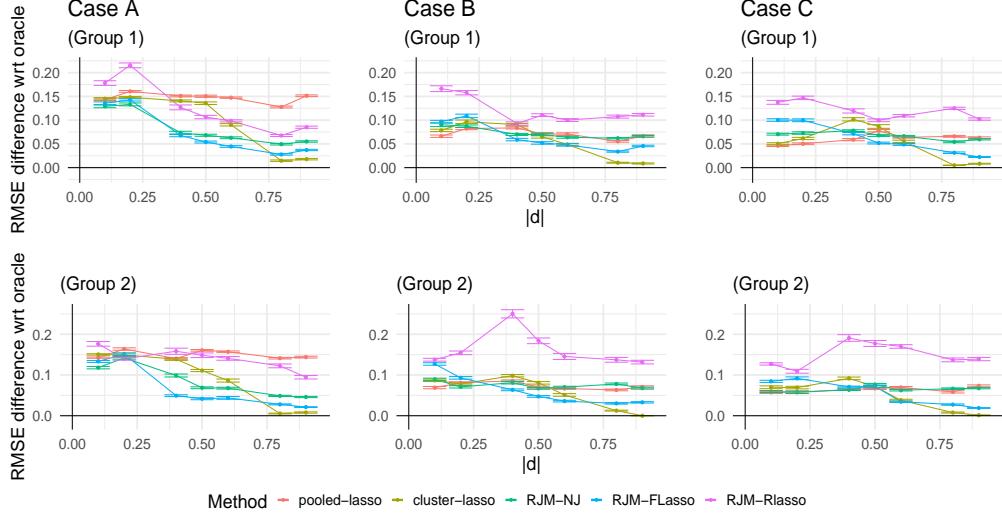


Figure 21: Second simulation, $p = 100$, regression coefficients estimation. Increase in RMSE relative to the oracle-lasso as a function of the absolute distance ($|d|$) of the group-wise covariate means, under group one (top) and group two (bottom), and for cases A (left), B (center) and C (right). [Error bars indicate standard errors from 20 repetitions.]

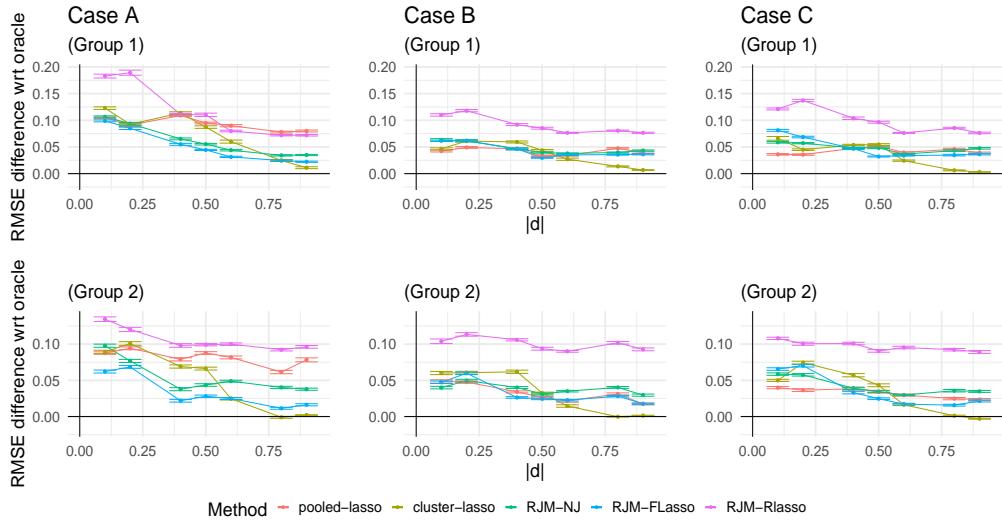


Figure 22: Second simulation, $p = 250$, regression coefficients estimation. Increase in RMSE relative to the oracle-lasso as a function of the absolute distance ($|d|$) of the group-wise covariate means, under group one (top) and group two (bottom), and for cases A (left), B (center) and C (right). [Error bars indicate standard errors from 20 repetitions.]

Appendix H. Selection of number of groups in Section 5.2

Here we consider all four cancer types (BRCA, KIRC, LUAD, THCA) and provide some results on cluster selection under unknown number of groups using the predictive approach described in Section 4. We consider three simulation settings where the respective true number of groups is $g^* = 2$ (using the BRCA and KIRC cancer types), $g^* = 3$ (BRCA, KIRC, LUAD) and $g^* = 4$ (further including THCA). For each setting we fit RJM models with two, three and four components. The simulations are along the lines of Section 5.2 considering Case A of Table 1 for $p = 100$. The conditions outlined in Table 1 for the β_j^* 's at the common locations are used again for $g^* \in \{3, 4\}$. Here we use the real group-sample proportions as they occur in the TCGA data set and assume that sample size grows with number of groups (for the simulations to be on an equal basis). Specifically, we set $n = 250 \times g^*$. The resulting group sample sizes for $g^* \in \{2, 3, 4\}$ are $(n_1 = 335, n_2 = 165)$, $(n_1 = 382, n_2 = 188, n_3 = 180)$ and $(n_1 = 410, n_2 = 200, n_3 = 200, n_4 = 190)$, respectively. We use 80% of the samples for training and 20% for testing.

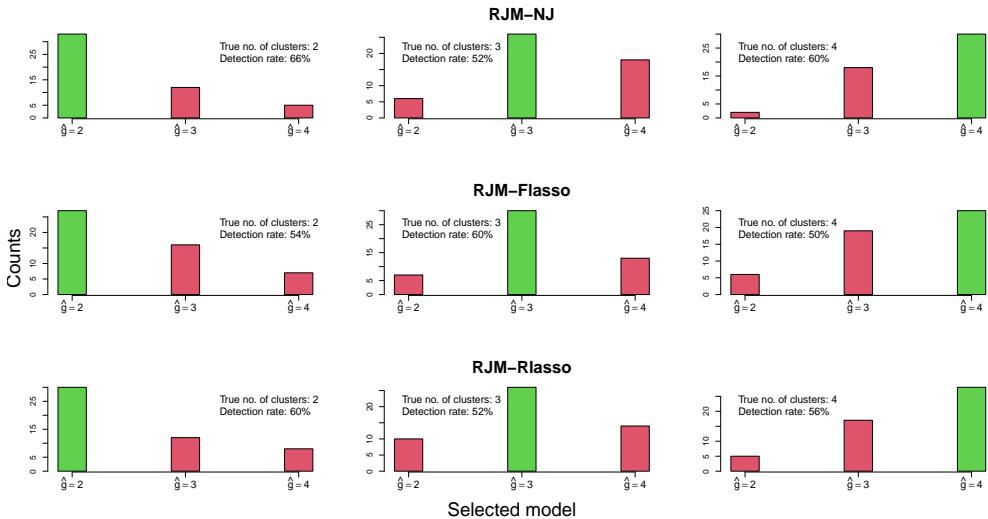


Figure 23: Second simulation, cluster selection. Barplots of selected clusters (50 repetitions) using the predictive approach in Section 4. Correct identification is highlighted in green; true number of clusters and detection rate of the correct model is annotated in each panel.

Figure 23 shows barplots of the selected number of clusters resulting from 50 repetitions of the simulations. As seen, the correct model is selected in majority under all cases. The detection rate of the correct model is also annotated in each panel of Figure 23. Here, RJM-NJ is slightly better with average overall detection rate of 59%, while RLasso and Flasso have 56% and 55%, respectively.

Appendix I. Selection of number of groups in Section 5.3

Here we present additional results concerning performance including a model selection step. The setting is as in Section 5.3 in the main text except that sample size is equal to 200. In particular, treating in turn each of the genes as response, with all others considered as features. A model selection step is included over the number of clusters, selecting between models with $g \in \{2, 3, 4\}$ components based on BIC. Here we compare with GMMs (`mclust`) and an MoE implementation from package `flexmix` (Grün and Leisch, 2008). Under the latter method we use elastic-net regularization (Zou and Hastie, 2005) in the expert networks and intercept-only multinomial gating functions (the latter is better than allowing 99 predictors to enter the gating functions in the absence of regularization for this part of the model). We note that results from `MoEclust` are not presented here as this method (based on incremental forward model search) never selected the correct model with four clusters. Results from the 100 attempts are summarized in Table 4.

Table 4: TCGA data application, performance including a model selection step. Number of times, out of 100 applications to the TCGA data, that the methods selected two, three and four clusters based on BIC. Each time a different gene expression was used as response variable with the predictor matrix containing the remaining 99 gene expressions. The correct number of clusters is four corresponding to the four cancer types included in the dataset.

Methods and cluster selection			
Estimated number of clusters	$\hat{g} = 2$	$\hat{g} = 3$	$\hat{g} = 4$
GMM (<code>mclust</code>)	5	91	4
MoE (<code>flexmix</code>)	18	67	15
RJM-NJ	17	18	65
RJM-FL	16	17	67
RJM-RL	17	19	64

References

- H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
- A. Anandkumar, D. J. Hsu, F. Huang, and S. M. Kakade. Learning mixtures of tree graphical models. In *NIPS*, pages 1061–1069, 2012.
- K. Bae and B. Mallick. Gene selection using a two-level hierarchical Bayesian model. *Bioinformatics*, 20(18):3423–3430, 2004.
- J. D. Banfield and A. E. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49(3):803–821, 1993.
- C. Carvalho, N. Polson, and J. Scott. The horseshoe estimator for sparse signal. *Biometrika*, 97(2):465–480, 2010.

- G. Celeux and G. Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793, 1995.
- F. Chamroukhi and B. T. Huynh. Regularized maximum-likelihood estimation of mixture-of-experts for regression and clustering. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018.
- U. J. Dang and P. D. McNicholas. Families of Parsimonious Finite Mixtures of Regression Models. In I. Morlini, T. Minerva, and M. Vichi, eds., *Advances in Statistical Models for Data Analysis*, pages 73–84. Springer International Publishing, 2015.
- U. J. Dang, A. Punzo, P. D. McNicholas, S. Ingrassia, and R. P. Browne. Multivariate response and parsimony for Gaussian cluster-weighted models. *Journal of Classification*, 34(1):4–34, 2017.
- C. Dayton and G. Macready. Concomitant-variable latent-class models. *Journal of the American Statistical Association*, 83(401):173–178, 1988.
- L. D. D. Desboulets. A review on variable selection in regression analysis. *Econometrics*, 6(4), 2018.
- D. L. Donoho and J. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.
- M. A. T. Figueiredo. Wavelet-based image estimation: An empirical Bayes approach using Jeffreys' noninformative prior. *IEEE Transactions on Image Processing*, 10(9):1322–1331, 2001.
- M. A. T. Figueiredo. Adaptive sparseness for supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1150–1159, 2003.
- M. Fop, T.B. Murphy, and L. Scrucca. Model-based clustering with sparse covariance matrices. *Statistics and Computing*, 29(4):791–819, 2019.
- C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.
- J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008a.
- J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2008b.
- S. Früwirth-Schnatter. *Finite Mixture and Markov Switching Models*. Springer, Heidelberg, 2005.
- J. Griffin and P. Brown. Alternative prior distributions for variable selection with very many more variables than observations. Technical report, University of Warwick, 2005.
- B. Grün and F. Leisch. FlexMix version 2: Finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, 28(4):1–35, 2008.

- Q. Gu and J. Han. Clustered support vector machines. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 307–315, 2013.
- C. Heinze-Deml and N. Meinshausen. Conditional variance penalties and domain shift robustness. *Machine Learning*, 110:303–348, 2021.
- S. Ingrassia, S. C. Minotti, and G. Vittadini. Local statistical modeling via a cluster-weighted approach with elliptical distributions. *Journal of Classification*, 29(3):363–401, 2012.
- S. Ingrassia, A. Punzo, G. Vittadini, and S. C. Minotti. The generalized linear mixed cluster-weighted model. *Journal of Classification*, 32(1):85–113, 2015.
- R. A. Jacobs. Bias/variance analyses of mixtures-of-experts architectures. *Neural Computation*, 9(2):369–383, 1997.
- R. A. Jacobs, M.I. Jordan, S.I. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.
- I. T. Jolliffe. A note on the use of principal components in regression. *Journal of the Royal Statistical Society, Series C*, 31(3):300–303, 1982.
- M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994.
- A. Khalili. New estimation and feature selection methods in mixture-of-experts models. *Canadian Journal of Statistics*, 38(4):519–539, 2010.
- A. Khalili and C. Jiahua. Variable selection in finite mixture of regression models. *Journal of the American Statistical Association*, 102(479):1025–1038, 2007.
- A. Khalili and S. Lin. Regularization in finite mixture of regression models with diverging number of parameters. *Biometrics*, 69(2):436–446, 2013.
- S. Liverani, D. I. Hastie, L. Azizi, M. Papathomas, and S. Richardson. PReMiuM: An R package for profile regression mixture models using Dirichlet processes. *Journal of Statistical Software*, 64(7):1–30, 2015.
- G.J. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley, New York, USA, 2000.
- D. P. McNicholas and T. B. Murphy. Parsimonious Gaussian mixture models. *Statistics and Computing*, 18(3):285–296, 2008.
- N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.
- X.-L. Meng and D. B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

- J. Molitor, M. Papathomas, M. Jerrett, and S. Richardson. Bayesian profile regression with an application to the National survey of children's health. *Biostatistics*, 11(3):484–498, 2010.
- K. Murphy and T. B. Murphy. Gaussian parsimonious clustering models with covariates and a noise component. *Advances in Data Analysis and Classification*, 14(2):293–325, 2020.
- T. Park and G. Casella. The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.
- N. G. Polson and J. G. Scott. Shrink Globally, Act Locally: Sparse Bayesian Regularization and Prediction. In J.M. Bernardo, M.J. Bayarri, J.O. Berger, A.P. David, D. Heckerman, A.F.M. Smith and M. West, eds., *Bayesian Statistics*, Vol. 9, pages 501–538. Oxford University Press, 2010.
- N. G. Polson and J. G. Scott. On the half-cauchy prior for a global scale parameter. *Bayesian Analysis*, 7(4):887–902, 2012.
- A. Punzo and S. Ingassia. Clustering bivariate mixed-type data via the cluster-weighted model. *Computational Statistics*, 31(3):989–1013, 2016.
- B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do CIFAR-10 classifiers generalize to CIFAR-10? *arXiv:1806.00451 [cs.LG]*, 2018.
- G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.
- L. Scrucca, M. Fop, T. B. Murphy, and Raftery A. E. mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1):289–317, 2016.
- N. Städler and S. Mukherjee. Penalized estimation in high-dimensional hidden Markov models with state-specific graphical models. *The Annals of Applied Statistics*, 7(4):2157–2179, 2013.
- N. Städler, P. Bühlmann, and S. van de Geer. ℓ_1 -penalization for mixture regression models. *Test*, 19:209–256, 2010.
- N. Städler, F. Dondelinger, S. M. Hill, R. Akbani, Y. Lu, G. B. Mills, and S. Mukherjee. Molecular heterogeneity at the network level: high-dimensional testing, clustering and a TCGA case study. *Bioinformatics*, 33(18):2890–2896, 2017.
- S. Subedi, A. Punzo, S. Ingassia, and P.D. McNicholas. Clustering and classification via cluster-weighted factor analyzers. *Advances in Data Analysis and Classification*, 7(1):5–40, 2013.
- M. A Sustik and B Calderhead. GLASSOFAST: An efficient GLASSO implementation. Technical Report TR-12-29:1-3, UTCS, 2012.
- B. Taschler, F. Dondelinger, and S. Mukherjee. Model-based clustering in very high dimensions via adaptive projections. *arXiv:1902.08472v1 [stat.ML]*, 2019.

REGULARIZED JOINT MIXTURES

- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58(1):267–288, 1996.
- S. van Erp, D. L. Oberski, and J. Mulder. Shrinkage priors for Bayesian penalized regression. *Journal of Mathematical Psychology*, 89:31–50, 2019.
- H. Wang. Bayesian graphical lasso models and efficient posterior computation. *Bayesian Analysis*, 7(4):867–886, 2012.
- C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.
- M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.
- H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.
- H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, 67(2):301–320, 2005.

Brendan Murphy

Material list:

Menezes T.P., Murphy T.B. and Fop M. (2024) A model-based approach to enhance individual matching across different databases by using household information. WG slides.

Menezes T.P., Murphy T.B. and Fop M. (2024) Hausdorff distance-based record linkage for improved matching of households and individuals in different databases. arXiv:2404.05566.

A Model-Based Approach to Enhance Individual Matching Across Different Databases by Using Household Information

Thais Pacheco Menezes, Brendan Murphy and Michael Fop

Department of Statistical Sciences Seminar, University College London.



1 / 37

Example 1: Historical Census

- ▶ The 1901 and 1911 censuses of Ireland have been digitised and are publicly available.
- ▶ Can we match individuals/households across censuses?
- ▶ What changed with individuals/households across this period?

2 / 37

Example: Murphy Family 1901

- This is my family entry in the 1901 census.

CENSUS OF IRELAND, 1901.

No. on Form B. 4

FORM A.

(Two Examples of the mode of filling up this Table are given on the other side.)

Christian Name, or Surname.	Surnames.	NAME and SURNAME. RELATION to Head of Family.	RELIGIOUS PROFESSION.	EDUCATION.	AGE.	SEX.	RANK, PROFESSION, OR OCCUPATION.	MARRIAGE.	WHERE BORN.	IRISH LANGUAGE.	If Dead and Name Died only; or Abstain or (dead) or Lunatic.
1 Denis	Murphy	Head of Family.	Roman Catholic	Born at Cork	Wife	60	m	Farmer	Married	To Cork	Irish & English
2 Edith	Murphy	wife	Roman Catholic	Reads & writes	50	f	-	-	Married	To Cork	Irish & English
3 Matthew	Murphy	son	Roman Catholic	Reads & writes	25	m	Farmers son	Married	To Cork	Irish & English	
4 Mary	Murphy	daughter	Roman Catholic	Reads & writes	25	f	-	-	Married	To Cork	Irish & English
5 Mary	Murphy	daughter	Roman Catholic	Reads & writes	25	f	Former's Daughter	Atmarrige	To Cork	Irish & English	
6 Eddie	Murphy	niece	Roman Catholic	Reads & writes	35	f	-	-	Atmarrige	To Cork	Irish & English
7 Michael	Murphy	son	Roman Catholic	Reads & writes	20	m	Farmers son	not married	To Cork	Irish & English	
8											
9											
10											
11											
12											
13											
14											
15											

I hereby certify, as required by the Act 63 Vict., cap. 6, &c. (1), that the foregoing Return is correct, according to the best of my knowledge and belief.

Patrick Murphy, Esq.
(Signature of Enumerator.)

I believe the foregoing to be a true Return.

Denis Murphy
(Signature of Head of Family.)

3 / 37

Example: Murphy Family 1911

- This is my family entry in the 1911 census.

CENSUS OF IRELAND, 1911.

No. on Form B. 6

FORM A.

(Two Examples of the mode of filling up this Table are given on the other side.)

Christian Name, or Surname.	Surnames.	NAME and SURNAME. RELATION to Head of Family.	RELIGIOUS PROFESSION.	EDUCATION.	AGE (See Births)	SEX.	RANK, PROFESSION, OR OCCUPATION.	PARTICULARS AS TO MARRIAGE.	WHERE BORN.	IRISH LANGUAGE.	If Dead and Name Died only; or Abstain or (dead); or Lunatic.
1 Mattie	Murphy	Head of Family.	Roman Catholic	Reads & writes	47	f	Farmer	Married	To Cork	Irish & English	
2 Mary	Murphy	Wife	Roman Catholic	Reads & writes	47	-	-	Married	To Cork	-	
3 Maria	Murphy	son	Roman Catholic	Reads & writes	9	m	Scholar	Single	To Cork	-	
4 Thomas	Murphy	Wife	Roman Catholic	Cannot read	3	-	-	Single	To Cork	-	
5 Mattie	Murphy	Daughter	Roman Catholic	Cannot read	11	m	-	Single	To Cork	-	
6 Mattie	Murphy	daughter	Roman Catholic	Reads & writes	23	f	Domestic Servant	Single	To Cork	-	
7 Eddie	Murphy	Daughter	Roman Catholic	Reads & writes	23	f	Domestic Servant	Single	To Cork	-	
8											
9											
10											
11											
12											
13											
14											
15											

I hereby certify, as required by the Act 10 Edw. VII, and 1 Geo. V., cap. 11, that the foregoing Return is correct, according to the best of my knowledge and belief.

Mattie Murphy
(Signature of Enumerator.)

I believe the foregoing to be a true Return.

Mattie Murphy
(Signature of Head of Family.)

4 / 37

Research Question

- ▶ Most record linkage methods try to match individuals directly.
 - ▶ Should we include household information when trying to match individuals from different databases?

Main Proposal

The main proposal of our work is to show that any group information available in the data (such as households) can help in the process of matching individuals.

5 / 37

Example: William Sealy Gosset 1901

- ▶ This is William Sealy Gosset's entry in 1901.

Example: William Sealy Gosset 1911

- ▶ This is William Sealy Gosset's entry in 1911.

- ▶ His household structure had changed considerably in the time period.

7 / 37

Example 2: Italian Survey of Household Income and Wealth

- ▶ The Italian Survey of Household Income and Wealth (SHIW) is conducted by the Bank of Italy.
 - ▶ The survey has been conducted since 1960.
 - ▶ We leverage data from the Italian survey conducted in 2014, 2016, and 2020.
 - ▶ Part of the sample comprises households that were interviewed in the previous study.

Advantage

The primary advantage of this database is the availability of true match status.

Variables

Variable	Description	Range
SEX	Individual's sex	2 levels
CIT	Indicator of whether or not it is an Italian citizen	2 levels
ANASC	Year of birth	Discrete
STUDIO	Educational qualification	8 levels
NASCREG	Region of birth	21 levels
NACE	Sector of activity of the company that the individual works	22 levels
IREG	Region of residence	20 levels
QUAL	Employment status	7 levels

9 / 37

Example: Households & Individuals

- Consider an individual in a household in 2014.

YEAR	SEX	ANASC	CIT	STUDIO	NACE	NASCREG	IREG	QUAL
2014	Male	1958	Yes	Licenza Media Inferiore	Wholesale and retail trade	Emilia Romagna	Emilia Romagna	sole proprietor/member of the arts or professions
2014	Female	1964	Yes	Diploma Professionale (3 anni)	Wholesale and retail trade	Emilia Romagna	Emilia Romagna	sole proprietor/member of the arts or professions
2014	Male	1984	Yes	Licenza Media Inferiore	Wholesale and retail trade	Emilia Romagna	Emilia Romagna	sole proprietor/member of the arts or professions
2014	Male	1987	Yes	Diploma Media Superiore	Wholesale and retail trade	Emilia Romagna	Emilia Romagna	sole proprietor/member of the arts or professions

- The household information helps with matching her to the correct individual in 2016.

YEAR	SEX	ANASC	CIT	STUDIO	NACE	NASCREG	IREG	QUAL
2016	Male	1958	Yes	Licenza Media Inferiore	Wholesale and retail trade	Emilia Romagna	Emilia Romagna	sole proprietor/member of the arts or professions
2016	Female	1964	Yes	Diploma Professionale (3 anni)	Wholesale and retail trade	Emilia Romagna	Emilia Romagna	sole proprietor/member of the arts or professions
2016	Male	1984	Yes	Licenza Media Inferiore	Wholesale and retail trade	Emilia Romagna	Emilia Romagna	sole proprietor/member of the arts or professions
2016	Male	1987	Yes	Diploma Media Superiore	Wholesale and retail trade	Emilia Romagna	Emilia Romagna	sole proprietor/member of the arts or professions

YEAR	SEX	ANASC	CIT	STUDIO	NACE	NASCREG	IREG	QUAL
2016	Female	1962	Yes	Diploma Media Superiore	Pensioner or Not employed	Emilia Romagna	Emilia Romagna	other not employed
2016	Male	1962	Yes	Licenza Media Inferiore	Wholesale and retail trade	Emilia Romagna	Emilia Romagna	sole proprietor/member of the arts or professions
2016	Female	1994	Yes	Diploma Media Superiore	Wholesale and retail trade	Emilia Romagna	Emilia Romagna	sole proprietor/member of the arts or professions

10 / 37

Motivation: Unsupervised

- ▶ We want to establish if household information can enhance the accuracy of individual matching.
- ▶ We previously developed a supervised learning approach where some known matches (household/individual) can be exploited. In the supervised model, we found that the household information was useful (Menezes, Murphy and Fop, 2024).

Motivation

In real-life applications, the true match labels are not available. Therefore, an unsupervised learning approach is needed.

11 / 37

Focus on Italian SHIW Data

12 / 37

Some Notation

- ▶ H_{st} = indicator that households \mathcal{H}_s and \mathcal{H}_t are a match;
- ▶ Z_{mn} = indicator that individual m from household \mathcal{H}_s matches individual n from household \mathcal{H}_t ;
- ▶ S = Number of households in file A;
- ▶ T = Number of households in file B;
- ▶ K = total number of features considered;
- ▶ M = Number of individuals in file A;
- ▶ N = Number of individuals in file B;

13 / 37

Matching Matrices

The indicator matrices (H and Z) have a very constrained structure.

Each row and column of the matrix has at most a single one, and all other entries are zero.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

14 / 37

Matching Matrices 2

The number of possible matrices grows rapidly in the number of rows and columns.¹

TABLE 1. The number $A_{n,m,1}$ of $n \times m$ binary matrices with at most one 1 in each row or column.

n	1	2	3	4	5	6	7	8	9
1	2								
2	3	7							
3	4	13	34						
4	5	21	73	209					
5	6	31	136	501	1546				
6	7	43	229	1045	4051	13327			
7	8	57	358	1961	9276	37633	130922		
8	9	73	529	3393	19081	93289	394353	1441729	
9	10	91	748	5509	36046	207775	1047376	4596553	17572114

¹From Mathar (2014) “The number of binary $n \times m$ matrices with at most k 1's in each row or column”

Probabilities

A model can be defined in terms of three main scenarios:

- ▶ $\theta_k = Pr(x_{kmn} = 1 | Z_{mn} = 1, H_{st} = 1)$ = probability that a pair of individuals match on feature k given that their households and they are a match.
- ▶ $\omega_k = Pr(x_{kmn} = 1 | Z_{mn} = 0, H_{st} = 1)$ = probability that a pair of individuals match on feature k given that their households are a match but they are not.
- ▶ $\delta_k = Pr(x_{kmn} = 1 | H_{st} = 0)$ = overall probability that the individuals match on feature k when their households are not a match.

Pair Match Probability

- ▶ Let $x_{mn} = (x_{1mn}, x_{2mn}, \dots, x_{Kmn})$ be the binary indicator vector of feature matches for the pair of individuals m and n from households s and t .
- ▶ We assume that probability of x_{mn} is given as,

$$\begin{aligned} P(x_{mn}|Z_{mn}, H_{st}) &= \left[\prod_{k=1}^K \theta_k^{x_{kmn}} (1 - \theta_k)^{1-x_{kmn}} \right] Z_{mn} H_{st} I_{m \in \mathcal{H}_s} I_{n \in \mathcal{H}_t} \\ &\quad \times \left[\prod_{k=1}^K \omega_k^{x_{kmn}} (1 - \omega_k)^{1-x_{kmn}} \right] (1 - Z_{mn}) H_{st} I_{m \in \mathcal{H}_s} I_{n \in \mathcal{H}_t} \\ &\quad \times \left[\prod_{k=1}^K \delta_p^{x_{kmn}} (1 - \delta_k)^{1-x_{kmn}} \right] (1 - H_{st}) I_{m \in \mathcal{H}_s} I_{n \in \mathcal{H}_t}. \end{aligned}$$

17 / 37

Likelihood

- ▶ The complete-data likelihood is given as,

$$L(Z, H, \Psi | X) = \prod_{s=1}^S \prod_{t=1}^T \prod_{m \in \mathcal{H}_s} \prod_{n \in \mathcal{H}_t} P(x_{mn}|Z_{mn}, H_{st}),$$

where $\Psi = \{\theta_1, \dots, \theta_K, \omega_1, \dots, \omega_K, \delta_1, \dots, \delta_K\}$.

- ▶ The likelihood is given as,

$$L(\Psi | X) = \sum_H \sum_Z \prod_{s=1}^S \prod_{t=1}^T \prod_{m \in \mathcal{H}_s} \prod_{n \in \mathcal{H}_t} P(x_{mn}|Z_{mn}, H_{st}).$$

- ▶ This is computationally intractable for moderate datasets because it involves summing over all possible H and Z matrices.

18 / 37

CEM Algorithm

- ▶ We propose using a CEM algorithm to fit the model.
This involves two steps:
 - ▶ **C-step:** Maximise $\log L(Z, H, \Psi | X)$ with respect to H and Z .
 - ▶ **M-step:** Maximise $\log L(Z, H, \Psi | X)$ with respect to Ψ .
- ▶ Repeat these steps until convergence.

19 / 37

Complete-Data Log-Likelihood

- ▶ The complete-data log-likelihood is:

$$\begin{aligned}\ell(Z, H, \Psi | X) = & \sum_{s=1}^S \sum_{t=1}^T \sum_{m \in \mathcal{H}_s} \sum_{n \in \mathcal{H}_t} Z_{mn} H_{st} \left[\sum_{k=1}^K x_{kmn} \log(\theta_k) + (1 - x_{kmn}) \log(1 - \theta_k) \right] \\ & + (1 - Z_{mn}) H_{st} \left[\sum_{k=1}^K x_{kmn} \log(\omega_k) + (1 - x_{kmn}) \log(1 - \omega_k) \right] \\ & + (1 - H_{st}) \left[\sum_{k=1}^K x_{kmn} \log(\delta_k) + (1 - x_{kmn}) \log(1 - \delta_k) \right]\end{aligned}$$

- ▶ For notational simplicity we write this as:

$$\ell(Z, H, \Psi | X) = \sum_{s=1}^S \sum_{t=1}^T \sum_{m \in \mathcal{H}_s} \sum_{n \in \mathcal{H}_t} Z_{mn} H_{st} a_{mn} + (1 - Z_{mn}) H_{st} b_{mn} + (1 - H_{st}) c_{mn}.$$

20 / 37

Model Fitting: Update H

- ▶ Assuming all the probability parameters and Z are fixed, we can write the complete-data log-likelihood as a function of H_{st} :

$$\ell(Z, H, \Psi | X) = \sum_{s=1}^S \sum_{t=1}^T \sum_{m \in \mathcal{H}_s} \sum_{n \in \mathcal{H}_t} H_{st} [Z_{mn} a_{mn} + (1 - Z_{mn}) b_{mn} - c_{mn}] + c_{mn}.$$

- ▶ Optimizing with respect to H is a linear programming problem:

$$\arg \max_H \sum_{s=1}^S \sum_{t=1}^T \sum_{m \in \mathcal{H}_s} \sum_{n \in \mathcal{H}_t} H_{st} [Z_{mn} a_{mn} + (1 - Z_{mn}) b_{mn} - c_{mn}],$$

21 / 37

Model Fitting: Update Z

- ▶ Assuming all the probability parameters and H are fixed, we can write the complete-data log-likelihood as a function of Z_{mn} :

$$\ell(Z, H, \Psi | X) = \sum_{s=1}^S \sum_{t=1}^T \sum_{m \in \mathcal{H}_s} \sum_{n \in \mathcal{H}_t} Z_{mn} [H_{st} (a_{mn} - b_{mn})] + C,$$

- ▶ Optimizing with respect to Z is a linear programming problem:

$$\arg \max_Z \sum_{m \in \mathcal{H}_s} \sum_{n \in \mathcal{H}_t} Z_{mn} [H_{st} (a_{mn} - b_{mn})]$$

22 / 37

Model Fitting: Update Parameters

- The parameter updates are straightforward:

$$\hat{\theta}_k = \frac{\sum_{s=1}^S \sum_{t=1}^T \sum_{m \in \mathcal{H}_s} \sum_{n \in \mathcal{H}_t} Z_{mn} H_{st} x_{kmn}}{\sum_{s=1}^S \sum_{t=1}^T \sum_{m \in \mathcal{H}_s} \sum_{n \in \mathcal{H}_t} Z_{mn} H_{st}},$$
$$\hat{\omega}_k = \frac{\sum_{s=1}^S \sum_{t=1}^T \sum_{m \in \mathcal{H}_s} \sum_{n \in \mathcal{H}_t} (1 - Z_{mn}) H_{st} x_{kmn}}{\sum_{s=1}^S \sum_{t=1}^T \sum_{m \in \mathcal{H}_s} \sum_{n \in \mathcal{H}_t} (1 - Z_{mn}) H_{st}},$$
$$\hat{\delta}_k = \frac{\sum_{s=1}^S \sum_{t=1}^T \sum_{m \in \mathcal{H}_s} \sum_{n \in \mathcal{H}_t} (1 - H_{st}) x_{kmn}}{\sum_{s=1}^S \sum_{t=1}^T \sum_{m \in \mathcal{H}_s} \sum_{n \in \mathcal{H}_t} (1 - H_{st})}.$$

- The estimates have an intuitive interpretation.

23 / 37

Blocking

- In many record linkage problems, blocking is used to improve computational efficiency.
- Blocking involves subsetting the data into non-overlapping groups. Matches are only allowed within blocks.
- We used Region of Residence (IREG) as the blocking variable because regional migration in Italy is quite low (eg. Bonifazi and Heins, 2000).
- We compare the results to when no blocking is used.

24 / 37

Initialisation Methods

- ▶ Initialise the parameters Z and H with the true values (**True Initial**).
- ▶ Use the Fellegi-Sunter model to initialise Z (**FS Initial**).
- ▶ Randomly Initialise the parameters (**Random Initial**).

Note: we will compare the performance of the proposed methodology with the method that matches individuals directly (fastLink R package).

25 / 37

Results - Households

- ▶ We can summarise the performance of method (with different initialisation methods) for matching households.

	True Initial	FS Initial	Random Initial	All Data
F_1 Score	62.13	62.14	60.72	61.31
Recall	90.51	90.56	88.35	90.33
Precision	47.29	47.30	46.25	46.40

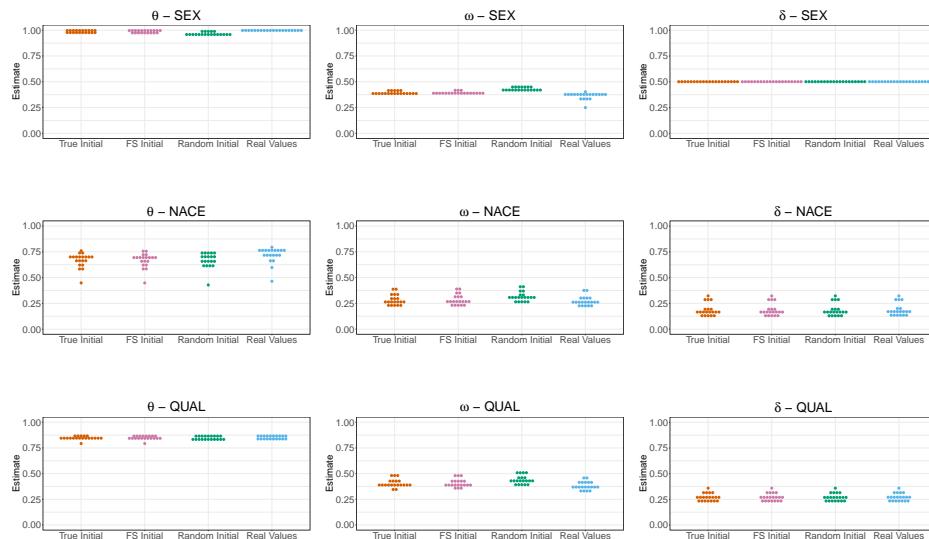
26 / 37

Results - Individuals

	uhlink True Initial	uhlink FS Initial	uhlink Random Initial	uhlink All Data	fastLink IREG	fastLink All
F_1 Score	80.05	80.08	75.03	82.52	45.63	47.06
Recall	94.39	94.36	79.11	94.19	58.67	60.68
Precision	69.50	69.55	71.36	74.21	37.33	38.43

27 / 37

Selected Parameters



28 / 37

Parameters

- ▶ The range of the parameter estimates across blocks is as follows:

	θ	ω	δ
SEX	0.958 - 1.000	0.373 - 0.423	0.500 - 0.503
ANASC	1.000 - 1.000	0.009 - 0.109	0.011 - 0.013
CIT	0.952 - 1.000	0.903 - 1.000	0.785 - 1.000
STUDIO	0.661 - 0.854	0.269 - 0.478	0.181 - 0.232
NACE	0.448 - 0.757	0.217 - 0.395	0.117 - 0.322
NASCREG	0.866 - 1.000	0.668 - 0.954	0.426 - 0.917
QUAL	0.793 - 0.871	0.343 - 0.488	0.217 - 0.359

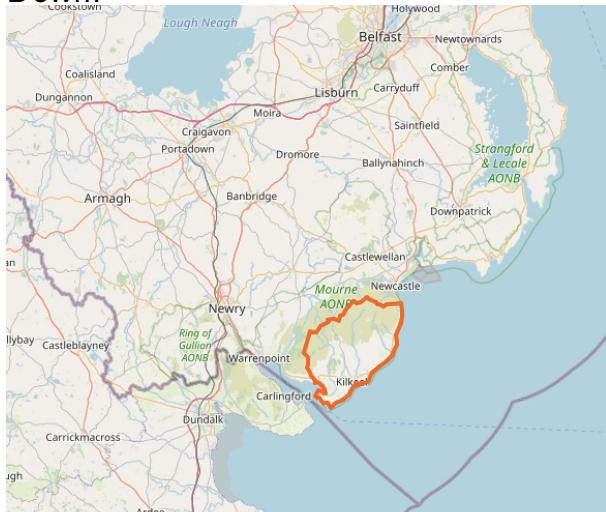
29 / 37

Returning to the Irish Census Data

30 / 37

Mourne Region

- Dermot Balson, a retired actuary, provided a carefully curated subset of the Irish Census from the Mourne region of County Down.



31 / 37

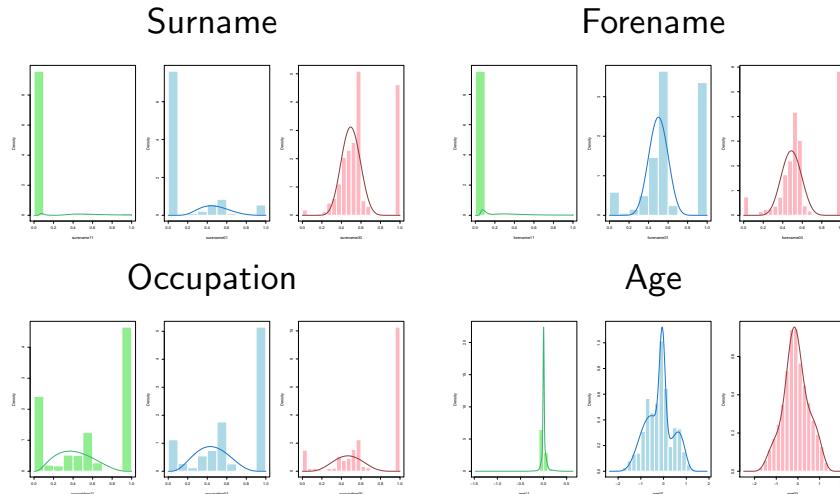
Differences Between Features

- The Irish census data has a number of different features.
- We propose comparing individuals for each feature using an appropriate discrepancy measure:
 - Text fields: Jaro-Winkler distance
 - Categorical: Hamming distance
 - Age: Difference (not absolute difference)

32 / 37

Differences Between Features

- ▶ Histograms of the distances observed are as follows:



33 / 37

Feature Model

- ▶ It is difficult to find an appropriate model for the distances.
- ▶ We currently propose dividing the distances into five bins.
- ▶ The bins can be equal spaced or found empirically using finite mixture model.
- ▶ Thereafter, the distances for each feature are modelled using a multinomial distribution with group specific parameters.

34 / 37

Preliminary Results: Households

- ▶ We have the following performance measures for matching households using the proposed approach.

Household		
	5 Bins	Mclust 5
F_1 Score	54.21	54.87
Recall	75.15	75.73
Precision	42.40	43.01

35 / 37

Preliminary Results Individuals

- ▶ We have the following performance measures for matching individuals using the proposed approach.

Individual			
	5 Bins	Mclust 5	fastLink
F_1 Score	74.44	74.97	38.76
Recall	77.73	78.47	47.95
Precision	71.42	71.76	32.53

36 / 37

Conclusions and Future work

- ▶ The model is able to recover the household and individual matches with good accuracy.
- ▶ Accounting for household information enhances the accuracy of individual matching.
- ▶ Performance is better compared to the method that matches individuals directly.

Future Work:

- ▶ Irish census data:
 - ▶ adapt model for urban areas
 - ▶ capture moving between households, split/merging of households
 - ▶ prepare for the release of 1926 census
- ▶ Enhanced inference using EM and/or Bayesian approach.

Hausdorff Distance-Based Record Linkage for Improved Matching of Households and Individuals in Different Databases

Thais Pacheco Menezes¹, Thomas Brendan Murphy^{1,2,3,4,*}, and Michael Fop¹*

¹School of Mathematics and Statistics, University College Dublin

²Insight Centre for Data Analytics, University College Dublin

³Institut d'Études Avancées, Université de Lyon

⁴ERIC, Université de Lyon

*Thomas Brendan Murphy and Michael Fop have contributed equally to this work

Corresponding author:thais.pachecomenezes@ucdconnect.ie

Abstract

Matching households and individuals across different databases poses challenges due to the lack of unique identifiers, typographical errors, and changes in attributes over time. Record linkage tools play a crucial role in overcoming these difficulties. This paper presents a multi-step record linkage procedure that incorporates household information to enhance the entity-matching process across multiple databases. Our approach utilizes the Hausdorff distance to estimate the probability of a match between households in multiple files. Subsequently, probabilities of matching individuals within these households are computed using a logistic regression model based on attribute-level distances. These estimated probabilities are then employed in a linear programming optimization framework to infer one-to-one matches between individuals. To assess the efficacy of our method, we apply it to link data from the Italian Survey of Household Income and Wealth across different years. Through internal and external validation procedures, the proposed method is shown to provide a significant enhancement in the quality of the individual matching process, thanks to the incorporation of household information. A comparison with a standard record linkage approach based on direct matching of individuals, which neglects household information, underscores the advantages of accounting for such information.

Keywords: Hausdorff distance; Household information; Linear programming; Matching databases; Record linkage

1 Introduction

Record linkage is the process of matching information from different sources that are believed to be related to the same entity (Herzog et al., 2007). Due to the digitization of census and survey-based data collection approaches, the application of record linkage methods to match entries across different databases is a field of growing interest, which enables the investigation of changes in population, demographic patterns, and family transitions over time (Ruggles et al., 2018; Abramitzky et al., 2020, 2021; Helgertz et al., 2022). The challenges regarding this task are related to the fact that the matching procedure is often based on information reported by the entity in the study, a process highly subject to typographical errors, inconsistencies, and changes over time (Abramitzky et al., 2021). Furthermore, errors in the data may appear as a consequence of how the survey was designed (Biancotti et al., 2008).

Concerning general record linkage methodology, a large body of work has been produced. One of the most famous approaches is the Fellegi-Sunter model (Fellegi and Sunter, 1969), which is widely used (e.g. Sadinle and Fienberg, 2013), and has also been recently implemented in conjunction with a linear programming framework (Moretti et al., 2019). Bayesian extensions of the Fellegi-Sunter

model and other approaches for record linkage have also been proposed in recent years (e.g. Steorts et al., 2016; Sadinle, 2017; Tancredi and Liseo, 2011; Fortini et al., 2001). Other approaches consider graph matching methods: Papadakis et al. (2022), for example, compares the performance of multiple bipartite graph matching algorithms.

One of the motivations behind our novel contribution derives from the fact that most of the widely used record linkage methods are focused solely on matching individuals, ignoring any available grouping information, such as household membership. The incorporation of such information in the linkage process has not been extensively explored. Frisoli and Nugent (2018) investigate the effect of including household information when matching records, comparing the results of record linkage procedures with and without such information. The model proposed by Fu et al. (2014) uses a graph matching framework which is shown to improve linkage accuracy when including the complete household structure in the matching process. Another work that considers household information is the one of Fu et al. (2011), where the main idea is to use the household membership to clean and link the data in a way that records containing errors and variations can be corrected, reducing the number of wrongly matched entities. Record linkage with grouping information is also explored by On et al. (2007), who propose a metric to measure similarity between groups that allows eliminating sets unlikely to be matching, enabling to focus the matching of entities in those groups having high similarity. Following this theme of including household information when matching entities, we define a general multi-step record linkage procedure that allows the incorporation of household information to improve the process of matching records across different databases. The methodology is developed and illustrated in application to record linkage of the Bank of Italy Survey of Household Income and Wealth (SHIW) databases (Bank of Italy, 2022).

An important step in matching databases is the quantification of the dissimilarity or similarity between pairs of records. Different metrics are employed to measure the dissimilarity between records, which are normally computed on the variables the entities are matched upon. According to the nature of these variables (text, numerical, categorical, etc.), different metrics can be used (Cohen et al., 2003; Herzog et al., 2007; Sayers et al., 2015). For example, for string variables, the Jaro-Winkler metric (Winkler, 1990) is the most commonly used. If the variables to be compared are categories chosen by the respondent, the comparison can be directly done so that the dissimilarity would be zero if the entities belong to the same category or one otherwise. When entities are grouped according to some structure, the standard distance metrics employed for record linkage need to be modified, as in this situation it is required to measure the dissimilarity between sets of individuals (see Eiter and Mannila, 1997, for a comprehensive overview of distance measures between two sets of points). In our proposed methodology, the membership to a given household is incorporated in the matching process, and to quantify the dissimilarity of two households across databases, we propose the use of the Hausdorff distance (Hausdorff, 1914). The Hausdorff distances between households are computed on the individual-level reported information and employed in a model used to predict the probability of a match between households. Subsequently, the matching of individuals is implemented by leveraging the information about the matched households and using a supervised learning method in combination with a linear programming optimization procedure. The use of the Hausdorff distance to incorporate household information is shown to be beneficial to the quality of the record linkage process.

The paper is organized as follows: Section 2 describes the motivating Bank of Italy SHIW data; Section 3 presents the Hausdorff distance and the approach used to measure the dissimilarity between records, also introducing the supervised learning models employed to estimate matching probabilities between households and individuals within households; Section 5 presents and discusses the results of the process of matching households and individuals for the SHIW databases; Section 6 concludes the paper with a discussion about limitations and potential developments.

2 Data: Italian Survey of Household Income and Wealth

The Bank of Italy has been conducting the Italian Survey of Household Income and Wealth (SHIW) since 1960 to collect information about the incomes and savings of Italians. Over the years, the survey has been extended to include information about wealth, financial behaviour, and other general economic aspects. The study uses a sample drawn in two stages, with municipalities and households as the primary and secondary sampling units, respectively. The sample represents the population officially resident in Italy, not accounting for people living in institutions (convents, hospitals, prisons, etc.) or those who are in the country illegally. The size of the sample has gradually increased reaching about 8000 households. In order to guarantee comparability across years, since 1989, part of the sample comprises households that were interviewed in the previous study. In this way, about 50% of the households are present in consecutive surveys.

Data from the surveys are published approximately every two years on the Bank of Italy website (Bank of Italy, 2022). The published microdata do not contain any information that could lead indirectly to the identification of the respondent, and, at the moment, they are available for download since 1989 in different formats. Additional material, containing the questionnaire and data description, is also provided.

Due to the nature of the survey procedure, the true match status between households is available via the unique identifier assigned to each household the first time it was included in the study and retained for all future inclusions. For individuals, their matching status can be validated due to the presence of individual IDs in prior surveys. These will enable the assessment of the matching performance of the proposed method in a supervised manner. An important point to highlight is that only matches between individuals who remained in the same household can be detected since an individual's ID is solely associated with their household. This constraint emerges from the data collection procedure, as transitioning to a new household, like through marriage, necessitates the creation of a new individual ID, thus erasing any previous linkage.

The record linkage framework proposed here is shown in application to the SHIW databases of 2014, 2016, and 2020. The 2014 database includes 19366 individuals spread across 8156 households, whereas the 2016 database consists of 16462 individuals within 7420 households. The 2020 database, on the other hand, comprises 15198 individuals distributed among 6239 households.

The questionnaire used in the survey includes several variables. Table 1 presents the name, the description, and the range of the variables considered in this work. Attention was given to variables conducive to constructing individual profiles, such as sex, year of birth, region of residence, and employment status. These variables offer a comprehensive view of the respondent's profile, although certain aspects may change over time, posing challenges in the matching process. Financial variables were omitted due to their sensitivity and volatility. In general, the majority of these variables are categorical, typically consisting of fewer than 10 distinct levels. There are, however, a few exceptions: variables such as those indicating the region of birth (NASCREG) and residence (IREG), as well as the variable detailing the sector of activity of the individual's employer (NACE), exhibit a more extensive range with over 20 levels. Additionally, the variable representing the year of birth (ANASC) is the

Table 1: Description of the variables considered for the matching procedure available in the 2014, 2016, and 2020 Italian survey.

Variable	Description	Range
SEX	Individual's sex	2 levels
CIT	Indicator if an individual is an Italian citizen or not	2 levels
ANASC	Year of birth	Discrete
STUDIO	Educational qualification	8 levels
NASCREG	Region of birth	21 levels
NACE	Sector of activity of the company where the individual works/worked	22 levels
IREG	Region of residence	20 levels
QUAL	Employment status	7 levels

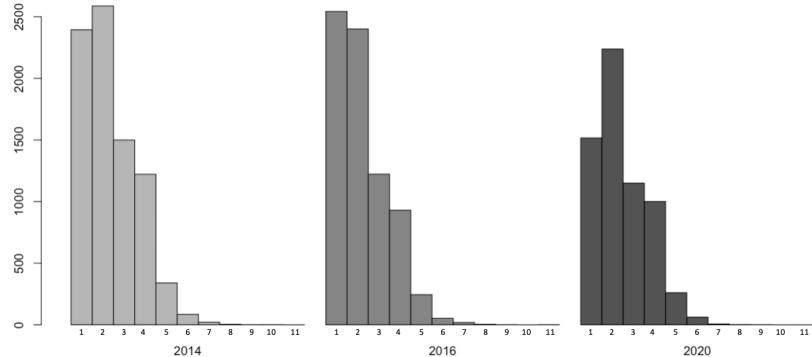


Figure 1: Barplot for the distribution of the size of the households in the 2014, 2016, and 2020 Italian survey.

only numerical variable in the data, characterized by a discrete range.

The databases include some missing values, generated under a not-missing-at-random mechanism. For the variable NASCREG, recording the region of birth, the absence of an answer is related to individuals not born in Italy. Hence, the missing entry is replaced by an extra category indicating that a subject is not born in Italy. The variable that indicates the sector of activity of the company (NACE) also has missing values, with most cases being individuals with working status corresponding to unemployed or pensioner. Also in this case a new category reflecting this information is created. However, eleven cases in 2016 still include missing information. These cases are missing at random since their working status is specified but no information about their activity sector is available. In practice, this means that, when compared with the other individuals, these eleven cases will be considered to have maximum dissimilarity with other cases for the variable NACE.

Across the 2014 and 2016 datasets, the SHIW data include 3804 matched households and 8660 matched individuals. For the 2016 and 2020 datasets, a total of 2983 household matches and 6434 individual matches are present. In the data, for most matched households, the size of the households across survey years remains the same. Figure 1 presents the barplot illustrating the distribution of household sizes for the years 2014, 2016, and 2020. A comparative analysis reveals a reduction in the count of single-person households within the 2020 survey. The growth in household size is evident in the calculated average household sizes: 2.37 for 2014, 2.22 for 2016, and an increase to 2.44 for 2020. However, it is worth noting that, aside from this shift, the overall distribution of household sizes remains relatively consistent across the three examined years with the majority of households being formed by up to two individuals. Regarding the area of residence, among the matched households across 2014 and 2016, only three changed their region of residence between surveys. This number increases to seven when comparing matched households between 2016 and 2020. These considerations underscore that the household structure tends to remain stable over survey years, and, given that most households are composed of two or more individuals, the inclusion of household information proves to be a valuable factor in the matching process.

It is essential to acknowledge that the SHIW data may be susceptible to errors and inconsistencies across survey years. A study by Biancotti et al. (2008) delves into data quality and measurement errors in the SHIW data, revealing that inconsistencies can arise in responses due to factors like interview duration, the interview process, and the broader survey design. These inconsistencies can pose challenges in identifying matching entities accurately. Thus, incorporating multiple sources of information at both household and individual levels could prove advantageous for enhancing record linkage accuracy.

The subsequent section introduces the record linkage framework, which is exemplified using the SHIW databases of 2014, 2016, and 2020. Our model primarily builds upon the 2014 and 2016 surveys, serving as foundational data for both the training and testing phases. Meanwhile, the 2020 database is exclusively employed in a specific test scenario, as elucidated later.

3 Record Linkage with Household Information

We propose the *household-and-Hausdorff-distance-based record linkage method*, `hhlink`, a two-step record linkage framework that uses the Hausdorff distance as the input to a supervised learning method for matching households between databases. The matched households are then employed as the basis for matching individuals using a combination of supervised learning and linear programming optimization. In this section, we present the main components of `hhlink`. Since the training process employs the 2014 and 2016 databases, we will illustrate the methodology using these survey years.

3.1 Hausdorff Distance Between Households

The Hausdorff distance (Hausdorff, 1914) measures the distance between two sets of points and it states that two sets are close if every point of either set is close to some point of the other one (Eiter and Mannila, 1997).

Consider that the individual i comes from the household \mathcal{H}_s from the set of all households in the 2014 database \mathcal{H}^{2014} , while the individual j is from household \mathcal{H}_t from the set \mathcal{H}^{2016} of all the households in the 2016 survey file. The distance between individuals i and j for a feature k is denoted with d_{ijk} . Consider $\beta_1, \dots, \beta_k, \dots, \beta_K$ a collection of non-negative coefficients. The linear combination of all K feature distances between individuals i and j is defined as

$$d_{ij} = \sum_{k=1}^K \beta_k d_{ijk}.$$

The Hausdorff distance between households \mathcal{H}_s and \mathcal{H}_t is then defined as:

$$\Delta_{st} = \max \left\{ \max_{i \in \mathcal{H}_s} \min_{j \in \mathcal{H}_t} d_{ij}; \max_{j \in \mathcal{H}_t} \min_{i \in \mathcal{H}_s} d_{ij} \right\}. \quad (1)$$

The Hausdorff distance corresponds to the greatest distance from any individual in one household to the most similar individual in the other household. Therefore, two households are considered close and likely to be a match if every individual in a household is close to the individuals in the other household. Consequently, pairs of households with the lowest Hausdorff distance are more likely to include matching individuals. Hence, the identification of similar households will enhance the process of linking individuals. The estimation of the β_k coefficients is implemented via maximum likelihood, as it is explained in Section 3.2.

For categorical variables, the distance d_{ijk} for two individuals is 0 if the individuals belong to the same category and 1 otherwise. For the numeric variable year of birth, the distance is calculated as the absolute difference between the years as $d_{ijk} = |ANASC_i - ANASC_j|/50$, where $ANASC_i$ and $ANASC_j$ are the year of birth of individual i and of individual j respectively; the factor 50 is to scale this distance so that it is close in range to 0 and 1. It is important to highlight that this factor is considered for interpretability only and does not affect the results.

3.2 Household Model

Let y_{st} be the indicator binary variable of matching status between households, taking the value of 1 if households \mathcal{H}_s and \mathcal{H}_t are a match and 0 otherwise. We are interested in determining the probability p_{st} of a match between the households \mathcal{H}_s and \mathcal{H}_t . The household model defines this probability as a logistic function dependent on the Hausdorff distance (1) between households:

$$p_{st} = P(y_{st} = 1 | \Delta_{st}) = \frac{e^{\beta_0 - \Delta_{st}}}{1 + e^{\beta_0 - \Delta_{st}}}, \quad \text{with } \beta_k \geq 0 \text{ for } k = 1, \dots, K, \quad (2)$$

where β_0 is the intercept. Note that the coefficients β_k are in the linear combination in the term Δ_{st} . Grouping all the parameters into the vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_k, \dots, \beta_K)$ and considering a Bernoulli model, the log-likelihood for two databases is given as follows:

$$\ell(\boldsymbol{\beta}) = \sum_{s=1}^{N^{2014}} \sum_{t=1}^{N^{2016}} \{y_{st} \log p_{st} + (1 - y_{st}) \log(1 - p_{st})\},$$

where N^{2014} and N^{2016} are the total number of households in 2014 and 2016 respectively.

The model is estimated and assessed within a supervised learning framework, hence consider a partition of the available data into a training and a test set. The procedure for dividing the data into training and test sets will be elucidated in Section 4.1. During the training process, the true match status y_{st} between households is known, and the parameter vector $\boldsymbol{\beta}$ is estimated through maximum likelihood:

$$\hat{\boldsymbol{\beta}} = \arg \max_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \sum_{s=1}^{N^{2014}} \sum_{t=1}^{N^{2016}} \{y_{st} \log p_{st} + (1 - y_{st}) \log(1 - p_{st})\}, \quad \text{with } \beta_k \geq 0 \text{ for } k = 1, \dots, K.$$

The above optimization is performed using standard box-constrained optimization via L-BFGS-B, as implemented in the R package `optimx` (R Core Team, 2022; Nash and Varadhan, 2011; Nash, 2014; Nash et al., 2022). Since we are dealing with distances, the model should encompass the fact that increasing the distance would cause the probability of a match to decrease. For this purpose, we impose the constraint that the β_k parameters are non-negative ($\beta_k \geq 0$ for $k = 1, \dots, K$). It is important to highlight that this log-likelihood is a complex non-linear function of the $\boldsymbol{\beta}$ parameters throughout the Hausdorff distance Δ_{st} in (1) presented in the calculation of the probability of the household being a match (2).

Upon obtaining the estimated set of parameters $\hat{\boldsymbol{\beta}}$ we can proceed to calculate the Hausdorff distance between any two households in the test data, using the equation:

$$\hat{\Delta}_{st}^* = \max \left\{ \max_{i \in \mathcal{H}_s^*} \min_{j \in \mathcal{H}_t^*} d_{ij}^*; \max_{j \in \mathcal{H}_t^*} \min_{i \in \mathcal{H}_s^*} d_{ij}^* \right\}.$$

Here, $d_{ij}^* = \sum_{k=1}^K \hat{\beta}_k d_{ijk}^*$ represents the linear combination of the individual's distances in the test data for the K features, taking into account the estimated parameter values $\hat{\beta}_k$. Following the computation of the Hausdorff distance, we can compute the estimated probability of a match between two households using the equation:

$$\hat{p}_{st}^* = P(\hat{y}_{st}^* = 1 | \hat{\Delta}_{st}^*) = \frac{e^{\hat{\beta}_0 - \hat{\Delta}_{st}^*}}{1 + e^{\hat{\beta}_0 - \hat{\Delta}_{st}^*}}.$$

Utilizing these estimated probabilities, we can proceed to estimate the indicator \hat{y}_{st}^* , which denotes whether two households in the test data \mathcal{H}_s^* and \mathcal{H}_t^* are classified as a match. Specifically, \hat{y}_{st}^* is assigned a value of 1 if the estimated probability \hat{p}_{st}^* associated with household \mathcal{H}_t^* is the highest among all possible matches for household \mathcal{H}_s^* and if $\hat{p}_{st}^* \geq \tau$. The threshold τ serves to control the proportion of matched households, enabling to determine that a household in one year does not have a match in the following survey if the highest probability of a match for that household is not sufficiently high. Spanning a range from 1 to 0, τ is defined as the highest value according to which the estimated proportion of matched households in the training data is as close as possible to the true proportion of matching households in the training set. It is essential to highlight that τ is estimated in the training phase and subsequently employed to determine the match status of households in the test data. Moreover, τ being equal to zero would imply that all households in 2014 will be matched with a household in 2016, while a value of one would signify that no households will be matched. In general, a higher value of τ corresponds to a lower proportion of matched households.

3.3 Individual Model

To estimate the probability of a match between individuals in linked households, an individual-level logistic regression model is employed. The model is defined in terms of the linear combination of distances between the individuals to be matched:

$$D_{ij} = \alpha_1 d_{ij1} + \cdots + \alpha_k d_{ijk} + \cdots + \alpha_K d_{ijK},$$

where d_{ijk} are the individual level distances between subjects i and j on variable k . The α_k parameters are the regression coefficients, which weigh the impact of the distance on the probability of a match between two individuals. Similar to the household model, to ensure that when increasing the distance the probability of a match must decrease, the α_k parameters are constrained to be non-negative.

To train this model, only the households in the training data that have a match ($y_{st} = 1$) are considered and all individuals inside these households are paired together. We define a binary variable z_{ij} which takes the value of 1 if individuals i and j are a match. The probability of a match for a pair of individuals from the two matching households, denoted as q_{ij} , is expressed as follows:

$$q_{ij} = P(z_{ij} = 1 | D_{ij}, y_{st} = 1) = \frac{e^{\alpha_0 - D_{ij}}}{1 + e^{\alpha_0 - D_{ij}}}, \quad \text{with } \alpha_k \geq 0 \text{ for } k = 1, \dots, K,$$

where α_0 is the intercept. Defining $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \dots, \alpha_k, \dots, \alpha_K)$ the vector of parameters, the model log-likelihood can be written as

$$\ell(\boldsymbol{\alpha}) = \sum_{s=1}^{N^{2014}} \sum_{t=1}^{N^{2016}} y_{st} \sum_{i=1}^{n^{2014}} \sum_{j=1}^{n^{2016}} \{z_{ij} \log q_{ij} + (1 - z_{ij}) \log(1 - q_{ij})\} I_{i \in \mathcal{H}_s} I_{j \in \mathcal{H}_t},$$

in which the indicator variables $I_{i \in \mathcal{H}_s}$ and $I_{j \in \mathcal{H}_t}$ take the value one if the individuals i and j belong to the households \mathcal{H}_s or \mathcal{H}_t , respectively, and zero otherwise; n^{2014} is the total number of individuals in the database for the year 2014, and n^{2016} is the total number of individuals in the database for the year 2016. The model training involves considering instances of matching individuals within matching households.

The model incorporates a non-negativity constraint on the coefficients. Furthermore, a ridge penalty is considered to correct for quasi-separation (Albert and Anderson, 1984; Heinze, 2006) and to obtain non-divergent and interpretable coefficient estimates. Consequently, to estimate the $\boldsymbol{\alpha}$ parameters, we maximize the following penalized log-likelihood:

$$\hat{\boldsymbol{\alpha}} = \arg \max_{\boldsymbol{\alpha}} \ell(\boldsymbol{\alpha}) = \sum_{s=1}^{N^{2014}} \sum_{t=1}^{N^{2016}} y_{st} \sum_{i=1}^{n^{2014}} \sum_{j=1}^{n^{2016}} \{z_{ij} \log q_{ij} + (1 - z_{ij}) \log(1 - q_{ij})\} I_{i \in \mathcal{H}_s} I_{j \in \mathcal{H}_t} + \lambda \sum_{k=1}^K \alpha_k^2,$$

subject to $\alpha_k \geq 0$ for $k = 1, \dots, K$.

This penalized log-likelihood is equivalent to a penalized logistic regression model with a response variable corresponding to the matching status of individuals. Hence, the estimation is implemented using the efficient routines available in the R package **glmnet** (Friedman et al., 2010, 2021), which allows the inclusion of the ridge penalty and the non-negativity constraints. Tuning of the penalty hyperparameter λ is performed using the default cross-validation procedure in the package.

With the estimated parameter vector $\hat{\boldsymbol{\alpha}}$, and conditioning upon the fact that the households \mathcal{H}_s^* and \mathcal{H}_t^* in the test data are predicted to be a match, we can estimate the probability of a match between two individuals inside these households, using,

$$\hat{q}_{ij}^* = P(z_{ij}^* = 1 | \hat{D}_{ij}^*, \hat{y}_{st}^* = 1) = \frac{e^{\hat{\alpha}_0 - \hat{D}_{ij}^*}}{1 + e^{\hat{\alpha}_0 - \hat{D}_{ij}^*}},$$

where $\hat{D}_{ij}^* = \hat{\alpha}_1 d_{ij1}^* + \cdots + \hat{\alpha}_k d_{ijk}^* + \cdots + \hat{\alpha}_K d_{ijK}^*$. This probability quantifies the likelihood of a match between the specific pair of individuals i and j in the test data, based on the estimated model parameters. Subsequently, we use these estimated probabilities, \hat{q}_{ij}^* , within a linear programming framework to compute the estimate \hat{z}_{ij}^* , taking the value of 1 if a match predicted between individuals i and j , and 0 otherwise. The details of the linear programming framework are presented in the following section.

3.4 Linear Programming

The proposed record linkage approach can be seen as a linear programming framework where a matrix of weights is assigned to each pair. In this context, the identification of the matches can be performed by maximizing the probability of a match under the constraint that each individual can be matched at most with one other individual. Following Moretti et al. (2019), we propose a similar approach to match individuals within households: the probabilities estimated from the individual-level logistic regression are used in a linear programming optimization framework to enforce one-to-one matches between individuals in the matched households across the databases. Consider two matched households \mathcal{H}_s , \mathcal{H}_t with N_s and N_t individuals to be matched, respectively, and for which the estimated $\hat{y}_{st} = 1$. Let \hat{q}_{ij} denote the estimated probability of a match for the pair of individuals i and j , and let z_{ij} be the binary indicator of whether individual i from household \mathcal{H}_s is a match for individual j from household \mathcal{H}_t to be estimated. The matching problem can be expressed as the following linear programming optimization problem:

$$\begin{aligned}\hat{z}_{ij} &= \arg \max_{z_{ij}} \sum_{i=1}^{N_s} \sum_{j=1}^{N_t} \hat{q}_{ij} z_{ij} \\ &\text{subject to } \sum_{i=1}^{N_s} z_{ij} \leq 1 \quad j = 1, \dots, N_t, \\ &\quad \sum_{j=1}^{N_t} z_{ij} \leq 1 \quad i = 1, \dots, N_s, \\ &\quad \sum_{i=1}^{N_s} z_{ij} \hat{q}_{ij} \geq \bar{q} \quad j = 1, \dots, N_t, \\ &\quad \sum_{j=1}^{N_t} z_{ij} \hat{q}_{ij} \geq \bar{q} \quad i = 1, \dots, N_s,\end{aligned}$$

where \bar{q} is the average estimated probability of a match between all the individuals in the matched households, $\bar{q} = \sum_{i,j} q_{ij} / (N_s N_t)$. In practice, the first two constraints indicate that each individual in 2014 can be matched with only up to one other individual in 2016, and vice versa. The last two constraints prevent matches for certain individuals, accounting for potential household changes such as individuals leaving or joining a household in the time between surveys. A pair of individuals within a matched household will not be matched if their probability of a match is below the average probability of a match for all pairs within that household. This criterion ensures that only individuals with a probability exceeding the average are assigned matches, while others remain unlinked.

We note that the same linear programming framework is implemented to estimate the matching status z_{ij}^* of individuals in the test data, using accordingly the corresponding estimated probabilities \hat{q}_{ij}^* and household matching labels \hat{y}_{st}^* .

4 Assessment of the Record Linkage Procedure

The `hhlink` approach proposed in the previous section is a supervised learning framework. Therefore, its performance is evaluated by considering a training and test splitting procedure of the SHIW data. Different metrics are employed for evaluation, and the performance of `hhlink` is compared with that one of a Fellegi-Sunter model employed to link all the individuals directly.

4.1 Training and Test Data

In the training process, the true match status between households and individuals is available and employed to estimate the parameters of the `hhlink` two-step approach. In the testing stage, the estimated models are used to assess the framework's performance in linking households and individuals. For this purpose, the available SHIW data are partitioned into training and test sets. Due to the structure of the data, the linkage goal, and the data collection process, two distinct methods are explored to split the data. These correspond to *internal* and *external* validation, respectively.

The first approach serves as an essential step in internally validating our methodology. Here, the training set is composed by selecting 60% of matching households and 60% of non-matching households for both 2014 and 2016 databases. Specifically, from the pool of households in 2014 that has a match in 2016, 60% are randomly chosen along with their corresponding matches. Simultaneously, among the households in the 2014 data that do not have a corresponding match in the 2016 database, 60% are randomly chosen to complete the training data for that year. This process is then repeated for the 2016 data, where 60% of the non-matched households from that year are similarly selected to create the corresponding training data. The remaining non-selected households will form the test set. The record linkage approach is subsequently deployed to predict match statuses on both the training and test data. To ensure that the results obtained are not merely the result of favorable training and test data partitions, the procedure is replicated ten times, and the results presented will reflect the model's average performance across all of these partitions.

The external validation approach involves training the model using the complete data from the 2014 and 2016 surveys and subsequently testing its performance by matching the 2016 database with the 2020 database. This method ensures the absence of data leakage, as none of the information from the 2014 database or the real matching status is employed in the testing phase. Moreover, there is no overlap across databases for training and testing, given that the testing phase includes an entirely new database whose instances are not present at all in the training phase. This validation approach provides a distinct demarcation between the training and testing datasets. Furthermore, by conducting external validation, we can assess the model's performance when a new survey is released, with the objective of matching the fresh data to the latest available survey.

4.2 Performance Measures

Given that information on actually matching households and individuals is available in the data, standard metrics can be used to assess the proposed record linkage framework's predictive performance.

The class distribution of matching/non-matching instances is considerably unbalanced due to the nature of the record linkage problem. For this reason, we consider the F_1 score to measure the performance of the models:

$$F_1 = \frac{2(\text{precision} \times \text{recall})}{\text{precision} + \text{recall}} = \frac{2\text{tp}}{2\text{tp} + \text{fp} + \text{fn}},$$

where tp represents true positives, fp stands for false positives, and fn denotes false negatives. Values of this score close to 1 indicate good performance. In the score calculation, precision measures the proportion of correct matches among all matches made, while recall reflects the percentage of correct matches considering the known true match status.

We consider also the false positive (FPR) and false negative (FNR) rates, defined as:

$$\text{FPR} = \frac{\text{fp}}{\text{fp} + \text{tn}}$$

$$\text{FNR} = \frac{\text{fn}}{\text{fn} + \text{tp}}.$$

The FPR measures the proportion of records that were wrongly assigned as matches while the FNR corresponds to the proportion of missed matching instances; lower values in both metrics are preferable.

These metrics will be utilized to evaluate the quality of the record linkage process and to compare the performance of the proposed `hhlink` with a model that matches individuals directly without taking into account household information.

4.3 Method for Comparison

The primary objective of this paper is to demonstrate the enhancement achieved in the individual matching process through the preliminary matching of households. To ascertain the degree of improvement in match quality resulting from the inclusion of household information, we will conduct a comparative analysis against an alternative linking methodology. This evaluation aims to contrast our proposed approach with an alternative method that matches individuals directly.

The considered alternative method implements the Fellegi-Sunter model, estimated using the Expectation-Maximization algorithm as implemented in the `fastLink` R package (Enamorado et al., 2019, 2020); in what follows, we denote with `fastLink` the Fellegi-Sunter model as implemented in the R package.

Given the computationally intensive nature of direct individual matching in `fastLink`, significant computational resources and time are essential prerequisites. This arises from the necessity to check the match possibilities for all potential pairs, a task that can become overwhelmingly large. For example, matching the 2014 and 2016 data would require evaluating an enormous number of pairs, totaling 19366×16462 . To alleviate this computational burden, blocking is applied, a common practice in record linkage scenarios.

Blocking partitions records into exclusive groups, such that only records within the same block are eligible for matching. This approach dramatically reduces the number of pairs that require evaluation (Steorts et al., 2014). We consider blocking on two factors: gender and region of birth, striving to achieve a balance between block size and the total number of blocks. Consequently, only individuals with the same gender and born in the same region are considered potential matches when applying the `fastLink` method.

5 Record Linkage of the SHIW Databases

As outlined in Section 3.2, we evaluate the model's performance by utilizing the estimated parameters to predict the match status between households \mathcal{H}_t and \mathcal{H}_s in training and test sets. Within each pair of households identified as a match, we assess the classification performance of the individual's model, as detailed in Section 3.3. In this context, the parameters derived from the logistic regression and the linear programming optimization process enable us to estimate the indicator variable of whether individuals i and j constitute a match.

5.1 Internal Validation

Initially, we present the outcomes obtained when the databases of 2014 and 2016 were randomly divided into training and test data ten times. We note that due to the tenfold repetition of model fitting and testing, we are able to assess not only the variability of the parameter estimates but also the variability in the model's performance metrics.

Table 2 presents the average estimates of the model coefficients β_k alongside the standard deviation of the estimates considering all splits. The estimates show that the variable accounting for the distance

Table 2: Average estimates of the household model’s parameters. The StDev represents the standard deviation associated with each estimate.

	Intercept	SEX	ANASC	CIT	STUDIO	NACE	NASCREG	IREG	QUAL
Estimate	-0.27	2.82	13.74	0.00	1.63	1.41	3.33	7.98	0.01
StDev	0.06	0.07	0.60	0.00	0.04	0.06	0.07	3.06	0.01

between the years of birth (ANASC) is the one with the largest weight in the calculation of the Hausdorff distance. We highlight that the values for this variable are continuous while all the others are equal to 1 if they are exactly the same for two individuals or 0 otherwise. The variable with the second largest weight is the variable indicating the region of residence of the household (IREG) while the indicator of Italian citizenship (CIT) does not contribute to the distance in the model.

Concerning variability, for the majority of variables, the estimates tend to vary closely around the mean value. The variable IREG exhibits the highest standard deviation, signifying greater variability in the estimations across model replications. Notably, despite the low standard deviation for the employment status variable (QUAL), the combination of a low average estimate suggests a substantial variability in this estimate, with a coefficient of variation of 100%.

Leveraging the available ground truth on matching households, Figure 2 presents quality measures for the household model considering all possible pairs of households. In this scenario, each household in 2014 was paired with all households in the 2016 set. The results indicate a high F_1 Score, driven by high recall and positive predictive values, suggesting that, in general, the model can correctly classify household pairs. Specifically, the average recall of 73% for the training data and 75% for the test data underscores the model’s ability to accurately identify household pairs known to be a match. It is important to note that the high false-negative rate may be attributed to situations where the highest probability of a match was not associated with the true matching household, or the estimated probability of a match fell below the defined threshold τ . In the training data, τ is determined as the value that makes the estimated total number of matches between households approximately equal to the proportion of matches in the training data. Given an average matching proportion of 46.65% in the training data, the estimated value for τ is 0.11 on average. This implies that to achieve around 46% matches between households in the training data, only households with probabilities greater than 0.11 are considered as potential matches. The estimated τ value is then applied to the test data to filter the potential matches in the household model.

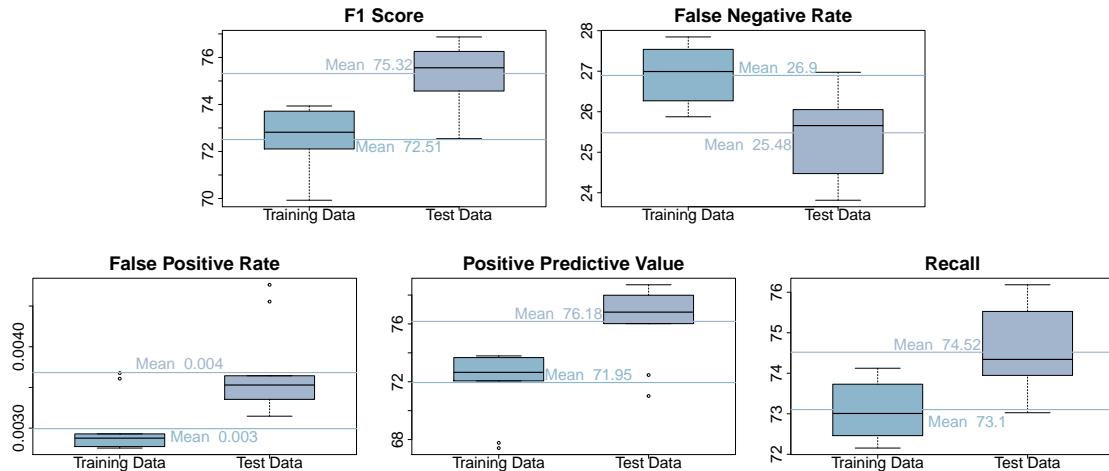


Figure 2: Boxplots illustrating performance metrics for household pair match status prediction.

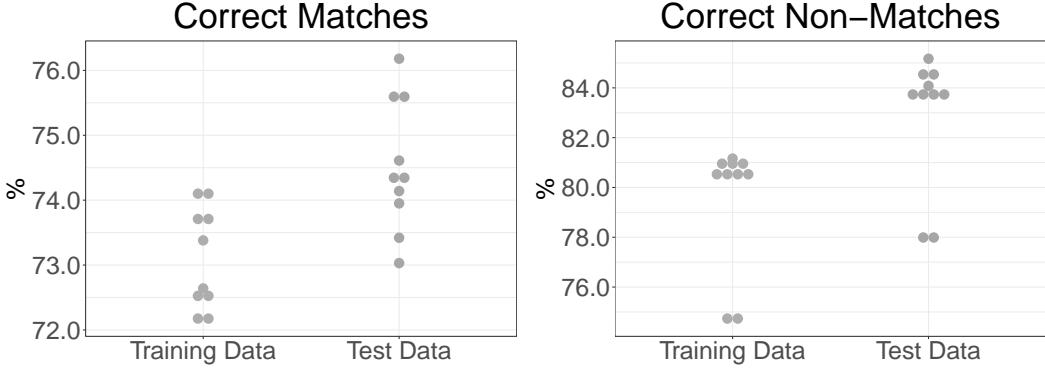


Figure 3: The left plot illustrates the proportion of correctly matched households among those with a match in each data split. The right plot shows the proportion of households accurately identified as not having a match among those without a match.

We also assess the household model’s performance by examining the predictive performance for each household in the 2014 database individually, rather than considering all possible pairs. We examine whether the model correctly matches households that have a match and whether it accurately identifies households without a match for each household in 2014. Figure 3 illustrates, for each data partition, the proportion of correctly matched households (Correct Matches) and the proportion of households correctly classified as not having a match (Correct Non-Matches). The results demonstrate that, among households with a match, approximately 72% to 74% were correctly matched in the training set, expanding to 73% to 76% in the test data. For households without a match, the proportion of correct non-matches is higher, ranging from around 75% to 81% in the training data and 78% to 85% in the test data. These results focused on assessing the model’s performance for each 2014 household in the training and test data underscores the model’s ability to effectively match these households or to accurately identify when they do not have a match.

To gain insights into the estimated probabilities of a match, we assess the rank of these probabilities for each household in 2014 in relation to the corresponding true matches in the 2016 database. For each split, the estimated probabilities of a match between a 2014 household and all 2016 households are ranked in descending order. Accurate model estimates would position the true matching household at the top. Table 3 presents the average percentage of actual matching households occupying the first position in the predicted probabilities, ranked in descending order, for both the training and test sets. The table shows that, on average, approximately 76.74% of the highest probability from the model corresponds to the true match in the training data. For the test data, this value is 79.55%. Moreover, the top three positions correspond, on average, to actually matching pairs in around 85% of the cases for both training and test sets. The results indicate that selecting the top three households with the highest probability of a match will likely include the true match in the selection.

Table 3: Rank of the probability of the correctly matching household. The rank 1 indicates that the highest probability is associated with the 2016 household which is the correct match. Likewise, rank 2 implies that the match had the second-highest probability, and so on.

Rank	Training	Test
1	76.74	79.55
2	4.92	4.19
3	1.76	1.14
4	1.09	0.64
≥ 5	15.49	14.48

Table 4: Average estimates of the individual’s model parameters.

	Intercept	SEX	ANASC	CIT	STUDIO	NACE	NASCREG	IREG	QUAL
Estimate	1.88	2.77	4.44	0.46	1.18	0.46	0.96	0.00	0.58
StDev	0.02	0.02	0.02	0.10	0.02	0.02	0.03	0.00	0.02

After having matched the households, the subsequent phase involves matching individuals within each linked household. Across each of the ten data splits into training and test subsets, conditionally on the matching households, we employ the logistic regression model outlined in Section 3.3 to predict the probability of a match between individuals. Table 4 provides the average parameter estimates for the individual’s model, along with their corresponding standard deviations. The variables associated with year of birth (ANASC) and sex (SEX) are the ones with the largest coefficients in the model, indicating that they contribute the most to the probability of two individuals being a match. The coefficient of the variable related to the region of residence (IREG) is shrunk to zero, in contrast to the household-level model where it had the second largest weight in the Hausdorff distance. This indicates that the region of residence largely contributes to linking two households, but it is no longer relevant when matching individuals within a pair of matched households. Regarding the variability of the estimates, the conclusion is similar to the one observed in the household model: most of the estimates do not vary too much around the mean.

Given the estimated matched households and the fitted individual-level logistic regression model, the probabilities from the logistic regression are employed in an optimization framework, as explained in Section 3.4. When assessing the individual’s model performance, we highlight that the results account only for the pair of individuals inside matched households.

As detailed in Section 4.3, we compare our proposed `hhlink`, which incorporates household information, with the `fastLink` method, which directly matches individuals. For each training and testing split, we compute performance metrics to assess the model’s ability to correctly classify pairs as matches or non-matches. Figure 4 illustrates the boxplots of these performance measures for the test sets.

As we examine the model’s performance on the unseen test data, `hhlink` demonstrates superior performance across all metrics except for the false positive rate. The lower false positive rate exhibited by `fastLink` can be attributed to its propensity to assign only a limited number of pairs as matches.

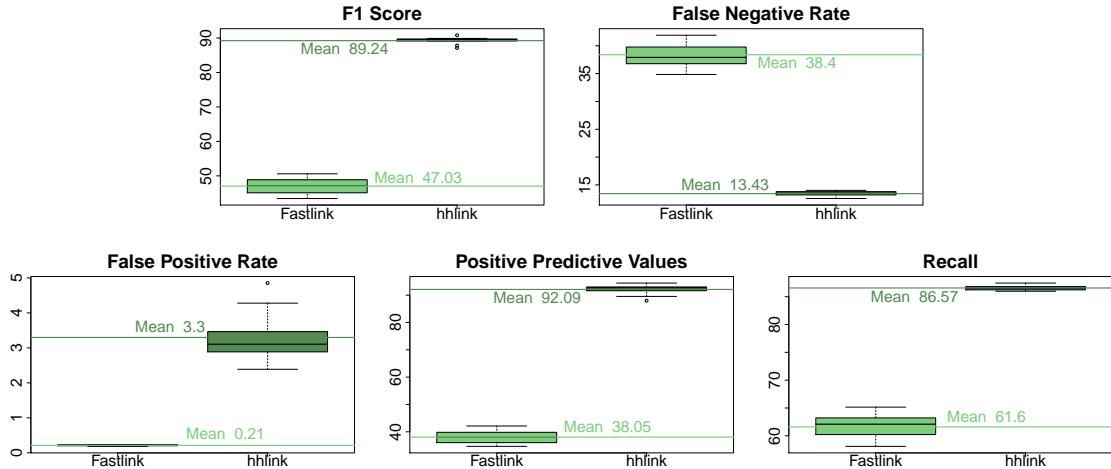


Figure 4: Boxplots illustrating performance metrics for individual pair match status prediction on the test data for `fastLink` and `hhlink` methods.

Consequently, it has a reduced chance of making false positive errors, as it predominantly categorizes matches as non-matches, incurring a higher false negative rate. The precision and recall values underscore the advantages of incorporating household information in the matching process. These values are significantly higher for `hhlink` compared to `fastLink`, which directly matches individuals. Specifically, the recall value indicates that `hhlink` correctly identifies approximately 87% of pairs known to match, in contrast to the 62% achieved by `fastLink`.

It is worth highlighting further that the number of pairs of individuals to be examined across databases is quite large, posing some computational burdens, since a large number of matching probabilities need to be estimated. Although blocking techniques can help alleviate this issue to some extent, `fastLink` still demands consideration of more than 3,000,000 potential pairs on the training data and more than 1,500,000 on the test data. In contrast, `hhlink` offers an effective remedy by reducing the number of individual pairs requiring evaluation. Notably, the detection of matching households essentially serves as an additional blocking step. Consequently, one only needs to consider pairs between households that have been identified as matches, significantly reducing the number of pairs to be examined to around 33,000 on the training set and 21,000 on the test data.

Given the substantial volume of pairs for comparison, and the unbalanced nature of the framework in which most of the individual's pairs are non-matching pairs, good performance measures can also be achieved by a method that assigns most pairs as non-matching, even if the method is not well designed. Additionally, previous results are only accounting for pairs of individuals within matched households. Individuals inside households that were not matched have not been paired with any other individual, making it impossible to assess the model's ability to correctly classify individual pairs in those cases. Therefore, it is of interest to assess the performance of the record linkage methods at identifying if an individual in the 2014 database has a match in the 2016 database, regardless of the number of total pairs, and if the matched individual in 2016 is correctly identified. In this case, we have two correct outcomes: the model correctly identifies the individual's match (Correct Matches) or it is able to correctly detect that the individual does not have a match (Correct Non-Matches).

In this regard, Figure 5 presents boxplots illustrating the difference between the percentage of correct matches and correct non-matches identified by our `hhlink` approach in comparison to the `fastLink` method. In this figure, positive values indicate that `hhlink`, incorporating household information, is better at identifying more correct matches or non-matches compared to `fastLink`. Conversely, negative values suggest that `fastLink` performed better. An examination of the results reveals that, on average, `hhlink` exhibits a higher proportion of correctly identified matches in the training data, surpassing the `fastLink` proportion by 10.59 percentage points. In the test data, this average difference is reduced to 4.39 points. Notably, `hhlink` excels in accurately identifying individuals without matches, exhibiting a substantial average improvement of 57.39 points in the training data and

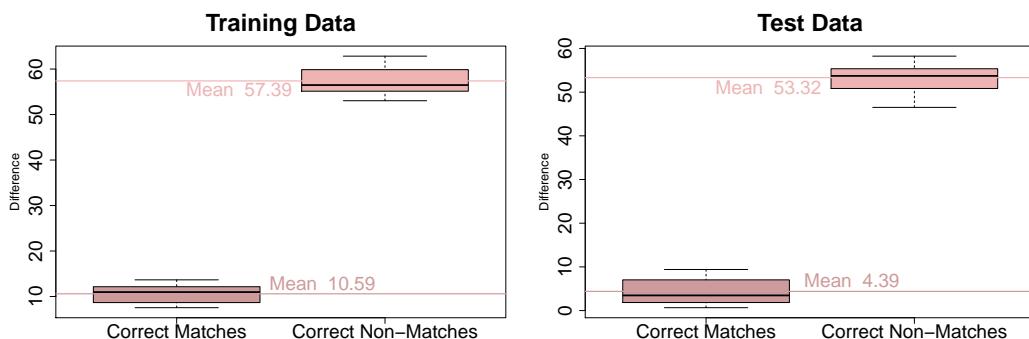


Figure 5: Boxplot displaying the difference in percentages between correctly identified matches and correct non-matches between `hhlink` and `fastLink`. The plot represents the average point-to-point difference between these approaches, with positive values denoting the `hhlink` better performance.

Table 5: Estimates of the household model parameters trained using the 2014 and 2016 survey data.

	Intercept	SEX	ANASC	CIT	STUDIO	NACE	NASCREG	IREG	QUAL
Estimate	-0.69	2.86	14.76	0.00	1.60	1.42	3.35	7.15	0.00

53.32 points in the test data when compared to the `fastLink` approach.

In summary, the results underscore the effectiveness of the `hhlink` method in correctly matching households. Additionally, the benefits of matching households become evident in the subsequent step of matching individuals, with consistently superior performance compared to methods directly matching individuals.

These findings are derived from the split of the survey databases from 2014 and 2016 into various training and test sets. In the following section, we will present the results of a validation involving an external test data set, where the model is trained using complete surveys from 2014 and 2016, and then evaluated by matching the 2016 survey with the 2020 data.

5.2 External Validation

This validation simulates a scenario where a new survey is available and a researcher is asked to match the individuals on this new database with the individuals on the previous survey. In this scenario, we will assess the method performance by using the surveys of 2014 and 2016 to train the method, then test it by matching the 2016 survey with the new 2020 data.

Table 5 provides the parameter estimates for the household model in this scenario. The estimates mirror those obtained in the internal validation, as detailed in Table 2. The variable ANASC (year of birth) is the one with the highest estimated weight in the Hausdorff distance, while CIT (Italian citizenship) and QUAL (employment status) do not contribute to the distance. We note that in this instance the model is trained on the entirety of the 2014 and 2016 survey data, hence no standard deviation is associated with the estimates.

Considering all possible pairs of households to be classified as matches or non-matches, Table 6 presents the performance measure for the household model considering the training (2014-2016) and test (2016-2020) scenario. The household model yields a positive predictive value (PPV) of 68.37% when matching households from the 2014 database with the 2016 database, and a value of 60.19% for matching households between the 2016 and 2020 surveys. These results indicate that, among all pairs classified as matches, the majority corresponds to true matches. Additionally, the high recall values suggest that the model is effective in identifying matching pairs. However, the results show an increase in the false negative rate in the test scenario. This discrepancy can be traced back to the threshold estimation process. The threshold is determined to achieve a proportion of estimated households in the training phase equal to the true percentage (46.64%) of matching households between 2014 and 2016. However, when this threshold is applied to match households in the 2016-2020 pair, which features a notably lower true proportion of matching households (40.20%), the estimated threshold

Table 6: Performance metrics for predicting the match status of household pairs. The 2014-2016 column displays results from the training phase utilizing the entire 2014 and 2016 databases. The 2016-2020 column reports results on the test scenario when matching the 2016 database with the 2020 data.

Metric	2014-2016	2016-2020
F_1 Score	70.30	57.17
FNR	27.66	45.56
FPR	0.002	0.002
PPV	68.37	60.19
Recall	72.34	54.44

Table 7: Ranking of correctly matching households by probability. The first column displays results from the training process using the 2014 and 2016 data, while the second column presents results for matching the 2016 survey with the 2020 data. A rank of 1 signifies that the highest probability corresponds to the correct household match. Similarly, rank 2 indicates the second-highest probability match, and so on.

Rank	2014-2016	2016-2020
1	74.21	64.06
2	5.16	5.49
3	1.97	2.43
4	1.39	2.18
≥ 5	17.26	25.85

may be deemed too high for the more recent surveys. Consequently, this discrepancy contributes to an increase in the false negative rate.

Table 7 provides the rank analysis for the household model. The rank serves as a metric indicating the position of the true match within the household matching process. The results consistently echo the previous findings obtained in the internal validation. In both scenarios, the highest match probability is predominantly associated with the true match in the subsequent survey for the majority of households. In particular, when matching households from 2016 to 2020, in 64.06% of cases the highest probability is associated with the true match. In general, across both scenarios, the true household match tends to fall within the top three highest match probabilities. However, for the 2014 – 2016 case, there are instances (17.36%) where a household’s probability of a match with its true match exceeds rank 4, while for the 2016 – 2020 case, this occurs in 25.85% of cases. As the scenario of training the model on the 2014-2016 databases and matching the 2016 survey to the 2020 data is more challenging, the likelihood of including the actually matching household in the top four positions is reduced.

After the household matching step across databases, the subsequent step involves fitting the individual model given the estimated household match status. Table 8 provides the parameter estimates for the individual model, which is trained using pairs of individuals within the matched households from the 2014 and 2016 surveys. The estimates are in line with those reported in the internal validation, where the model was trained only on random subsets of the 2014 and 2016 databases, as reported in Table 4. Also in this case ANASC and SEX have the largest weights, and IREG does not exert a significant impact in matching individuals. This consistency underscores the robustness of the estimates across different evaluation scenarios.

As elaborated in Section 3.4, following the estimation of the parameters in the individual model, we compute the probability of a match for all the pairs of individuals. Subsequently, leveraging a linear programming framework, we link individuals across databases. We apply `hhlink` to match pairs of individuals across the 2014 and 2016 surveys in the training phase, as well as link individuals between the 2016 and 2020 surveys in the testing phase. We also use the `fastLink` approach to match individuals across the databases for comparison. We remark that, in applying `fastLink`, blocking is employed to mitigate computational workload, as detailed in Section 4.3. The match performance in each of these scenarios is reported in Table 9.

The `hhlink` approach has an F_1 score higher than `fastLink` in both linking the 2014 with the 2016 survey and the 2016 and 2020 database. This indicates that the record linkage of individuals based on matched household information outplays `fastLink` which matches individuals directly. As before, the lower false positive rate (FPR) of `fastLink` is associated with fewer matches being assigned in

Table 8: Estimates of the individual model parameters trained using the data of 2014 and 2016

	Intercept	SEX	ANASC	CIT	STUDIO	NACE	NASCREG	IREG	QUAL
Estimate	1.88	2.77	4.43	0.50	1.17	0.47	0.95	0.00	0.58

Table 9: Comparison of individuals matching quality between `hhlink` and `fastLink`.

	<code>hhlink</code>		<code>fastLink</code>	
	2014-2016	2016-2020	2014-2016	2016-2020
F_1 Score	87.29	82.93	30.37	26.57
FNR	13.25	16.69	57.47	61.88
FPR	5.05	7.44	0.12	0.12
PPV	87.84	82.55	23.62	20.03
Recall	86.75	83.31	42.53	38.12

this method, resulting in a higher false negative rate (FNR). Finally, the precision and recall values reinforce that the inclusion of household information is beneficial to the process of matching individuals since such values for `hhlink` are more than double than those of `fastLink`.

As before, the extensive number of individual pairs across multiple databases presents significant computational challenges, especially when dealing with complete sets of databases from 2014, 2016, and 2020, reaching evaluation of 7,000,000 pairs in the `fastLink`. Such a large volume of pairs can potentially skew performance assessments. Therefore, it's crucial to assess record linkage methods based on their ability to accurately identify matches in the 2016 database for individuals in 2014, or matches in the 2020 database for individuals in 2016, regardless of the total number of pairs. This evaluation focuses on two outcomes: correctly identifying matches (Correct Matches) and accurately discerning non-matches (Correct Non-Matches). Table 10 provides a comprehensive overview of these performance metrics for the examined record linkage approaches.

The `hhlink` approach consistently outperforms the direct individual linkage approach. When household information is factored in, it results in the detection of 5578 (or 64.41%) of the 8660 actual matches between individuals in the 2014 and 2016 datasets and 2985 (46.39%) of 6434 for the individuals in the 2016 and 2020 datasets. In stark contrast, the `fastLink` approach lags behind with detection of only 3659 (42.25%) and 2420 (37.61%) respectively. It's crucial to note that the `hhlink` approach can correctly identify more than 90% of the individuals that do not have a match while the percentage is around 30% for the `fastLink` approach. We highlight that the reduced performance when matching the 2016 survey with the 2020 database in the testing phase, in comparison to the internal validation results, may be attributed to the amends done in 2020 in the traditional sampling design to improve the sample representativeness of some population groups, as informed in the Bank of Italy website. The four-year gap between these surveys, instead of the usual two years can also contribute to this. Nonetheless, these findings underscore the significant enhancement achieved by incorporating household information, leading to an increased number and quality of correctly identified matches.

6 Discussion and Conclusions

This work introduced a novel record linkage approach, `hhlink`, contributing to two key aspects. Firstly, it introduces the Hausdorff distance as a valuable metric for effectively measuring the dissimilarity between households during the matching process. Secondly, it underscores the advantages of initi-

Table 10: Number of correctly detected individual matches and non-matches for `hhlink` and `fastLink`.

	<code>hhlink</code>		<code>fastLink</code>	
	2014-2016	2016-2020	2014-2016	2016-2020
Correct Matches	5578 (64.41%)	2985 (46.39%)	3659 (42.25%)	2420 (37.61%)
Correct Non-Matches	10125 (94.57%)	9615 (95.88%)	3103 (28.98%)	3473 (34.63%)

ating the matching process at the household level when linking individual records across databases, ultimately improving data integration and the quality of the matched results.

The proposed `hlink` approach is a multi-step methodology. The first step employs the Hausdorff distance to estimate the probability of a match between pairs of households, based on linear combinations of distances between individual features. The following step employs logistic regression and linear programming optimization to match individual records within identified matched households.

The `hlink` method is showcased and evaluated in application to record linkage of the Italian Survey of Household Income and Wealth (SHIW) data, demonstrating the substantial benefits of considering household information when linking individual records across databases. Across internal and external validation frameworks, evaluation metrics consistently indicate superior performance of `hlink` compared to a method that directly matches individual records without leveraging household information.

A limitation of the proposed approach is in the supervised nature of the method, which requires the availability of labeled data where identifiers of matching households and individuals between databases are needed for training. This opens interesting avenues for future research. Future work will explore extensions to unsupervised learning for record linkage, where grouping information of the instances is available but not identifiers that can be used for matching. Unsupervised extensions of the proposed approach could be particularly useful in matching surveys with a larger time gap.

The proposed approach has been developed in application to record linkage of survey data collected at the household level. However, we remark that the proposed framework holds the potential for record linkage in other databases with grouping and hierarchical structures. The methodology's applicability extends beyond the specific data used, making it a valuable tool for data integration and analysis in various domains where grouping information on the individual records to be linked is available.

Declarations

Funding: This publication has emanated from research conducted with the financial support of Science Foundation Ireland under grant numbers 18/CRT/6049 and 12/RC/2289_P2 and a visiting period at Collegium de Lyon.

Conflicts of interest: The authors declare that there is no conflict of interest.

Ethical Conduct: The manuscript is only submitted to the Journal of Classification. The submitted work is original and is not published elsewhere in any form or language.

Data Availability: The data that support the findings of this study are openly available on the Bank of Italy website (Bank of Italy, 2022).

References

- Abramitzky, R., Boustan, L., Eriksson, K., Feigenbaum, J., and Pérez, S. (2021). Automated linking of historical data. *Journal of Economic Literature*, 59(3):865–918.
- Abramitzky, R., Mill, R., and Pérez, S. (2020). Linking individuals across historical sources: A fully automated approach. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 53(2):94–111.
- Albert, A. and Anderson, J. A. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, 71(1):1–10.
- Bank of Italy (2022). Bilanci delle famiglie Italiane. <https://www.bancaditalia.it/statistiche/tematiche/indagini-famiglie-imprese/bilanci-famiglie/documentazione/index.html> (Accessed: 2022-10-11 and 2023-08-03).
- Biancotti, C., D’Alessio, G., and Neri, A. (2008). Measurement error in the Bank of Italy’s Survey of Household Income and Wealth. *Review of Income and Wealth*, 54(3):466–493.
- Cohen, W. W., Ravikumar, P., and Fienberg, S. E. (2003). A comparison of string distance metrics for name-matching tasks. In *Proceedings of the 2003 International Conference on Information Integration on the Web*, pages 73—78. AAAI Press.
- Eiter, T. and Mannila, H. (1997). Distance measures for point sets and their computation. *Acta Informatica*, 34(2):109–133.
- Enamorado, T., Fifield, B., and Imai, K. (2019). Using a probabilistic model to assist merging of large-scale administrative records. *American Political Science Review*, 113(2):353–371.
- Enamorado, T., Fifield, B., and Imai, K. (2020). *fastLink: Fast Probabilistic Record Linkage with Missing Data*. R package version 0.6.0.
- Fellegi, I. P. and Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210.
- Fortini, M., Liseo, B., Nuccitelli, A., and Scanu, M. (2001). On Bayesian record linkage. *Research in Official Statistics*, 4(1):185–198.
- Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.
- Friedman, J., Hastie, T., Tibshirani, R., Narasimhan, B., Tay, K., Simon, N., Qian, J., and Yang, J. (2021). *glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models*. R package version 4.1-1.
- Frisoli, K. and Nugent, R. (2018). Exploring the effect of household structure in historical record linkage of early 1900s Ireland census records. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 502–509.
- Fu, Z., Christen, P., and Boot, M. (2011). Automatic cleaning and linking of historical census data using household information. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 413–420.
- Fu, Z., Christen, P., and Zhou, J. (2014). A graph matching method for historical census household linkage. In Tseng, V. S., Ho, T. B., Zhou, Z.-H., Chen, A. L. P., and Kao, H.-Y., editors, *Advances in Knowledge Discovery and Data Mining*, pages 485–496. Springer International Publishing.
- Hausdorff, F. (1914). *Grundzüge der Mengenlehre*. Leipzig, Von Veit.

- Heinze, G. (2006). A comparative investigation of methods for logistic regression with separated or nearly separated data. *Statistics in Medicine*, 25(24):4216–4226.
- Helgertz, J., Price, J., Wellington, J., Thompson, K. J., Ruggles, S., and Fitch, C. A. (2022). A new strategy for linking U.S. historical censuses: A case study for the IPUMS multigenerational longitudinal panel. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 55(1):12–29.
- Herzog, T. N., Scheuren, F. J., and Winkler, W. E. (2007). *Data Quality and Record Linkage Techniques*, volume 1. Springer.
- Moretti, D., Valentino, L., and Tuoto, T. (2019). Optimization routines for enforcing one-to-one matches in record linkage problems. *The R Journal*, 11(1):185.
- Nash, J. C. (2014). On best practice optimization methods in R. *Journal of Statistical Software*, 60(2):1–14.
- Nash, J. C. and Varadhan, R. (2011). Unifying optimization algorithms to aid software system users: optimx for R. *Journal of Statistical Software*, 43(9):1–14.
- Nash, J. C., Varadhan, R., and Grothendieck, G. (2022). *optimx: Expanded Replacement and Extension of the ‘optim’ Function*. R package version 2022-4.30.
- On, B.-W., Koudas, N., Lee, D., and Srivastava, D. (2007). Group linkage. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 496–505.
- Papadakis, G., Efthymiou, V., Thanos, E., and Hassanzadeh, O. (2022). Bipartite graph matching algorithms for clean-clean entity resolution: An empirical evaluation. In *Proceedings of the 25th International Conference on Extending Database Technology (EDBT)*, pages 462–474.
- R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Ruggles, S., Fitch, C. A., and Roberts, E. (2018). Historical census record linkage. *Annual Review of Sociology*, 44(1):19–37.
- Sadinle, M. (2017). Bayesian estimation of bipartite matchings for record linkage. *Journal of the American Statistical Association*, 112(518):600–612.
- Sadinle, M. and Fienberg, S. E. (2013). A generalized Fellegi–Sunter framework for multiple record linkage with application to homicide record systems. *Journal of the American Statistical Association*, 108(502):385–397.
- Sayers, A., Ben-Shlomo, Y., Blom, A. W., and Steele, F. (2015). Probabilistic record linkage. *International Journal of Epidemiology*, 45(3):954–964.
- Steorts, R. C., Hall, R., and Fienberg, S. E. (2016). A Bayesian approach to graphical record linkage and deduplication. *Journal of the American Statistical Association*, 111(516):1660–1672.
- Steorts, R. C., Ventura, S. L., Sadinle, M., and Fienberg, S. E. (2014). A comparison of blocking methods for record linkage. In Domingo-Ferrer, J., editor, *Privacy in Statistical Databases*, pages 253–268. Springer International Publishing.
- Tancredi, A. and Liseo, B. (2011). A hierarchical Bayesian approach to record linkage and population size problems. *The Annals of Applied Statistics*, 5(2B):1553–1585.
- Winkler, W. E. (1990). String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods*, pages 354–359. American Statistical Association.

Dimitris Karlis

Material list:

Iannario, M. and Dimitris, K. (2024) Dyadic multivariate mixture models for the analysis of elite swimmers. Unpublished paper.

Dyadic multivariate mixture models for the analysis of elite swimmers

Maria Iannario^{1*}† and Dimitris Karlis^{2†}

^{1*}Department of Political Sciences, University of Naples Federico II, Via L. Rodinò, 22, Naples, 80133, Italy.

²Department of Statistics, Athens University of Economics and Business, 76 Patission, str, Athens, 10434, Greece.

*Corresponding author(s). E-mail(s): maria.iannario@unina.it;

Contributing authors: karlis@aueb.gr;

†These authors contributed equally to this work.

Abstract

The analysis of sports data has attracted increasing attention; it ever more involves non-standard data structures from different sources, often in highly dimensional and complex formats. The most recent studies are based on the advantage of setting together information from different observed variables to infer underlying attributes or latent processes. Data based on objective performance are combined with information collected through questionnaires and based on the psychological evaluation of athletes. The inherent dependencies of the latter data, however, pose unique challenges. The authors propose a methodological approach that takes these dependencies into account and allows the variation in latent dyadic traits between athletes and coaches to be modelled parsimoniously. The authors propose a dyadic multivariate mixture model (DMMM) for measuring interactions of pairs of individuals when the responses to items on Likert scales represent the latent perceptions of each individual (actor) realised within the context of a dyad formed with another individual (partner). A study based on elite swimmers demonstrates the usefulness of the approach for three important areas of research: the strength of ties in the coach-athlete relationship, group differences in how the discussion of personal problems is related to the strength of bonds, and the evaluation of possible covariates related to personality traits affecting the study of latent dimensions.

Keywords: Athletes'/coaches' perception, dyadic data, latent variables, multivariate mixture models, ordinal data models

1 Introduction

The study of athlete data is of growing interest. The wide availability of objective measurement tools and performance data makes the study of these more and more interesting. However, not considering the performance indicators used to classify the results of athletes and teams, and positioning data involving non-standard data structures like movement-trajectories, there is still a lack of understanding of why and how certain behaviours emerge in performance contexts [20]. Recently, increasing interest has been devoted to understanding the psychological behaviour of certain athletes and how personality traits influence their performance (see [2], [17], [27], [30], among others), including through complex statistical models [5]; [10].

This data analysis work would be useful for professional analytics, enabling effective decision-making based on behaviour before and during competitions, improving the effects of training and in general the performance. A few papers focused on interpersonal relationships in athlete-athlete and coach-athlete dyads, studying their interdependence to improve outcomes [3]; [12]; [28].

The contribution proposed here fits into this strand. It aims to analyse the strength of ties in the coach-athlete relationship, group differences in how the discussion of personal problems is related to the strength of bonds, and the possible impact of demographic or personality trait-based variables on the relationship of the dyad. These objectives are addressed through the introduction of a mixture model that takes into account both the multiple items administered on ordinal scales by questionnaire and the dyadic data structure of the answers.

The analysis here reported concerns several observed items collected on the dyads (athlete, coach) on a 5-point Likert scale consisting of the below points – (1) Strongly Disagree; (2) Disagree; (3) Neither Agree nor Disagree; (4) Agree; (5) Strongly Agree. Data come from a survey concerning 100 couples collected in 2019 for the Statistical Modelling and Data Analytics for Sports project, which involved the University of Naples Federico II and the Italian Swimmer Federation (Campania Regional Committee). The survey gathered information on the psychosocial aspects that influence swimmers' performance with a focus on their relationship with their coaches.

The article is organised as follows: in Section 2, we introduce the modelling framework. In Section 3, we consider inferential issues regarding the EM algorithm used to obtain maximum likelihood (ML) estimation, selection criteria to identify the number of components and relevant covariates. Section 4 discusses the application of data analysis and Section 5 concludes the article.

2 Dyadic multivariate mixture model

2.1 The basic model

Consider the case that we observe p items for two actors (athlete, coach), called the pair. So, Y_{ijk} , $i = 1, \dots, n$, $j = 1, \dots, p$, $k = 1, 2$, is the response for item j of actor k for the i -th pair. Each response is expressed on a Likert scale with $r = 1, \dots, R$

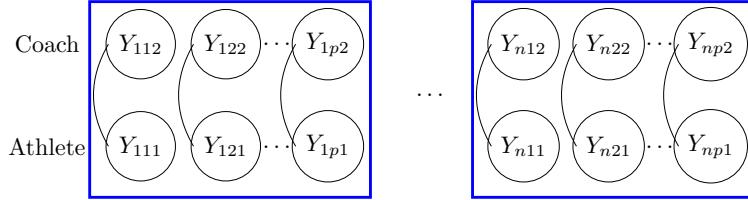


Fig. 1 Path diagram of the Dyadic multivariate mixture model.

categories. For the i -th pair the likelihood $L_i(\Theta)$ will be

$$L_i(\Theta) = \prod_{j=1}^p P(Y_{ij1}, Y_{ij2}; \theta_j),$$

where $P(Y_{ij1}, Y_{ij2}; \theta_j)$ is a bivariate discrete distribution defined through a copula (details in section 2.3) and $\theta_j = (\beta_{1j}, \beta_{2j}, \alpha_j)$ is a vector of parameters for the j -th item that contains the marginal parameters (β_{1j}, β_{2j}) for the two actors. More specifically, β_{mj} , $m = 1, 2$ are some regression coefficients from some marginal model for the item j specific to actor m of the pair, $m = 1, 2$ and the dependence parameter α_j that measures the association between the two actors for item j . In the next section we define this association with a copula and therefore α_j refers to the copula parameter(s). We assume that for each item we observe a different association; perhaps it is too restrictive to assume that all items show the same level of association. Finally, $\Theta = (\theta_1, \dots, \theta_p)$ represents the totality of the parameters.

The full likelihood for the data will be

$$\begin{aligned} L(\Theta) &= \prod_{i=1}^n L_i(\Theta) \\ &= \prod_{i=1}^n \prod_{j=1}^p P(Y_{ij1}, Y_{ij2}; \theta_j). \end{aligned}$$

Remark 1: The above specification assumes that there is independence across different items. So, we have dependence only within each item due to the pairs we observed but across items we have independence. We would like to introduce some association between the different items. The independence assumption will be relaxed later.

Remark 2: We further assume that the association is the same across pairs. This can be extended by assuming that it differs based on some covariate information about the pair. As an example consider that X_i is the age of the athlete at pair i . Then we may assume that the association parameter used above as α_j is not the same across pairs but it changes using the following relationship

$$g(\alpha_{ij}) = \gamma_j + \delta_j X_i$$

for some function $g(\cdot)$ to ensure the admissible support of the copula parameter. Here γ_j and δ_j are the regression coefficients for the j -th item. For $\delta_j = 0$ we have the reduced model with the same association parameter across pairs.

2.2 A finite mixture model

The model so far assumes no association between the items. This can be interpreted as independence conditional to some latent class [21]. Suppose that we have G classes. So, knowing the class the items are independent. We may extend by assuming a finite mixture model, namely Dyadic multivariate mixture model (DMMM), which presents for the i -th observation the likelihood

$$L_i(\Theta^*) = \sum_{g=1}^G \pi_g \prod_{j=1}^p P(Y_{ij1}, Y_{ij2}; \theta_{gj}). \quad (1)$$

Here $\pi_g > 0$, for $g = 1, \dots, G$, is the mixing proportion with $\sum \pi_g = 1$, i.e. the probability that a randomly selected pair belongs to the g -th latent class, and the parameter(s) θ_{gj} are now class specific. Also in this case, it is possible to have some regression type relationship and marginal parameters that can also depend on the class g . We assume that such marginal regression parameters are different across different classes.

The full model log-likelihood can have the form

$$\begin{aligned} \ell(\Theta^*) &= \sum_{i=1}^n \log L_i(\Theta^*) \\ &= \sum_{i=1}^n \log \left[\sum_{g=1}^G \pi_g \prod_{j=1}^p P(Y_{ij1}, Y_{ij2}; \theta_{gj}) \right] \end{aligned}$$

where now $\Theta^* = (\Theta_1, \dots, \Theta_G)$.

Note that we assume that G is known, but perhaps this may also be of interest to estimate, disclosing how many classes we have. For the problem of finding the value of G , namely the number of classes, see [26]; [32]; [8], among the others.

Being a finite mixture, estimation is feasible via an EM algorithm whose description is given in section 3. All other results related to finite mixtures (see [21]) apply here. The model can be seen as an extension of what is worked in [16].

Such a model, starting from the conditional independence of items given the class, can produce correlation between the items, and has the typical for finite mixture model inhomogeneity interpretation. Some pairs belong to some class (group/cluster) and hence they have a different behavior, which may be interpreted using some external covariates, like demographics or -as in the case study we deal with- some psychological respondents' latent traits.

In the next section we describe the marginal model specification.

2.3 A bivariate ordinal regression model using copulas

To help the exposition we drop the subscripts at the moment and we will introduce them again when necessary to describe the model.

- Suppose we have only one item. Let Y_i be an ordinal outcome (item) with R categories for the i -th observation. Also assume that we have available a vector of K covariates so $X_i = (X_{i1}, \dots, X_{iK})'$ for the i -th individual. Then $P(Y_i < r)$ is the cumulative probability of Y_i less than or equal to a specific category $r = 1, \dots, R-1$ yielding the cumulative model [19]. The odds of being less than or equal a particular category can be defined as

$$\frac{P(Y_i \leq r)}{P(Y_i > r)}$$

for $r = 1, \dots, R-1$ since $P(Y_i > R) = 0$. Then the log odds are

$$\log \frac{P(Y_i \leq r)}{P(Y_i > r)} = \text{logit}[P(Y_i \leq r)].$$

We further assume that

$$\text{logit}[P(Y_i \leq r)] = \beta_{r0} - \eta_1 x_{i1} - \dots - \eta_K x_{iK}.$$

It can be seen that a different intercept is considered for each category so β_{r0} is specific for the category r , while the regression coefficients η_r , $r = 1, \dots, K$, are the same for all categories (i.e. we are assuming the parallel assumption, see [1]). Hence the vector of parameters to be estimated in such a model is $\theta = (\beta_1, \dots, \beta_{R-1}, \eta_1, \dots, \eta_K)$, namely we have $R + K - 1$ parameters.

- Consider now a bivariate model, i.e. when we have two responses - say Y_{i1}, Y_{i2} for the i -th observation/pair -. We can model them through a copula whose model specification is based on a copula representation of the joint distribution of Y_{i1} and Y_{i2} .

A bivariate copula $C(u, v)$ is any function $C(u, v) : [0, 1]^2 \rightarrow [0, 1]$ which is a bivariate cumulative distribution function with uniform marginals. Theoretical details can be found in [22] to whose contribution we refer for a full discussion. The central idea is that one can join two univariate probability density functions to create a bivariate distribution with given marginals but also dependence. The choice of the copula function can result to different dependence structure. Usually the copula is defined as a cdf $C(u, v; \alpha)$ where α is some parameter measuring the association/dependence of the pair.

- Extensions to the multivariate case are simple but typically they may have problems to create flexible models. For continuous random variables the bivariate density can be derived simply by taking derivatives of the bivariate cumulative function. For discrete random variable finite differences are needed.

For our case concerning ordinal random variables, denote as $p_j(y) = P(Y_j \leq r)$, for $j = 1, 2$, the two marginal probability functions. Assuming a copula function $C(u, v)$

the bivariate probability mass function can be derived as

$$\begin{aligned} P(Y_1 = r_1, Y_2 = r_2) &= C(p_1(r_1), p_2(r_2); \alpha) - \\ &\quad C(p_1(r_1 - 1), p_2(r_2); \alpha) - \\ &\quad C(p_1(r_1), p_2(r_2 - 1); \alpha) + \\ &\quad C(p_1(r_1 - 1), p_2(r_2 - 1); \alpha) \end{aligned}$$

Hiring that $r \in \{1, \dots, R\}$, we assume that $P(Y \leq 1) = 0$ in the above notation, so for example $P(Y_1 = 0, Y_2 = 0) = C(p_1(0), p_2(0); \alpha)$. Of course in our case the marginal probability functions p_1 and p_2 are determined by some ordinal logistic regression model as defined above. Hence the bivariate model with copulas is also based on covariates. From the former definition and on the basis of an appropriate copula, we can easily calculate the probability and thus derive the likelihood for inference [see, e.g. 25]. In this contribution we present results based on the Galambos copula [11, p. 174] with

$$C(u, v; \alpha) = uv \exp \left[\{(-\log u)^\alpha + (-\log v)^\alpha\}^{-1/\alpha} \right], u, v \in [0, 1], \quad \alpha > 0.$$

The model described above is also reported in [9] based on the latent representation with underlying continuous random variables. Furthermore, as already mentioned in Remark 2, section 2.1, we may assume that the copula parameter α depends on some covariates. For example, in our case, we may assume that the magnitude of the correlation/dependence between coach and athlete depends on some characteristics, e.g. the duration of their relationship.

3 Inference

3.1 EM algorithm

In order to fit the model described in Section 2 we used the EM algorithm. Being a finite mixture the application of the EM algorithm is relatively simple, even though it is not possible to obtain a closed-form expression in the M-step.

Assume that $Z_i = (Z_{i1}, \dots, Z_{iG})$ are latent indicators such as $Z_{ig} = 1$ if the i -th observation comes from the g -th component and 0 otherwise. Then at the E-step we need to calculate the expectations of Z_{ig} given the data and the current estimates of the parameters. The algorithm proceeds as

E-step:
Calculate

$$w_{ig} = E(Z_{ig} | \text{data}, \Theta^*) = \frac{\pi_g \prod_{j=1}^p P(Y_{ij1}, Y_{ij2}; \theta_{gj})}{\sum_{g=1}^G \pi_g \prod_{j=1}^p P(Y_{ij1}, Y_{ij2}; \theta_{gj})}$$

using the current values of the parameters.

M-step: We need to update the parameters. For the mixing proportions we get that

$$\hat{\pi}_g = \frac{\sum_{i=1}^n w_{ig}}{n},$$

where the w_{ig} are those obtained from the E-step above, while for the parameters associated with each component we need to run a weighted likelihood with weights for the g -th component given by w_{ig} , for $i = 1, \dots, n$. So we update $\theta_g = (\theta_{g1}, \dots, \theta_{gp})$ by

$$\text{argmax}_{\theta_g} \sum_{i=1}^n w_{ig} \sum_{j=1}^p \log P(Y_{ij1}, Y_{ij2}; \theta_{gj}).$$

Since no closed form solutions exist we used typical numerical maximization¹.

The algorithm iterates between the E and M-steps until convergence. The latter is detected by the relative change between two successive log-likelihood values. Initial values for the EM algorithm can be obtained as follows:

- For the cases when $G = 2$, we just perturb the estimates from the model with one component (i.e. $G = 1$), by multiplying with 0.95 and 1.05, respectively, the estimated parameter of that simple model, while the mixing proportions are set equal to 0.5.
- For the case when $G > 2$ we use as initial values the estimates from the model with $G - 1$ components plus a new component with initial values equal to the mean of the other components and mixing proportion of 0.05. This mixing proportion was removed from the old components so as to sum to 1.

Having tested this approach we found that it works very well for simulated datasets.

3.2 Further inferential issues

Some more issues related to inference on the proposed model are the following:

Number of Components: One can use any of the criteria applied to typical finite mixture models including information based criteria like AIC or BIC. Among information criteria the Bayesian Information Criterion (BIC) [31] provides the most parsimonious solution.

Copula selection and its interpretation: As already mentioned the choice of the copula is quite important as different copulas give rise to different dependence structures and hence they can reveal important information about the underlying structure, see the discussion in [24]. For example, copulas that allow for upper tail dependence can fit better responses that concentrate more towards the upper values of the Likert scale. This is an interesting and important issue to consider in the context of such a complex treatment.

¹Specifically we used `optim` function in R.

Variable Selection: Since the DMMM uses covariates for the marginal models, i.e. for each item, but also for both members of the dyad, each covariate can be present in a large number of different regressions. This creates a large variable selection problem. It can be even more complicated in the sense that one can assume covariates in the mixing proportions [see for example, 7], as well as covariates in the copula parameters. This is a huge variable selection problem that is not dealt with in this paper but opens up the possibility of future studies. In the discussion here, the selection of variables follows some preliminary studies suggested in the literature and is based on the standard forward stepwise selection (see 4.2).

4 Elite swimmer application

4.1 Data

A sample of 100 elite swimmers enrolled in professional level registers has been selected within the Statistical Modelling and Data Analytics for Sports project. The latter conducted by the University of Naples Federico II and the Italian Swimmer Federation (Campania Regional Committee) aimed at gathering information on the psychosocial aspects that influence the performance of elite swimmers also in connection with the evaluations expressed by the coaches. The sample of elite swimmers has been randomly selected by coaches who in turn were sampled by the Italian Swimming Federation (Campania unit) list. The athletes answered several questions concerning their mental strategies and skills based on one of the main theoretical frameworks for analysing personality and coping behaviour in sport, i.e. the sport performance psychological inventory IPPS-48 [29]. The latter consists of 48 items in which respondents state how often they describe their sporting experience on a rating scale with six categories ranging from never to always. IPPS-48 factors assess both cognitive and emotional aspects relevant to athletes' performance measuring the following eight dimensions: Self-talk, Goal setting, Self-confidence, Emotional arousal control, Cognitive anxiety, Concentration disruption, Mental practice, and Race preparation. The questionnaire also collects some demographics, partially reported in Table 1, which may influence the response behaviour. Notably, age, gender, and the level of competitions are considered relevant predictors whereas other factors have not been extensively explored in sports literature.

Furthermore, athletes and coaches answer to $J = 7$ questions assessed by means of a rating scale with five categories ranging from total disagreement to total agreement about their perceived assessments on some topics related to the performance. The questions are named as Anxiety, Challenging, Improvement, Pressure, Recovery, Talent, and Workout². Responses are in Figures 2 and 3. Figure 2 shows the observed

² *Anxiety*: Athletes get anxious during performance; *Challenging*: The athlete performs better when the competition is more challenging; *Improvement*: The athlete has improved over the past year; *Pressure*: The athlete's performance is better when under pressure; *Recovery*: If the performance starts badly, the athlete has difficulty in recovering the right state of mind; *Talent*: The athlete's achievements are attributable to a natural talent that possesses; *Workout*: The athlete always follows precise and detailed pre-competition/match training.

Table 1 Demographics concerning the characteristics of the 100 athletes of the survey

Gender
Female (42) - Male (58)
Age
Min (10) - Max (30) - Mean (14.25)
Diet
No (48) - Flexible (44) - Rigid (8)
Engaged in a romantic relationship
No (72) - Yes (28)
Time spent with family
Never (21), Rarely (15), Sometimes (20), Always (33), Very Often (11)
Style
Dolphin crawl (9), Freestyle stroke (56), Backstroke (11), Breaststroke (24)
Years of competition
< 5 (38), 5 – 10 (33), 11 – 15 (21), > 15 (8)
Level of competition
International (9), National (22), Regional (69)
Official competitions organized by FIN
0 (4), 1 – 5 (11), 5 – 10 (15), 11 – 15 (16), > 15 (54)
Hours of training (per week)
< 10 (19), 10 – 15 (56), > 15 (25)
Numbers of podium
0 (49), < 5 (24), 5 – 10 (9), 11 – 15 (8), > 15 (10)
Coached an athlete or team
No (93) - Yes (7)
Practising other sports
No (80) - Yes (20)
Hours of training practising other sports (per week)
0 (74), 0 – 5 (16), 5 – 10 (1), 11 – 15 (5), > 15 (4)
Distance covered
50 mt (14), 100 mt (46), 200 mt (16), 400 mt (11), 800 mt (3), 1500 mt (10)

frequencies for all the 100 pairs for each question (item) separately. There it is possible to see the differences between the responses. Figure 3, instead, shows the joint frequencies as an attempt to see the correspondence between the responses.

Mental skills evaluated using the IPPS-48 are summarized in Table 2.

Some further details and analyses are also given in [5]; [10].

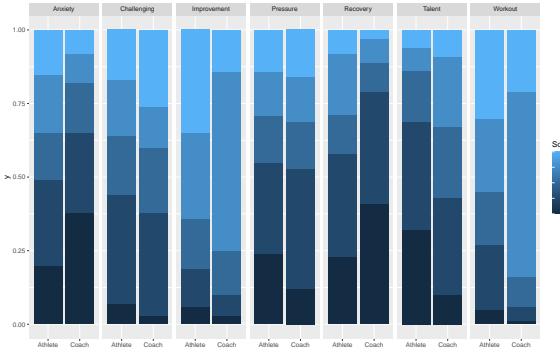


Fig. 2 Observed relative frequencies for all the questions (items) of interest.

Table 2 Descriptive statistics of the IPPS-48 factors

Constructs	Items	Mean	Sd.Dev
Cognitive anxiety	6	3.652	1.202
Self confidence	6	4.282	1.102
Concentration disruption	6	2.197	0.915
Emotional arousal control	6	3.758	0.941
Goal setting	6	4.205	1.091
Race preparation	6	3.838	0.937
Self talk	6	3.880	1.343
Mental practice	6	3.767	0.992

4.2 Fitting the DMMM

We are using $J = 7$ 5-points scale variables observed for both the athletes and the coaches as responses. Focusing on mental skills, for each response we consider $K = 8$ candidate variables (see Table 2). As a starting point we fitted a model with just $G = 1$ component, which corresponds to the independence. We used both Frank's and Galambos' copula, but we report Galambos' results because Frank's copula results are very similar and the differences observed are minimal³.

As already mentioned the problem of variable selection is a huge one for a complex structure like the one we are dealing with in this study. Since the interest lies on examining the effect of the regressors to the whole set of responses.

For the evaluation of the most relevant covariates we used a forward stepwise methodology selecting from the above list of covariates in Table 2. Using the AIC as criterion to enter new variables in the model we ended up with a model with 6 covariates. To clarify, the inclusion of one variable was done with respect all the responses and not per response; so we included one variable in the model when the

³All results are available if requested.

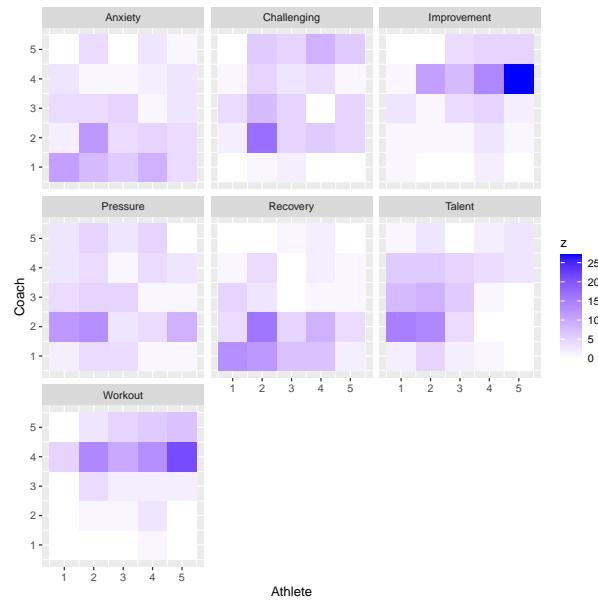


Fig. 3 Heat-map for the frequencies for each of the variables for pairs of athletes and coaches.

variable was included in all the responses (this imply 14 additional parameters) and the AIC was improved.

Then we fitted a 2-finite mixture model for the selected model. The log-likelihood improved a lot (from -1792.67 for the model with $G = 1$ to -1652.96 when $G = 2$). The two components were well separated. The estimated parameters are in Table 3. What it is interesting is that for some covariates and responses the sign of the coefficient is different between coaches and athletes implying a different point of view inside the pair.

Table 3 Estimated Parameters for the 2-finite mixture.

		1st Component						2nd Component					
		Cognitive Anxiety	Self Confidence	Concentration disruption	Emotional control	Goal setting	Race Preparation	0—1	1—2	2—3	3—4		
Anxiety	Athlete	1.729	-0.523	-0.108	0.226	0.413	-1.026	-0.023	2.177	3.323	5.624		
	Coach	0.064	-0.807	0.653	-0.488	0.326	0.681	-2.605	0.076	1.302	2.400		
Recovery	Athlete	0.845	-0.516	-0.293	-0.265	0.416	-0.164	-1.915	0.638	1.357	3.053		
	Coach	-0.078	-0.920	0.663	-0.059	0.139	0.812	-2.361	1.458	2.407	4.417		
Pressure	Athlete	0.241	0.019	0.563	0.593	-0.122	-0.415	1.425	2.638	3.374	3.811		
	Coach	0.333	0.111	-0.622	0.199	-0.152	-0.175	-1.290	2.418	2.420	28.548		
Challenging	Athlete	-0.158	-0.356	-0.764	0.786	0.913	0.531	1.226	4.458	5.933	6.843		
	Coach	0.382	0.465	-0.251	-0.066	0.106	-0.633	-2.055	2.032	3.047	3.548		
Talent	Athlete	-0.600	0.659	0.635	-0.017	-0.139	0.788	3.551	5.556	7.526	34.851		
	Coach	0.521	0.252	-0.351	0.200	0.275	-0.053	2.458	4.603	6.678	8.318		
Workout	Athlete	0.774	-0.432	-0.106	1.194	0.589	-0.409	2.832	5.292	6.282	7.421		
	Coach	0.255	-0.499	-0.484	0.113	0.203	-4.509	-2.771	-1.522	1.565			
Improvement	Athlete	0.440	1.583	-0.199	-0.233	-0.447	-0.141	1.161	2.605	3.899	5.677		
	Coach	0.429	0.216	-0.260	0.120	0.186	-0.071	-0.328	1.430	2.637	6.886		

Table 4 Estimated copula parameters for each pair and each component.

	1st comp	2nd comp
Anxiety	0.163	0.335
Recovery	0.152	0.091
Pressure	0.276	0.125
Challenging	0.784	0.502
Talent	0.366	0.453
Workout	0.182	0.303
Improvement	0.916	0.215

Table 4 presents the estimated copula parameters for each pair and each component. It can be seen that these are not values close to zero which would imply the absence of correlation. Note also that for some responses, the magnitude of the parameters is different, showing a different dependency in the two clusters.

4.3 Goodness of fit

A plot of the implied probabilities for each item is shown in Figure 4. Based on the fitted model and the estimated parameters, the probabilities for each paired response (for athlete and coach) have been computed. We then derived the margins and estimated how many responses we expected based on the model. We plot each component separately to make the distinction between the two components of the mixture visible.

For example looking on the *Anxiety* response it can be seen that the athletes of both components are quite close, but the coaches differ greatly. For the second component, coaches answered values for the first part of the scale much more often than for the 1st component. The differences are much larger for coaches than for athletes in all answers. For the athletes the *Recovery* response has the higher difference. To quantify the differences between the two components we calculated a simple mean absolute deviation distance between two distributions. Namely consider two discrete distributions $\mathbf{f} = (f_1, \dots, f_R)$ and $\mathbf{q} = (q_1, \dots, q_R)$ as

$$MAD(\mathbf{f}, \mathbf{q}) = \frac{1}{R} \sum_{i=1}^R |f_i - q_i|.$$

Table 5 shows the distance for each question (item) between the two components of the mixture.

For an overall goodness of fit of the model we report Figure 5 where for each pair we calculated the expected frequency taking into account the two components and the mixing proportion. Figure 5 shows how well the model fitted the data, in the sense that what we expected from the model is close to what we observed. Running individual χ^2 test of goodness of fit we never reject the null hypothesis for any of the 14 items (7 responses for each dyad). The same when considering the paired responses. An overall goodness of fit test is not possible due to the small sample size and the huge number of cells representing a study element for subsequent work. Figure 5 however provides evidence of this.

Table 5 MAD distance between the two components for athletes and coaches. Coaches have much larger distance for all the questions (items)

	Athlete	Coach
Anxiety	0.0291	0.1967
Recovery	0.0811	0.2232
Pressure	0.0837	0.2616
Challenging	0.0676	0.2693
Talent	0.0291	0.1543
Work-out	0.0347	0.1299
Improvement	0.0455	0.1337

4.4 Variable Selection issues in DMMM

Variable selection in finite mixture of regressions is more demanding than standard variable selection in simple regression setting. The reason is that the contributions of variables to the response may vary from one component to another of the mixture model. Hence the total number of possible combinations increases too fast, creating a complex and challenging variable selection problem. For a detailed description of the problem see [15]. They also introduced a penalized likelihood approach for variable selection in FMR models that also took into account the mixture structure while the approach reduced the computational effort.

Here we use a different approach. Since we are mostly interested on the variables that contribute to the dyad we want to see whether a variable contributes to the entire set of responses rather one by one. For this reason, for the regression finite mixture model, we removed one variable at a time and fitted the model without this variable.

Note that each variable contributed to the 7 responses, two dyads and the 2 components, so we had 28 parameters associated to that variable only. So, removing one variable we reduced the number of parameters by 28. One can see the effect in the log-likelihood when a variable is removed in Table 6. The reported p-value is the one of the associated likelihood ratio test with 28 degrees of freedom.

Cognitive anxiety and *Self Confidence* are the two most important variables with a significant change in the log-likelihood if removed. For the rest the changes are not very large.

Table 6 Likelihood Ratio test statistics for each covariate. The log-likelihood refers to what we get if we remove the covariate from the model. The p-value refers to whether all the regression coefficients of that variable are all equal to zero, and hence the variable has not any contribution.

Variable	Log-lik	LRT	p-value
Cognitive Anxiety	-1714.5798	123.2425	0.0000
Self Confidence	-1682.0423	58.1675	0.0007
Concentration disruption	-1664.6732	23.4293	0.7113
Emotional control	-1666.5418	27.1665	0.5092
GoalSetting	-1663.4180	20.9190	0.8287
Race Preparation	-1666.5766	27.2360	0.5054
Full model	-1652.9586	-	-

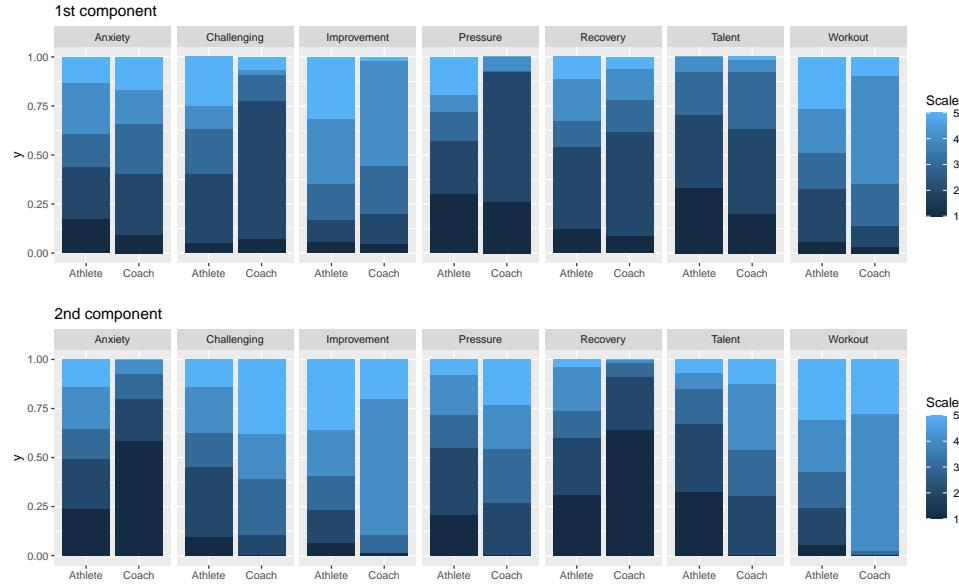


Fig. 4 The probabilities of all 5-points scales derived from the fitted model. We plot separately the two components of the mixture.

4.5 Cluster selection

One further aspect of a finite mixture model is to verify whether the identified clusters can have any interpretation. For this reason and using maximum a posteriori probability, readily available when the EM algorithm finishes through the w_{ij} weights, we assigned each pair to one of the two clusters. We also note that while we run with larger value for $G > 2$, the improvement of the log-likelihood was not large, while the model became very complicated with a huge number of estimated parameters and too difficult to be interpreted and provide insights. For this reason we comment here the case with $G = 2$.

4.6 Cluster description

Cluster 1 is characterized by more severe coaches (in the sense that they are more demanding from their athletes and they judge them more strictly). They score higher the variables about *Anxiety* and *Workout* and lower those about *Talent*, *Pressure*, *Improvement*, *Recovery* and *Challenging*, implying that they have a more strict opinion

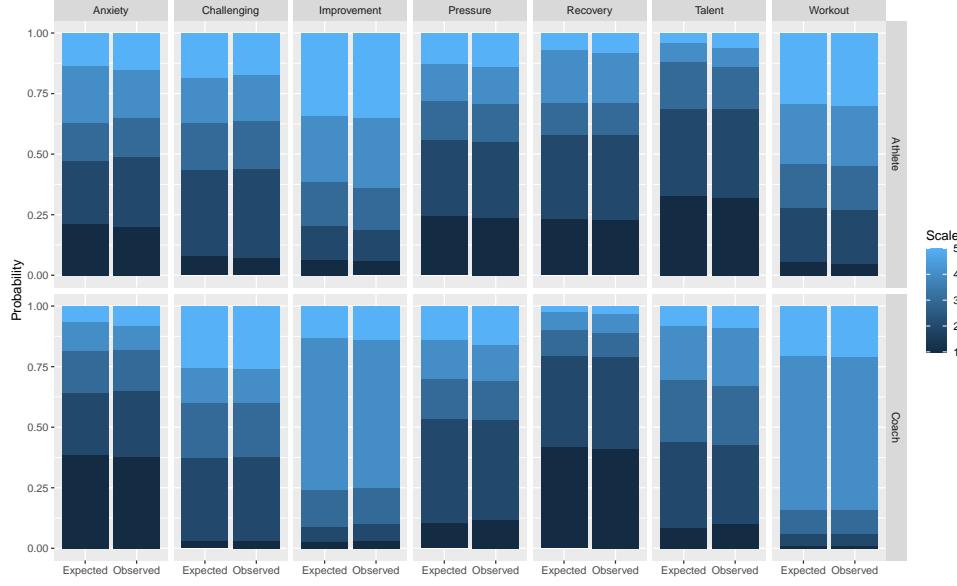


Fig. 5 Goodness of fit for the model. We estimated the expected frequencies for each of the response. Here we plot the expected frequencies next to the observed frequencies to see their closeness. Overall, the expected frequencies are very close to those observed.

about their athletes. For athletes, the differences between the two clusters are smaller, we can only say that the highest score is in the *Recovery* item. Also trying to connect the demographic variables presented in Table 1 we found that cluster 1 has those dyads that participated in more official competitions, practicing other sport and finally participate in larger distances like 1500 meters which can be described with those more involved in competition. Note that only those three demographics were found significant, recall the small sample size ($n = 100$).

5 Concluding Remarks

Recent studies analysing data based on objective sports performance with information collected via questionnaires and based on the psychological assessment of athletes constitute the frame of reference of this contribution. In this proposal, the inherent dependencies of the data also based on joint evaluations of a dyadic nature (coach and athlete) allowed for the introduction of an innovative methodological approach. Indeed, we introduced a dyadic multivariate mixture model (DMMM) including individual and

dyad-level latent traits; modelling larger groupings than dyads and including covariates for explaining the connections between personality traits and the relationship between the dyad.

The initial proposals for the methodology addressed therein refer to the Social Relations Model (SRM) - see [33]; [14] -. Here the specific behaviour of an actor when paired with a partner depends on a dyad-level composite latent variable consisting of three parts: (i) a latent trait at the individual level that reflects a general inclination of the actor to a specific behaviour when paired with a partner, (ii) a latent trait of the partner to elicit such behaviour, and (iii) a dyad-level latent trait characterising the effect of the direct relationship between the two parties on the actor's behaviour, independent of the two individual-level latent traits. The subsequent dyadic Item Response Theory model introduced in [6] extends the SRM to the situation where the perception of the actor is a latent variable measured by multiple indicators (items). The latter is an alternative to the multivariate SRM (e.g., [13]; [4]; [23]) which corresponds to a set of univariate basic SRMs with additional correlations of individual-level and dyad-level latent traits across items. Both the proposals are challenging because require the estimation of a large number of parameters and sometimes present an interpretative difficulty due to the complexity of the results and relationships. The DMMM overcomes some of these limitations; it incorporates the key features of both the complete SRM and dyadic Item Response Theory (IRT) model including individual and dyad-level latent traits, and including covariates for explaining the connections between personality traits and the couple relationship.

The proposal therein also lends itself to being:

1. Extended to higher dimensions (modelling larger groupings than dyads). The probability mass function calculation implies 2^d differences but this is pretty stable if the number of levels is not huge.
2. Implemented in the factor models context. Namely it is very common in surveys, to deal with datasets with large number of items (ordinal variables) that are naturally divided into subgroups, in such, each group of items has homogeneous dependence. Factor models are a unified tool for the analysis of such datasets with dependence coming from a few latent variables/factors. [24] describe such a model with copulas. We emphasize that all such models are univariate. Here we observed each item twice, that we would like to join together. So we need to extend the model in the bivariate case.
3. Structured to meet a complex need concerning a full variable selection approach. The latter implies that the regression models in different sub-populations may use different subsets of predictor variables (covariates) to explain the response variable. If the memberships of the observations are unobserved, then we naturally have a finite mixture model of linear regressions, where each mixture component is a Generalized Linear Model (GLM) with its own subset of predictor variables. In our case due to the large number of responses this becomes even more complicated. For example having 8 candidate variables one can see that we have 2^8 models for each response and hence in total 7×2^8 possible models. Possible penalisation techniques also involve a complex contribution based on non-standard structures.

An approach to clustering that explicitly considers multidimensional models but also assumes that the response is affected by dependence in the responses within a dyad is for the first time developed in a context such as sports performance by addressing some assessments of the psychological perception and characteristics of athletes that will certainly contribute to the studies proposed in this area.

The main findings here highlight some common behaviours among athletes, their self-assessments always being underestimated compared to those of their coaches, and variables related to psychology as a driver. Associated with these are socio-demographic variables that allow for the determination of clusters of respondents, highlighting those more involved in competition.

The paper has some limitations due to computational complexity, but it allows - for the first time - to combine different elements to address the evaluation of subjective perception in the sporting context, opening up a new study opportunity.

Acknowledgments. This research was carried out in the context of the project “Statistical Modelling and Data Analytics for Sports. Psychosocial aspects to assess the performance: the case of swimmers” (University of Naples Federico II—Italian Swimming Federation, Campania Regional Committee).

References

- [1] Agresti A (2010). *Analysis of Ordinal Categorical Data*, 2nd edition, J.Wiley & Sons, Hoboken.
- [2] Aidman E, Schofield G (2004). Personality and individual differences in sport. In T. Morris, & J. Summers (Eds.), Sport psychology: Theory, applications and issues, 2nd edition (pp. 22–47). Wiley.
- [3] Bell S T (2007). Deep-level composition variables as predictors of team performance: A meta analysis. *Journal of Applied Psychology*, 92(3), 595–615.
- [4] Card N A, Little T D, Selig J P (2008). Using the bivariate social relations model to study dyadic relationships: Early adolescents' perceptions of friends' aggression and prosocial behavior. In N. A. Card, T. D. Little, & J. P. Selig (Eds.), Modeling dyadic and interdependent data in the developmental and behavioral sciences (pp. 245–276). New York: Routledge.
- [5] Fabbricatore R, Iannario M, Romano R, Vistocco D (2023). Component-based structural equation modelling for the assessment of psycho-social aspects and performance of athletes. *AStA Advances in Statistical Analysis*, 107, 343–367.
- [6] Gin B, Sim N, Skrondal A. et al. (2020). A Dyadic IRT Model. *Psychometrika*, 85, 815–836.
- [7] Grun B, Leisch F (2008). FlexMix Version 2: Finite Mixtures with Concomitant Variables and Varying and Constant Parameters. *Journal of Statistical Software*, 28(4):1–35.

- [8] Hennig C, Liao TF (2013). How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 62(3):309–369.
- [9] Hirk R, Hornik K, Vana L (2019). Multivariate ordinal regression models: an analysis of corporate credit ratings. *Statistical Methods & Applications*. 28 pp. 507–539
- [10] Iannario M, Romano R, Vistocco, D (2023). Dyadic analysis for multi-block data in sport surveys analytics. *The Annals of Operations Research*, 325, 701–714.
- [11] Joe, H. (2014) Dependence Modeling with Copulas. CRC Press, New York
- [12] Jowett S, Nezlek J (2012). Relationship interdependence and satisfaction with important outcomes in coach-athlete dyads. *Journal of Social and Personal Relationships*, 29, 287–301.
- [13] Kenny DA (1994). Interpersonal perception: A social relations analysis. New York: Guilford.
- [14] Kenny D A, La Voie L (1984). The social relations model. *Advances in Experimental Social Psychology*, 18, 141–182
- [15] Khalili A, Chen J (2007). Variable selection in finite mixture of regression models. *Journal of the American Statistical Association*, 102, 1025–1038.
- [16] Kosmidis, I. and Karlis, D. (2016) Model-based clustering using copulas with applications. *Statistics and Computing*, 26, 1079-1099
- [17] Laborde S, Allen M S, Katschak K, Mattonet K, Lachner N (2020). Trait personality in sport and exercise psychology: A mapping review and research agenda. *International Journal of Sport and Exercise Psychology*, 18(6), 701–716.
- [18] Li J, Chen B, Zhang Y (2021). Adopting evaluative conditioning to improve coach-athlete relationships. *Frontiers in Psychology*, 12:751990.
- [19] McCullagh P (1980). Regression models for ordinal data (with discussion), *Journal of the Royal Statistical Society, Series B*, 42, 109–142.
- [20] McGarry T (2009). Applied and theoretical perspectives of performance analysis in sport: Scientific issues and challenges. *International Journal of Performance Analysis in Sport*, 9(1), 128–140.
- [21] McLachlan G, Peel D (2000). Finite Mixture Models. Wiley, New York
- [22] Nelsen RB (1999). An Introduction to Copulas. Springer, New York.

- [23] Nestler S (2018). Likelihood estimation of the multivariate social relations model. *Journal of Educational and Behavioral Statistics*, 43, 387–406.
- [24] Nikoloulopoulos A, Joe H (2015). Factor copula models for item response data. *Psychometrika*, 80, 126–150.
- [25] Nikoloulopoulos A (2013). Copula-based models for multivariate discrete response data. *Copulae In Mathematical And Quantitative Finance: Proceedings of the Workshop held in Cracow, 10-11 July 2012*, 231-249.
- [26] Nylund KL, Asparouhov T, Muthén BO (2007). Deciding on the number of classes in latent class analysis and growth mixture modeling: a Monte Carlo simulation study. *Structural Equations Modeling*, 14(4):535–569.
- [27] Piepiora P (2021). Assessment of personality traits influencing the performance of men in team sports in terms of the big five. *Frontiers in Psychology*, 12:679724.
- [28] Rhind D J A, Jowett S (2011). Working with coach–athlete relationships: Their quality and maintenance. In S. Mellalieu, & S. Hanton (Eds.), Professional practice in sport psychology: A review (pp. 219–248). Routledge.
- [29] Robazza C, Bortoli L, Gramaccioni G (2009). L'inventario psicologico della prestazione sportiva (IPPS-48). *Giornale Italiano di Psicologia dello Sport*, 4, 14–20.
- [30] Shuai Y, Wang S, Liu X, Kueh YC and Kuan G (2023). The influence of the five-factor model of personality on performance in competitive sports: a review. *Frontiers in Psychology*, 14:1284378.
- [31] Schwarz G (1978). Estimating the dimension of a model. *Annals of Statistics*, 6, 461–464.
- [32] Tofghi D, Enders CK (2008). Identifying the correct number of classes in growth mixture models. In: Hancock GR, Samuelsen KM (eds) Advances in latent variable mixture models. Information Age, Charlotte, 317–341.
- [33] Warner R, Kenny D A, Stoto M (1979). A new round robin analysis of variance for social interaction data. *Journal of Personality and Social Psychology*, 37, 1742–1757.

Sylvia Frühwirth-Schnatter

Material list:

Grushanina M. and Frühwirth-Schnatter S. (2024) Dynamic mixture of finite mixtures of factor analysers with automatic inference on the number of clusters and factors. Unpublished manuscript.

Dynamic Mixture of Finite Mixtures of Factor Analysers with Automatic Inference on the Number of Clusters and Factors

Margarita Grushanina* and Sylvia Frühwirth-Schnatter†

Abstract. Mixtures of factor analysers represent a popular tool for finding structure in data. While in many applications the number of clusters and latent factors within clusters is held constant, some recent models automatically infer cluster and/or factor dimensionalities. This is done by employing nonparametric priors and allowing the number of clusters and factors to potentially be infinite. MCMC estimation is performed via adaptive algorithms, where parameters associated with the redundant factors are discarded. The current work contributes to the literature by allowing automatic inference on the number of clusters and cluster-specific factors while keeping both dimensions finite. For automatic inference on the cluster structure we employ the dynamic mixture of finite mixtures model with a prior on the number of mixture components. Automatic inference on cluster-specific factors is performed by assigning an exchangeable shrinkage process (ESP) prior which can be interpreted as a generalized cumulative shrinkage process (CUSP) prior for the columns of the factor loading matrices. Extensive simulation studies and applications to benchmark as well as real data sets demonstrate that our model outperforms competing alternatives, in particular based on the multiplicative gamma process prior, with respect to recovering the correct number of cluster-specific factors.

MSC2020 subject classifications: Primary 62H25, 62H30; secondary 62F15, 62G05.

Keywords: factor analysis, hierarchical model, adaptive Gibbs sampling, spike-and-slab prior, Dirichlet prior, finite mixture models, Indian buffet process, Pitman-Yor process prior, telescoping sampling.

1 Introduction

Mixtures of factor analysers (MFA) models combine clustering with local dimensionality reduction performed separately in each cluster and are particularly useful for modelling data with complex and nonhomogeneous structure. First works involving MFA models appeared already in the 1990s when Ghahramani and Hinton (1996) developed an expectation-maximization (EM) algorithm for inference on the parameters of such a model. This algorithm was further developed for the purpose of clustering high-dimensional data in McLachlan et al. (2003), see also McLachlan and Peel (2000,

*Department of Epidemiology and Biostatistics, School of Public Health, Imperial College London, London, United Kingdom, m.grushanina@imperial.ac.uk

†Institute for Statistics and Mathematics, Vienna University of Economics and Business, Vienna, Austria, sylvia.fruehwirth-schnatter@wu.ac.at

Chapter 9). A Bayesian treatment of MFA via a variational approximation was first considered by [Ghahramani and Beal \(2000\)](#). At the same time, [Fokoue \(2000\)](#) discussed an algorithm for MCMC inference of an MFA model which was further ameliorated in [Fokoue and Titterington \(2003\)](#). The following years have seen a fair amount of literature on various versions of MFAs. The most notable ones include [McNicholas and Murphy \(2008\)](#), who assessed an MFA model in the context of parsimonious Gaussian mixture models, and [Viroli \(2010\)](#), who introduced a mixture of factor mixture analysers (MFMA). The key feature of the MFMA model is that it assumes that the data are generated according to several factor models with a certain prior probability (thus performing a local dimension reduction at the first level), and that in each factor model the factors are described by a multivariate mixture of Gaussians (thus performing a global dimension reduction at the second level).

Determining the number of clusters and the number of cluster-specific factors has always been a challenging issue. Many authors treat both quantities as known while others, like [McNicholas and Murphy \(2008\)](#) and [Viroli \(2010\)](#), run their models for various number of components in the mixture and factors in the factor analytical part and use model selection criteria to choose the best fitting model. In an early attempt to find a way to learn the model dimensions from data, [Fokoue and Titterington \(2003\)](#) developed a Birth-and-Death MCMC algorithm which uses the fact that the posterior distributions of both factor models and finite mixture models are invariant to permutations of the order of their parameters and thus the collection of their parameters can be viewed as a point process. More recently, [Papastamoulis \(2018\)](#) introduced an overfitting Bayesian MFA (BMFA), which estimates the unknown number of mixture components assuming a fixed number of factors. The optimal number of factors is then determined using information criteria.

The most flexible Bayesian MFA model to date is the infinite mixture of infinite factor analysers (IMIFA) model introduced in [Murphy et al. \(2020\)](#), which allows automatic inference on both the numbers of clusters and cluster-specific factors by assigning nonparametric priors to both cluster weights and cluster-specific factor loadings. For the automatic inference on the number of clusters they employ a Pitman-Yor process prior, using its stick-breaking representation and a slice sampler for MCMC estimation. Automatic inference on the cluster-specific number of factors is achieved with the multiplicative gamma process (MGP) prior of [Bhattacharya and Dunson \(2011\)](#) and an adaptive Gibbs sampler is used to facilitate estimation with varying dimensions. In fact, [Murphy et al. \(2020\)](#) offer an entire family of Bayesian MFA models which are briefly reviewed in Section 2. While this model class has the clear advantage of eliminating the need to predefine the model dimensions, this flexibility comes with certain shortcomings. These concern, for instance, subjective truncation criteria on which models based on the MGP prior are highly dependent as well as information loss when redundant cluster-specific factors are discarded in adaptive MCMC algorithms. At last, identification of the cluster-specific factor models remains an open issue for infinite MFA models.

In this paper we complement the IMIFA family and suggest an innovative way to specify a Bayesian MFA model which allows for the automatic inference on the number of clusters and cluster-specific factors. This is achieved by exploring a finite-dimensional

representation of infinite nonparametric priors. For the mixture part, we employ the dynamic mixture of finite mixtures (MFM) model, introduced in [Frühwirth-Schnatter et al. \(2021\)](#). This model puts a prior on the number of mixture components and allows inference with respect to the number of filled components which can be regarded as the number of clusters in the data. For each cluster-specific factor model, we generalize the cumulative shrinkage process (CUSP) prior introduced by [Legramanti et al. \(2020\)](#), which in its more general version becomes an exchangeable shrinkage process (ESP) prior. As shown by [Frühwirth-Schnatter \(2023\)](#), such a prior implicitly shrinks the factor loadings toward zero as the column index increases and allows inference on the number of active columns in the loading matrix which can be regarded as the cluster-specific number of factors. With this prior, the dimensionality of the cluster-specific factor model stays fixed while the number of active factors is random.

The rest of the paper is organised as follows. Section 2 introduces the general notion of a Bayesian MFA model, reviews the IMIFA model family and discusses how to extend this model class through alternative prior choices. Section 3 outlines the main contributions of the present paper, with details on how clustering is achieved through a dynamic MFM model given in Section 3.1 and the ESP prior on factor loadings explained in Section 3.2. Section 4 presents our four block MCMC algorithm, based on the telescoping sampler introduced in [Frühwirth-Schnatter et al. \(2021\)](#). The performance of our method is illustrated in an extensive simulation study in Section 5 and through real data applications in Section 6 and in the Supplementary material, Section E. The paper concludes in Section 7.

2 Bayesian MFA

In this section we provide a brief review of Bayesian MFA models and discuss how this model class can be extended. Given T observations $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_T)$ of a multivariate p -dimensional random variable, a BMFA model is formulated as follows. It is assumed that these observations can be grouped into K groups (clusters) and within each of these K clusters, labelled by $k = 1, \dots, K$, the variability of the observations can be described by H_k latent factors the number of which can vary across clusters. The model can be formalised in the following way. The variation of the observations within each cluster k is described by a cluster-specific factor-analytical model:

$$\mathbf{y}_t - \boldsymbol{\mu}_k = \boldsymbol{\Lambda}_k \mathbf{f}_t^k + \boldsymbol{\epsilon}_t,$$

where $\boldsymbol{\mu}_k$ is a p -dimensional vector of cluster-specific means, $\boldsymbol{\Lambda}_k$ is a $p \times H_k$ -dimensional cluster-specific factor loading matrix, \mathbf{f}_t^k is a H_k -dimensional vector of latent factors, and $\boldsymbol{\epsilon}_t$ is a p -dimensional vector of idiosyncratic errors. The idiosyncratic errors are assumed to follow a normal distribution with cluster-specific diagonal covariance matrices:

$$\boldsymbol{\epsilon}_t \sim N_p(\mathbf{0}, \boldsymbol{\Xi}_k), \quad \boldsymbol{\Xi}_k = \text{diag}(\xi_{1k}^2, \dots, \xi_{pk}^2). \quad (2.1)$$

This assumption implies that conditional on \mathbf{f}_t^k all p elements of \mathbf{y}_t are independent, so all dependencies between the variables are explained by the common factors \mathbf{f}_t^k . It is

usually assumed that the latent factors are orthogonal, namely:

$$\mathbf{f}_t^k \sim N_{H_k}(\mathbf{0}, \mathbf{I}_{H_k}), \quad (2.2)$$

and that \mathbf{f}_t^k , \mathbf{f}_s^k , $\boldsymbol{\epsilon}_t$ and $\boldsymbol{\epsilon}_s$ are pairwise independent for all $t \neq s$. The assumptions (2.2) and (2.1) imply that the data within each cluster arise from a multivariate normal distribution. Hence, we can formulate the following mixture model independently for each observation \mathbf{y}_t :

$$f(\mathbf{y}_t | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k, \boldsymbol{\Xi}_k) = \sum_{k=1}^K \eta_k N_p(\mathbf{y}_t; \boldsymbol{\mu}_k, \boldsymbol{\Omega}_k), \quad \boldsymbol{\Omega}_k = \boldsymbol{\Lambda}_k \boldsymbol{\Lambda}_k^T + \boldsymbol{\Xi}_k, \quad (2.3)$$

where $\boldsymbol{\Omega}_k$ denotes the cluster-specific covariance matrix of the data and η_k , ($k = 1, \dots, K$) are cluster weights with $\sum_{k=1}^K \eta_k = 1$. Note that this decomposition of $\boldsymbol{\Omega}_k$ into the sum of the cross-covariance matrix $\boldsymbol{\Lambda}_k \boldsymbol{\Lambda}_k^T$ and the idiosyncratic errors' covariance matrix is possible only under the assumptions (2.2) and (2.1).

The main challenge lies in establishing the values of K and H_1, \dots, H_K . Most of the frequentist MFA literature requires the model dimension to be pre-specified and usually information criteria are used for model choice. Recently, [Murphy et al. \(2020\)](#) have demonstrated that it is possible to recover both the number of clusters as well as the cluster-specific factor dimensions from the data in a one sweep algorithm within a Bayesian framework. To this aim, two classes of prior families, both well-known in Bayesian non-parametric inference, are introduced within the MFA framework, namely a Pitman-Yor process prior to learn the number of clusters and a multiplicative gamma process prior to learn the cluster-specific factor dimensions. These prior choices are briefly reviewed below and alternative choices are discussed.

2.1 Learning the number of clusters

To identify the number of clusters, [Murphy et al. \(2020\)](#) assign a Pitman-Yor process (PYP) prior to the mixture weights with the stick-breaking representation:

$$\eta_k = v_k \prod_{l=1}^{k-1} (1 - v_l), \quad v_k \sim \mathcal{B}(1 - \sigma, d + k\sigma), \quad (2.4)$$

where the reinforcement parameter σ and the concentration parameter d are required to satisfy the conditions $\sigma \in [0, 1]$ and $d > -\sigma$. The PYP prior apriori allows $K = \infty$ components in (2.3) and defines an infinite MFA model. For a given sample size T , only finitely many components are occupied and the number of clusters, denoted by K_+ , is defined as the number of non-empty components. Choosing $\sigma = 0$ yields an infinite MFA based on the Dirichlet process (DP) prior with concentration parameter equal to d . However, [Miller and Harrison \(2013\)](#) raise concerns about posterior consistency for the number of clusters for DP mixtures, even if recent work by [Ascolani et al. \(2023\)](#) suggests that putting a hyperprior on d might alleviate this problem.

It is well-known that both DP and PYP mixtures with $\sigma > 0$ imply that the number of clusters K_+ is increasing and diverges as T goes to infinity, with a rate depending on σ . In the present paper, we are interested in applications in which the data arise from a moderate number of clusters, even if the sample size increases. As alternatives to infinite MFAs, the overfitting MFA (OMFA) model suggested by [Papastamoulis \(2018\)](#) can be applied. In this specification of model (2.3), a large, but fixed K is combined with the symmetric Dirichlet prior $\eta_1, \dots, \eta_K \sim Dir_K(\frac{\alpha_M}{K})$ on the mixture weights. For appropriate choices of the hyperparameter α_M , a sparse mixture results, where $K_+ < K$ with high probability ([Frühwirth-Schnatter and Malsiner-Walli, 2019](#)).

Different prior choices on the mixture weights can be characterized and compared for given hyperparameters by the induced exchangeable partition probability function (EPPF), defined as the prior $p(\mathcal{C}|T)$ over all random partitions $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_{K_+}\}$ of the T observations into K_+ clusters, with each cluster \mathcal{C}_j containing $T_j > 0$ observations. An OMFA specification can be regarded as an “approximation” to an infinite MFA based on the DP prior, since the EPPF $p(\mathcal{C}|T, K, \alpha_M)$ of such a model converges to the EPPF of a DP mixture with concentration parameter α_M , given by the Ewens distribution $p_{DP}(\mathcal{C}|T, \alpha_M)$, as K increases. On the other hand, the EPPF of an OMFA model with a given value of K can be matched to the EPPF of a second family of PYP priors, where the reinforcement parameter $\sigma < 0$ and the concentration parameter is equal to $d = K|\sigma|$. In the corresponding stick-breaking representation, stick $v_K = 1$ and this PYP prior yields a mixture with infinitely many components, of which only K have non-zero weights, with the symmetric Dirichlet distribution $\mathcal{D}_K(|\sigma|) = \mathcal{D}_K(d/K)$ acting as prior on the weight distribution ([Gnedin, 2010; De Blasi et al., 2015](#)). Based on [Gnedin and Pitman \(2006\)](#), we find that the OMFA model of [Papastamoulis \(2018\)](#) with concentration parameter $\alpha_M > 0$ and K fixed and a Bayesian MFA model with a PYP prior on the mixtures weights with concentration parameter $d = \alpha_M$ and reinforcement parameter $\sigma = -\alpha_M/K$ imply the same EPPF and are equivalent regarding the implied prior on the partitions and the number of clusters.

In Section 3.1, we generalize the OMFA specification by putting a prior on the number of components K , while keeping the symmetric Dirichlet prior $Dir_K(\frac{\alpha_M}{K})$ as a conditional prior on the mixture weights, given K . This yields a so-called dynamic mixture of finite MFA, abbreviated as $(MF)^2A$, which is a special case of the class of generalized MFM introduced by [Frühwirth-Schnatter et al. \(2021\)](#). The latter paper derives many interesting properties of this model class, among them the EPPF $p(\mathcal{C}|T, \alpha_M)$ for any given prior $p(K)$. Based on these results, the EPPF of a dynamic $(MF)^2A$ reads:

$$p(\mathcal{C}|T, \alpha_M) = p_{DP}(\mathcal{C}|T, \alpha_M) \times \sum_{K=K_+}^{\infty} p(K) R(K, \alpha_M, K_+), \quad (2.5)$$

$$R(K, \alpha_M, K_+) = \prod_{j=1}^{K_+} \frac{\Gamma(T_j + \frac{\alpha_M}{K})(K - j + 1)}{\Gamma(1 + \frac{\alpha_M}{K})\Gamma(T_j)K},$$

where $p_{DP}(\mathcal{C}|T, \alpha_M)$ is the probability mass function (pmf) of the Ewens distribution. The paper by [Greve et al. \(2022\)](#) derives the induced prior on the number of clusters $p(K_+|T, \alpha_M)$ and other functionals from this EPPF which are helpful for a deeper understanding of the clustering behaviour of this specific Bayesian MFA model.

2.2 Learning the number of cluster-specific factors

While the OMFA model of [Papastamoulis \(2018\)](#) is able to learn the number of clusters, it pursues a traditional approach regarding the unknown factor dimensions H_1, \dots, H_K . They assume that $H_1 = \dots = H_K = H$ is fixed and work with the prior $\lambda_{ihk} | \phi_h \sim N(0, \theta_h), \theta_h \sim G^{-1}(g_0, h_0)$, $h = 1, \dots, H$, with a column specific scale parameter θ_h that is identical for all clusters. Traditional model selection criteria such as AIC, BIC and various variants of DIC are then used for selecting H .

The work of [Murphy et al. \(2020\)](#) provides a major improvement over this approach. They introduce a well-known shrinkage prior on the factor loadings to the framework of a Bayesian MFA, namely the multiplicative gamma process (MGP) prior ([Bhattacharya and Dunson, 2011](#)) which allows the factor loading matrix Λ_k in each cluster to have potentially infinitely many columns. More specifically, with $h = 1, \dots, \infty$ denoting the column index, the prior on the factor loadings λ_{ihk} in the i th row is formulated for $i = 1, \dots, p$ as follows:

$$\begin{aligned} \lambda_{ihk} | \phi_{ihk}, \tau_{hk}, \sigma_k &\sim N(0, \phi_{ihk}^{-1} \tau_{hk}^{-1} \sigma_k^{-1}), \quad \phi_{ihk} \sim G(\nu_1/2, \nu_2/2), \quad \tau_{hk} = \prod_{l=1}^h \delta_{lk}, \\ \delta_{1k} &\sim G(a_1, b_1), \quad \delta_{lk} \sim G(a_2, b_2), \quad l \geq 2, \quad \sigma_k \sim G(\rho_1, \rho_2), \end{aligned} \quad (2.6)$$

where δ_{lk} ($l = 1, \dots, \infty$) are independent and τ_{hk} is a shrinkage parameter for the h -th column of the k -th cluster-specific loading matrix Λ_k . The τ_{hks} are stochastically increasing under the restriction $a_2 > 1$, which favours growing shrinkage as the column index h increases. ϕ_{ihk} are local shrinkage parameters for the factor loadings λ_{ihk} in Λ_k and are designed to favour sparsity while also preserving non-zero loadings. Finally, σ_k is a global cluster-specific shrinkage parameter reflecting the belief that the overall degree of shrinkage varies across the clusters.

This MGP prior defines a mixture of infinite FA (MIFA) model and, in combination with the PYP prior on the mixture weights, yields the IMIFA model, while in combination with the overfitting mixture the OMIFA model results ([Murphy et al., 2020](#)). While this model class represents an important benchmark in nonparametric factor models, its implementation comes with certain challenges. First of all, to learn the cluster-specific factor dimensions H_k in the various components $k = 1, \dots, \infty$, a clear-cut decision which factors are active and which ones are inactive is needed. However, as for any continuous shrinkage prior, only soft thresholding is feasible for a MGP prior. Following [Bhattacharya and Dunson \(2011\)](#), factor h in cluster k is discarded during MCMC sampling, if a certain proportion of elements of the h -th column of the k -th cluster loading matrix Λ_k is within an ϵ_0 neighbourhood of zero. While this allows to learn possibly different cluster-specific factor dimensions H_k , the choice of the truncation criteria can be rather influential, as shown in [Schiavon and Canale \(2020\)](#). This leads to uncertainty, in particular, when working with data sets where there is no clear indication of the probable number of latent factors. Second, the hyperparameters a_1 and a_2 in (2.6) control at the same time the shrinkage rate and the prior for loadings on active factors, which creates a trade off between the need to maintain a rather diffuse prior for the active terms and to introduce shrinkage for the redundant ones. As illustrated in [Durante \(2017\)](#), this leads

to the problem that the efficient shrinkage conditions imposed on the hyperparameters provide too strong shrinkage in larger data sets.¹

For this reason, we consider alternative shrinkage priors for the factor loading matrices in a Bayesian MFA model in the present paper. One such family is the cumulative shrinkage process (CUSP) prior of Legramanti et al. (2020) which defines an alternative factor model with a priori infinitely many columns in Λ_k and largely corrects the drawbacks of the MGP prior. This prior has not yet been implemented for Bayesian MFA models, but can easily be adjusted to define an alternative MIFA model. The CUSP prior is a nonparametric prior on the variances of the elements of the factor loading matrix and represents a sequence of spike-and-slab distributions which assign growing mass to the spike as the model complexity grows. Active loadings are controlled by the slab parameters, while inactive loadings are controlled by the spike parameters. For any component k of a MIFA model, regardless whether the number of components K is finite or infinite, the prior on the factor loadings λ_{ihk} in the i th row of the h -th column of Λ_k is formulated for $i = 1, \dots, p$ as follows:

$$\begin{aligned}\lambda_{ihk} | \theta_{hk} &\sim N(0, \theta_{hk}), \\ \theta_{hk} | \pi_{hk} &\sim (1 - \pi_{hk})p_{\text{slab}}(\theta_{hk} | \phi_\theta) + \pi_{hk}\delta_{\theta_\infty},\end{aligned}\quad (2.7)$$

$$\pi_{hk} = \sum_{l=1}^h w_{lk}, \quad w_{lk} = v_{lk} \prod_{m=1}^{l-1} (1 - v_{mk}), \quad (2.8)$$

where $\pi_{hk} \in (0, 1)$ is the probability of the spike. Note that the definition of π_{hk} in (2.8) as a cumulative sum over w_{1k}, \dots, w_{hk} implies that the sequence $\pi_{1k}, \pi_{2k}, \pi_{3k}, \dots$ is increasing, pulling in this way all factor loadings a priori toward zero as the column index h increases, similar in spirit to the MGP prior. w_{lk} exhibits the usual stick-breaking representation of a DP prior (Sethuraman (1994)) where the v_{hk} are generated independently from a $\mathcal{B}(1, \alpha_C)$ -distribution. θ_{hk} is a column shrinkage parameter for the h th column of the cluster-specific factor loading matrix Λ_k , ϕ_θ denotes the hyperparameters of the slab distribution and θ_∞ is fixed in Legramanti et al. (2020) at 0.05.

The CUSP prior and its properties were further studied and generalised in Kowal and Canale (2023) and Frühwirth-Schnatter (2023) with applications, respectively, in the context of nonparametric functional bases as well as sparse Bayesian factor analysis. Their insights are exploited in the present paper in the context of Bayesian MFA. First of all, the construction of the cluster-specific spike probabilities $\pi_{1k}, \pi_{2k}, \pi_{3k}, \dots$ from the stick-breaking representation of a DP prior with concentration parameter α_C can be generalized by involving an arbitrary sequence of sticks v_{1k}, \dots, v_{Hk} , where H can be both finite or infinite. Second, the Dirac spike in (2.7) can be replaced by a continuous distribution $p_{\text{slab}}(\theta_{hk})$ without affecting the key properties of the prior, provided that the slab distribution is stochastically dominated by the spike distribution around 0 (Frühwirth-Schnatter, 2023, Proposition 2.2). The resulting spike-and-slab prior yields a generalized CUSP prior which can, in principle, be based on the stick-breaking representation of any finite or infinite mixture. The adaptation of the factor

¹Durante (2017) showed that the condition $a_2 > 1$ is not sufficient for efficient shrinkage and two more conditions, namely, $a_2 > b_2 + 1$ and $a_2 > a_1$, are required.

dimensionality H_k in each cluster k of an MFA model based on a generalized CUSP prior avoids the ambiguity of the MGP prior, since the "inactive" columns of Λ_k are identified as those which are assigned to the spike and H_k is defined as the number of "active" columns assigned to the slab.

An interesting class of generalized CUSP priors results from independent sticks $v_{hk} \sim B(a_h, b_h)$, $h = 1, \dots, H$, where the decreasing slab probabilities $\pi_{hk}^* = 1 - \pi_{hk}$ have a representation as a multiplicative beta process (Frühwirth-Schnatter, 2023):

$$\pi_{hk}^* = \prod_{\ell=1}^h (1 - v_{\ell k}) = \prod_{\ell=1}^h v_{\ell k}^*, \quad v_{hk}^* \sim B(b_h, a_h), \quad h = 1, \dots, H. \quad (2.9)$$

Various generalized CUSP priors can be applied for Bayesian MFA by specific choices of H and (a_h, b_h) , $h = 1, \dots, H$ in (2.9). Choosing $H = \infty$ in combination with a prior on the sticks, where a_h and b_h are constant yields new MIFA models. E.g., $a_h = 1$ and $b_h = \alpha_B$ as in Legramanti et al. (2020) or $a_h = \beta$ and $b_h = \beta\alpha_B$ as in Kowal and Canale (2023) yields MIFA models based on, respectively, a one-parameter (Teh et al., 2007) or a two-parameter (Ghahramani et al., 2007) Indian buffet process (IBP) prior. The MCMC procedure introduced by Legramanti et al. (2020) for the CUSP prior could, in principle, be extended to MIFA, however, such a sampling strategy quickly becomes computationally intensive. To perform MCMC for the CUSP prior, Legramanti et al. (2020) truncate the infinite representation (2.7) at a finite upper limit $H < \infty$ for the number of factors (which can be chosen adaptively during MCMC) by defining $v_{Hk} = 1$. Despite being an intrinsically binary classification problem, the assignment of the columns of the factor loading matrices into spike or slab requires in each cluster data augmentation based on H categorical variables z_{hk} , each with H realisations in $\{1, 2, \dots, H\}$, based on the discrete prior $Pr(z_{hk} = \ell | w_{\ell k}) = w_{\ell k}$, $\ell = 1, \dots, H$. The number of active columns in the cluster-specific loading matrix Λ_k is then defined as $H_k = \sum_{h=1}^H I\{z_{hk} > 0\}$. As a result, sampling the categorical indicators z_{1k}, \dots, z_{Hk} to perform classifications of all H columns of Λ_k into active or inactive ones, operates for each cluster on a separate $H \times H$ grid and requires H^2 density evaluations.

Frühwirth-Schnatter (2023) discusses an alternative (generalized) CUSP prior with finite H which can be used to define the (new) class of mixture of (finite) overfitting FA (MOFA). In the context of Bayesian MFA, this prior involves the following sticks,

$$v_{hk} \sim B(a_h, b_h), \quad h = 1, \dots, H, \quad (2.10)$$

where $a_h = 1$ and $b_h = \alpha_B(H - h + 1)/H$. For a wide range of hyperparameters α_B (which can be learned from the data), a MOFA specification results, where $H_k < H$ with high probability. As will be discussed in Section 3.2, prior (2.10) emerges as the CUSP representation of an exchangeable shrinkage process (ESP) prior on the slab probabilities which is invariant to permutations of the column index within each cluster. Working with the ESP prior, which is commonly applied in finite sparse Bayesian factor analysis, is advantageous from a computational points of view, in particular in the context of Bayesian MFA. Due to the exchangeability on the weights (rather than the sticks), classification of the columns into spike and slab involves data augmentation based on

H binary (rather than categorical) variables $\gamma_{1k}, \dots, \gamma_{Hk}$ and requires only $2H$ density evaluations in each cluster. Finally, it is evident that prior (2.10) converges to the CUSP prior of Legramanti et al. (2020) with the concentration parameter α_B as H increases. Hence, a MOFA model based on the finite generalized CUSP prior (2.10) offers an attractive alternative to the truncated version of the CUSP prior of Legramanti et al. (2020), in particular from a computational viewpoint.

3 Extending the IMIFA model family

Note that the clustering properties of any prior on the mixture weights hold, in general, independently of the chosen family of component distributions and, therefore, independently of the prior distribution on the component-specific parameters. Hence, for any Bayesian MFA, the prior on the factor loadings within each cluster can be chosen independently from the prior on the mixture weights and a range of new Bayesian MFA models can be constructed to expand the IMIFA model family introduced by Murphy et al. (2020) by combining the priors discussed, respectively, in Section 2.1 and Section 2.2. As mentioned earlier, any MIFA model can be based on the CUSP prior of Legramanti et al. (2020) instead of the MGP prior to learn the cluster specific factor dimensions H_k . In combination with the PYP prior for the number of clusters, this yields an alternative to the IMIFA model of Murphy et al. (2020) and avoids the drawbacks of the MGP prior. Efficient computation for this model class can be based on an infinite MOFA (IMOFA) model, where the PYP prior is combined with prior (2.10) with a finite, fixed H for learning the factor dimension in each cluster. As discussed, such a model can be regarded as a finite approximation of an IMIFA model based on the CUSP prior.

In the present paper, we focus on Bayesian MFA models with automatic inference on the dimensionality of both cluster and factor structure of the data, which, at the same time, allows to keep both K and H_1, \dots, H_K finite, while being random variables that are inferred from the data. To this aim, we employ the dynamic mixture of finite mixtures model to identify the cluster structure of the data. For the cluster-specific factor-analytical part of the model, we rely on the finite generalized CUSP prior introduced in (2.10) and exploit its relationship to the class of exchangeable shrinkage process (ESP) priors (Frühwirth-Schnatter, 2023). We call the resulting novel MFA model a mixture of finite mixtures of factor analysers, or dynamic (MF)²A for short. In addition to the reasons outlined in Section 2.1, this choice is also guided by computational considerations. Learning the number of clusters in a dynamic (MF)²A model gives the additional advantage of making the MCMC estimation possible solely within Gibbs sampler steps relying on telescoping sampling and eliminating the need to refer to additional methods such as slice sampling.

3.1 Dynamic mixture of finite mixtures of factor analysers

Let $k = 1, \dots, K$ denote the cluster index and $h = 1, \dots, H$ denote the indices of factors within a cluster. The dynamic mixture of finite mixtures (MFM) model is a mixture

model with a prior $p(K)$ on the number of mixture components K and can be written in the following hierarchical way ([Frühwirth-Schnatter et al. \(2021\)](#)):

$$\begin{aligned}
 K &\sim p(K), \\
 \eta_1, \dots, \eta_K | K, \alpha_{\mathcal{M}} &\sim Dir_K \left(\frac{\alpha_{\mathcal{M}}}{K} \right), \\
 \boldsymbol{\mu}_k | \mathbf{b}_0, \mathbf{B}_0 &\sim N_p(\mathbf{b}_0, \mathbf{B}_0), \quad \text{cluster means for } k = 1, \dots, K, \\
 S_t | \eta_1, \dots, \eta_K &\sim \mathcal{M}(1; \eta_1, \dots, \eta_K), \quad \text{latent allocation variables for } t = 1, \dots, T, \\
 \mathbf{y}_t | S_t = k, \boldsymbol{\mu}_k, \Omega_k &\sim N_p(\boldsymbol{\mu}_k, \Omega_k), \quad \text{for each data point in } 1, \dots, T. \\
 \Omega_k &= \boldsymbol{\Lambda}_k \boldsymbol{\Lambda}_k^T + \boldsymbol{\Xi}_k, \quad \text{cluster covariance matrices for } k = 1, \dots, K.
 \end{aligned} \tag{3.1}$$

The model is called "dynamic" due to the fact that the Dirichlet concentration parameter $\alpha_{\mathcal{M}}/K$ is inversely proportional to the number of components K , which favours more sparse solutions as the number of components grows. Under this model, the joint distribution of the data $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_T)$ has a representation as a countably infinite MFM with K p -variate Gaussian components:

$$p(\mathbf{y}) = \sum_{K=1}^{\infty} p(K) \prod_{t=1}^T \sum_{k=1}^K \eta_k N_p(\mathbf{y}; \boldsymbol{\mu}_k, \Omega_k).$$

In this framework, K is defined as the (theoretical) number of components in the mixture, while the number of clusters K_+ is defined as the number of filled components that generated the data, namely $K_+ = \sum_{k=1}^K \mathcal{I}\{|T_k| > 0\}$, where the set $T_k = \{t : S_t = k\}$ collects the indices of all observations generated by the component k and the cardinality $|T_k|$ is the number of such observations. Assigning a prior to K has the advantage that both K and K_+ are random a priori. Depending both on the number of observations T and the choice of hyperparameters, they can be close or rather different, see the detailed investigation in [Greve et al. \(2022\)](#). Having the Dirichlet concentration parameter decrease with increasing K allows a gap between K_+ and K and thus ensures randomness in the prior distribution of K_+ for a broad range of hyperparameters $\alpha_{\mathcal{M}}$.

The EPPF of a dynamic $(MF)^2A$ has been provided in [\(2.5\)](#). In the limiting case where the prior $p(K)$ increasingly concentrates all prior mass at $K = +\infty$, this model converges to an infinite MFA where clustering is based on a DP mixture. An OMFA model results as that special case where the prior $p(K)$ puts all prior mass on a fixed value $K = K_f$. For more general priors $p(K)$, the connection of the dynamic $(MF)^2A$ to infinite MFA based on the PYP prior is of special interest. The EPPF [\(2.5\)](#) arises from mixing a PYP prior over the reinforcement parameter $\sigma_K = -\alpha_{\mathcal{M}}/K$, while the concentration parameter $d = \alpha_{\mathcal{M}}$ is independent of K . Hence, the dynamic $(MF)^2A$ model represents a model beyond the class of Gibbs-type priors.

It remains to be discussed how to choose $p(K)$. Following the considerations in [Frühwirth-Schnatter et al. \(2021\)](#) and [Grün et al. \(2022\)](#), we chose the suggested translated beta-negative-binomial (BNB) prior $K - 1 \sim BNB(\alpha_{\lambda}, a_{\pi}, b_{\pi})$, which represents a hierarchical generalisation of the Poisson, the geometric and the negative-binomial

distributions. The p.m.f. takes the following form for $K = 1, 2, \dots$:

$$p(K) = \frac{\Gamma(\alpha_\lambda + K - 1)B(\alpha_\lambda + a_\pi, K - 1 + b_\pi)}{\Gamma(\alpha_\lambda)\Gamma(K)B(a_\pi, b_\pi)} \quad (3.2)$$

where $\alpha_\lambda > 0$, $a_\pi > 0$ and $b_\pi > 0$ are hyperparameters. The choice of hyperparameters can be governed by the desired value of the prior mean $E(K) = 1 + \alpha_\lambda b_\pi / (a_\pi - 1)$, which exists as long as $\alpha_\pi > 1$. An important advantage of this prior is that the three parameters α_λ , a_π and b_π allow simultaneous control over both the expectation of $p(K)$ and its tails, as well as the implied prior on K_+ and its expectation (see [Frühwirth-Schnatter et al. \(2021\)](#) and [Greve et al. \(2022\)](#) for details on the induced prior on K_+).

Since the hyperparameter α_M in the Dirichlet concentration parameter α_M/K plays an important role for the prior distribution induced on the number of filled clusters K_+ and the partitions, we learn it from the data by assigning a prior to it and updating it from the posterior distribution in a random walk Metropolis-Hastings step. We choose the F-distribution prior $\alpha_M \sim \mathcal{F}(\nu_l, \nu_r)$ as it is flexible enough to allow various cluster solutions by modeling the behaviour close to zero and in the tail independently (see [Frühwirth-Schnatter et al. \(2021\)](#) for further motivation of this prior choice).

3.2 ESP prior for factor loadings

[Frühwirth-Schnatter \(2023\)](#) introduces the general class of exchangeable shrinkage process priors, which take the form of unordered spike-and-slab priors. Adjusted for the MFA framework, this prior is defined as follows. For each cluster $k = 1, \dots, K$, assume $\tau_k = \{\tau_{hk} \in (0, 1) : h = 1, \dots, H\}$ to be a finite sequence of i.i.d. random probabilities. Let $\Theta_k = \{\theta_{hk}\}$, $h = 1, \dots, H$ be a finite sequence of column-specific shrinkage parameters assumed to be independent conditional on τ_k and independent of all τ_{lk} , $l \neq h$ for all h . If $p(\theta_{hk} | \tau_{hk})$ takes the following spike-and-slab form:

$$\theta_{hk} | \tau_{hk} \sim (1 - \tau_{hk})p_{spike}(\theta_{hk} | \phi_0) + \tau_{hk}p_{slab}(\theta_{hk} | \phi_\theta), \quad (3.3)$$

where ϕ_θ and ϕ_0 are the hyperparameters of the slab and the spike distributions, respectively, then Θ_k follows an exchangeable shrinkage process (ESP) prior. The prior is exchangeable in the sense that it is invariant to permutations of the column indices h within each cluster and to permutations of the cluster indices across clusters.

It is often assumed in the literature that the slab probabilities $\tau_{1k}, \dots, \tau_{Hk}$ follow a beta distribution, where the first parameter depends on H . H is here the same in all clusters and can be considered as the maximum possible number of factors, which the data allows. A typical choice of the beta prior for τ_{hk} would be

$$\tau_{hk} | H \sim \mathcal{B}\left(b_0 \frac{\alpha_B}{H}, b_0\right), \quad h = 1, \dots, H.$$

This prior was proposed in [Frühwirth-Schnatter et al. \(2024\)](#) in the context of sparse finite Bayesian factor models. For $H \rightarrow \infty$ it converges to the infinite two-parameter beta prior introduced by [Ghahramani et al. \(2007\)](#) in the framework of Bayesian non-parametric latent feature models. With $b_0 = 1$, this prior becomes the one-parameter

beta prior employed by [Ročková and George \(2016\)](#), which converges to the Indian buffet process prior for $H \rightarrow \infty$ (see [Teh et al. \(2007\)](#) for more details):

$$\tau_{hk}|H \sim \mathcal{B}\left(\frac{\alpha_B}{H}, 1\right), \quad h = 1, \dots, H. \quad (3.4)$$

It is shown in [Frühwirth-Schnatter \(2023\)](#) that any ESP prior admits a finite generalised CUSP representation. In the context of our Bayesian MFA model, it is obtained by the permutation of the columns index h of the parameters $\theta_{1k}, \dots, \theta_{Hk}$ according to the decreasing slab probabilities $\tau_{(1k)} > \dots > \tau_{(Hk)}$ for each cluster k . The CUSP prior of [Legramanti et al. \(2020\)](#), defined in (2.7), can be considered as the limiting case, with $H \rightarrow \infty$, of the exchangeable spike-and-slab prior (3.3) on $\theta_{hk}|\tau_{hk}$ where τ_{hk} arises from prior (3.4). For $H \rightarrow \infty$, the hyperparameter α_B coincides with the hyperparameter α_C of the stick distribution of the CUSP prior. However, while [Legramanti et al. \(2020\)](#) assume a fixed value $\alpha_C = 5$, we adapt α_B to data under a suitable prior, see the Supplementary material, Appendix A. Increasing spike probabilities π_{hk} for $h = 1, \dots, H$, as in the case of the CUSP prior, are obtained for each cluster from the decreasing order statistics $\tau_{(1k)} > \dots > \tau_{(Hk)}$ by defining $\pi_{hk} = 1 - \tau_{(hk)}$. Representation (3.3) allows to choose an upper limit for the number of factors in each cluster, H , and keep it fixed throughout the entire inference procedure. By performing classification between spike and slab independently for each column, we end up with defining an effective number of active factors H_k in each cluster k , which is random both a priori and a posteriori, typically smaller than H , and varies across clusters.

This relationship between the combined priors (3.3) and (3.4) and the CUSP prior holds regardless of the distributions of the spike and the slab, both of which are allowed to depend, respectively, on (random) hyperparameters ϕ_0 and ϕ_θ . Following [Legramanti et al. \(2020\)](#) and [Kowal and Canale \(2023\)](#), we combine the spike and slab distributions $p_{spike}(\theta_{hk}|\phi_0)$ and $p_{slab}(\theta_{hk}|\phi_\theta)$ with the conditionally Gaussian prior $\lambda_{ihk}|\theta_{hk} \sim N(0, \theta_{hk})$ for the factor loadings in row i in column h in cluster k . This allows to work out the marginal prior for the h th column $\boldsymbol{\lambda}_{hk} = (\lambda_{1hk}, \dots, \lambda_{phk})^\top$ of the k th cluster factor loading matrix for specific spike and slab priors. E.g., under the slab prior $\theta_{hk}|a_\theta, b_\theta \sim \mathcal{G}^{-1}(a_\theta, b_\theta)$, a Student- t distribution results for $\boldsymbol{\lambda}_{hk}$, i.e. $\boldsymbol{\lambda}_{hk}|a_\theta, b_\theta \sim t_{2a_\theta}(\mathbf{0}, b_\theta/a_\theta \mathbf{I}_p)$, while $\boldsymbol{\lambda}_{hk}|a_0, b_0 \sim t_{2a_0}(\mathbf{0}, b_0/a_0 \mathbf{I}_p)$ under the spike prior $\theta_{hk}|a_0, b_0 \sim \mathcal{G}^{-1}(a_0, b_0)$. Thus, the full specification of the prior on the factor loadings in each cluster of the MFA model can be formalised as follows:

$$\begin{aligned} \lambda_{ihk} | \theta_{hk} &\sim N(0, \theta_{hk}), \quad \text{where } i = 1, \dots, p, \quad h = 1, \dots, H \text{ and } k = 1, \dots, K, \\ \theta_{hk} | \tau_{hk} &\sim \tau_{hk} \mathcal{G}^{-1}(a_\theta, b_\theta) + (1 - \tau_{hk}) \mathcal{G}^{-1}(a_0, b_0), \quad \tau_{hk} \sim \mathcal{B}\left(\frac{\alpha_B}{H}, 1\right). \end{aligned}$$

$\xi_{ik}^2 \sim \mathcal{G}^{-1}(a_\xi, b_{\xi i})$, $b_{\xi i} \sim \mathcal{G}(a_g, b_{gi})$. Details on other priors and hyperparameters are given in the Supplementary material, Section A.

For MCMC estimation, we use the usual technique of data augmentation for ESP priors. We introduce KH latent binary indicator variables γ_{hk} , one for each column $h = 1, \dots, H$ of the loading matrix $\boldsymbol{\Lambda}_k$ in each cluster k , to classify the columns into “active” and “inactive” ones. The indicator γ_{hk} takes the value of either zero or one for each column $h = 1, \dots, H$ and follows the Bernoulli prior $P(\gamma_{hk} = 1 | \tau_{hk}) = \tau_{hk}$.

4 Posterior computations and MCMC algorithm

4.1 Nested Gibbs sampler

Despite the relatively complex nature of the model, with separate factor-analytical models nested within a cluster structure, posterior inference can be done mostly within Gibbs sampling steps. The sampler consists of four major blocks, where in the first block the partition is updated and the number K_+ of non-empty clusters is identified. In the second block the parameters of the factor model are updated for every filled cluster and the number of active factors H_k in each cluster $k = 1, \dots, K_+$ is identified via the non-zero elements in the corresponding columns of the binary indicator matrix $\gamma = \{\gamma_{hk}\}$, where by γ_k we denote the row of γ which corresponds to cluster k . In the same way, $\theta_k = \{\theta_{hk}, h = 1, \dots, H\}$ denotes the column-specific shrinkage parameters for the factor loadings in cluster k . In the third block, the new number of mixture components $K \geq K_+$ is sampled and the Dirichlet parameter α_M is updated via a random walk Metropolis-Hastings step. Finally, in the fourth block, we fill the empty clusters including the parameters of the underlying factor models from the corresponding priors. Thus, the first and the third blocks are the standard telescoping sampler clustering steps as described in [Frühwirth-Schnatter et al. \(2021\)](#). The full details of the sampler are provided in Algorithm 1. As for such an algorithm starting values can be rather influential in forming the chain path, they are carefully chosen to ensure that the sampler moves through all areas of high posterior support, see details in the supplementary material, Section B.1.

Note, that in Block 4, step (a) of Algorithm 1, Λ_k , Ξ_k , and $\theta_k = \{\theta_{hk}\}$ for the added empty clusters are sampled using, respectively, the hyperparameters $\mathbf{b}_\xi = (b_{\xi 1}, \dots, b_{\xi p})$, b_θ and b_0 learned in Block 2, step (b) of Algorithm 1 from the K_+ filled components. This is a specific feature of the telescoping sampler for MFMs developed in [Frühwirth-Schnatter et al. \(2021\)](#), which ensures that the hyperparameters of the filled components inform the hyperparameters of the empty components.

A separate factor-analytical procedure needs to be run in Block 2, step (a) for each of the filled clusters $1, \dots, K_+$ (see Algorithm 2). The first three steps are standard Gibbs sampler steps for factor models, with the first step used for updating the factors \mathbf{f}_t^k for all observations $t \in T_k$ assigned to cluster k . In the following two steps, factor loadings λ_{ik} in the i th row of Λ_k and idiosyncratic variances ξ_{ik}^2 are updated for $i = 1, \dots, p$. Since classification in Block 1, step (a) of Algorithm 1 is carried out marginalized w.r.t. the factors \mathbf{f}_t^k , it is important to update the factors in the first step of Algorithm 2. In this way, factors derived for observations currently assigned to a cluster are used in the subsequent steps of updating the cluster-specific factor loadings and idiosyncratic variances. In step 4 of Algorithm 2, cluster-specific means μ_k are sampled based on the observations assigned to cluster k and the updated parameters of the cluster-specific factor models.

The remaining steps deal with the classification of the columns of the cluster-specific factor loading matrices, $\{\lambda_{hk}\}, h = 1, \dots, H$, into "active" (assigned to the slab) and "inactive" (assigned to the spike). As already described in Section 3.2, this is done by introducing a latent binary indicator γ_{hk} for each column $h = 1, \dots, H$ of each matrix

Algorithm 1 Telescoping sampling for the dynamic $(MF)^2 A$ model

Block 1.

- (a) Update the partition \mathcal{C} of the data by sampling latent allocation indicators S_t for $t = 1, \dots, T$, from $Pr(S_t = k | \eta_K, \mu_1, \dots, \mu_K, \Omega_1, \dots, \Omega_K, K) \propto \eta_k N_p(\mathbf{y}_t; \mu_k, \Omega_k)$, $k = 1, \dots, K$.
- (b) Compute the number of observation points in each cluster $|T_k| = \#\{t | S_t = k\}$, the number of non-empty components $K_+ = \sum_{k=1}^K \mathcal{I}\{|T_k| > 0\}$, and relabel the components so that the first K_+ clusters are non-empty.

Block 2.

- (a) For each of the filled clusters $k = 1, \dots, K_+$ run the factor analytical procedure with the spike-and-slab prior on factor loadings, sequentially updating \mathbf{f}_t^k , Λ_k , Ξ_k , θ_k , γ_k and τ_k . Conditional on Λ_k and Ξ_k update cluster means μ_k for the filled clusters $k = 1, \dots, K_+$ (see Algorithm 2 for further details).
- (b) Update the hyperparameters $\mathbf{b}_\xi = (b_{\xi 1}, \dots, b_{\xi p})$, b_θ , b_0 and α_B of the factor-analytical model conditional on K_+ , θ_k , γ_k , Ξ_k and τ_k (see Algorithm 3 for further details).

Block 3.

- (a) Conditional on the partition \mathcal{C} , draw a new value of $K \geq K_+$ from

$$p(K | \mathcal{C}, \alpha_M) \propto p(K) \frac{\alpha_M^{K_+} K!}{K^{K_+} (K - K_+)!} \prod_{k=1}^{K_+} \frac{\Gamma(|T_k| + \alpha_M/K)}{\Gamma(1 + \alpha_M/K)}, \quad K = K_+, K_+ + 1, \dots$$

- (b) Using a random walk MH step, sample $\alpha_M | \mathcal{C}, K$ from

$$p(\alpha_M | \mathcal{C}, K) \propto p(\alpha_M) \frac{\alpha_M^{K_+} \Gamma(\alpha_M)}{\Gamma(T + \alpha_M)} \prod_{k=1}^{K_+} \frac{\Gamma(|T_k| + \alpha_M/K)}{\Gamma(1 + \alpha_M/K)}.$$

Block 4.

- (a) If $K > K_+$, add $K - K_+$ empty clusters and sample their cluster means μ_k and the corresponding factor model parameters, i.e. Λ_k and Ξ_k from the priors.
 - (b) Sample cluster weights from $\eta_K \sim \mathcal{D}(\alpha_M/K + |T_1|, \dots, \alpha_M/K + |T_K|)$.
-

Λ_k , which takes the value of 0 if the corresponding column λ_{hk} is assigned to the spike and of 1 if the corresponding column is assigned to the slab. The classification itself is performed in step 5 of Algorithm 2, where the values 0 or 1 are assigned to γ_{hk} according to the marginal probabilities of λ_{hk} arising from either the spike or the slab distribution. In step 6, slab probabilities τ_{hk} are updated based on the binary indicators γ_{hk} . Finally, in step 7, the cluster-specific factor loading variances θ_{hk} are sampled separately for the columns assigned to the spike and for those assigned to the slab.

Algorithm 3 describes the procedure in Block 2, step (b) of Algorithm 1, where the hyperparameters of the factor-analytical models are updated based on the information derived from the K_+ filled clusters in Block 2, step (a).

Note that in step 5 of Algorithm 3 we marginalise over τ_{hk} and sample α_B using only the information from the classification of the columns of the factor loading matrices

Algorithm 2 Details of the step (a) in Block 2 of the Algorithm 1

for (k in $1 : K_+$) **do**

1. Sample \mathbf{f}_t^k for $t : t \in T_k$ from

$$\mathbf{f}_t^k | - \sim N_H \left((\Phi_H + \Lambda_k^T \Xi_k^{-1} \Lambda_k)^{-1} \Lambda_k^T \Xi_k^{-1} (\mathbf{y}_t - \boldsymbol{\mu}_k), (\Phi_H + \Lambda_k^T \Xi_k^{-1} \Lambda_k)^{-1} \right),$$

where $\Xi_k = \text{diag}(\xi_{1k}^2, \dots, \xi_{pk}^2)$ and $\Phi_H = \mathbf{I}_H$.

2. Sample the i th row λ_{ik} of the k th cluster loading matrix for i in $(1, \dots, p)$ from

$$\lambda_{ik}^\top | - \sim N_H \left((\Psi_k^{-1} + \xi_{ik}^{-2} \mathbf{F}_k \mathbf{F}_k^T)^{-1} \mathbf{F}_k \xi_{ik}^{-2} (\mathbf{y}_i - \mu_{ik})^T, (\Psi_k^{-1} + \xi_{ik}^{-2} \mathbf{F}_k \mathbf{F}_k^T)^{-1} \right),$$

where $\Psi_k = \text{diag}(\theta_{1k}, \dots, \theta_{Hk})$, $\mathbf{F}_k = \{\mathbf{f}_t^k : t \in T_k\}$ is a matrix of factors of the cluster k , and \mathbf{y}_i is a vector of observations of the variable i , for which $S_t = k$.

3. Sample ξ_{ik}^{-2} for i in $(1, \dots, p)$ from

$$\xi_{ik}^{-2} | - \sim \mathcal{G} \left(a_\xi + \frac{|T_k|}{2}, b_\xi + \frac{1}{2} \sum_{t:t \in T_k} (y_{it} - \mu_{ik} - \lambda_{ik} \mathbf{f}_t^k)^2 \right).$$

4. Update the cluster-specific mean $\boldsymbol{\mu}_k$ from $\boldsymbol{\mu}_k | - \sim N_p(\mathbf{b}_k, \mathbf{B}_k)$, where

$$\mathbf{b}_k = \mathbf{B}_k \left(\mathbf{B}_0^{-1} \mathbf{b}_0 + \Xi_k^{-1} \sum_{t:t \in T_k} (\mathbf{y}_t - \Lambda_k \mathbf{f}_t) \right), \quad \mathbf{B}_k = (\mathbf{B}_0^{-1} + |T_k| \Xi_k^{-1})^{-1}.$$

5. Sample the binary indicators γ_{hk} for each column $\{\lambda_{hk}\}$, $h = 1, \dots, H$ of the loading matrix as

$$\begin{aligned} P(\gamma_{hk} = 0 | \lambda_{hk}, b_0, \alpha_B, H) &\propto \frac{H}{\alpha_B + H} t_{2a_0}(\lambda_{hk}; \mathbf{0}, (b_0/a_0)\mathbf{I}_p), \\ P(\gamma_{hk} = 1 | \lambda_{hk}, b_\theta, \alpha_B, H) &\propto \frac{\alpha_B}{\alpha_B + H} t_{2a_\theta}(\lambda_{hk}; \mathbf{0}, (b_\theta/a_\theta)\mathbf{I}_p). \end{aligned}$$

6. Sample the (unordered) slab probabilities τ_{hk} for h in $(1, \dots, H)$:

$$\tau_{hk} | \gamma_{hk} \sim \mathcal{B} \left(\frac{\alpha_B}{H} + \gamma_{hk}, 2 - \gamma_{hk} \right).$$

7. Given γ_{hk} and the h th column λ_{hk} of the loading matrix, for each h in $(1, \dots, H)$, sample $\theta_{hk} | \gamma_{hk}, \lambda_{hk}$ depending on γ_{hk} :

$$\begin{aligned} \theta_{hk} | \gamma_{hk} = 0, \lambda_{hk} &\sim \mathcal{G}^{-1} \left(a_0 + \frac{1}{2} p, b_0 + \frac{1}{2} \sum_{i=1}^p \lambda_{ihk}^2 \right), \\ \theta_{hk} | \gamma_{hk} = 1, \lambda_{hk} &\sim \mathcal{G}^{-1} \left(a_\theta + \frac{1}{2} p, b_\theta + \frac{1}{2} \sum_{i=1}^p \lambda_{ihk}^2 \right). \end{aligned}$$

end for

Algorithm 3 Details of the step (b) in Block 2 of the Algorithm 1

1: Sample $b_{\xi i}$ for $i = 1, \dots, p$ from

$$b_{\xi i} | - \sim \mathcal{G} \left(a_g + K_+ a_\xi, b_{gi} + \sum_{k=1}^{K_+} \frac{1}{\xi_{ik}^2} \right).$$

2: Calculate the effective number of “active” columns H_k in cluster k as $H_k = \sum_{h=1}^H \gamma_{hk}$. Define $H^{++} = \sum_{k=1}^{K_+} H_k$ as the total number of “active” columns in all filled clusters and $H^\infty = HK_+ - H^{++}$ as the total number of “inactive” columns in all filled clusters.

3: Sample b_0 from

$$b_0 | - \sim \mathcal{G} \left(a_1 + H^\infty a_0, b_1 + \sum_{k=1}^{K_+} \sum_{h:\gamma_{hk}=0} \frac{1}{\theta_{hk}} \right).$$

4: Sample b_θ from

$$b_\theta | - \sim \mathcal{G} \left(a_2 + H^{++} a_\theta, b_2 + \sum_{k=1}^{K_+} \sum_{h:\gamma_{hk}=1} \frac{1}{\theta_{hk}} \right).$$

5: Use a random walk MH step to sample $\alpha_B | H_1, \dots, H_{K_+}, H$ from

$$p(\alpha_B | H_1, \dots, H_{K_+}, H) \propto \left(\frac{\alpha_B}{\alpha_B + H} \right)^{H^{++}} \left(\frac{H}{\alpha_B + H} \right)^{H^\infty} p(\alpha_B).$$

into “active” and “inactive” ones and the corresponding number of active factors in each filled cluster, H_1, \dots, H_{K_+} (see Frühwirth-Schnatter (2023) for the single factor model solution). This is done via a random walk Metropolis-Hastings step with proposal $\log \alpha_B^{new} \sim \mathcal{N}(\log \alpha_B, s_\alpha^2)$. As the acceptance rate depends on the dimension of the data set p through H , we made the step size s_α dependent on H exponentially $s_\alpha = 1 + \alpha_1(1 - \alpha_2)^H$, thus making sure that the step size is getting smaller as p (and hence H) increases. In our empirical settings we used $\alpha_1 = 2$ and $\alpha_2 = 0.11$. Alternatively, it is also possible to sample α_B in a Gibbs sampling step conditioning on τ_{hk} , H and K_+ from

$$\alpha_B | \tau_{hk}, H, K_+ \sim \mathcal{G} \left(a_\alpha + HK_+, b_\alpha - \frac{1}{H} \sum_{k=1}^{K_+} \sum_{h=1}^H \log \tau_{hk} \right).$$

However, we found that sampling α_B conditional on τ_{hk} in some cases leads to α_B being stuck at relatively high values and results in an overestimation of H_1, \dots, H_{K_+} , while marginalising out τ_{hk} leads to a more stable performance of the algorithm.

4.2 Post-processing and stratification

Before conducting any further inference, the MCMC output should first undergo a post-processing treatment to resolve the label switching problem inherent in Bayesian inference for finite mixture models to ensure a correct grouping of the model components into clusters. We estimate the number of active components \hat{K}_+ as the mode of the posterior distribution $p(K_+ | \mathbf{y})$ derived from all MCMC draws and keep only those draws where the realisation of K_+ is equal to \hat{K}_+ . For these draws, label switching between clusters is accounted for by applying a variant of k -means clustering to the point process representation of the MCMC draws in the spirit of [Frühwirth-Schnatter \(2011\)](#), see Section B.2 in the Supplementary Material for details on this procedure.

For inference on the parameters of the cluster-specific factor models, these draws are processed further. First, we estimate the number of factors \hat{H}_k in each of the K_+ clusters as the mode of the corresponding posterior distribution. In a second step, we remove all draws in which the number of active factors H_k is not equal to \hat{H}_k in any cluster. For the remaining draws, the cluster-specific factor loading matrices exhibit \hat{H}_k active and $H - \hat{H}_k$ inactive columns (with the respective indicator $\gamma_{hk} = 1$ and $\gamma_{hk} = 0$) and we keep only the active columns for further inference. These sparsified cluster-specific factor loading matrices $\boldsymbol{\Lambda}_k$ with \hat{H}_k columns are then used for the calculation of the cluster-specific covariance matrices $\boldsymbol{\Omega}_k = \boldsymbol{\Lambda}_k \boldsymbol{\Lambda}_k^\top + \boldsymbol{\Xi}_k$, see Section 5.

The cluster-specific factor loading matrices $\boldsymbol{\Lambda}_k$, however, are not uniquely identified, since we did not impose any constraints on these matrices during MCMC sampling to prevent rotational invariance. Rather, we apply the MatchAlign algorithm introduced in [Poworoznek et al. \(2021\)](#) independently in each cluster k to resolve rotational invariance for the draws of $\boldsymbol{\Lambda}_k$. More details of this approach are discussed in Section B.2 of the Supplementary Material.

5 Simulation studies

The performance of the dynamic $(MF)^2A$ model is first demonstrated on simulation studies. We use several different settings to assess the model's ability to correctly infer the cluster and factor dimensionality of the data sets. In Section 5.1 we demonstrate the performance of the model for a range of various p and T settings² on data sets with balanced cluster sizes and a common number of factors. The simulation study in Section 5.2 is more challenging with a larger number of clusters, some of which are small, and a varying number of cluster-specific factors. The hyperparameter specifications are reported in Table 1. The maximum number of factors H is equal to the largest integer which satisfies the variance identification condition of [Anderson and Rubin \(1956\)](#) $H \leq (p - 1)/2$. Unless otherwise specified, the sampler is run for 50,000 iterations, with 20% of them discarded as burn-in. In both sections, we compare the performance of the dynamic $(MF)^2A$ model with the IMIFA model for all simulated data sets.

²We consider only $T > p$ settings as this is a precondition for a stable performance of a model based on the mixture of Gaussian distributions. In the case $p > T$ the cluster covariance matrices are not well defined which leads to difficulties in both clustering assignment and factor model performance.

Parameter(s)	Hyperparameters	Hyperparameter values
μ_k	$\mathbf{b}_0, \mathbf{B}_0$	$median(\mathbf{y}), diag(R_1^2, \dots, R_p^2)$
K	$\alpha_\lambda, \alpha_\pi, \beta_\pi$	1, 4, 3
α_M	ν_l, ν_r	6, 3
α_B	a_α, b_α	6, 2
ξ_{ik}^2	a_ξ, a_g, b_{gi}	1, 3, $100/R_i^2$
$\theta_{hk} \gamma_{hk} = 1$	a_θ, a_2, b_2	3, 2, 1
$\theta_{hk} \gamma_{hk} = 0$	a_0, a_1, b_1	21, 1, 1

Table 1: Hyperparameter specifications for the $(MF)^2 A$ model. In the prior for μ_k , R_i denotes the range of the data in dimension i .

The clustering performance is assessed using the adjusted Rand index (ARI; [Hubert and Arabie \(1985\)](#)) and the misclassification rate is estimated as the percentage of mislabelled observations compared to the true cluster labels used to simulate the data. To assess the accuracy of the model in estimating the true cluster-specific covariance matrices $\Omega_k^0 = \Lambda_k^0(\Lambda_k^0)^\top + \Xi_k^0$ of the data via the estimated covariance matrices

$$\widehat{\Omega}_k = \frac{1}{M} \sum_{m=1}^M \Omega_k^{(m)}, \quad \Omega_k^{(m)} = \Lambda_k^{(m)}(\Lambda_k^{(m)})^\top + \Xi_k^{(m)},$$

where $\Lambda_k^{(m)}$ and $\Xi_k^{(m)}$ is the m -th among the M posterior draws passing the post-processing procedure described in Section 4.2, we compute for each simulation in each of the scenarios a Monte-Carlo estimate of the mean squared error (MSE) defined by

$$MSE_{\Omega_k} = \sum_{i=1}^p \sum_{l=i}^p \mathbb{E}((\Omega_{k,il} - \Omega_{k,il}^0)^2 | \mathbf{y}) / (p(p+1)/2).$$

Following [Murphy et al. \(2020\)](#), the data are standardised before feeding them into the model. More specifically, the data are transformed as $\tilde{\mathbf{y}}_t = \mathbf{S}^{-1}(\mathbf{y}_t - \mathbf{m})$, where \mathbf{m} denotes the vector of means of \mathbf{y} and the scale matrix $\mathbf{S} = \sqrt{\text{diag}(S_{y,1} \dots S_{y,p})}$ is defined from the empirical variances $S_{y,i}$ of the data y_{it} over $t = 1, \dots, T$. Given the true cluster-specific factor loading and covariance matrices Λ_k^0 and Ω_k^0 of the original data, the corresponding matrices then take the form $\mathbf{S}^{-1}\Lambda_k^0$ and $\mathbf{S}^{-1}\Omega_k^0\mathbf{S}^{-1}$ for the transformed data.

5.1 Simulation study 1

The aim of this simulation study is to evaluate the performance of our model on data sets of various sizes, i.e. with various settings of p and T , with the clusters approximately equally sized, but not very well separated from each other. Three different settings of (p, T) were considered to test the performance of the model on small, middle sized and relatively large data sets, and also to evaluate the results against an increasing dimensionality of the data set, namely $(20, 100)$, $(30, 200)$ and $(50, 500)$. Note, that for a reliable performance of clustering based on mixture distributions, the number of observations in each cluster should be reasonably bigger than the number of variables.

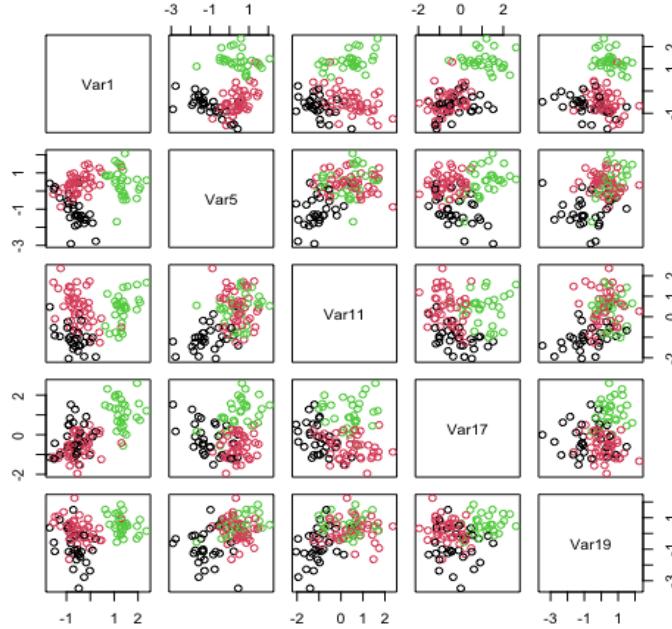


Figure 1: Pairwise scatter plots of five randomly chosen variables from one of the replicate data sets in Simulation Study 1 with $p = 10$ and $T = 100$ demonstrating the overlap between the three clusters.

The data are simulated with $K_+ = 3$ clusters, $H_k = 4$ factors in each cluster and cluster weights $\boldsymbol{\eta}_3 = (1/3, 1/3, 1/3)$. The model parameters are simulated as $\xi_{ik}^2 \sim \mathcal{G}^{-1}(2, 1)$ and $\boldsymbol{\lambda}_{ik}^\top \sim N_4(\mathbf{0}, \mathbf{I}_4)$ for all i and k . To ensure that clusters are overlapping, the means are generated similarly to [Murphy et al. \(2020\)](#) as $\boldsymbol{\mu}_k \sim N_p((2k-4)\mathbf{1}, \mathbf{I}_p)$. Each \mathbf{y}_t , $t = 1, \dots, T$, is simulated from a conditional mixture model with $\mathbf{f}_t^k \sim N_4(\mathbf{0}, \mathbf{I}_4)$:

$$p(\mathbf{y}_t | \{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k, \mathbf{f}_t^k, \boldsymbol{\Xi}_k\}, \boldsymbol{\eta}_{K_+}) = \sum_{k=1}^{K_+} \eta_k N_p(\mathbf{y}_t; \boldsymbol{\mu}_k + \boldsymbol{\Lambda}_k \mathbf{f}_t^k, \boldsymbol{\Xi}_k).$$

To test the robustness of the model performance, each simulation setting was replicated 25 times and each time the data set was newly generated. For the (50, 500) setting we ran the sampler for 75,000 iterations, as convergence was slower to achieve for a bigger data set. As in the other settings, 20% of the draws were discarded as burn-in. As the cluster-specific parameters $\boldsymbol{\Lambda}_k$ and $\boldsymbol{\Xi}_k$ could induce some degree of separation between clusters, pairwise scatter plots from a randomly chosen data set are shown in Figure 1 to demonstrate the extent of overlap amongst clusters. For the sake of clear visibility, five randomly chosen variables from a data set with $p = 20$ variables and $T = 100$ observations are depicted.

The results provided in Table 2 demonstrate that the dynamic $(MF)^2A$ model per-

Dimension	\hat{K}_+	\hat{K}	\hat{H}_1	\hat{H}_2	\hat{H}_3	ARI	Error (%)	MSE_{Ω_1}	MSE_{Ω_2}	MSE_{Ω_3}
(20, 100)	2.80	2.80	3.95	4.00	4.00	0.91	6.4	0.027	0.032	0.032
(30, 200)	2.84	2.84	4.33	4.43	4.33	0.93	4.84	0.017	0.016	0.014
(50, 500)	2.80	2.80	4.40	4.30	4.40	0.92	6.36	0.005	0.005	0.005

Table 2: Simulation results for the $(MF)^2A$ model under different dimensionality settings. The true numbers are $K_+ = 3$ and $H_1 = H_2 = H_3 = 4$. For each performance measure, the average across 25 simulated data sets is reported. The performance of K_+ and H_k is based on modal estimates. Clustering performance is assessed via the ARI and the percentage error rate against the known cluster labels.

forms generally well for all three settings of (p, T) , exhibiting the capability to uncover the true structure of the simulated data in most cases. The partition has been identified correctly in the majority of cases for all settings of (p, T) . In the few cases where the number of clusters was underestimated, two clusters were merged into one. The number of cluster-specific factors was occasionally slightly overestimated for the larger settings of (p, T) , however, was recovered correctly in most cases. In general, the model exhibited a stable performance both on small and relatively large data sets. The MSE of the cluster-specific covariance matrices was in all cases very small, with the estimation becoming more exact as the size of the data set grew. This shows that the model generally correctly estimated cluster structure even in the cases when the number of cluster-specific factors is overestimated.

To provide some benchmark, we compare the performance of our dynamic $(MF)^2A$ model with the IMIFA model of [Murphy et al. \(2020\)](#) by running the R-package `IMIFA` on the same 25 simulated data sets. To keep the settings of the two models as close as possible, we set the PYP parameters of the IMIFA model, namely σ and d , to be learned from data and run the model for exactly the same number of iterations. The hyperparameters of the MGP prior are left as default, i.e. $a_1 = 2.1$ and $a_2 = 3.1$. The results presented in Table 3 show that the clustering performance of the IMIFA model is very good even on small data sets where the number of observations within clusters is not much higher than the number of variables. This is due to an advantage of the PYP prior over the MFM prior in the case of rather small number of observations within clusters. In the factor-analytical part, however, IMIFA shows a clear tendency to overestimate the number of cluster-specific factors compared to the dynamic $(MF)^2A$ model (see again Table 2). This can be attributed to the intrinsic inefficiency of the MGP prior described in Section 2.2.

5.2 Simulation study 2

The design of the simulation study presented in this section is more challenging as the clusters are of different sizes and the number of factors varies between clusters. The data is generated with $p = 20$ and $T = 700$, and is allocated into $K_+ = 6$ clusters with varying numbers of cluster-specific factors. The clusters are given weights $\boldsymbol{\eta}_6 = (0.25, 0.25, 0.2, 0.15, 0.1, 0.5)$, thus including rather small clusters (a setting which often

Dimension	\hat{K}_+	\hat{H}_1	\hat{H}_2	\hat{H}_3	ARI	Error (%)
(20, 100)	3	4.92	5	5	1	0
(30, 200)	3	5	5	5	1	0
(50, 500)	3	5	5	5	1	0

Table 3: Simulation results for the IMIFA model under different dimensionality settings. The true numbers are $K_+ = 3$ and $H_1 = H_2 = H_3 = 4$. For each performance measure, the average across 25 simulated data sets is reported. The performance of K_+ and H_k is based on modal estimates. Clustering performance is assessed via the ARI and the percentage error rate against the known cluster labels.

appears in Bayesian nonparametric models). The number of factors H_1, \dots, H_{K_+} are drawn randomly from $1, \dots, 6$, with the upper limit being smaller than $(p - 1)/2$ and thus satisfying the variance identification constraint. Otherwise, the same parameter settings as in Section 5.1 are used to generate the data. Figure D.5 in the Supplementary Material, Section D illustrates the extent of intermixing between the clusters by showing pairwise scatter plots for five randomly chosen variables for the first replicate data set.

The sampler was run for 50,000 iterations, which was enough for convergence, with 20% of the draws discarded as burn-in. The true model dimensions, namely the number of clusters K_+ , the cluster sizes $|T_1|, \dots, |T_{K_+}|$, and the cluster-specific number of factors $\mathbf{H} = (H_1, \dots, H_{K_+})$, of ten simulated data sets are summarised in Table 4. The estimated model dimensions as well as the ARI and the clustering error are presented in Table 5. In half of the cases, the model correctly identified the partition and cluster assignments. In the other five cases, where the model identified 5 clusters, two original clusters were merged together. Regarding the inference on the number of factors, in most cases the number of cluster-specific factors was determined correctly, with a few cases of overestimation, especially in the case when two clusters were merged together. As shown on Figure D.5, there is a high degree of overlapping between the clusters. This presents a rather challenging task for a Gaussian mixture model when some of the modes are situated close together, in which case the model tends to merge clusters. However, even in this challenging environment the model exhibited a reasonably good clustering performance with ARI more or close to 0.8 in case of merged clusters and a reliable performance in detecting the number of factors in each cluster.

The results of running the IMIFA model on exactly the same simulated data sets are presented in Table 6. They confirm the strong tendency of the model to overestimate the number of factors, which was also evident in Table 3. The clustering performance, on the other hand, is perfect which can be explained by the general good performance of the PYP prior in situations with a large number of differently sized clusters.

6 Empirical data: Eurozone inflation rates

To illustrate the model's performance on real data, we employ the dynamic $(MF)^2A$ model to analyse the structure of data consisting of the Harmonised Index of Consumer

	K_+	$ T_1 , \dots, T_{K_+} $	\mathbf{H}
1st replicate	6	(170, 185, 137, 99, 79, 30)	(2, 3, 4, 6, 5, 6)
2nd replicate	6	(179, 168, 134, 118, 60, 41)	(3, 4, 3, 4, 2, 3)
3rd replicate	6	(157, 178, 134, 114, 81, 36)	(6, 5, 5, 5, 4, 3)
4th replicate	6	(183, 173, 126, 111, 74, 33)	(2, 4, 4, 5, 3, 6)
5th replicate	6	(179, 178, 133, 98, 76, 36)	(5, 5, 4, 4, 6, 4)
6th replicate	6	(160, 185, 141, 117, 70, 27)	(5, 4, 6, 5, 5, 2)
7th replicate	6	(194, 153, 132, 106, 73, 42)	(6, 3, 2, 3, 4, 5)
8th replicate	6	(171, 194, 145, 98, 57, 35)	(1, 4, 2, 5, 3, 3)
9th replicate	6	(197, 159, 127, 113, 65, 39)	(5, 1, 5, 5, 5, 6)
10th replicate	6	(175, 183, 141, 102, 72, 27)	(1, 5, 3, 2, 5, 3)

Table 4: True model dimensions of ten simulated data sets in the situation with unbalanced cluster sizes and different number of cluster-specific factors.

	\hat{K}_+	$ \hat{T}_1 , \dots, \hat{T}_{\hat{K}_+} $	$\hat{\mathbf{H}}$	ARI	Error (%)
1st replicate	6	(170, 185, 137, 99, 79, 30)	(2, 3, 4, 6, 5, 6)	1	0
2nd replicate	5	(179, 168, 252, 60, 41)	(3, 4, 9, 2, 3)	0.82	16.8
3rd replicate	6	(157, 178, 134, 114, 81, 36)	(6, 5, 5, 5, 4, 3)	1	0
4th replicate	5	(183, 173, 200, 111, 33)	(2, 4, 8, 5, 7)	0.89	10.6
5th replicate	6	(179, 178, 133, 98, 76, 36)	(5, 5, 4, 4, 6, 4)	1	0
6th replicate	5	(160, 185, 141, 117, 97)	(5, 4, 6, 5, 8)	0.98	3.9
7th replicate	5	(194, 285, 106, 73, 42)	(6, 6, 3, 4, 6)	0.77	18.9
8th replicate	6	(171, 194, 145, 98, 57, 35)	(1, 4, 2, 5, 3, 3)	1	0
9th replicate	5	(197, 286, 113, 65, 39)	(5, 7, 5, 5, 6)	0.78	18.1
10th replicate	6	(175, 183, 141, 102, 72, 27)	(1, 5, 3, 2, 5, 4)	1	0

Table 5: Estimation results for the dynamic $(MF)^2A$ model in the situation with unbalanced cluster sizes and different number of cluster-specific factors.

Prices (HICP) inflation rates for $p = 19$ Eurozone countries for the period from February 1997 to October 2019, which yields in total $T = 273$ observations. Figure 2 illustrates the path of these time series for the reported period. In this case, clustering is performed with respect to the time dimension, which seems a natural choice as in different countries factors which drive their inflation rates may differ in various time periods (for example, some countries in the data set joined the single currency area later than others). The sampler was run for 50,000 iteration and 20% were discarded as a burn-in. The data were demeaned and unit-scaled and the same hyperparameters were used as in Table 1.

The results of applying the dynamic $(MF)^2A$ model to the HICP inflation rates are presented in Table 7. The six clusters estimated by the dynamic $(MF)^2A$ model show a clear time-related pattern as they are situated one after another on the time

	\hat{K}_+	$ \hat{T}_1 , \dots, \hat{T}_{\hat{K}_+} $	$\hat{\mathbf{H}}$	ARI	Error (%)
1st replicate	6	(170, 185, 137, 99, 79, 30)	(3, 4, 5, 7, 6, 7)	1	0
2nd replicate	6	(179, 168, 137, 118, 60, 41)	(4, 5, 4, 5, 3, 4)	1	0
3rd replicate	6	(157, 178, 134, 114, 81, 36)	(6, 6, 6, 5, 4)	1	0
4th replicate	6	(183, 173, 126, 111, 74, 33)	(3, 5, 5, 6, 4, 6)	1	0
5th replicate	6	(179, 178, 133, 98, 76, 36)	(6, 6, 5, 5, 6, 5)	1	0
6th replicate	6	(160, 185, 141, 117, 70, 27)	(6, 5, 7, 6, 6, 3)	1	0
7th replicate	6	(194, 153, 132, 106, 73, 42)	(7, 4, 3, 4, 5, 6)	1	0
8th replicate	6	(171, 194, 145, 98, 57, 35)	(2, 5, 3, 6, 4, 4)	1	0
9th replicate	6	(197, 159, 127, 113, 65, 39)	(6, 2, 6, 6, 6, 7)	1	0
10th replicate	6	(175, 183, 141, 102, 72, 27)	(2, 6, 4, 3, 6, 4)	1	0

Table 6: Estimation results for the IMIFA model in the situation with unbalanced cluster sizes and different number of cluster-specific factors.

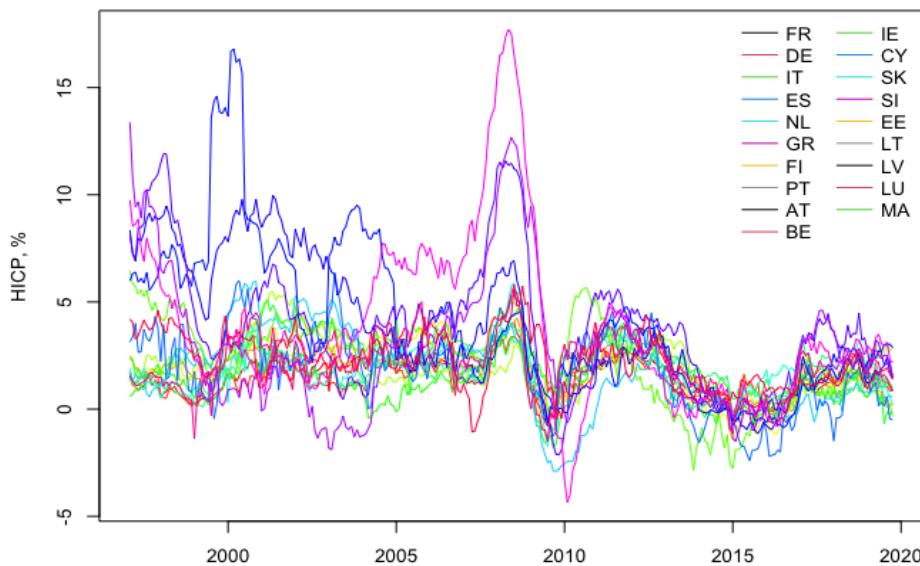


Figure 2: HICP year-on-year inflation rates of 19 Eurozone countries for the period February 1997 - October 2019.

line, see Figure 3. Cluster 1, marked by the coral colour, contains observations in the period February 1997 - June 1999, which roughly corresponds to the period before the

Model	# clusters	# factors
Dynamic $(MF)^2 A$	6	3,2,2,3,3,2
IMIFA	20	2,2,3,2,2,2,2,2,2,3,2,3,3,2,3,2,2,2,2,3

Table 7: Results of fitting the dynamic $(MF)^2 A$ and IMIFA models on the Eurozone inflation rates data set.

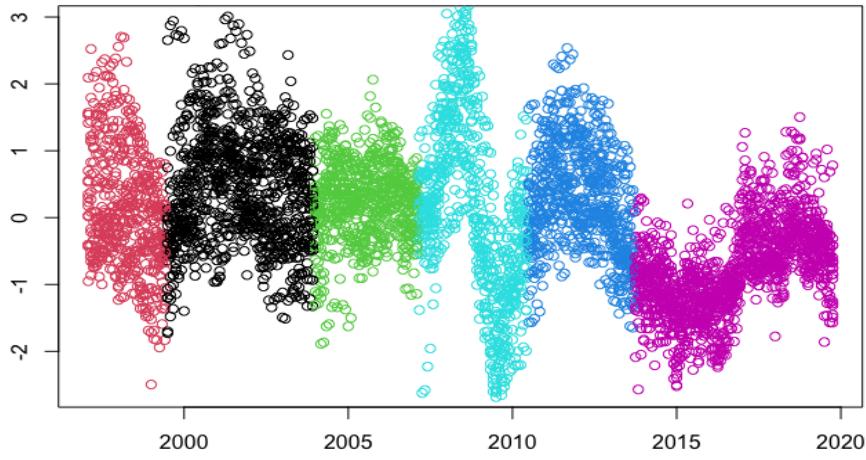


Figure 3: Cluster assignments of the Eurozone inflation rates data set according to the dynamic $(MF)^2 A$ model.

introduction of the Euro³. Cluster 2 (black) covers the period from July 1999 till December 2003. The third cluster, depicted green in Figure 3, contains observations from January 2004 till February 2007, and corresponds to the period between the extension of the European Union by ten new members⁴ and the financial crisis. The fourth cluster coloured light blue on the chart covers the period between March 2007 and June 2010 and encompasses the financial crisis 2007 – 2008 and the subsequent recession. The period from July 2010 to September 2013, assigned to cluster 5 (darker blue), was marked by the European sovereign debt crisis which resulted in bailout packages for several Eurozone countries. Finally, October 2013 - October 2019 was a period of extremely low and at times even negative inflation rates amongst the Eurozone countries, during which the European Central Bank struggled to stimulate inflation with a very loose monetary policy.

³The Euro was launched as a currency for accounting purposes and electronic payments on January 1, 1999 while coins and banknotes were introduced on January 1, 2002.

⁴On May 1, 2004, ten new members joined the EU, namely, Cyprus, Malta, Czechia, Estonia, Hungary, Latvia, Lithuania, Poland, Slovakia and Slovenia.

The estimated number of cluster-specific latent factors fluctuates between two and three, depending on the cluster. It is interesting to note that the number of cluster-specific factors estimated by the model is higher in periods marked by crises. In fact, in cluster 4, which covers the period of the financial crisis 2007 – 2008 and the following recession, the model estimates three active factors. In the subsequent period, marked by the sovereign debt crisis, the estimated number of active factors is also three, as in the first period which precedes the introduction of the Euro. In all other periods (clusters 2, 3 and 6) the estimated number of latent factors is two.

For comparison, we also ran the IMIFA model on the Eurozone inflation rates data set. The IMIFA model estimated 20 clusters, nearly as much as the 21 years covered by the observation period, with either two or three latent factors in each cluster. This example illustrates that the dynamic $(MF)^2A$ model is well suited for analysing also time series data, exhibiting better clustering performance in this case than the IMIFA model. The comparative performance of the dynamic $(MF)^2A$ model on several benchmark data sets is investigated further in the Supplementary material, Section E.

7 Conclusion

We proposed a novel model in the MFA framework which allows fully automatic inference on the number of non-empty components in the mixture and the number of latent factors in the cluster-specific factor-analytical models while keeping both dimensions finite. This was done by employing the connection between nonparametric Bayesian process priors and their finite representations, connecting the MFM framework with the ESP class of priors (Frühwirth-Schnatter (2023)) in the factor-analytical part. This approach allowed to eliminate some of the drawbacks of the popular MGP prior. Posterior inference is performed solely within Gibbs sampler steps and all influential parameters are learned from the data, which makes it possible to use the dynamic $(MF)^2A$ model on various data sets with no or little additional tuning. Some hyperparameter tuning may become necessary when working with data sets of essentially different nature. However, the hyperparameter values we provided proved to be rather universal and can be employed for data sets of various sizes and structure, including time series data.

Future research directions could include, for example, introducing element-wise shrinkage for the columns of cluster-specific factor loading matrices, which could help to achieve more exact identification in sparse factor models. Making hyperparameters of cluster-specific factor models, namely α_B , b_0 and b_θ , cluster-specific could improve the model's performance in settings with differently sized clusters, where the dynamic $(MF)^2A$ model sometimes struggled to distinguish smaller clusters. Alternatively, the triple gamma prior (Cadonna et al., 2020) could be employed for the spike and the slab distributions instead of the inverse gamma priors (Frühwirth-Schnatter, 2023). This could improve mixing and uncertainty quantification of the number of cluster-specific factors. Another interesting and promising direction of research would be relaxing the assumption of independent factors, which is particularly restrictive in the case of time series data, where observations at different times are often correlated.

References

- Anderson, T. and Rubin, H. (1956). “Statistical inference in factor analysis.” *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, Volume V: 111–150.
- Ascolani, F., Lijoi, A., Rebaudo, G., and Zanella, G. (2023). “Clustering consistency with Dirichlet process mixtures.” *Biometrika*, 110: 551–558.
- Bhattacharya, A. and Dunson, D. (2011). “Sparse Bayesian infinite factor models.” *Biometrika*, 98(2): 291–306.
- Cadonna, A., Frühwirth-Schnatter, S., and Knaus, P. (2020). “Triple the gamma - A unifying shrinkage prior for variance and variable selection in sparse state space and TVP models.” *Econometrics*, 8(2): 1–36.
- De Blasi, P., Favaro, S., Lijoi, A., Mena, R., and Ruggiero, M. (2015). “Are Gibbs-type priors the most natural generalization of the Dirichlet process?” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37: 212–229.
- Durante, D. (2017). “A note on the multiplicative gamma process.” *Statistics & Probability Letters*, 122: 198–204.
- Fokoue, E. (2000). “A Markov chain Monte Carlo (MCMC) approach to the Bayesian analysis of mixtures of factor analysers.” In *Proceedings in Computational Statistics 2000, Short Communication and Posters*, 19–30. Statistics The Netherlands.
- Fokoue, E. and Titterington, D. (2003). “Mixtures of factor analysers. Bayesian estimation and inference by stochastic simulation.” *Machine Learning*, 50(1): 73–94.
- Frühwirth-Schnatter, S. (2011). *Dealing with Label Switching under Model Uncertainty*, chapter 10, 213–239. John Wiley & Sons, Ltd.
- (2023). “Generalized Cumulative Shrinkage Process Priors with Applications to Sparse Bayesian Factor Analysis.” *Philosophical Transactions of the Royal Society A*, (381): 381:20220148.
- Frühwirth-Schnatter, S., Hosszejni, D., and Lopes, H. F. (2023). “When It Counts - Econometric Identification of the Basic Factor Model Based on GLT Structures.” *Econometrics*, 11(4): 26.
- (2024). “Sparse finite Bayesian factor analysis when the number of factors is unknown.” *Bayesian Analysis*, accepted for publication.
- Frühwirth-Schnatter, S. and Malsiner-Walli, G. (2019). “From here to infinity: Sparse finite versus Dirichlet process mixtures in model-based clustering.” *Advances in Data Analysis and Classification*, 13: 33–64.
- Frühwirth-Schnatter, S., Malsiner-Walli, G., and Grün, B. (2021). “Generalised Mixtures of Finite Mixtures and Telescoping Sampling.” *Bayesian Analysis*, 16(4): 1279–1307.
- Ghahramani, Z. and Beal, M. (2000). “Variational inference for Bayesian mixture of

- factor analyzers.” In *Advanced in Neural Information Processing system*, volume 12, 449–455.
- Ghahramani, Z., Griffiths, T. L., and Sollich, P. (2007). “Bayesian nonparametric latent feature models (with discussion and rejoinder).” In Bernardo, J. M., Bayarri, M. J., Berger, J. O., Dawid, A. P., Heckerman, D., Smith, A. F. M., and West, M. (eds.), *Bayesian Statistics 8*, 1–25. Oxford: Oxford University Press.
- Ghahramani, Z. and Hinton, G. (1996). “The EM algorithm for mixtures of factor analyzers.” Technical report, Department of Computer Science, University of Toronto.
- Gnedin, A. (2010). “A Species Sampling Model with Finitely Many Types.” *Electronic Communications in Probability*, 15: 79–88.
- Gnedin, A. and Pitman, J. (2006). “Exchangeable Gibbs partitions and Stirling triangles.” *Journal of Mathematical Sciences*, 138: 5674–5684.
- Greve, J., Grün, B., Malsiner-Walli, G., and Frühwirth-Schnatter, S. (2022). “Spying on the prior of the number of data clusters and the partition distribution in Bayesian cluster analysis.” *Australian & New Zealand Journal of Statistics*, 64: 205–229.
- Grün, B., Malsiner-Walli, G., and Frühwirth-Schnatter, S. (2022). “How many data clusters are in the Galaxy data set? Bayesian cluster analysis in action.” *Advances in Data Analysis and Classification*, 16(2): 325–249.
- Grün, B. and Leisch, F. (2009). “Dealing with label switching in mixture models under genuine multimodality.” *Journal of Multivariate Analysis*, 100(5): 851–861.
URL <https://www.sciencedirect.com/science/article/pii/S0047259X08001929>
- Hubert, L. and Arabie, P. (1985). “Comparing partitions.” *Journal of Classification*, 2(1): 193–218.
- Kowal, D. R. and Canale, A. (2023). “Semiparametric Functional Factor Models with Bayesian Rank Selection.” *Bayesian Analysis*, 18: 1161–1189.
- Legramanti, S., Durante, D., and Dunson, D. (2020). “Bayesian cumulative shrinkage for infinite factorizations.” *Biometrika*, 107(3): 745–752.
- McLachlan, G., Peel, D., and Bean, R. (2003). “Modelling high-dimensional data by mixtures of factor analyzers.” *Computational Statistics & Data Analysis*, 41: 379–388.
- McLachlan, G. J. and Peel, D. (2000). *Finite Mixture Models*. Wiley Series in Probability and Statistics. New York: Wiley.
- McNicholas, P., ElSherbiny, A., McDaid, A., and Murphy, T. (2018). “pgmm: parsimonious Gaussian mixture models. R package version 1.2.3.”
URL <https://cran.r-project.org/package=pgmm>
- McNicholas, P. and Murphy, T. (2008). “Parsimonious Gaussian mixture models.” *Statistics and Computing*, 18(3): 285–296.

- Miller, J. W. and Harrison, M. T. (2013). “A simple example of Dirichlet process mixture inconsistency for the number of components.” In *Advances in Neural Information Processing Systems*, 199–206.
- Murphy, K., Viroli, C., and Gormley, I. (2020). “Infinite Mixtures of Infinite Factor Analysers.” *Bayesian Analysis*, 15(3): 937–963.
- Papastamoulis, P. (2016). “label.switching: An R Package for Dealing with the Label Switching Problem in MCMC Outputs.” *Journal of Statistical Software*, 69(1): 1–24.
- (2018). “Overfitting Bayesian mixtures of factor analyzers with an unknown number of components.” *Computational Statistics and Data Analysis*, 124: 220–234.
- Poworoznek, E., Ferrari, F., and Dunson, D. (2021). “Efficiently resolving rotational ambiguity in Bayesian matrix sampling with matching.” *ArXiv: 2107.13783*.
- Redner, R. and Walker, H. (1984). “Mixture densities, maximum likelihood and the EM algorithm.” *SIAM Review*, 26(2): 195–239.
- Ročková, V. and George, E. (2016). “Fast Bayesian factor analysis via automatic rotation to sparsity.” *Journal of the American Statistical Association*, 111(516): 1608–1622.
- Schiavon, L. and Canale, A. (2020). “On the truncation criteria in infinite factor models.” *Stat*, 9(1): e298.
- Sethuraman, J. (1994). “A constructive definition of Dirichlet priors.” *Statistica Sinica*, 4: 639–650.
- Stephens, M. (2002). “Dealing With Label Switching in Mixture Models.” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 62(4): 795–809.
- Teh, Y., Görür, D., and Ghahramani, Z. (2007). “Stick-breaking Construction for the Indian Buffet Process.” In Meila, M. and Shen, X. (eds.), *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, 556–563. San Juan, Puerto Rico: PMLR.
- Viroli, C. (2010). “Dimensionally reduced model-based clustering through mixtures of factor mixture analyzers.” *Journal of Classification*, 27(3): 363–388.
- Wang, C., Pan, G., T., T., and L., Z. (2015). “Shrinkage estimation of large dimensional precision matrix using random matrix theory.” *Statistica Sinica*, 25(3): 993–1008.

Acknowledgments

The authors would like to thank the anonymous reviewers and the associate editor for valuable comments and suggestions which helped to significantly improve this work.

**Supplementary material for:
“Dynamic Mixture of Finite Mixtures of Factor
Analysers with Automatic Inference on the
Number of Clusters and Factors”**

Appendix A: More details on priors and hyperparameters

We use the BNB prior on the number of components K as in (3.2), with the parameters $\alpha_\lambda = 1$, $\alpha_\pi = 4$ and $\beta_\pi = 3$, which results in the a priori expectation of the number of components $E(K) = 2$. The reasoning behind this choice of hyperparameters can be found in [Frühwirth-Schnatter et al. \(2021\)](#) and [Grün et al. \(2022\)](#) along with a comparative study of the performance of various translated priors for $K - 1$ in the MFMs context. For the hyperparameters ν_l and ν_r of the F distribution prior on the concentration parameter $\alpha_{\mathcal{M}}$ used in the prior for mixture component weights $\boldsymbol{\eta}_K = \{\eta_k, k = 1, \dots, K\}$, we chose $\alpha_{\mathcal{M}} \sim \mathcal{F}(6, 3)$ following the reasoning in [Frühwirth-Schnatter et al. \(2021\)](#).

For the cluster means, we follow [Malsiner-Walli et al. \(2016\)](#) and choose in (3.1) the independence prior $\boldsymbol{\mu}_k \sim N_p(\mathbf{b}_0, \mathbf{B}_0)$ with the data-dependent hyperparameters

$$\mathbf{b}_0 = \text{median}(\mathbf{y}), \quad \mathbf{B}_0 = \text{diag}(R_1^2, \dots, R_p^2),$$

where R_i is the range of the data in dimension i .

In the application of mixture models to clustering multivariate data, it is often suggested in the literature to employ a hierarchical data-driven inverse Wishart prior for cluster covariance matrices (see, e.g. [Malsiner-Walli et al. \(2016\)](#), [Frühwirth-Schnatter \(2006\)](#)). In the MFA context, where each cluster contains a factor-analytical model, the cluster covariance matrices are computed at each iteration of the MCMC sampler as $\boldsymbol{\Omega}_k = \boldsymbol{\Lambda}_k \boldsymbol{\Lambda}_k^T + \boldsymbol{\Xi}_k$, where $\boldsymbol{\Lambda}_k$ is the $p \times H$ factor loading matrix in cluster k and $\boldsymbol{\Xi}_k$ is the $p \times p$ diagonal matrix of uniquenesses of the factor model in cluster k . Thus, the prior on $\boldsymbol{\Omega}_k$ has a more general structure than an inverse Wishart prior and is driven by the prior choices for $\boldsymbol{\Lambda}_k$ and $\boldsymbol{\Xi}_k$.

The choice of the maximum possible number of factors H is governed by the variance identification constraints. As mentioned in below in Section C, the variance identification of a factor model is guaranteed only when the number of latent factors satisfies the constraint $H_k \leq (p - 1)/2$. Consequently, we set H equal to the largest integer which is less or equal to $(p - 1)/2$. We noticed, however, that in practical implementation in cases when the data dimensionality p is rather small, like $p \leq 10$, which consequently leads to the upper limit on the number of factors being less than five, setting $H = p$ leads to better mixing and thus better performance of the model. Nevertheless, the effective number of active factors discovered by the model usually satisfies the identification constraint $H_k \leq (p - 1)/2$.

The cluster-specific idiosyncratic variance parameters ξ_{ik}^2 are given a hierarchical prior:

$$\xi_{ik}^2 \sim \mathcal{G}^{-1}(a_\xi, b_{\xi i}), \quad b_{\xi i} \sim \mathcal{G}(a_g, b_{gi}),$$

where the rate hyperparameters b_{gi} are assigned the data-driven values $100/R_i^2$, following the considerations in [Stephens \(1997\)](#) and [Frühwirth-Schnatter \(2006\)](#). Assigning a data-driven hierarchical prior to ξ_{ik}^2 is particularly beneficial in the context of MFA models due to the specific structure of the cluster-specific covariance matrices $\boldsymbol{\Omega}_k = \boldsymbol{\Lambda}_k \boldsymbol{\Lambda}_k^T + \boldsymbol{\Xi}_k$, where the cluster-specific idiosyncratic covariance matrix $\boldsymbol{\Xi}_k$ represents an important part of the covariance structure specific to cluster k . With the priors for the elements of $\boldsymbol{\Lambda}_k$ containing no data-related information, the prior for ξ_{ik}^2 provides $\boldsymbol{\Omega}_k$ with a link to information from the data. This is especially important because, as explained in details in Section 4, this prior is used to fill the parameters of the newly generated empty clusters during MCMC sampling, which makes the prior choice highly influential for the performance of the algorithm.

With the prior on factor loadings described in Section 3.2, the parameters of the spike and the slab distributions deserve some closer attention. As these parameters (especially of the spike) are rather influential in classifying factors into “active” and “inactive” ones (see, for example, [Schiavon and Canale \(2020\)](#) for a discussion of this subject), we let them be adjusted to the data by assigning hyperpriors to the scale parameters b_0 and b_θ of, respectively, the spike and the slab distribution in (3.5). To the scale parameter of the slab distribution, b_θ , we assign a gamma hyperprior $b_\theta \sim \mathcal{G}(a_2, b_2)$ with the hyperparameters a_2 and b_2 chosen such as to allow a rather flat distribution. With the spike distribution we aim at setting the hyperparameters in such a way, that the prior expectation of θ_{hk} is some small number, for example 0.05 as in [Legramanti et al. \(2020\)](#). Choosing such a hyperprior introduces flexibility also in the spike as θ_{hk} can be bigger or smaller than the expected value, depending on the data. The mean of the inverse gamma distribution with parameters α and β is defined as $\beta/(\alpha - 1)$, which leads to the condition $b_0/(\alpha_0 - 1) = 0.05$. For α_0 reasonably big (the reason why this is a reasonable choice for our model is explained below), we need to choose the parameters of the hyperprior on b_0 in such a way that the mean of θ_{hk} at the spike is approximately 0.05, which is easily done with the gamma hyperprior $b_0 \sim \mathcal{G}(a_1, b_1)$.

The choices of the spike and slab shape parameters a_0 and a_θ have to guarantee that the regions where the spike distribution of the elements λ_{ihk} of the factor loading matrix dominates the slab distribution are centered around 0, while the slab distribution dominates the spike distribution in the tails. Since marginally, λ_{ihk} follows a univariate t -distribution, a necessary condition for that is that the degrees of freedom parameter a_0 in the spike is considerably larger than the degrees of freedom parameter a_θ in the slab.

Finally, following the idea to learn all the influential parameters from data, we assign a gamma hyperprior to the strength parameter of the beta prior for the slab probabilities $\alpha_B \sim \mathcal{G}(a_\alpha, b_\alpha)$. Our specific choices of hyperparameters are summarized in Table 1.

Appendix B: Further details of the MCMC algorithm

B.1 Initialisation of the MCMC algorithm

When constructing an MCMC algorithm involving mixture and factor models, it is often the case that starting values are influential in defining the path of the chain. Hence, in order to minimise the probability of the chain being stuck in areas with low posterior probability, initialisations of the model parameters should be chosen carefully. Here we discuss the initialisations and starting values for our model in more details.

The initial splitting of the data into the starting number of clusters K_0 is done via k-means clustering (using R-package *mclust*) to achieve reasonably balanced initial cluster sizes, as using hierarchical clustering to initialise cluster labels often gives heavily imbalanced starting values. K_0 is chosen conservatively and should be clearly overfitting. We follow the suggestion in [Frühwirth-Schnatter et al. \(2021\)](#) and take it approximately two or three times the expected number of clusters in the data set. Cluster means are initialised with the k-means cluster centres and the initial values of the cluster weights η_k , $k = 1, \dots, K$ are sampled from the symmetric Dirichlet distribution $\mathcal{D}_K(\frac{1}{K})$. The Dirichlet concentration parameter $\alpha_{\mathcal{M}}$ is initialised as the mean of its prior distribution $\mathcal{F}(\nu_l, \nu_r)$.

The strength parameter of the prior for slab probabilities α_B is initiated at the mean of its prior distribution $\mathcal{G}(a_\alpha, b_\alpha)$. The initial allocation of the columns of the cluster-specific factor loading matrices Λ_k to spike and slab is done according to the slab probabilities τ_{hk} initiated from the $\mathcal{B}(\frac{\alpha_B}{H}, 1)$ prior. The spike and slab variances θ_{hk} are initiated as the means of their respective prior distributions.

Special attention should be given to the initialisation of the cluster covariance matrices Ω_k . In a classical clustering model, they would be given an inverse Wishart prior with some carefully tuned hyperparameters and initialised from this prior. However, in the $(MF)^2A$ model, the cluster covariance matrices are defined as $\Omega_k = \Lambda_k \Lambda_k^T + \Xi_k$ and thus have a far more flexible structure than the inverse Wishart prior. To ensure a proper functioning of the algorithm in the initial phase, it is important for the cluster-specific covariance matrices Ω_k to be closely linked to the data. If both Λ_k and Ξ_k are initiated from their respective priors, this would very likely take some iterations to achieve and might lead to the chain being stuck in a region of parameter space with low likelihood. As a solution to this problem, we suggest initiating Ω_k for all $k = 1, \dots, K$ from the estimator suggested in [Frühwirth-Schnatter and Lopes \(2018\)](#) for the sample precision matrix in the context of sparse Bayesian factor models. This estimator combines the sample information with an inverse Wishart prior $\Omega_k \sim \mathcal{IW}_p(v_0, (v_0 \mathbf{S}_0)^{-1})$. Provided that the data are standardized, this yields following estimator:

$$\widehat{\Omega} = (v_0 + T/2)^{-1} (v_0 \mathbf{S}_0 + 0.5 \sum_{t=1}^T \mathbf{y}_t \mathbf{y}_t^T). \quad (\text{B.1})$$

Based on the hyperparameters $v_0 = 3$ and $\mathbf{S}_0 = \mathbf{I}_p$, the estimator $\widehat{\Omega}$ is used to initiate the cluster covariance matrix Ω_k for each of the K clusters, thus they all are the same at

the first iteration of the MCMC sampler. For unstandardised (but demeaned) data, this estimator can be viewed as an estimator for the sample correlation matrix. In this case the estimator should be appropriately scaled using the diagonal entries of the sample covariance matrix ([Wang et al. \(2015\)](#)).

B.2 Details of the post-processing algorithm

One of the properties of finite mixture models is their invariance to relabelling of the components of the mixture, a phenomenon first investigated in [Redner and Walker \(1984\)](#) and attributed to the symmetry in the likelihood of the model parameters. Therefore, before conducting any further inference, the MCMC output should undergo a post-processing treatment to ensure the correct grouping of the model components into clusters and the correct representation of the parameters of the cluster-specific factor models. There are multiple various solutions to this problem suggested in the literature, see, for example, [Grün and Leisch \(2009\)](#), [Stephens \(2002\)](#), and [Papastamoulis \(2016\)](#), amongst others.

Following [Frühwirth-Schnatter \(2011\)](#), we handle the label switching problem by working with the point process representation of the MCMC draws and choosing only the \tilde{M} draws with the number of active components equal to the posterior mode \hat{K}_+ of K_+ . This representation allows to identify the model and solve the label switching problem by performing an unsupervised clustering of the mixture parameters at each simulated draw. Providing the MCMC algorithm converged, these draws should cluster around the underlying true finite mixture distributions. Clustering is performed via a k -means algorithm (with a slight modification), available in the R package *mclust*. The package performs clustering with Gaussian mixtures (of which k -means clustering is a special case, see [Frühwirth-Schnatter et al. \(2019\)](#), Chapter 8 for more details) to determine the predefined number of clusters, which we set equal to $t\hat{K}_+$.

We modify the k -means algorithm as follows. To include information from the cluster-specific factor models, we cluster around $(\mu_k^\top \log |\Omega_k| \log(\text{tr}(\Omega_k)) \log(v_k^{\max}/v_k^{\min}))^\top$, where v_k^{\max} and v_k^{\min} are, respectively, the biggest and the smallest eigenvalue of Ω_k . This produces a classification index $J_k^{(m)} \in \{1, \dots, \hat{K}_+\}$ for all draws $m = 1, \dots, \tilde{M}$ with \hat{K}_+ clusters. If $\rho_m = (J_1^{(m)}, \dots, J_{\hat{K}_+}^{(m)})$ is a permutation of $\{1, \dots, \hat{K}_+\}$, a unique labelling is achieved and the cluster-specific model parameters and the latent cluster allocation indicators $S_t^{(m)}$, $t = 1, \dots, T$ are reordered through ρ_m . The draws corresponding to ρ_m s which are not a permutation of $\{1, \dots, \hat{K}_+\}$ are removed and we proceed with the remaining \tilde{M} draws.

We compute the inferred number of factors \hat{H}_k in each of the clusters as the mode of the number of active factors in each cluster over the \tilde{M} draws. The estimated factor-dimensions \hat{H}_k should satisfy the constraint $\hat{H}_k \leq \frac{p-1}{2}$ to ensure the variance identification condition of [Anderson and Rubin \(1956\)](#) and to enable identification of the cluster-specific factor models in each cluster (see Appendix C). We remove all draws in which the number of active factors in any cluster is not equal to \hat{H}_k and denote the number of remaining posterior draws by M . Thus, at this stage, the cluster-specific factor

loading matrices have \hat{H}_k active and $H - \hat{H}_k$ inactive columns (with the respective indicator $\gamma_{hk} = 1$ and $\gamma_{hk} = 0$). For each of the M draws, we keep only the active columns of each cluster-specific loading matrix and these sparsified factor loading matrices Λ_k with \hat{H}_k columns are then used for the calculation of cluster-specific covariance matrices $\Omega_k = \Lambda_k \Lambda_k^\top + \Xi_k$, see Section 5.

The M draws $\Lambda_k^{(m)}, m = 1, \dots, M$ of the cluster-specific factor loading matrices Λ_k , however, are not uniquely identified, since we did not impose any constraints on these matrices during MCMC sampling to prevent rotational invariance such as, for instance, a GLT structure (Frühwirth-Schnatter et al., 2023). Rather, we address rotational invariance *a posteriori* in a post-processing step and apply the MatchAlign algorithm introduced in Poworoznek et al. (2021) to resolve rotational invariance for all M posterior draws of the cluster-specific factor loading matrices Λ_k . Details of this algorithm are discussed in Appendix C.1 and illustrated for simulated data in Appendix C.2.

Appendix C: Identification of cluster-specific factor loading matrices

C.1 Identification of cluster-specific factor loading matrices

The design of the dynamic $(MF)^2A$ model allows the possibility of econometric identification of the cluster-specific factor models. The finite and fixed dimension of the factor loading matrices greatly simplify the application of identification methods, especially when variance identification is considered.

This section provides a brief review of the identification problem in factor models and its solutions. One issue concerns how to ensure that for any two representation

$$\Omega_k = \Lambda_k \Lambda_k^\top + \Xi_k \quad \text{and} \quad \Omega_k = \Theta_k \Theta_k^\top + \Delta_k \quad (\text{C.1})$$

of the cluster-specific covariance matrices Ω_k , the idiosyncratic covariance matrices $\Xi_k = \Delta_k$ are identical and, hence, the cross-covariance matrix $\Lambda_k \Lambda_k^\top = \Theta_k \Theta_k^\top$ is uniquely identified. This is the problem of variance identification, which was first addressed by Anderson and Rubin (1956). Their row deletion property, which states that whenever an arbitrary row is deleted from Λ_k , two disjoint submatrices of rank H_k remain, is a sufficient condition for variance identification. This imposes an upper bound on the number of factors, namely

$$H_k \leq (p - 1)/2, \quad \forall k = 1, 2, \dots \quad (\text{C.2})$$

and variance identification can easily fail, if the number of factors H_k is too high. A failure of variance identification means that the variance decomposition in (C.1) is not unique and $\Lambda_k \Lambda_k^\top$ is not uniquely identified from a given Ω_k . A recent, detailed discussion of variance identification and how it can be verified is provided by Frühwirth-Schnatter et al. (2023). For dense factor models, where all factor loadings are unconstrained, condition (C.2) is sufficient for variance identification (except for a

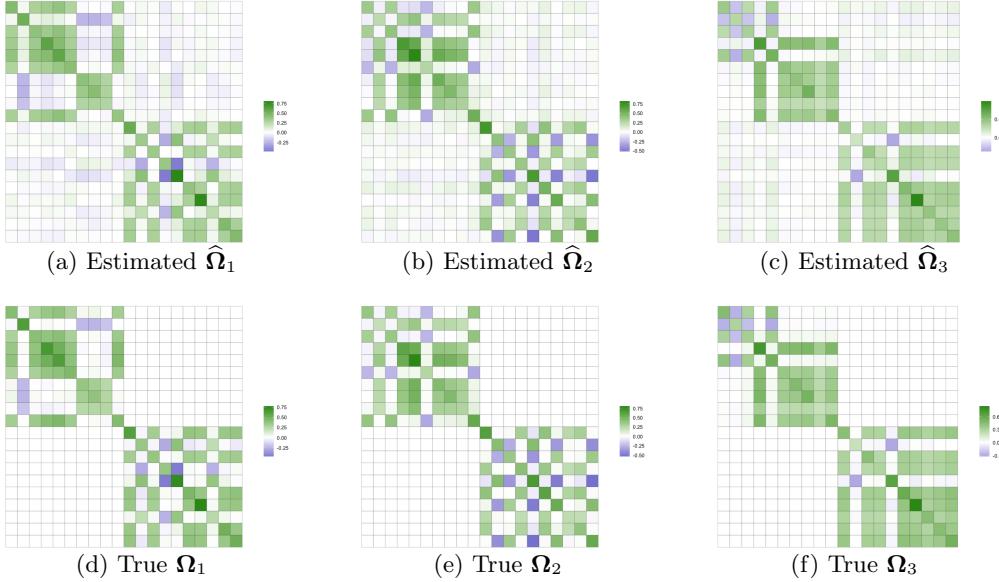


Figure C.1: Cluster-specific covariance matrices for a data set with block-diagonal Λ_k s reconstructed by the dynamic $(MF)^2A$ model (top row) and the true cluster-specific covariance matrices (bottom row).

set of measure zero). For sparse factor models, where each factor loading can take the value 0 with a positive probability this problem is more complicated and additional conditions on the number of non-zero elements in each column of a factor loading matrix need to be applied (see Frühwirth-Schnatter et al. (2023) for a extensive discussion of this issue). As in our model we set an upper bound on the highest possible number of factors in each cluster at $H_k \leq (p - 1)/2$ and deal with a dense setting, the variance identification condition is satisfied.

Once the variance identification has been achieved and $\Lambda_k \Lambda_k^T$ has been uniquely identified, there remains a rotational invariance problem. It arises from the fact, that the decomposition $\Omega_k = \Lambda_k \Lambda_k^T + \Xi_k$ will also hold for any semi-orthogonal matrix \mathbf{P} ($\mathbf{P} \mathbf{P}^\top = \mathbf{I}$) and $\Theta_k = \Lambda_k \mathbf{P}$, $\mathbf{g}_t^k = \mathbf{P}^T \mathbf{f}_t^k$, in which case the two models

$$\mathbf{y}_t - \boldsymbol{\mu}_k = \Lambda_k \mathbf{f}_t^k + \boldsymbol{\epsilon}_t \quad \text{and} \quad \mathbf{y}_t - \boldsymbol{\mu}_k = \Theta_k \mathbf{g}_t^k + \boldsymbol{\epsilon}_t$$

are observationally equivalent. This problem can either be addressed *a priori* via imposing restrictions on the elements of Λ_k , such as, for example, setting the upper diagonal elements equal to 0 and requiring the diagonal elements to be positive so that Λ_k represents a positive lower triangular (PLT) matrix. This approach is very popular in econometrics literature and has first been implemented by Geweke and Zhou (1996) followed by many others (see, for example, Lopes and West (2004) and Carvalho et al. (2008)). However, this approach introduces a particular order dependence amongst observations which can be rather restrictive and requires some prior knowledge of the

dependencies within the data. Another popular approach is to address the rotational invariance *a posteriori* in a post-processing step by using some type of orthogonalisation procedure to rotationally align the samples from each iteration of the MCMC algorithm. In the paper, we employ such an a posteriori identification method to solve rotational invariance of cluster-specific factor loading matrices.

Before applying any identification procedure, the MCMC posterior draws should undergo the post-processing treatment described in detail in Section 4.2. In a first step, only the draws with the estimated number of clusters are chosen and label switching between clusters is accounted for. In a second step, only the draws with the estimated number of active factors in each cluster are taken and the columns corresponding to the inactive factors are deleted from the cluster-specific factor loading matrices $\Lambda_k^{(m)}$. To solve rotational invariance, we apply the MatchAlign algorithm introduced in Poworoznek et al. (2021). This algorithm is applied to MCMC samples of the cluster-specific factor loading matrices $\Lambda_k^{(m)}$ and cluster specific factors $\mathbf{f}_k^{(m)}$ in a post-processing step. The idea of the method is to split the problem of rotational invariance into two steps, where at the first one the orthogonal ambiguity is handled by performing a Varimax procedure (Kaiser (1958)) on each sample of $\Lambda_k^{(m)}$, and in the second step the columns of each posterior sample of $\Lambda_k^{(m)}$ are matched with the columns of a reference matrix, thus solving the column permutation and sign switching problem. The reference matrix is constructed by computing the ratio of the largest and smallest singular values of $\Lambda_k^{(m)}$ and choosing the matrix with the median value of this ratio. The posterior samples are then matched with the reference matrix via minimizing a loss function in a greedy procedure.

C.2 Illustrating identification for simulated data

We illustrate the results of applying the above described identification procedure to the estimated factor loadings on two synthetic examples. First, we generated a data set with $p = 20$ variables, 3 equally weighted clusters and $H_k = 4$ factors in each cluster, and $T = 300$ observations. The rest of the parameters are similar to those in Section 5.1, apart from the factor loadings. The factor loadings are generated in a similar manner to one of the simulation study scenarios in Frühwirth-Schnatter et al. (2023). More specifically, they form two blocks, with the first 10 variables loading on factors 1 and 2 and the remaining 10 variables loading on factors 3 and 4. The non-zero factor loadings are generated as $\lambda_{ihk} = (-1)^{b_{ihk}}(1 + 0.1\mathcal{N}(0, 1))$, where the exponent b_{ihk} is a binary variable with $Pr(b_{ihk} = 1) = 0.2$. All the loadings within a block are non-zero, which ensures that there are no zero rows in the factor loading matrices and variance identification is guaranteed with the maximum possible number of latent factors in each cluster $H = 9$ satisfying the upper limit. The sampler was run for 50,000 iterations with 10,000 of them discarded as a burn-in both for this as well as for the second example.

The model correctly recovered the partition of the data into three clusters and the true number of four factors in each cluster. Figure C.1 shows the estimated (top row) and the true (bottom row) cluster-specific covariance matrices. The true covariance matrices have been appropriately scaled to match the standardised data, as described in Section 5. The block structure of the covariance matrix is nicely recovered in all

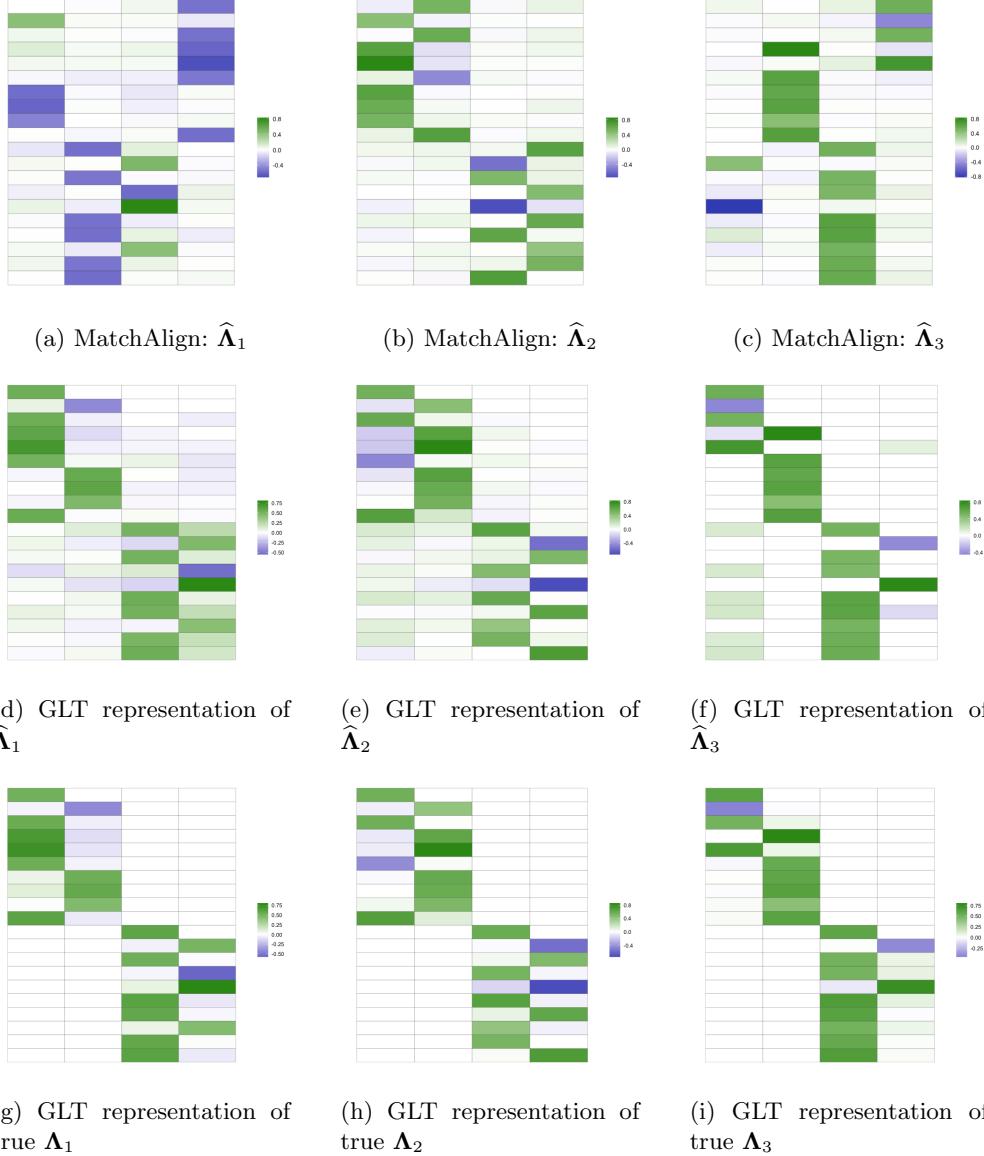


Figure C.2: Cluster-specific factor loading matrices (with block-diagonal structure) identified with the MatchAlign algorithm via the R-package "infinitefactor", raw (top row) and corresponding GLT representation (second row from the top), and GLT representation of the true cluster-specific factor loading matrices of the simulated data (bottom row).

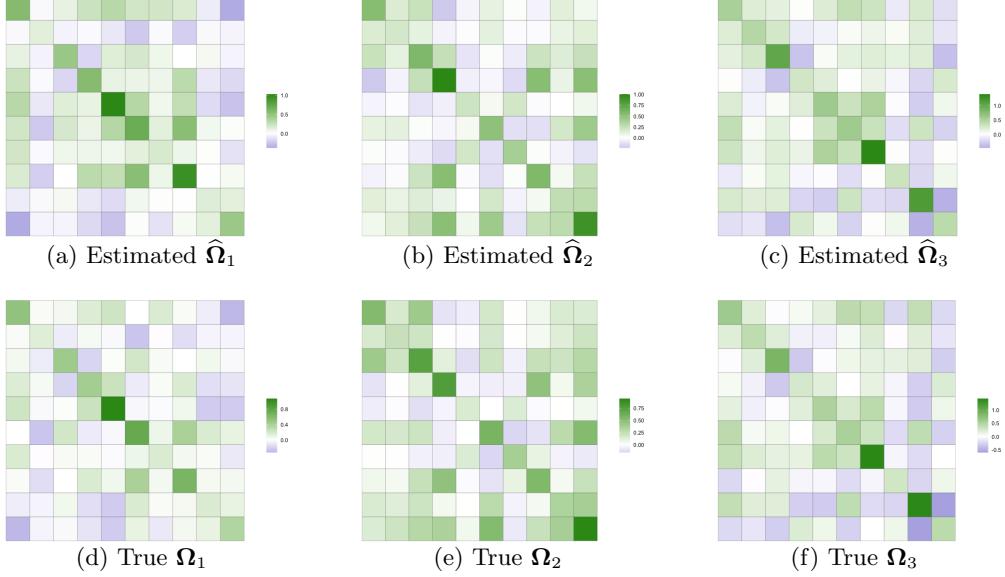


Figure C.3: Cluster-specific covariance matrices for a data set with dense Λ_{ks} reconstructed by the dynamic $(MF)^2A$ model (top row) and the true cluster-specific covariance matrices (bottom row).

three clusters and the MSE between the estimated and true covariance matrices are $MSE_{\Omega_1} = 0.041$, $MSE_{\Omega_2} = 0.038$, and $MSE_{\Omega_3} = 0.034$.

The result of applying the MatchAlign algorithm is illustrated in Figure C.2. The top row depicts the loading matrices identified via the MatchAlign algorithm of [Poworoznek et al. \(2021\)](#) using the R-package *infinitefactor*. The question is how to compare the estimated with the true, simulated factor loading matrices. As shown recently in [Frühwirth-Schnatter et al. \(2023\)](#), any unconstrained loading matrix can be rotated into a uniquely defined loading matrix which takes the form of a generalised lower triangular (GLT) matrix. Hence, this GLT representations can be used to compare loading matrices that are defined w.r.t. to different factor bases. To enable better visual comparability of the estimated and simulated factor loading matrices, they are presented in their GLT form in, respectively, the second and the third row. The true factor loading matrices are also appropriately scaled, i.e. premultiplied by \mathbf{S}^{-1} . One can see that the block structure and the sign pattern of column elements are recovered quite well. As the dynamic $(MF)^2A$ model in its current form does not include shrinkage of single factor loadings in active columns of factor loading matrices, some noise in the block of zero factor loadings is unavoidable, however, the non-zero elements are clearly distinguishable.

As another illustration, a data set with $p = 10$ variables, $T = 200$ observations, 3 clusters and $H_k = 4$ factors was generated, with all other parameters exactly as in Simulation Study 1, meaning that this time all cluster-specific factor loading matrices

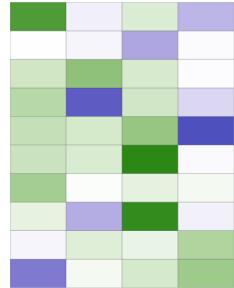
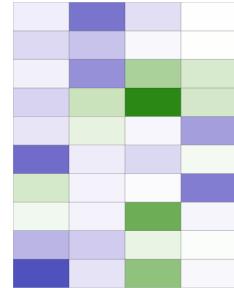
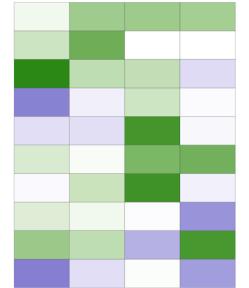
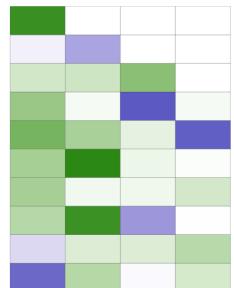
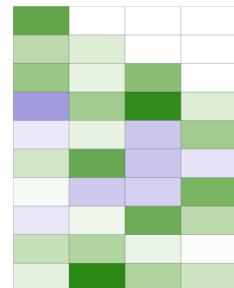
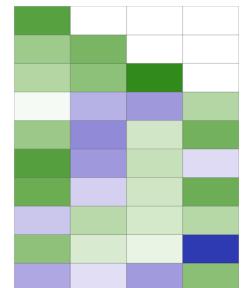
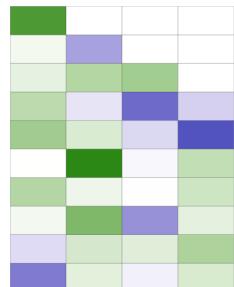
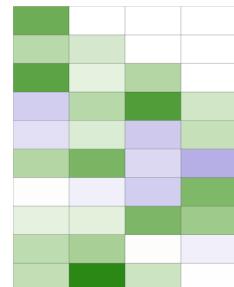
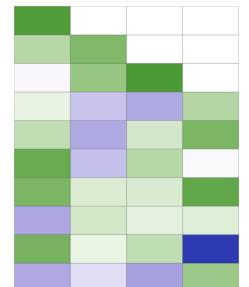
(a) MatchAlign: $\hat{\Lambda}_1$ (b) MatchAlign: $\hat{\Lambda}_2$ (c) MatchAlign: $\hat{\Lambda}_3$ (d) GLT representation of $\hat{\Lambda}_1$ (e) GLT representation of $\hat{\Lambda}_2$ (f) GLT representation of $\hat{\Lambda}_3$ (g) GLT representation of true Λ_1 (h) GLT representation of true Λ_2 (i) GLT representation of true Λ_3

Figure C.4: Cluster-specific factor loading matrices (dense) identified with the MatchAlign algorithm via the R-package "infinitefactor", raw (top row) and corresponding GLT representation (second row from the top), and the GLT representation of the true cluster-specific factor loading matrices of the simulated data (bottom row).

are dense. The model recovered the partition into 3 clusters almost correctly (with the ARI of 0.95 and three misclassified observations out of 200) and 4 active factors were identified in each cluster. The estimated and true (scaled) covariance matrices are presented in Figure C.3. The MSE between the estimated and true covariance matrices are $MSE_{\Omega_1} = 0.016$, $MSE_{\Omega_2} = 0.012$, and $MSE_{\Omega_3} = 0.013$. Loading matrices were again identified with MatchAlign algorithm and presented in Figure C.4, where again the GLT representation is used in each cluster to compare the estimated factor loading matrix with the true loading matrix underlying the simulated data.

Appendix D: Further details for Simulation Study 2

Figure D.5 illustrates the extent of intermixing between the clusters by showing pairwise scatter plots for five randomly chosen variables for the first replicate data set in Simulation Study 2.

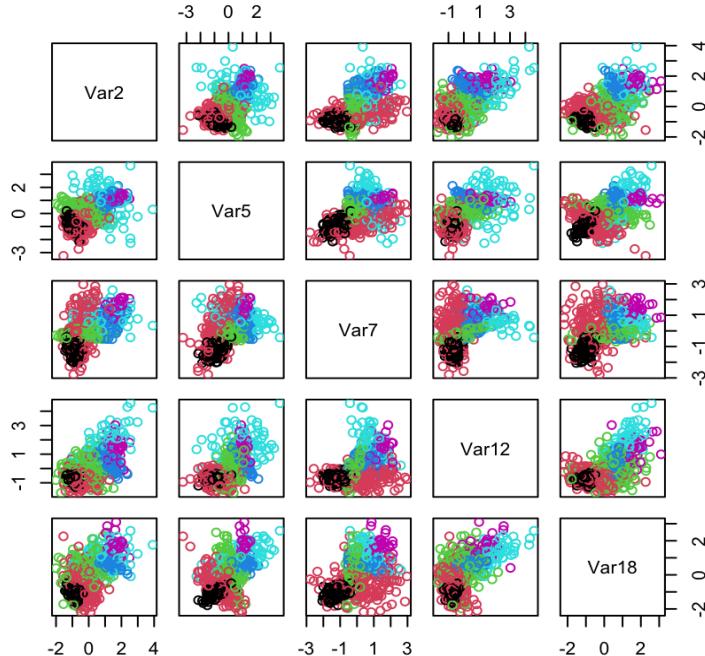


Figure D.5: Pairwise scatter plots of five randomly chosen variables from one of the replicate data sets in Simulation Study 2 with $p = 20$ and $T = 700$ demonstrating the overlap between the six clusters.

Appendix E: Benchmark data applications

In this section we evaluate the performance of the dynamic $(MF)^2 A$ on several publicly available benchmark data sets, which are often used in the literature to test MFA models. We compare the performance of our model against three other MFA models. The first model is the IMIFA model, fit via the R package *IMIFA* ([Murphy et al. \(2020\)](#)), which is the only one that can be compared with our model in terms of flexibility, in a sense that it also provides fully automatic inference on the number of clusters and cluster-specific factors and allows factors to differ between clusters. The other two models, namely overfitting Bayesian mixtures of factor analysers, fit via the R package *fabMix* ([Papastamoulis \(2020\)](#)), and parsimonious Gaussian mixture models, fit via the R package *pgmm* ([McNicholas et al. \(2018\)](#)), are less flexible and only allow the same number of factors in each cluster. For the sake of simplicity, we will refer to those models by the names of the R packages which were used to fit them, namely *IMIFA*, *fabMix* and *pgmm*. Unless otherwise specified, the data are standardised (demeaned and unit-scaled) before fitting our Bayesian MFA model and the same values of hyperparameters are used as reported in Table 1 for the simulated data. Unless otherwise specified, the sampler is run for 50,000 iterations with 20% of them discarded as a burn-in.

E.1 Coffee data set

The coffee data set, first introduced in [Streuli \(1973\)](#), is one of the benchmark data sets often used to evaluate the performance of clustering and MFA models (see, e.g. [McNicholas and Murphy \(2008\)](#), [Papastamoulis \(2018\)](#)) and is available in the *pgmm* R package. The data consists of $T = 43$ coffee samples from 29 countries collected from beans corresponding to the Arabica and Robusta species. For each sample 13 variables are observed: water, pH value, fat, chlorogenic acid, bean weight, free acid, caffeine, neochlorogenic acid, extract yield, mineral content, trigonelline, isochlorogenic acid and total chlorogenic acid. Following [McNicholas and Murphy \(2008\)](#), we excluded the total chlorogenic acid from the analysed data since it is the sum of the chlorogenic, neochlorogenic and isochlorogenic acids, so in the end there are $p = 12$ variables in the data set.

Table E.1 illustrates the performance of all four models in estimation of the coffee data set. It is natural to assume that different coffee bean species, namely Arabica and Robusta, should correspond to different clusters. The ARI is computed on the basis of the known classifications into Arabica and Robusta coffee bean species. All models except the parsimonious Gaussian mixture model (*pgmm*), were able to identify the correct number of clusters and to uncover the correct partition.

Regarding the poor performance of the parsimonious Gaussian mixture model (*pgmm*) for this data set, it must be mentioned that the parsimonious Gaussian mixture model is very sensitive to the specification of the initial range of possible values for the number of clusters and factors which the model explores as well as the choice of either random or k -means starting points (this problem has also been mentioned in [Papastamoulis \(2018\)](#)). For example, the classification results for the coffee data set with two clusters and the correct partition, which are reported in [McNicholas and Murphy \(2008\)](#),

Model	# clusters	# factors	ARI
Dynamic $(MF)^2A$	2	1, 2	1
IMIFA	2	3, 5	1
<i>pgmm</i>	5	1	0.32
<i>fabMix</i>	2	1	1

Table E.1: Results of fitting the dynamic $(MF)^2A$ model against a range of benchmark MFA models on the coffee data set. Note that the number of factors is estimated to be same for all clusters in *pgmm* and *fabMix* by model design.

emerge when the range of possible number of cluster is set between two and three and the range of possible number of factors is set between one and three. However, having found that choice rather restrictive, we choose the slightly wider range of one to five for both the number of clusters and cluster-specific factors⁵, thus aiming for somewhat more flexibility. The model chosen on the basis of the integrated complete-data likelihood (ICL) criterion reports five clusters with one factor in each cluster, splitting the "Arabica" beans into three groups and "Robusta" beans into two.

E.2 Italian wines data set

The Italian wines data set ([Forina et al. \(1986\)](#)), available in the *pgmm* R package, is another benchmark data set employed for assessing the performance of clustering and MFA models (see, e.g. [Papastamoulis \(2018\)](#), [McNicholas and Murphy \(2008\)](#)). It contains $p = 27$ variables measuring chemical and physical properties of $T = 178$ wines collected over the period 1970 – 1979. The wines originate from the Piemont region of Italy and belong to one of three types, namely Barolo, Grignolino and Barbera. We expect the classification algorithm to recognise three clusters which correspond to the three wine types.

The results of applying the dynamic $(MF)^2A$ model and the three alternative MFA models to the Italian wines data set are presented in Table E.2. The assumed true cluster assignments were computed on the basis of the known classifications into Barolo, Grignolino and Barbera wine types. The confusion matrices between the estimated and the true cluster assignments are given in Table E.3.

For this data set, the best clustering performance is delivered by the *pgmm* model, which produced an almost perfect classification, while the other three models overestimated the number of clusters. The dynamic $(MF)^2A$ model essentially put most of Barolo and Grignolino wines in one cluster, and the Barbera wines into one separate cluster. Similarly, the *fabMix* model put all observations belonging to Barbera wine brand to a separate cluster but struggled with Barolo and especially Grignolino wines spreading them across four other clusters. The IMIFA model estimated ten clusters,

⁵We used the same range of possible values for the number of cluster-specific factors for the simulation with overfitting Bayesian mixtures of factor analysers via *fabMix* package.

Model	# clusters	# factors	ARI
$Dynamic (MF)^2A$	4	4, 1, 4, 1	0.48
IMIFA	10	3, 6, 4, 5, 3, 2, 3, 5, 3, 4	0.72
<i>pgmm</i>	3	4	0.96
<i>fabMix</i>	5	1	0.66

Table E.2: Results of fitting the dynamic $(MF)^2A$ model against a range of benchmark MFA models on the Italian wines data set. Note that the number of factors is estimated to be same for all clusters in *pgmm* and *fabMix* by model design.

Dynamic $(MF)^2A$					IMIFA										
Cluster	1	2	3	4	Cluster	1	2	3	4	5	6	7	8	9	10
Barolo	0	2	57	0	Barolo	49	0	0	0	7	0	0	0	3	0
Grignolino	0	2	65	4	Grignolino	0	54	12	0	1	0	0	1	0	3
Barbera	48	0	0	0	Barbera	0	0	0	41	0	5	2	0	0	0

pgmm			fabMix							
Cluster	1	2	3	Cluster	1	2	3	4	5	
Barolo	0	59	0	Barolo		54	4	1	0	0
Grignolino	1	1	69	Grignolino		9	3	14	1	44
Barbera	48	0	0	Barbera		0	0	0	48	0

Table E.3: Confusion matrices between the estimated and true cluster assignments of the Italian wines data set. The estimated cluster assignments are provided by $(MF)^2A$, IMIFA, *pgmm* and *fabMix* models.

with most of the observations being concentrated in four clusters. Regarding the number of cluster-specific latent factors, both the dynamic $(MF)^2A$ and the *pgmm* models estimated 4 factors in each (significantly filled) cluster. The estimated number of factors by the IMIFA model in bigger clusters is between three and six, while the *fabMix* found only one latent factor.

E.3 Italian olive oils data set

The Italian olive oils data set ([Forina et al. \(1983\)](#)) has also been widely used in the literature for testing clustering and factor-analytical models (see, e.g. [Murphy et al. \(2020\)](#)) and is available in the R package *FlexDir*. The data describe the composition of

Model	# clusters	# factors	ARI (areas)	ARI (regions)
Dynamic $(MF)^2A$	5	2, 1, 1, 4, 3	0.60	0.77
IMIFA	5	2, 3, 3, 6, 3	0.90	0.54
<i>pgmm</i>	5	5	0.59	0.76
<i>fabMix</i>	5	4	0.59	0.76

Table E.4: Results of fitting the dynamic $(MF)^2A$ model against a range of benchmark MFA models on the Italian olive oils data set. Note that the number of factors is estimated to be same for all clusters in *pgmm* and *fabMix* by model design.

Dynamic $(MF)^2A$						IMIFA					
Cluster	1	2	3	4	5	Cluster	1	2	3	4	5
Northern Italy	0	91	0	60	0	Northern Italy	48	0	50	0	53
Sardinia	0	0	0	0	98	Sardinia	0	98	0	0	0
Southern Italy	197	0	126	0	0	Southern Italy	0	0	0	323	0

<i>pgmm</i>						<i>fabMix</i>					
Cluster	1	2	3	4	5	Cluster	1	2	3	4	5
Northern Italy	0	0	88	63	0	Northern Italy	60	0	0	0	91
Sardinia	0	0	0	0	98	Sardinia	0	98	0	0	0
Southern Italy	195	128	0	0	0	Southern Italy	0	0	201	122	0

Table E.5: Confusion matrices between the estimated and true cluster assignments to three areas of the Italian olive oils data set. The estimated cluster assignments are provided by $(MF)^2A$, IMIFA, *pgmm* and *fabMix* models.

eight fatty acids in $T = 572$ Italian olive oils, which originate from three areas: southern and northern Italy and Sardinia. Each area breaks down into several 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, one can assume that the true number of clusters should probably correspond to either three areas or nine regions.

Table E.4 presents the results of applying our dynamic $(MF)^2A$ model and the other three MFA models to the Italian olive oils data. Due to a rather small number of $p = 8$ variables in the data set, the initial number of cluster-specific factors in the $(MF)^2A$

Cluster	Dynamic $(MF)^2A$					Cluster	IMIFA				
	1	2	3	4	5		1	2	3	4	5
North Apulia	0	0	25	0	0	North Apulia	0	0	0	25	0
South Apulia	197	0	9	0	0	South Apulia	0	0	0	206	0
Calabria	0	0	56	0	0	Calabria	0	0	0	56	0
Sicily	0	0	36	0	0	Sicily	0	0	0	36	0
Inland Sardinia	0	0	0	0	65	Inland Sardinia	0	65	0	0	0
Coastal Sardinia	0	0	0	0	33	Coastal Sardinia	0	33	0	0	0
Umbria	0	50	0	0	0	Umbria	0	0	47	0	3
East Liguria	0	40	0	10	0	East Liguria	0	0	0	0	50
West Liguria	0	0	0	51	0	West Liguria	48	0	3	0	0

Cluster	pgmm					Cluster	fabMix				
	1	2	3	4	5		1	2	3	4	5
North Apulia	0	25	0	0	0	North Apulia	0	0	0	25	0
South Apulia	195	11	0	0	0	South Apulia	0	0	197	9	0
Calabria	0	56	0	0	0	Calabria	0	0	1	55	0
Sicily	0	36	0	0	0	Sicily	0	0	3	33	0
Inland Sardinia	0	0	0	0	65	Inland Sardinia	0	65	0	0	0
Coastal Sardinia	0	0	0	0	33	Coastal Sardinia	0	33	0	0	0
Umbria	0	0	50	0	0	Umbria	10	0	0	0	40
East Liguria	0	0	37	13	0	East Liguria	50	0	0	0	0
West Liguria	0	0	0	51	0	West Liguria	0	0	0	0	51

Table E.6: Confusion matrices between the estimated and true cluster assignments to nine regions of the Italian olive oils data set. The estimated cluster assignments are provided by $(MF)^2A$, IMIFA, pgmm and fabMix models.

algorithm, which is usually set at $H = \lfloor (p - 1)/2 \rfloor$, was replaced by $H = p$. As it is unclear if the clustering should be done according to areas or regions, we calculated the ARI for both cases. All four models discovered five clusters and all placed Sardinia into a separate cluster while the clustering assignment of northern and southern Italy differs between models (see Table E.5 and E.6 for the classification into areas and regions, respectively). Interestingly, the IMIFA model delivers a better performance in

clustering into big areas, while the other three models achieve a significantly better clustering result in terms of smaller regions. The major difference in the performance of the IMIFA model is that it places southern Italy into one cluster, while the other models split it into South Apulia and the rest. The IMIFA model also splits northern Italy into three groups, roughly corresponding to the three regions. While the other three models split northern Italy into two groups, the allocation of the regions into these groups varies between models. The dynamic $(MF)^2A$ and the *pgmm* models allocate Umbria and the biggest part of the East Liguria into one cluster, the *fabMix* model groups the biggest part of Umbria with West Liguria. The ARIs of the dynamic $(MF)^2A$, *pgmm* and *fabMix* models are very similar to each other, with the ARI of the dynamic $(MF)^2A$ model being marginally better than the ARIs of the *pgmm* and *fabMix* models.

References

- Anderson, T. and Rubin, H. (1956). “Statistical inference in factor analysis.” *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, Volume V: 111–150.
- Carvalho, C., Chang, J., Lucas, J., Nevins, J., Wang, Q., and West, M. (2008). “High-Dimensional Sparse Factor Modeling: Applications in Gene Expression Genomics.” *Journal of American Statistical Association*, 103(484): 1438–1456.
- Forina, M., Armanino, C., Castino, M., and Ubigli, M. (1986). “Multivariate data analysis as a discriminating method of the origin of wines.” *Vitis*, 25(3): 189–201.
- Forina, M., Armanino, C., Lanteri, S., and Tiscornia, E. (1983). “Classification of olive oils from their fatty acid composition.” *Food Research and Data Analysis*, 189–214.
- Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models*. Springer-Verlag, New York.
- (2011). *Dealing with Label Switching under Model Uncertainty*, chapter 10, 213–239. John Wiley & Sons, Ltd.
- Frühwirth-Schnatter, S., Celeux, G., and Robert, C. P. (eds.) (2019). *Handbook of Mixture Analysis*. Boca Raton, FL: CRC Press.
- Frühwirth-Schnatter, S., Hosszejni, D., and Lopes, H. F. (2023). “When It Counts - Econometric Identification of the Basic Factor Model Based on GLT Structures.” *Econometrics*, 11(4): 26.
- Frühwirth-Schnatter, S. and Lopes, H. (2018). “Sparse Bayesian Factor Analysis when the Number of Factors is Unknown.” *ArXiv 1804.04231*.
- Frühwirth-Schnatter, S., Malsiner-Walli, G., and Grün, B. (2021). “Generalised Mixtures of Finite Mixtures and Telescoping Sampling.” *Bayesian Analysis*, 16(4): 1279–1307.
- Geweke, J. and Zhou, G. (1996). “Measuring the pricing error of the arbitrage pricing theory.” *Review of Financial Studies*, 9(2): 557–587.

- Grün, B., Malsiner-Walli, G., and Frühwirth-Schnatter, S. (2022). “How many data clusters are in the Galaxy data set? Bayesian cluster analysis in action.” *Advances in Data Analysis and Classification*, 16(2): 325–249.
- Kaiser, H. (1958). “The varimax criterion for analytic rotation in factor analysis.” *Psychometrika*, 23(3): 187–200.
- Legramanti, S., Durante, D., and Dunson, D. (2020). “Bayesian cumulative shrinkage for infinite factorizations.” *Biometrika*, 107(3): 745–752.
- Lopes, H. and West, M. (2004). “Bayesian model assessment in factor analysis.” *Statistica Sinica*, 14(1): 41–67.
- Malsiner-Walli, G., Frühwirth-Schnatter, S., and Grün, B. (2016). “Model-based clustering based on sparse finite Gaussian mixtures.” *Statistics and Computing*, 26: 303–324.
- McNicholas, P. and Murphy, T. (2008). “Parsimonious Gaussian mixture models.” *Statistics and Computing*, 18(3): 285–296.
- Murphy, K., Viroli, C., and Gormley, I. (2020). “Infinite Mixtures of Infinite Factor Analyzers.” *Bayesian Analysis*, 15(3): 937–963.
- Papastamoulis, P. (2018). “Overfitting Bayesian mixtures of factor analyzers with an unknown number of components.” *Computational Statistics and Data Analysis*, 124: 220–234.
- (2020). “Clustering multivariate data using factor analytic Bayesian mixtures with an unknown number of components.” *Statistics and Computing*, (30): 485–506.
- Poworoznek, E., Ferrari, F., and Dunson, D. (2021). “Efficiently resolving rotational ambiguity in Bayesian matrix sampling with matching.” *ArXiv: 2107.13783*.
- Schiavon, L. and Canale, A. (2020). “On the truncation criteria in infinite factor models.” *Stat*, 9(1): e298.
- Stephens, M. (1997). “Bayesian Methods for Mixtures of Normal Distributions.” Ph.D. thesis, University of Oxford.
- Streuli, H. (1973). “Der heutige Stand der Kaffeechemie.” In *6th International Colloquium on Coffee Chemistry, Association Scentifique International du Cafe, Bogata, Columbia*, 61–72.

Aaron Molstad

Material list:

Jin, Y. (2024) Kernelized discriminant analysis for multivariate categorical response regression. WG slides.

Jin, Y., Zhang, X. and Molstad, A. J. (2024) Kernelized discriminant analysis for regression with multivariate categorical responses. Unpublished manuscript.

Kernelized discriminant analysis for multivariate categorical response regression

Yisen Jin

Department of Statistics
University of Florida

y.jin@ufl.edu

In collaboration with
Dr. Aaron Molstad (UMN) & Dr. Xin Zhang (FSU)



About myself

I am a Ph.D. candidate in Department of Statistics at University of Florida.

Research interests:

- Statistical & Machine Learning • Multivariate Analysis
- Numerical Optimization • Dimension reduction • Statistical Computing
- Geometric Deep Learning

Advanced courses:

- Probability Theory Statistical Inference Generalized Linear Models
- Computer Vision Machine Learning Differential Geometry
- Convex Optimization Experimental Design Advanced Regression



Review of Linear Discriminant Analysis (LDA)

LDA model assumes that

$$X | Y = k \sim N_p(\mu_k, \Sigma), \quad k \in \mathcal{K}$$

- $\mu_k \in \mathbb{R}^p$: mean vector for the k -th class
- $\Sigma \in \mathbb{S}_+^p$: covariance matrix.

Review of Linear Discriminant Analysis (LDA)

LDA model assumes that

$$X | Y = k \sim N_p(\mu_k, \Sigma), \quad k \in \mathcal{K}$$

- $\mu_k \in \mathbb{R}^p$: mean vector for the k -th class
- $\Sigma \in \mathbb{S}_+^p$: covariance matrix.

And using Bayes rule, we have

$$P(Y = k | X = x) \propto f_k(x) \cdot P(Y = k)$$

Multivariate categorical response regression

We want to model

$$\Pr(Y_1 = v_1, \dots, Y_M = v_M \mid X = x), \quad (v_1, \dots, v_M) \in \mathcal{C}$$

where $X \in \mathbb{R}^p$, and $Y_m \in \{1, \dots, c_m\}$ for $m = 1, \dots, M$.

- Separate Modeling:

$$\Pr(Y_1 = v_1, \dots, Y_M = v_M \mid X = x) \approx \prod_{m=1}^M \Pr(Y_m = v_m \mid X = x)$$

- Aggregate Modeling: construct a synthetic response Y' with $c^* = \prod_{m=1}^M c_m$ many categories, and model $\Pr(Y' \mid X)$ directly.

Multivariate categorical response regression

Our method assumes a variation of normal LDA model

$$(X \mid Y_1 = j_1, \dots, Y_M = j_M) \sim N_p(\mu_{*j_1, \dots, j_M}, \Sigma_*),$$

where $\mu_{*j_1, \dots, j_M} \in \mathbb{R}^p$ and $\Sigma_* \in \mathbb{S}_+^p$ is the covariance matrix.

Bayes' classification rule

$$\arg \max_{\mathbf{v} \in \mathcal{C}} \left\{ \mu_{*\mathbf{v}}^\top \Sigma_*^{-1} (2x_{\text{new}} - \mu_{*\mathbf{v}}) + 2 \log \pi_{*\mathbf{v}} \right\}. \quad (1)$$

Multivariate response LDA

Bayes' classification rule

$$\arg \max_{\mathbf{v} \in \mathcal{C}} \left\{ \mu_{*\mathbf{v}}^\top \Sigma_*^{-1} (2x_{\text{new}} - \mu_{*\mathbf{v}}) + 2 \log \pi_{*\mathbf{v}} \right\}. \quad (2)$$

Problems when sample size n is small-to-moderate:

- $\prod_{m=1}^M c_m$ many mean vectors to estimate.
- Some \mathbf{v} may not be observed in the training data.

Mean estimation via discrete kernelized regression

A natural assumption:

- If \mathbf{v} is close to \mathbf{v}' , then $\mu_{\mathbf{v}}$ should be close to $\mu_{\mathbf{v}'}$.

Mean estimation via discrete kernelized regression

A natural assumption:

- If \mathbf{v} is close to \mathbf{v}' , then $\mu_{\mathbf{v}}$ should be close to $\mu_{\mathbf{v}'}$.

Assumption

There exists a transformation $\phi : \mathcal{C} \rightarrow \mathbb{R}^d$ ($d \geq 1$) such that $\forall \varepsilon > 0$, there exists $\delta > 0$,

$$\|\phi(\mathbf{v}) - \phi(\mathbf{v}')\|_2 \leq \delta \implies \|\mu_{*\mathbf{v}} - \mu_{*\mathbf{v}'}\|_2 \leq \varepsilon$$

Mean estimation via discrete kernelized regression

Let $g_* : \mathcal{C} \rightarrow \mathbb{R}^p$ be the function

$$g_*(\mathbf{v}) := \mu_{*\mathbf{v}} \quad \text{for each } \mathbf{v} \in \mathcal{C}$$

We propose to approximate $g_{*\ell}$ with

$$g_\ell(\cdot) = \eta_\ell + \tilde{g}_\ell(\cdot), \quad \eta_\ell \in \mathbb{R},$$

where

$$\tilde{g}_\ell(\cdot) \in \text{span} \{k(\cdot, \mathbf{y}_i) : i \in [n]\}, \quad \ell \in [p].$$

where $k(\mathbf{y}_i, \mathbf{y}_j) = \langle \phi(\mathbf{y}_i), \phi(\mathbf{y}_j) \rangle$ is the induced kernel function.

Choices for kernel functions

- The weighted Hamming distance kernel

$$k^H(\mathbf{y}_i, \mathbf{y}_j) = \frac{1}{w} \sum_{m=1}^M \mathbf{1}(y_{im} = y_{jm}) w_m$$

- The Eskin measure kernel

$$k^E(\mathbf{y}_i, \mathbf{y}_j) = \frac{1}{M} \sum_{m=1}^M \left\{ \frac{c_m^2}{c_m^2 + 2} + \frac{2}{c_m^2 + 2} \mathbf{1}(y_{im} = y_{jm}) \right\}.$$

- The inverse occurrence frequency (IOF) kernel

$$k^{IOF}(\mathbf{y}_i, \mathbf{y}_j) = \frac{1}{M} \sum_{m=1}^M \left\{ \frac{1}{1 + d_{im}d_{jm}} + \frac{d_{im}d_{jm}}{1 + d_{im}d_{jm}} \mathbf{1}(y_{im} = y_{jm}) \right\}$$

Maximum Likelihood Estimation and Regularization

The negative log-likelihood is given by

$$\frac{1}{n} \sum_{i=1}^n \{x_i - \eta - \alpha^\top k_{\tilde{\mathbf{Y}}}(\mathbf{y}_i)\}^\top \Omega \{x_i - \eta - \alpha^\top k_{\tilde{\mathbf{Y}}}(\mathbf{y}_i)\} - \log \det(\Omega), \quad (3)$$

where $\alpha \in \mathbb{R}^{\tilde{n} \times p}$, $k_{\tilde{\mathbf{Y}}}(\cdot) = (k(\cdot, \tilde{\mathbf{y}}_1), \dots, k(\cdot, \tilde{\mathbf{y}}_{\tilde{n}}))^\top$

Maximum Likelihood Estimation and Regularization

The negative log-likelihood is given by

$$\frac{1}{n} \sum_{i=1}^n \{x_i - \eta - \alpha^\top k_{\tilde{\mathbf{Y}}}(\mathbf{y}_i)\}^\top \Omega \{x_i - \eta - \alpha^\top k_{\tilde{\mathbf{Y}}}(\mathbf{y}_i)\} - \log \det(\Omega), \quad (3)$$

where $\alpha \in \mathbb{R}^{\tilde{n} \times p}$, $k_{\tilde{\mathbf{Y}}}(\cdot) = (k(\cdot, \tilde{\mathbf{y}}_1), \dots, k(\cdot, \tilde{\mathbf{y}}_{\tilde{n}}))^\top$

Recall Bayes' classification rule

$$\arg \max_{\mathbf{v} \in \mathcal{C}} \left\{ \mu_{*\mathbf{v}}^\top \Sigma_*^{-1} (2x_{\text{new}} - \mu_{*\mathbf{v}}) + 2 \log \pi_{*\mathbf{v}} \right\}.$$

In high-dimensional settings, there are two types of sparsity we can explore.

- Sparsity in mean vectors $\mu_{*\mathbf{v}}$.
- Sparsity in discriminant vectors $\beta_{*\mathbf{v}} = \Sigma_*^{-1} \mu_{*\mathbf{v}}$.

Regularized Maximum Likelihood Estimation

- Sparsity in mean vectors (KLDA-M)

$$\frac{1}{n} \text{tr} \{(X - K_{\tilde{\mathbf{Y}}} \alpha) \Omega (X - K_{\tilde{\mathbf{Y}}} \alpha)^\top\} - \log \det(\Omega) + \lambda \|\alpha\|_{1,2} + \frac{\gamma}{2} \|\Omega\|_F^2 \quad (4)$$

- Sparsity in discriminant vectors (KLDA-D), let $\Theta = \alpha \Omega$

$$\frac{1}{n} \text{tr} \{(X \Omega - K_{\tilde{\mathbf{Y}}} \Theta) \Omega^{-1} (X \Omega - K_{\tilde{\mathbf{Y}}} \Theta)^\top\} - \log \det(\Omega) + \lambda \|\Theta\|_{1,2} + \frac{\eta}{2} \|\Omega\|_F^2 \quad (5)$$

where $\|\alpha\|_{1,2} := \sum_{j=1}^p (\sum_{i=1}^{\tilde{n}} \alpha_{i,j}^2)^{1/2}$, and $K_{\tilde{\mathbf{Y}}} \in \mathbb{R}^{n \times \tilde{n}}$ with (i,j) th entry $k(\mathbf{y}_i, \tilde{\mathbf{y}}_j)$

Theoretical results

Theorem 1. Average in-sample mean estimation error

Under some regularity conditions, defining $\hat{g}(\mathbf{v}) = \hat{\alpha}^\top K_{\tilde{\mathbf{Y}}}(\mathbf{v})$ for all $\mathbf{v} \in \mathcal{C}$, it follows that

$$\frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \|g_*(\tilde{\mathbf{y}}_i) - \hat{g}(\tilde{\mathbf{y}}_i)\|_2^2 \leq \frac{36w_*^2 s}{\varphi_{\min}(\Omega_*) \kappa_S} \left(\frac{\tilde{n}_{\max}}{\tilde{n}_{\min}^2} \right) \left\{ \frac{\|K_{\tilde{\mathbf{Y}}}^\dagger\|_F}{\tilde{n}} + \left(\frac{\|K_{\tilde{\mathbf{Y}}}^\dagger\|}{\tilde{n}} \right) \sqrt{2c_2 \log p} \right\}^2$$

with probability at least $1 - p^{1-c_2}$.

Theorem 2. Out-of-sample mean estimation error

Under some regularity conditions, if $\lambda = 4\omega_*(\|K_{\tilde{\mathbf{Y}}}\|_F + \|K_{\tilde{\mathbf{Y}}}\|\sqrt{2c_2 \log p})/n$, then for any $\mathbf{v} \in \mathcal{C}$, we have

$$\begin{aligned} \|\hat{g}(\mathbf{v}) - g_*(\mathbf{v})\|_2 &\leq \|g_*(\mathbf{v}) - M_*^\top K_{\tilde{\mathbf{Y}}}^{\dagger-1} k_{\tilde{\mathbf{Y}}}(\mathbf{v})\|_2 + \\ &\quad \frac{6\|k_{\tilde{\mathbf{Y}}}(\mathbf{v})\|_2 w_* \sqrt{s}}{\kappa_S} \left(\frac{\sqrt{\tilde{n}_{\max}}}{\tilde{n}_{\min}} \right) \left\{ \frac{\|K_{\tilde{\mathbf{Y}}}^\dagger\|_F}{\tilde{n}} + \left(\frac{\|K_{\tilde{\mathbf{Y}}}^\dagger\|}{\tilde{n}} \right) \sqrt{2c_2 \log p} \right\}, \end{aligned}$$

with probability at least $1 - p^{1-c_2}$.

Simulation Study

Randomly select 10 distinct elements of $[p]$, say $\{k_1, \dots, k_{10}\}$.

- **Model A.** Let

$$g_{*k}(\mathbf{y}) = \begin{cases} b_\ell^\top \mathbf{y} / \nu & : k = k_\ell \text{ for some } \ell \in \{1, \dots, 10\} \\ 0 & : \text{otherwise} \end{cases}, \quad k \in [p]$$

- **Model B.** Let

$$\beta_k(\mathbf{y}) = \begin{cases} b_\ell^\top \mathbf{y} / \nu & : k = k_\ell \text{ for some } \ell \in \{1, \dots, 10\} \\ 0 & : \text{otherwise} \end{cases}, \quad k \in [p]$$

And let $g_*(\mathbf{y}) = \Omega_*^{-1} \beta(\mathbf{y})$.

$b_\ell \in \{0, 2\}^M$ with the collection $\{b_\ell\}_{\ell=1}^{10}$ having $10M/2$ components equal to two and $10M/2$ equal to zero in randomly chosen positions.

Simulation study

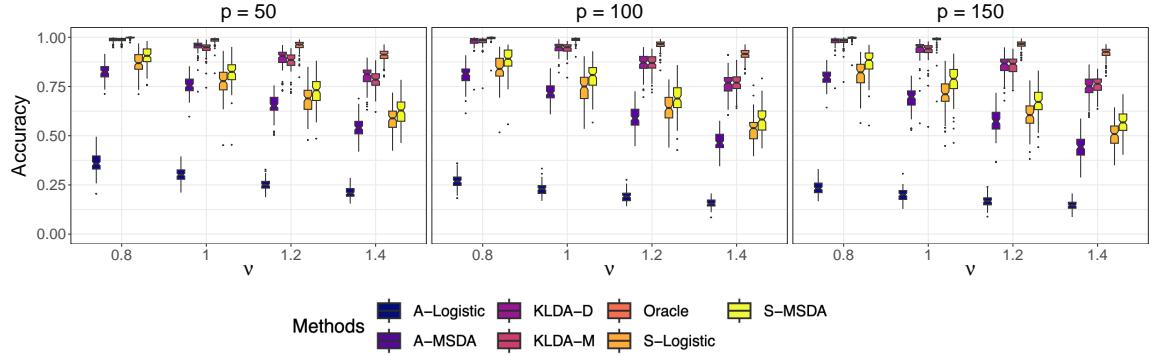


Figure: Model A. Prediction accuracy over 100 independent replications with $(p, \nu) \in \{50, 100, 150\} \times \{0.8, 1.0, 1.2, 1.4\}$. $M = 6, c_m = 2$ for $m = 1, \dots, M$. $n = 200$.

Simulation study

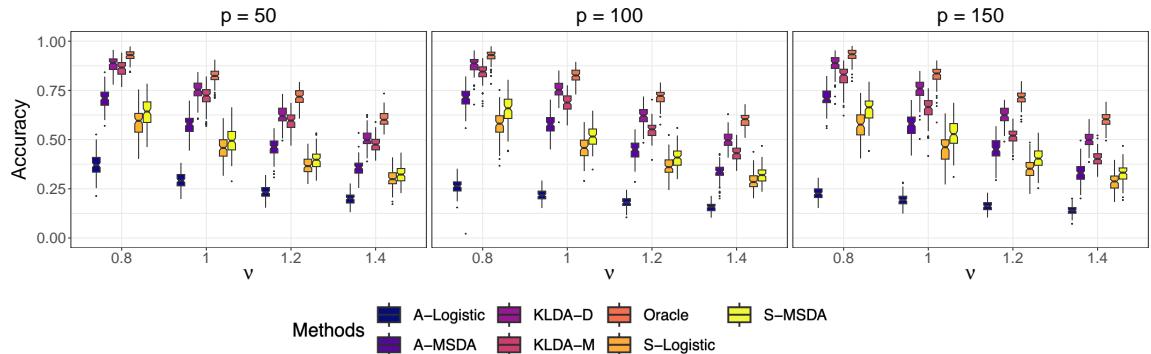


Figure: Model B. Prediction accuracy over 100 independent replications with $(p, \nu) \in \{50, 100, 150\} \times \{0.8, 1.0, 1.2, 1.4\}$. $M = 6, c_m = 2$ for $m = 1, \dots, M$. $n = 200$.

Real Data Analysis

We demonstrate the application of our method on a dataset consisting of gene expression profiles from colon biopsies (Noble et al., 2008).

- 44290 gene expression levels.
 - 202 tissue samples.
 - 3 labels for each sample: patient state (normal/ulcerative colitis), tissue state (inflamed/uninflamed) and anatomical locations (sigmoid colon/terminal ileum/descending colon/ascending colon).



Real Data Analysis

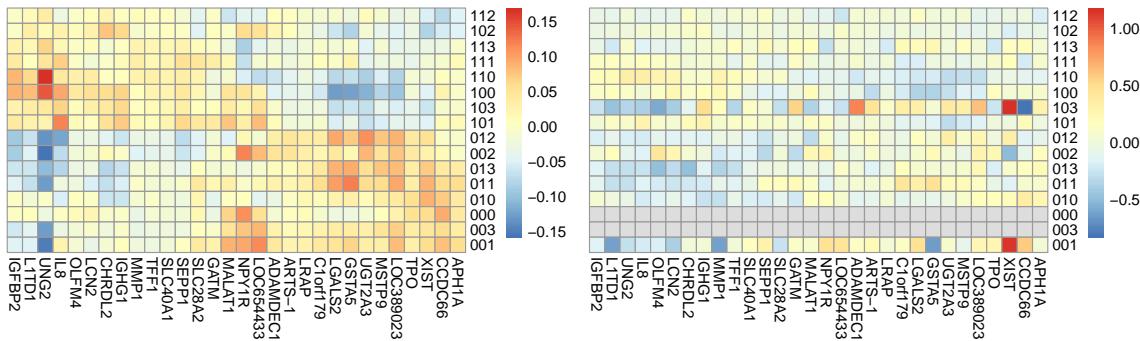


Figure: Left: Mean estimation using KLDA-M with $p = 200$. Each column corresponds to a gene and we include 30 genes. Each row corresponds to a combination of response categories. Right: Sample mean estimation.



Real Data Analysis

n	p	KLDA-M	KLDA-D	S-Logistic	S-MSDA	A-Logistic	A-MSDA
50	100	0.163	0.153	0.148	0.145	0.149	0.157
	200	0.164	0.163	0.152	0.148	0.161	0.152
	300	0.165	0.158	0.153	0.147	0.157	0.144
	400	0.168	0.167	0.147	0.146	0.158	0.146
	500	0.172	0.178	0.151	0.150	0.164	0.153
100	100	0.192	0.191	0.177	0.160	0.187	0.177
	200	0.188	0.180	0.169	0.164	0.181	0.174
	300	0.191	0.184	0.173	0.174	0.190	0.171
	400	0.202	0.186	0.162	0.165	0.187	0.175
	500	0.206	0.200	0.170	0.174	0.191	0.182

Table: Prediction accuracy on GDS3268 dataset over 100 independent replications with $p \in \{100, 200, 300, 400, 500\}$ and $n \in \{50, 100\}$. When $n = 50$, standard errors were never larger than 0.005; when $n = 100$ standard errors were never larger than 0.006.

Kernelized discriminant analysis for regression with multivariate categorical responses

Yisen Jin[†], Xin Zhang^{*}, and Aaron J. Molstad^{*}

Department of Statistics^{†*}, University of Florida, Gainesville, FL, USA

Department of Statistics^{*}, Florida State University, Tallahassee, FL, USA

School of Statistics^{*}, University of Minnesota, Minneapolis, MN, USA

Abstract

Modeling the joint probability mass of multiple categorical variables as a function of predictors is a foundational task in categorical data analysis. When the number of response variables, number of categories per response, and/or the number of predictors is large, existing likelihood-based methods cannot be applied or perform poorly. In this article, we propose a novel approach to multivariate categorical response regression which assumes a variation of the normal linear discriminant analysis model. In order to estimate unknown parameters in way that exploits dependence amongst the response variables, we propose a new penalized likelihood method based on discrete kernel regression. We propose two estimators, each of which can lead to interpretable and parsimonious fitted models. Theoretically, we establish statistical properties of our method and demonstrate a tradeoff between the statistical error and approximation error. Through simulation studies and an application to genomic data, we demonstrate that our method yields better classification accuracy and more interpretable fitted models than existing methods. Software implementing our method, as well as code for reproducing the results in this article, are available for download at <https://github.com/yjin07/kernelizedDA>.

Keywords: Multivariate categorical response regression, categorical data analysis, linear discriminant analysis, kernel methods, convex optimization

1 Introduction

Consider a regression model where the response consists of M distinct categorical variables, each with two or more response categories. Specifically, let Y_1, \dots, Y_M be the random responses where Y_m has numerically coded categorical support $\{1, \dots, c_m\} =: [c_m]$ ($c_m \geq 2$)

*Correspondence: amolstad@umn.edu. X. Zhang's and A. J. Molstad's contributions were supported by NSF DMS-2113590 and NSF DMS-2113589, respectively.

for $m \in \{1, \dots, M\} =: [M]$ ($M \geq 2$) and let $X \in \mathbb{R}^p$ be the predictor. In a multivariate categorical response regression, the goal is to model the conditional probability mass function

$$\Pr(Y_1 = v_1, \dots, Y_M = v_M \mid X = x), \quad (v_1, \dots, v_M) \in \mathcal{C} \quad (1)$$

where $\mathcal{C} = [c_1] \times [c_2] \times \dots \times [c_M]$. Throughout this article, we will use both (v_1, \dots, v_M) and \mathbf{v} to denote arbitrary elements of \mathcal{C} .

There are two extreme approaches to modeling (1). The first and arguably simplest approach is to fit a separate model for each $(Y_m \mid X)$ —for example, using multinomial logistic regression or linear discriminant analysis—then obtain estimates of (1) through the product $\prod_{m=1}^M \widehat{\Pr}(Y_m = v_m \mid X = x)$, where $\widehat{\Pr}(Y_m = v_m \mid X = x)$ is an estimate of the probability that the m th response takes category v_m given the predictor $X = x$. We refer to this approach as “separate modeling” (of Y_1, \dots, Y_M given X). If the responses Y_1, \dots, Y_M are conditionally (on X) independent, this can work well. However, if the response components are conditionally dependent, this approach will be overly restrictive.

On the other extreme, one could instead construct a synthetic (univariate) categorical response variable Y' with $c^* = \prod_{m=1}^M c_m$ many categories, and model $(Y' \mid X)$ directly. That is, one would treat every element of \mathcal{C} as its own category and simply fit a univariate categorical response regression model for $(Y' \mid X)$. We call this approach “aggregate modeling”. Aggregate modeling is the regression analog of modeling counts in a M -way contingency table as a multinomial random variable (Agresti, 2012). In contrast to separate modeling, this approach allows for arbitrary conditional dependence between components of the response (Molstad and Rothman, 2023). However, this additional flexibility comes at a substantial cost: aggregate modeling ignores that Y' is constructed from M distinct response variables. Unless the model fitting procedure explicitly accounts for this, such an approach can be inefficient. To make matters worse, if the sample size is small relative to c^* , it is probable that one will not observe a response from at least one of the c^* many categories

of Y' . In such a scenario, one cannot use maximum likelihood in a straightforward way. If we used a multinomial logistic regression model for $(Y' | X)$, for example, and one of the categories of Y' was never observed the training data, the maximum likelihood estimator of the unknown regression coefficients would not be finite (Agresti, 2012, Section 6.5). More generally, aggregate modeling requires the estimation of an enormous number of parameters relative to separate modeling.

There are many methods for fitting (1) in the categorical data analysis literature. For example, many have proposed link functions that have parameters that can be interpreted in terms of their effects on marginal probabilities and higher-order associations (Molenberghs and Lesaffre, 1999; Ekholm et al., 2000; Glonek and McCullagh, 1995; Glonek, 1996; Lupparelli and Roverato, 2017). The generalized log-linear model of Lang (1996) covers many such models as special cases. These methods are, loosely, intermediate to the two extremes as they explicitly account for the multivariate construction of the response. However, these methods are not applicable when p is large, and/or are difficult to compute when $M \geq 3$.

In machine learning, many adopt a more general version of the separate modeling approach based on the notion of a “classifier chain” (Read et al., 2009; Zhang and Zhou, 2013; Read et al., 2021). A classifier chain is constructed by fitting a particular sequence of conditional models. For example, one would first model $(Y_1 | X = x)$, then $(Y_2 | X = x, Y_1 = y_1)$, and so on. The classification rule for a new observation with predictor x_{new} would thus be the argument $(v_1, \dots, v_m) \in \mathcal{C}$ maximizing the product of the successive conditional probabilities. This approach is more flexible than separate modeling, but requires many ad-hoc choices that may have a significant impact on performance (e.g., in what order to fit the chain or whether to condition on predicted or observed response) and yields fitted models whose parameters are difficult to interpret in terms of (1), the mass function of interest.

To handle large p , Molstad and Rothman (2023) recently proposed to fit the multinomial logistic regression aggregate model using penalized maximum likelihood. Their penalty

allowed predictors to be interpreted as being either irrelevant, affecting only marginal probabilities, or affecting all higher-order associations. When $M \geq 3$, however, the method from Molstad and Rothman (2023) becomes too computationally burdensome to be useful in practice. Along different lines, Molstad and Zhang (2022) proposed to model (1) using a mixture of regressions model motivated by the assumption of a low-rank “functional” probability tensor decomposition. Their method assumes that the tensor-valued function defined by probabilities $\Pr(Y_1 = j_1, \dots, Y_M = j_M | X = x)$ has a low rank decomposition for all $x \in \mathbb{R}^p$ and performs variable selection under this paradigm. However, the low-rank model assumption made by Molstad and Zhang (2022) is not capable of handling arbitrary dependence among all the responses, as is the method we will describe shortly.

In this article, we propose a new model-based method for fitting (1) under the assumption of a normal linear discriminant analysis model (Hastie et al., 2009, Chapter 12). Our method allows one to model complex conditional dependencies between the response variables, and as we will explain, elegantly handles the case where many combinations of response categories are not observed in the training data. Furthermore, our method has parameters which are easily interpretable, and can handle large M , p , and c_m without issue.

2 Kernelized discriminant analysis

2.1 Multivariate linear discriminant analysis model

Our method assumes a variation of the normal linear discriminant analysis model. Letting \mathbb{S}_+^p denote the set of $p \times p$ symmetric positive definite matrices, we assume

$$(X | Y_1 = v_1, \dots, Y_M = v_M) \sim N_p(\mu_{*v_1, \dots, v_M}, \Sigma_*), \quad (v_1, \dots, v_M) \in \mathcal{C}, \quad (2)$$

where $\Sigma_*^{-1} =: \Omega_* \in \mathbb{S}_+^p$ is the unknown precision (inverse covariance) matrix and $\mu_{*v_1, \dots, v_M} \in \mathbb{R}^p$ is the unknown mean vector corresponding to the response category M -tuple (v_1, \dots, v_M) . That is, we assume that conditioned on the M -dimensional categorical response (Y_1, \dots, Y_M)

the predictor X follows a p -dimensional multivariate normal distribution whose mean vector depends on the combination of response categories, but whose covariance is identical across category combinations. Note that indeed, (2) is exactly the linear discriminant analysis model under the aggregate model described in the previous section. While this generality provides the flexibility of the aggregate model, we will estimate the parameters from (2) in a way that exploits the multivariate nature of the response.

Under (2), Bayes' classification rule for a new predictor $x_{\text{new}} \in \mathbb{R}^p$ is given by the M -tuple (v_1, \dots, v_M) maximizing $\Pr(Y_1 = v_1, \dots, Y_M = v_M | X = x_{\text{new}}) \propto f_{*v_1, \dots, v_M}(x_{\text{new}}) \Pr(Y_1 = v_1, \dots, Y_M = v_M)$ where $f_{*v_1, \dots, v_M}(x_{\text{new}})$ is the density of $N_p(\mu_{*v_1, \dots, v_M}, \Sigma_*)$ evaluated at x_{new} and $\Pr(Y_1 = v_1, \dots, Y_M = v_M) = \pi_{*v_1, \dots, v_M}$ is the marginal probability that $(Y_1, \dots, Y_M) = (v_1, \dots, v_M)$. Naturally, $\pi_{*\mathbf{v}} \geq 0$ for all $\mathbf{v} \in \mathcal{C}$ and $\sum_{v_1=1}^{c_1} \cdots \sum_{v_M=1}^{c_M} \pi_{*v_1, \dots, v_M} = 1$. For our regression problem to make sense, however, we require the $\pi_{*\mathbf{v}}$ satisfy $\Pr(Y_m = v_m) > 0$ for all $v_m \in [c_m]$ and $m \in [M]$. Thus restated, Bayes' classification rule is

$$\arg \max_{\mathbf{v} \in \mathcal{C}} \left\{ \mu_{*\mathbf{v}}^\top \Omega_* (2x_{\text{new}} - \mu_{*\mathbf{v}}) + 2 \log \pi_{*\mathbf{v}} \right\}. \quad (3)$$

In practice, one would replace unknown parameters $\pi_{*\mathbf{v}}, \mu_{*\mathbf{v}}$, and Ω_* appearing in (3) with estimates thereof. Classification with respect to a single response component, say Y_1 , also requires estimation of $\pi_{*\mathbf{v}}, \mu_{*\mathbf{v}}$, and Ω_* , as Bayes' classification rule for Y_1 under (2) is the $v_1 \in [c_1]$ maximizing $\Pr(Y_1 = v_1 | X = x_{\text{new}})$, i.e.,

$$\arg \max_{v_1 \in [c_1]} \sum_{v_2=1}^{c_2} \cdots \sum_{v_M=1}^{c_M} \pi_{*v_1, \dots, v_M} \exp \left\{ \left(x_{\text{new}} - \frac{\mu_{*v_1, \dots, v_M}}{2} \right)^\top \Omega_* \mu_{*v_1, \dots, v_M} \right\}. \quad (4)$$

Based on (4), one can see that it is possible for the v_1 maximizing the marginal probability to disagree with the v'_1 from the M -tuple (v'_1, \dots, v'_M) maximizing $\Pr(Y_1 = v'_1, \dots, Y_M = v'_M | X = x_{\text{new}})$. Equation (4) suggests that marginally, the appropriate model for $Y_1 | X$ is the mixture discriminant analysis model (Hastie and Tibshirani, 1996). If one or more

responses from (2) were unobservable, then (2) generates the mixture discriminant analysis model with the same number of mixtures for every class.

Finally, we make two important remarks on the normality assumption in our model (2). First, although the normal assumption may seem restrictive, it is well-known that the standard LDA classifier (with $M = 1$) performs well in a variety of classification tasks (see Michie et al., 1994; Hand, 2006, for example). As explained in Hastie et al. (2009, Section 4.4.5), though LDA makes stronger model assumptions than logistic regression, both rely on linear decision boundaries, and the more restrictive assumptions on the marginal distribution of predictors can improve efficiency and have a beneficial regularizing effect.

Second, in subsequent sections, we will propose new ways to (i) estimate the $\mu_{*\mathbf{v}}$ and the precision matrix Ω_* , and (ii) estimate the discriminant vectors $\beta_{*\mathbf{v}} = \Omega_*\mu_{*\mathbf{v}}$ directly. These discriminant vectors span the same subspace as Fisher-Rao's discriminant vectors (Fisher, 1936; Rao, 1948), which (without the normality assumption) sequentially maximize the ratio of between-class variance and within-class variance. Therefore, our methodology can be used for off-the-shelf dimension reduction, visualization, and plug-in estimation for Fisher-Rao-type discriminant analysis, which has been used for “multi-label” classification in the literature (Park and Lee, 2008; Wang et al., 2010).

2.2 Conditional dependence under model (2)

As mentioned, a major deficiency of the separate modeling approach is that it implicitly assumes that responses are conditionally independent. In this section, we characterize the manner in which the linear discriminant analysis model (2) induces conditional dependence in $(Y_1, \dots, Y_M | X)$. Perhaps surprisingly, under certain restrictions on the precision Ω_* , the mean vectors $\mu_{*\mathbf{v}}$, and their products, the model (2) can imply conditional independence.

To begin, consider an example with $M = 2$ and $c_1 = c_2 = 2$. In this case, the responses

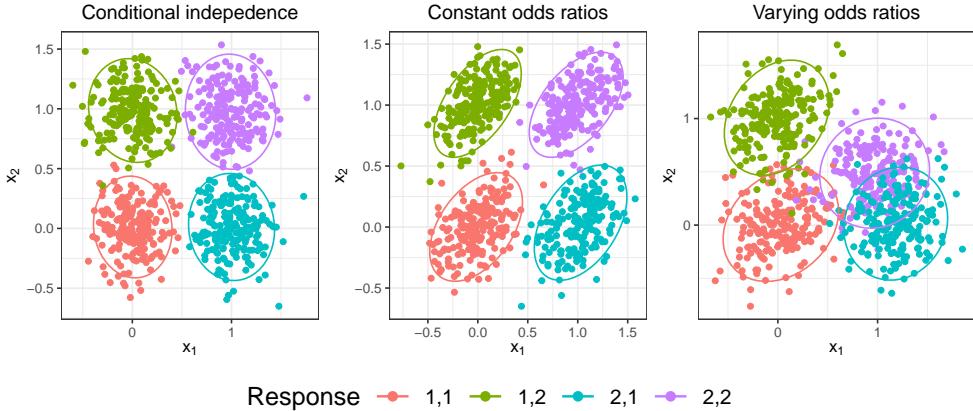


Figure 1: Data generated from $X \mid Y_1, Y_2$ under different types of conditional dependence. Each color represents a different response category combination conditioned on, as denoted in the legend.

are conditionally independent if and only if the odds ratio equals one for all x where

$$\text{OR}(x) = \frac{\Pr(Y_1 = 1, Y_2 = 1 \mid X = x)\Pr(Y_1 = 2, Y_2 = 2 \mid X = x)}{\Pr(Y_1 = 1, Y_2 = 2 \mid X = x)\Pr(Y_1 = 2, Y_2 = 1 \mid X = x)}$$

is the odds ratio at x . To characterize settings when this occurs, we first define the set of discriminant vectors $\beta_{*\mathbf{v}} = \Omega_*\mu_{*\mathbf{v}} \in \mathbb{R}^p$ for all $\mathbf{v} \in \mathcal{C}$.

Remark 1. If $M = 2$, $c_1 = c_2 = 2$ and $\pi_{*\mathbf{v}} = 1/4$ for each $\mathbf{v} \in \mathcal{C}$, then Y_1 and Y_2 are conditionally independent if and only if for all $x \in \mathbb{R}^p$,

$$x^\top(\beta_{*1,1} + \beta_{*2,2} - \beta_{*1,2} - \beta_{*2,1}) = \frac{1}{2} \left\{ \sum_{j=k} \mu_{*j,k}^\top \beta_{*j,k} - \sum_{j \neq k} \mu_{*j,k}^\top \beta_{*j,k} \right\}. \quad (5)$$

When the predictors are uncorrelated, it is easy to construct examples where (5) holds.

Example 1. Under the conditions of Remark 1, suppose $p = 2$ and $\Omega_* \propto I_2$. If $\mu_{*j,k} = (g_{*1}(j), g_{*2}(k))^\top$ for any bijective functions $g_{*m} : [c_m] \rightarrow \mathbb{R}$, $m \in [2]$, then (5) holds.

We display data generated under Example 1 in the leftmost panel of Figure 1. More generally, if $p \geq M$ is arbitrary, Ω_* is proportional to the identity, and each component

of $\mu_{*\mathbf{v}}$ takes a value determined by a single component of \mathbf{v} , (5) will hold. Conditional dependence between predictors can be allowed when some components of $\mu_{*\mathbf{v}}$ do not vary as a function of \mathbf{v} . However, if we instead assumed $\Omega_{*j,k} = \rho^{|j-k|}$ for $\rho > 0$ in Example 1, (5) will not hold in general. We display such a scenario in the middle panel of Figure 1, where the mean vectors are identical to the leftmost panel. Evidently, conditional dependence between components of the response can be induced both through the mean vectors $\mu_{*\mathbf{v}}$ (e.g., see the rightmost panel of Figure 1) and/or through the precision matrix Ω_* .

Examining the parametric form of $\log \text{OR}(x)$ reveals an insightful fact about how the predictor affects the association between Y_1 and Y_2 .

Proposition 1. *Suppose $M = 2$, $c_1 = c_2 = 2$ and $\pi_{*\mathbf{v}} = 1/4$ for each $\mathbf{v} \in \mathcal{C}$. If $x^\top(\beta_{*1,1} + \beta_{*2,2} - \beta_{*1,2} - \beta_{*2,1}) = 0$ for all x , then $\text{OR}(x) = c$ for some $c \in \mathbb{R}$ for all $x \in \mathbb{R}^p$. Restated, if $\beta_{*1,1} + \beta_{*2,2} = \beta_{*1,2} + \beta_{*2,1}$, then the predictor can only affect the marginal distributions of the response $(Y_m \mid X)$ as the odds ratio is constant, depending upon Ω_* and the $\mu_{*\mathbf{v}}$ alone.*

The proof of Proposition 1 and all subsequent results can be found in the Supplementary Material. If $\beta_{*1,1} + \beta_{*2,2} = \beta_{*1,2} + \beta_{*2,1}$ under the conditions of Proposition 1, then x could only affect the marginal (conditional) probabilities $\Pr(Y_1 = v_1 \mid X = x)$ and $\Pr(Y_2 = v_2 \mid X = x)$. This follows from the fact that the joint distribution $(Y_1, Y_2 \mid X)$ is uniquely defined by its marginal distributions and odds ratio (Agresti, 2012).

Remark 2. *The result of Proposition 1 could also be applied componentwise on x . That is, if $[\beta_{*1,1} + \beta_{*2,2}]_j = [\beta_{*1,2} + \beta_{*2,1}]_j$, then the j th component of x , the j th predictor, does not affect the odds ratios, but can still affect the marginal distributions $(Y_m \mid X)$. For the j th predictor to be entirely irrelevant, it cannot affect odds ratios or marginal distributions. Examining (3) and (4), we can see that this would be the case only if $[\beta_{*\mathbf{v}}]_k = c$ for some $c \in \mathbb{R}$ for all $\mathbf{v} \in \mathcal{C}$.*

In a later section, we will propose a new method for estimating the discriminant vectors

under (2) such that we will be able to identify predictors which are irrelevant for classification.

2.3 Considerations for maximum likelihood estimation

In standard applications of the linear discriminant analysis model—when there is only a single (univariate) categorical response—one can expect to observe realizations of X conditional on each response category. As such, one can straightforwardly use standard maximum likelihood estimators for the unknown mean vectors and covariance. Similarly, marginal probabilities for the response categories can be estimated in a straightforward way. For the setting where $p > n$, in which case the maximum likelihood estimator of Ω_* does not exist, there are many methods for regularized estimation of the parameters from (2) (Rothman et al., 2008; Witten and Tibshirani, 2009; Xu et al., 2015; Price et al., 2015; Molstad and Rothman, 2018), and the discriminant vectors (Cai and Liu, 2011; Mai et al., 2012, 2019).

In the multivariate categorical response context, however, estimation of the parameter from (2) becomes more challenging. For example, in order to estimate the mean vectors from (2) using maximum likelihood, one must observe at least one realization of X from each of the $\prod_{m=1}^M c_m$ many response categories. This requires a sample size of at least $\prod_{m=1}^M c_m$, though in practice, a much larger sample size is needed to observe predictors from every response category combination with reasonably high probability. Moreover, estimating the marginal probabilities $\pi_{*\mathbf{v}}$ is challenging: when n is small relative to $\prod_{m=1}^M c_m$, this is essentially the problem of estimating probabilities from a sparse contingency table (Agresti, 1992).

One approach for handling this problem is to simply treat response category combinations not appearing in the sample data as occurring with probability zero. That is, for every $\mathbf{v} \in \mathcal{C}$ not observed in our training data, we would assume $\pi_{*\mathbf{v}} = 0$, which makes estimating $\mu_{*\mathbf{v}}$ unnecessary for the task of classification. Of course, if we know certain category combinations occur with nonzero probability, this may be inappropriate. Moreover, the ability to interpret the parameters of (2) is one of the primary reasons for employing the model (2).

Instead, we propose a new approach for estimating all parameters from (2) for the purpose of classification. Loosely speaking, our method exploits the assumption that if (v_1, \dots, v_M) is similar to (v'_1, \dots, v'_M) (e.g., many $v_k = v'_k$), then μ_{*v_1, \dots, v_M} will be similar to $\mu_{*v'_1, \dots, v'_M}$. In the case that p is large, as in our real data application, we will also consider two (distinct) assumptions which will reduce the number of parameters to be estimated. The first is that many components of the mean vectors $\mu_{*\mathbf{v}} \in \mathbb{R}^p$ do not vary across the response category combinations. The second, which is not necessarily implied by the first, is that many of the p variables are irrelevant for classification.

2.4 Mean estimation with discrete kernelized regression

For the remainder of this section, let $g_* : \mathcal{C} \rightarrow \mathbb{R}^p$ denote the function $g_*(\mathbf{v}) := \mu_{\mathbf{v}}$ for each $\mathbf{v} \in \mathcal{C}$. Accordingly, (2) can be equivalently characterized $(X \mid Y_1 = v_1, \dots, Y_M = v_M) \sim N_p\{g_*(v_1, \dots, v_M), \Sigma_*\}$ for each $(v_1, \dots, v_M) \in \mathcal{C}$. That is, g_* is a function whose domain is \mathcal{C} and codomain is \mathbb{R}^p . Define the components of g_* at \mathbf{v} as $g_*(\mathbf{v}) = (g_{*1}(\mathbf{v}), \dots, g_{*p}(\mathbf{v}))^\top \in \mathbb{R}^p$ where $g_{*\ell} : \mathcal{C} \rightarrow \mathbb{R}$ for $\ell \in [p]$.

To exploit the notion that similar combinations of response categories correspond to similar mean vectors, we use a variation of kernelized regression. We assume there exists a transformation $\phi : \mathcal{C} \rightarrow \mathbb{R}^d$ ($d \geq 1$) such that $\|\phi(\mathbf{v}) - \phi(\mathbf{v}')\|_2$ small implies $\|g_*(\mathbf{v}) - g_*(\mathbf{v}')\|_2$ is small, loosely speaking. This requires the existence of some transformation from the space of the response, \mathcal{C} , to \mathbb{R}^d such that if two response combinations \mathbf{v} and \mathbf{v}' are close in the transformed space, their corresponding mean vectors are close in \mathbb{R}^p . Such transformations ϕ are called “feature maps” in nonparametric regression (Schölkopf and Smola, 2002).

At a high level, we first apply the transformation ϕ to the collection of observed responses, then quantify the similarity between two any response category combinations via the Euclidean inner product in \mathbb{R}^d . Specifically, let $k : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}$ be a symmetric positive-semidefinite kernel function such that for any collection of n responses $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^n$, where

$\mathbf{y}_i = (y_{i1}, \dots, y_{iM}) \in \mathcal{C}$, the $n \times n$ matrix with (i, j) th entry $k(\mathbf{y}_i, \mathbf{y}_j) = \langle \phi(\mathbf{y}_i), \phi(\mathbf{y}_j) \rangle$ is positive semidefinite. Loosely, $k(\mathbf{y}_i, \mathbf{y}_j)$ will be large if \mathbf{y}_i and \mathbf{y}_j are similar, and vice versa.

Formally, we propose to approximate the ℓ th component of the mean function, $g_{*\ell}$, with a function g_ℓ belonging to the hypothesis space of functions

$$g_\ell(\cdot) = \eta_\ell + \tilde{g}_\ell(\cdot), \quad \eta_\ell \in \mathbb{R}, \quad \tilde{g}_\ell(\cdot) \in \text{span} \{k(\cdot, \mathbf{y}_i) : i \in [n]\}, \quad \ell \in [p]. \quad (6)$$

That is, $g_\ell(\cdot)$ is the set of functions that can be decomposed into a constant plus a function depending on the input element of \mathcal{C} . To see how such a function g_ℓ satisfies our heuristic, notice that by definition of the hypothesis space, we can see that every $g_\ell(\cdot) = \eta_\ell + \sum_{i=1}^n a_{(\ell)i} k(\cdot, \mathbf{y}_i)$ for some $a_{(\ell)} \in \mathbb{R}^n$. Therefore,

$$|g_\ell(\mathbf{v}) - g_\ell(\mathbf{v}')| = \left| \sum_{i=1}^n a_{(\ell)i} \{k(\mathbf{v}, \mathbf{y}_i) - k(\mathbf{v}', \mathbf{y}_i)\} \right| \leq \|\phi(\mathbf{v}) - \phi(\mathbf{v}')\|_2 \left\| \sum_{i=1}^n a_{(\ell)i} \phi(\mathbf{y}_i) \right\|_2$$

so that for a given $a_{(\ell)}$, $\|\phi(\mathbf{v}) - \phi(\mathbf{v}')\|_2$ small roughly implies $|g_\ell(\mathbf{v}) - g_\ell(\mathbf{v}')|$ is small.

We can also justify our hypothesis space of functions (6) more formally. Given positive semidefinite kernel function k with domain $\mathcal{C} \times \mathcal{C}$, we may define a reproducing kernel Hilbert space of functions, \mathcal{H} , where for all $\mathbf{v} \in \mathcal{C}$, (i) $k(\cdot, \mathbf{v}) \in \mathcal{H}$ and (ii) for all $f \in \mathcal{H}$, $\langle f, k(\cdot, \mathbf{v}) \rangle_{\mathcal{H}} = f(\mathbf{v})$ (Wainwright, 2019, Chapters 12-13). Suppose, momentarily, Ω_* were known. It is then natural to consider the (nonparametric) maximum likelihood estimator of g_* given by $\arg \min_{h \in \mathcal{H}} \sum_{i=1}^n \{x_i - h(\mathbf{y}_i)\}^\top \Omega_* \{x_i - h(\mathbf{y}_i)\}$. By arguments similar to the generalized representer theorem (Schölkopf et al., 2001, Theorem 1; see discussion of relaxing strict monotonicity of regularizing function), we can show that one minimizer with respect to h is given by a function of the form $\hat{h}_\ell(\cdot) = \sum_{i=1}^n a_{(\ell)i} k(\cdot, \mathbf{y}_i)$ for some $a_{(\ell)} \in \mathbb{R}^n$ for each $\ell \in [p]$. Thus, it is natural to focus our attention to the space of functions (6), the set of all functions having the same linear representation as \hat{h} , a maximum likelihood estimator.

For discrete kernel regression, there are many transformations $\phi : \mathcal{C} \rightarrow \mathbb{R}^d$, and corresponding kernel functions k , that could be employed. These include the weighted Hamming

distance kernel, the Eskin measure kernel, and a kernel based on Goodall similarity, to name a few. The weighted Hamming distance kernel is given by $k^H(\mathbf{y}_i, \mathbf{y}_j) = \frac{1}{w} \sum_{m=1}^M w_m \mathbf{1}(y_{im} = y_{jm})$, where $w_m \geq 0$ are user specified weights (e.g., $w_m = \sqrt{c_m}$) and $w = \sum_{m=1}^M w_m$. This kernel computes the (weighted) number of agreements between its two inputs. The Eskin measure kernel is defined as $k^E(\mathbf{y}_i, \mathbf{y}_j) = \frac{1}{M} \sum_{m=1}^M \left\{ \frac{c_m^2}{c_m^2 + 2} + \frac{2}{c_m^2 + 2} \mathbf{1}(y_{im} = y_{jm}) \right\}$. The kernel k^E more harshly penalizes mismatches that occur in response components with a smaller number of categories. Finally, a variant of Goodall's measure $k^G(\mathbf{y}_i, \mathbf{y}_j) = \frac{1}{M} \sum_{m=1}^M \left\{ \frac{d_{im}(d_{im}-1)}{n(n-1)} \mathbf{1}(y_{im} = y_{jm}) \right\}$ where $d_{im} = \sum_{v=1}^{c_m} f_{mv} \mathbf{1}(y_{im} = v)$ and $f_{mv} = \sum_{i=1}^n \mathbf{1}(y_{im} = v)$ is the frequency of category v in m th response. The kernel k^G assigns higher similarity if the matches are frequent. Note that this is a special case of the weighted Hamming distance kernel. Many other kernel functions exist for quantifying the similarity between two elements of \mathcal{C} : see Boriah et al. (2008) for an overview.

Before we formally describe how we will estimate g_* , we note that the span of $\{k(\cdot, \mathbf{y}_i), i \in [n]\}$, the space from which we will estimate \tilde{g} , is determined by the set of unique responses \mathbf{y}_i . For example, if $\mathbf{y}_n = \mathbf{y}_{n-1}$, then $\text{span}\{k(\cdot, \mathbf{y}_i) : i \in [n]\} = \text{span}\{k(\cdot, \mathbf{y}_i) : i \in [n-1]\}$. Consequently, we define $\tilde{\mathbf{Y}} = \{\tilde{\mathbf{y}}_i\}_{i=1}^{\tilde{n}}$ as the set of distinct response category combinations observed in $\{\mathbf{y}_i\}_{i=1}^n$ (i.e., $\tilde{n} \leq n$ and $\tilde{\mathbf{Y}} \subseteq \mathbf{Y}$) where $\tilde{\mathbf{y}}_i \neq \tilde{\mathbf{y}}_j$ for all $i \neq j$. We then define $k_{\tilde{\mathbf{Y}}}(\cdot) : \mathcal{C} \rightarrow \mathbb{R}^{\tilde{n}}$ as $k_{\tilde{\mathbf{Y}}}(\cdot) = (k(\cdot, \tilde{\mathbf{y}}_1), \dots, k(\cdot, \tilde{\mathbf{y}}_{\tilde{n}}))^{\top}$, define $K_{\tilde{\mathbf{Y}}} \in \mathbb{R}^{n \times \tilde{n}}$ as the matrix with (i, j) th entry $k(\mathbf{y}_i, \tilde{\mathbf{y}}_j)$, and define $K_{\tilde{\mathbf{Y}}}^{\dagger} \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$ as the matrix with (i, j) th entry $k(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j)$. It is easy to see that $\text{span}\{k(\cdot, \tilde{\mathbf{y}}_i) : i \in [\tilde{n}]\} = \text{span}\{k(\cdot, \mathbf{y}_i) : i \in [n]\}$. The implication of this fact is that any $\tilde{g}(\cdot) = (\tilde{g}_1(\cdot), \dots, \tilde{g}_p(\cdot))^{\top}$ has be represented as $\alpha^{\top} k_{\tilde{\mathbf{Y}}}(\cdot)$ where $\alpha \in \mathbb{R}^{\tilde{n} \times p}$. Thus, in contrast to applications of the representer theorem with predictors drawn from a continuous distribution (wherein the coefficient dimension is n), we need only estimate \tilde{n} coefficients per component of \tilde{g} .

Thus, for a given set of n independent observations $\{(\mathbf{y}_1, x_1), \dots, (\mathbf{y}_n, x_n)\}$, we approximate the function $g_*(\cdot)$ with a function of the form $\eta + \alpha^{\top} k_{\tilde{\mathbf{Y}}}(\cdot)$. To fit the model (2), $\eta \in \mathbb{R}^p$

and $\alpha \in \mathbb{R}^{\tilde{n} \times p}$ will be estimated using penalized maximum likelihood.

To establish some of the results in this article, we will often require an assumption about the user-chosen kernel function k . Let $\varphi_{\min}(A)$ be the smallest singular value of a matrix A .

Assumption 1. The kernel function k is chosen so that for any $\tilde{\mathbf{Y}}$, there exists a constant $c_0 > 0$ such that $\varphi_{\min}(K_{\tilde{\mathbf{Y}}}^\dagger) \geq c_0 > 0$.

Assumption 1 would be satisfied by a kernel function k' for any sample \mathbf{Y} , if, for example, we defined $k'(\mathbf{y}_i, \mathbf{y}_j) = k(\mathbf{y}_i, \mathbf{y}_j) + c_0 \mathbf{1}(\mathbf{y}_i = \mathbf{y}_j)$ with k being any of the three example kernels. This amounts to upweighting an exact match between the two arguments of the kernel function by positive constant c_0 .

3 Maximum likelihood estimation

3.1 Regularized estimation of g_*

In order to estimate the coefficients (η, α) and the precision matrix Ω_* , we propose to minimize the negative log-likelihood. Specifically, the negative log-likelihood of (2) with $g_*(\cdot)$ approximated by $\eta + \alpha^\top k_{\tilde{\mathbf{Y}}}(\cdot)$, ignoring constants, is given by

$$\frac{1}{n} \sum_{i=1}^n \{x_i - \eta - \alpha^\top k_{\tilde{\mathbf{Y}}}(\mathbf{y}_i)\}^\top \Omega \{x_i - \eta - \alpha^\top k_{\tilde{\mathbf{Y}}}(\mathbf{y}_i)\} - \log \det(\Omega), \quad (7)$$

where \det is the determinant operator. In low-dimensional settings, minimizing (7) with respect to α, η , and Ω may work well. Specifically, we have the following result.

Proposition 2. Define $g_{\text{MLE}}(\cdot) = \eta_{\text{MLE}} + \alpha_{\text{MLE}}^\top k_{\tilde{\mathbf{Y}}}(\cdot)$ where η_{MLE} and α_{MLE} are minimizers of (7) with respect to η and α . Let $\bar{x} = n^{-1} \sum_{i=1}^n x_i$. For each $\{\tilde{\mathbf{y}}_i\}_{i=1}^{\tilde{n}}$, define $\bar{x}_{\tilde{\mathbf{y}}_i} = \frac{\sum_{k=1}^n \mathbf{1}\{\mathbf{y}_k = \tilde{\mathbf{y}}_i\} x_k}{\sum_{k=1}^n \mathbf{1}\{\mathbf{y}_k = \tilde{\mathbf{y}}_i\}}$ as the sample mean for the observed response category combination $\tilde{\mathbf{y}}_i \in \mathcal{C}$, and define $\bar{x}_{\tilde{\mathbf{y}}_i}^0 = \frac{\sum_{k=1}^n \mathbf{1}\{\mathbf{y}_k = \tilde{\mathbf{y}}_i\} (x_k - \bar{x})}{\sum_{k=1}^n \mathbf{1}\{\mathbf{y}_k = \tilde{\mathbf{y}}_i\}}$. If Assumption 1 holds, then

$$g_{\text{MLE}}(\mathbf{v}) = \begin{cases} \bar{x}_{\mathbf{v}} & : \text{if } \mathbf{y}_i = \mathbf{v} \text{ for any } i \in [n] \\ \sum_{i=1}^{\tilde{n}} w_i(\mathbf{v}) \bar{x}_{\mathbf{y}_i}^0 + \bar{x} & : \text{otherwise} \end{cases},$$

where $w(\mathbf{v}) = (w_1(\mathbf{v}), \dots, w_{\tilde{n}}(\mathbf{v}))^\top = K_{\tilde{\mathbf{Y}}}^{0\dagger -1} k_{\tilde{\mathbf{Y}}}(\mathbf{v})$ and $K_{\tilde{\mathbf{Y}}}^{0\dagger} \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$ is a matrix with (i, j) th entry $k(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j) - n^{-1} \sum_{i'=1}^n k(\mathbf{y}_{i'}, \tilde{\mathbf{y}}_j)$. The minimizer with respect to Ω , if it exists, is $(n^{-1} \sum_{i=1}^n \{x_i - g_{\text{MLE}}(\mathbf{y}_i)\} \{x_i - g_{\text{MLE}}(\mathbf{y}_i)\}^\top)^{-1}$.

Proposition 2 establishes that when we minimize (7) with respect to η and α , our estimate of g_* , g_{MLE} , is equivalent to the conditional sample mean for all response category combinations $\mathbf{v} \in \mathcal{C}$ that are observed in the training data. For category combinations that are not observed, g_{MLE} is a weighted sum of the overall sample mean and the conditional sample means for observed response category combinations. The weights are determined by the choice of kernel function k and the collection of observed responses $\{\mathbf{y}_i\}_{i=1}^n$. Thus, if p and c^* were fixed, and the $\pi_{*\mathbf{v}}$ are bounded away from zero, as $n \rightarrow \infty$, our method will perform identically to standard maximum likelihood.

However, in finite samples when p and c^* are large relative to n , this may be problematic. First, the $\bar{x}_{\mathbf{v}}$ may be computed from a small number of observations, and relatedly, it is well known that with p diverging quickly relative to n , linear discriminant analysis will eventually perform no better than random guessing due to noise accumulation in the mean estimates (Fan and Fan, 2008; Elman et al., 2020). Second, when $p \geq n$, the maximum likelihood estimator of Ω will not exist because $n^{-1} \sum_{i=1}^n \{x_i - g_{\text{MLE}}(\mathbf{y}_i)\} \{x_i - g_{\text{MLE}}(\mathbf{y}_i)\}^\top$ will not be invertible. Even when $p < n$, this matrix will be singular when c^* is large relative to n .

Consequently, we need an alternative approach for estimating the coefficients in (7). As mentioned, in this article we consider two schemes for shrinkage estimation which exploit different assumptions about the model (2). The first assumption we consider is that many components of $\mu_{*\mathbf{v}}$ do not vary across all $\mathbf{v} \in \mathcal{C}$. That is, there are many components j such that $[\mu_{*\mathbf{v}}]_j = \eta_j \in \mathbb{R}$ for all $\mathbf{v} \in \mathcal{C}$ where $[a]_j$ denotes the j th component of a vector a . To

encourage fitted models with this property, we need a way in which to estimate g such that many of the $\tilde{g}_\ell = 0$ for all inputs.

Notice that for any fixed α , $n^{-1} \sum_{i=1}^n x_i - n^{-1} \sum_{i=1}^n \alpha^\top k_{\tilde{\mathbf{Y}}}(\mathbf{y}_i)$ minimizes the negative log-likelihood with respect to η . Thus, we need only focus on minimizing

$$\mathcal{L}(\alpha, \Omega) = \frac{1}{n} \text{tr}\{(X^0 - K_{\tilde{\mathbf{Y}}}^0 \alpha) \Omega (X^0 - K_{\tilde{\mathbf{Y}}}^0 \alpha)^\top\} - \log \det(\Omega), \quad (8)$$

where $X^0 = (x_1 - n^{-1} \sum_{i=1}^n x_i, \dots, x_n - n^{-1} \sum_{i=1}^n x_i)^\top \in \mathbb{R}^{n \times p}$ and $K_{\tilde{\mathbf{Y}}}^0 \in \mathbb{R}^{n \times \tilde{n}}$ is a matrix with (j, k) th entry $k(\mathbf{y}_j, \tilde{\mathbf{y}}_k) - n^{-1} \sum_{i=1}^n k(\mathbf{y}_i, \tilde{\mathbf{y}}_k)$.

If, for example, the j th column of α is entirely zero, then we are ensured $[\alpha^\top k_{\tilde{\mathbf{Y}}}(\mathbf{v})]_j = 0$ for all $\mathbf{v} \in \mathcal{C}$, which would imply that $[\hat{g}(\mathbf{v})]_j$ is constant. Thus, to encourage estimates of g such that $\tilde{g}_\ell(\mathbf{v}) = 0$ for all \mathbf{v} , we apply a group lasso penalty on the columns α . Specifically, we propose to estimate the pair (α, Ω) with $(\hat{\alpha}, \hat{\Omega})$, defined as

$$\arg \min_{\alpha \in \mathbb{R}^{\tilde{n} \times p}, \Omega \in \mathbb{S}_+^p} \{\mathcal{L}(\alpha, \Omega) + \lambda \|\alpha\|_{1,2} + \frac{\gamma}{2} \|\Omega\|_F^2\}, \quad (9)$$

where $\|\alpha\|_{1,2} := \sum_{j=1}^p (\sum_{i=1}^{\tilde{n}} \alpha_{i,j}^2)^{1/2}$, $\|\Omega\|_F := \{\text{tr}(\Omega^\top \Omega)\}^{1/2}$, and $(\lambda, \gamma) \in (0, \infty) \times (0, \infty)$ are user-specified tuning parameters. The ridge penalty on Ω serves to shrink the sum of squared elements of Ω and ensures that with α fixed, a minimizer with respect to Ω exists. Though the optimization problem in (9) is nonconvex, it is biconvex. That is, with α held fixed, the optimization is convex with respect to Ω and vice versa.

Our choice of ridge penalty on Ω is primarily for computational convenience. With α fixed, the minimizer of (9) with respect to Ω has a closed form. One could instead penalize the sum of the absolute values of the off-diagonals of Ω if it is reasonable to assume that Ω_* is sparse (Rothman et al., 2008). This approach, however, would in general require an iterative algorithm to minimize (9) with respect to Ω .

With $(\hat{\alpha}, \hat{\Omega})$ (and consequently, $\hat{\eta}$) in hand, we classify a subject with predictors x_{new} into response category set given by $\arg \max_{\mathbf{v} \in \mathcal{C}} [\{\hat{\eta} + \hat{\alpha}^\top k_{\tilde{\mathbf{Y}}}(\mathbf{v})\}^\top \hat{\Omega} \{2x_{\text{new}} - \hat{\eta} - \hat{\alpha}^\top k_{\tilde{\mathbf{Y}}}(\mathbf{v})\} + 2 \log \hat{\pi}_{\mathbf{v}}]$.

where $\hat{\pi}_{\mathbf{v}}$ is an estimate of $\pi_{*\mathbf{v}}$, which we discuss in Section 5.

3.2 Convexifying reparameterization for direct variable selection

The estimator in (9) is motivated by the assumption that many elements of the mean vectors $\mu_{*\mathbf{v}}$ do not differ across all $\mathbf{v} \in \mathcal{C}$. While this affords interpretability in terms of how the parameters from (2) depend on \mathbf{v} , this does lead to variable selection (i.e., removal of irrelevant variables) without additional constraints on Ω_* . Recall that $\beta_{*\mathbf{v}} := \Omega_* \mu_{*\mathbf{v}}$ is the discriminant vector for category combination $\mathbf{v} \in \mathcal{C}$. As mentioned in Remark 2, for the j th variable to be irrelevant for distinguishing between all combinations of response categories, it must be that $[\beta_{*\mathbf{v}}]_j = \nu_j \in \mathbb{R}$ for all $\mathbf{v} \in \mathcal{C}$. Intuitively, it is insufficient that $[\mu_{*\mathbf{v}}]_j = [\mu_{*\mathbf{v}'}]_j$ for all \mathbf{v}, \mathbf{v}' because if the j th variable is conditionally correlated with a variable whose means are unequal, the j th variable will affect the decision rule (Xu et al., 2015).

When we approximate $g_*(\cdot)$ with $\eta + \alpha^\top k_{\tilde{\mathbf{Y}}}(\cdot)$, we approximate $\beta_{*\mathbf{v}} = \Omega_* g_*(\mathbf{v})$ with $\Omega_* \{ \eta + \alpha^\top k_{\tilde{\mathbf{Y}}}(\mathbf{v}) \}$ for some $\eta \in \mathbb{R}^p$ and $\alpha \in \mathbb{R}^{\tilde{n} \times p}$. Therefore, if we want to estimate Ω_* and (η, α) so that our fitted model can be interpreted directly in terms of which variables are irrelevant for discriminating between response categories, we need to encourage fitted models such that many elements of $\hat{\Omega} \hat{\alpha}^\top k_{\tilde{\mathbf{Y}}}(\mathbf{v})$ will be zero for all $\mathbf{v} \in \mathcal{C}$ for estimates $(\hat{\Omega}, \hat{\alpha})$. If the j th row of $\hat{\Omega} \hat{\alpha}^\top$ were entirely zero, then $[\hat{\Omega} \hat{g}(\mathbf{v})]_j = [\hat{\Omega} \hat{\eta}]_j = c \in \mathbb{R}$ for all \mathbf{v} , i.e., the j th predictor has no effect on the estimated decision rule. Therefore, if we let $\Theta := \alpha \Omega$, imposing sparsity on the columns of Θ would correspond to componentwise equality cross the discriminant vectors. Under this parameterization, we can write the negative log-likelihood as

$$\mathcal{L}_c(\Theta, \Omega) = \frac{1}{n} \text{tr} \{ (X^0 \Omega - K_{\tilde{\mathbf{Y}}}^0 \Theta) \Omega^{-1} (X^0 \Omega - K_{\tilde{\mathbf{Y}}}^0 \Theta)^\top \} - \log \det(\Omega), \quad (10)$$

where the subscript c denotes that this is the negative log-likelihood under the parameterization $\Theta = \alpha \Omega$. Analogous to (9), our estimator for direct variable selection is

$$\arg \min_{\Theta \in \mathbb{R}^{\tilde{n} \times p}, \Omega = \Omega^\top} \left\{ \mathcal{L}_c(\Theta, \Omega) + \lambda \|\Theta\|_{1,2} + \frac{\eta}{2} \|\Omega\|_F^2 \right\} \quad \text{subject to } \Omega \geq \epsilon I_p, \quad (11)$$

where $\epsilon > 0$ is a lower bound on the smallest eigenvalue of Ω_* . Remarkably, the optimization problem in (11) is jointly convex in (Θ, Ω) (Zhu, 2020, Theorem 1). This implies that (11) is also biconvex as both $\Theta \mapsto \mathcal{L}_c(\Theta, \Omega)$ and $\Omega \mapsto \mathcal{L}_c(\Theta, \Omega)$ are convex. Though this convexifying reparameterization has been studied in regression (Yu and Bien, 2019; Zhu, 2020), to the best of our knowledge, it has not been employed for variable selection in the linear discriminant analysis model.

In practice, we impose a lower bound on the smallest eigenvalue of Ω , ϵ , making the feasible set closed (as opposed to \mathbb{S}_+^p , which is open). This simplifies our computational algorithm for solving (11). Though ϵ is a tuning parameter, we find that simply setting ϵ equal to some reasonably small constant (e.g., $\epsilon = 10^{-4}$) seems to work well across a variety of settings.

With a solution to (11) in hand, say $(\check{\Theta}, \check{\Omega})$, we use classification rule

$$\arg \max_{\mathbf{v} \in \mathcal{C}} \left[\{\check{\eta}^\top \check{\Omega} + k_{\check{\mathbf{Y}}}(\mathbf{v})^\top \check{\Theta}\} \{2x_{\text{new}} - \check{\eta} + \check{\Omega}^{-1} \check{\Theta}^\top k_{\check{\mathbf{Y}}}(\mathbf{v})\} + 2 \log \hat{\pi}_{\mathbf{v}} \right], \quad (12)$$

where $\check{\eta} = n^{-1} \sum_{i=1}^n x_i - n^{-1} \sum_{i=1}^n k_{\check{\mathbf{Y}}}(\mathbf{y}_i) \check{\Theta} \check{\Omega}^{-1}$. Examining (12), it is immediate to see that if the j th column of $\check{\Theta}$ is entirely zero, then the j th component of x_{new} has no effect on the decision rule (since $\check{\eta}^\top \check{\Omega} x_{\text{new}}$ is constant with respect to \mathbf{v}).

We recommend selecting all tuning parameters using cross-validation.

3.3 Finite sample properties of kernelized mean estimation

In this section, we study the finite sample properties of our estimator of g_* based on kernelized regression in (9). Let $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^n$ be the observed responses, and let $\tilde{\mathbf{Y}} = \{\tilde{\mathbf{y}}_i\}_{i=1}^{\tilde{n}}$ be the set of distinct response category combinations. The following results will apply conditional on the set of observed responses \mathbf{Y} . In our context, the mean vectors and discriminant vectors are of primary interest. The precision matrix Ω_* plays a crucial role in estimation and classification, but mainly serves to bridge the mean vectors and the discriminant vectors. As

such, we assume Ω_* is known in order to focus our attention specifically on recovery of g_* . Without loss of generality, we also assume that the predictors are centralized in the sense that their componentwise marginal expectation is zero. In this setting, if for a particular $\ell \in [p]$, $g_{*\ell}$ is constant across all response category combinations, then $g_{*\ell}(\mathbf{v}) = 0$ for all $\mathbf{v} \in \mathcal{C}$. Hence, we consider estimating $g_*(\mathbf{v}) = E(X | Y_1, \dots, Y_M = \mathbf{v})$ with $\hat{g}(\mathbf{v}) = \sum_{j=1}^{\tilde{n}} \hat{\alpha}_j k(\mathbf{v}, \tilde{\mathbf{y}}_j)$ where we define $\hat{\alpha}$ as

$$\arg \min_{\alpha \in \mathbb{R}^{\tilde{n} \times p}} \left[\frac{1}{n} \sum_{i=1}^n \{x_i - \alpha^\top k_{\tilde{\mathbf{Y}}}(\mathbf{y}_i)\}^\top \Omega_* \{x_i - \alpha^\top k_{\tilde{\mathbf{Y}}}(\mathbf{y}_i)\} + \lambda \|\alpha\|_{1,2} \right]. \quad (13)$$

Notice that we can write the random predictor (conditional on \mathbf{y}_i) as $X_i = \Omega_*^{-1/2} Z_i + g_*(\mathbf{y}_i)$, where entries of $Z_i \in \mathbb{R}^p$ are independent standard normals for $i \in [n]$. Let $M_* \in \mathbb{R}^{n \times p}$ have i th row $g_*(\mathbf{y}_i)$. Because many rows of M_* will be duplicated if we observed $\mathbf{y}_i = \mathbf{y}_j$ for many $i \neq j$, it is convenient to define $M_*^\dagger \in \mathbb{R}^{\tilde{n} \times p}$ as the matrix that contains one mean vector corresponding to each element of $\tilde{\mathbf{Y}}$ and define $Q \in \mathbb{R}^{n \times \tilde{n}}$ as a matrix with (i, j) th entry $Q_{i,j} = \mathbf{1}(\mathbf{y}_i = \tilde{\mathbf{y}}_j)$, so that $M_* = QM_*^\dagger$.

Now, let us define $\alpha_* \in \arg \min_{\alpha \in \mathbb{R}^{\tilde{n} \times p}} \|M_* - K_{\tilde{\mathbf{Y}}} \alpha\|_F^2$. If Assumption 1 holds, then α_* is unique and given by $\alpha_* = (K_{\tilde{\mathbf{Y}}}^\top K_{\tilde{\mathbf{Y}}})^{-1} K_{\tilde{\mathbf{Y}}}^\top M_*$. Moreover, it can be easily verified that under Assumption 1, $\min_{\alpha \in \mathbb{R}^{\tilde{n} \times p}} \|M_* - K_{\tilde{\mathbf{Y}}} \alpha\|_F^2 = \|M_* - K_{\tilde{\mathbf{Y}}} \alpha_*\|_F^2 = 0$. For completeness, we provide a proof of this fact in the Supplementary Material. The equality $\|M_* - K_{\tilde{\mathbf{Y}}} \alpha_*\|_F^2 = 0$ implies the existence of an α such that $K_{\tilde{\mathbf{Y}}} \alpha_*$ perfectly recovers M_* . This suggests that we may treat α_* as an estimand, and $\hat{\alpha}$ as our estimator thereof.

Next, let us state our second assumption.

Assumption 2. There exists a constant c_1 such that $0 < c_1 \leq \varphi_{\min}(\Omega_*) \leq \varphi_{\max}(\Omega_*) \leq 1/c_1 < \infty$, where φ_{\max} and φ_{\min} denote the largest and smallest eigenvalue of their argument, respectively.

Recall from Section 2.4 that we assume few components of g_* differ as a function of its argument. In terms of M_* , this would imply that $M_{*,j} = 0$ for many $j \in [p]$. If $M_{*,j} = 0$,

this implies that $\alpha_{*,j} = 0$ (since M_* is the rightmost term in the product defining α_*). Hence, define $\mathcal{S} = \{j : \alpha_{*,j} \neq 0\}$, define $\mathcal{S}^c = [p] \setminus \mathcal{S}$, and let s be the cardinality of the set \mathcal{S} . Recall that $K_{\tilde{\mathbf{Y}}}^\dagger \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$ is the matrix with (i,j) th entry $k(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j)$ (i.e., $K_{\tilde{\mathbf{Y}}} = Q K_{\tilde{\mathbf{Y}}}^\dagger$). Define $\kappa_{\mathcal{S}} = \inf_{\Delta \in \mathbb{C}(\mathcal{S}), \|\Delta\|_F=1} \|K_{\tilde{\mathbf{Y}}}^\dagger \Delta \Omega_*^{1/2}\|_F^2 / \tilde{n}$ where $\mathbb{C}(\mathcal{S}) = \{\nu \in \mathbb{R}^{\tilde{n} \times p} : \|\nu_{\cdot, \mathcal{S}^c}\|_{1,2} \leq 3\|\nu_{\cdot, \mathcal{S}}\|_{1,2}\}$. Under Assumption 1 and 2, $\kappa_{\mathcal{S}} \geq c_0^2 c_1 / \tilde{n} > 0$, but for \mathcal{S} with small cardinality, $\kappa_{\mathcal{S}}$ may be larger. Notice that $\kappa_{\mathcal{S}}$ is effectively a restricted eigenvalue of the matrix $\tilde{n}^{-1}(\Omega_* \otimes K_{\tilde{\mathbf{Y}}}^\dagger K_{\tilde{\mathbf{Y}}}^\dagger)$ (Wainwright, 2019, Chapter 7.3.1), hence the stated lower bound.

We are now prepared to present our first result. For the remainder of the article, let $\|\cdot\|$ denote the spectral norm of matrix.

Theorem 1 (Average in-sample mean estimation error). *Suppose Assumption 1 and 2 hold. Let $c_2 > 1$ be a fixed constant and let $\omega_* = \max_{j \in [p]} \|\Omega_{*,j}\|_2$. If $\lambda = 4\omega_*(\|K_{\tilde{\mathbf{Y}}}\|_F + \|K_{\tilde{\mathbf{Y}}}\| \sqrt{2c_2 \log p})/n$, then with probability at least $1 - p^{1-c_2}$,*

$$\frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \|g_*(\tilde{\mathbf{y}}_i) - \hat{g}(\tilde{\mathbf{y}}_i)\|_2^2 \leq \frac{36w_*^2 s}{\varphi_{\min}(\Omega_*) \kappa_{\mathcal{S}}} \left(\frac{\tilde{n}_{\max}}{\tilde{n}_{\min}^2} \right) \left\{ \frac{\|K_{\tilde{\mathbf{Y}}}^\dagger\|_F}{\tilde{n}} + \frac{\|K_{\tilde{\mathbf{Y}}}^\dagger\|}{\tilde{n}} \sqrt{2c_2 \log p} \right\}^2,$$

where $\tilde{n}_{\max} := \max_{j \in [\tilde{n}]} \sum_{i=1}^n \mathbf{1}(\mathbf{y}_i = \tilde{\mathbf{y}}_j)$ and $\tilde{n}_{\min} := \min_{j \in [\tilde{n}]} \sum_{i=1}^n \mathbf{1}(\mathbf{y}_i = \tilde{\mathbf{y}}_j)$.

Theorem 1 demonstrates how well we can recover the mean vectors, on average, corresponding to the response category combinations observed in the training data. If \tilde{n}_{\min} is small relative to \tilde{n}_{\max} , the bound would be worse than, say, in the best-case scenario when $\tilde{n}_{\min} = \tilde{n}_{\max}$. Note that our result in Theorem 1 is distinct from standard results for kernel regression estimators. Our proof technique, which focuses on estimation error for the “optimal” coefficients α_* , allows us to account for sparsity in M_* in a direct way using the proof strategy from Negahban et al. (2012).

Next, to illustrate how judicious choice of kernel can improve estimation of means for all category combinations—including those not observed in the training data—we require a more general result.

Lemma 1 (Statistical versus approximation error). *For any $\mathbf{v} \in \mathcal{C}$, we have*

$$\|\hat{g}(\mathbf{v}) - g_*(\mathbf{v})\|_2 \leq \inf_{w \in \mathbb{R}^{\tilde{n}}} \{h_w^{g*}(\mathbf{v}) + h_w^\phi(\mathbf{v})\} + \|(\hat{\alpha} - \alpha_*)^\top k_{\tilde{\mathbf{Y}}}(\mathbf{v})\|_2,$$

where $h_w^{g*}(\mathbf{v}) := \|g_*(\mathbf{v}) - \sum_{i=1}^{\tilde{n}} w_i g_*(\tilde{\mathbf{y}}_i)\|_2$ and $h_w^\phi(\mathbf{v}) := \|\alpha_*^\top \{\sum_{i=1}^{\tilde{n}} w_i k_{\tilde{\mathbf{Y}}}(\tilde{\mathbf{y}}_i) - k_{\tilde{\mathbf{Y}}}(\mathbf{v})\}\|_2$. If $\mathbf{v} \in \tilde{\mathbf{Y}}$, then $\inf_{w \in \mathbb{R}^{\tilde{n}}} \{h_w^{g*}(\mathbf{v}) + h_w^\phi(\mathbf{v})\} = 0$.

The generic error bound from Lemma 1 can be decomposed into two parts: approximation error, $\inf_{w \in \mathbb{R}^{\tilde{n}}} \{h_w^{g*}(\cdot) + h_w^\phi(\cdot)\}$, and statistical error, $\|(\hat{\alpha} - \alpha_*)^\top k_{\tilde{\mathbf{Y}}}(\cdot)\|_F$. The approximation error can be further decomposed into two pieces, represented by h_w^{g*} and h_w^ϕ . The magnitude of $h_w^{g*}(\mathbf{v})$ quantifies how well we can approximate $g_*(\mathbf{v})$ with any linear combination of the $\{g_*(\tilde{\mathbf{y}}_i)\}_{i=1}^{\tilde{n}}$. If we observe a sufficiently large number of response category combinations in our training data, we could expect there to exist w such that this term is small. However, h_w^{g*} cannot be disentangled from h_w^ϕ . The term h_w^ϕ reflects the quality of our choice of kernel function k . In particular, we can write $h_w^\phi(\mathbf{v}) = \|\sum_{\ell=1}^{\tilde{n}} \alpha_\ell^* \langle \sum_{i=1}^{\tilde{n}} w_i \phi(\tilde{\mathbf{y}}_i) - \phi(\mathbf{v}), \phi(\tilde{\mathbf{y}}_\ell) \rangle\|_2$, which will be small if $\sum_{i=1}^{\tilde{n}} w_i \phi(\tilde{\mathbf{y}}_i) - \phi(\mathbf{v})$ is small. Ideally, we could select a kernel k (and consequently, ϕ) for which there exists a w such that $\|\sum_{i=1}^{\tilde{n}} w_i \phi(\tilde{\mathbf{y}}_i) - \phi(\mathbf{v})\|_2$ and $\|\sum_{i=1}^{\tilde{n}} w_i g_*(\tilde{\mathbf{y}}_i) - g_*(\mathbf{v})\|_2$ are both small. The optimal choice of kernel, then, is one in which both $\sum_{i=1}^{\tilde{n}} w_i \phi(\tilde{\mathbf{y}}_i) = \phi(\mathbf{v})$ and $\sum_{i=1}^{\tilde{n}} w_i g_*(\tilde{\mathbf{y}}_i) = g_*(\mathbf{v})$ for a single vector w for all $\mathbf{v} \in \mathcal{C}$.

Finally, we apply Lemma 1 to establish the following.

Theorem 2 (Out-of-sample mean estimation). *Suppose Assumption 1 and 2 hold. If $\lambda = 4\omega_*(\|K_{\tilde{\mathbf{Y}}}\|_F + \|K_{\tilde{\mathbf{Y}}}\|\sqrt{2c_2 \log p})/n$, then for any $\mathbf{v} \in \mathcal{C}$, with probability at least $1 - p^{1-c_2}$*

$$\begin{aligned} \|\hat{g}(\mathbf{v}) - g_*(\mathbf{v})\|_2 &\leq \|g_*(\mathbf{v}) - M_*^{\dagger \top} K_{\tilde{\mathbf{Y}}}^{-1} k_{\tilde{\mathbf{Y}}}(\mathbf{v})\|_2 + \\ &\quad \frac{6\|k_{\tilde{\mathbf{Y}}}(\mathbf{v})\|_2 w_* \sqrt{s}}{\kappa_S} \left(\frac{\sqrt{\tilde{n}_{\max}}}{\tilde{n}_{\min}} \right) \left\{ \frac{\|K_{\tilde{\mathbf{Y}}}^\dagger\|_F}{\tilde{n}} + \left(\frac{\|K_{\tilde{\mathbf{Y}}}^\dagger\|}{\tilde{n}} \right) \sqrt{2c_2 \log p} \right\}. \end{aligned}$$

The result of Theorem 2 follows from the fact that under Assumption 1, there always

exists a w such that $h_w^\phi(\mathbf{v}) = 0$. We plug this w into the expression for $h_w^{g*}(\mathbf{v})$, then apply a probabilistic bound for $\|\hat{\alpha} - \alpha_*\|_2$ to establish the right hand side of the inequality from Theorem 1. Proofs of all results can be found in the Supplementary Material.

4 Computation

4.1 Blockwise coordinate descent algorithm

To exploit the biconvexity of the objective function from (9), we use a blockwise coordinate descent algorithm. That is, we iteratively update α with Ω held fixed and vice versa. With α fixed at its (t) th iterate, $\alpha^{(t)}$, obtaining the (t) th iterate for Ω requires solving a ridge penalized normal precision matrix estimation problem, $\Omega^{(t)} = \arg \min_{\Omega \in \mathbb{S}_+^p} [\text{tr}\{S(\alpha)\Omega\} - \log \det(\Omega) + \frac{\eta}{2}\|\Omega\|_F^2]$, with $S(\alpha) = n^{-1}(X^0 - K_{\tilde{\mathbf{Y}}}^0\alpha)^\top(X^0 - K_{\tilde{\mathbf{Y}}}^0\alpha)$. It can be shown that $\Omega^{(t+1)} = \frac{1}{2\eta}V\{-D + (D^2 + 4\eta I_p)^{1/2}\}V^\top$ where $S(\alpha) = VDV^\top$ is the eigendecomposition of $S(\alpha)$ where $V \in \mathbb{R}^{p \times p}$ is orthogonal and $D \in \mathbb{R}^{p \times p}$ diagonal. See Witten and Tibshirani (2009) or Price et al. (2015) for a derivation.

With Ω fixed at $\Omega^{(t)}$, the $(t+1)$ th iterate for α is $\alpha^{(t+1)} \in \arg \min_{\alpha \in \mathbb{R}^{n \times p}} \{\mathcal{L}(\alpha, \Omega) + \gamma\|\alpha\|_{1,2}\}$. We use a variation of the proximal gradient descent algorithm to compute $\alpha^{(t+1)}$ (Beck and Teboulle, 2009; Polson et al., 2015). For this subalgorithm, we will use r as an iteration counter. Specifically, given step size $s > 0$ sufficiently small and (r) th iterate of α , $\alpha^{(r)}$, the $(r+1)$ th iterate of the proximal gradient descent subalgorithm is defined as

$$\alpha^{(r+1)} = \arg \min_{\alpha \in \mathbb{R}^{n \times p}} \left\{ \frac{1}{2}\|\alpha - \alpha^{(r)} + s\nabla_\alpha \mathcal{L}(\alpha^{(r)}, \Omega)\|_F^2 + s\gamma\|\alpha\|_{1,2} \right\} \quad (14)$$

where $\nabla_\alpha \mathcal{L}(\alpha, \Omega) = -\frac{2}{n}\{K_{\tilde{\mathbf{Y}}}^{0\top} X^0 \Omega - K_{\tilde{\mathbf{Y}}}^{0\top} K_{\tilde{\mathbf{Y}}}^0 \alpha \Omega\}$. One can use subgradient calculus to show that (14) can be solved column-by-column in closed form. Namely,

$$\alpha_{:,j}^{(r+1)} = \max \left(1 - \frac{s\gamma}{\|\alpha_{:,j}^{(r)} - s[\nabla_\alpha \mathcal{L}(\alpha^{(r)}, \Omega)]_{:,j}\|_2}, 0 \right) \left(\alpha_{:,j}^{(r)} - s[\nabla_\alpha \mathcal{L}(\alpha^{(r)}, \Omega)]_{:,j} \right) \quad (15)$$

where $\alpha_{:,j}^{(r)}$ is the j th column of $\alpha^{(r)}$ and $[\nabla_\alpha \mathcal{L}(\alpha^{(r)}, \Omega)]_{:,j}$ is the j th column of $\nabla_\alpha \mathcal{L}(\alpha^{(r)}, \Omega)$

(Yuan and Lin, 2006; Simon et al., 2013). We repeat (15) for $r = 1, 2, 3, \dots$ until the objective function value converges.

In our implementation, we use an accelerated variation of this algorithm (Parikh et al., 2014, Chapter 4.3). Briefly, the accelerated version replaces search point $\alpha^{(r)} - s\nabla_\alpha \mathcal{L}(\alpha^{(r)}, \Omega)$ with $\alpha^{(r,r-1)} - s\nabla_\alpha \mathcal{L}(\alpha^{(r,r-1)}, \Omega)$ where $\alpha^{(r,r-1)} = \alpha^{(r)} + \frac{r-1}{r+2}(\alpha^{(r)} - \alpha^{(r-1)})$. It is well known that if s is fixed and chosen sufficiently small—or if s is chosen by backtracking line search—the objective function value converges at a quadratic rate (Beck and Teboulle, 2009; Parikh et al., 2014). We provide an outline of the algorithm to solve (9), Algorithm 1 in the Supplementary Material.

4.2 Modifications for convex estimator

To solve (11), we use a blockwise coordinate descent scheme similar to that described in the previous section. With Ω fixed at (t) th iterate $\Omega^{(t)}$, the update for Θ is $\Theta^{(t+1)} \in \arg \min_{\Theta \in \mathbb{R}^{\tilde{n} \times p}} \{\mathcal{L}_c(\Theta, \Omega) + \gamma \|\Theta\|_{1,2}\}$. We use the same accelerated proximal gradient descent scheme as in the α update: we only replace $\nabla_\alpha \mathcal{L}$ with $\nabla_\Theta \mathcal{L}_c$. With Θ fixed at its (t) th iterate, $\Theta^{(t)}$, to compute the (t) th iterate of Ω , we need to solve

$$\arg \min_{\Omega: \Omega = \Omega^\top, \Omega \geq \epsilon I_p} \left[\frac{1}{n} \text{tr}\{(X^0 \Omega - K_{\tilde{Y}}^0 \Theta) \Omega^{-1} (X^0 \Omega - K_{\tilde{Y}}^0 \Theta)^\top\} - \log \det(\Omega) + \frac{\eta}{2} \|\Omega\|_F^2 \right]. \quad (16)$$

Unlike the update in the nonconvex case, (16) cannot be solved in closed form. Instead, we use a projected gradient descent algorithm to solve (16). Letting $\mathcal{L}_c^\eta(\Theta, \Omega) := \mathcal{L}_c(\Theta, \Omega) + \frac{\eta}{2} \|\Omega\|_F^2$ and using that $\nabla_\Omega \mathcal{L}_c^\eta(\Theta, \Omega) = n^{-1} \{X^{0\top} X^0 - \Omega^{-1} \Theta^\top K_{\tilde{Y}}^{0\top} K_{\tilde{Y}}^0 \Theta \Omega^{-1}\} - \Omega^{-1} + \gamma \Omega$, the $(r+1)$ th iterate of the projected gradient descent subalgorithm is defined as $\Omega^{(r+1)} = \arg \min_{\Omega \geq \epsilon I_p} \|\Omega - \Omega^{(r)} + s \nabla_\Omega \mathcal{L}_c^\eta(\Theta, \Omega^{(r)})\|_F^2 = U D_{\epsilon+} U^\top$, where $D_{\epsilon+} = \text{diag}[\{\max(d_j, \epsilon)\}_{j=1}^p]$, UDU^\top is the eigendecomposition of $\Omega^{(r)} - s \nabla_\Omega \mathcal{L}_c^\eta(\Theta, \Omega^{(r)})$ and d_j is the (j, j) th element of D . That is, $\Omega^{(r+1)}$ is simply the search point $\Omega^{(r)} - s \nabla_\Omega \mathcal{L}_c^\eta(\Theta, \Omega^{(r)})$ projected onto the closed convex set $\{\Omega : \Omega = \Omega^\top, \Omega \geq \epsilon I_p\}$. The step size $s > 0$ is chosen via backtracking line search.

The complete algorithm we use for computing (11) can be found in the Supplementary Material Algorithm 2.

5 Estimation of marginal response probability tensor

In the case that n is small relative to c^* , sample estimates of the marginal probabilities $\pi_{*\mathbf{v}}$ will be unreliable. For example, when $c^* > n$, at least $c^* - n$ maximum likelihood estimates of the $\pi_{*\mathbf{v}}$ will be zero. Of course, $\pi_{*v_1, \dots, v_M} = 0$ implies $\Pr(Y_1 = v_1, \dots, Y_M = v_M | X) = 0$. When we know there exists some small positive constant κ such that $\Pr(Y_1 = v_1, \dots, Y_M = v_M | X = x) \geq \kappa$ for all $(v_1, \dots, v_M) \in \mathcal{C}$ (e.g., based on prior knowledge of the application), this is problematic because using (3) with $\hat{\pi}_{v_1, \dots, v_M} = 0$ forces $\widehat{\Pr}(Y_1 = v_1, \dots, Y_M = v_M | X = x) = 0$ for all $x \in \mathbb{R}^p$. Hence, we need a more reliable way to estimate the $\pi_{*\mathbf{v}}$ when n is small relative to c^* .

Let $\Pi_* \in \mathbb{R}^{c_1} \times \dots \times \mathbb{R}^{c_M}$ denote the M -way probability tensor with (v_1, \dots, v_M) th element $[\Pi_*]_{v_1, \dots, v_M} = \pi_{*v_1 \dots v_M}$, for $(v_1, \dots, v_M) \in \mathcal{C}$, such that $\pi_{*v_1 \dots v_M} \geq 0$ and $\sum_{v_1, \dots, v_M} \pi_{*v_1 \dots v_M} = 1$. To resolve the aforementioned issues in the small n , large c^* setting, we approximate the tensor Π_* using a low-rank tensor decomposition (Kolda, 2001). Note that this approximation is necessary to provide a reliable estimate, but low-rankness is not an essential part of our model assumption. In simulations, we do not impose such low-rank assumption in the data generating process, yet we observe that the estimation approach described in the following works well for approximating the marginal probabilities. Let $\Delta^{(r-1)} := \{v \in \mathbb{R}^r : 1_r^\top v = 1, v_k \geq 0, k \in [r]\}$. We assume Π_* can be decomposed as

$$\Pi_* = \sum_{k=1}^R \delta_k \Psi_k, \quad \Psi_k = \psi_k^{(1)} \circ \psi_k^{(2)} \circ \dots \circ \psi_k^{(M)} \quad (17)$$

where each $\psi_k^{(\ell)} \in \Delta^{c_\ell-1}$, $\delta = (\delta_1, \dots, \delta_R)^\top \in \Delta^{(M-1)}$ with $\delta_1 \geq \delta_2 \geq \dots \geq \delta_R > 0$, and \circ denotes the outer product such that $\Pi_{*v_1, \dots, v_M} = \sum_{k=1}^R \delta_k \prod_{\ell=1}^M \psi_{kv_\ell}^{(\ell)}$. It is easy to check that the constraints on $\psi_k^{(\ell)}$ and δ make Π_* a valid probability tensor (Dunson and Xing, 2009;

Johndrow et al., 2017; Molstad and Zhang, 2022). It can be also be shown that (17) always holds for a finite $R \leq \prod_{m=1}^M c_m / \max_{k \in [M]} c_k$. When the dependence structure of (Y_1, \dots, Y_M) is constrained, this can imply a reduction in the upper bound on R such that (17) holds. For example, if the M responses form $L \leq M$ mutually independent groups indexed by $\{G_k\}_{k=1}^L$ where $G_1 \cup \dots \cup G_L = [M]$, $G_k \subseteq [M]$, and $G_k \cap G_{k'} = \emptyset$ for $k \neq k'$, then Π_* can be decomposed as (17) with $R \leq \prod_{l=1}^L (\prod_{m \in G_l} c_m / \max_{k \in G_l} c_k)$ (Molstad and Zhang, 2022). Thus, low-rank probability tensors can approximate otherwise complex dependencies among categorical response variables. The decomposition of a probability tensor as in (17) is related to classical latent structure analysis (Anderson, 1954; Goodman, 1974).

Based on the decomposition in (17), in order to estimate Π_* , we need to estimate the unknown parameters $\theta = \{\delta, \psi_1^{(1)}, \psi_2^{(1)}, \dots, \psi_R^{(M)}\} \in \Delta^{(M-1)} \times \Delta^{(c_1-1)} \times \dots \times \Delta^{(c_M-1)}$. The observed data log-likelihood evaluated at θ is $\sum_{i=1}^n \log[\sum_{k=1}^R \delta_k \{\prod_{\ell=1}^M \prod_{j=1}^{c_\ell} (\psi_{kj}^{(\ell)})^{\mathbf{1}(y_{i\ell}=j)}\}]$. We use the expectation-maximization (EM) algorithm to maximize the observed data log-likelihood with respect to θ . To do so, we introduce a latent variable Z , which is assumed to follow a categorical distribution with R categories and marginal probability vector $\delta \in \Delta^{R-1}$ (Johndrow et al., 2017, Remark 3.1). That is, $\Pr(Z = r) = \delta_r$ and $\psi_r^{(k)}$ is the vector of probabilities corresponding to $(Y_k | Z = r)$ for $k \in [M]$. Intuitively, this model assumes that conditioned on (latent) variable Z , the Y_m are independent. Since we cannot observe Z , the marginal dependence among Y_1, \dots, Y_M is induced through their dependence on Z . The complete EM algorithm we use is given in the Supplementary Material.

In practice, we treat R as a tuning parameter to be selected alongside (λ, η) using cross-validation or an information criterion.

6 Simulation studies

6.1 Data generating models and competing method

In this section, we illustrate the performance of our method through simulation studies. We compare our estimators to competitors under a variety of data generating models. For 100 independent replications under each setting, we first generate $n = 200$ independent responses (Y_1, \dots, Y_M) . To do so, we generate a c^* -variate vector which has independent Uniform(0,1) components, say $u \in (0, 1) \times \dots \times (0, 1)$, then divide by its sum so that $\pi_* = u / \sum_{j=1}^{c^*} u_j$ belongs to the $(c^* - 1)$ -dimensional probability simplex. Then, we generate realizations of (Y_1, \dots, Y_M) , denoted \mathbf{y} , from the categorical distribution with probabilities π_* . Thus the components of the response are marginally dependent with arbitrary dependencies.

Given \mathbf{y} , we then generate x from the p -dimensional multivariate normal distribution (2). In each scenario, we set $\Omega_{*s,t}^{-1} = 0.7^{|s-t|}$ for all $(s, t) \in [p] \times [p]$. We consider two different models (Models A and B) for the mean vectors determined by g_* . Specifically, the two models differ in terms of how the mean vectors from (2) depend on the response categories. For both models, we vary p and a parameter controlling the difficulty of classification.

- **Model A.** We randomly select 10 distinct elements of $[p]$, say $\{k_1, \dots, k_{10}\}$ and set

$$g_{*k}(\mathbf{y}) = \begin{cases} b_\ell^\top \mathbf{y} / \nu & : k = k_\ell \text{ for some } \ell \in \{1, \dots, 10\} \\ 0 & : \text{otherwise} \end{cases}, \quad k \in [p]$$

where $b_\ell \in \{0, 2\}^M$ with the collection $\{b_\ell\}_{\ell=1}^{10}$ having $10M/2$ components equal to two and $10M/2$ equal to zero in randomly chosen positions.

Note that here (and in Model B), we use the numeric form $\mathbf{y} \in [c_1] \times \dots \times [c_M]$ so that $b_\ell^\top \mathbf{y} \in \mathbb{R}$.

Model A is ideal for the nonconvex estimator: only ten elements of the $\mu_{*\mathbf{v}}$ differ as a function of \mathbf{v} , which the nonconvex estimator is designed to exploit. The convex estimator, on the other hand, exploits sparsity in the collection of discriminant vectors $\Omega_*(\mu_{*\mathbf{v}} - \mu_{*\mathbf{v}'})$

for $\mathbf{v} \neq \mathbf{v}'$. Under Model A, $\Omega_*(\mu_{*\mathbf{v}} - \mu_{*\mathbf{v}'})$ can have as many as 30 nonzero elements because Ω_* is tridiagonal. Model B, in contrast, imposes sparsity on the discriminant vectors directly.

- **Model B.** We randomly select 10 distinct elements of $[p]$, say $\{k_1, \dots, k_{10}\}$ and set

$$[\beta_{*\mathbf{y}}]_k = \begin{cases} b_\ell^\top \mathbf{y} / \nu & : k = k_\ell \text{ for some } \ell \in \{1, \dots, 10\} \\ 0 & : \text{otherwise} \end{cases}, \quad k \in [p]$$

where $b_\ell \in \{0, 2\}^M$ with the collection $\{b_\ell\}_{\ell=1}^{10}$ having $10M/2$ components equal to two and $10M/2$ equal to zero in randomly chosen positions. Then, we set $g_*(\mathbf{v}) = \Omega_*^{-1} \beta_{*\mathbf{v}}$ so that $\beta_{*\mathbf{v}} - \beta_{*\mathbf{v}'} = \Omega_*(\mu_{*\mathbf{v}} - \mu_{*\mathbf{v}'})$.

Under Model B, all components of g_* can differ as a function of the response categories. However, by construction, only ten variables are relevant for classification. Model B is thus ideal for (11).

We consider two versions of each model. In **Model A-4** and **Model B-4**, we set $M = 4$ and $c_1 = \dots = c_4 = 3$; in **Model A-6** and **Model B-6**, we set $M = 6$ and $c_1 = \dots = c_6 = 2$. Throughout our simulations, we will consider $p \in \{50, 100, 150\}$ and $\nu \in \{0.8, 1.0, 1.2, 1.4\}$. Under both models, ν controls the difficulty of the classification problem. If ν is small, differences between category combination means are large, so the problem is easier.

We compare our two estimators: (9) (KLDA-M) and (11) (KLDA-D)—both using Hamming distance kernel—to competitors that either fit separate models for each response, or formulate a synthetic (univariate) categorical response and fit a singular model (i.e., the aggregate model). The first competitor is the separate multinomial logistic regression estimator (**S-Logistic**), which fits a separate group-lasso penalized multinomial logistic regression model for each response. We also consider an aggregate version, **A-Logistic**, which fits a single group-lasso penalized multinomial logistic regression model to the synthetic $c^* = \prod_{\ell=1}^M c_m$ -category response. Note that if a category combination is not observed in the

training data, this method sets its conditional probability to zero. We also consider separate multiclass sparse discriminant analysis models (Mai et al., 2019, **S-MSDA**), and an aggregate sparse discriminant analysis model **A-MSDA**. The latter correctly specifies the LDA model from which we generate data. Finally, we also compared to **oracle**, which uses the true parameters from (2) plugged into Bayes' classification rule. This serves as an upper bound for the performance of any estimation method, but is not available in practice.

The first performance metric we considered is the mean estimation error. Since none of the competitors are capable of estimating the means corresponding different categories, we compare the mean estimation error of the sample mean (i.e., the MLE) with the mean estimates obtained using our methods. Specifically we display mean estimation error, defined as $\frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \|g_*(\tilde{\mathbf{y}}_i) - \bar{x}_{\tilde{\mathbf{y}}_i}\|_2^2$ for the MLE, and $\frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \|g_*(\tilde{\mathbf{y}}_i) - \hat{g}(\tilde{\mathbf{y}}_i)\|_2^2$ for the **KLDA** variants, where $\bar{x}_{\tilde{\mathbf{y}}_i} = \frac{1}{\sum_{i=1}^n \mathbf{1}(\mathbf{y}_i = \tilde{\mathbf{y}}_i)} \sum_{i=1}^n x_i \mathbf{1}(\mathbf{y}_i = \tilde{\mathbf{y}}_i)$ is the sample mean corresponding to $\tilde{\mathbf{y}}_i$. The other two performance metrics we considered are prediction accuracy and Hamming distance, which are defined as $\frac{1}{n} \sum_{i=1}^n \mathbf{1}(\mathbf{y}_i = \hat{\mathbf{y}}_i)$, and $\frac{1}{nM} \sum_{i=1}^n \sum_{\ell=1}^M \mathbf{1}(\mathbf{y}_{i\ell} = \hat{\mathbf{y}}_{i\ell})$, respectively.

6.2 Results

In Table 1, we present the MEE results with varying parameters p , ν and different models. Notably, our proposed estimators result in considerably smaller errors in mean estimation compared to the sample mean estimates across all considered scenarios. The nonconvex estimator **KLDA-M**, which directly estimates the mean vectors, does not invariably outperform the convex estimator **KLDA-D**, which approximates the discriminant vectors directly. Under Model A, the mean vectors are sparse, and **KLDA-M** is able to exploit this, whereas **KLDA-D** is not. Conversely, under Model B, it is the discriminant vectors that are sparse, whereas the mean vectors are nonsparse. Naturally, this favors **KLDA-D** in the mean estimation.

In Figure 2 and 3, we display the prediction accuracy on testing set across various models. Note that we omit the results of **C-Logistic** under Model A-4 and Model B-4 since it

p	ν	Model A-4			Model A-6			Model B-4			Model B-6		
		MLE	KLDA-M	KLDA-D									
50	0.8	24.475	1.149	2.176	21.072	1.022	1.747	24.475	2.945	2.174	21.072	2.636	1.864
	1.0	24.346	0.945	2.262	20.737	0.787	1.703	24.346	2.815	2.044	20.737	2.153	1.759
	1.2	24.361	0.854	2.314	20.950	0.716	1.731	24.361	2.833	2.130	20.950	2.161	1.920
	1.4	24.364	0.813	2.359	20.970	0.657	1.741	24.364	2.626	2.131	20.970	2.192	1.851
100	0.8	48.267	1.435	4.301	41.685	1.299	3.247	48.267	5.228	3.646	41.685	4.607	3.456
	1.0	48.955	1.255	4.460	42.290	1.028	3.252	48.955	4.944	3.541	42.290	4.259	3.358
	1.2	48.551	1.179	4.557	42.178	0.993	3.282	48.551	5.073	3.486	42.178	4.101	3.379
	1.4	48.670	1.088	4.498	41.878	0.983	3.156	48.670	4.828	3.316	41.878	4.299	3.165
150	0.8	72.536	1.657	6.310	62.511	1.519	4.637	72.536	7.714	5.119	62.511	6.011	4.777
	1.0	72.256	1.537	6.575	62.976	1.378	4.616	72.256	6.811	4.855	62.976	5.726	4.465
	1.2	72.193	1.465	6.498	63.267	1.309	4.547	72.193	6.726	4.639	63.267	5.658	4.302
	1.4	72.222	1.425	6.266	63.715	1.226	4.330	72.222	6.974	4.563	63.715	5.707	4.057

Table 1: Average mean estimation errors for the MLE versus KLDA averaged over 100 independent replications under Model A and Model B.

performs so poorly compared to others. These figures clearly demonstrate our proposed estimators, **KLDA-M** and **KLDA-D**, have superior prediction accuracy relative to the competitors, particularly noticeable with a larger number of responses as in Model A-6 and Model B-6.

For Model A-4, **KLDA-M** and **KLDA-D** have comparable performance when p is 50 or 100. However, when p is increased to 150, **KLDA-M** clearly outperforms **KLDA-D**. This coheres with expectations, given that **KLDA-M** leverages the sparsity of the mean vectors. In contrast, the discriminant vectors, onto which **KLDA-D** imposes sparsity, are nonsparse in this scenario, so the regularization scheme of **KLDA-D** may impose unhelpful bias. Consequently, **S-MSDA** mirrors the performance of **KLDA-D** under these conditions. Under Model B, **KLDA-D** consistently outperforms **KLDA-M**. This can be attributed to the fact that the mean vectors, which **KLDA-M** regularizes, are nonsparse. As expected, however, **KLDA-D** outperforms the other competitors as it exploits the sparsity of the discriminant vectors.

In the Supplementary Materials, we also include Hamming distance and variable selection results under all simulation settings. To summarize briefly, in terms of Hamming distance, we observe similar general trends as in Figure 2 and 3, except that **S-MSDA** can, at times, outperform our proposed methods. This can be understood from the fact that Hamming distance is inherently measuring quality of estimation of marginal probabilities, whereas our proposal is focused on estimation of the joint probability mass of $(Y_1, \dots, Y_M \mid X)$.

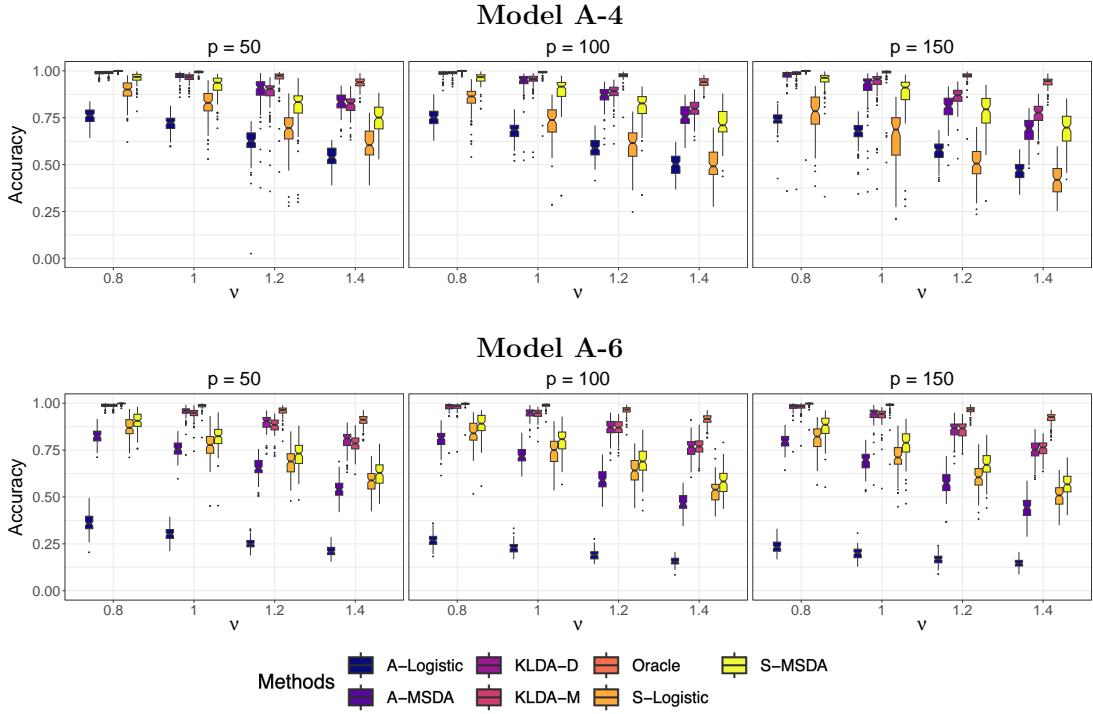


Figure 2: Prediction accuracy over 100 independent replications under **Model A-4** and **Model A-6** with $(p, \nu) \in \{50, 100, 150\} \times \{0.8, 1.0, 1.2, 1.4\}$.

7 Classification of colon tissue samples

In this section, we demonstrate the application of our method on a dataset consisting of gene expression profiles from colon biopsies (Noble et al., 2008). In particular, this dataset contains 44290 gene expression levels from 202 tissue samples. There are three labels for each sample: patient state (normal/ulcerative colitis), tissue state (inflamed/uninflamed) and anatomical locations (sigmoid colon/terminal ileum/descending colon/ascending colon). The dataset can be download from the Gene Expression Omnibus (GDS3268).

Following our simulation studies, we analyze the data using the proposed **KLDA-M** and **KLDA-D**, in conjunction with **S-Logistic**, **S-MSDA**, **A-Logistic** and **A-MSDA**. We partitioned the data by randomly selecting n samples for the training set, allocating 50 samples for the

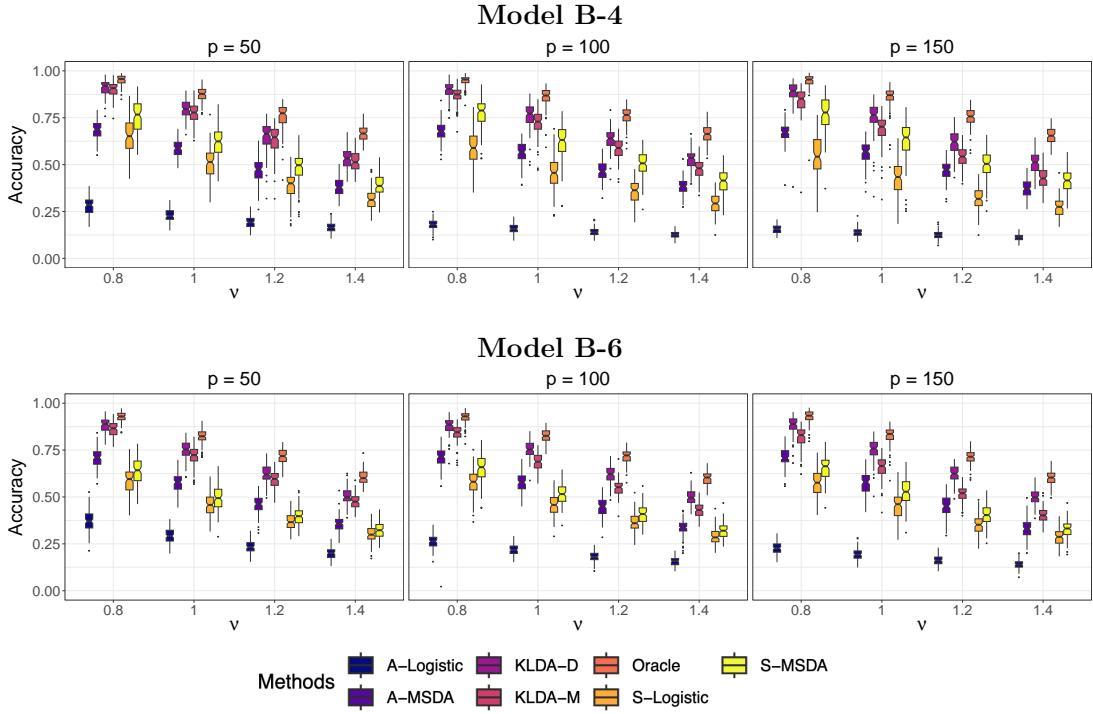


Figure 3: Prediction accuracy over 100 independent replications under **Model B-4** and **Model B-6** with $(p, \nu) \in \{50, 100, 150\} \times \{0.8, 1.0, 1.2, 1.4\}$.

validation set, and designating the remaining $152 - n$ samples for the testing set. To mitigate computational demands, we undertook a screening process on the genes. This involved ranking gene expression levels based on their median absolute deviation and subsequently selecting the top p genes. To avoid issues of collinearity, genes exhibiting high correlation were pruned. For our analysis, we consider $p \in \{100, 200, 300, 400, 500\}$ and $n \in \{50, 100\}$.

The results based on 100 replicates are listed in Table 2. It can be seen that our proposed methods achieve the highest accuracy under all choices of n and p , and KLDA-M outperforms KLDA-D in all but one setting. This underscores the potential advantage of regularizing mean vectors as opposed to discriminant vectors for this particular problem.

Next, we turn our attention to the mean estimation utilizing KLDA-M. For this analysis, we set $p = 200$, allocate 152 samples for training, and select tuning parameters with the

n	p	KLDA-M	KLDA-D	S-Logistic	S-MSDA	A-Logistic	A-MSDA
50	100	0.163	0.153	0.148	0.145	0.149	0.157
	200	0.164	0.163	0.152	0.148	0.161	0.152
	300	0.165	0.158	0.153	0.147	0.157	0.144
	400	0.168	0.167	0.147	0.146	0.158	0.146
	500	0.172	0.178	0.151	0.150	0.164	0.153
100	100	0.192	0.191	0.177	0.160	0.187	0.177
	200	0.188	0.180	0.169	0.164	0.181	0.174
	300	0.191	0.184	0.173	0.174	0.190	0.171
	400	0.202	0.186	0.162	0.165	0.187	0.175
	500	0.206	0.200	0.170	0.174	0.191	0.182

Table 2: Prediction accuracy on GDS3268 dataset over 100 independent replications with $p \in \{100, 200, 300, 400, 500\}$ and $n \in \{50, 100\}$. When $n = 50$, standard errors were never larger than 0.005; when $n = 100$, standard errors were never larger than 0.006.

remaining 50 samples. The results are shown in Figure 4, where we only include 30 genes to save space. Our fitted model estimated 111 genes’ means varied as a function of the response category combinations. The three-digit numbers along the rows denote distinct combinations of response labels. The first digit represents the patient state: 0 for normal and 1 for ulcerative colitis. The second digit represents the tissue’s state, with 0 indicating inflamed and 1 denoting uninflamed. The final digit represents anatomical locations: 0 for the ascending colon, 1 for the descending colon, 2 for the sigmoid colon, and 3 for the terminal ileum. It is important to note that both the estimated and sample mean vectors have been centralized; we’ve subtracted the global mean from each. Note that the combinations of response categories 000 (normal patient, inflamed tissue, ascending colon) and 003 (normal patient, inflamed tissue, terminal ileum) are not observed in the dataset, so we don’t have corresponding sample mean estimates.

Our estimates reveal a more distinct pattern of gene expression levels across various response category combinations compared to the sample mean estimates. Notably, genes such as IGFBP2, L1TD1, and UNG2 exhibit lower expression levels in normal patients and heightened levels in those diagnosed with ulcerative colitis. Conversely, genes like LGALS2,

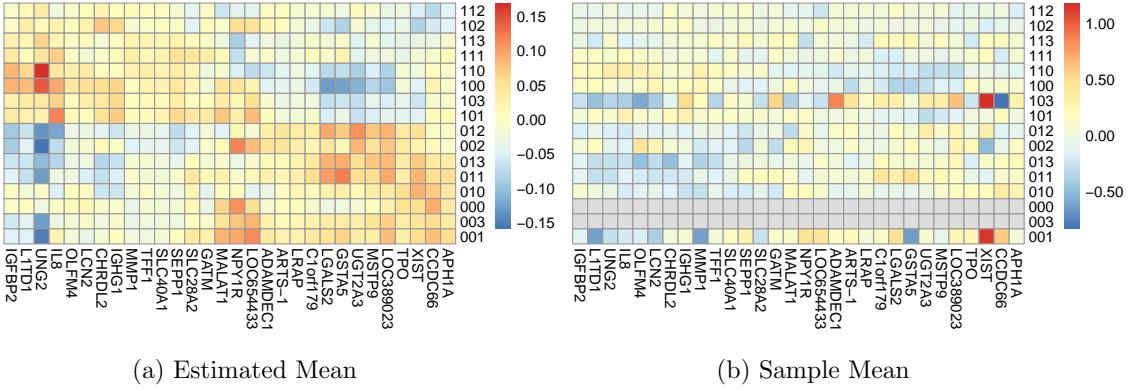


Figure 4: Mean estimates (minus the columnwise global average) using KLDA-M and the MLE with $p = 200$. Each column corresponds to a gene and we include 30 genes. Each row corresponds to a combination of response categories.

GSTA5, and UGT2A3 demonstrate an inverse pattern. Regarding UNG2, within both groups—normal patients and those diagnosed with ulcerative colitis—it is observed that the ascending colon has a higher expression level relative to other anatomical locations.

References

- Agresti, A. (1992). A survey of exact inference for contingency tables. *Statistical science*, 7(1):131–153.
- Agresti, A. (2012). *Categorical data analysis*, volume 792. John Wiley & Sons.
- Anderson, T. W. (1954). On estimation of parameters in latent structure analysis. *Psychometrika*, 19(1):1–10.
- Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202.
- Boriah, S., Chandola, V., and Kumar, V. (2008). Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the 2008 SIAM international conference on data mining*, pages 243–254. SIAM.
- Cai, T. and Liu, W. (2011). A direct estimation approach to sparse linear discriminant analysis. *Journal of the American statistical association*, 106(496):1566–1577.
- Dunson, D. B. and Xing, C. (2009). Nonparametric bayes modeling of multivariate categorical data. *Journal of the American Statistical Association*, 104(487):1042–1051.

- Ekholm, A., McDonald, J. W., and Smith, P. W. (2000). Association models for a multivariate binary response. *Biometrics*, 56(3):712–718.
- Elman, M. R., Minnier, J., Chang, X., and Choi, D. (2020). Noise accumulation in high dimensional classification and total signal index. *The Journal of Machine Learning Research*, 21(1):1383–1405.
- Fan, J. and Fan, Y. (2008). High dimensional classification using features annealed independence rules. *Annals of statistics*, 36(6):2605.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.
- Glonek, G. F. (1996). A class of regression models for multivariate categorical responses. *Biometrika*, 83(1):15–28.
- Glonek, G. F. and McCullagh, P. (1995). Multivariate logistic models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(3):533–546.
- Goodman, L. A. (1974). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*, 61(2):215–231.
- Hand, D. J. (2006). Classifier technology and the illusion of progress. *Statistical science*, 21(1):1–14.
- Hastie, T. and Tibshirani, R. (1996). Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):155–176.
- Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.
- Johndrow, J. E., Bhattacharya, A., and Dunson, D. B. (2017). Tensor decompositions and sparse log-linear models. *Annals of statistics*, 45(1):1.
- Kolda, T. G. (2001). Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 23(1):243–255.
- Lang, J. B. (1996). Maximum likelihood methods for a generalized class of log-linear models. *The Annals of Statistics*, 24(2):726–752.
- Lupparelli, M. and Roverato, A. (2017). Log-mean linear regression models for binary responses with an application to multimorbidity. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 66(2):227–252.
- Mai, Q., Yang, Y., and Zou, H. (2019). Multiclass sparse discriminant analysis. *Statistica Sinica*, 29(1):97–111.

- Mai, Q., Zou, H., and Yuan, M. (2012). A direct approach to sparse discriminant analysis in ultra-high dimensions. *Biometrika*, 99(1):29–42.
- Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (1994). Machine learning, neural and statistical classification.
- Molenberghs, G. and Lesaffre, E. (1999). Marginal modelling of multivariate categorical data. *Statistics in Medicine*, 18(17-18):2237–2255.
- Molstad, A. J. and Rothman, A. J. (2018). Shrinking characteristics of precision matrix estimators. *Biometrika*, 105(3):563–574.
- Molstad, A. J. and Rothman, A. J. (2023). A likelihood-based approach for multivariate categorical response regression in high dimensions. *Journal of the American Statistical Association*, 118(542):1402–1414.
- Molstad, A. J. and Zhang, X. (2022). Conditional probability tensor decompositions for multivariate categorical response regression.
- Negahban, S. N., Ravikumar, P., Wainwright, M. J., and Yu, B. (2012). A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557.
- Noble, C. L., Abbas, A. R., Cornelius, J., Lees, C. W., Ho, G.-T., Toy, K., Modrusan, Z., Pal, N., Zhong, F., Chalasani, S., et al. (2008). Regional variation in gene expression in the healthy colon is dysregulated in ulcerative colitis. *Gut*, 57(10):1398–1405.
- Parikh, N., Boyd, S., et al. (2014). Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239.
- Park, C. H. and Lee, M. (2008). On applying linear discriminant analysis for multi-labeled problems. *Pattern recognition letters*, 29(7):878–887.
- Polson, N., Scott, J. G., and Willard, B. T. (2015). Proximal algorithms in statistics and machine learning. *Statistical science*, 30(4):559–581.
- Price, B. S., Geyer, C. J., and Rothman, A. J. (2015). Ridge fusion in statistical learning. *Journal of Computational and Graphical Statistics*, 24(2):439–454.
- Rao, C. R. (1948). The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203.
- Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2009). Classifier chains for multi-label classification. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part II* 20, pages 254–269. Springer.

- Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2021). Classifier chains: A review and perspectives. *Journal of Artificial Intelligence Research*, 70:683–718.
- Rothman, A. J., Bickel, P. J., Levina, E., and Zhu, J. (2008). Sparse permutation invariant covariance estimation.
- Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). A generalized representer theorem. In *Computational Learning Theory: 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001 Amsterdam, The Netherlands, July 16–19, 2001 Proceedings 14*, pages 416–426. Springer.
- Schölkopf, B. and Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.
- Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *Journal of computational and graphical statistics*, 22(2):231–245.
- Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press.
- Wang, H., Ding, C., and Huang, H. (2010). Multi-label linear discriminant analysis. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part VI 11*, pages 126–139. Springer.
- Witten, D. M. and Tibshirani, R. (2009). Covariance-regularized regression and classification for high dimensional problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):615–636.
- Xu, P., Zhu, J., Zhu, L., and Li, Y. (2015). Covariance-enhanced discriminant analysis. *Biometrika*, 102(1):33–45.
- Yu, G. and Bien, J. (2019). Estimating the error variance in a high-dimensional linear model. *Biometrika*, 106(3):533–546.
- Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1):49–67.
- Zhang, M.-L. and Zhou, Z.-H. (2013). A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837.
- Zhu, Y. (2020). A convex optimization formulation for multivariate regression. *Advances in Neural Information Processing Systems*, 33:17652–17661.

Luca Scrucca

Material list:

Scrucca L. and Karlis D. (2024) A model-based approach to shot charts estimation in basketball. arXiv:2405.17265.

Scrucca L. (2019) A transformation-based approach to Gaussian mixture density estimation for bounded data. Biometrical Journal, 61:4, 873–888.

A Model-Based Approach to Shot Charts Estimation in Basketball

Luca Scrucca 

Department of Economics, Università degli Studi di Perugia

Dimitris Karlis 

Department of Statistics, Athens University of Economics

May 3, 2024

Abstract

Shot charts in basketball analytics provide an indispensable tool for evaluating players' shooting performance by visually representing the distribution of field goal attempts across different court locations. However, conventional methods often overlook the bounded nature of the basketball court, leading to inaccurate representations, particularly along the boundaries and corners. In this paper, we propose a novel model-based approach to shot chart estimation and visualization that explicitly considers the physical boundaries of the basketball court. By employing Gaussian mixtures for bounded data, our methodology allows to obtain more accurate estimation of shot density distributions for both made and missed shots. Bayes' rule is then applied to derive estimates for the probability of successful shooting from any given locations, and to identify the regions with the highest expected scores. To illustrate the efficacy of our proposal, we apply it to data from the 2022-23 NBA regular season, showing its usefulness through detailed analyses of shot patterns for two prominent players.

Keywords: Shot charts; visualization of shooting patterns; density estimation; transformation-based Gaussian mixtures for bounded data; probability of successful shooting; expected points scored.

Contents

1	Introduction	2
2	Methods	4
2.1	Model specification	5
2.2	Estimation and model selection	6
3	Applications	7
3.1	Stephen Curry	7
3.2	Joel Embiid	8
4	Conclusions	10

1 Introduction

Basketball is among the most popular sports game worldwide. It not only enjoys widespread popularity as a sport but has also generated substantial economic benefits through its associated industries. The National Basketball Association (NBA) is widely recognized as the world's leading league, attracting international interest with tremendous amounts spent in related marketing. In Europe, the Euroleague represents the pinnacle of professional men's club basketball competition and is regarded as the top-tier men's league on the continent. The increasing interest on basketball has led quite early to the development of advanced statistical methodologies for measuring performance (Kubatko et al., 2007), while several other proposal have been made after this. For a broad picture of academic and non-academic research on basketball analytics we recommend the book of Zuccolotto and Manisera (2020) and the broad review paper by Terner and Franks (2021).

One of the basic characteristics of basketball is that it is a fast-paced contact game in which the players are constantly moving in heated confrontations, thus leading to quick transitions from defense to offense or vice versa. In practice basketball is a game of space (see e.g. Goldsberry, 2012, p1). The teams that makes better use of spatial aspects can have an advantage and hence several tactics related to better enhancement of spatio-temporal game aspects (Sandholtz et al., 2020).

Advancements in sports information systems and technology has allowed the collection of a number of detailed spatio-temporal data that capture various aspects of basketball Papalexakis and Pelechrinis (2018); Shortridge et al. (2014). Such data can help considerably to understand the game and the effects of space on that while they also provide interesting information for all stakeholders of the game, including trainers, team managers, players, scouters of new players, spectators and journalists. Visualizations of basketball games can provide important information about the game (Perin et al., 2018). An increasing number of visualization research has been conducted that includes as visual analysis of player trajectories, visualization of field goals of a player, and visualization of basic statistics of different players in different games (Chen et al., 2016).

Shots are a key-ingredient of the sport. The final score of a team is defined by the number of successful shots and their quality, (2 or 3 points plus the 1 point for free throws). As such considerable interest has been made on understanding and predicting shot tactics and success. For example, Zuccolotto et al. (2018) utilized several techniques to model scoring probability under high-pressure conditions in basketball based play-by-play data from the Italian "Serie A2" Championship 2015/2016. Shortridge et al. (2014) discussed and proposed different measures about shot efficiency that take into account the spatial effect and they also proposed visualizations related to shot efficiency. Oughali et al. (2019) tried to predict shot success based on several machine learning approaches. Fichman and O'Brien (2019) discussed the optimal shot selection strategy for a basketball team. Jiao et al. (2021) proposed a marked spatial point process for modeling basketball shots based on the observation that the success rate of a basketball shots may be higher at locations where a player makes more shots. Related to the spatial aspect are also the so-called corner 3's, which are those shots that while producing 3 points are taken closer to the basket, thus allowing for larger probability of success and distinguished tactic for that shots (Pelechrinis and Goldsberry, 2021).

Visualizing shots can be a powerful tool for better understanding the different tactics. Quite early it has been noted that spatial visualizations like shot charts can be very valuable to reveal the tactical performance of the teams and hence be a valuable tool in the hands of trainers (Reich et al., 2006). For example, shot charts, that is, maps capturing locations of (made or missed) shots, and spatio-temporal trajectories for the players on the court can capture information about the offensive and defensive tendencies, as well as, schemes used by a team. Characterization of these processes is important for player and team comparisons, scouting, game preparation etc. Since then there has been extensive literature related to shots in basketball including effective

visualizations that can produce insights. Since shots are the most important aspect as it leads to gaining points, it is quite common to produce statistics related to shot success but also to shot patterns, including spatio-temporal aspects of shots. The radical choice of most teams towards different shooting styles that include more 3-points is perhaps partially due to the improved visualizations available.

Shot charts in basketball analytics are a fundamental tool for visually examining the distribution of players' field goal attempts and their efficiency in different court locations. Typically such charts visualize the locations of all shots made, either by cutting the courts in cells (or hexagons or other areas) of the same size and representing their frequency by some color (Chu, 2010). Ehrlich and Sanders (2024) proposed alternative ways to improve the information provided using some model based estimate of the shot efficiency. See also the work on Fu and Stasko (2024) about the importance of visualizing the shooting performance.

However, despite their utility, current shot charts representations face certain limitations. Predominantly, when constructed from observed data using hexagons or derived from standard density estimation procedures, they often fail to take into account the bounded nature of the basketball court. This limitation can result in misleading representations, especially at the boundaries and corners of the court. Consequently, the analysis may not fully account for the contextual constraints imposed by the court's physical boundaries, potentially skewing the assessment of shooting patterns and efficiency, particularly in areas where players are more inclined to attempt shots due to strategic advantages or positional play. These discrepancies underscore the importance of refining shot charts methodologies to accurately depict the nuanced spatial dynamics inherent in basketball shot data.

Figure 1a illustrates the approach commonly used to visualize the spatial distribution of a player's shot attempts. Typically, a two-dimensional kernel density estimate is used (Scott, 2009). However, if the boundaries of a basketball court are not taken into account, some artifacts are noticeable, particularly in the corner 3-point areas, behind the backboard, and in front of the center 3-point line. In contrast, by adopting the methodology proposed in this paper we obtain a density estimate that remains confined within the physical boundaries of the basketball court and, by providing more accurate spatial estimates, effectively remove the above mentioned artifacts.

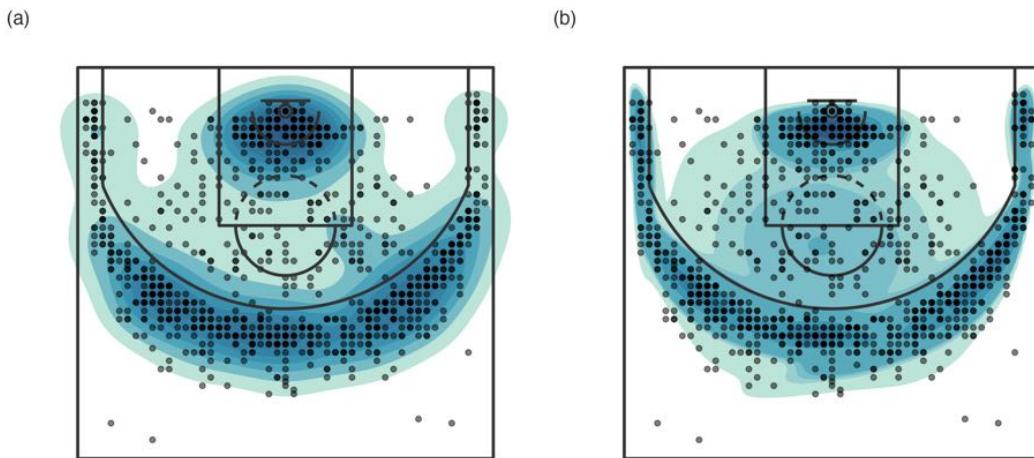


Figure 1: Distribution of Stephen Curry's shot attempts during the 2022-23 NBA regular season. Panel (a) shows the density estimate obtained using two-dimensional kernel density estimation, while panel (b) the estimate obtained by fitting Gaussian mixtures for bounded data, which allows the physical boundaries of the basketball court to be taken into account.

To summarize, in our proposal we embrace a model-based approach to shot charts estimation and visualization that: 1) employs Gaussian mixtures to estimate the density distribution of made and missed shots; 2) takes into account the physical boundaries of the basketball court; 3) applies Bayes' rule to derive estimates for the probability of successful shooting from any location; and 4) identifies regions with the highest expected scores.

The paper is organized as follows: Section 2 describes the model and the estimation procedure; Section 3 illustrates the proposed methodology using the data from the 2022-23 NBA regular season for two players, namely Stephen Curry, perhaps the GOAT (*Greatest Of All Time*) 3-point shooter, and Joel Embiid, the MVP (*Most Valuable Player*) for that season; the final section contains some concluding remarks and potential future extensions to this paper.

2 Methods

Shot charts in basketball analytics provide a visual representation of a player or team's shooting performance by analyzing data on shots attempted from various spots on the court. However, basketball courts come in many different sizes. In the NBA, the court is 94 by 50 feet (28.7 by 15.2 m), while under the International Basketball Federation (FIBA) rules, the court is slightly smaller, measuring 28 by 15 meters (91.9 by 49.2 ft). The 3-point line is also different, being located at 23 feet 9 inches (7.24 m) from the center of the basket in the NBA (22 ft or 6.70 m at the corner), and 6.75 m (22 ft 1.75 in) for FIBA (6.60 m or 21 ft 8 in at the corner). As discussed in reference to the results shown in Figure 1, these physical constraints on the basketball court must be given due consideration in density estimation from shots spatial information.

Figure 2 shows the shots attempted by Stephen Curry (left panel) and Joel Embiid (right panel) during the 2022-23 NBA regular season with each data point marked by shot outcome. The significant presence of shots from beyond the arc of the 3-point line is evident for Curry, while a greater number of attempts in the mid-range can be traced for Embiid. However, partly because of the presence of overlapping points, it is difficult to identify the spots from which the two players preferentially and most effectively shoot at the basket. Thereby, density estimation becomes crucial for gaining insights into shooting patterns and optimizing players performance, or to set up an efficient defense that limits shooting opportunities at preferred positions.

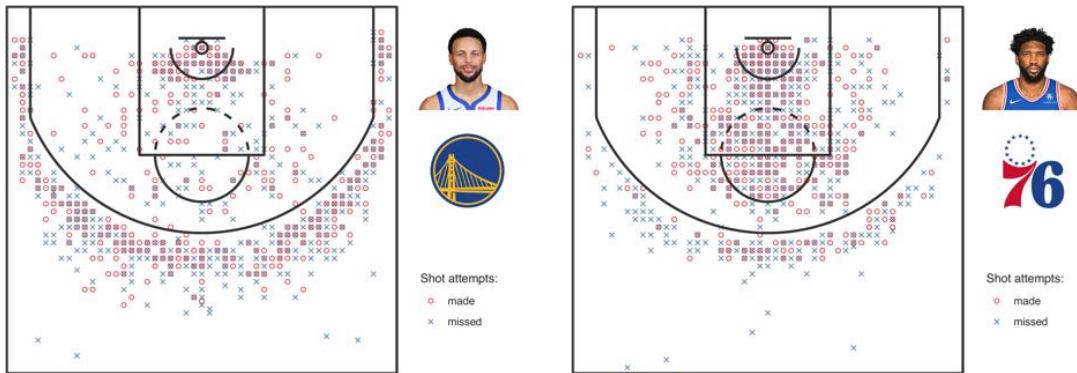


Figure 2: Shots attempted by Stephen Curry and Joel Embiid during the 2022-23 NBA regular season.

Gaussian mixtures (McLachlan and Peel, 2000; Fraley and Raftery, 2002) offer a semiparametric approach to density estimation. In this approach, the density of the data is expressed as a convex linear combination of one or more probability density functions. Gaussian mixtures are a popular choice obtained by using Gaussian densities as components of the mixture.

Gaussian Mixtures Models (GMMs) carry several advantages due to their intrinsic probabilistic generative nature. In particular, maximum likelihood estimation of parameters is available via the EM algorithm (see Section 2.2), with estimates that remain efficient even for multidimensional data. Moreover, GMMs require no hyperparameters tuning, with the problem of selecting the complexity of the mixture that can be recast as model selection problem (see Section 2.2).

Despite the fact that GMMs can approximate any continuous density with arbitrary accuracy, provided the mixture has an adequate number of components (see Ferguson, 1983; Escobar and West, 1995, among others), it is crucial to consider the inherent physical constraints of the basketball half-court when estimating densities in shot charts. This can be achieved by adopting the transformation-based approach to Gaussian mixture density estimation for bounded data proposed by Scrucca (2019). This approach is particularly suitable for this scenario because it explicitly considers the natural bounds of the basketball half-court. Next section briefly reviews the methodology of our proposal.

2.1 Model specification

The transformation-based approach for GMMs discussed in Scrucca (2019) extends density estimation using mixture modeling to the case of bounded variables. The basic idea is to carry out density estimation not on the original data but on appropriately transformed scale. Then, the density for the original data can be simply obtained by a change of variables.

Let (x_i, y_i) denote the coordinates of the position on the court where a player attempts a shot, for $i = 1, \dots, n$, where n is the number of shots attempted, and $C_i = \{0, 1\}$ the corresponding binary outcome, where 1 indicates a made shot and 0 a missed shot. Consider the coordinate-wise range-logit transformation defined as

$$t(x, y) = \begin{bmatrix} t(x) \\ t(y) \end{bmatrix} = \begin{bmatrix} \log\left(\frac{x - \ell_x}{u_x - x}\right) \\ \log\left(\frac{y - \ell_y}{u_y - y}\right) \end{bmatrix},$$

where (ℓ_x, u_x) and (ℓ_y, u_y) are the lower and upper bounds along, respectively, the x -axis and the y -axis. Figure 3 shows the coordinates of the half-court we consider in our study for a 94 by 50 feet NBA basketball court. Thus, half-court court boundaries are set at $(\ell_x = -25, u_x = 25)$ and $(\ell_y = 0, u_y = 47)$.

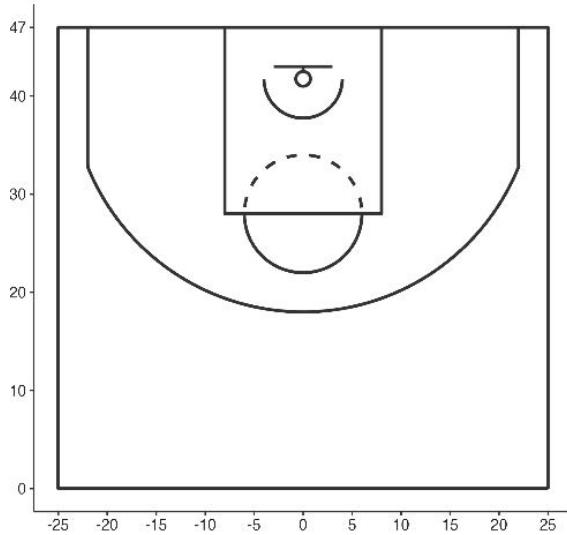


Figure 3: NBA half-court dimensions and coordinates (in feet) used in the present paper.

In the logit-range transformed scale the density of a shot from location (x, y) can be expressed using the following Gaussian mixture

$$h(t(x, y)) = \sum_{g=1}^G \pi_g \mathcal{N}(t(x), t(y) | \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g), \quad (1)$$

where G is the number of mixture components, π_g the mixing probabilities (with $\pi_g > 0$ and $\sum_{g=1}^G \pi_g = 1$), $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$, respectively, the mean vector and covariance matrix for Gaussian component g . Upon re-expressing it in the original coordinate scale, the density function can be formulated as follows:

$$f(x, y) = h(t(x, y)) \times |\mathbf{J}(t(x, y))|, \quad (2)$$

where $|\mathbf{J}(t(x, y))|$ is the Jacobian of the transformation. According to the coordinate-wise transformation approach adopted, the matrix of first derivatives is diagonal, so the Jacobian reduces to the product of first derivatives, i.e.

$$|\mathbf{J}(t(x, y))| = t'(x) \times t'(y) = \left(\frac{1}{x - \ell_x} + \frac{1}{u_x - x} \right) \times \left(\frac{1}{y - \ell_y} + \frac{1}{u_y - y} \right).$$

The density in the transformed coordinates from (1) can be estimated separately for made ($C = 1$) and missed shots ($C = 0$), and then back-transformed in the original scale using (2). Subsequently, the probability of scoring a basket from a specific location can be calculated using Bayes' theorem. Specifically, the density at location (x, y) for shot outcome $C = k$, with $k = \{0, 1\}$, is given by

$$f(x, y | C = k) = \left(\sum_{g=1}^{G_k} \pi_{g|k} \mathcal{N}(t(x), t(y) | \boldsymbol{\mu}_{g|k}, \boldsymbol{\Sigma}_{g|k}) \right) \times |\mathbf{J}(t(x, y))|, \quad (3)$$

where G_k represents the number of mixture components for shot outcome $C = k$. The $\pi_{g|k}$ terms denote the mixing probabilities for outcome $C = k$ ($\pi_{g|k} > 0$ and $\sum_{g=1}^{G_k} \pi_{g|k} = 1$), and $\boldsymbol{\mu}_{g|k}$ along with $\boldsymbol{\Sigma}_{g|k}$ stand for the mean vectors and covariance matrices for component g of outcome $C = k$.

Once the density is estimated for both made shots, $f(x, y | C = 1)$, and missed shots, $f(x, y | C = 0)$, the probability of a successful shot can be obtained using Bayes' rule as:

$$\Pr(C = 1 | x, y) = \frac{\tau_1 f(x, y | C = 1)}{\tau_0 f(x, y | C = 0) + \tau_1 f(x, y | C = 1)}, \quad (4)$$

where τ_1 and τ_0 are the outcome prior probabilities of, respectively, made and missed shots.

The estimated probabilities of making shots from various positions on the court in (4) can be multiplied by the point value of those shots (2 or 3 points) to derive the *expected points scored*:

$$\text{EPS}(x, y) = \begin{cases} 2 \times \Pr(C = 1 | x, y) & \text{if } (x, y) \text{ is within the 3-point line} \\ 3 \times \Pr(C = 1 | x, y) & \text{if } (x, y) \text{ is beyond the 3-point line} \end{cases}$$

This represents an important metric which provides valuable insights into offensive strategies and efficiency from different positions on the court.

2.2 Estimation and model selection

Estimation of unknown parameters, $\pi_{g|k}$, $\boldsymbol{\mu}_{g|k}$, $\boldsymbol{\Sigma}_{g|k}$, for $g = 1, \dots, G_k$ and $k = \{0, 1\}$, in (3) can be pursued via the EM algorithm. For details see Scrucca (2019, Sec. 3.3). Moreover, outcome prior probabilities, τ_1 and τ_0 , in (4) can be estimated from, respectively, the proportions of made and missed shots.

Without imposing any constraints on the covariance matrices of Gaussian components, empirical evidence suggests the inclusion of a Bayesian regularization prior to increase smoothness of the density estimate over the basketball court and avoid singularities and degeneracies in maximization of the likelihood. This can be achieved by adopting the approach of Fraley and Raftery (2007), who proposed weakly informative conjugate priors to regularize the estimation process. The EM algorithm can still be used for model fitting, but maximum likelihood estimates (MLEs) are replaced by maximum a posteriori (MAP) estimates. For details see Scrucca et al. (2023, Sec. 7.2).

A crucial aspect in mixture modeling is the choice of the number of mixture components, G_k , for each outcome. Typically, the Bayesian Information Criterion (BIC; Schwarz, 1978) is used as model selection criterion in finite mixture models. This choice is justified by Keribin (2000), who demonstrated that BIC is consistent for choosing the number of components in a mixture model, assuming a bounded likelihood (which is guaranteed by the introduction of the regularized prior mentioned earlier). However, when Bayesian regularization is introduced a slightly modified version of BIC should be used for model selection, with the maximized log-likelihood replaced by the log-likelihood evaluated at the MAP.

3 Applications

In this section we analyzed the player-by-player data of some selected NBA players for the 2022-23 NBA regular season. The data are obtained from the R package `hoopR` (Gilani, 2023), which provides easy access to data available on ESPN analytics at <https://www.espn.com/nba/>.

3.1 Stephen Curry

Figure 4 shows the estimated densities for made (a) and missed (b) shots, respectively $f(x, y|C = 1)$ and $f(x, y|C = 0)$ from (3). Regions are highlighted by highest density regions (HDRs) corresponding to specific percentages of the data. Note, however, that these cannot be directly compared, but they can be used for computing shot probabilities using (4). Required prior probabilities are estimated using proportions of made and missed shots during the regular season, giving $\hat{\tau}_1 = 0.4724$ and $\hat{\tau}_0 = 0.5276$.

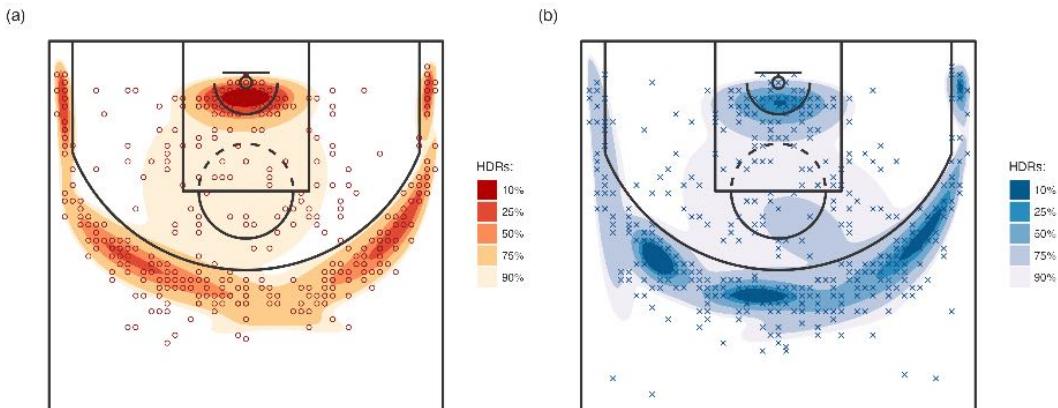


Figure 4: Highest density regions (HDRs) from mixture-based estimated densities for made (a) and missed (b) shots for Stephen Curry during the 2022-23 NBA regular season.

Figure 5a presents the estimated shot chart highlighting regions of high and low probability for made shots by Stephen Curry. The chart reveals a remarkable consistency in Curry's

shooting ability across various regions, with particularly high probabilities close to the basket and extending well beyond the three-point line. Notably, two key exceptions emerge: very far locations and positions approximately 2-3 feet from the three-point line at the top of the key. Additionally, a closer look suggests a reduced probability in the right mid-range area.

Building upon the estimated shot chart discussed above, Figure 5b presents the corresponding graph of expected points scored. This visualization highlights regions of high scoring efficiency, primarily concentrated around close-range shots and extending to all areas beyond the three-point arc, with a notable preference for the left side. Interestingly, these high-efficiency regions align with areas of higher shot probability observed in Figure 5a, while regions with lower expected points coincide with areas of lower shot probability.

Lastly, the table below Figure 5 summarizes key statistics for both two-point and three-point attempts: number of attempts, observed made shot proportions, estimated average probabilities, observed average points per attempt, and estimated expected score. Notably, the empirical and estimated values exhibit close agreement, highlighting the accuracy of the model. These data showcase Stephen Curry's remarkable offensive efficiency beyond the three-point arc, reflected in an estimated expected score of 1.27 points per attempt compared to 1.10 points for closer shots.

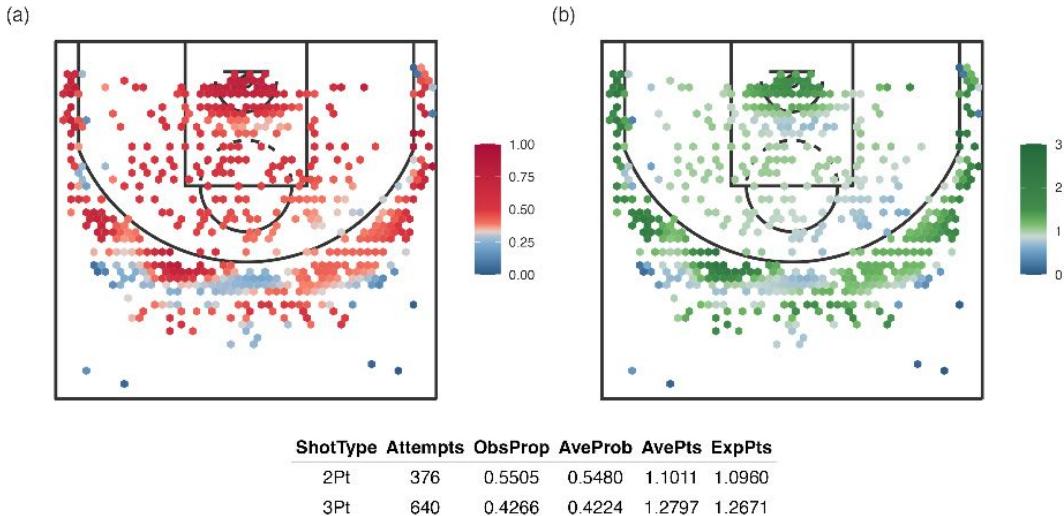


Figure 5: Shot charts depicting (a) estimated probabilities and (b) expected points scored per attempt for Stephen Curry during the 2022-23 NBA regular season. The table below the charts reports a summary of empirical and estimated key statistics for both two-point and three-point shots.

3.2 Joel Embiid

As a second player we analyze Joel Embiid of the Philadelphia 76ers. Compared to Stephen Curry's role as shooting guard, Embiid plays as a center, is much taller and stronger physically, but at the same time has an excellent aptitude for shooting from mid-range and beyond the arc. During the 2022-23 regular season Embiid had the highest average points per game (30.6) and won the MVP award.

Charts in Figure 6 show the highest density regions (HDRs) obtained from mixture-based estimated densities for made (a) and missed (b) shots. The majority of shots are concentrated in the paint and near the free-throw line, while beyond the three-point arc Embiid's favorite position appears to be the central one.

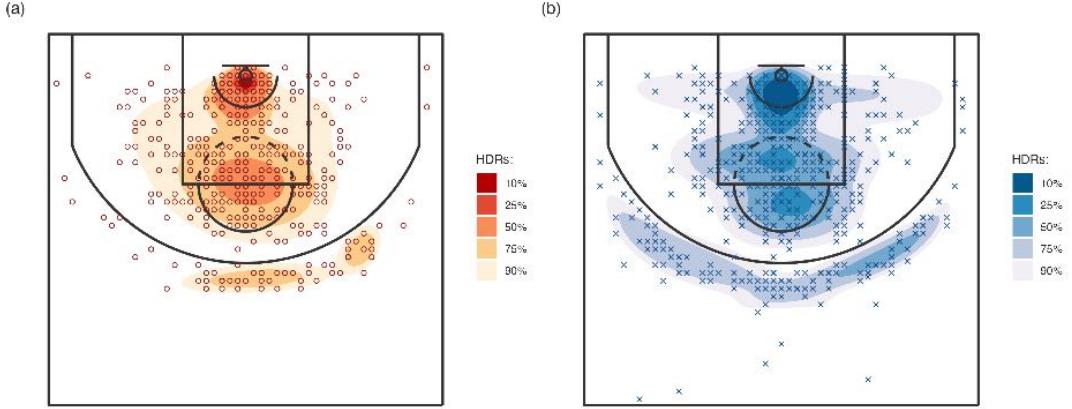


Figure 6: Highest density regions (HDRs) from mixture-based estimated densities for made (a) and missed (b) shots for Joel Embiid during the 2022-23 NBA regular season.

Embiid's shooting efficiency is very high, as can be seen from the chart in Figure 7a, with estimated success probabilities well above 50% in almost all mid-range and close-to-basket positions. For three-point shots, two preferred positions with very high success rates emerge: in front of the basket and slightly to the right. In other positions beyond the arc, the estimated probabilities appear significantly lower.

In terms of expected points scored from different positions, the most profitable ones are near the basket, thanks to the high shooting percentage, and those with the highest efficiency beyond the arc, due to the fact that more points are obtained for each basket made.

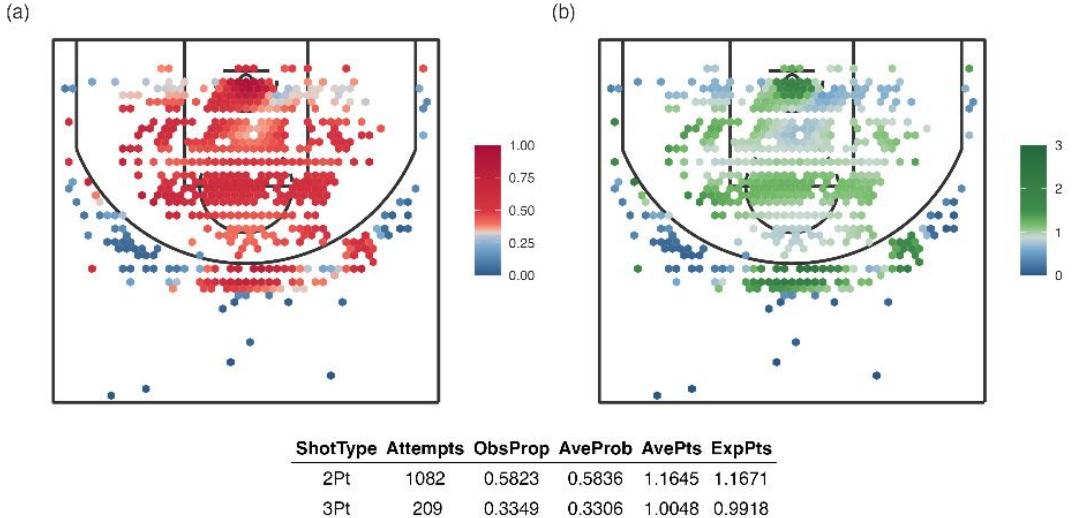


Figure 7: Shot charts depicting (a) estimated probabilities and (b) expected points scored per attempt for Joel Embiid during the 2022-23 NBA regular season. The table below the charts reports a summary of empirical and estimated key statistics for both two-point and three-point shots.

Finally, it is interesting to compare the different shooting choices of Stephen Curry and Joel Embiid, and their relative effectiveness and efficiency (see tables at the bottom of Figures

7 and 5). Curry favors long-distance shots, attempting approximately 70% more three-point shots (640 attempts compared to 376), while Embiid notably focuses on within the arc shots (1082 attempts compared to 209). Curry exhibits high estimated probabilities of scoring for both 2-point (54%) and, especially, 3-point (42%) shots, whereas Embiid demonstrates a higher percentage in the mid- and close-range shots (58%), but only a moderate 3-point percentage (33%), which is nonetheless excellent for his role. These translate into excellent expected points for both 2-point and 3-point attempts, with Curry astonishingly averaging about 1.27 points per 3-point attempt.

4 Conclusions

The availability of good quality spatial data in sports has increased a lot their usage, including spatial visualizations. For example, we are all familiar with heatmaps that represent the location density of players as an attempt to describe their playing behavior but also to identify tactics. Shot charts are pivotal tools in basketball analytics, offering valuable insights into players' shooting tendencies and efficiencies across different areas of the court. Existing shot chart representations often fall short in accurately capturing shooting spatial distribution, primarily due to their inability to account for the bounded nature of the basketball court. In the present paper we proposed a new approach that employs Gaussian mixtures to estimate the density distribution using a transformation-based approach that takes into account the physical boundaries of the court. We demonstrate the effectiveness of our methodology through case studies involving real-world data from the 2022-23 NBA regular season.

The easiness of applying and fitting Gaussian mixtures to estimate the spatial distribution creates additional opportunities. An explicit extension of the proposed work relates to all other sports where spatial location data are used. Recall also that this may extend to other non-sport related applications where boundaries need to be taken into account. As a proposal for further investigation, we also mention the use of mixture models as the basis for *conditional heatmaps*. So far, most of the sports visualization based on tracking data is based on the position of a player in the court. Sometimes it is interesting to visualize the conditional heatmap, i.e. the position of a player conditional on the position of some other player. For example, in basketball (but also in football and other team sports) this can reveal important tactical aspects and space creation strategies for the teams, which is an important ingredient of the game. Gaussian mixtures allow easily to work on that since one can easily obtain/estimate the joint distribution of the location of two players as the joint distribution in 4 dimensions, allowing also for dependence. From the joint distribution one can estimate the conditional density in a straightforward manner and thus produce a conditional heatmap.

References

- Chen, W., Lao, T., Xia, J., Huang, X., Zhu, B., Hu, W., and Guan, H. (2016). Gameflow: narrative visualization of NBA basketball games. *IEEE Transactions on Multimedia*, 18(11):2247–2256.
- Chu, S. (2010). Information visualization in the NBA: The shot chart. Technical report, University of California, Berkeley.
- Ehrlich, J. and Sanders, S. (2024). Estimating NBA team shot selection efficiency from aggregations of true, continuous shot charts: A generalized additive model approach. Available at SSRN: <https://ssrn.com/abstract=4697111>.
- Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588.
- Ferguson, T. (1983). Bayesian density estimation by mixtures of normal distributions. In Rizvi, M. H., Rustagi, J. S., and Siegmund, D., editors, *Recent Advances in Statistics*, pages 287–302. Academic Press.
- Fichman, M. and O'Brien, J. R. (2019). Optimal shot selection strategies for the NBA. *Journal of Quantitative Analysis in Sports*, 15(3):203–211.
- Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631.
- Fraley, C. and Raftery, A. E. (2007). Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of Classification*, 24(2):155–181.
- Fu, Y. and Stasko, J. (2024). HoopInSight: Analyzing and comparing basketball shooting performance thr visualization. *IEEE Transactions on Visualization & Computer Graphics*, 30(1):858–868.
- Gilani, S. (2023). *hoopR: Access Men's Basketball Play by Play Data*. R package version 2.1.0.
- Goldsberry, K. (2012). CourtVision: New visual and spatial analytics for the NBA. In *2012 MIT Sloan Sports Analytics Conference*, volume 9, pages 12–15.
- Jiao, J., Hu, G., and Yan, J. (2021). A Bayesian marked spatial point processes model for basketball shot chart. *Journal of Quantitative Analysis in Sports*, 17(2):77–90.
- Keribin, C. (2000). Consistent estimation of the order of mixture models. *Sankhya Ser. A*, 62(1):49–66.
- Kubatko, J., Oliver, D., Pelton, K., and Rosenbaum, D. T. (2007). A starting point for analyzing basketball statistics. *Journal of Quantitative Analysis in Sports*, 3(3).
- McLachlan, G. J. and Peel, D. (2000). *Finite Mixture Models*. Wiley, New York.
- Oughali, M. S., Bahloul, M., and El Rahman, S. A. (2019). Analysis of NBA players and shot prediction using random forest and XGBoost models. In *2019 International Conference on Computer and Information Sciences (ICCIS)*, pages 1–5. IEEE.
- Papalexakis, E. and Pelechrinis, K. (2018). thoops: A multi-aspect analytical framework for spatio-temporal basketball data. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 2223–2232.

- Pelechrinis, K. and Goldsberry, K. (2021). The anatomy of corner 3s in the NBA: What makes them efficient, how are they generated and how can defenses respond? *arXiv preprint arXiv:2105.12785*.
- Perin, C., Vuillemot, R., Stolper, C. D., Stasko, J. T., Wood, J., and Carpendale, S. (2018). State of the art of sports data visualization. *Computer Graphics Forum*, 37(3):663–686.
- Reich, B. J., Hodges, J. S., Carlin, B. P., and Reich, A. M. (2006). A spatial analysis of basketball shot chart data. *The American Statistician*, 60(1):3–12.
- Sandholtz, N., Mortensen, J., and Bornn, L. (2020). Measuring spatial allocative efficiency in basketball. *Journal of Quantitative Analysis in Sports*, 16(4):271–289.
- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.
- Scott, D. W. (2009). *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 2nd edition.
- Scrucca, L. (2019). A transformation-based approach to Gaussian mixture density estimation for bounded data. *Biometrical Journal*, 61(4):873–888.
- Scrucca, L., Fraley, C., Murphy, T. B., and Raftery, A. E. (2023). *Model-Based Clustering, Classification, and Density Estimation Using mclust in R*. Chapman & Hall/CRC.
- Shortridge, A., Goldsberry, K., and Adams, M. (2014). Creating space to shoot: quantifying spatial relative field goal efficiency in basketball. *Journal of Quantitative Analysis in Sports*, 10(3):303–313.
- Terner, Z. and Franks, A. (2021). Modeling player and team performance in basketball. *Annual Review of Statistics and Its Application*, 8:1–23.
- Zuccolotto, P. and Manisera, M. (2020). *Basketball data science: With applications in R*. CRC Press.
- Zuccolotto, P., Manisera, M., and Sandri, M. (2018). Big data analytics for modeling scoring probability in basketball: The effect of shooting under high-pressure conditions. *International Journal of Sports Science & Coaching*, 13(4):569–589.

A transformation-based approach to Gaussian mixture density estimation for bounded data

Luca Scrucca 

Department of Economics, Università degli Studi di Perugia, Italy

Correspondence

Luca Scrucca, Department of Economics, Università degli Studi di Perugia, Via A. Pascoli 20, 06123 Perugia, Italy.
Email: luca.scrucca@unipg.it

Abstract

Finite mixture of Gaussian distributions provide a flexible semiparametric methodology for density estimation when the continuous variables under investigation have no boundaries. However, in practical applications, variables may be partially bounded (e.g., taking nonnegative values) or completely bounded (e.g., taking values in the unit interval). In this case, the standard Gaussian finite mixture model assigns nonzero densities to any possible values, even to those outside the ranges where the variables are defined, hence resulting in potentially severe bias. In this paper, we propose a transformation-based approach for Gaussian mixture modeling in case of bounded variables. The basic idea is to carry out density estimation not on the original data but on appropriately transformed data. Then, the density for the original data can be obtained by a change of variables. Both the transformation parameters and the parameters of the Gaussian mixture are jointly estimated by the expectation-maximization (EM) algorithm. The methodology for partially and completely bounded data is illustrated using both simulated data and real data applications.

KEY WORDS

bounded support, density estimation, EM algorithm, Gaussian mixture models, range-power transformation

1 | INTRODUCTION

Density estimation is the problem of inferring a probability density function given a finite number of sample data points drawn from a population described by a probability distribution. Broadly speaking, three alternative approaches to density estimation can be distinguished. In the parametric approach, a parametric distribution is assumed for the density with unknown parameters, which are estimated by fitting the parametric function using the observed data. Conversely, in the nonparametric approach no density function is assumed a priori, but its form is completely determined by the data. Histograms and kernel density estimation (KDE) are two popular methods that belong to this class, and both are characterized by the number of parameters growing with the size of the data set. Furthermore, extension to higher dimensionality is problematic. A third approach is based on finite mixture models, where the unknown density is expressed as a convex combination of one or more probability density functions. In this class, a popular model is the Gaussian mixture model (GMM) that assumes the Gaussian distribution for the underlying component densities. GMMs can approximate any continuous density with arbitrary accuracy provided the model has a sufficient number of components and the parameters of the model are correctly estimated (Escobar & West, 1995; Roeder & Wasserman, 1997).

Bounded data are quite common in biomedical data analyses because of the measurement scale of the data, or the type of variables under study. However, the standard GMM for density estimation does not take into account whether or not a variable has bounded support. Consider the graphs in Figure 1, which show some histograms for random samples drawn

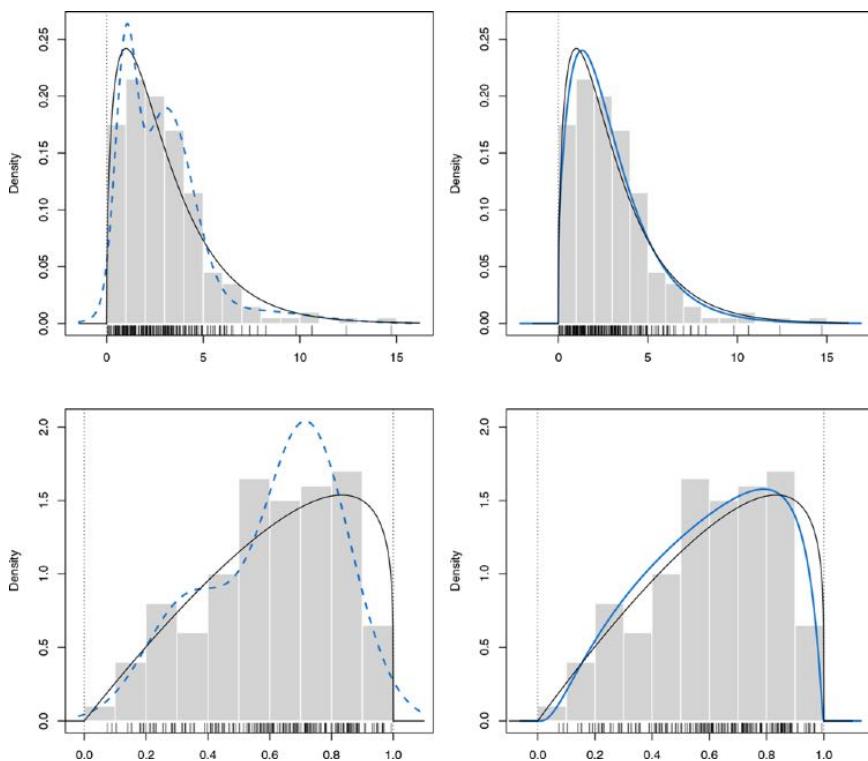


FIGURE 1 Histograms for random samples drawn from a $\chi^2(3)$ (top panels) and a beta(2, 1.2) (bottom panels) distributions with the corresponding density functions (solid lines) and boundaries of the random variable (vertical dotted lines)

Note: Panels on the left show the estimated densities obtained by fitting a GMM on the original scale (blue dashed lines). Panels on the right show the densities estimated by the GMDEB transformation approach (blue solid lines).

from two distributions, one bounded from below (top panels), and one having both lower and upper bounds (bottom panels). In these graphs boundaries are shown as vertical dots, true densities are represented as solid lines, and density estimates based on GMMs as dashed lines (see left panels of Figure 1). In both cases, the estimated densities are unsatisfactory not only at the boundaries but also in the range of admissible values. A possible way to tackle these problems is to abandon the use of GMMs in favor of alternative component distributions. Another option is to remain in the realm of the Gaussian mixtures framework, but analyze the data in a transformed scale. The right panels of Figure 1 show the density estimates obtained with the Gaussian mixture density estimation for bounded (GMDEB) data approach proposed in this paper. In both cases, the true underlying densities appear to be well approximated, and with natural boundaries constraints clearly satisfied.

In this paper, a transformation-based approach to density estimation based on GMMs is proposed and discussed. The basic idea is to use an invertible function to map a bounded variable to an unbounded support, estimate the density of the transformed variable, and then back-transform to the original scale. This approach seems very natural and it has been around for a long time (see Marron & Ruppert, 1994; Wand, Marron & Ruppert, 1991, for KDE), but a simple and efficient implementation of this methodology is not yet available in the context of mixture density estimation. Note that a similar approach based on Manly transformation has been recently proposed by Zhu and Melnykov (2018) for modeling skewed data in model-based clustering. However, our proposal differs in two main respects: first, it has been designed to allow variables with bounded support, second, it aims at a different goal, that is, density estimation as compared to clustering.

In Section 2, the GMMs approach to density estimation is reviewed. Then, Section 3 presents the proposed range-power transformation method for density estimation using GMMs in case of bounded variables. The model is described and the corresponding maximum likelihood estimates are derived through the EM algorithm. Section 4 contains the results of some simulation studies carried out to evaluate the proposed methodology and to compare it with other available methods. In Section 5, some real-world data sets are analyzed. The final section provides some concluding remarks.

2 | FINITE MIXTURE MODELING

2.1 | Finite mixture for density estimation

Consider a vector of random variables \mathbf{x} taking values in the sample space $\mathcal{S}_{\mathcal{X}} \subseteq \mathbb{R}^p$ with $p \geq 1$, and assume that the probability density function can be written as a finite mixture density of G components of the form

$$f(\mathbf{x}; \boldsymbol{\Psi}) = \sum_{g=1}^G \pi_g f_g(\mathbf{x}; \boldsymbol{\theta}_g),$$

where $\boldsymbol{\Psi} = (\pi_1, \dots, \pi_{G-1}, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_G)^\top$ is the parameters vector. The mixing weights (π_1, \dots, π_G) must satisfy the constraints $\pi_g > 0$ for all $g = 1, \dots, G$, and $\sum_{g=1}^G \pi_g = 1$. The g th component density $f_g(\mathbf{x}; \boldsymbol{\theta}_g)$ is usually taken as known except for the associated parameter(s) $\boldsymbol{\theta}_g$. Most applications assume that all component densities arise from the same parametric distribution family, although this need not be the case in general. In particular, a popular model specifies $f_g(\mathbf{x}; \boldsymbol{\theta}_g) \equiv \phi(\mathbf{x}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$, where $\phi(\cdot)$ is the Gaussian density with mean $\boldsymbol{\mu}_g$ and covariance matrix $\boldsymbol{\Sigma}_g$. Then, the GMM can be written as

$$f(\mathbf{x}; \boldsymbol{\Psi}) = \sum_{g=1}^G \pi_g \phi(\mathbf{x}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g), \quad (1)$$

where in this case $\boldsymbol{\Psi} = (\pi_1, \dots, \pi_{G-1}, \boldsymbol{\mu}_1^\top, \dots, \boldsymbol{\mu}_G^\top, \text{vech}\{\boldsymbol{\Sigma}_1\}^\top, \dots, \text{vech}\{\boldsymbol{\Sigma}_G\}^\top)$ represents the entire parameters vector. Note that $\text{vech}\{\cdot\}$ is an operator that forms a vector by extracting unique elements of a symmetric matrix.

In this paper, we refer to (1) as the Gaussian mixture density estimate (GMDE) model. The usual nonparametric KDE can be viewed as a mixture of $G = n$ components with uniform weights, that is, $\pi_g = 1/n$ (Titterington, Smith & Makov, 1985, pp. 28–29). Compared to KDE, finite mixture modeling uses a smaller number of components (i.e., less parameters), so it has smaller variance. Conversely, compared to parametric density estimation, finite mixture modeling has the advantage of (potentially) using more parameters, so introducing less estimation bias. There are also disadvantages related to mixture modeling, such as an increased learning complexity and lack of closed-form solution, so it needs to resort to numerical procedures (e.g., EM algorithm), and in certain cases there can be identifiability issues.

2.2 | Estimation of Gaussian finite mixture model

Consider a random sample $\mathbf{x}_1, \dots, \mathbf{x}_n$ of n observations on p variables drawn from the mixture distribution in (1). Then, the log-likelihood is given by

$$\ell(\boldsymbol{\Psi}) = \sum_{i=1}^n \log \sum_{g=1}^G \pi_g \phi(\mathbf{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g). \quad (2)$$

Direct maximization of the log-likelihood function is not straightforward, so MLEs are usually obtained via the expectation–maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977; McLachlan & Peel, 2000).

An incomplete-data formulation of the mixture problem is introduced by associating to each observation a latent component-label vector \mathbf{z}_i ($i = 1, \dots, n$). This is a G -dimensional vector, with the generic element $z_{ig} = 1$ or 0 according to whether or not \mathbf{x}_i arises from the g th component of the mixture. Assuming independence of the complete-data vector $(\mathbf{x}_i^\top, \mathbf{z}_i^\top)^\top$, and the multinomial distribution for the component-label vectors \mathbf{z}_i s, the complete-data log-likelihood is given by

$$\ell_C(\boldsymbol{\Psi}) = \sum_{i=1}^n \sum_{g=1}^G z_{ig} \{ \log \pi_g + \log \phi(\mathbf{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \}. \quad (3)$$

The log-likelihood (2) is maximized using the EM algorithm, an iterative algorithm that alternates two steps, called E-step and M-step, which guarantees, under fairly general conditions, the convergence to at least a local maximizer. The objective function at iteration $(m+1)$ of the EM algorithm is the conditional expectation of the complete-data log-likelihood (3), the so-called Q -function:

$$Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(m)}) = \sum_{i=1}^n \sum_{g=1}^G \hat{z}_{ig}^{(m)} \{ \log \pi_g + \log \phi(\mathbf{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \},$$

where $\hat{z}_{ig}^{(m)} = E(I(z_i = g)|\mathbf{x}_i, \Psi^{(m)})$, that is, the estimated posterior probability at iteration m of the EM algorithm, with $I(\cdot)$ the indicator function, which equals 1 if the condition is fulfilled and 0 otherwise.

In the E-step the Q -function is evaluated, using the parameter values π_g, μ_g, Σ_g obtained at the previous step, to get the updated posterior probabilities

$$\hat{z}_{ig}^{(m+1)} = \frac{\hat{\pi}_g^{(m)} \phi(\mathbf{x}_i; \hat{\mu}_g^{(m)}, \hat{\Sigma}_g^{(m)})}{\sum_{k=1}^G \hat{\pi}_k^{(m)} \phi_k(\mathbf{x}_i; \hat{\mu}_k^{(m)}, \hat{\Sigma}_k^{(m)})}.$$

Then, in the M-step the parameters vector Ψ is updated by maximizing the Q -function given the previous values $\hat{\Psi}^{(m)}$ and the updated posterior probabilities $\hat{z}_{ig}^{(m+1)}$, that is,

$$\hat{\Psi}^{(m+1)} = \arg \max_{\Psi} Q(\Psi; \hat{\Psi}^{(m)}).$$

In the case of a multivariate Gaussian mixture the M-step yields

$$\hat{\pi}_g^{(m+1)} = \frac{\sum_{i=1}^n \hat{z}_{ig}^{(m+1)}}{n} \quad \text{and} \quad \hat{\mu}_g^{(m+1)} = \frac{\sum_{i=1}^n \hat{z}_{ig}^{(m+1)} \mathbf{x}_i}{\sum_{i=1}^n \hat{z}_{ig}^{(m+1)}}.$$

The update formula for the covariance matrix depends on the structure of the within-component covariance matrices. Parsimonious parameterization of the covariance matrices can be expressed through the eigendecomposition $\Sigma_g = v_g \mathbf{O}_g \mathbf{A}_g \mathbf{O}_g^\top$, where v_g is a scalar controlling the volume of the corresponding ellipsoid, \mathbf{A}_g is a diagonal matrix specifying the shape of the density contours, and \mathbf{O}_g is an orthogonal matrix, which determines the orientation of the ellipsoid (Banfield & Raftery, 1993; Celeux & Govaert, 1995). For instance, assuming an unconstrained covariance matrix, the updating formula is

$$\hat{\Sigma}_g^{(m+1)} = \frac{\sum_{i=1}^n \hat{z}_{ig}^{(m+1)} (\mathbf{x}_i - \hat{\mu}_g^{(m+1)}) (\mathbf{x}_i - \hat{\mu}_g^{(m+1)})^\top}{\sum_{i=1}^n \hat{z}_{ig}^{(m+1)}}.$$

Scrucca, Fop, Murphy, and Raftery (2016, table 3) summarize some parameterizations of within-component covariance matrices, and the corresponding geometric characteristics, currently available in the `mclust` software. The previous unconstrained covariance matrix is indicated as VVV model. Note, however, that for some models no closed-formula is available, so numerical optimization is required.

The EM algorithm requires the specification of initial values for the parameters, say $\Psi^{(0)}$. Alternatively, an initial assignment of observations to the components of the mixture can be made, basically starting the EM algorithm from the M-step. The initialization of the EM algorithm is often crucial because the likelihood surface tends to have multiple modes, and in certain cases it can be even unbounded. Nevertheless, the EM algorithm usually produces sensible results when started from reasonable starting values (Wu, 1983, p. 150). For a further discussion on this point and a recent proposal, see Scrucca and Raftery (2015).

Information criteria based on penalized forms of the log-likelihood are routinely used in finite mixture modeling for model selection, that is, to decide not only how many components should be included in the mixture but also which covariance parameterizations to adopt in the Gaussian case. Two popular criteria are the Bayesian information criterion (BIC; Fraley & Raftery, 1998; Schwartz, 1978) and the integrated complete-data likelihood (ICL) criterion (Biernacki, Celeux & Govaert, 2000). When the goal is density estimation, Roeder and Wasserman (1997) showed that the GMDE model selected using BIC is a consistent estimator of the true density. If only the order of the mixture is needed, formal hypothesis testing can also be pursued by the likelihood ratio test (LRT). However, standard regularity conditions do not hold for the null distribution of the LRT statistic to have its usual chi-squared distribution (McLachlan & Peel, 2000, Chap. 6), and significance must be assessed by resampling approaches. For a recent review, see McLachlan and Rathnayake (2014), and for an implementation in the `mclust` software see Scrucca et al. (2016).

3 | METHODOLOGY

3.1 | GMDE for variables with bounded support

Let \mathbf{x} be a p -variate random vector from a distribution with density f having bounded support $S_{\mathcal{X}} \subset \mathbb{R}^p$, and $\{t(\mathbf{x}; \lambda); \lambda \in \Lambda\}$ be some family of continuous monotonic transformations that map $S_{\mathcal{X}}$ to an unbounded p -dimensional support. Then, we can write $\mathbf{y} = t(\mathbf{x}; \lambda)$ as the transformed set of variables with density h having unbounded support $S_{\mathcal{Y}}$.

Suppose that the density of the transformed data can be expressed through a Gaussian finite mixture density of the form

$$h(\mathbf{y}; \boldsymbol{\Psi}) = \sum_{g=1}^G \pi_g \phi(\mathbf{y}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g). \quad (4)$$

Then, by the continuous change of variable theorem, the density of the untransformed data can be expressed as

$$f(\mathbf{x}; \boldsymbol{\Psi}, \lambda) = h(t(\mathbf{x}; \lambda)) \cdot |\mathbf{J}(t(\mathbf{x}; \lambda))|,$$

where $\mathbf{J}(t(\mathbf{x}; \lambda))$ is the Jacobian of the transformation, that is, the determinant of the matrix of partial derivatives.

3.2 | Range–power transformation for variables with bounded support

3.2.1 | Lower bound case

Suppose x is a univariate random variable with lower bounded support $S_X \equiv (l, \infty)$, where $l > -\infty$ and density $f(x)$. Consider a preliminary range transformation defined as $x \mapsto (x - l)$, which maps $S_X \rightarrow \mathbb{R}^+$. Let $\{t(x; \lambda \in \Lambda)\}$ be a continuous monotonic transformation. Based on the well-known Box–Cox transformation (Box & Cox, 1964), we consider the following *range–power* transformation

$$t(x; \lambda) = \begin{cases} \frac{(x - l)^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(x - l) & \text{if } \lambda = 0, \end{cases} \quad (5)$$

which has continuous first derivative equal to $t'(x; \lambda) = (x - l)^{\lambda-1}$ for any $\lambda \in \Lambda$.

The original Box–Cox power transformation method is restricted to the univariate case, but it can be extended also to the multivariate case as described in Velilla (1993). However, further development of the multivariate case $\mathbf{x} = (x_1, \dots, x_p)^\top$ can be greatly simplified by working in a coordinate-wise fashion. Thus, in this paper, we propose the use of the range–power transformation in (5) for each dimension separately.

3.2.2 | Lower and upper bound case

Suppose now that x is a univariate random variable with bounded support $S_X \equiv (l, u)$, where $-\infty < l < u < +\infty$. Consider the preliminary range transformation $x \mapsto (x - l)/(u - x)$, which maps $S_X \rightarrow \mathbb{R}^+$. As in the previous case, adopting a *range–power* transformation we can write

$$t(x; \lambda) = \begin{cases} \frac{\left(\frac{x-l}{u-x}\right)^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log\left(\frac{x-l}{u-x}\right) & \text{if } \lambda = 0, \end{cases} \quad (6)$$

with continuous first derivative given by

$$t'(x; \lambda) = \begin{cases} \left(\frac{x-l}{u-x}\right)^{\lambda-1} \frac{u-l}{(u-x)^2} & \text{if } \lambda \neq 0 \\ \frac{1}{x-l} + \frac{1}{u-x} & \text{if } \lambda = 0. \end{cases}$$

Following the approach discussed in Section 3.2.1, the multivariate case can be tackled by working in a coordinate-wise fashion, hence applying the range–power transformation in (6) to each variable separately.

3.3 | Estimation

Maximum likelihood estimation can be pursued via the EM algorithm under the assumption that the density on the transformed scale can be expressed as in (4), with $\mathbf{y} = t(\mathbf{x}; \lambda)$ the vector of range–power transformed variables according to (5) or (6). If the previous assumption holds, then the density function on the original scale is given by

$$f(\mathbf{x}; \Psi, \lambda) = \sum_{g=1}^G \pi_g \phi(t(\mathbf{x}; \lambda); \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \cdot |\mathbf{J}(t(\mathbf{x}; \lambda))|, \quad (7)$$

where $t(\mathbf{x}; \lambda) = (t(x_1; \lambda_1), \dots, t(x_p; \lambda_p))^\top$ and $\mathbf{J}(t(\mathbf{x}; \lambda))$ is the Jacobian of the transformation. Note that as a consequence of the coordinate independent approach to multivariate range–power transformation, the matrix of first derivatives is diagonal, so the Jacobian reduces to

$$\mathbf{J}(t(\mathbf{x}; \lambda)) = \det \left[\frac{\partial t(\mathbf{x}; \lambda)}{\partial \mathbf{x}} \right] = \prod_{j=1}^p \frac{\partial t(x_j; \lambda_j)}{\partial x_j}.$$

The conditional expectation of the complete-data log-likelihood given the observed data can be expressed as

$$Q(\Psi; \Psi^{(m)}) = \sum_{i=1}^n \sum_{g=1}^G \hat{z}_{ig}^{(m)} \{ \log \pi_g + \log \phi(t(\mathbf{x}_i; \lambda); \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) + \log |\mathbf{J}(t(\mathbf{x}_i; \lambda))| \},$$

where $\hat{z}_{ig}^{(m)} = E(I(z_i = g) | \mathbf{x}_i, \Psi^{(m)})$. Therefore, in the E-step the posterior probabilities are updated using

$$\hat{z}_{ig}^{(m+1)} = \frac{\hat{\pi}_g^{(m)} \phi(t(\mathbf{x}_i; \hat{\lambda}^{(m)}); \hat{\boldsymbol{\mu}}_g^{(m)}, \hat{\boldsymbol{\Sigma}}_g^{(m)})}{\sum_{k=1}^G \hat{\pi}_k^{(m)} \phi(t(\mathbf{x}_i; \hat{\lambda}^{(m)}); \hat{\boldsymbol{\mu}}_k^{(m)}, \hat{\boldsymbol{\Sigma}}_k^{(m)})}.$$

In the M-step, the parameters (Ψ, λ) are updated by maximizing the Q -function given the previous values of the parameters and the updated posterior probabilities. According to the expectation-conditional-maximization (ECM) algorithm proposed by Meng and Rubin (1993), this maximization can be pursued in two steps. In the first step, an updated value $\hat{\lambda}^{(m+1)}$ is computed by numerically maximizing the Q -function with respect to λ because no closed-form expression is available. To this goal, a Newton-type numerical optimization algorithm can be used. Although, in theory, λ parameters are unrestricted, in practice reasonable values are typically found in a narrower range, for example, between -3 and 3 . For this reason, in our implementation we used the L-BFGS-B method of Byrd, Lu, Nocedal, and Zhu (1995) available in the `optim()` function for the R statistical software. The remaining parameters are then obtained as in standard EM algorithm but accounting for the updated transformation parameters $\hat{\lambda}^{(m+1)}$, that is,

$$\hat{\pi}_g^{(m+1)} = \frac{\sum_{i=1}^n \hat{z}_{ig}^{(m+1)}}{n} \quad \text{and} \quad \hat{\boldsymbol{\mu}}_g^{(m+1)} = \frac{\sum_{i=1}^n \hat{z}_{ig}^{(m+1)} t(\mathbf{x}_i; \hat{\lambda}^{(m+1)})}{\sum_{i=1}^n \hat{z}_{ig}^{(m+1)}}.$$

Again, the update formula for the covariance matrix depends on the assumed eigendecomposition model. In the most general case of an unconstrained covariance matrix, that is, the VVV model, we have

$$\hat{\boldsymbol{\Sigma}}_g^{(m+1)} = \frac{\sum_{i=1}^n \hat{z}_{ig}^{(m+1)} (t(\mathbf{x}_i; \hat{\lambda}^{(m+1)}) - \hat{\boldsymbol{\mu}}_g^{(m+1)}) (t(\mathbf{x}_i; \hat{\lambda}^{(m+1)}) - \hat{\boldsymbol{\mu}}_g^{(m+1)})^\top}{\sum_{i=1}^n \hat{z}_{ig}^{(m+1)}}.$$

Initialization of the above EM algorithm is obtained by first estimating the optimal marginal transformations, then using the final classification from a k -means algorithm on the range–power transformed variables. This initial partition of data points is used to start the algorithm from the M-step. Finally, the EM algorithm is stopped when the log-likelihood improvement falls below a specified tolerance value or a maximum number of iterations is reached.

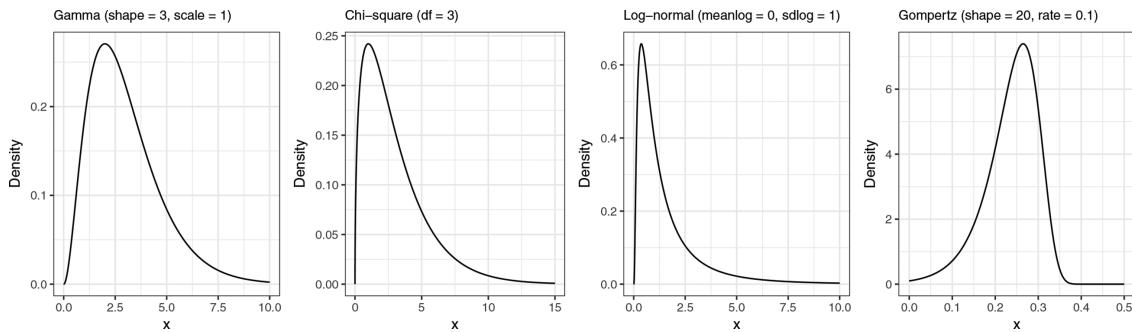


FIGURE 2 Univariate densities with lower bound considered in the simulation study

4 | SIMULATION STUDIES

In this section, we present some simulation studies designed to compare the proposed GMDEB approach to some density estimators for bounded variables discussed in the literature. The comparison is based on the integrate squared error (ISE):

$$\text{ISE}(\hat{f}) = \int \left[\hat{f}(x) - f(x) \right]^2 dx,$$

where f is the unknown true density and \hat{f} is its estimate based on a random sample of n observations. Thus, the ISE is a measure of discrepancy between the true and the estimated density based on a squared loss criterion. It is equal to 0 when the estimated density perfectly coincides with the true density, and increases as the differences between the two densities get larger. For more details, see Scott (2009, section 2.3). In the following sections, the ISE is computed via numerical integration.

4.1 | Distributions with lower bound

The univariate densities with lower bounded support considered in this study are shown in Figure 2. They are all bounded at zero with different degrees of skewness, positive for the first three densities and negative for the last one.

The proposed method (GMDEB) is compared with the following density estimators:

- simpleKern, which refers to the simple boundary correction method proposed by Jones (1993), which is equivalent to a kernel weighted local linear fitting near the boundary;
- reflectKern, which indicates the reflection method of Schuster (1985), which amounts to reflect the observed data points at the origin, then a density estimate is obtained using this augmented data set with a simple correction to ensure that integrate to one;
- cutnormKern, which is the cut and normalization method of Gasser and Müller (1979), where the kernel is truncated at the boundary and renormalized to unity;
- logtransKern, which is the method proposed by Marron and Ruppert (1994), which fits a KDE on the log-scale and then back-transforms the result with an explicit normalization step;
- logSpline, which estimates a density using cubic splines to approximate the log-density using knots located as described in Stone, Hansen, Kooperberg, and Truong (1997);
- GaMixDE, which estimates a density by fitting a mixture of Gamma densities;
- GMDE, which is the standard density estimate from GMMs with no boundary correction.

The first four methods mentioned above are implemented in the `evmix` R package (Hu & Scarrott, 2018; Scarrott, Hu, Akbar & of Canterbury, 2018), whereas the `logSpline` estimator is available in the `logspline` R package (Kooperberg, 2016). For `GaMixDE`, the code is available in the R package `mixtools` (Benaglia, Chauveau, Hunter & Young, 2009; Young et al., 2017), whereas `GMDE` is obtained from the `mclust` R package (Fraley, Raftery, & Scrucca, 2017). In the last two cases, the number of mixture components is selected using the BIC criterion.

Figure 3 graphically summarizes the simulation results obtained on 1,000 replications. Overall, the GMDEB estimator appears to be able to approximate the true density better than the other KDE methods with boundary correction, in particular when the

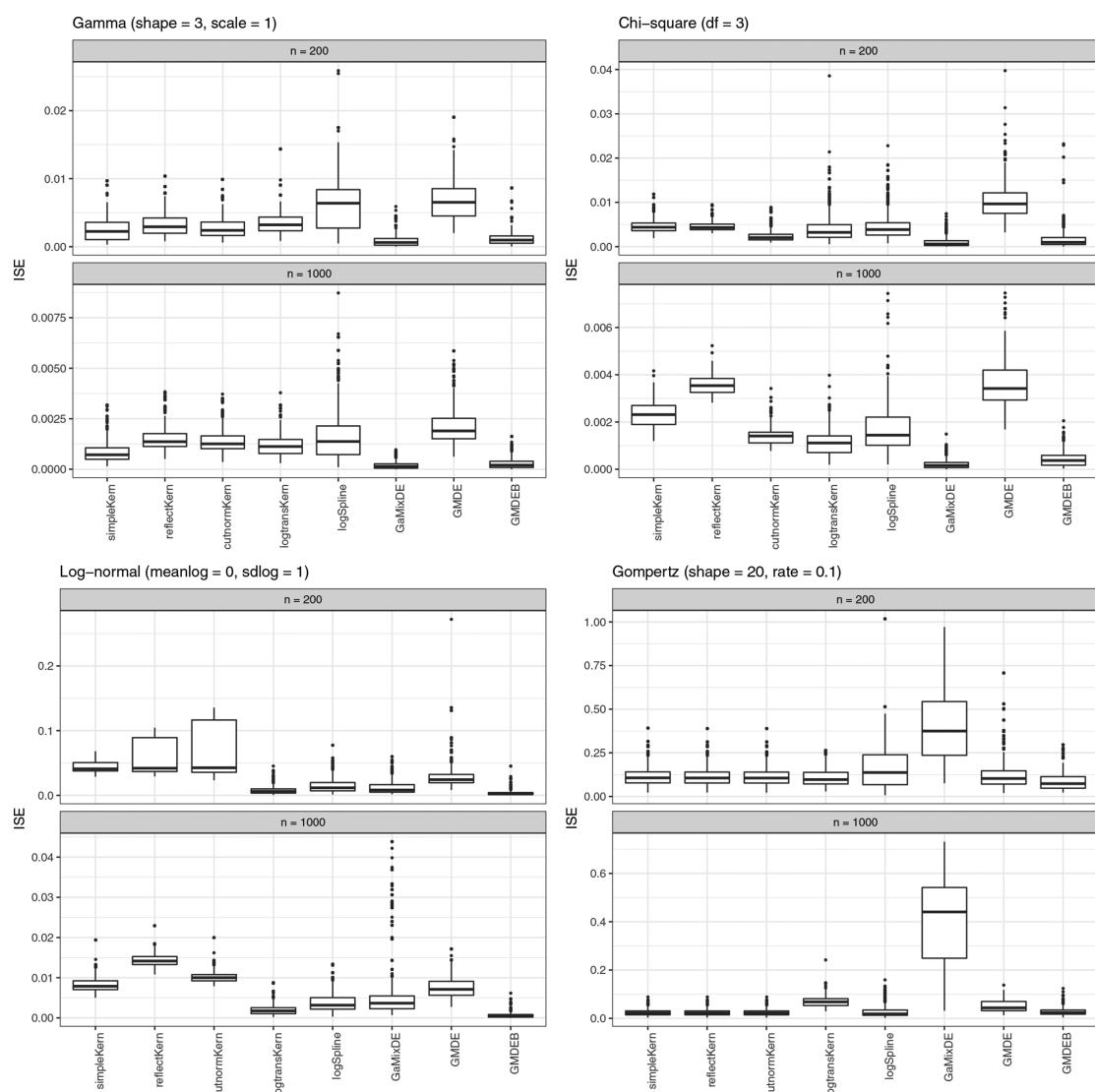


FIGURE 3 Boxplots of ISE distribution from 1,000 replications of the simulation study for the selected univariate densities with lower bound

sample size is small. The proposed approach also shows less variability, which decreases as the sample size increases. Clearly, the GMDEB approach appears to be inferior to the Gamma mixture density estimator, although not by much, when the true density belongs to the Gamma family of distributions (i.e., in the first two cases), but it is better in the last two cases, in particular for the left-skewed Gompertz distribution.

4.2 | Distributions with lower and upper bounds

For the case of univariate densities with both lower and upper bounded support, a list of distributions considered in the simulation study is shown in Figure 4. The first two settings involve the beta distribution with parameters selected to produce a symmetric case and a skewed case. The last two settings consider two further asymmetric cases, the Kumaraswamy distribution with density $f(x) = \alpha\beta x^{\alpha-1}(1-x^\alpha)^{\beta-1}$ on $[0,1]$ and the logarithmic peak distribution with density $f(x) = -\log(x)$ on $(0, 1)$.

The cases described above should provide a broad spectrum of examples for comparing different estimators. To this goal, the proposed method (GMDEB) is compared with the following density estimators:

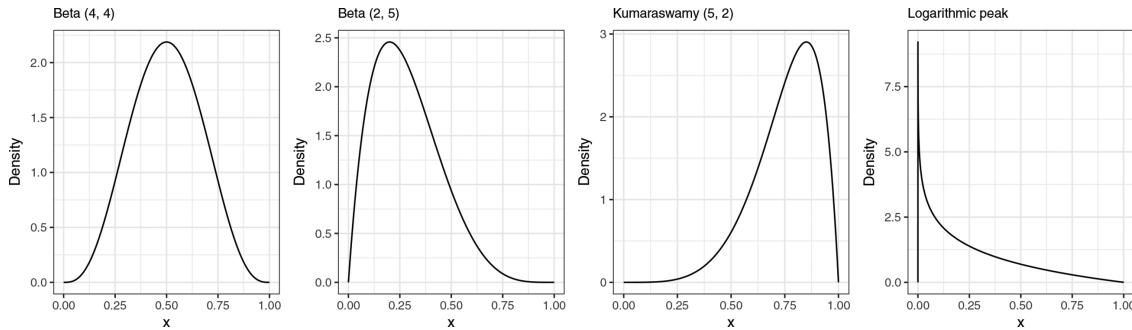


FIGURE 4 Univariate densities with lower and upper bounds considered in the simulation study

- `beta1Kern`, `beta2Kern`, which use the beta and modified beta kernels proposed by Chen (1999) followed by a renormalization to ensure a proper density;
- `copulaKern`, which uses the bivariate Gaussian copula-based kernels of Jones and Henderson (2007);
- `logSpline`, which fits a density using cubic splines to approximate the log-density using knots located as described in Stone et al. (1997);
- `BeMixDE`, which estimates the density by fitting a mixture of beta distributions.

The first two estimators are available in the R package `evmix` (Hu & Scarrott, 2018; Scarrott et al., 2018). The `logSpline` estimator is available in the `logspline` R package (Kooperberg, 2016). For the `BeMixDE` the `betareg` R package (Grün, Kosmidis & Zeileis, 2012) is used with the number of mixture components selected using BIC.

Figure 5 reports the simulation results obtained on 1,000 replications. By looking at the boxplots, the GMDEB approach appears to be more accurate than the other nonparametric density estimators. Furthermore, its accuracy is slightly lower than the beta mixture density estimator in the first three cases (which, however, are all cases related to the beta distribution), but is better in the last case. Overall, GMDEB seems to provide robust reliable density estimates when both lower and upper bounds are present.

4.3 | A note on computing time

A major concern might be the computational effort required by the estimation of the λ parameter through numerical optimization within the EM algorithm. To investigate the runtime of the proposed procedure, we designed a small simulation study where we generated data from a chi-square distribution with 3 degrees of freedom for sample sizes $n = \{200, 1000, 10000\}$. A multivariate case was also investigated by considering a 10-dimensional variable with independent marginals drawn as in the univariate case. We estimated the density of GMDEB by both estimating the λ parameter, and by fixing it at the corresponding MLE value. Furthermore, we executed the algorithm both sequentially and in parallel (over the mixture components and covariance parameterizations). Experiments were carried out on an iMac with 4 cores i5 Intel CPU running at 2.8 GHz and with 16GB of RAM.

Figure 6 shows the results averaged over 100 replications of the experiments. Clearly, in the univariate case the effect of estimating the transformation parameter is negligible. On the contrary, the effect is visible in the multivariate case, but a considerable speedup can be achieved by parallelization. The worst case, that is, transformation parameter to be estimated sequentially with a sample of size 10,000 on 10 dimensions, required on average just over 1 minute.

5 | REAL DATA ANALYSES

5.1 | Acidity data

This data set provides the values of an acidity index (acid-neutralizing capacity, ANC) measured in a sample of 155 lakes in North-Central Wisconsin. Several authors have previously analyzed the data using a mixture of Gaussian distributions on the log-scale (Crawford, 1994; Crawford, DeGroot, Kadane, & Small, 1992; McLachlan & Peel, 2000; Richardson & Green, 1997).

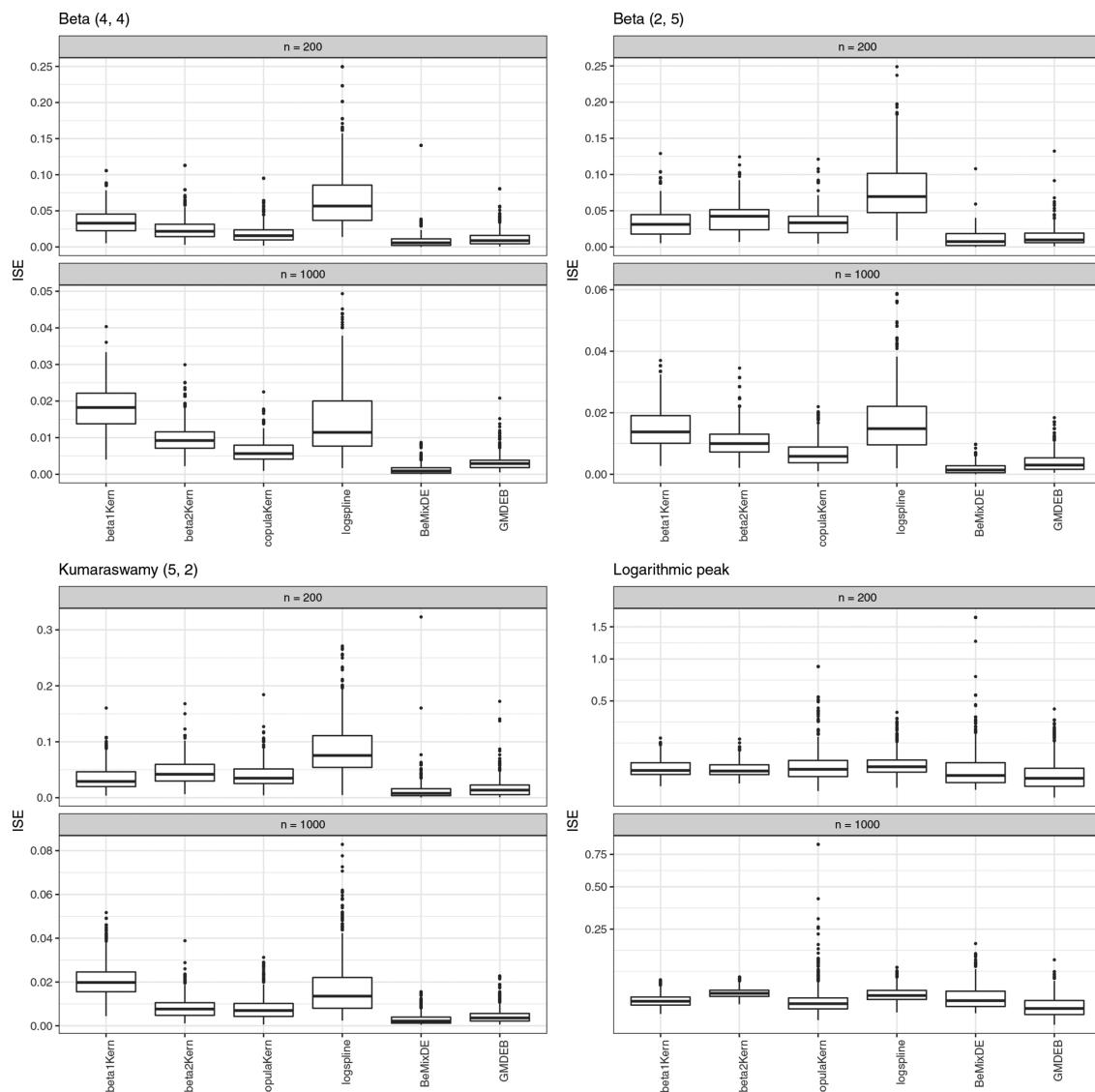


FIGURE 5 Boxplots of ISE distribution from 1,000 replications of the simulation study for the selected univariate densities with lower and upper bounds

On the contrary, we analyze the data in the original scale because the proposed method automatically selects the “optimal” transformation and takes into account the implicit lower bound of the index that cannot assume negative values.

From the left panel of Figure 7 we can see that according to BIC the best model is the one with two mixture components having different variances (V,2), closely followed by models (E,2) and (V,3). The right panel of Figure 7 shows the histogram of the data and the density estimated with model (V,2) using the GMDEB approach with transformation parameter $\hat{\lambda} = -0.293$ (blue thick line). For comparison, we also draw the density estimated by GMM without any boundary correction (black dashed line). The density estimated by GMDEB appears to accurately follow the distribution of the data, indicating the presence of two separated skewed distributions having different dispersions, smaller for the component close to the origin and larger for higher values of ANC. This is also confirmed by the graphs in Figure 8, which show the component densities scaled by the estimated prior probabilities $\hat{\pi} = (0.6322, 0.3678)$ (left panel) and the estimated posterior probabilities \hat{z}_{ij} . Using the standard cutoff value of 0.5, lakes with ANC smaller than about 232 are assigned to the first group, otherwise to the second group. This is in agreement with previous findings.

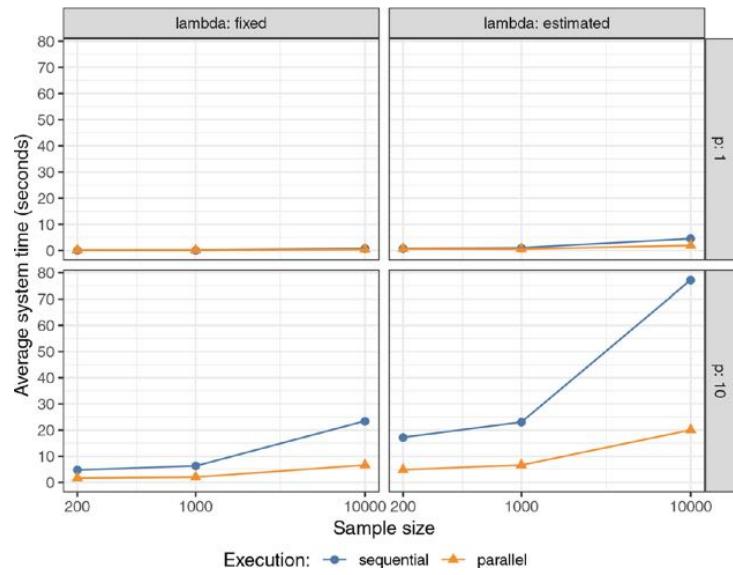


FIGURE 6 Average run times for different sample sizes (expressed in \log_{10} -scale) obtained by considering the transformation parameter either fixed or to be estimated, by running the GMDEB algorithm sequentially or in parallel, and for number of variables 1 and 10

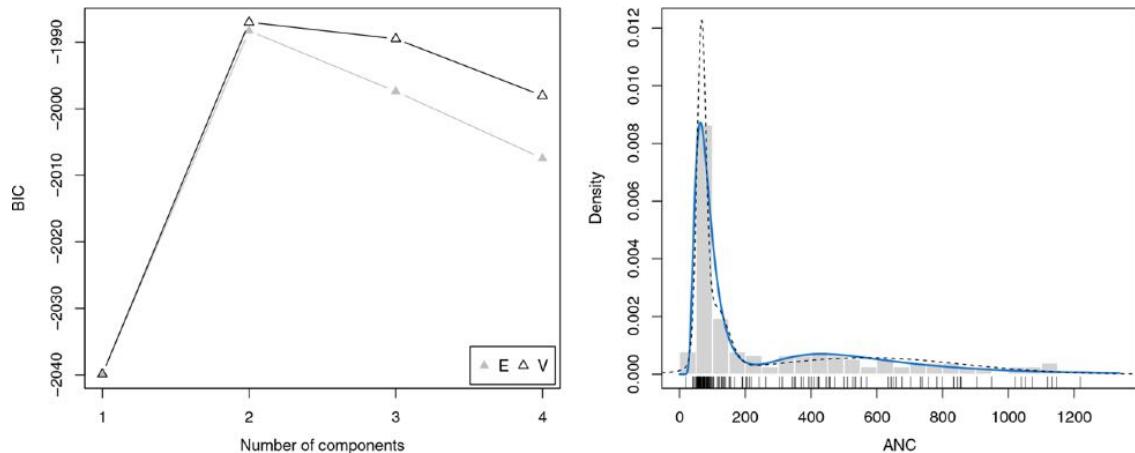


FIGURE 7 Plot of BIC values for different number of mixture components and within-component variances (left panel); histogram of acidity data with GMDEB (blue thick line) and GMM (black dashed line) density estimates (right panel)

5.2 | Racial data

Geenens (2013) presented an analysis on data giving the proportion of white students enrolled in 56 school districts in Nassau County (Long Island, New York), for the 1992–1993 school year. The density estimate for this data set should only be supported on the $[0,1]$ range. See also Simonoff (1996, section 3.2).

The selected model on the transformed scale is (E,1) with $\hat{\lambda} = 0.387$, and the corresponding density estimate on the original scale is shown in Figure 9. This can be compared graphically with the beta density and the beta mixture using two components. Both models were estimated by maximum likelihood using the betareg R package (Grün et al., 2012), with a single component in the first case, and the optimal number of mixture components selected using BIC in the second case. The beta density seems to put too much emphasis close to the upper boundary and in the middle values of the distribution, while completely missing the bulk of the data between 70 and 90% of white students. On the contrary, the proposed GMDEB approach provides an estimate of the density that correctly identify the majority of the data with at least 70% of white students, but also the small peak near the lower boundary containing schools with almost 0% white students. The two-component beta mixture density is quite close to that provided by GMDEB, but the latter should be preferred according to BIC (see table in Figure 9). These findings largely agree with those reported in Geenens (2013, figure 3).

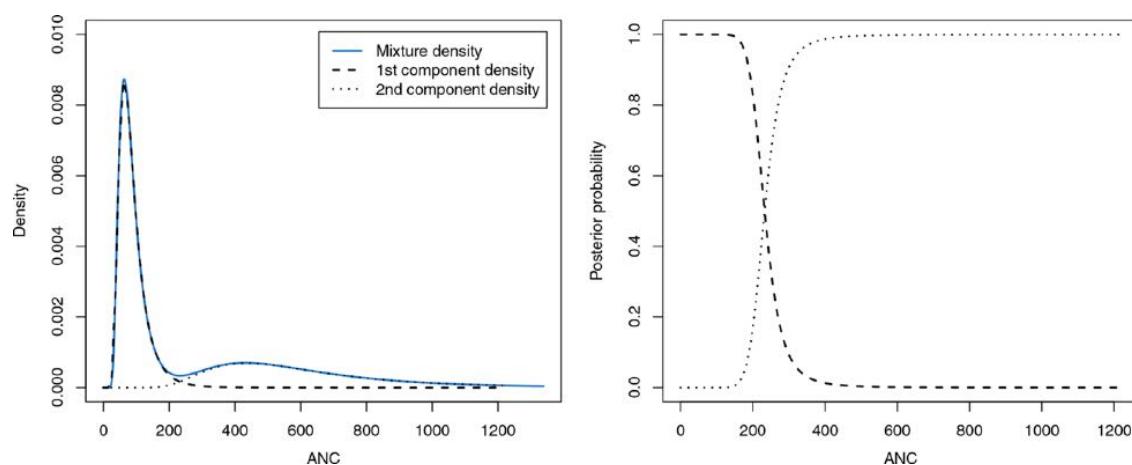


FIGURE 8 Plot of estimated mixture density and rescaled component densities (left panel), and corresponding estimated posterior probabilities (right panel) for the acidity data

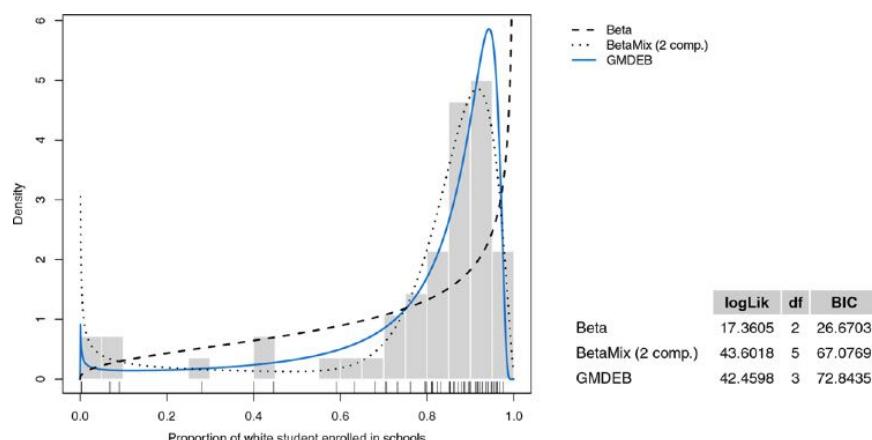


FIGURE 9 Density estimates for the racial data obtained using the GMDEB method, the standard beta distribution, and a two-component beta mixture

Note: The included table reports the log-likelihood, the number of estimated parameters, and the BIC (larger values are preferred).

5.3 | Plasma data

Consider the data from a study on the association between the low plasma concentrations of retinol, beta-carotene, or other carotenoids on the increased risk of developing certain types of cancer (Nierenberg et al., 1989). The joint distribution of plasma retinol (ng/mL) and plasma beta-carotene (ng/mL) is bounded below at zero for both variables. The left panel of Figure 10 shows the scatterplot of data points observed on 314 patients.

If the bivariate density is estimated using the standard GMM, the model with the largest $BIC = -8101.773$ has three components and unconstrained covariance matrices (VVV). This relatively large number of components is related to the presence of a strong skewness in the data distribution. Furthermore, a nonnegligible mass of density is assigned to negative values of plasma beta-carotene.

Both issues can be solved using the proposed range-power transformation approach. The selected model for the bivariate density estimation has $BIC = -8044.852$, with a diagonal equal variance structure and a single component (EII,1). The transformation parameters are estimated as $\hat{\lambda} = (0.155, 0.0295)$. The right panel of Figure 10 shows the highest density regions (HDRs; Hyndman, 1996) corresponding to proportions $(0.25, 0.5, 0.75, 0.9)$. In this case, the joint data distribution appears to be well approximated by the estimated density.

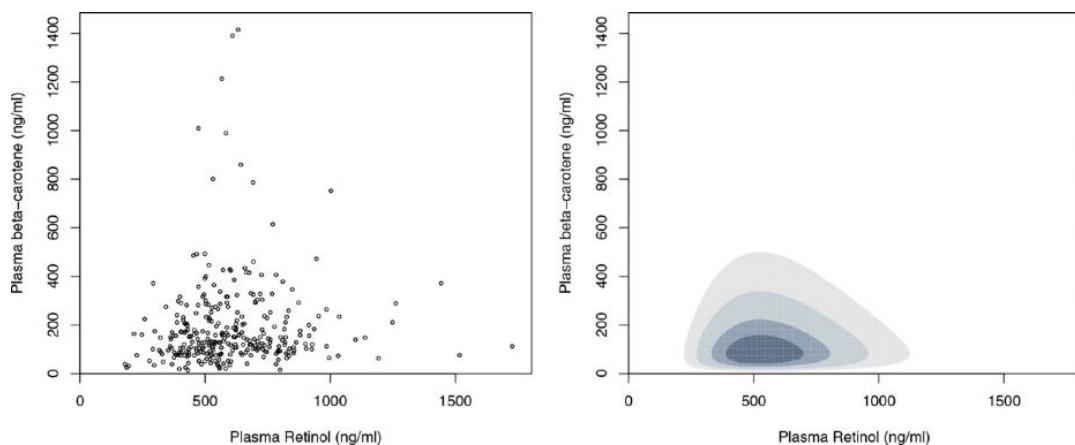


FIGURE 10 Plot of data points for the plasma data set (left panel) and the corresponding bivariate density estimated using the GMDEB approach (right panel)

Note: In the latter case, the graph shows the highest density regions corresponding to proportions (0.25, 0.5, 0.75, 0.9).

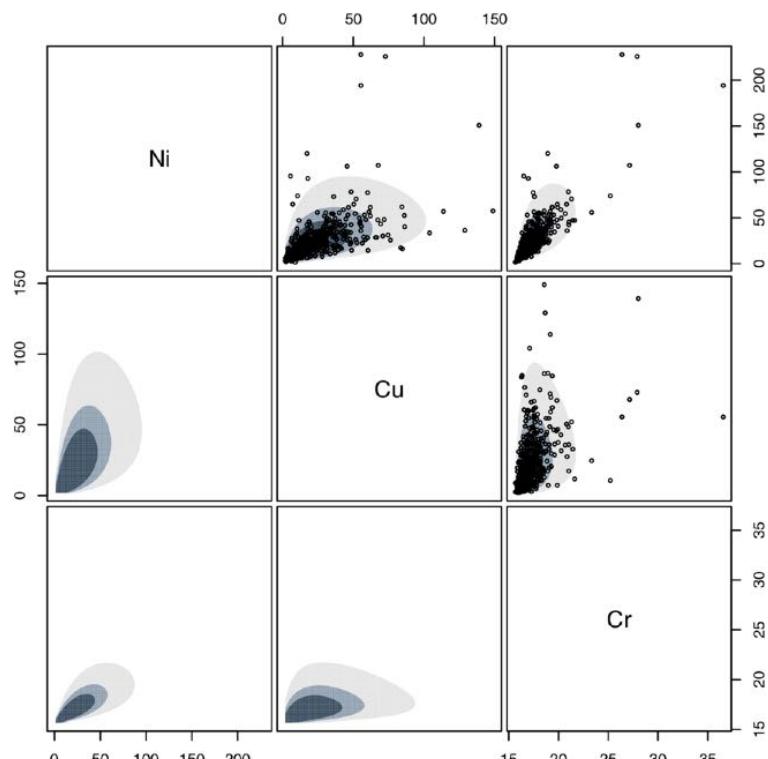


FIGURE 11 Scatterplot matrix of chromium (Cr), copper (Cu), and nickel (Ni) concentrations on the C-horizon layer

Note: The density estimated using the GMDEB approach is shown using HDRs corresponding to 25%, 50%, and 75% probability regions.

5.4 | C-horizon layer of the Kola data

The Kola Ecogeochemistry Project (1993–1998) collected data on more than 50 chemical elements on four different primary sample materials: terrestrial moss, and the O-, B-, and C-horizon of podzolic soils located in parts of northern Finland, Norway, and Russia. The main aim of the project was the documentation of the impact of the Russian nickel industry on the Arctic environment. The data are available on Reimann et al. (1998), see also Reimann, Filzmoser, Garrett, and Dutter (2011). Here, we analyze the distribution of nickel (Ni), copper (Cu), and chromium (Cr) on the C-horizon layer. For each of the 605 sites, the detected concentrations of the abovementioned heavy metals are provided. Clearly, concentrations are bounded below at zero, and a preliminary data exploration suggests that the joint distribution is highly skewed.

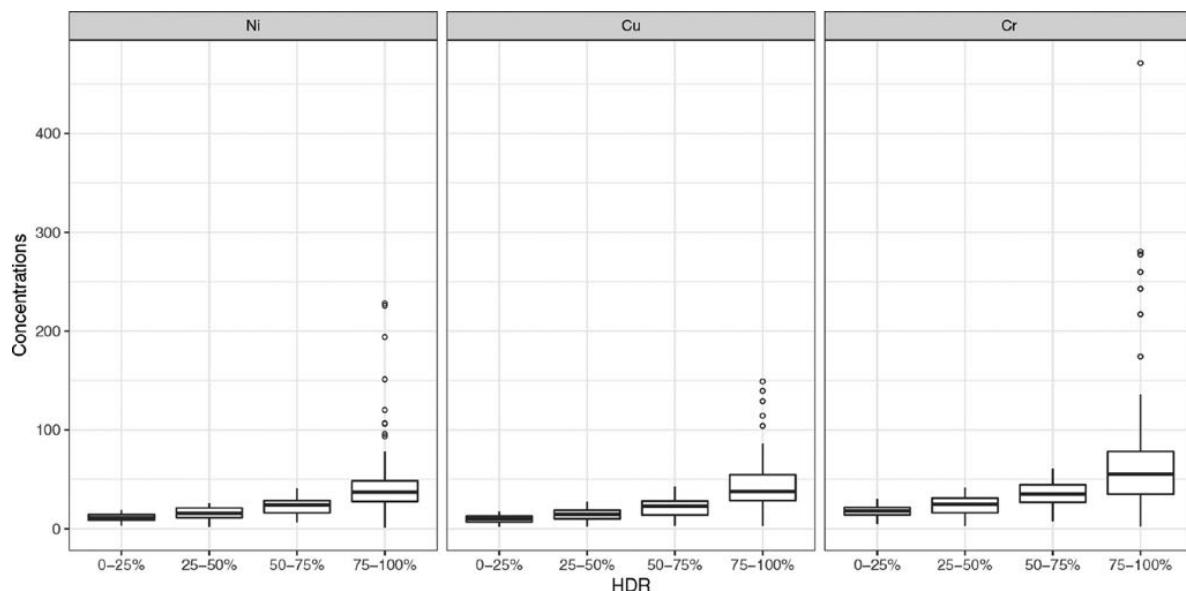


FIGURE 12 Boxplots for the distributions of heavy metals conditional on HDR regions from the GMDEB density estimate

The selected GMDEB model according to the BIC criterion is a two-component mixture model with variable volume and equal shape and orientation, that is, VEE in *mclust* nomenclature. The estimated vector of transformation parameters is $\hat{\lambda} = (-0.0384, 0.0010, -0.0975)$. Figure 11 contains the scatterplot matrix of heavy metal concentrations with the estimated density projected onto the marginal bivariate subspaces. The latter are shown as HDRs corresponding to 25%, 50%, and 75% probability regions. The distribution of nickel, copper, and chromium on the C-horizon layer is clearly skewed, with most sites having concentrations close to the origin. However, there are also a number of sites with relatively high concentrations. Further insights can be obtained by examining the distribution of metal concentrations conditional on the HDR to which the observed sites belong, as shown in Figure 12. Looking at the boxplots for the conditional distributions, we can see that the central part of the distribution, that is, that corresponding to 0–25% HDR, is characterized by the lowest concentration levels, whereas higher concentrations of heavy metals can be found as we move to regions of lower density.

6 | DISCUSSION

This paper addressed the problem of density estimation using GMMs when variables are partially or completely bounded. By introducing a range-power transformation of the data, it is possible to obtain a GMM for density estimation on the transformed data, and then to derive an accurate estimate of the density on the original scale, which takes into account the natural bounds of the variables. The proposed model is estimated by maximum likelihood using the EM algorithm. We showed that this transformation-based approach is able to deal with variables having either lower bounds or both lower and upper bounds, and the results obtained are often better than those provided by other methods usually based on modified versions of KDE.

The transformation-based approach seems to be very promising and, in principle, it could be applied to other types of non-Gaussian variables, for example, skewed variables, and for other purposes outside density estimation, for instance, in clustering. A straightforward extension to investigate is the use of other families of transformations, such as those proposed by Manly (1976) and Yeo and Johnson (2000). Furthermore, although the paper deals with the problem of density estimation, the proposed methodology has implications also on model-based clustering for bounded data. These very important issues are deferred to future works.

CONFLICT OF INTEREST

The author has declared no conflict of interest.

ORCID

Luca Scrucca  <https://orcid.org/0000-0003-3826-0484>

REFERENCES

- Banfield, J., & Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49, 803–821.
- Benaglia, T., Chauveau, D., Hunter, D. R., & Young, D. (2009). mixtools: An R package for analyzing finite mixture models. *Journal of Statistical Software*, 32(6), 1–29.
- Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7), 719–725.
- Box, G., & Cox, D. (1964). An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 26(2), 211–252.
- Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5), 1190–1208.
- Celeux, G., & Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28, 781–793.
- Chen, S. X. (1999). Beta kernel estimators for density functions. *Computational Statistics & Data Analysis*, 31(2), 131–145.
- Crawford, S. L. (1994). An application of the Laplace method to finite mixture distributions. *Journal of the American Statistical Association*, 89(425), 259–267.
- Crawford, S. L., DeGroot, M. H., Kadane, J. B., & Small, M. J. (1992). Modeling lake-chemistry distributions: Approximate Bayesian methods for estimating a finite-mixture model. *Technometrics*, 34(4), 441–453.
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39, 1–38.
- Escobar, M. D., & West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430), 577–588.
- Fraley, C., & Raftery, A. E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 41, 578–588.
- Fraley, C., Raftery, A. E., & Scrucca, L. (2017). *mclust: Gaussian mixture modelling for model-based clustering, classification, and density estimation*. R package version 5.4.
- Gasser, T., & Müller, H.-G. (1979). *Kernel estimation of regression functions*. Heidelberg: Springer.
- Geenens, G. (2013). Probit transformation for kernel density estimation on the unit interval. *Journal of the American Statistical Association*, 109(505), 346–358.
- Grün, B., Kosmidis, I., & Zeileis, A. (2012). Extended beta regression in R: Shaken, stirred, mixed, and partitioned. *Journal of Statistical Software*, 48(11), 1–25.
- Hu, Y., & Scarrott, C. (2018). evmix: An R package for extreme value mixture modeling, threshold estimation and boundary corrected kernel density estimation. *Journal of Statistical Software*, 84(5), 1–27.
- Hyndman, R. J. (1996). Computing and graphing highest density regions. *The American Statistician*, 50(2), 120–126.
- Jones, M. C. (1993). Simple boundary correction for kernel density estimation. *Statistics and Computing*, 3(3), 135–146.
- Jones, M. C., & Henderson, D. A. (2007). Miscellanea kernel-type density estimation on the unit interval. *Biometrika*, 94(4), 977–984.
- Kooperberg, C. (2016). *logspline: Logspline Density Estimation Routines*. R package version 2.1.9.
- Manly, B. F. J. (1976). Exponential data transformations. *The Statistician*, 25(1), 37–42.
- Marron, J. S., & Ruppert, D. (1994). Transformations to reduce boundary bias in kernel density estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 56, 653–671.
- McLachlan, G., & Peel, D. (2000). *Finite mixture models*. New York: Wiley.
- McLachlan, G. J., & Rathnayake, S. (2014). On the number of components in a Gaussian mixture model. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(5), 341–355.
- Meng, X.-L., & Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2), 267–278.
- Nierenberg, D. W., Stukel, T. A., Baron, J. A., Dain, B. J., Greenberg, E. R., & Group, S. C. P. S. (1989). Determinants of plasma levels of beta-carotene and retinol. *American Journal of Epidemiology*, 130(3), 511–521.
- Reimann, C., Äyräs, M., Chekushin, V., Bogatyrev, I., Boyd, R., de Caritat, P., Volden, T., et al. (1998). *Environmental geochemical atlas of the Central Barents Region*. Trondheim, Norway: Geological Survey of Norway.
- Reimann, C., Filzmoser, P., Garrett, R., & Dutter, R. (2011). *Statistical data analysis explained: Applied environmental statistics with R*. Chichester, UK: John Wiley & Sons.

- Richardson, S., & Green, P. J. (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.
- Roeder, K., & Wasserman, L. (1997). Practical Bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association*, 92(439), 894–902.
- Scarrott, C., Hu, Y., Akbar, A., & Canterbury, U. (2018). *evmix: Extreme value mixture modelling, threshold estimation and boundary corrected kernel density estimation*. R package version 2.10.
- Schuster, E. F. (1985). Incorporating support constraints into nonparametric estimators of densities. *Communications in Statistics—Theory and Methods*, 14(5), 1123–1136.
- Schwartz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6, 31–38.
- Scott, D. W. (2009). *Multivariate density estimation: Theory, practice, and visualization* (2nd ed.). New Jersey: John Wiley & Sons.
- Scrucca, L., Fop, M., Murphy, T. B., & Raftery, A. E. (2016). mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1), 205–233.
- Scrucca, L., & Raftery, A. E. (2015). Improved initialisation of model-based clustering using Gaussian hierarchical partitions. *Advances in Data Analysis and Classification*, 4(9), 447–460.
- Simonoff, J. (1996). *Smoothing methods in statistics*. New York: Springer.
- Stone, C. J., Hansen, M. H., Kooperberg, C., & Truong, Y. K. (1997). Polynomial splines and their tensor products in extended linear modeling. *Annals of Statistics*, 25(4), 1371–1470.
- Titterington, D. M., Smith, A. F., & Makov, U. E. (1985). *Statistical analysis of finite mixture distributions*. Chichester, New York: Wiley.
- Velilla, S. (1993). A note on the multivariate Box-Cox transformations to normality. *Statistics and Probability Letters*, 17, 441–451.
- Wand, M. P., Marron, J. S., & Ruppert, D. (1991). Transformations in density estimation. *Journal of the American Statistical Association*, 86(414), 343–353.
- Wu, C. J. (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1), 95–103.
- Yeo, I.-K., & Johnson, R. A. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4), 954–959.
- Young, D., Benaglia, T., Chauveau, D., & Hunter, D. (2017). *mixtools: Tools for Analyzing Finite Mixture Models*. R package version 1.1.0.
- Zhu, X., & Melnykov, V. (2018). Manly transformation in finite mixture modeling. *Computational Statistics & Data Analysis*, 121, 190–208.

SUPPORTING INFORMATION

Additional supporting information including source code to reproduce the results may be found online in the Supporting Information section at the end of the article.

How to cite this article: Scrucca L. A transformation-based approach to Gaussian mixture density estimation for bounded data. *Biometrical Journal*. 2019;61:873–888. <https://doi.org/10.1002/bimj.201800174>

Alessandro Casa

Material list:

Casa A. (2024) Sparse model-based clustering of three-way data via lasso-type penalties. WG Slides.

Cappozzo A., Casa A., Fop M. (2023) Sparse model-based clustering of three-way data via lasso-type penalties. arXiv:2307.10673.

Sparse model-based clustering of three-way data via lasso-type penalties

Working Group on Model-Based Clustering

 Alessandro Casa

Joint work with: A. Cappozzo & M. Fop

 Dipartimento di Scienze Economiche
Università degli Studi di Bergamo

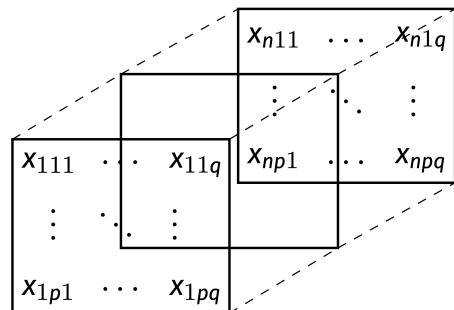
 alessandro.casa@unibg.it

Bertinoro, 26th July 2024



➤ Introduction | Three-way data

- Three-way or **matrix-variate** data are increasingly common in different applications
- Arising for example when multiple variables are measured over time for a set of units



As stated in Anderlucci and Viroli (2015) we need tools able to “account simultaneously for the three goals of the analysis which arise from the **three layers of the data structure**: heterogeneous units, correlated occasions and dependent variables”

➤ Introduction | Three-way clustering

Idea

Clustering tools, searching for groups of observed matrices, may be used to reduce these complexities, provide parsimonious summaries and highlights data patterns

Several approaches have been explored, some of them collapsing the structure into a two-way matrix

- **Distance-based**, based on least-square approaches, not requiring distributional assumptions
- **Density-based**, links clusters to some specific features of the density underlying the data
 - Model-based, mixture modelling resorting to available matrix-variate distributions
 - Modal, relying on nonparametric estimators of matrix-variate distributions and searching for modes of the estimated density

2/24

➤ Model-based matrix-variate clustering

- **Matrix Gaussian mixture models (MGMM)** resorts to matrix Gaussian distribution to provide model-based three-way data generalization
- Let $\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\}$ be a set of n matrices, with $\mathbf{X}_i \in \mathbb{R}^{p \times q}$. According to MGMM we have that

$$f(\mathbf{X}_i; \Theta) = \sum_{k=1}^K \tau_k \phi_{p \times q}(\mathbf{X}_i; \mathbf{M}_k, \boldsymbol{\Omega}_k, \boldsymbol{\Gamma}_k)$$

- $\phi_{p \times q}(\cdot)$, $p \times q$ matrix Gaussian density
- τ_k , mixing proportions
- $\mathbf{M}_k \in \mathbb{R}^{p \times q}$, k -th component mean matrix
- $\boldsymbol{\Omega}_k \in \mathbb{R}^{p \times p}$ and $\boldsymbol{\Gamma}_k \in \mathbb{R}^{q \times q}$, k -th component rows and columns precision matrices
- $\Theta = \{\tau_k, \mathbf{M}_k, \boldsymbol{\Omega}_k, \boldsymbol{\Gamma}_k\}_{k=1}^K$ denotes the whole set of parameters to be estimated

3/24

➤ Estimation and extensions

- Θ usually estimated by means of tailored **EM-algorithm** with estimates available in closed-form
 - Link between matrix Gaussian and vectorized Gaussian forces constraints on trace or determinant of Ω_k or Γ_k
$$\mathbf{X} \sim m\mathcal{N}_{p \times q}(\mathbf{M}, \boldsymbol{\Omega}, \boldsymbol{\Gamma}) \iff \text{vec}(\mathbf{X}) \sim \mathcal{N}_{pq}(\text{vec}(\mathbf{M}), \boldsymbol{\Omega} \otimes \boldsymbol{\Gamma}),$$
- Interestingly, Viroli (2011) proposed a birth and death MCMC algorithm to estimate the model in a **Bayesian framework**
- **Several extensions** have been proposed to deal with
 - Skewed or heavy-tailed data, contamination, data-transformations and count data among the others
 - Tensor-variate higher order data
 - Regression framework

4/24

➤ Overparameterization | Why?

Bouveyron and Brunet-Saumard (2014) noted that
“[...] classical model-based clustering techniques show a disappointing behavior in high-dimensional spaces. This is mainly due to the fact that model-based clustering methods are dramatically over-parametrized.”

- In standard model-based clustering the number of parameters to estimate scales quadratically with the number of variables
- What about **matrix-variate**?

$$|\Theta| = (K - 1) + Kpq + Kp(p - 1)/2 + Kq(q - 1)/2$$

Cardinality of Θ scales quadratically with **both p and q**

- Difficult to use even with moderate dimensions
- Interpretation of the results is quite tricky

5/24

➤ Overparameterization | Two-way solutions

(Non-exhaustive) taxonomy of the possible solutions proposed in the two-way framework

- [Constrained parameterizations](#)
eigen-decomposition or specific parameterizations of the component covariance matrices
- [Variable selection](#)
identification of most relevant variables for clustering purposes, either via model-selection or via sparse estimation
- [Sparsity-inducing procedures](#)
penalized estimation to reduce the number of non-zero parameters

6/24

➤ Overparameterization | Three-way extensions

Approaches in the matrix-variate framework have been developed coherently with the introduced taxonomy

- [Sarkar et al. \(2020\) - Constrained parameterizations](#)
extend two-way constrained parameterization approaches to the matrix-variate scenario by considering eigendecomposition of both row and column component covariance matrices
- [Wang & Melnykov \(2020\) - Variable selection](#)
introduce a stepwise variable selection procedure, accounting for redundant and irrelevant variables, with the best model being selected according to information criteria

7/24

› Overparameterization | What's missing?

The mentioned approaches have two main [drawbacks](#)

- Computationally intensive
- Potentially rigid, as they do not allow varying cluster-dependent association patterns and mean matrices structures

Idea & starting point

Assume that all the matrices in Θ possess their own cluster-dependent degrees of sparsity

Aim

Extend the framework of sparse and penalized mixture models to matrix-variate data

8/24

› Sparse matrix mixture models

We introduce a sparse matrix Gaussian mixture model. Estimation is carried out by maximizing the following [penalized log-likelihood](#)

$$\ell_p(\Theta; \mathbf{X}) = \sum_{i=1}^n \log \left\{ \sum_{k=1}^K \tau_k \phi_{p \times q}(\mathbf{X}_i; \mathbf{M}_k, \boldsymbol{\Omega}_k, \boldsymbol{\Gamma}_k) \right\} - p_{\lambda}(\mathbf{M}_k, \boldsymbol{\Omega}_k, \boldsymbol{\Gamma}_k)$$

where

- $p_{\lambda}(\mathbf{M}_k, \boldsymbol{\Omega}_k, \boldsymbol{\Gamma}_k)$ is a [penalty term](#) to be defined
- $\lambda = (\lambda_M, \lambda_\Omega, \lambda_\Gamma)$ is a vector of penalty hyperparameters, to be selected to [tune the intensity](#) of the penalty

9/24

› Penalty specification | Precision matrices

Different choices can be taken when specifying $p_{\lambda}(\mathbf{M}_k, \boldsymbol{\Omega}_k, \boldsymbol{\Gamma}_k)$

In this work we consider

$$p_{\lambda}(\mathbf{M}_k, \boldsymbol{\Omega}_k, \boldsymbol{\Gamma}_k) = p_{\lambda_M}(\mathbf{M}_k) + \sum_{k=1}^K \lambda_{\Omega} \|\mathbf{P}_{\Omega} \odot \boldsymbol{\Omega}_k\|_1 + \sum_{k=1}^K \lambda_{\Gamma} \|\mathbf{P}_{\Gamma} \odot \boldsymbol{\Gamma}_k\|_1$$

The second and third terms impose **graphical lasso** penalties on row and column precision matrices with $\|A\|_1 = \sum_{jh} |A_{jh}|$

Advantages

- Reduce number of non-zero parameters
- Enhance interpretation, with connection to conditional dependence patterns
- Allows for component-dependent sparsity patterns

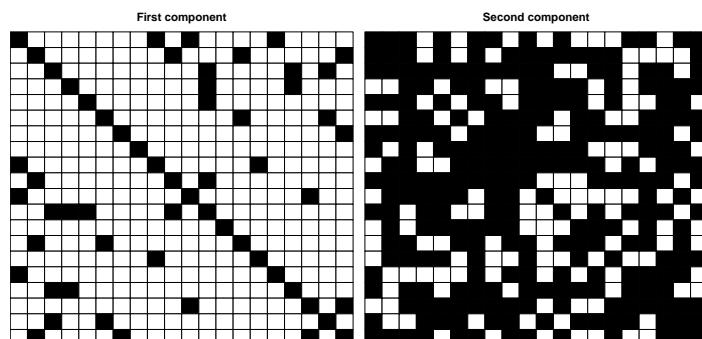
10/24

› Penalty specification | Additional thoughts

Inclusion of matrices \mathbf{P}_{Ω} and \mathbf{P}_{Γ} provides **higher flexibility** as clever specifications can encode peculiar structures and prior belief

Casa et al. (2022) show that **component-dependent weighting matrices** can be used to scale the penalty effect to account for component-specific sparsity intensities. We could consider for example

$$p_{\lambda_{\Omega}}(\boldsymbol{\Omega}_k) = \sum_{k=1}^K \lambda_{\Omega} \|\mathbf{P}_{\Omega_k} \odot \boldsymbol{\Omega}_k\|_1$$



11/24

› Penalty specification | Mean matrices

In this work we consider **group lasso penalty** on \mathbf{M}_k

$$p_{\lambda_M}(\mathbf{M}_k) = \sum_{k=1}^K \lambda_M \sum_{r=1}^p \|\mathbf{m}_{r,k}\|_2$$

where $\mathbf{m}_{r,k}$ is the r -th row of \mathbf{M}_k and $\|\cdot\|_2$ the L_2 -norm

Additional thoughts on group lasso

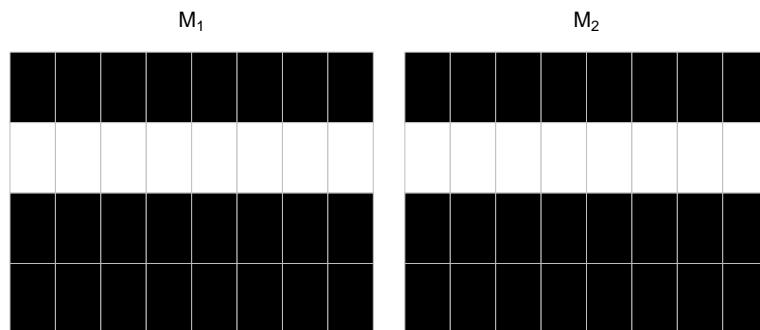
- Grouping structure is given by the rows of the mean matrices
- In the estimation, either the whole row $\mathbf{m}_{r,k}$ is estimated to be zero or its elements are shrunk towards (but not equal to) zero by an amount depending on λ_M

12/24

› Penalty specification | More on group lasso

Idea

If data are centered, $\hat{\mathbf{m}}_{r,k} = 0$ for all k the r -th variable is not useful to discriminate clusters mean-levels across q time instants



Word of caution

It allows to assess variable importance from a mean-perspective. The r -th variable might still contain cluster-discriminating information in its dependencies with other variables

13/24

➤ Penalty specification | Note on alternatives

An alternative choice for $p_{\lambda_M}(\mathbf{M}_k)$ could be

$$p_{\lambda_M}(\mathbf{M}_k) = \sum_{k=1}^K \lambda_M \|\mathbf{P}_M \odot \mathbf{M}_k\|_1$$

with \mathbf{P}_M a weighting matrix as before

Idea

- Cell-wise L_1 -penalty offers a less-structured way to induce sparsity
- Harder to provide indications about feature selection/importance
- We generalize Heo & Baek (2021) which built a similar model but assuming diagonal Ω_k and Γ_k , hindering clusters recovery

14/24

➤ Model estimation | Set-up

For fixed K and λ , EM-algorithm is employed to maximize the [penalized complete log-likelihood](#)

$$\begin{aligned} \ell_c(\Theta; \mathbf{X}, \mathbf{Z}) \propto & \sum_{i,k} z_{ik} \left[\log \tau_k + \frac{q}{2} \log |\Omega_k| + \frac{p}{2} \log |\Gamma_k| + \right. \\ & \left. - \frac{1}{2} \text{tr} \{ \Omega_k (\mathbf{X}_i - \mathbf{M}_k) \Gamma_k (\mathbf{X}_i - \mathbf{M}_k)' \} \right] - p_\lambda(\mathbf{M}_k, \Omega_k, \Gamma_k) \end{aligned}$$

[Initialization of the EM-algorithm](#)

Connections between MGMM and GMM allows to resort to vectorization and standard two-way initialization techniques such as model-based hierarchical agglomerative clustering (Scrucca & Raftery, 2015)

15/24

➤ Model estimation | EM steps

E-step

standard updates for a posteriori probabilities \hat{z}_{ik}

M-step

partial optimization strategy is required, alternating between four steps

- closed-form updates for $\hat{\tau}_k$
- updates of $\hat{\mathbf{M}}_k$ depend on the considered penalty
 - group lasso, proximal gradient descent algorithm
 - lasso, cell-wise coordinate ascent algorithm
- coordinate descent graphical lasso algorithm for $\hat{\Omega}_k$
- coordinate descent graphical lasso algorithm for $\hat{\Gamma}_k$

16/24

➤ Model selection

- We need a way to choose the number of cluster K and to tune the hyperparameters $\lambda_{\mathbf{M}}, \lambda_{\Omega}, \lambda_{\Gamma}$
- Estimate different models for different configurations and select the best one according to the BIC

$$\text{BIC} = 2 \log L(\hat{\Theta}) - d_0 \log(n)$$

where d_0 is the number of non-zero estimated parameters

- Possible alternatives?
 - Stochastic optimization algorithms
 - Conditional searches schemes
 - E-MS algorithm

17/24

Some results | Crime data

- **crime data**

available in the package MatTransMix (Zhu et al., 2022), analyzed in Melnykov & Zhu (2019)

- Crime frequency and rate records between 2000 and 2012 ($q = 13$) for $n = 236$ cities in the US.

Measured variables ($p = 7$)

Violent crimes

- murder, rape, robbery, aggravated assault

Property crimes

- motor vehicle theft, burglary, larceny-theft

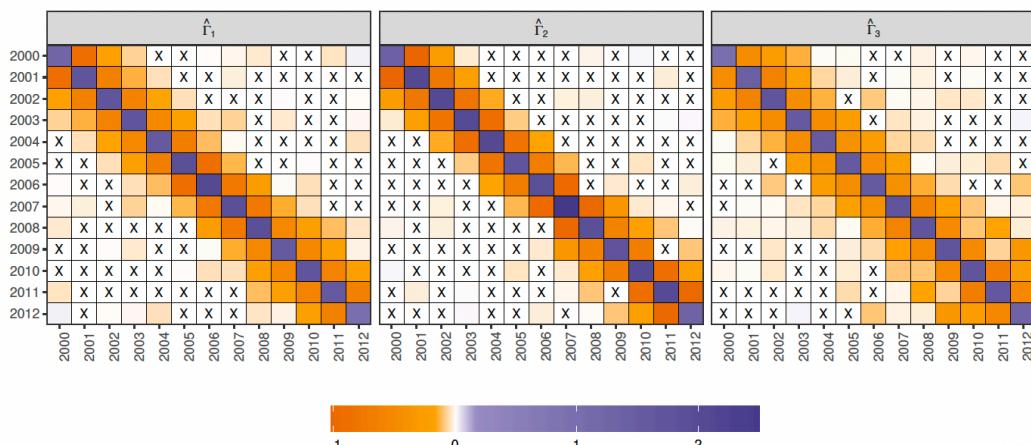
Aim

Cluster cities with similar crime trends and identify which crime types show difference in time patterns across clusters

18/24

Some results | Crime data

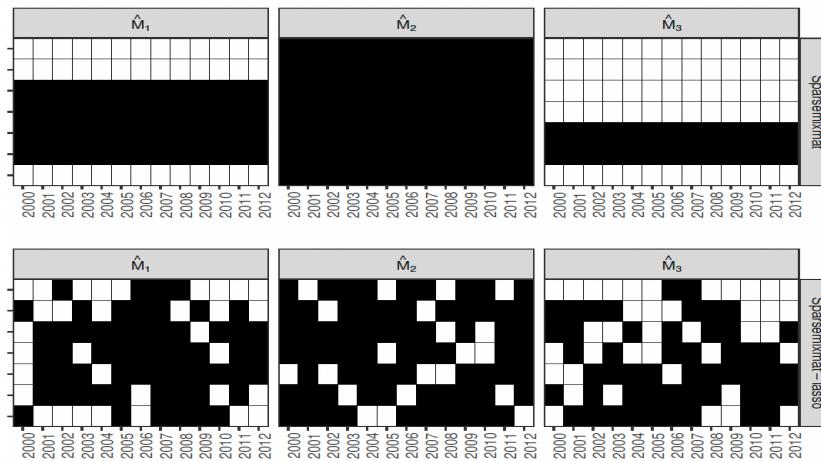
- BIC selects $K = 3$ and $\lambda_{\Omega} = 0$ implying non-sparse conditional dependencies among crime types across clusters
- $\lambda_{\Gamma} \neq 0$ implies certain degree of sparsity for Γ_k which show a **banded structure**, in line with expectations



19/24

➤ Some results | Crime data

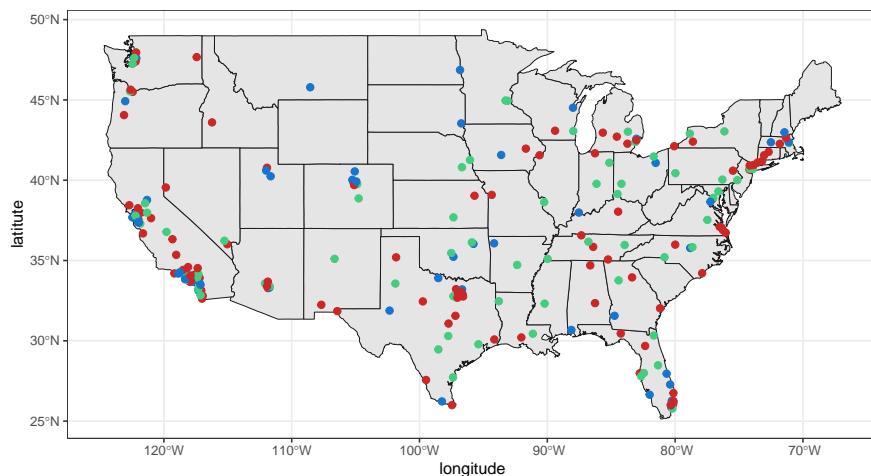
- A closer look to the $K = 3$ clusters allow to identify
 - cluster 1, larger population, medium crime rates
 - cluster 2, medium population, highest crime rates
 - cluster 3, medium population, safest cities
- No overall variable selection but more structured sparsity and interpretable indications about feature importance



20/24

➤ Some results | Crime data

- Indications closed to the ones in Melnykov & Zhu (2019)
 - Easter USA more dangerous
 - Something along Mississippi belt
 - Large cities are more dangerous than their surroundings



21/24

➤ Conclusion and Discussion

We propose a modeling framework alleviating limitations of three-way model-based clustering

- Easier interpretation of dependence patterns
- Flexible way to induce parsimony
- Indications about variable importance

What's next?

- Extend the approach to other reasonable penalties such as sparse group lasso and fused lasso
- Evaluate alternative model selection strategies
- Explore if the proposal can be used with alternative choices for the component densities

22/24

➤ Some references

Cappozzo, A., Casa, A., & Fop, M. (2024+).
Sparse model-based clustering of three-way data via lasso-type penalties.
arXiv preprint arXiv:2307.10673.

- Bouveyron, C. & Brunet-Saumard, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis*, 71:52–78
- Casa, A., Cappozzo, A., & Fop, M. (2022). Group-wise shrinkage estimation in penalized model-based clustering. *Journal of Classification*, 39(3), 648–674
- Heo, J. & Baek, J. (2021). A penalized matrix normal mixture model for clustering matrix data. *Entropy*, 23(10):1249
- Leng, C. & Tang, C. (2012). Sparse matrix graphical models. *Journal of the American Statistical Association*, 107(499): 1187–1200

23/24

Some references

- Melnykov, V. & Zhu, X. (2019). Studying crime trends in the USA over the years 2000-2012. *Advances in Data Analysis and Classification*, 13(1): 325–341
- Sarkar, S., Zhu, X., Melnykov, V., and Ingrassia, S. (2020). On parsimonious models for modeling matrix data. *Computational Statistics & Data Analysis*, 142:106822.
- Scrucca, L. & Raftery, A.E. (2015). Improved initialisation of model-based clustering using Gaussian hierarchical partitions. *Advances in Data Analysis and Classification*, 9(4): 447–460
- Viroli, C. (2011a). Finite mixtures of matrix normal distributions for classifying three-way data. *Statistics and Computing*, 21(4):511–522
- Viroli, C. (2011b). Model based clustering for three-way data structures. *Bayesian Analysis*, 21(4):511–522
- Wang, Y. and Melnykov, V. (2020). On variable selection in matrix mixture modelling. *Stat*, 9(1):e278
- Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Societies Series B*, 68(1):49–67

Sparse model-based clustering of three-way data via lasso-type penalties

Andrea Cappozzo*

MOX, Department of Mathematics, Politecnico di Milano

Alessandro Casa*

Faculty of Economics and Management, Free University of Bozen-Bolzano

Michael Fop

School of Mathematics & Statistics, University College Dublin

July 21, 2023

Abstract

Mixtures of matrix Gaussian distributions provide a probabilistic framework for clustering continuous matrix-variate data, which are becoming increasingly prevalent in various fields. Despite its widespread adoption and successful application, this approach suffers from over-parameterization issues, making it less suitable even for matrix-variate data of moderate size. To overcome this drawback, we introduce a sparse model-based clustering approach for three-way data. Our approach assumes that the matrix mixture parameters are sparse and have different degree of sparsity across clusters, allowing to induce parsimony in a flexible manner. Estimation of the model relies on the maximization of a penalized likelihood, with specifically tailored group and graphical lasso penalties. These penalties enable the selection of the most informative features for clustering three-way data where variables are recorded over multiple occasions and allow to capture cluster-specific association structures. The proposed methodology is tested extensively on synthetic data and its validity is demonstrated in application to time-dependent crime patterns in different US cities.

Keywords: Group lasso, Matrix-variate data, Model-based clustering, Penalized likelihood, Sparse estimation

1 Introduction

Matrix-variate data, where a matrix is observed for each statistical unit, are becoming more common in a large number of applications and data analysis routines. This data

*These authors contributed equally to this work

structure is often referred to as *three-way* and characterized by the presence of three different layers or modes, namely the units, the variables and the occasions. These data are nowadays often occurring in applications such as multivariate time-dependent analysis ([Anderlucci and Viroli, 2015](#)), the analysis of crime patterns ([Melnykov and Zhu, 2019](#)), basketball analytics ([Yin et al., 2023](#)), the analysis of export trade networks ([Melnykov et al., 2021](#)), image and brain scan data analysis ([Gao et al., 2021; Liu et al., 2022](#)). In spite of their potential in terms of informative content, matrix-variate data introduce several challenges which need to be dealt with in the modeling process. In fact, each of the three different layers induce specific peculiarities in terms of intricate dependency structures.

In this landscape, clustering is often of interest to reduce the aforementioned complexities by proposing parsimonious summaries of the data and highlighting their most relevant patterns. To this extent, both distance-based ([Vichi, 1999; Vichi et al., 2007](#)) and nonparametric techniques ([Ferraccioli and Menardi, 2023](#)) have been proposed. Nevertheless, parametric or model-based approaches are undoubtedly the ones that have received the most attention: taking steps from [Basford and McLachlan \(1985\)](#) and building on mixtures of matrix-variate Gaussian distributions, the seminal papers by [Viroli \(2011a,b\)](#) have paved the way for a new and lively stream of research. Recently, several flexible approaches have been proposed to deal with data of different nature. These approaches considered alternative distributional assumptions for skewed data ([Chen and Gupta, 2005; Melnykov and Zhu, 2018; Gallaugher and McNicholas, 2018](#)), transformations (see, among others, [Chen and Gupta, 2005; Melnykov and Zhu, 2018; Gallaugher and McNicholas, 2018; Tomarchio et al., 2020, 2022; Tomarchio, 2022](#)) and alternative models for count data ([Silva et al., 2023; Subedi, 2023](#)).

Despite being practically useful, matrix-variate model-based clustering faces significant limitations in high-dimensional settings. These limitations are particularly pronounced in the three-way framework where the tendency to over-parameterization, inherited from the vector-valued setting ([Bouveyron and Brunet-Saumard, 2014](#)), becomes even more challenging. In fact, in the context of the matrix Gaussian distribution, two covariance matrices are employed for each component to accommodate the data structure. Consequently,

when dense parameterizations are assumed for these matrices, the number of parameters to be estimated grows quadratically with both the number of rows and columns. This undermines the practical utility of the approach, even when a moderate number of variables and/or occasions are observed.

To address these limitations, in this work we introduce a novel approach where each parameter involved in the specification of the matrix Gaussian mixture model has its own cluster-specific degree of sparsity. This greatly increases the flexibility of the model, leads to a parsimonious modeling framework, and provides more interpretable insights regarding the clustering partition. The approach relies on the maximization of a penalized likelihood which automatically enforces sparsity. More specifically, we impose a graphical lasso penalty on rows and columns precision matrices, promoting a reduction in the number of non-zero parameters while facilitating interpretation in terms of conditional dependencies, thanks to the connection with Gaussian graphical models. Additionally, we impose a group lasso penalty on the rows of the component mean matrices. In the common scenario where variables are observed over time for a set of statistical units, this penalization scheme allows to perform automatic variable selection in a three-way model-based clustering framework. As a supplementary contribution, we briefly generalize the applicability of the work by [Heo and Baek \(2021\)](#), where they consider a lasso-type entry-wise penalty for the elements of the mean matrices.

The remainder of the paper is structured as follows. Section 2 overviews model-based clustering of matrix-variate data, with a specific focus on the issues arising in high-dimensional spaces. In Section 3, our proposal is introduced and motivated, alongside with the description of the associated estimation and model selection methods. In Section 4 and 5, the performance of the proposed framework is tested on synthetic and real data, respectively. Conclusions and considerations about further improvements and future research directions end the paper in Section 6.

2 Model-based matrix-variate clustering

2.1 Mixture of matrix normal distributions

Model-based clustering (Fraley and Raftery, 2002; Bouveyron et al., 2019) assumes that the data are generated by a finite mixture distribution, which describes the presence of heterogeneous sub-populations. In this context, typically maximum likelihood estimation is usually implemented by means of the EM algorithm (Dempster et al., 1977), resorting to a data augmentation scheme where the latent group indicator variables are treated as missing data. Operationally, once the model is fitted, a partition is obtained by assuming a one-to-one correspondence between the groups and the mixture components, and assigning the i -th observation to a given cluster according to the maximum a posteriori (MAP) rule (see Fraley and Raftery, 2002; Bouveyron et al., 2019, for a detailed treatment).

When dealing with standard continuous vector-variate data, where a number of variables are measured for a set of units, it is routine to assume that the mixture components correspond to multivariate Gaussian distributions (Fraley and Raftery, 2002). Nonetheless, nowadays it is becoming increasingly common to encounter three-way data structures, where multiple variables are measured over different occasions. This additional layer (or mode) introduces new modeling challenges that need to be taken into account when clustering samples is the final goal. Indeed, as noted by Anderlucci and Viroli (2015), models have to “*account simultaneously for three goals of the analysis, which arise from the three layers of the data structure; heterogeneous units, correlated occasions and dependent variables*”. Matrix Gaussian mixture models have originally been proposed by Viroli (2011a,b) with the aim of simultaneously accounting for these sources of complexity.

Formally, let $\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\}$, be a sample of $p \times q$ matrices, with $\mathbf{X}_i \in \mathbb{R}^{p \times q}$, $i = 1, \dots, n$. While in general the dimensions can be relate to any type of measurement, in the following we assume that p variables are observed in q different occasions, as appropriate for most applications. The natural GMM extension for model-based clustering of three-way data is given by the matrix Gaussian mixture model (MGMM), expressed as follows:

$$f(\mathbf{X}_i; \Theta) = \sum_{k=1}^K \tau_k \phi_{p \times q}(\mathbf{X}_i; \mathbf{M}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\Psi}_k), \quad (1)$$

where τ_k 's are the mixing proportions with $\tau_k > 0, \forall k = 1, \dots, K$ and $\sum_{k=1}^K \tau_k = 1$; K is the number of mixture components, while Θ denotes the collection of all mixture parameters. Here, $\phi_{p \times q}(\cdot; \mathbf{M}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\Psi}_k)$ denotes the density of a $p \times q$ matrix normal distribution (Dawid, 1981), reading as

$$\begin{aligned}\phi_{p \times q}(\mathbf{X}_i; \mathbf{M}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\Psi}_k) &= (2\pi)^{-\frac{pq}{2}} |\boldsymbol{\Sigma}_k|^{-\frac{q}{2}} |\boldsymbol{\Psi}_k|^{-\frac{p}{2}} \\ &\quad \exp \left\{ -\frac{1}{2} \text{tr}(\boldsymbol{\Sigma}_k^{-1}(\mathbf{X}_i - \mathbf{M}_k) \boldsymbol{\Psi}_k^{-1}(\mathbf{X}_i - \mathbf{M}_k)') \right\},\end{aligned}$$

where \mathbf{M}_k is the $p \times q$ mean matrix of the k -th component, and $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\Psi}_k$ are the component rows and columns covariance matrices, with dimensions $p \times p$ and $q \times q$, respectively. Coherently with the two-way scenario, the model in (1) can be estimated by means of the EM-algorithm, see for example Viroli (2011a); Glanz and Carvalho (2018); Gao et al. (2021). Alternatively, the model can also be formulated and estimated under a Bayesian framework, as for example Viroli (2011a); Yin et al. (2023).

An alternative specification of the matrix-variate Gaussian distribution may be given, since the following relation holds

$$\mathbf{X} \sim m\mathcal{N}_{p \times q}(\mathbf{M}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}) \iff \text{vec}(\mathbf{X}) \sim \mathcal{N}_{pq}(\text{vec}(\mathbf{M}), \boldsymbol{\Psi} \otimes \boldsymbol{\Sigma}), \quad (2)$$

where $\text{vec}(\cdot)$ and \otimes denote respectively the vectorization operator and the Kronecker product; $m\mathcal{N}_{p \times q}$ denotes a matrix Normal distribution of dimensions p and q . From this relation, the matrix-variate Gaussian can be regarded as a direct generalization of the normal distribution to the three-way matrix framework. For more details about the matrix Gaussian distribution, its properties, and its connection to the multivariate normal distribution, readers can refer to Gupta and Nagar (2018). The presence of the Kronecker product in (2) highlights an identifiability issue, since $\boldsymbol{\Psi} \otimes \boldsymbol{\Sigma} = c\boldsymbol{\Psi} \otimes c^{-1}\boldsymbol{\Sigma}$ for any $c \in \mathbb{R}^+$. Enforcing constraints on the trace or on the determinant of one of the two matrices is regarded as a viable solution to solve the problem (see e.g., Viroli, 2012; Melnykov and Zhu, 2018; Glanz and Carvalho, 2018); the latter approach will be considered in the rest of the paper.

2.2 Issues in matrix mixture models for high-dimensional data

Finite mixture models are routinely used for probabilistic cluster analysis. Nonetheless, both in the two-way and the three-way framework, they present a cumbersome issue which is related to their tendency to be over-parameterized even with a moderate number of variables. When dealing with vector-variate data, the cardinality of the parameter space $|\Theta|$ scales quadratically with the number of variables; this problem is even more exacerbated in the matrix-variate scenario, where $|\Theta|$ scales quadratically with both dimensions p and q of the component row and column covariance matrices. In order to deal with this challenge, different approaches have been proposed in the two-way setting (see e.g., [Bouveyron and Brunet-Saumard, 2014](#); [Fop and Murphy, 2018](#), for exhaustive reviews of the topic), which can be grouped into three distinct types: constrained modeling, variable selection, and sparse modeling; a brief overview is provided in [Casa et al. \(2022\)](#).

In line with this classification, recent efforts have been devoted to addressing the issue of over-parameterization within the framework of matrix mixture modeling. Specifically, some of the existing approaches either adopt parsimonious parametrizations, or implement variable selection to discard irrelevant variables and reduce the number of parameters. In [Sarkar et al. \(2020\)](#), the authors extend the family of covariance eigendecomposition models considered for vector-valued data ([Banfield and Raftery, 1993](#); [Celeux and Govaert, 1995](#)) to the matrix-variate scenario. They introduced a collection of 98 constrained models and further enhanced parsimony by proposing an additive formulation for the mean matrices, resulting in a family of 196 matrix mixture models. On the other hand, [Wang and Melnykov \(2020\)](#) propose a variable selection approach where the work by [Maugis et al. \(2009\)](#) is extended to the matrix-variate framework. A stepwise variable selection procedure is proposed, which alternates variable inclusion and exclusion steps, where the resulting models are compared by means of the Bayesian Information Criterion (BIC, [Schwarz, 1978](#)). These two approaches present some relevant drawbacks: they can be computationally intensive, since involve fitting and comparing of a large number of models, and they implement a rigid way to induce parsimony, not allowing the association structures among the variables and the structure of the mean matrices to vary from one cluster to the other.

For the above reasons, in this work we take a different perspective, based on the formulation of a sparse matrix mixture model, by extending the framework of sparse and penalized mixture models (Fop and Murphy, 2018; Fop et al., 2019; Casa et al., 2022, among others) to matrix-variate data. Building primarily upon the literature on sparse matrix graphical models (Leng and Tang, 2012; Chen and Liu, 2019, for example) and sparse model-based clustering (Zhou et al., 2009), sparse approaches for matrix-variate data clustering have been recently introduced and are gaining increasing attention. In application to brain imaging data, Gao et al. (2021) develop a penalized Gaussian matrix mixture model, where penalty functions on the entries of the component mean matrices are introduced to shrink the mean parameters. The method is shown to recover the low rank mean signal, however, it does not allow a flexible modeling of the association structure across the clusters. On a similar vein, Liu et al. (2022) presents a multi-step approach for clustering and sparse correlation estimation in application to functional magnetic resonance imaging data. Here, in contrast to Gao et al. (2021) and motivated by the application, the authors propose an optimization framework that focuses on recovering the different association structures across the clusters, but covariance parameters rather than the means are employed to cluster the units, which could be a limitation if clusters are well separated in terms of mean signals. Additionally, the authors remark that the method suffer from the need to pre-specify the number of clusters beforehand and the lack of a principled method for selecting this number. In Heo and Baek (2021), the authors describe a penalized matrix normal mixture model for clustering that employs penalty functions on both means and covariance matrix parameters to induce sparse estimation. However, this approach relies on implicit restrictive independence assumptions during estimation, posing potential problems. Moreover, the specific formulation of the penalty functions on the mean parameters does not allow for an effective variable selection in the context of three-way data where variables are measured over multiple occasions.

In what follows, we propose a sparse matrix Gaussian mixture model where we overcome the drawbacks of the aforementioned frameworks for three-way data clustering. Our proposed approach offers several advantages: it allows clusters to be characterized by different

association structures, it accommodates estimation of sparse component matrix means and inverse covariance matrices, it uses a principled criterion for model selection, it leverages a computationally efficient framework for estimation based on lasso-type penalties, it allows mean parameters to have different sparse patterns across clusters, and it implements variable selection in a matrix-variate context where the variables are observed over multiple time occasions. The proposed method is based on a maximum penalized likelihood framework, presented in the next section.

3 Sparse matrix mixture models

3.1 Model specification

A sparsity-inducing procedure relying on a penalized likelihood estimation approach is hereafter proposed. Following from the model in (1), we aim to maximize the general penalized log-likelihood below:

$$\ell_P(\boldsymbol{\Theta}; \mathbf{X}) = \sum_{i=1}^n \log \left\{ \sum_{k=1}^K \tau_k \phi_{p \times q}(\mathbf{X}_i; \mathbf{M}_k, \boldsymbol{\Omega}_k, \boldsymbol{\Gamma}_k) \right\} - p_{\boldsymbol{\lambda}}(\mathbf{M}_k, \boldsymbol{\Omega}_k, \boldsymbol{\Gamma}_k), \quad (3)$$

where the first addend represents the standard MGMM log-likelihood and $p_{\boldsymbol{\lambda}}(\mathbf{M}_k, \boldsymbol{\Omega}_k, \boldsymbol{\Gamma}_k)$ is a penalty term depending on a set of shrinkage factors generally denoted with $\boldsymbol{\lambda}$, while $\boldsymbol{\Omega}_k = \boldsymbol{\Sigma}_k^{-1}$ and $\boldsymbol{\Gamma}_k = \boldsymbol{\Psi}_k^{-1}$, for $k = 1, \dots, K$ are the rows and column precision matrices, respectively. The collection of parameters is $\boldsymbol{\Theta} = \{\tau_k, \mathbf{M}_k, \boldsymbol{\Omega}_k, \boldsymbol{\Gamma}_k\}_{k=1}^K$. The choice to parameterize the MGMM density in terms of precision matrices is motivated by their relation to Gaussian graphical models and their interpretation in terms of conditional dependencies (Whittaker, 1990; Leng and Tang, 2012). However, other options could be considered, and a discussion is reported in Section 6.

Different routes can be taken when specifying the penalty $p_{\boldsymbol{\lambda}}(\mathbf{M}_k, \boldsymbol{\Omega}_k, \boldsymbol{\Gamma}_k)$ to obtain sparse estimates of the mixture component matrix parameters; readers may refer to the recent book by Hastie et al. (2019) for a detailed discussion. In this work, we consider the following penalty term

$$p_{\boldsymbol{\lambda}}(\mathbf{M}_k, \boldsymbol{\Omega}_k, \boldsymbol{\Gamma}_k) = \sum_{k=1}^K \lambda_1 \sum_{r=1}^p \|\mathbf{m}_{r \cdot k}\|_2 + \sum_{k=1}^K \lambda_2 \|\mathbf{P}_2 * \boldsymbol{\Omega}_k\|_1 + \sum_{k=1}^K \lambda_3 \|\mathbf{P}_3 * \boldsymbol{\Gamma}_k\|_1, \quad (4)$$

where $\mathbf{m}_{r,k}$ is the r -th row of matrix \mathbf{M}_k , while $\|\cdot\|_1$ and $\|\cdot\|_2$ are the L_1 and the L_2 -norm respectively, with $\|A\|_1 = \sum_{jh} |A_{jh}|$. Moreover, $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ is a vector of positive shrinkage hyper-parameters controlling the strength of the penalization. Lastly, \mathbf{P}_2 and \mathbf{P}_3 are matrices with non-negative entries, and $*$ denotes the element-wise product.

The first term in (4) corresponds to a group lasso penalty (Yuan and Lin, 2006), imposed on the rows of the K mean matrices \mathbf{M}_k . Group lasso aims to simultaneously shrink to zero a set of grouped parameters, and it has been mainly used in a regression framework, where some covariates might be structurally connected (see Ch.4.3 in Hastie et al., 2019, and references therein). In this work, we generalize this penalty to an unsupervised setting with matrix-variate mean parameters. Here, we consider the parameters as being grouped according to the rows of \mathbf{M}_k . Therefore, for a given k , either the whole row $\mathbf{m}_{r,k} = (m_{r1,k}, \dots, m_{rq,k})$ is estimated to be zero, or else its elements are shrunk towards zero (but not resulting equal to zero) by an amount depending on λ_1 . This penalization scheme is adopted to perform variable selection in model-based clustering of three-way data in the common scenario when p variables are observed over q time instants or occasions. Indeed, when $\mathbf{m}_{r,k} = \mathbf{0}$ for all k , the r -th row of \mathbf{M}_k is constant across all occasions and clusters. Therefore, it is not useful for discriminating the mean levels of the clusters. Even when $\mathbf{m}_{r,k} = \mathbf{m}_{r,h} = \mathbf{0}$ for some components k and h , the r -th variable does not contain discriminative information to separate them, resulting in overlap along that dimension. Note that the proposed approach can be seen as the adaptation of the support union recovery methodology (Obozinski et al., 2009, 2011) to the matrix-variate model-based clustering context.

With the second and the third term in (4), we impose a graphical lasso penalty (see Banerjee et al., 2008; Friedman et al., 2008; Witten et al., 2011) on the group-specific precision matrices. This represents an extension of the work by Leng and Tang (2012) to the framework of mixture models. By shrinking to zero some parameters, the penalty terms allow to alleviate the problems outlined in Section 2.2 when dealing with high-dimensional data, providing a parsimonious and flexible model for the association structure between row and column variables across clusters. The resulting sparse representation of $\boldsymbol{\Omega}_k$ and $\boldsymbol{\Gamma}_k$,

for $k = 1, \dots, K$, provides a convenient interpretation of the dependencies among rows and columns of the observed matrices. In fact, zero entries in the precision matrices imply that the corresponding variables are conditionally independent given the others, following the principles of Gaussian graphical models (Whittaker, 1990). The matrices \mathbf{P}_2 and \mathbf{P}_3 in the graphical lasso penalty term introduce an higher degree of flexibility, since particular specifications allow to introduce prior beliefs regarding the dependencies between the variables. Indications on how to choose these matrices can be found in Bien and Tibshirani (2011). Here the authors suggest to use all-ones matrices, ensuring homogeneous and uninformed penalization for all the precision terms. To prevent shrinkage of the diagonal entries, zeros can be placed on the main diagonal. Alternatively, \mathbf{P}_2 and \mathbf{P}_3 can be defined as adjacency matrices with user-defined patterns, thus allowing the a priori specification of the expected conditional dependence structures. More recently, Casa et al. (2022) introduced a data-driven method for specifying these matrices, which promotes cluster separation within the context of sparse model-based clustering and does not require initial knowledge of the association structure between the variables. In what follows we employ all-one matrices with zero diagonal entries for both \mathbf{P}_2 and \mathbf{P}_3 , as this aspect is not the primary focus of the present paper.

The above-mentioned methodology is based on the assumption that all the parameter matrices in (3), namely $\{\mathbf{M}_k, \boldsymbol{\Omega}_k, \boldsymbol{\Gamma}_k\}_{k=1}^K$, have different component-specific levels of sparsity. This leads to a realistic and flexible modeling framework, where a variable may be relevant only for a subset of clusters, and where the conditional dependence patterns are allowed to vary across groups. Our proposal represents a natural extension to the three-way data scenario of the approach outlined by Zhou et al. (2009). Coherently with their work, the penalty on \mathbf{M}_k aims to perform variable selection. On the other hand, the penalizations on $\boldsymbol{\Omega}_k$ and $\boldsymbol{\Gamma}_k$ are needed in high-dimensional settings, to obtain sparse representations of the matrix mixture precision matrices and to reduce the number of free parameters to be estimated.

3.2 Model estimation

For a fixed number of components K and penalty vector $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ comprehensively, the parameters are estimated by maximizing (3) with respect to Θ . The maximization is carried out by means of a tailored EM algorithm for maximum penalized likelihood estimation (Green, 1990; McLachlan and Krishnan, 2008), where the maximization step (M-step) is comprised of three partial optimization cycles. Let us firstly define the *penalized complete-data log-likelihood* associated with (3) as

$$\ell_C(\Theta; \mathbf{X}) = \sum_{i=1}^n \sum_{k=1}^K z_{ik} \left[\log \tau_k - \frac{pq}{2} \log 2\pi + \frac{q}{2} \log |\Omega_k| + \frac{p}{2} \log |\Gamma_k| + \right. \\ \left. - \frac{1}{2} \text{tr} \{ \Omega_k (\mathbf{X}_i - \mathbf{M}_k) \Gamma_k (\mathbf{X}_i - \mathbf{M}_k)' \} \right] - p_\lambda(\mathbf{M}_k, \Omega_k, \Gamma_k), \quad (5)$$

where z_{ik} is the realization of \mathbf{Z}_{ik} , the latent group membership indicator variable, with $z_{ik} = 1$ if matrix \mathbf{X}_i belongs to the k -th component, and 0 otherwise. The posterior probability of \mathbf{Z}_{ik} is updated at each expectation step (E-step), allowing to obtain the conditional expectation of (5), usually called Q -function, which defines the objective function to be maximized in the M-step. The devised algorithm is described in detail in the next subsections.

3.2.1 Initialization strategy

Initialization plays a crucial role when resorting to EM-type algorithms to perform model estimation. In fact, whenever the likelihood surface has multiple modes, the convergence to the global maximum is not guaranteed and poorly chosen initial values may lead to sub-optimal solutions (McLachlan and Krishnan, 2008). Thanks to the correspondence between GMM and MGMM in Equation (2), initialization strategies developed for vector-variate data samples can be directly employed in the matrix-variate framework. In this regard, after the data have been vectorized, we resort to model-based agglomerative hierarchical clustering (Scrucca and Raftery, 2015). This initialization strategy, already employed in the popular `mclust` software (Scrucca et al., 2016), has been proven effective in partitioning the data into K initial groups.

Once the starting partition is obtained, the first iteration of the M-step requires also initialization of the matrices Ω_k and Γ_k , $k = 1, \dots, K$. For the purpose, identity matrices of dimensions respectively equal to $p \times p$ and $q \times q$ are employed as initial values.

3.2.2 E-step

At iteration t , the estimated a posteriori probabilities $\hat{z}_{ik}^{(t)} = \widehat{\Pr}(\mathbf{Z}_{ik} = 1 | \mathbf{X}_i)$ are updated as follows:

$$\hat{z}_{ik}^{(t)} = \frac{\hat{\tau}_k^{(t-1)} \phi_{p \times q}(\mathbf{X}_i; \hat{\mathbf{M}}_k^{(t-1)}, \hat{\Omega}_k^{(t-1)}, \hat{\Gamma}_k^{(t-1)})}{\sum_{v=1}^K \hat{\tau}_v^{(t-1)} \phi_{p \times q}(\mathbf{X}_i; \hat{\mathbf{M}}_v^{(t-1)}, \hat{\Omega}_v^{(t-1)}, \hat{\Gamma}_v^{(t-1)})}, \quad i = 1, \dots, n,$$

where with the superscript $(t-1)$ we denote parameter estimates obtained in the previous EM iteration.

3.2.3 M-step

The M-step requires the maximization of the (penalized) Q -function, defined as

$$Q(\boldsymbol{\tau}, \{\mathbf{M}_k, \Omega_k, \Gamma_k\}_{k=1}^K) = \sum_{i=1}^n \sum_{k=1}^K \hat{z}_{ik}^{(t)} \left[\log \tau_k + \frac{q}{2} \log |\Omega_k| + \frac{p}{2} \log |\Gamma_k| + \right. \\ \left. - \frac{1}{2} \text{tr} \{ \Omega_k (\mathbf{X}_i - \mathbf{M}_k) \Gamma_k (\mathbf{X}_i - \mathbf{M}_k)' \} \right] + \\ - \sum_{k=1}^K \lambda_1 \sum_{r=1}^p \|\mathbf{m}_{r,k}\|_2 - \sum_{k=1}^K \lambda_2 \|\mathbf{P}_2 * \Omega_k\|_1 - \sum_{k=1}^K \lambda_3 \|\mathbf{P}_3 * \Gamma_k\|_1. \quad (6)$$

The direct maximization of $Q(\cdot)$ with respect to all parameters at once is an unfeasible task, so a partial optimization strategy is required. The closed-form expression for the mixing proportions $\boldsymbol{\tau}$ is readily available:

$$\hat{\tau}_k^{(t)} = \frac{\hat{n}_k^{(t)}}{n}, \quad \hat{n}_k^{(t)} = \sum_{i=1}^n \hat{z}_{ik}^{(t)}, \quad k = 1, \dots, K.$$

Custom procedures are devised for obtaining updates for \mathbf{M}_k , Ω_k , and Γ_k , $k = 1, \dots, K$.

Sparse estimation of the mean matrices \mathbf{M}_k

When maximization of (6) is performed with respect to \mathbf{M}_k , given current estimates of the precision matrices $\hat{\Omega}_k^{(t-1)}$ and $\hat{\Gamma}_k^{(t-1)}$, the Q -function simplifies as follows

$$\begin{aligned} Q_M(\mathbf{M}_k) &= \sum_{i=1}^n \hat{z}_{ik}^{(t)} \left[\text{tr} \left\{ \hat{\Omega}_k^{(t-1)} \mathbf{X}_i \hat{\Gamma}_k^{(t-1)} \mathbf{M}'_k \right\} - \frac{1}{2} \text{tr} \left\{ \hat{\Omega}_k^{(t-1)} \mathbf{M}_k \hat{\Gamma}_k^{(t-1)} \mathbf{M}'_k \right\} \right] - \lambda_1 \sum_{r=1}^p \|\mathbf{m}_{r,k}\|_2 \\ &= \text{tr} \left\{ \hat{\Omega}_k^{(t-1)} \mathbf{S}_M \hat{\Gamma}_k^{(t-1)} \mathbf{M}'_k \right\} - \frac{\hat{n}_k^{(t)}}{2} \text{tr} \left\{ \hat{\Omega}_k^{(t-1)} \mathbf{M}_k \hat{\Gamma}_k^{(t-1)} \mathbf{M}'_k \right\} - \lambda_1 \sum_{r=1}^p \|\mathbf{m}_{r,k}\|_2, \end{aligned} \quad (7)$$

where \mathbf{S}_M is the sum of the matrix-variate observations weighted by $\hat{z}_{ik}^{(t)}$:

$$\mathbf{S}_M = \sum_{i=1}^n \hat{z}_{ik}^{(t)} \mathbf{X}_i.$$

The optimization of (7) with respect to \mathbf{M}_k is solved via a proximal gradient descent algorithm (Parikh and Boyd, 2014). Briefly, proximal gradient methods address a general class of convex problems where the objective function may be decomposed into two terms: the first, generally denoted with $f(\cdot)$, is convex and differentiable, while the other, $g(\cdot)$, may not be everywhere differentiable. On that account, proximal gradient methods, also known as forward backward splitting procedures, can be seen as an extension of gradient descent for optimization problems whose gradient is not available for the entire objective function. In recent years, such approaches gained increasing popularity in the field of statistics and machine learning, as they provide reliable and numerically efficient solutions to regularized models with non-differentiable penalties (Mosci et al., 2010; Klosa et al., 2020). In our case, the maximization of (7) can be recast as follows:

$$\underset{\mathbf{M}_k}{\text{minimize}} f(\mathbf{M}_k) + g(\mathbf{M}_k),$$

where

$$f(\mathbf{M}_k) = \frac{\hat{n}_k^{(t)}}{2} \text{tr} \left\{ \hat{\Omega}_k^{(t-1)} \mathbf{M}_k \hat{\Gamma}_k^{(t-1)} \mathbf{M}'_k \right\} - \text{tr} \left\{ \hat{\Omega}_k^{(t-1)} \mathbf{S}_M \hat{\Gamma}_k^{(t-1)} \mathbf{M}'_k \right\} \quad \text{and} \quad g(\mathbf{M}_k) = \lambda_1 \sum_{r=1}^p \|\mathbf{m}_{r,k}\|_2.$$

Define $\nabla \mathbf{m}_{l,k}$ to be the l -th row, $l = 1, \dots, p$, of

$$\frac{\partial f(\mathbf{M}_k)}{\partial \mathbf{M}_k} = \hat{n}_k^{(t)} \hat{\Omega}_k^{(t-1)} \mathbf{M}_k \hat{\Gamma}_k^{(t-1)} - \hat{\Omega}_k^{(t-1)} \mathbf{S}_M \hat{\Gamma}_k^{(t-1)}, \quad (8)$$

where (8) is the $p \times q$ matrix of first-order partial derivatives of $f(\cdot)$ with respect to \mathbf{M}_k .

A proximal gradient update for the l -th row of matrix \mathbf{M}_k is constructed as follows:

$$\mathbf{b} = \mathbf{m}_{l,k} - \nu \nabla \mathbf{m}_{l,k}, \quad (9a)$$

$$\hat{\mathbf{m}}_{l,k} = \text{prox}_{\nu \lambda_1}(\mathbf{b}), \quad (9b)$$

where ν is a step-size parameter and $\text{prox}_{\nu \lambda_1}(\cdot)$ is the proximity operator of the considered group lasso penalty, namely the row-wise soft thresholding operator:

$$\text{prox}_{\nu \lambda_1}(\mathbf{b}) = \begin{cases} \mathbf{b} \left(1 - \frac{\lambda_1 \nu}{\|\mathbf{b}\|_2}\right) & \text{if } \|\mathbf{b}\|_2 > \lambda_1 \nu, \\ \mathbf{0} & \text{if } \|\mathbf{b}\|_2 \leq \lambda_1 \nu. \end{cases} \quad (10)$$

Iterating equations (9a) and (9b) until convergence sequentially along the p rows retrieves $\hat{\mathbf{M}}_k^{(t)}$, the estimate of the mean matrix mixture parameters for the t -th iteration of the EM algorithm, as the proximal gradient solution to the maximization problem in (7). When λ_1 is sufficiently large, the rows of $\hat{\mathbf{M}}_k^{(t)}$ are set to zero as a result of the proximity operator. Operationally, the weighted sample mean matrix $\sum_{i=1}^n \hat{z}_{ik}^{(t)} \mathbf{X}_i / \hat{n}_k^{(t)}$ is employed as an initial guess for starting the proximal gradient search, while the step-size parameter ν is kept fixed at 10^{-4} .

Sparse estimation of the row-precision matrices Ω_k

When (6) is maximized with respect to Ω_k , given current estimates of the precision matrices $\hat{\Gamma}_k^{(t-1)}$ and of the mean parameters $\hat{\mathbf{M}}_k^{(t)}$, the Q -function simplifies as follows:

$$Q_\Omega(\Omega_k) = \sum_{i=1}^n \hat{z}_{ik}^{(t)} \left[\frac{q}{2} \log |\Omega_k| - \frac{1}{2} \text{tr} \left\{ \Omega_k (\mathbf{X}_i - \hat{\mathbf{M}}_k^{(t)}) \hat{\Gamma}_k^{(t-1)} (\mathbf{X}_i - \hat{\mathbf{M}}_k^{(t)})' \right\} \right] - \lambda_2 \|\mathbf{P}_2 * \Omega_k\|_1. \quad (11)$$

By rearranging terms in (11), we obtain:

$$Q_\Omega(\Omega_k) = \log |\Omega_k| - \text{tr} \{ \Omega_k \mathbf{S}_\Omega \} - \frac{2}{\hat{n}_k q} \lambda_2 \|\mathbf{P}_2 * \Omega_k\|_1, \quad (12)$$

where

$$\mathbf{S}_\Omega = \sum_{i=1}^n \hat{z}_{ik}^{(t)} \frac{(\mathbf{X}_i - \hat{\mathbf{M}}_k^{(t)}) \hat{\Gamma}_k^{(t-1)} (\mathbf{X}_i - \hat{\mathbf{M}}_k^{(t)})'}{\hat{n}_k^{(t)} q}.$$

Maximization of (12) with respect to Ω_k corresponds a graphical lasso problem, which is solved using the coordinate descent algorithm by Friedman et al. (2008), where in our context their penalty coefficient is equal to $\frac{2}{\hat{n}_k q} \lambda_2 \mathbf{P}_2$. The algorithm is implemented in the R (R Core Team, 2023) package `glassoFast` (Sustik et al., 2018) and returns the estimates of the row precision matrices $\hat{\Omega}_k^{(t)}$, for $k = 1, \dots, K$.

Sparse estimation of the column-precision matrices Γ_k

In the maximization of (6) with respect to Γ_k , given current estimates $\hat{\Omega}_k^{(t)}$ and $\hat{\mathbf{M}}_k^{(t)}$, the Q -function simplifies to:

$$Q_\Gamma(\Gamma_k) = \sum_{i=1}^n \hat{z}_{ik}^{(t)} \left[\frac{p}{2} \log |\Gamma_k| - \frac{1}{2} \text{tr} \left\{ \Gamma_k (\mathbf{X}_i - \hat{\mathbf{M}}_k^{(t)})' \hat{\Omega}_k^{(t)} (\mathbf{X}_i - \hat{\mathbf{M}}_k^{(t)}) \right\} \right] - \lambda_3 \|\mathbf{P}_3 * \Gamma_k\|_1. \quad (13)$$

By rearranging terms in (13), we obtain the following objective function:

$$Q_\Gamma(\Gamma_k) = \log |\Gamma_k| - \text{tr} \{ \Gamma_k \mathbf{S}_\Gamma \} - \frac{2}{\hat{n}_k p} \lambda_3 \|\mathbf{P}_3 * \Gamma_k\|_1, \quad (14)$$

where

$$\mathbf{S}_\Gamma = \sum_{i=1}^n \hat{z}_{ik}^{(t)} \frac{(\mathbf{X}_i - \hat{\mathbf{M}}_k^{(t)})' \hat{\Omega}_k^{(t)} (\mathbf{X}_i - \hat{\mathbf{M}}_k^{(t)})}{\hat{n}_k p}.$$

Maximization of (14) with respect to Γ_k corresponds again to the graphical lasso, where in this case the original penalty coefficient is equal to $\frac{2}{\hat{n}_k p} \lambda_3 \mathbf{P}_3$. Also in this case the estimation is performed using the algorithm implemented in the package `glassoFast`, giving the estimates of the column precision matrices $\hat{\Gamma}_k^{(t)}$, for $k = 1, \dots, K$.

The updates based on the graphical lasso expressions (12) and (14) are iterated sequentially within the M-step at each cycle of the EM algorithm until convergence is reached, returning sparse estimates of the precision matrices Ω_k and Γ_k . The global convergence is evaluated by monitoring the increase in the penalized log-likelihood at each full EM iteration. The algorithm is considered to have reached convergence when $\ell_P(\hat{\Theta}^{(t)}; \mathbf{X}) - \ell_P(\hat{\Theta}^{(t-1)}; \mathbf{X}) < \varepsilon$ for a given $\varepsilon > 0$. In our analyses, ε is set equal to 10^{-5} .

The procedure described in this section is available within an R package at <https://github.com/AndreaCappozzo/sparsemixmat>, where some of the routines have been implemented in C++ to reduce the overall computing time.

3.3 A note on related penalty specifications

As briefly mentioned in Section 3.1, several options can be considered when specifying the penalty term in (3). A viable alternative to our proposal would consist in considering a standard lasso penalty on the matrices \mathbf{M}_k 's, coherently with the penalty adopted for the precision matrices. In this case, the penalty term would read as follows

$$p_{\lambda}(\mathbf{M}_k, \Omega_k, \Gamma_k) = \sum_{k=1}^K \lambda_1 \|\mathbf{P}_1 * \mathbf{M}_k\|_1 + \sum_{k=1}^K \lambda_2 \|\mathbf{P}_2 * \Omega_k\|_1 + \sum_{k=1}^K \lambda_3 \|\mathbf{P}_3 * \Gamma_k\|_1, \quad (15)$$

where \mathbf{P}_1 is a $p \times q$ matrix with non-negative entries, while the other quantities are defined as in the previous sections. Compared to the one introduced in Section 3.1, this penalty represents a less-structured way to induce sparsity in the mean matrices. In general, it does not allow to perform proper variable selection, since dimensions of the mean matrices are not jointly shrunk to zero. Nonetheless, the sparsity patterns could provide relevant insights and the method can be useful in some specific applications, as for example when no temporal dimension is present in the data. As highlighted in Section 2.2, Gao et al. (2021) consider lasso cell-wise penalization of matrix mixture mean parameters. However, the authors do not consider penalization of the component covariance matrices. As a result, the method may still require the estimation of a large number of parameters and does not provide a flexible model for the association structures between row and column variables. To overcome these limitations, in their recent work, Heo and Baek (2021) derive a penalized matrix normal mixture model where sparsity is also induced on the precision matrices, by using a penalty function similar to (15). Nonetheless, in their proposed estimation procedure, and in particular in the M-step update for \mathbf{M}_k 's, the authors implicitly assume that both the rows and the columns component precision matrices are diagonal. This assumption can lead to inaccurate estimates, especially in those applications where complex conditional dependency patterns are present. For these reasons, in the following we derive an estimation scheme where the independence assumption is not required. Note that Ω_k and Γ_k are estimated as in Section 3.2.3, therefore in what follows we only outline the updating formula for \mathbf{M}_k . Furthermore, the E-step and the considerations about the initialization strategy and the convergence criterion remain unchanged.

Consider the current estimates of the precision matrices $\hat{\Omega}_k$ and $\hat{\Gamma}_k$, where we omit the

iteration superscript for ease of notation. When the penalty term is defined as in (15), in the maximization step with respect to \mathbf{M}_k , the Q -function can be expressed as follows

$$Q_M(\mathbf{M}_k) = \sum_{i=1}^n \hat{z}_{ik}^{(t)} \left[\text{tr} \left\{ \hat{\Omega}_k \mathbf{X}_i \hat{\Gamma}_k \mathbf{M}'_k \right\} - \frac{1}{2} \text{tr} \left\{ \hat{\Omega}_k \mathbf{M}_k \hat{\Gamma}_k \mathbf{M}'_k \right\} \right] - \lambda_1 \|\mathbf{P}_1 * \mathbf{M}_k\|_1. \quad (16)$$

We propose a cell-wise coordinate ascent estimation for $m_{ls,k}$, where $m_{ls,k}$ denotes the element in the l -th row and s -th column of matrix \mathbf{M}_k . Likewise, let $\hat{\omega}_{ls,k}$, $\hat{\gamma}_{ls,k}$ and $p_{ls,1}$ denote the elements in the l -th row and s -th column of matrices $\hat{\Omega}_k$, $\hat{\Gamma}_k$ and \mathbf{P}_1 respectively. Lastly, $x_{ls,i}$ is similarly defined in relation to a matrix observation \mathbf{X}_i . The following proposition characterizes the updating formula:

Proposition 1: *The sufficient and necessary conditions for $\hat{m}_{ls,k}$ to be a (global) maximizer of (16) (for fixed l , s and k) are*

$$\sum_{i=1}^N \hat{z}_{ik} \sum_{r=1}^p \sum_{c=1}^q \hat{\omega}_{lr,k} x_{rc,i} \hat{\gamma}_{cs,k} - \hat{n}_k \sum_{r=1}^p \sum_{c=1}^q \hat{\omega}_{lr,k} \hat{m}_{rc,k} \hat{\gamma}_{cs,k} = \lambda_1 p_{ls,1} \text{sign}(\hat{m}_{ls,k}), \quad \text{if } \hat{m}_{ls,k} \neq 0 \quad (17)$$

and

$$\left| \sum_{i=1}^n \hat{z}_{ik} \left[\sum_{\substack{r=1 \\ r \neq l}}^p \hat{\omega}_{lr,k} \left(\sum_{c=1}^q (x_{rc,i} - \hat{m}_{rc,k}) \hat{\gamma}_{cs,k} \right) + \hat{\omega}_{ll,k} \left(\sum_{\substack{c=1 \\ c \neq s}}^q (x_{lc,i} - \hat{m}_{lc,k}) \hat{\gamma}_{cs,k} \right) + \hat{\omega}_{ll,k} x_{ls,i} \hat{\gamma}_{ss,k} \right] \right| \leq \lambda_1 p_{ls,1}, \quad \text{if } \hat{m}_{ls,k} = 0. \quad (18)$$

Thus, at the t -th iteration of the EM algorithm $\hat{m}_{ls,k}^{(t)} = 0$ if

$$\left| \sum_{i=1}^n \hat{z}_{ik}^{(t)} \left[\sum_{\substack{r=1 \\ r \neq l}}^p \hat{\omega}_{lr,k}^{(t-1)} \left(\sum_{c=1}^q (x_{rc,i} - \hat{m}_{rc,k}^{(t)}) \hat{\gamma}_{cs,k}^{(t-1)} \right) + \hat{\omega}_{ll,k}^{(t-1)} \left(\sum_{\substack{c=1 \\ c \neq s}}^q (x_{lc,i} - \hat{m}_{lc,k}^{(t)}) \hat{\gamma}_{cs,k}^{(t-1)} \right) + \hat{\omega}_{ll,k}^{(t-1)} x_{ls,i} \hat{\gamma}_{ss,k}^{(t-1)} \right] \right| \leq \lambda_1 p_{ls,1}, \quad (19)$$

otherwise, $\hat{m}_{ls,k}^{(t)}$ is obtained by solving

$$\begin{aligned} \hat{n}_k^{(t)} \hat{\omega}_{ll,k}^{(t-1)} \hat{m}_{ls,k}^{(t)} \hat{\gamma}_{ss,k}^{(t-1)} + \lambda_1 p_{ls,1} \operatorname{sign}(\hat{m}_{ls,k}^{(t)}) &= \sum_{i=1}^n \hat{z}_{ik}^{(t)} \sum_{r=1}^p \sum_{c=1}^q \hat{\omega}_{lr,k}^{(t-1)} x_{rc,i} \hat{\gamma}_{cs,k}^{(t-1)} + \\ &\quad - \hat{n}_k^{(t)} \left(\sum_{\substack{r=1 \\ r \neq l}}^p \sum_{\substack{c=1 \\ c \neq s}}^q \hat{\omega}_{lr,k}^{(t-1)} \hat{m}_{rc,k}^{(t)} \hat{\gamma}_{cs,k}^{(t-1)} \right) \end{aligned} \quad (20)$$

with respect to $\hat{m}_{ls,k}^{(t)}$.

The proof of Proposition 1 is reported in the Supplementary Material. This result corrects an inaccuracy introduced in Equation (5) of [Heo and Baek \(2021\)](#) and it can be seen as the matrix-variate extension of Theorem 1 of [Zhou et al. \(2009\)](#). Convergence to the global maximum is assured thanks to theoretical properties of coordinate descent algorithms (see e.g., [Wright, 2015](#)). The described procedure, for sufficiently large λ_1 , forces some $\hat{m}_{ls,k}^{(t)}$ to be shrunk to 0, ultimately inducing sparsity in \mathbf{M}_k , $k = 1, \dots, K$. Notice however, as already mentioned, that such a penalty does not allow to directly perform variable selection within a matrix-variate data framework. The latter is achieved only employing a group-lasso penalization scheme, as highlighted in Section 3.1.

3.4 Model selection

The model estimation strategy in Section 3.2 has been outlined by considering K and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ fixed. However, in practical applications, the number of clusters and the penalty hyperparameters are not known a priori and need to be chosen using model selection strategies. In this work, we select K and $\boldsymbol{\lambda}$ which maximize a modified version of the Bayesian Information Criterion (BIC, [Schwarz, 1978](#)), already considered in [Pan and Shen \(2007\)](#) and [Casa et al. \(2022\)](#). In detail, we use the following criterion:

$$BIC = 2 \log L(\hat{\Theta}) - d_0 \log(n), \quad (21)$$

where d_0 is the number of parameters not shrunk to zero and $\log L(\hat{\Theta})$ is the log-likelihood evaluated at $\hat{\Theta}$.

The adequacy of the BIC for selecting the number of mixture components has been thoroughly studied (see e.g., [Roeder and Wasserman, 1997](#); [Keribin, 2000](#)), and the criterion

has been widely used both in the two-way and, more recently, in the three-way model-based clustering frameworks (Sarkar et al., 2020; Tomarchio et al., 2022; Sharp et al., 2022). Moreover, the formulation in (21) has been proven useful also to tune the intensity of the penalization both in the lasso (Zou et al., 2007) and in the sparse precision matrix estimation contexts (Lian, 2011). Nonetheless, other model selection strategies may be pursued, especially in situations where exhaustive grid searches are considered too computational demanding. Possible alternatives are provided by stochastic optimization algorithms, such as genetic algorithms (Holland, 1992), or to conditional search schemes. Another interesting approach is outlined in Jiang et al. (2015), where the authors develop the E-MS algorithm, in which model selection is performed within each iteration of the standard EM algorithm.

4 Simulation study

4.1 Experimental Setup

In this section, we assess the performance of the proposed method on synthetic data, evaluating its ability in recovering the underlying sparse patterns and the clustering structure. For each replication of the simulation experiment, we generate $n = 150$ samples from a 3-component matrix Gaussian mixture model, in which mean matrices and both row and column precision matrices have some degree of sparsity. The row and column precision matrices have dimensions $p \times p$ and $q \times q$, with p and q equal to 10 and 5, respectively. The 10×5 mean matrices \mathbf{M}_k , $k = 1, 2, 3$ have a row-wise sparse structure, visually displayed in Figure 1. The data generating process purposely reproduce a situation in which some of the p variables measured in q occasions are irrelevant for clustering. In this specific context, the second, fourth, sixth, eighth and tenth row do not convey any grouping information, being identically equal to 0 in all clusters. We consider two distinct scenarios according to the sparsity structure enforced for the row precision matrices Ω_k :

- *Alternated-blocks row precision matrices:* the 10×10 row precision matrices Ω_k , $k = 1, 2, 3$ have a block-wise sparse structure, as visually displayed in the upper panels of Figure 2.

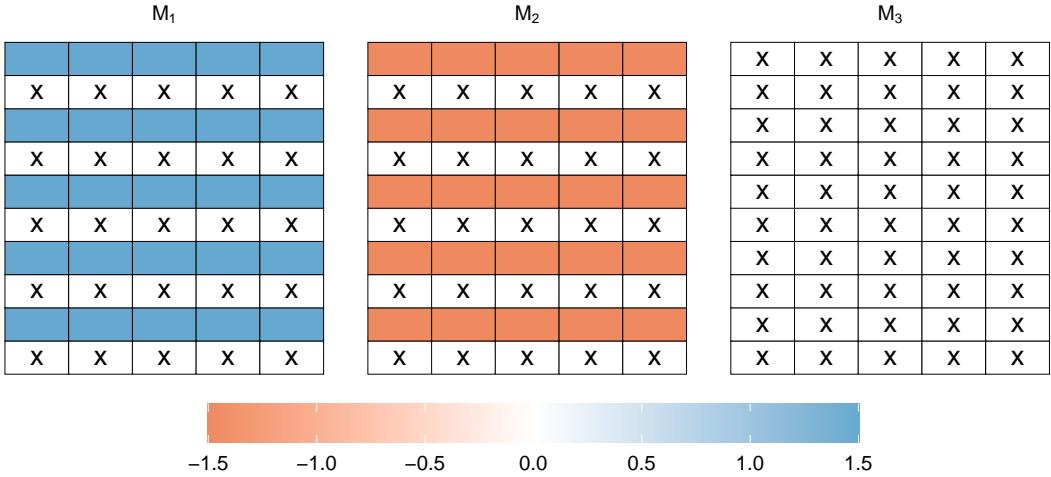


Figure 1: Heatmaps of the true 10×5 mean matrices \mathbf{M}_k , $k = 1, 2, 3$, considered in the simulated data experiment. A zero entry in the matrices is indicated with the symbol \times .

- *Sparse-at-random row precision matrices:* the row precision matrices have a sparse at random Erdős-Rényi graph structure (Erdős and Rényi, 1960) with probabilities of connection equal to 0.2, 0.5 and 0.8 for Ω_1 , Ω_2 and Ω_3 , respectively. These are visually displayed in the upper panels of Figure 6.

In both scenarios, the column precision matrices Γ_k are generated according to a sparse at random Erdős-Rényi graph structure, while the mixing proportions τ_k are assumed equal to $1/K$, $K = 3$. The experiment is repeated 100 times, and for each replication the following models are fitted to the synthetic data samples:

- *Full MGMM:* the finite mixtures of matrix normal distributions originally introduced in Viroli (2011a), where full matrix parameters are estimated for each component. This model specification corresponds to a G-VVV-VV model following the nomenclature introduced in Sarkar et al. (2020).
- *Sparsemixmat:* the penalized MGMM method introduced in this paper, with a group-lasso penalization imposed on the rows of the mean matrices according to the penalty term in (4).

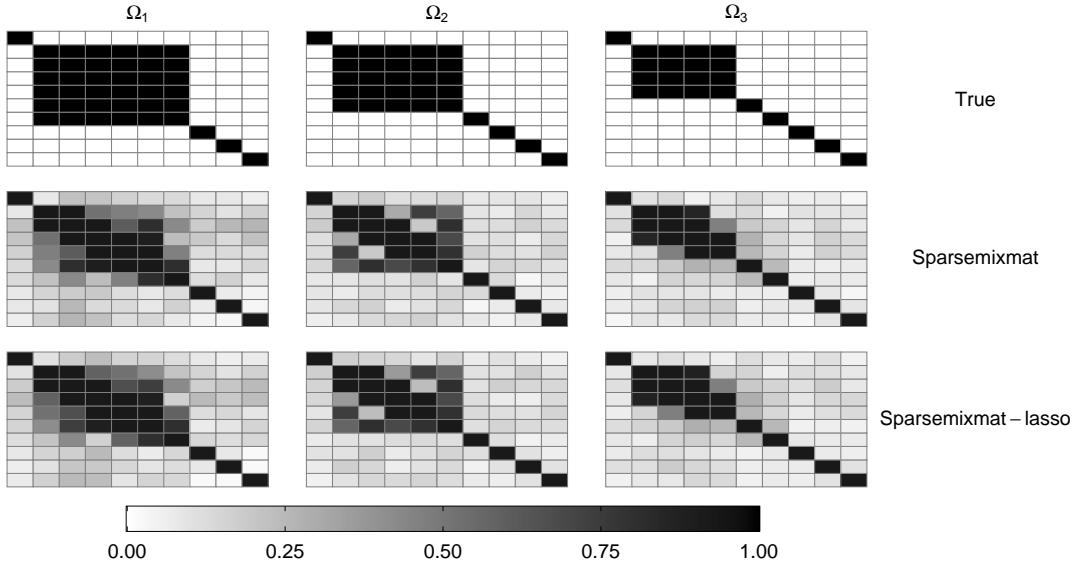


Figure 2: Alternated-blocks row precision matrices scenario. True association structures (top) and estimated association structures averaged over 100 replications (middle and bottom) for the row precision matrices Ω_k , $k = 1, 2, 3$. Black squares denote a non-zero parameter between two variables.

- *Sparsemixmat-lasso*: the penalized MGMM methodology introduced in Heo and Baek (2021), with a entry-wise lasso penalization on the mean matrices according to the penalty term in (15), and estimated following the steps outlined in Section 3.3.

For the *Sparsemixmat* and *Sparsemixmat-lasso* models, a search over an equispaced grid of elements for each penalty term λ_1 , λ_2 and λ_3 is considered, and the best model according to the BIC criterion introduced in Section 3.4 is retained. All competing methods are initialized via model-based agglomerative hierarchical clustering as discussed in Section 3.2.1. The methods are evaluated according to their ability in performing variable selection, recovering the true sparsity structure, and correctly retrieving the cluster allocations. The issue of matching the estimated clustering with the actual classification is addressed using the `matchClasses` function from the `e1071` R package (Meyer et al., 2020). Simulation results are reported in the next subsection.

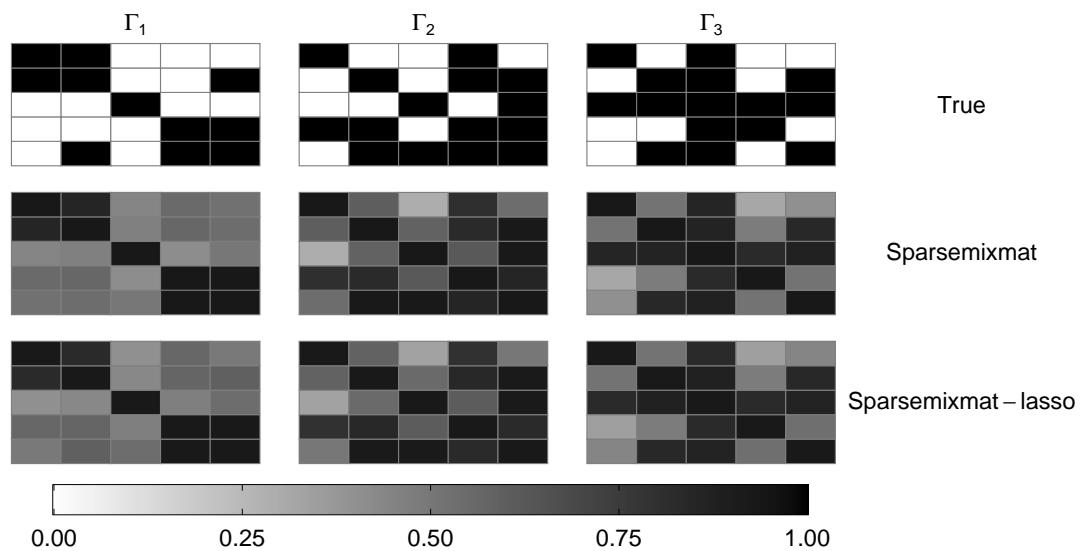


Figure 3: Alternated-blocks row precision matrices scenario. True association structures (top) and estimated association structures averaged over 100 replications (middle and bottom) for the column precision matrices Γ_k , $k = 1, 2, 3$. Black squares denote a non-zero parameter between two occasions.

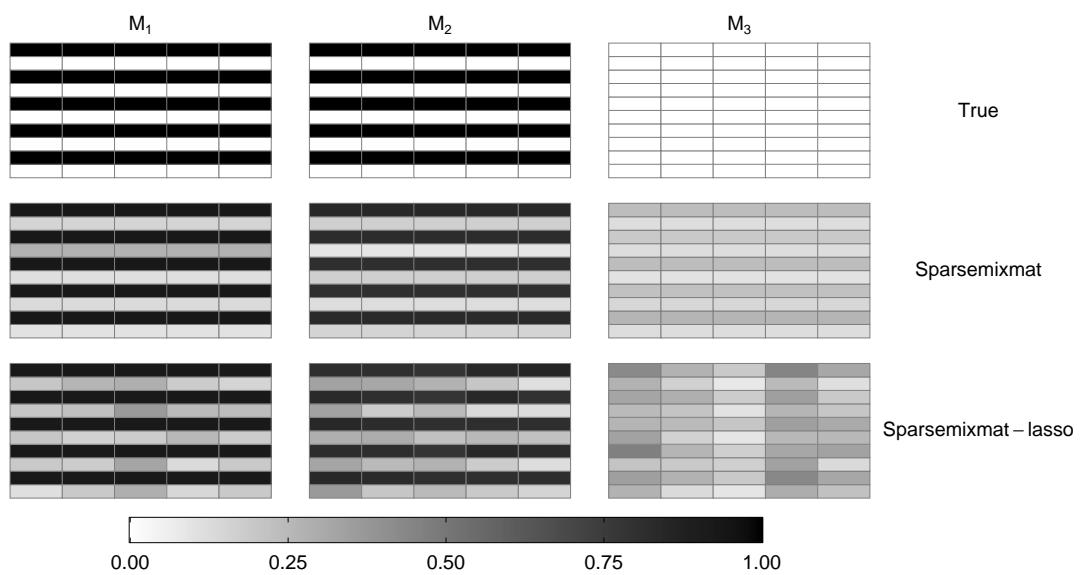


Figure 4: Alternated-blocks row precision matrices scenario. True mean matrices (top) and estimated mean matrices averaged over 100 replications (middle and bottom) associated to the data generating mean matrices \mathbf{M}_k , $k = 1, 2, 3$. Black squares denote a non-zero parameter.

4.2 Simulation study results

4.2.1 Alternated-blocks row precision matrices

In Figure 2, we report the heatmap plots associated to the 10×10 row precision matrices Ω_k , $k = 1, 2, 3$ for the *alternated-blocks row precision matrices* scenario. In the top row, each heatmap represents the association structure corresponding to a component row precision matrix, where each black square denotes the presence of a non zero parameter, and hence an association between a pair of variables. The second and third rows are the heatmaps of the proportion of times a non-zero precision parameter has been estimated between a pair of variables. As it emerges from the graphs, we note that both *Sparsemixmat* and *Sparsemixmat-lasso* satisfactorily recover the true underlying sparsity structure. A moderate penalty on the row-precision matrices allows for the shrinkage to zero of some of the elements of Ω_k , which allow the correct identification of the conditional association structures among the variables in the clusters. Figure 3 reports similar heatmaps related to the 5×5 column precision matrices. Also for this dimension of the matrix data, the association structure is correctly identified by both methods.

Different results are observed when examining the estimates of the cluster mean matrices \mathbf{M}_k , reported in Figure 4. In the figure, the heatmaps report the non-zero entries of the data generating mean matrices and the the proportion of zero entries for the estimated ones, averaged over 100 replications. The row-wise shrinkage of *Sparsemixmat*, enforced by the group-lasso penalty, favors a better recovery of the mean matrices structure compared to the entry-wise lasso shrinkage of *Sparsemixmat-lasso*. This conclusion is further supported by the metrics displayed in Table 1 where, we report the average Frobenius distance between true and estimated parameters for each mixture component. Notably, *Sparsemixmat* outperforms the competing methods, exhibiting the lowest average distance for every mean matrix across all three clusters. While *Sparsemixmat-lasso* and *Full MGMM* seem to perform slightly better when looking at row and column precision matrices, the difference is often negligible. Moreover, our proposed approach achieves superior results in terms of recovery the underlying cluster partition, as measured by the adjusted Rand index (ARI, Hubert and Arabie, 1985), as well as overall model parsimony,

Table 1: Alternated-blocks row precision matrices *scenario*. Frobenius distance between true and estimated parameters, adjusted Rand index (ARI), and number of non-zero parameters (d_0) averaged over 100 repetitions. Bold numbers indicate the best performing method according to the considered metric. Standard errors are reported in brackets.

	<i>Full MGMM</i>	<i>Sparsemixmat</i>	<i>Sparsemixmat-lasso</i>
$\ \mathbf{M}_1 - \hat{\mathbf{M}}_1\ _F$	38.617 (77.98)	32.965 (76.716)	34.624 (77.839)
$\ \mathbf{M}_2 - \hat{\mathbf{M}}_2\ _F$	36.773 (78.055)	13.876 (37.567)	14.678 (37.592)
$\ \mathbf{M}_3 - \hat{\mathbf{M}}_3\ _F$	16.382 (30.183)	7.714 (17.721)	8.161 (18.563)
$\ \boldsymbol{\Omega}_1 - \hat{\boldsymbol{\Omega}}_1\ _F$	1.136 (0.97)	3.218 (0.712)	3.028 (0.647)
$\ \boldsymbol{\Omega}_2 - \hat{\boldsymbol{\Omega}}_2\ _F$	1.256 (1.573)	1.47 (0.412)	1.383 (0.393)
$\ \boldsymbol{\Omega}_3 - \hat{\boldsymbol{\Omega}}_3\ _F$	1.529 (2.07)	0.796 (0.513)	0.75 (0.494)
$\ \boldsymbol{\Gamma}_1 - \hat{\boldsymbol{\Gamma}}_1\ _F$	2.767 (6.117)	2.807 (6.002)	2.797 (6.095)
$\ \boldsymbol{\Gamma}_2 - \hat{\boldsymbol{\Gamma}}_2\ _F$	3.794 (6.661)	2.272 (4.251)	2.239 (4.336)
$\ \boldsymbol{\Gamma}_3 - \hat{\boldsymbol{\Gamma}}_3\ _F$	5.376 (10.043)	4.24 (8.431)	4.202 (8.609)
ARI	0.948 (0.156)	0.992 (0.058)	0.991 (0.058)
d_0	362 (0)	166.602 (19.931)	175.913 (15.51)

quantified by the number of estimated parameters. *Sparsemixmat* shows a higher ARI and a lower number of non-zero parameters compared to *Full MGMM* and *Sparsemixmat-lasso*. It is important to note that *Full MGMM* does not employ any shrinkage, resulting in a total of $(K - 1) + K(pq + p(p + 1)/2 + q(q + 1)/2)$ estimated parameters in all cases.

Another aspect to examine is the performance of the proposed approach in terms of variable selection. Specifically, given the matrix-variate nature of the data, we are interested in monitoring the method's ability in correctly identifying the zero rows of the mean matrices, and hence correctly detect those variables that are constantly equal to zero across occasions and clusters. To measure this, we make use of the F_1 score, defined as follows:

$$F_1 = \frac{\text{tp}}{\text{tp} + 0.5(\text{fp} + \text{fn})}, \quad (22)$$

where tp denotes the number of zero rows in \mathbf{M}_k correctly estimated as such, while fp and fn denote the number of non-zero rows wrongly shrunk to 0 and the number of zero rows

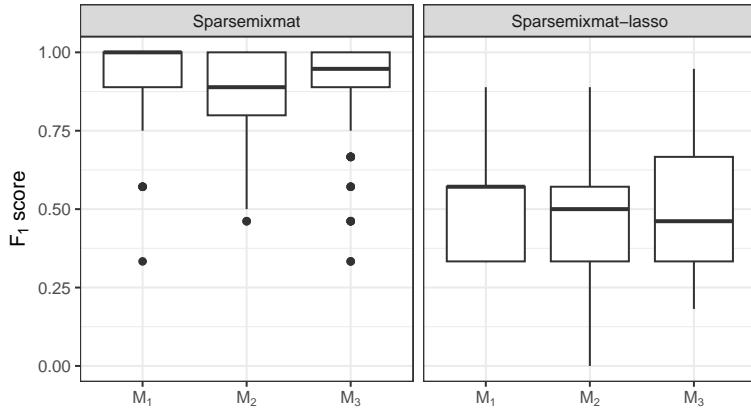


Figure 5: Alternated-blocks row precision matrices scenario. Boxplots of the F_1 score for 100 replications of the experiment.

not shrunk to 0, respectively. Figure 5 displays boxplots of the F_1 score for the *Sparsemixmat* and *Sparsemixmat-lasso* methods. By enforcing entire rows of $\hat{\mathbf{M}}_k$ to be shrunk to zero by means of the group-lasso penalty, the *Sparsemixmat* approach achieves better variable selection performance. Conversely, for the *Sparsemixmat-lasso*, which applies entry-wise lasso shrinkage, there is no guarantee that entire rows will be ultimately set to 0. Therefore, when the primary aim is multivariate variable selection or solving the support union problem within a matrix mixture context, our proposed approach is preferable.

Similar results are observed when more complex dependence structures between the p variables are considered, as it will be reported in the next subsection.

4.2.2 Sparse-at-random row precision matrices

In the second scenario, the row precision matrices are constructed having a sparse-at-random Erdős-Renyi graph structure. Figure 6 shows the data-generating and estimated association structures, with interpretation similar to previous similar figures. We note how the more challenging dependence patterns among the variables affect the performance of the penalized models. Irrespective of the considered methods, the number of non-zero parameters is consistently overestimated, resulting in solutions where the levels of sparsity

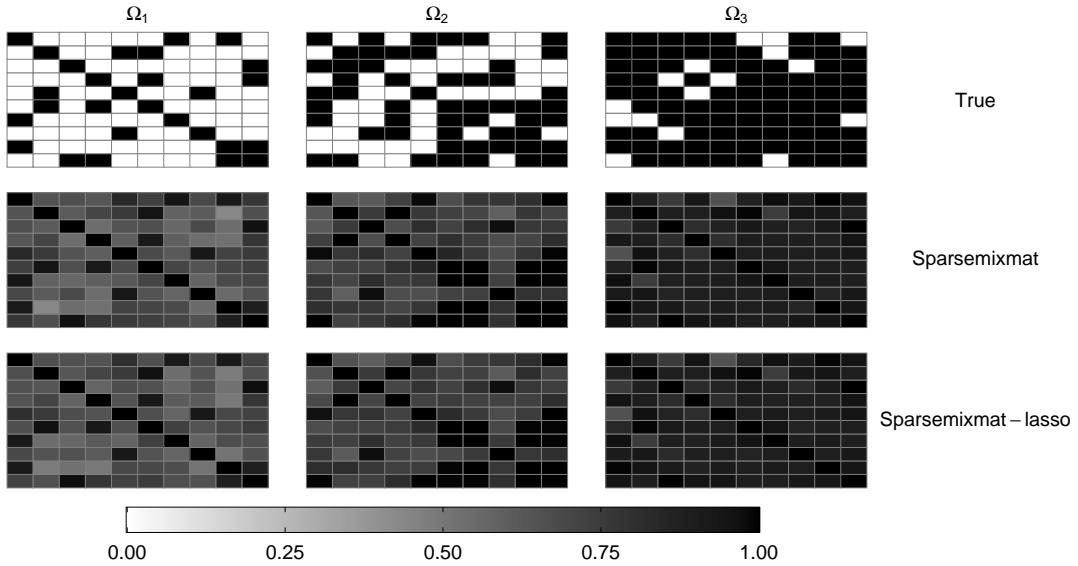


Figure 6: Sparse-at-random row precision matrices scenario. True association structures (top) and estimated association structures averaged over 100 replications (middle and bottom) for the row precision matrices Ω_k , $k = 1, 2, 3$. Black squares denote a non-zero parameter between two variables.

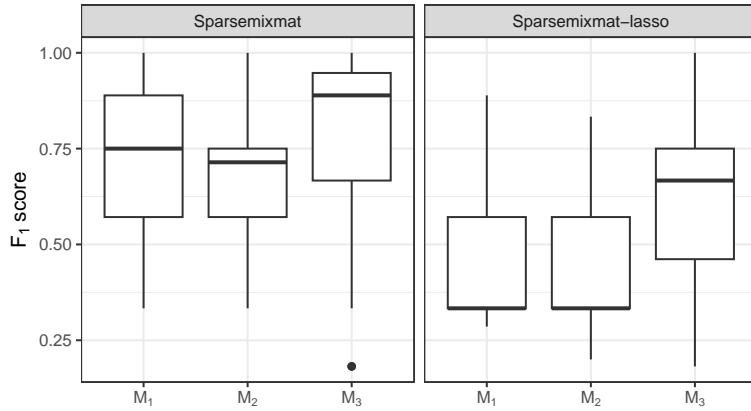


Figure 7: Sparse-at-random row precision matrices scenario. Boxplots of the F_1 score for 100 replications of the experiment.

Table 2: Sparse-at-random row precision matrices *scenario*. Frobenius distance between true and estimated parameters, adjusted Rand index (ARI), and number of non-zero parameters (d_0) averaged over 100 repetitions. Bold numbers indicate the best performing method according to the considered metric. Standard errors are reported in brackets.

	<i>Full MGMM</i>	<i>Sparsemixmat</i>	<i>Sparsemixmat-lasso</i>
$\ \mathbf{M}_1 - \hat{\mathbf{M}}_1\ _F$	60.188 (93.856)	51.972 (90.38)	53.639 (91.107)
$\ \mathbf{M}_2 - \hat{\mathbf{M}}_2\ _F$	39.912 (58.11)	17.181 (31.259)	20.183 (29.645)
$\ \mathbf{M}_3 - \hat{\mathbf{M}}_3\ _F$	37.871 (45.368)	11.685 (21.207)	12.44 (22.506)
$\ \boldsymbol{\Omega}_1 - \hat{\boldsymbol{\Omega}}_1\ _F$	4.704 (7.408)	3.19 (5.278)	3.152 (5.35)
$\ \boldsymbol{\Omega}_2 - \hat{\boldsymbol{\Omega}}_2\ _F$	5.277 (7.607)	4.353 (5.774)	4.233 (5.758)
$\ \boldsymbol{\Omega}_3 - \hat{\boldsymbol{\Omega}}_3\ _F$	6.525 (9.775)	5.694 (5.622)	5.628 (5.666)
$\ \boldsymbol{\Gamma}_1 - \hat{\boldsymbol{\Gamma}}_1\ _F$	2.623 (4.448)	2.884 (6.512)	2.981 (6.661)
$\ \boldsymbol{\Gamma}_2 - \hat{\boldsymbol{\Gamma}}_2\ _F$	12.287 (23.165)	9.49 (18.434)	11.866 (22.579)
$\ \boldsymbol{\Gamma}_3 - \hat{\boldsymbol{\Gamma}}_3\ _F$	18.079 (26.515)	17.61 (23.537)	16.404 (24.64)
ARI	0.944 (0.162)	1 (<0.01)	1 (<0.01)
d_0	362 (0)	257.204 (20.289)	251.071 (12.01)

of the $\boldsymbol{\Omega}_k$ matrices are underestimated. Similar results are observed for the column precision matrices and the mean matrices (not reported here). Nonetheless, *Sparsemixmat* seems to outperform *Full MGMM* and *Sparsemixmat-lasso* when evaluating the performance in terms of Frobenius distance and recovering of the true clustering, as it is indicated in Table 2). Particularly, the mean and column-precision matrices are quite satisfactorily estimated by *Sparsemixmat*, with only a slightly higher total number of parameters in comparison to *Sparsemixmat-lasso*. Similarly to the previous scenario, with regard to the ability of performing variable selection, a group-lasso penalty on the rows of \mathbf{M}_k is to be preferred, as highlighted in the boxplots of Figure 7, where *Sparsemixmat* shows consistently higher F_1 score values compared to *Sparsemixmat-lasso*. Interestingly, the variable selection performance of both methods in terms of the F_1 score is lower in this scenario compared to the previous one. This finding suggests that the performance in variable selection does

not only depend on the penalty imposed to the mean matrices, but it is also affected by how well the dependence structure among the p variables in the K clusters is recovered.

In summary, the proposed approach adequately tackle the problem of clustering matrix-variate data with sparse model parameters. The method is flexible, it is capable of capturing cluster-wise different dependence structures in both variables and occasions, it enables row-wise variable selection when variables are recorded over multiple occasions, and it detects effectively the clustering structure in the matrix data. These considerations hold true not only in an experimental setup but also in the analysis of real-world data, as reported in the next section.

5 Application: criminal trends in the US

5.1 Data description

We analyze data from the United States Department of Justice Federal Bureau of Investigation concerning violent and property crimes of 236 American cities. The aim of the analysis is to cluster cities with similar crime trends and to identify which crime types exhibit relevant differences in the time patterns across clusters. In the data, for each city, $p = 7$ variables capturing the rates of murder, rape, robbery, aggravated assault, burglary, larceny-theft, and motor vehicle theft are measured over $q = 13$ years in the period between 2000 and 2012. Thus, the data can be conveniently arranged in a $7 \times 12 \times 236$ array, where each statistical unit \mathbf{X}_i , $i = 1, \dots, 236$, takes the form of a 7×13 matrix. The dataset is publicly available within the `MatTransMix` R package ([Zhu et al., 2022](#)) and has been previously analyzed in [Melnykov and Zhu \(2019\)](#), where the authors introduced a method based on mixture of matrix transformation regression time series. The next subsection includes discussion of the results of our modeling approach and comparisons with the findings from [Melnykov and Zhu \(2019\)](#).

5.2 Results

We implement an initial pre-processing step in which the statistical units are cell-wise centered and log-transformed to alleviate skewness. Subsequently, the *Sparsemixmat* model introduced in Section 3 is fitted to the crime data. The shrinkage parameters are varied within a pre-specified grid of values, and considering $K \in \{3, 4, 5, 6\}$.

The BIC as introduced in Equation (21) selects $K = 3$ clusters, with corresponding shrinkage hyperparameters λ_1 , λ_2 , and λ_3 equal to 3.81, 0 and 14.3, respectively. The penalty coefficient $\lambda_2 = 0$ implies that the estimated row precision matrices $\hat{\Omega}_k$ for the selected model are non-sparse, indicating relevant associations between the crime types across clusters. On the other hand, with the selected λ_1 and λ_3 greater than zero, both the estimated mean matrices and the column precision matrices measuring the dependence between the time occasions have certain degrees of sparsity. Visual representations of these estimated parameters are displayed in figures 8 and 9. From Figure 8, we observe that no crime type presents estimated cluster mean rates equal to zero across all clusters, indicating that all variables contain some discriminating information. However, in light of the considerations of Section 3.1, clusters tend to be differentiated over the rates of certain crimes across the years. For example, all clusters have dissimilar burglary and larceny-theft rates, while cluster 1 and 3 tend to overlap in terms of murder, rape, and motor vehicle theft rates. In addition, robbery and assault crime rates tend to stay constant over time for the cities in cluster 3, while they vary for those in clusters 1 and 2. Figure 9 shows that the estimated column precision matrices, which embed the conditional association structure of the crime rates between years, tend to have a banded structure. The entries along the diagonal are generally non-zero, while entries between far in time occasions are generally shrunk to zero, indicating higher levels of association between consecutive years.

The clustering of the cities in the data is displayed in the map of Figure 10, while the mean rate profiles of the resulting partition, computed for the crime types in the original scale, are reported in Figure 11. More in detail, cluster 3 (blue color) identifies the safest cities in the country, which tend also to be the smallest in size. Higher concentration of safe cities can be observed in Northern Texas, the Los Angeles-San Diego area, and parts of the

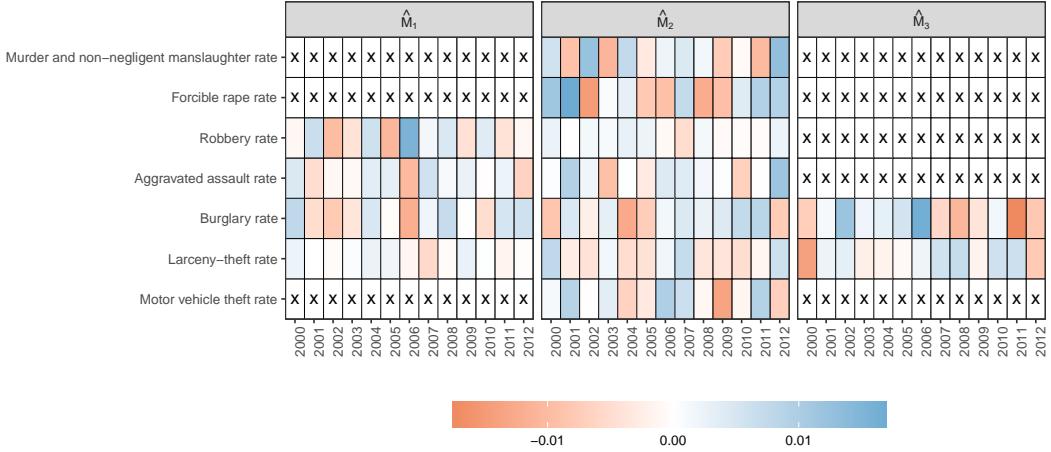


Figure 8: Crime data. Estimated mean matrices \hat{M}_k , $k = 1, 2, 3$ for the sparsenmixmat model. Colors denote the values of the estimates; a 0 entry in the matrices is indicated by the symbol \times .

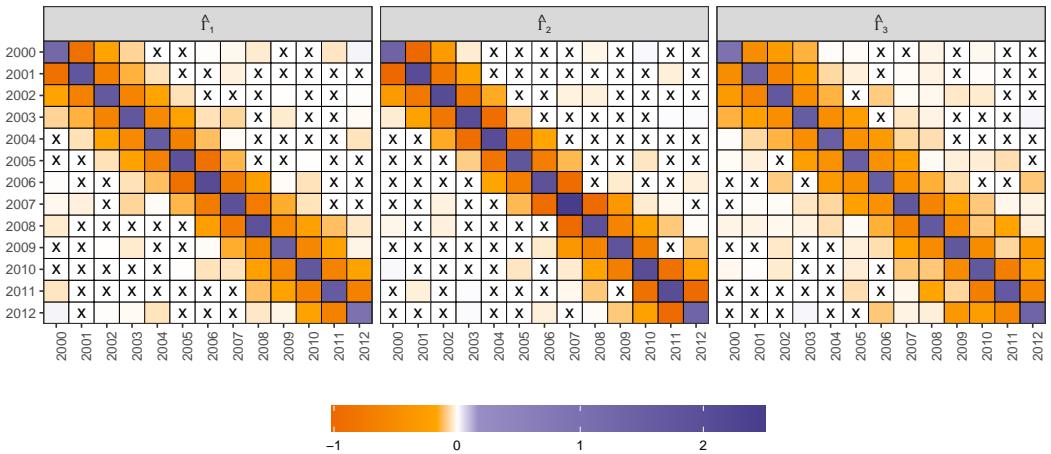


Figure 9: Crime data. Estimated column precision matrices $\hat{\Gamma}_k$, $k = 1, 2, 3$, for the sparsenmixmat model. Colors denote the values of the entries; a 0 entry in the matrices is indicated by the symbol \times .

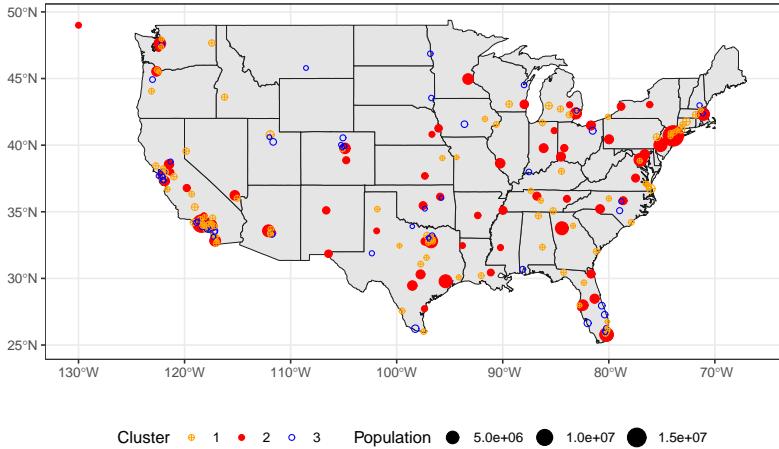


Figure 10: Crime data. Map of the USA showing the clustering of the cities obtained from the sparsemixmat model. The sizes of the circles is proportional to the city population. Colors and symbols indicate different clusters.

northern states, together with a few coastal areas in Florida and in the south of Indiana. Cluster 2 (red color) includes the cities with the highest crime rates of the considered types. From the map, it appears that these cities tend to be unevenly distributed across the US, with a concentration in the eastern part of the country, which is also the most densely populated. Lastly, cluster 1 (orange color) comprises cities that are slightly less safe, for which the mean crime rates over time tend to be higher than those in cluster 3. However, as remarked previously, the cities in these clusters tend to overlap in terms of murder, rape, and motor vehicle theft rates over time (see Figure 8).

We highlight several similarities in the results discussed here and those of the analyses reported in [Melnykov and Zhu \(2019\)](#). First off, compared to the partition obtained in their 3-cluster model we observe an agreement of approximately 75% of cases, along with a very similar interpretation of the resulting clusters. Dependence patterns across time similar to those displayed in Figure 9 have also been observed in [Melnykov and Zhu \(2019\)](#), in which a first order autoregressive model was employed to reduce the number of parameters and model the time dependence. While this is indeed a sensible modeling choice given the temporal dependence of these data, we remark the flexibility of our procedure in automatically

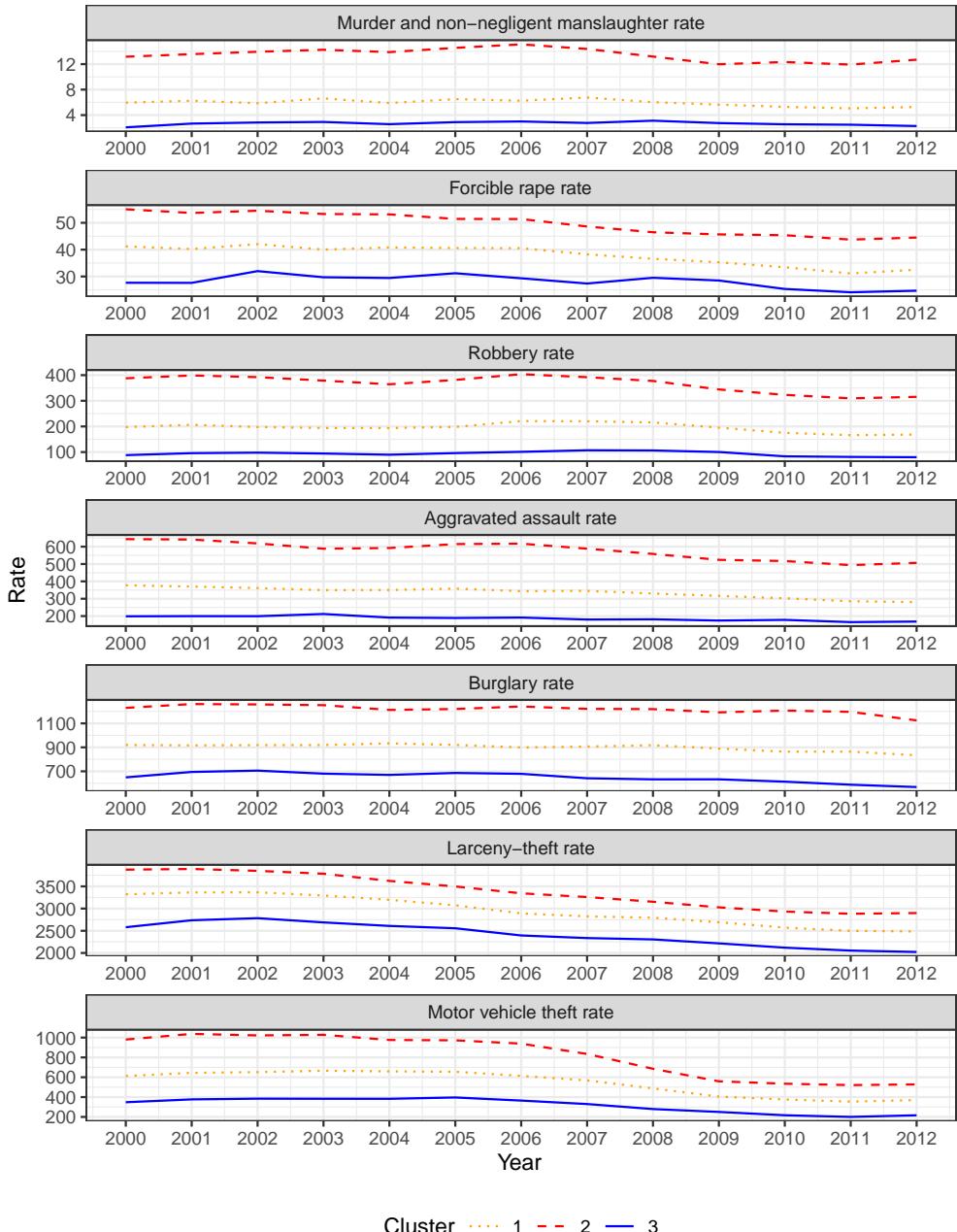


Figure 11: Crime data. Mean profiles for the sparsenixmat model. The mean profiles are computed for the variables in original scale. Colors and line types illustrate different clusters.

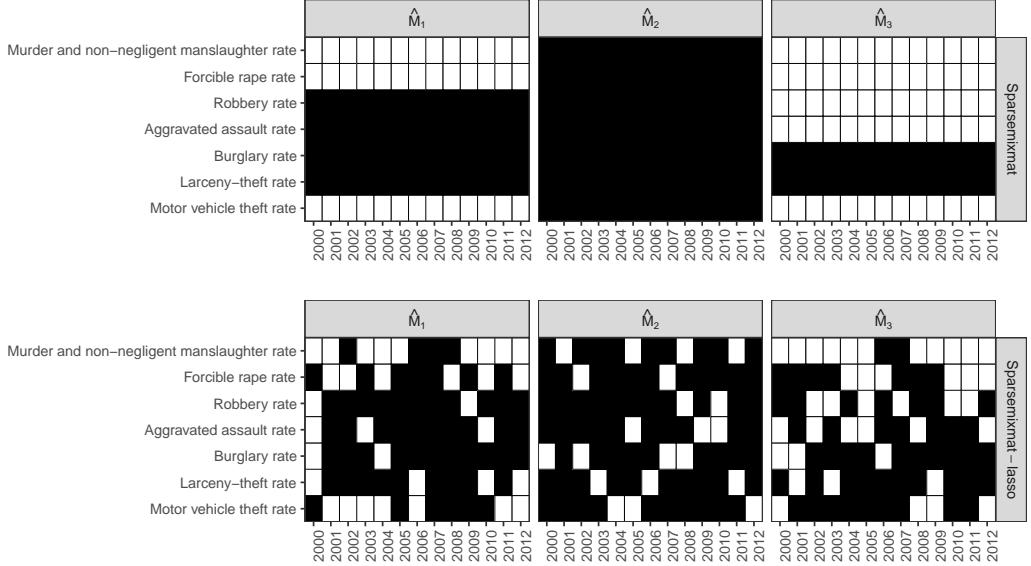


Figure 12: Crime data. Sparse structure associated to the mean matrices $\hat{\mathbf{M}}_k$, $k = 1, 2, 3$ for the *sparsenmixmat* and *sparsenmixmat-lasso* models. Black squares denote an entry different from 0.

capturing an autoregressive-like structure in the time occasions. This is achieved through the penalization imposed on the column precision matrices without the need to pre-specify any pattern or dependence structure. In addition, in [Melnykov and Zhu \(2019\)](#), to overcome the overparametrization issue associated to the mean matrices, the authors consider regressing crime rates on years. In contrast, our proposed method employs a group-lasso penalty which effectively serves the same purpose, without the specification of a regression model.

We conclude this section by comparing the results obtained with our *sparsenmixmat* procedure with the *sparsenmixmat-lasso* of [Heo and Baek \(2021\)](#). The two models provide very similar partitions of the cities in the data, having almost perfect agreement and with only 9 cities assigned to different clusters. Figure 12 shows the estimated sparse structures associated with the estimated cluster mean matrices, obtained under the two different penalties. For *Sparsenmixmat-lasso*, all the crime types have non-zero mean rates for some of

the years and clusters, making difficult to differentiate the clusters in terms of overall mean crime rate patters across years. Once again, it is worth to highlight the ease of interpretation induced by the group-lasso penalty of *Sparsemixmat*, making it a more favorable option when clustering with matrix-variate data where variables are recorded over multiple time occasions.

6 Conclusion

The complex structure entailed by three-way data makes clustering matrices a particularly challenging task. By framing the problem into a well-defined probabilistic context, model-based approaches are unarguably among the most commonly adopted to address these challenges. Nonetheless, these approaches have to face severe issues and limitations even when dealing with three-way data of moderate dimensions. In this work, we propose a modeling framework that alleviates these drawbacks, thus allowing to cluster matrix-variate data even when the number of variables p or the number of occasions q is moderate. In particular, the presented method relies on a penalized likelihood approach that allows to induce sparsity in the model parameters. The penalties on the row and column precision matrices reduce greatly the number of parameters to estimate, while simultaneously easing the interpretation of the dependence patterns, thanks to the connection with Gaussian graphical models. Additionally, the group lasso penalty on the rows of the component mean matrices allows to perform variable selection in the situation where the three-way data arise from variables recorded over multiple occasions. This increases even further the model parsimony and provides useful indications regarding the ability of the variables in separating the clusters across the occasions. Assessments on both synthetic data and data concerning crime rates in the US have shown the validity of our proposed method for sparse model-based clustering of three-way data, overcoming some of the drawbacks of the approaches currently present in the literature.

The paper leaves several paths open for future research. Firstly, while effectively performing variable selection, the group lasso penalty, could be quite rigid in some applications. In fact, as highlighted in Section 3.1, this specification sets to zero entire rows of the mixture

component mean matrices. Nonetheless, sometimes sparsity could be desirable also within the rows, thus enforcing only some elements and not the entire variable to be shrunk to zero. This could be achieved by adapting the so-called sparse group lasso (Simon et al., 2013) to the framework considered in our work. In fact, this penalty is a convex combination of the group-lasso and the entry-wise lasso penalty briefly described in Section 3.3, and it could extend the application of the proposed framework to other contexts. Throughout the manuscript, matrix Gaussian mixture models have been parameterized in terms of precision matrices. Nonetheless, the penalized approach can be adapted to a setting where sparsity is imposed on the covariance matrices, thus generalizing the work by Fop et al. (2019) to the matrix-variate case. This approach would still lend itself to a convenient representation in terms of the so-called *covariance graphs*, where a missing edge between two nodes implies that the corresponding variables are marginally independent (Chaudhuri et al., 2007). Furthermore, in this work we focused on matrix Gaussian distributions, since they are a widely adopted choice to model continuous data. Nonetheless, it would be interesting to explore if the proposed penalized method could be employed in conjunction with other choices for the component densities, potentially encompassing situations with heavy-tails or skewness (see e.g., Melnykov and Zhu, 2018; Tomarchio et al., 2020). Lastly, alternative model selection strategies might be devised. The adopted grid search produced good results in our numerical assessments. Nonetheless, as mentioned in Section 3.4, it could be too computationally demanding in some applications. For this reason, stochastic optimization techniques could be borrowed and adapted to our setting, as well as the so-called E-MS algorithm introduced by Jiang et al. (2015).

As a final worthy observation, we noted that even in the matrix-variate scenario, the works focusing on precision matrices estimation in multi-class settings often enforce similarities between the underlying graphical models (Huang and Chen, 2015). This assumption, reasonable in different applications, could deteriorate the quality of the results when clustering is the final aim. Therefore, we believe that the strategy adopted in Casa et al. (2022) could be combined with the procedure proposed in this paper, to encompass those situations where different component precision matrices have markedly different degrees of

sparsity.

Acknowledgments

Andrea Cappozzo acknowledges the support by MUR, grant Dipartimento di Eccellenza 2023-2027.

Conflicts of interest

The authors report there are no competing interests to declare.

SUPPLEMENTARY MATERIAL

The supplementary material reports the proof of Proposition 1.

Proof of Proposition 1. For easing the notation, we subsequently drop the “hat” from any parameter estimate and, without loss of generality, we prove the result for \mathbf{P}_1 equal to an all-ones matrix. Similarly to the case outlined in Theorem 1 of Zhou et al. (2009), $Q_M(\mathbf{M}_k)$ is differentiable with respect to $m_{ls,k}$ when $m_{ls,k} \neq 0$, while non-differentiable at $m_{ls,k} = 0$.

The following two cases are considered:

1. If $m_{ls,k} \neq 0$ is a maximum, given that $Q_M(\mathbf{M}_k)$ is concave and differentiable, the sufficient and necessary condition for $m_{ls,k}$ to be the global maximum of $Q_M(\mathbf{M}_k)$ is

$$\frac{\partial Q_M(\mathbf{M}_k)}{\partial m_{ls,k}} = 0 \iff \sum_{i=1}^n z_{ik} \sum_{r=1}^p \sum_{c=1}^q \omega_{lr,k} x_{rc,i} \gamma_{cs,k} - n_k \sum_{r=1}^p \sum_{c=1}^q \omega_{lr,k} m_{rc,k} \gamma_{cs,k} - \lambda_1 \text{sign}(m_{ls,k}) = 0 \quad (23)$$

from which (20) is easily derived by solving (23) with respect to $m_{ls,k}$.

2. If $m_{ls,k} = 0$ is a maximum, we compare $Q_M(0, \cdot)$ with $Q_M(\Delta m_{ls,k}, \cdot)$, the values of $Q_M(\mathbf{M}_k)$ at $m_{ls,k} = 0$ and $m_{ls,k} = \Delta m_{ls,k}$ respectively (while the other entries of \mathbf{M}_k are fixed at their maximum). By definition, we have $Q_M(0, \cdot) \geq Q_M(\Delta m_{ls,k}, \cdot)$ for

any $\Delta m_{ls,k}$ near 0

\iff

$$\sum_{i=1}^n z_{ik} \left[-2 \operatorname{tr} \left\{ \Omega_k \mathbf{X}_i \Gamma_k \mathbf{M}'_k \right\} \Big|_{m_{ls,k}=\Delta m_{ls,k}} + \operatorname{tr} \left\{ \Omega_k \mathbf{M}_k \Gamma_k \mathbf{M}'_k \right\} \Big|_{m_{ls,k}=\Delta m_{ls,k}} + \right. \\ \left. + 2 \operatorname{tr} \left\{ \Omega_k \mathbf{X}_i \Gamma_k \mathbf{M}'_k \right\} \Big|_{m_{ls,k}=0} - \operatorname{tr} \left\{ \Omega_k \mathbf{M}_k \Gamma_k \mathbf{M}'_k \right\} \Big|_{m_{ls,k}=0} \right] \geq -2\lambda_1 |\Delta m_{ls,k}|$$

\iff

$$\sum_{i=1}^n z_{ik} \left[-2 \left(\operatorname{tr} \left\{ \Omega_k \mathbf{X}_i \Gamma_k \mathbf{M}'_k \right\} \Big|_{m_{ls,k}=\Delta m_{ls,k}} - \Omega_k \mathbf{X}_i \Gamma_k \mathbf{M}'_k \Big|_{m_{ls,k}=0} \right) + \right. \\ \left. + \operatorname{tr} \left\{ \Omega_k \mathbf{M}_k \Gamma_k \mathbf{M}'_k \right\} \Big|_{m_{ls,k}=\Delta m_{ls,k}} - \Omega_k \mathbf{M}_k \Gamma_k \mathbf{M}'_k \Big|_{m_{ls,k}=0} \right] \geq -2\lambda_1 |\Delta m_{ls,k}|$$

\iff

$$\sum_{i=1}^n z_{ik} \left[-2 \left[\sum_{r=1}^p \omega_{rl,k} \left(\sum_{c=1}^q x_{rc,i} \gamma_{cs,k} \right) \Delta m_{ls,k} \right] + \right. \\ \left. 2 \sum_{\substack{r=1 \\ r \neq l}}^p \omega_{rl,k} \left(\sum_{c=1}^q m_{rc,k} \gamma_{cs,k} \right) \Delta m_{ls,k} + \omega_{ll,k} \left(\Delta m_{ls,k} \gamma_{ss,k} + 2 \sum_{\substack{c=1 \\ c \neq s}}^q m_{lc,k} \gamma_{cs,k} \right) \Delta m_{ls,k} \right] \geq -2\lambda_1 |\Delta m_{ls,k}|$$

\iff

$$\sum_{i=1}^n z_{ik} \left[2 \left[\sum_{\substack{r=1 \\ r \neq l}}^p \omega_{rl,k} \left(\sum_{c=1}^q x_{rc,i} \gamma_{cs,k} \right) \Delta m_{ls,k} \right] + 2 \omega_{ll,k} \left(\sum_{c=1}^q x_{lc,i} \gamma_{cs,k} \right) \Delta m_{ls,k} - \right. \\ \left. 2 \sum_{\substack{r=1 \\ r \neq l}}^p \omega_{rl,k} \left(\sum_{c=1}^q m_{rc,k} \gamma_{cs,k} \right) \Delta m_{ls,k} - \omega_{ll,k} \left(\Delta m_{ls,k} \gamma_{ss,k} + 2 \sum_{\substack{c=1 \\ c \neq s}}^q m_{lc,k} \gamma_{cs,k} \right) \Delta m_{ls,k} \right] \leq 2\lambda_1 |\Delta m_{ls,k}|$$

\iff

$$\sum_{i=1}^n z_{ik} \left[2 \Delta m_{ls,k} \left[\sum_{\substack{r=1 \\ r \neq l}}^p \omega_{rl,k} \left(\sum_{c=1}^q (x_{rc,i} - m_{rc,k}) \gamma_{cs,k} \right) \right] + 2 \omega_{ll,k} \left(\sum_{c=1}^q x_{lc,i} \gamma_{cs,k} \right) \Delta m_{ls,k} - \right. \\ \left. \omega_{ll,k} \left(\Delta m_{ls,k} \gamma_{ss,k} + 2 \sum_{\substack{c=1 \\ c \neq s}}^q m_{lc,k} \gamma_{cs,k} \right) \Delta m_{ls,k} \right] \leq 2\lambda_1 |\Delta m_{ls,k}|$$

$$\begin{aligned}
& \iff \\
& \left| \sum_{i=1}^n z_{ik} \left[\sum_{\substack{r=1 \\ r \neq l}}^p \omega_{rl,k} \left(\sum_{c=1}^q (x_{rc,i} - m_{rc,k}) \gamma_{cs,k} \right) + \omega_{ll,k} \left(\sum_{c=1}^q x_{lc,i} \gamma_{cs,k} \right) \right. \right. \\
& \quad \left. \left. - \omega_{ll,k} \left(\frac{\Delta m_{ls,k}}{2} \gamma_{ss,k} + \sum_{\substack{c=1 \\ c \neq s}}^q m_{lc,k} \gamma_{cs,k} \right) \right] \right| \leq \lambda_1 \\
& \iff \\
& \left| \sum_{i=1}^n z_{ik} \left[\sum_{\substack{r=1 \\ r \neq l}}^p \omega_{rl,k} \left(\sum_{c=1}^q (x_{rc,i} - m_{rc,k}) \gamma_{cs,k} \right) + \right. \right. \\
& \quad \left. \left. \omega_{ll,k} \left(\sum_{\substack{c=1 \\ c \neq s}}^q (x_{lc,i} - m_{lc,k}) \gamma_{cs,k} \right) + \omega_{ll,k} x_{ls,i} \gamma_{cs,k} \right] \right| \leq \lambda_1 \text{ as } \Delta m_{ls,k} \rightarrow 0
\end{aligned}$$

□

References

- Anderlucci, L. and Viroli, C. (2015). Covariance pattern mixture models for the analysis of multivariate heterogeneous longitudinal data. *The Annals of Applied Statistics*, 9(2):777–800.
- Banerjee, O., El Ghaoui, L., and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516.
- Banfield, J. D. and Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49(3):803.
- Basford, K. E. and McLachlan, G. J. (1985). The mixture method of clustering applied to three-way data. *Journal of Classification*, 2:109–125.
- Bien, J. and Tibshirani, R. J. (2011). Sparse estimation of a covariance matrix. *Biometrika*, 98(4):807–820.

- Bouveyron, C. and Brunet-Saumard, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis*, 71:52–78.
- Bouveyron, C., Celeux, G., Murphy, T. B., and Raftery, A. E. (2019). *Model-based clustering and classification for data science: with applications in R*. Cambridge University Press.
- Casa, A., Cappozzo, A., and Fop, M. (2022). Group-wise shrinkage estimation in penalized model-based clustering. *Journal of Classification*, 39(3):648–674.
- Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793.
- Chaudhuri, S., Drton, M., and Richardson, T. S. (2007). Estimation of a covariance matrix with zeros. *Biometrika*, 94(1):199–216.
- Chen, J. T. and Gupta, A. K. (2005). Matrix variate skew normal distributions. *Statistics*, 39(3):247–253.
- Chen, X. and Liu, W. (2019). Graph estimation for matrix-variate Gaussian data. *Statistica Sinica*, 29:479–504.
- Dawid, A. P. (1981). Some matrix-variate distribution theory: Notational considerations and a Bayesian application. *Biometrika*, 68(1):265–274.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.
- Erdős, P. and Rényi, A. (1960). On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, 5(1):17–60.
- Ferraccioli, F. and Menardi, G. (2023). Modal clustering of matrix-variate data. *Advances in Data Analysis and Classification*, 17:323–345.

- Fop, M. and Murphy, T. B. (2018). Variable selection methods for model-based clustering. *Statistics Surveys*, 12:18–65.
- Fop, M., Murphy, T. B., and Scrucca, L. (2019). Model-based clustering with sparse covariance matrices. *Statistics and Computing*, 29(4):791–819.
- Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association*, 97:611–631.
- Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.
- Gallaugher, M. P. and McNicholas, P. D. (2018). Finite mixtures of skewed matrix variate distributions. *Pattern Recognition*, 80:83–93.
- Gao, X., Shen, W., Zhang, L., Hu, J., Fortin, N. J., Frostig, R. D., and Ombao, H. (2021). Regularized matrix data clustering and its application to image analysis. *Biometrics*, 77(3):890–902.
- Glanz, H. and Carvalho, L. (2018). An expectation–maximization algorithm for the matrix normal distribution with an application in remote sensing. *Journal of Multivariate Analysis*, 167:31–48.
- Green, P. J. (1990). On use of the EM for penalized likelihood estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 443–452.
- Gupta, A. K. and Nagar, D. K. (2018). *Matrix variate distributions*, volume 104. CRC Press.
- Hastie, T., Tibshirani, R., and Wainwright, M. (2019). *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC.
- Heo, J. and Baek, J. (2021). A penalized matrix normal mixture model for clustering matrix data. *Entropy*, 23(10):1249.
- Holland, J. H. (1992). Genetic algorithms. *Scientific American*, 267(1):66–73.

- Huang, F. and Chen, S. (2015). Joint Learning of Multiple Sparse Matrix Gaussian Graphical Models. *IEEE Transactions on Neural Networks and Learning Systems*, 26(11):2606–2620.
- Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.
- Jiang, J., Nguyen, T., and Rao, J. S. (2015). The E-MS algorithm: model selection with incomplete data. *Journal of the American Statistical Association*, 110(511):1136–1147.
- Keribin, C. (2000). Consistent estimation of the order of mixture models. *Sankhyā: The Indian Journal of Statistics, Series A*, 62(1):49–66.
- Klosa, J., Simon, N., Westermark, P. O., Liebscher, V., and Wittenburg, D. (2020). Seagull: lasso, group lasso and sparse-group lasso regularization for linear regression models via proximal gradient descent. *BMC Bioinformatics*, 21(1):407.
- Leng, C. and Tang, C. (2012). Sparse matrix graphical models. *Journal of the American Statistical Association*, 107(499):1187–1200.
- Lian, H. (2011). Shrinkage tuning parameter selection in precision matrices estimation. *Journal of Statistical Planning and Inference*, 141(8):2839–2848.
- Liu, D., Zhao, C., He, Y., Liu, L., Guo, Y., and Zhang, X. (2022). Simultaneous cluster structure learning and estimation of heterogeneous graphs for matrix-variate fMRI data. *Biometrics*, Online.
- Maugis, C., Celeux, G., and Martin-Magniette, M.-L. (2009). Variable selection in model-based clustering: A general variable role modeling. *Computational Statistics & Data Analysis*, 53(11):3872–3882.
- McLachlan, G. J. and Krishnan, T. (2008). *The EM Algorithm and Extensions*, volume 54 of *Wiley Series in Probability and Statistics*. John Wiley & Sons, Inc.

- Melnykov, V., Sarkar, S., and Melnykov, Y. (2021). On finite mixture modeling and model-based clustering of directed weighted multilayer networks. *Pattern Recognition*, 112:107641.
- Melnykov, V. and Zhu, X. (2018). On model-based clustering of skewed matrix data. *Journal of Multivariate Analysis*, 167:181–194.
- Melnykov, V. and Zhu, X. (2019). Studying crime trends in the USA over the years 2000–2012. *Advances in Data Analysis and Classification*, 13(1):325–341.
- Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2020). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071)*, TU Wien. R package version 1.7-4.
- Mosci, S., Rosasco, L., Santoro, M., Verri, A., and Villa, S. (2010). Solving Structured Sparsity Regularization with Proximal Methods. In Balcázar, J. L., Bonchi, F., Gionis, A., and Sebag, M., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 418–433, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Obozinski, G., Wainwright, M. J., and Jordan, M. I. (2009). High-dimensional support union recovery in multivariate regression. *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference*, pages 1217–1224.
- Obozinski, G., Wainwright, M. J., and Jordan, M. I. (2011). Support union recovery in high-dimensional multivariate regression. *Annals of Statistics*, 39(1):1–47.
- Pan, W. and Shen, X. (2007). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, 8:1145–1164.
- Parikh, N. and Boyd, S. (2014). Proximal Algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239.
- R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

- Roeder, K. and Wasserman, L. (1997). Practical Bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association*, 92(439):894–902.
- Sarkar, S., Zhu, X., Melnykov, V., and Ingrassia, S. (2020). On parsimonious models for modeling matrix data. *Computational Statistics & Data Analysis*, 142:106822.
- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.
- Scrucca, L., Fop, M., Murphy, T. B., and Raftery, A. E. (2016). mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models. *The R Journal*, 8(1):289–317.
- Scrucca, L. and Raftery, A. E. (2015). Improved initialisation of model-based clustering using Gaussian hierarchical partitions. *Advances in Data Analysis and Classification*, 9(4):447–460.
- Sharp, A., Chalatov, G., and Browne, R. P. (2022). A dual subspace parsimonious mixture of matrix normal distributions. *Advances in Data Analysis and Classification*.
- Silva, A., Qin, X., Rothstein, S. J., McNicholas, P. D., and Subedi, S. (2023). Finite mixtures of matrix variate poisson-log normal distributions for three-way count data. *Bioinformatics*, 39(5):btad167.
- Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *Journal of computational and graphical statistics*, 22(2):231–245.
- Subedi, S. (2023). Clustering matrix variate longitudinal count data. *Analytics*, 2(2):426–437.
- Sustik, M. A., Calderhead, B., and Clavel, J. (2018). *glassoFast: Fast Graphical LASSO*. R package version 1.0.
- Tomarchio, S. D. (2022). Matrix-variate normal mean-variance birnbaum-saunders distributions and related mixture models. *Computational Statistics*, pages 1–28.

- Tomarchio, S. D., Gallaugher, M. P., Punzo, A., and McNicholas, P. D. (2022). Mixtures of Matrix-Variate Contaminated Normal Distributions. *Journal of Computational and Graphical Statistics*, 31(2):413–421.
- Tomarchio, S. D., Punzo, A., and Bagnato, L. (2020). Two new matrix-variate distributions with application in model-based clustering. *Computational Statistics & Data Analysis*, 152:107050.
- Vichi, M. (1999). One-mode classification of a three-way data matrix. *Journal of Classification*, 16(1):27–44.
- Vichi, M., Rocci, R., and Kiers, H. A. (2007). Simultaneous component and clustering models for three-way data: within and between approaches. *Journal of Classification*, 24(1):71–98.
- Viroli, C. (2011a). Finite mixtures of matrix normal distributions for classifying three-way data. *Statistics and Computing*, 21(4):511–522.
- Viroli, C. (2011b). Model based clustering for three-way data structures. *Bayesian Analysis*, 6(4):573–602.
- Viroli, C. (2012). On matrix-variate regression analysis. *Journal of Multivariate Analysis*, 111:296–309.
- Wang, Y. and Melnykov, V. (2020). On variable selection in matrix mixture modelling. *Stat*, 9(1):e278.
- Whittaker, J. (1990). *Graphical Models in Applied Multivariate Statistics*. Wiley.
- Witten, D. M., Friedman, J. H., and Simon, N. (2011). New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4):892–900.
- Wright, S. J. (2015). Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34.

- Yin, F., Hu, G., and Shen, W. (2023). Analysis of professional basketball field goal attempts via a bayesian matrix clustering approach. *Journal of Computational and Graphical Statistics*, 32(1):49–60.
- Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67.
- Zhou, H., Pan, W., and Shen, X. (2009). Penalized model-based clustering with unconstrained covariance matrices. *Electronic Journal of Statistics*, 3:1473–1496.
- Zhu, X., Sarkar, S., and Melnykov, V. (2022). MatTransMix: an R Package for Matrix Model-Based Clustering and Parsimonious Mixture Modeling. *Journal of Classification*, 39(1):147–170.
- Zou, H., Hastie, T., and Tibshirani, R. (2007). On the “degrees of freedom” of the lasso. *The Annals of Statistics*, 35(5):2173–2192.

Claire Gormley

Material list:

Melnykov V., Melnykov I. and Semhar M. (2016) Semi-supervised model-based clustering with positive and negative constraints. *Advances in Data Analysis and Classification*, 10, 327–349.

Babu G., Gowen A., Fop M. and Gormley I. C. (2024) A consensus-constrained parsimonious Gaussian mixture model for clustering hyperspectral images. arXiv:2403.03349.

Semi-supervised model-based clustering with positive and negative constraints

Volodymyr Melnykov · Igor Melnykov ·
Semhar Michael

Received: 18 February 2014 / Revised: 24 January 2015 / Accepted: 9 February 2015 /
Published online: 25 February 2015
© Springer-Verlag Berlin Heidelberg 2015

Abstract Cluster analysis is a popular technique in statistics and computer science with the objective of grouping similar observations in relatively distinct groups generally known as clusters. Semi-supervised clustering assumes that some additional information about group memberships is available. Under the most frequently considered scenario, labels are known for some portion of data and unavailable for the rest of observations. In this paper, we discuss a general type of semi-supervised clustering defined by so called positive and negative constraints. Under positive constraints, some data points are required to belong to the same cluster. On the contrary, negative constraints specify that particular points must represent different data groups. We outline a general framework for semi-supervised clustering with constraints naturally incorporating the additional information into the EM algorithm traditionally used in mixture modeling and model-based clustering. The developed methodology is illustrated on synthetic and classification datasets. A dendrochronology application is considered and thoroughly discussed.

Keywords Semi-supervised clustering · Model-based clustering · Finite mixture models · Positive and negative constraints · BIC

Mathematics Subject Classification 62H30

V. Melnykov (✉) · S. Michael
Department of Information Systems, Statistics, and Management Science,
University of Alabama, Tuscaloosa, AL 35487, USA
e-mail: vmelnykov@ua.edu

I. Melnykov
Department of Mathematics and Physics, Colorado State
University-Pueblo, Pueblo, CO 81001, USA

1 Introduction

The goal of cluster analysis is to locate distinct groups of similar observations within a data set (Fralej and Raftery 1998, 2002). The rise in the number of clustering applications and the necessity to deal with increasingly large amounts of data motivated the interest in improving existing algorithms and methods as well as development of new techniques. The procedures of one class, called hierarchical methods, utilize distances between data groups for merging or splitting certain clusters at each step of the procedure (Ward 1963; Johnson 1967). Other clustering procedures are oriented towards the optimization of a dataset partition. The most common representative of this group of methods is the k -means algorithm that uses the within-cluster sum of squares as an optimization criterion and modifies the partition over a number of iterations (MacQueen 1967; Forgy 1965). A large class of methods interprets each cluster as a set of observations from a particular probability distribution. Then, the clustering task involves the choice of specific probability distributions to be used in the model as well as parameter estimation for these distributions.

Semi-supervised clustering comprises a variety of methods that operate in the presence of some restrictions on point membership in classes (Basu et al. 2002, 2004; Huang and Hasegawa-Johnson 2009). For example, the labels of certain data points may be available and these points can be immediately used for inference about the classes that they belong to. However, the data with known labels can be expensive or difficult to obtain; for instance, such data may be supplied by a human expert. As a result, in the majority of applications, the amount of labeled data is considerably smaller than that of unlabeled observations (Basu et al. 2002). As we will further discuss in this paper, specifying the membership of points can be problematic if it is unclear which class is associated with a particular set of labels. For a massive resource of ideas on constrained clustering algorithms and their implementation, we refer the reader to a recent book by Basu et al. (2008).

The restriction on class membership can come in the form of a requirement that some specified points need to be included in the same cluster, a so called positive constraint. On the other hand, in a negative constraint, certain points are required to belong to different clusters (Shental et al. 2003). A modification of the K -means algorithm involving both types of restrictions was considered by Wagstaff et al. (2001). Positive and negative constraints provide useful information in a large variety of practical applications (Huang and Hasegawa-Johnson 2009; Nigam et al. 2000), but there are more potential difficulties in their implementation than in the case of predetermined labels, as constraints can lead to complex structures within a dataset. Some examples of applications of these models include image processing techniques (Shental et al. 2003; Martinez-Uso et al. 2010), speech recognition (Digalakis et al. 1995), genome association and sequencing studies (Liu et al. 2013) as well as gene microarray analysis (Wang et al. 2007). Positive constraints are easier to accommodate in computation thanks to their transitivity, i.e., should we know that points x_i and x_j belong to the same cluster and x_j and x_k are in the same cluster, it can be concluded that x_i and x_k are also joined together. As negative constraints are not transitive, their treatment requires more sophisticated methods that can also be more demanding in terms of computational resources. Thus, Shental et al. (2003) suggested an approximate proce-

dure using a Markov network to accommodate such constraints. In the framework of semi-supervised clustering, researchers also consider models that put restrictions on class memberships in the form of weights or penalties for inclusion of observations into a certain cluster. By implementing such “soft” constraints, these models try to achieve more flexibility in finding the clustering solution (Lu and Leen 2007; Law et al. 2005; Pan et al. 2006). In the present work, we do not consider such scenarios and focus on strict constraints.

We propose a novel model that naturally incorporates the information about positive and negative constraints in the expectation-maximization (EM) algorithm for the inference on finite mixture models. The structure of the paper is as follows. In Sect. 2, we consider the use of model-based clustering in the presence of constraints, develop the necessary notation, and introduce the modifications to the EM algorithm compared to modeling without constraints. Section 3 contains examples that illustrate the use of our method on synthetic as well as real-life classification datasets. A thorough study of an application of constraints to dendrochronology is described in Sect. 4. The paper is concluded with a discussion.

2 Methodology

2.1 Mixture modeling and model-based clustering

Let $X = \{X_1, X_2, \dots, X_n\}$ be a sample of size n consisting of independent random vectors distributed according to the probability distribution function

$$f(\mathbf{x}_i | \Psi) = \sum_{k=1}^K \tau_k f_k(\mathbf{x}_i | \vartheta_k),$$

where $f_k(\cdot | \vartheta_k)$ represents the k th mixture component with a corresponding parameter vector ϑ_k , K is the total number of components involved in the mixture, and $\tau_k > 0$ is the k th mixing proportion such that $\sum_{k=1}^K \tau_k = 1$. The overall parameter vector Ψ involves all model parameters that need to be estimated including mixing proportions and component-specific parameters. Although there are several estimation procedures discussed in literature, the most popular approach is the maximum likelihood estimation. The straightforward maximization of the likelihood function

$$\mathcal{L}(\Psi | \mathbf{x}) = \prod_{i=1}^n \sum_{k=1}^K \tau_k f_k(\mathbf{x}_i | \vartheta_k) \quad (1)$$

is often troublesome due to inconvenient functional form. An appealing alternative is to employ a general procedure commonly used in problems involving missing information which is called the EM algorithm (Dempster et al. 1977). The EM algorithm is an iterative scheme consisting of two steps (called E and M) and relying on the notion of the complete-data likelihood function, given in the mixture modeling framework by

$$\mathcal{L}_c(\Psi | \mathbf{x}, \mathbf{Z}) = \prod_{i=1}^n \prod_{k=1}^K [\tau_k f_k(\mathbf{x}_i | \boldsymbol{\vartheta}_k)]^{I(Z_i=k)}, \quad (2)$$

where $\mathbf{Z} = (Z_1, Z_2, \dots, Z_n)'$ is the classification vector assumed to be missing and $I(Z_i = k)$ is the indicator function equal to 1 if k represents the correct label for the i th observation and equal to 0 otherwise. The conditional expectation of the log function of (2) given the observed data is calculated at the E-step of the EM algorithm. This expectation is usually denoted as the Q -function and is given by

$$Q(\Psi | \dot{\Psi}, \mathbf{x}) = \sum_{i=1}^n \sum_{k=1}^K \ddot{\pi}_{ik} \{\log \tau_k + \log f_k(\mathbf{x}_i | \boldsymbol{\vartheta}_k)\}, \quad (3)$$

where $\ddot{\pi}_{ik} = E(Z_i = k | \mathbf{x}_i, \dot{\Psi})$ is the current estimate of the posterior probability based on the parameter vector estimate $\dot{\Psi}$ obtained from the previous iteration; here, two dots represent the current iteration estimate and one dot denotes the previous one. It can be shown that the E-step of the EM algorithm reduces to the estimation of posterior probabilities π_{ik} by

$$\ddot{\pi}_{ik} = \frac{\dot{\tau}_k f_k(\mathbf{x}_i | \dot{\boldsymbol{\vartheta}}_k)}{\sum_{k'=1}^K \dot{\tau}_{k'} f_{k'}(\mathbf{x}_i | \dot{\boldsymbol{\vartheta}}_{k'})}.$$

The M-step of the EM algorithm involves maximizing the Q -function (3) with respect to Ψ . The E- and M-steps then are iterated till the convergence is reached. In the case of the mixture with multivariate Gaussian components with unrestricted covariance matrices, the E-step is given by

$$\ddot{\pi}_{ik} = \frac{\dot{\tau}_k \phi(\mathbf{x}_i | \dot{\boldsymbol{\mu}}_k, \dot{\Sigma}_k)}{\sum_{k'=1}^K \dot{\tau}_{k'} \phi(\mathbf{x}_i | \dot{\boldsymbol{\mu}}_{k'}, \dot{\Sigma}_{k'})},$$

while the M-step is implemented using the expressions

$$\ddot{\tau}_k = \frac{\sum_{i=1}^n \ddot{\pi}_{ik}}{n}, \quad \ddot{\boldsymbol{\mu}}_k = \frac{\sum_{i=1}^n \ddot{\pi}_{ik} \mathbf{x}_i}{\sum_{i=1}^n \ddot{\pi}_{ik}}, \quad \ddot{\Sigma}_k = \frac{\sum_{i=1}^n \ddot{\pi}_{ik} (\mathbf{x}_i - \ddot{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \ddot{\boldsymbol{\mu}}_k)'}{\sum_{i=1}^n \ddot{\pi}_{ik}}, \quad (4)$$

where $\phi(\cdot | \boldsymbol{\mu}_k, \Sigma_k)$ is the k th Gaussian density function with the corresponding mean vector $\boldsymbol{\mu}_k$ and covariance matrix Σ_k .

The number of components in the mixture can be either pre-specified or unknown depending on the nature of a particular problem. In the latter case, Bayesian information criterion (BIC) (Schwarz 1978) is a popular choice for selecting K due to its good performance in the mixture modeling framework. A traditional convergence criterion is based on the relative log likelihood change: $(\log \mathcal{L}(\ddot{\Psi}) - \log \mathcal{L}(\dot{\Psi})) / |\log \mathcal{L}(\ddot{\Psi})| < \epsilon$, where ϵ is the required tolerance level. When the convergence is reached, the last iteration of the EM algorithm yields the maximum likelihood parameter estimate $\hat{\Psi}$ and

posterior probability estimate $\hat{\pi}_{ik}$. Then, observation memberships can be estimated according to the Bayes decision rule, i.e., using $\hat{Z}_i = \operatorname{argmax}_k \hat{\pi}_{ik}$. This relationship provides the connection between finite mixture modeling and model-based clustering.

2.2 Model-based clustering with constraints

Now, we focus on the situation when additional information about class membership is available for some part of data. It has been discussed in literature (McLachlan and Peel 2000; Côme et al. 2009; Melnykov and Maitra 2010) that the presence of some labels does not affect the flavor of the EM algorithm and model-based clustering: posterior probabilities still have to be obtained for unlabeled observations while the membership for the rest of the data is known with certainty as labels z_i are provided. However, such an approach has a flaw—the membership of data points is not available with certainty before the EM algorithm is run. Figure 1 illustrates this idea. There are two clusters simulated from two Gaussian components. The majority of observations are unlabeled except the three solid points from the black component, whose membership is assumed available. In plot (a), clusters are well-separated and it is trivial to identify the solution with no misclassifications by running semi-supervised clustering. In plot (b), we move both data groups towards each other and rerun the EM algorithm. As we can see, the membership of the three solid points has changed as they are fitted better by the other component. This example suggests that, unless in a trivial case, it can be dangerous to assign points to components as the components are yet to be formed. In other words, specifying labels, i.e., saying that some points belong to a special component, say #1, with probability 1, is often questionable as it is even unknown which component will be denoted as #1. This suggests that some other approaches to specifying the information about labels should be considered.

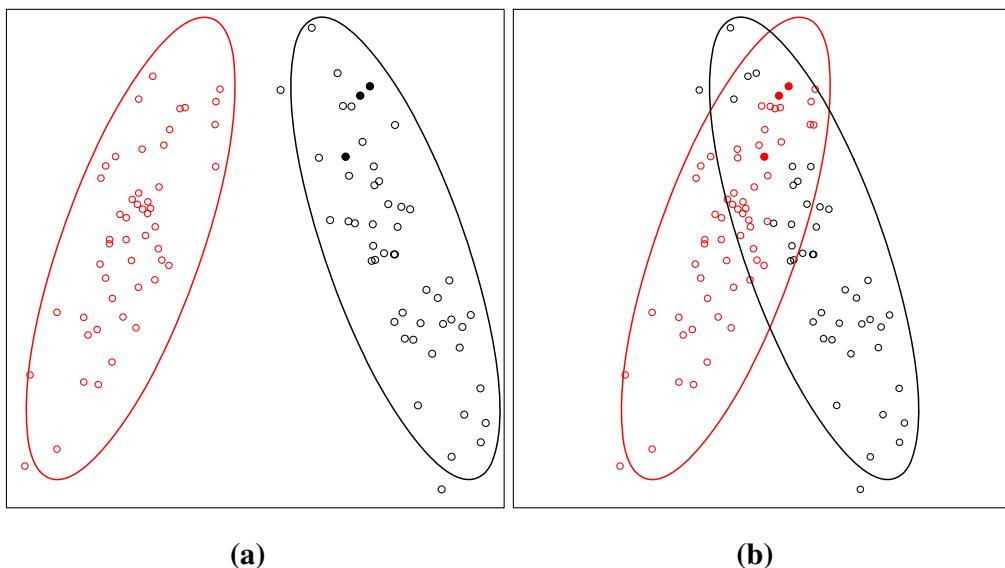


Fig. 1 Different model-based clustering solutions for **a** well-separated and **b** overlapping clusters. Red and black colors illustrate the assignments obtained by model-based clustering. Solid points represent observations with known labels. Ellipsoids provide 95 % confidence regions associated with the original covariance matrices of the two components (color figure online)

Suppose some relations among observations are available in addition to the data sample. Some observations must belong to the same mixture component (positive constraint), some others, on the contrary, can be prohibited from originating in a common component (negative constraint). In this case, the likelihood function provided in Eq. (1) is not directly applicable as it does not account for the set of existing constraints. In the meantime, it is nearly unrealistic or at least very challenging to specify a valid likelihood function in this setting. Fortunately, this issue can be resolved by employing the EM algorithm discussed in the previous section to find the maximum likelihood estimates in the parameter space restricted by the set of data constraints. The maximized likelihood value obtained this way can be at best equal to that in the unrestricted parameter space. The main distinction from the EM algorithm outlined in Sect. 2.1 is that the observed information in the conditional expectation of the complete-data log likelihood function will be extended by the presence of additional constraints. In other words, the Q -function transforms to $Q(\Psi|\dot{\Psi}, \mathbf{x}, \mathcal{Z})$, where \mathcal{Z} denotes the set of all available restrictions. This modification does not directly affect the derivations associated with the M-step, however, the E-step involving the calculation of $\ddot{\pi}_{ik} = E(Z_i = k|\mathbf{x}, \mathcal{Z}, \dot{\Psi})$ requires substantial changes. In the following sections, we introduce the notation and discuss the modifications of the E-step in cases with positive and negative constraints.

2.3 Notation and terminology

For the matter of notational and mathematical convenience, we introduce the notion of a *block*, which is also equivalent to that of a *chunklet* defined in Shental et al. (2003). The block is defined as a set of data points that are required to belong to the same cluster. Here, it is assumed that each block has its maximum size and cannot be extended any further, i.e., there are no positive relations with observations from other blocks. Let B be the total number of blocks and \mathcal{B}_b represent the set of the id's of data points in the b th block that should be treated jointly. Then, the number of the block the i th observation belongs to can be identified by $b(i) = \arg_b\{I(i \in \mathcal{B}_b) = 1\}$ and, therefore, $\mathcal{B}_{b(i)} = \{j : Z_i = Z_j\}$, where Z_i represents the membership of the i th point. The simplest block involves a single observation not related to other blocks. Therefore, for B blocks, we observe that $\bigcup_{b=1}^B \mathcal{B}_b = \{1, 2, \dots, n\}$ and $\mathcal{B}_b \cap \mathcal{B}_{b'} = \emptyset$ for all $b \neq b'$, where \emptyset denotes the empty set. It can be remarked that the entire structure of positive constraints introduced this way can be seen as a disconnected undirected graph with vertices representing individual observations and edges reflecting positive relations. The components of such a graph (i.e., the connected parts in the disconnected graph) are all complete in the sense that all vertices within every component are connected by adjacent edges (i.e., directly connected by an edge). Thus, the introduced notion of a block is equivalent to a complete component in a disconnected undirected graph.

If there are two blocks with points related to each other by a negative constraint, data points from one block cannot belong to the same cluster with data points from the other block. Thus, the notion of blocks is convenient for formalizing negative restrictions in data. Finally, it is important to remark that the sets of positive and negative constraints must be in agreement, i.e., no contradictions occur as a result of their implementation.

2.4 Estimation with positive constraints

As we discussed in the beginning of Sect. 2.2, in our framework, the EM algorithm involves dealing with the Q -function $Q(\Psi|\dot{\Psi}, \mathbf{x}, \mathcal{Z})$, in particular, calculating $\pi_{ik} = E(Z_i = k|\mathbf{x}, \mathcal{Z})$ at the E-step. Since for now the focus is on positive constraints, we denote the set of all such restrictions as \mathcal{Z}^+ . The corresponding posterior probability π_{ik}^+ can be rewritten as follows:

$$\begin{aligned}\pi_{ik}^+ &= P(Z_i = k|\mathbf{x}, \mathcal{Z}^+) \\ &= \frac{P(Z_i = k, Z_i = Z_j, \forall j \in \mathcal{B}_{b(i)}|\mathbf{x}_j, \forall j \in \mathcal{B}_{b(i)})}{\sum_{k'=1}^K P(Z_i = k', Z_i = Z_j, \forall j \in \mathcal{B}_{b(i)}|\mathbf{x}_j, \forall j \in \mathcal{B}_{b(i)})} \\ &= \frac{P(Z_j = k, \forall j \in \mathcal{B}_{b(i)})P(\mathbf{x}_j, \forall j \in \mathcal{B}_{b(i)}|Z_j = k, \forall j \in \mathcal{B}_{b(i)})}{\sum_{k'=1}^K P(Z_j = k', \forall j \in \mathcal{B}_{b(i)})P(\mathbf{x}_j, \forall j \in \mathcal{B}_{b(i)}|Z_j = k', \forall j \in \mathcal{B}_{b(i)})}.\end{aligned}$$

Based on this expression, it can be noted that $\pi_{ik}^+ = \pi_{jk}^+, \forall j \in \mathcal{B}_{b(i)}$. For notational simplicity, this posterior probability, common for all $j \in \mathcal{B}_b$, can be denoted as π_{bk}^+ and estimated by

$$\ddot{\pi}_{bk}^+ = \frac{\dot{\tau}_k^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} f_k(\mathbf{x}_j; \dot{\vartheta}_k)}{\sum_{k'=1}^K \dot{\tau}_{k'}^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} f_{k'}(\mathbf{x}_j; \dot{\vartheta}_{k'})},$$

where $|\mathcal{B}_b|$ represents the cardinality of \mathcal{B}_b . Since $\ddot{\pi}_{bk}^+$ is the same for all observations within the b th block, all data points that belong to the same block are guaranteed to be assigned to the same mixture component. At the M-step, the Q -function (3) has to be maximized with respect to the parameter vector Ψ . It can be alternatively written as

$$Q(\Psi|\dot{\Psi}, \mathbf{x}, \mathcal{Z}^+) = \sum_{b=1}^B |\mathcal{B}_b| \sum_{k=1}^K \ddot{\pi}_{bk}^+ \log \tau_k + \sum_{b=1}^B \sum_{k=1}^K \ddot{\pi}_{bk}^+ \sum_{j \in \mathcal{B}_b} \log f_k(\mathbf{x}_j; \dot{\vartheta}_k).$$

For the multivariate Gaussian mixture, Eq. (4) can be also rewritten as follows:

$$\begin{aligned}\ddot{\tau}_k &= \frac{\sum_{b=1}^B |\mathcal{B}_b| \ddot{\pi}_{bk}^+}{n}, \quad \ddot{\mu}_k = \frac{\sum_{b=1}^B \ddot{\pi}_{bk}^+ \sum_{j \in \mathcal{B}_b} \mathbf{x}_j}{\sum_{b=1}^B |\mathcal{B}_b| \ddot{\pi}_{bk}^+}, \\ \ddot{\Sigma}_k &= \frac{\sum_{b=1}^B \ddot{\pi}_{bk}^+ \sum_{j \in \mathcal{B}_b} (\mathbf{x}_j - \ddot{\mu}_k)(\mathbf{x}_j - \ddot{\mu}_k)'}{\sum_{b=1}^B |\mathcal{B}_b| \ddot{\pi}_{bk}^+}.\end{aligned}$$

It should be pointed out that Shental et al. (2003) derived identical expressions for posterior probabilities. Unfortunately, due to a complete-data likelihood function misspecification, the authors failed to obtain the solution for mixing proportions recalculated at the M-step.

2.5 Estimation with negative constraints

The situation with negative constraints is not so straightforward. As we discussed before, such constraints can be applied to blocks of observations formed based on positive restrictions. We denote the new set of constraints as \mathcal{Z}^\pm emphasizing that both types of constraints are present. Then, the corresponding posterior probabilities are defined by $\pi_{ik}^\pm = P(Z_i = k | \mathbf{x}, \mathcal{Z}^\pm)$. The process of calculating probabilities π_{ik}^\pm is more complicated than that for probabilities π_{ik}^+ since the result depends on the number of blocks disconnected by negative constraints, the structure of these constraints, as well as the specific location of the block within this structure.

2.5.1 Two blocks

First, we consider the simplest case when there are two blocks disconnected with a negative constraint. Figure 2a provides a graph with two nodes illustrating this setting. Red circles surrounding the nodes, denoted by solid points, imply that expressions for both nodes are symmetric with respect to each other. A crossed line stands for the negative constraint. According to the notation introduced in Sect. 2.3, the set of observations $\{\mathbf{x}_j, \forall j \in \mathcal{B}_{b(i)}\}$ forms a block associated with the i th observation. Then, the other block can be defined as $\{\mathbf{x}_j, \forall j \in \mathcal{B}_{b(i)}^-\}$, where $\mathcal{B}_{b(i)}^- = \{l : Z_i \neq Z_l, Z_l = Z_{l'}, l' \in \{1, 2, \dots, n\}\}$. The posterior probability π_{ik}^\pm can be obtained as follows:

$$\begin{aligned} \pi_{ik}^\pm &= \frac{P(Z_i = k, Z_i = Z_j, \forall j \in \mathcal{B}_{b(i)}, Z_i \neq Z_l, Z_l = Z_{l'}, \forall l, l' \in \mathcal{B}_{b(i)}^- | \mathbf{x}_j, \forall j \in \mathcal{B}_{b(i)}, \mathcal{B}_{b(i)}^-)}{\sum_{k'=1}^K P(Z_i = k', Z_i = Z_j, \forall j \in \mathcal{B}_{b(i)}, Z_i \neq Z_l, Z_l = Z_{l'}, \forall l, l' \in \mathcal{B}_{b(i)}^- | \mathbf{x}_j, \forall j \in \mathcal{B}_{b(i)}, \mathcal{B}_{b(i)}^-)} \\ &= \frac{\sum_{\substack{s=1 \\ s \neq k}}^K P(Z_j = k, \forall j \in \mathcal{B}_{b(i)}, Z_l = s, \forall l \in \mathcal{B}_{b(i)}^- | \mathbf{x}_j, \forall j \in \mathcal{B}_{b(i)}, \mathcal{B}_{b(i)}^-)}{\sum_{\substack{k'=1 \\ s' \neq k'}}^K P(Z_j = k', \forall j \in \mathcal{B}_{b(i)}, Z_l = s', \forall l \in \mathcal{B}_{b(i)}^- | \mathbf{x}_j, \forall j \in \mathcal{B}_{b(i)}, \mathcal{B}_{b(i)}^-)}. \end{aligned}$$

It can be noted that $\pi_{ik}^\pm = \pi_{jk}^\pm, \forall j \in \mathcal{B}_{b(i)}$ (denote this quantity as π_{bk}^\pm for all observations from the b th block) and $P(Z_j = k, \forall j \in \mathcal{B}_{b(i)}, Z_l = s, \forall l \in \mathcal{B}_{b(i)}^- | \mathbf{x}_j, \forall j \in \mathcal{B}_{b(i)}, \mathcal{B}_{b(i)}^-)$ can be written as $P(Z_j = k, \forall j \in \mathcal{B}_{b(i)}, Z_l = s, \forall l \in \mathcal{B}_{b(i)}^-)P(\mathbf{x}_j, \forall j \in \mathcal{B}_{b(i)}, \mathcal{B}_{b(i)}^- | Z_j = k, \forall j \in \mathcal{B}_{b(i)}, Z_l = s, \forall l \in \mathcal{B}_{b(i)}^-)$. Then, it follows that the E-step of the EM algorithm involves estimating posterior probabilities by

$$\ddot{\pi}_{bk}^\pm = \frac{\dot{\tau}_k^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} f_k(\mathbf{x}_j; \dot{\vartheta}_k) \sum_{s=1, s \neq k}^K \dot{\tau}_s^{|\mathcal{B}_b^-|} \prod_{l \in \mathcal{B}_b^-} f_s(\mathbf{x}_l; \dot{\vartheta}_s)}{\sum_{k'=1}^K \dot{\tau}_{k'}^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} f_{k'}(\mathbf{x}_j; \dot{\vartheta}_{k'}) \sum_{s'=1, s' \neq k'}^K \dot{\tau}_{s'}^{|\mathcal{B}_b^-|} \prod_{l \in \mathcal{B}_b^-} f_{s'}(\mathbf{x}_l; \dot{\vartheta}_{s'})}. \quad (5)$$

The M-step of the EM algorithm remains unchanged but involves posterior probabilities $\ddot{\pi}_{bk}^\pm$.

2.5.2 Three blocks

Now, we focus on a slightly more complicated situation with three nodes. The graphs in Fig. 2b–d illustrate the considered cases. There are two possible scenarios for the three-node case. The first one, with two negative constraints, is illustrated in plots (b) and (c), while the other one, with all blocks disconnected through negative constraints, is highlighted in plot (d). First, we derive posterior probabilities for the edge blocks highlighted by red circles in plot (b). For each of the circled blocks, there is only one direct negative constraint. We denote an edge block under consideration as \mathcal{B}_b , the middle block is called \mathcal{B}_b^- , and the remaining block is $\mathcal{B}_b^=$, where “=” emphasizes the fact that there are two negative constraints in between the two edge blocks. Then, applying the logic similar to the one demonstrated in Sect. 2.5.1, it can be shown that $\ddot{\pi}_{bk}^\pm$ for the edge blocks can be calculated by the following expression:

$$\ddot{\pi}_{bk}^\pm = \frac{\dot{\tau}_k^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} f_k(\mathbf{x}_j; \dot{\vartheta}_k) \sum_{s=1}^K \dot{\tau}_s^{|\mathcal{B}_b^-|} \prod_{l \in \mathcal{B}_b^-} f_s(\mathbf{x}_l; \dot{\vartheta}_s) \sum_{h=1, h \neq s}^K \dot{\tau}_h^{|\mathcal{B}_b^=|} \prod_{r \in \mathcal{B}_b^=} f_h(\mathbf{x}_r; \dot{\vartheta}_h)}{\sum_{k'=1}^K \dot{\tau}_{k'}^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} f_{k'}(\mathbf{x}_j; \dot{\vartheta}_{k'}) \sum_{s'=1, s' \neq k'}^K \dot{\tau}_{s'}^{|\mathcal{B}_b^-|} \prod_{l \in \mathcal{B}_b^-} f_{s'}(\mathbf{x}_l; \dot{\vartheta}_{s'}) \sum_{h'=1, h' \neq s'}^K \dot{\tau}_{h'}^{|\mathcal{B}_b^=|} \prod_{r \in \mathcal{B}_b^=} f_{h'}(\mathbf{x}_r; \dot{\vartheta}_{h'})}. \quad (6)$$

For the middle block circled in Fig. 2c, there are two blocks related to it via negative constraints. Let us denote them as \mathcal{B}_b^{-1} and \mathcal{B}_b^{-2} . Then, it can be shown that the posterior probability for the observations that belong to the middle block \mathcal{B}_b is calculated by

$$\ddot{\pi}_{bk}^\pm = \frac{\dot{\tau}_k^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} f_k(\mathbf{x}_j; \dot{\vartheta}_k) \sum_{s=1}^K \dot{\tau}_s^{|\mathcal{B}_b^{-1}|} \prod_{l \in \mathcal{B}_b^{-1}} f_s(\mathbf{x}_l; \dot{\vartheta}_s) \sum_{h=1, h \neq k}^K \dot{\tau}_h^{|\mathcal{B}_b^{-2}|} \prod_{r \in \mathcal{B}_b^{-2}} f_h(\mathbf{x}_r; \dot{\vartheta}_h)}{\sum_{k'=1}^K \dot{\tau}_{k'}^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} f_{k'}(\mathbf{x}_j; \dot{\vartheta}_{k'}) \sum_{s'=1, s' \neq k'}^K \dot{\tau}_{s'}^{|\mathcal{B}_b^{-1}|} \prod_{l \in \mathcal{B}_b^{-1}} f_{s'}(\mathbf{x}_l; \dot{\vartheta}_{s'}) \sum_{h'=1, h' \neq k'}^K \dot{\tau}_{h'}^{|\mathcal{B}_b^{-2}|} \prod_{r \in \mathcal{B}_b^{-2}} f_{h'}(\mathbf{x}_r; \dot{\vartheta}_{h'})}. \quad (7)$$

Fig. 2 Two- and three-node negative constraint graphs. Red circles highlight symmetric nodes for each figure

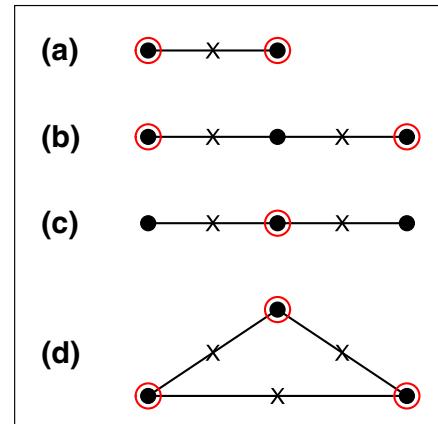
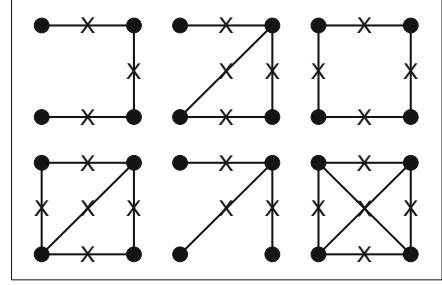


Fig. 3 Six possible four-node negative constraint graphs



Finally, we consider the case displayed in Fig. 2d. All blocks are symmetric, hence the following formula for calculating posterior probabilities can be applied to all of them:

$$\ddot{\pi}_{bk}^{\pm} = \frac{\dot{\tau}_k^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} f_k(\mathbf{x}_j; \dot{\vartheta}_k) \sum_{\substack{s=1 \\ s \neq k}}^K \dot{\tau}_s^{|\mathcal{B}_b^{-1}|} \prod_{l \in \mathcal{B}_b^{-1}} f_s(\mathbf{x}_l; \dot{\vartheta}_s) \sum_{\substack{h=1 \\ h \neq k, h \neq s}}^K \dot{\tau}_h^{|\mathcal{B}_b^{-2}|} \prod_{r \in \mathcal{B}_b^{-2}} f_h(\mathbf{x}_r; \dot{\vartheta}_h)}{\sum_{k'=1}^K \dot{\tau}_{k'}^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} f_{k'}(\mathbf{x}_j; \dot{\vartheta}_{k'}) \sum_{\substack{s'=1 \\ s' \neq k'}}^K \dot{\tau}_{s'}^{|\mathcal{B}_b^{-1}|} \prod_{l \in \mathcal{B}_b^{-1}} f_{s'}(\mathbf{x}_l; \dot{\vartheta}_{s'}) \sum_{\substack{h'=1 \\ h' \neq k', h' \neq s'}}^K \dot{\tau}_{h'}^{|\mathcal{B}_b^{-2}|} \prod_{r \in \mathcal{B}_b^{-2}} f_{h'}(\mathbf{x}_r; \dot{\vartheta}_{h'})}. \quad (8)$$

An important conclusion can be drawn from the comparison of expressions (6)–(8): the only distinction between the expressions is the set of restrictions defined for sums such as $s \neq k$, $h \neq k$, and $h \neq s$. It readily suggests an approach for writing down expressions for more complicated graphs with multiple nodes.

2.5.3 General cases

It can be shown that the number of unique graphs, G_M , with M nodes constructed based on negative relations is equivalent to the number of connected labeled graphs and is given by the sequence 1, 2, 6, 21, 112, 853, ... for $M = 2, 3, 4, 5, 6, 7, \dots$, respectively (Sloane 2014). Figure 3 provides an illustration of 6 unique graphs that can be constructed with 4 nodes. It can be noted based on the material presented in previous sections that the derivation of the expressions for posterior probabilities π_{bk}^{\pm} follows the same principle. The researcher just needs to include the required number of sums and carefully set the restrictions for them. For instance, for the top right node of the middle graph in the first row of Fig. 3, the posterior probability can be calculated by

$$\ddot{\pi}_{bk}^{\pm} = \frac{\dot{\tau}_k^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} \dot{f}_{kj} \sum_{\substack{s=1 \\ s \neq k}}^K \dot{\tau}_s^{|\mathcal{B}_b^{-1}|} \prod_{l \in \mathcal{B}_b^{-1}} \dot{f}_{sl} \sum_{\substack{h=1 \\ h \neq k}}^K \dot{\tau}_h^{|\mathcal{B}_b^{-2}|} \prod_{r \in \mathcal{B}_b^{-2}} \dot{f}_{hr} \sum_{\substack{g=1 \\ g \neq k, g \neq h}}^K \dot{\tau}_g^{|\mathcal{B}_b^{-3}|} \prod_{u \in \mathcal{B}_b^{-3}} \dot{f}_{gu}}{\sum_{k'=1}^K \dot{\tau}_{k'}^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} \dot{f}_{k'j} \sum_{\substack{s'=1 \\ s' \neq k'}}^K \dot{\tau}_{s'}^{|\mathcal{B}_b^{-1}|} \prod_{l \in \mathcal{B}_b^{-1}} \dot{f}_{s'l} \sum_{\substack{h'=1 \\ h' \neq k'}}^K \dot{\tau}_{h'}^{|\mathcal{B}_b^{-2}|} \prod_{r \in \mathcal{B}_b^{-2}} \dot{f}_{h'r} \sum_{\substack{g'=1 \\ g' \neq k', g' \neq h'}}^K \dot{\tau}_{g'}^{|\mathcal{B}_b^{-3}|} \prod_{u \in \mathcal{B}_b^{-3}} \dot{f}_{g'u}},$$

where $\dot{f}_{ac} = f_a(\mathbf{x}_c; \dot{\vartheta}_a)$ and \mathcal{B}_b^{-1} , \mathcal{B}_b^{-2} , and \mathcal{B}_b^{-3} represent the three nodes disconnected from the studied one with \mathcal{B}_b^{-1} being the node in the top left corner.

The complexity of calculations grows rapidly along with the increase in M . Therefore, handling cases with multiple blocks and complex graph structures, although not difficult in realization, can be time consuming.

2.6 Computational aspects

Calculations associated with Eq. (5) and other similar expressions are likely to encounter computational issues due to the presence of elements such as $\dot{\tau}_k^{|\mathcal{B}_b|}, \dot{\tau}_s^{|\mathcal{B}_b^-|}$, $\prod_{j \in \mathcal{B}_b} f_k(\mathbf{x}_j; \dot{\vartheta}_k)$, and $\prod_{l \in \mathcal{B}_b^-} f_s(\mathbf{x}_l; \dot{\vartheta}_s)$. If blocks consist of multiple observations, these terms can be approaching zero. In the meantime, one can rewrite Eq. (5) in the following way:

$$\ddot{\pi}_{bk}^{\pm} = \left[\sum_{k'=1}^K \sum_{\substack{s'=1 \\ s' \neq k'}}^K \left\{ \frac{\sum_{s=1}^K \dot{\tau}_k^{|\mathcal{B}_b|} \dot{\tau}_s^{|\mathcal{B}_b^-|} \prod_{j \in \mathcal{B}_b} f_k(\mathbf{x}_j; \dot{\vartheta}_k) \prod_{l \in \mathcal{B}_b^-} f_s(\mathbf{x}_l; \dot{\vartheta}_s)}{\dot{\tau}_{k'}^{|\mathcal{B}_b|} \dot{\tau}_{s'}^{|\mathcal{B}_b^-|} \prod_{j \in \mathcal{B}_b} f_{k'}(\mathbf{x}_j; \dot{\vartheta}_{k'}) \prod_{l \in \mathcal{B}_b^-} f_{s'}(\mathbf{x}_l; \dot{\vartheta}_{s'})} \right\}^{-1} \right]^{-1}.$$

Then, it readily reduces to

$$\begin{aligned} \ddot{\pi}_{bk}^{\pm} &= \left[\sum_{k'=1}^K \left(\frac{\dot{\tau}_{k'}}{\dot{\tau}_k} \right)^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} \frac{f_{k'}(\mathbf{x}_j; \dot{\vartheta}_{k'})}{f_k(\mathbf{x}_j; \dot{\vartheta}_k)} \sum_{\substack{s'=1 \\ s' \neq k'}}^K \left\{ \sum_{\substack{s=1 \\ s \neq k}}^K \left(\frac{\dot{\tau}_s}{\dot{\tau}_{s'}} \right)^{|\mathcal{B}_b^-|} \prod_{l \in \mathcal{B}_b^-} \frac{f_s(\mathbf{x}_l; \dot{\vartheta}_s)}{f_{s'}(\mathbf{x}_l; \dot{\vartheta}_{s'})} \right\}^{-1} \right]^{-1}. \end{aligned} \quad (9)$$

The main advantage of Eq. (9) over Eq. (5) is that it can be calculated with the use of an approach standard in such situations: $\prod_i \frac{a_i}{b_i} = \exp\{\sum_i (\log a_i - \log b_i)\}$. Hence, severe computational problems can be resolved or at least relaxed. A similar approach to overcoming computational issues can be undertaken in the cases of more complicated negative relation structure.

3 Experimental evaluations

This section is devoted to illustrating the developed semi-supervised methodology and studying its properties. First, we consider several examples in Sect. 3.1. In Sect. 3.2, we assess the computational cost of semi-supervised clustering. The analysis of two famous classification datasets is provided in Sect. 3.3.

3.1 Illustrative example

In this section, we explore the impact of positive and negative constraints on the obtained solutions. A three-component mixture with bivariate Gaussian components was simulated using the R package MIXSIM (Melnikov et al. 2012). MIXSIM simulates random mixture models according to a pre-specified level of maximum and/or average pairwise overlap, where the pairwise overlap between mixture components is defined as the sum of two misclassification probabilities $\omega_{j|k} + \omega_{k|j}$ (Maitra and Melnykov

(2010). Here, we used maximum overlap of 0.10 and average overlap of 0.05. The parameters of the simulated mixture are provided in Table 1. A sample of size 100 was simulated from the obtained mixture. Figure 4a represents the corresponding unsupervised clustering solution, *i.e.*, the solution with no constraints imposed. Solid inscribed characters represent the true classification of observations, while characters around them stand for the estimated membership. The log likelihood value associated with the obtained solution is $\ell = -26.24$. The largest overlap is $\omega_{12} = 0.10$. The total number of misclassified observations is 5. Since the developed semi-supervised procedure searches for a local maximum, subject to constraints, all log likelihood values associated with semi-supervised clustering solutions cannot exceed the value -26.24 . In display (b), we consider implementing two negative constraints between observations 1 and 2 as well as between 3 and 4. Observations 1 and 2 originated from different components but are assigned to the black one, marked with squares, according to unsupervised clustering. If we know that observations 1 and 2 represent different components, we can separate them by a negative relation. Similarly, with observations 3 and 4: if we know their origins, a negative constraint can be established for them.

The provided negative restrictions are shown via crossed lines. As we can see, the addition of even a minor piece of information can cause changes in the obtained solution: compared with the plot (a), 3 observations are classified differently now. Indeed, observations 1 and 2 as well as 3 and 4 represent different mixture components. We also observe some decrease in log likelihood value: it dropped to $\ell = -26.91$. While observations 3 and 4 are classified correctly now, we can see that data points 1 and 2 are both misclassified. This fact suggests that providing additional information can actually lead to an improved as well as degraded classification. Figure 4 also illustrates that providing incorrect positive [plot (c), $\ell = -27.48$] or negative [plot (d), $\ell = -31.73$] constraints can lead to rather substantial changes in the assignments of other points. In both cases, the relations established for observations 5 and 6 as well as 7 and 8 result in multiple misclassifications.

3.2 Assessment of computational cost

In this section, we conduct a brief study of computational costs associated with semi-supervised clustering in the presence of positive and negative constraints. One hundred mixture models involving six components with maximum overlap 0.01 were simulated

Table 1 Parameters and misclassification probabilities of the 3-component mixture model from the illustrative example in Sect. 3.1

k	τ_k	$\boldsymbol{\mu}_k$	$\boldsymbol{\Sigma}_k$	$\omega_{j k}$
1	$\frac{1}{3}$	$\begin{pmatrix} 0.878 \\ 0.949 \end{pmatrix}$	$\begin{pmatrix} 0.040 & 0.010 \\ 0.010 & 0.012 \end{pmatrix}$	$\omega_{2 1} = 0.055$ $\omega_{3 1} = 0.005$
2	$\frac{1}{3}$	$\begin{pmatrix} 0.330 \\ 0.796 \end{pmatrix}$	$\begin{pmatrix} 0.030 & 0.028 \\ 0.028 & 0.083 \end{pmatrix}$	$\omega_{1 2} = 0.045$ $\omega_{3 2} = 0.014$
3	$\frac{1}{3}$	$\begin{pmatrix} 0.626 \\ 0.079 \end{pmatrix}$	$\begin{pmatrix} 0.021 & 0.017 \\ 0.017 & 0.080 \end{pmatrix}$	$\omega_{1 3} = 0.016$ $\omega_{2 3} = 0.015$

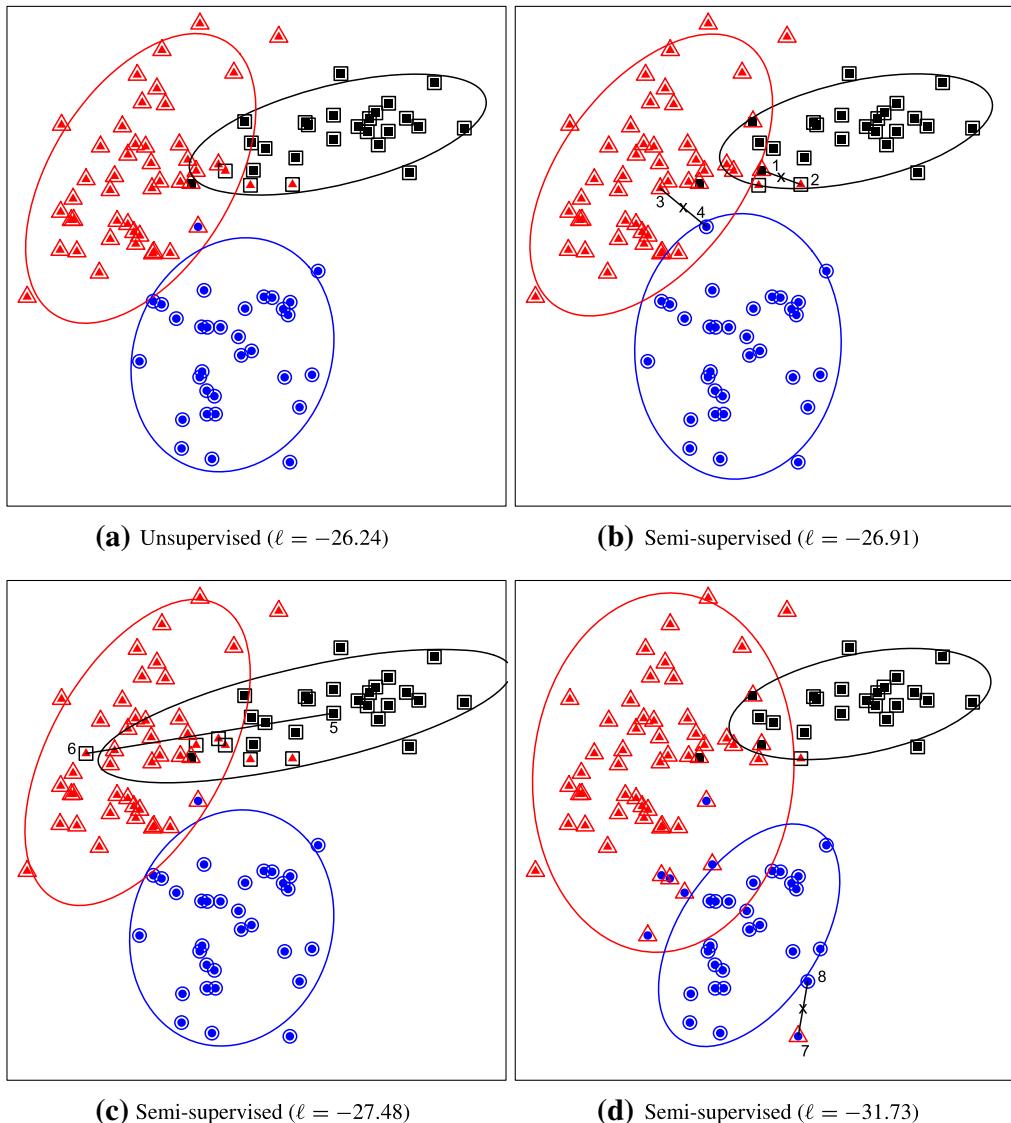


Fig. 4 Performance of **a** unsupervised clustering and **b–d** semi-supervised clustering with various constraints representing illustrative examples from Sect. 3.1

by means of the package MIXSIM. Mixing proportions were specified to be equal. From every generated mixture, one dataset of size 300 was simulated. Since obtained data are synthetic, the true partitioning of each dataset is available. In cases with positive constraints, we formed a block of size $|\mathcal{B}_b|$ in each true group. Thus, in the positive relation section of Table 2, column $|\mathcal{B}_b| = 1$ represents the case of unsupervised clustering and $|\mathcal{B}_b| = 25$ assumes that a block involving 25 observations was created in each true cluster. Four cases with $|\mathcal{B}_b| = 1, 5, 15, 25$ were considered. In cases with negative constraints, we use the same setting but provide three additional negative restrictions: the first and second blocks cannot belong to the same cluster, the third and fourth blocks cannot belong to the same cluster, and the fifth and sixth blocks cannot belong to the same group. Table 2 contains the results of the simulation study.

Table 2 Results of the simulation study from Sect. 3.2 devoted to studying computational costs of semi-supervised clustering

	Positive relations				Negative relations			
	$ \mathcal{B}_b = 1$	$ \mathcal{B}_b = 5$	$ \mathcal{B}_b = 15$	$ \mathcal{B}_b = 25$	$ \mathcal{B}_b = 1$	$ \mathcal{B}_b = 5$	$ \mathcal{B}_b = 15$	$ \mathcal{B}_b = 25$
$\bar{T}_{\bullet}^{EM}/\bar{T}_{unsup}^{EM}$	1.000	0.661	0.290	0.113	0.875	0.418	0.280	0.088
\bar{i}^{EM}	14.45	10.26	5.58	2.83	12.14	6.21	5.04	1.96

$|\mathcal{B}_b|$ represents the size of a block in each simulated cluster, $\bar{T}_{\bullet}^{EM}/\bar{T}_{unsup}^{EM}$ denotes the ratio of average time required by the EM algorithm in cases of semi-supervised and unsupervised clustering, and \bar{i}^{EM} stands for the average number of iterations of the EM algorithm

To assess the computational cost of the methods in different settings, we recorded the average number of EM iterations \bar{i}^{EM} required to reach convergence (initialization stage is omitted) and the ratio $\bar{T}_{\bullet}^{EM}/\bar{T}_{unsup}^{EM}$, where \bar{T}_{unsup}^{EM} represents the average time required by the EM algorithm in the unsupervised clustering case and \bar{T}_{\bullet}^{EM} represents the same quantity obtained in the semi-supervised clustering setting. Thus, this ratio provides the time improvement achieved by semi-supervised clustering compared with the unsupervised case.

As we can see from Table 2, all settings with semi-supervised clustering allow improving the computational time. The explanation of this phenomenon lies, first of all, in the considerable reduction of the number of iterations required by the EM algorithm. The other reason for the observed speed improvement is related to the lower number of posterior probability calculations at the E-step. Another interesting observation can be made from the comparison of settings with negative and positive constraints. As we can see, additional negative constraints allowed further time improvement. This was achieved by reducing the space of possible solutions due to the additional information that prevents merging blocks that belong to different clusters.

Indeed, the considered simulation study is by no means a comprehensive one. Various sample sizes, component overlap levels, and the number of components can be investigated. Also, negative constraints of more complicated forms can lead to longer computational times. Another interesting aspect that might be considered in the future is the performance of semi-supervised clustering procedures under the model misspecification. Despite the fact that the carried out simulation study is short and preliminary, we can conclude that semi-supervised clustering with constraints is highly practical and can lead to considerable time improvements in the performance of the EM algorithm.

3.3 Classification datasets

In this section, we consider the application of the proposed methodology to two popular classification datasets: *Iris* (Anderson 1935; Fisher 1936) and *Crabs* (Campbell and Mahon 1974).

Table 3 Unsupervised and semi-supervised clustering solutions obtained for different K values for the dataset *Iris*

	<i>I. Setosa</i>	<i>I. Versicolor</i>	<i>I. Virginica</i>	BIC	\mathcal{AR}
Unsupervised					
$K = 1$	50	50	50	829.98	0.000
$K = 2$	50	0	0	574.02	0.568
	0	50	50		
$K = 3$	50	0	0	580.84	0.904
	0	45	0		
	0	5	50		
Semi-supervised					
$K = 2$	1	1	2	764.47	0.568
	1	2	1	862.43	0.568
	1	2	2	574.02	0.568
$K = 3$	1	2	3	586.31	1.000

Bold font in the BIC column highlights the smallest BIC value under unsupervised or semi-supervised setting

3.3.1 Iris dataset

The dataset *Iris* contains 150 data points representing three different iris species: *I. Setosa*, *I. Versicolor*, and *I. Virginica*. There are four measurements taken on each subject: sepal length, sepal width, petal length, and petal width. Multiple studies involving this dataset suggest that *I. Setosa* represents an extremely well-separated cluster while *I. Versicolor* and *I. Virginica* overlap considerably. Melnykov (2013) reports that the sum of two misclassification probabilities between *I. Versicolor* and *I. Virginica* under Gaussian mixture model is equal to 0.049. While this value is not extreme, it is high enough to cause considerable challenges in making a decision regarding the preferable number of components and clusters. Thus, it is not clear whether this gives sufficient grounds to justify the three-component solution over the simpler one with just two components. Table 3 provides the results of unsupervised and semi-supervised model-based clustering with Gaussian components. For unsupervised clustering, classification agreement tables are provided. As expected, in the case with $K = 2$, groups *I. Versicolor* and *I. Virginica* are merged together. For $K = 3$, these two species are successfully separated with just five observations from *I. Versicolor* incorrectly classified to *I. Virginica*. Unrestricted mixture modeling relying on BIC recommends the two-component solution with $\mathcal{AR} = 0.568$. The difference in BIC values between two- and three-component models is rather small: 6.82. In the meantime, this difference is obtained assuming that the membership of observations is unknown, which is not the case here.

Would the results be much different if the available information about the membership of data points were incorporated into the model? Table 3 provides all possible solutions for mixture modeling with constraints and suggests that the best solution with constraints is obtained for $K = 2$ ($\{1, 2, 2\}$). Specifically, *I. Versicolor* and *I.*

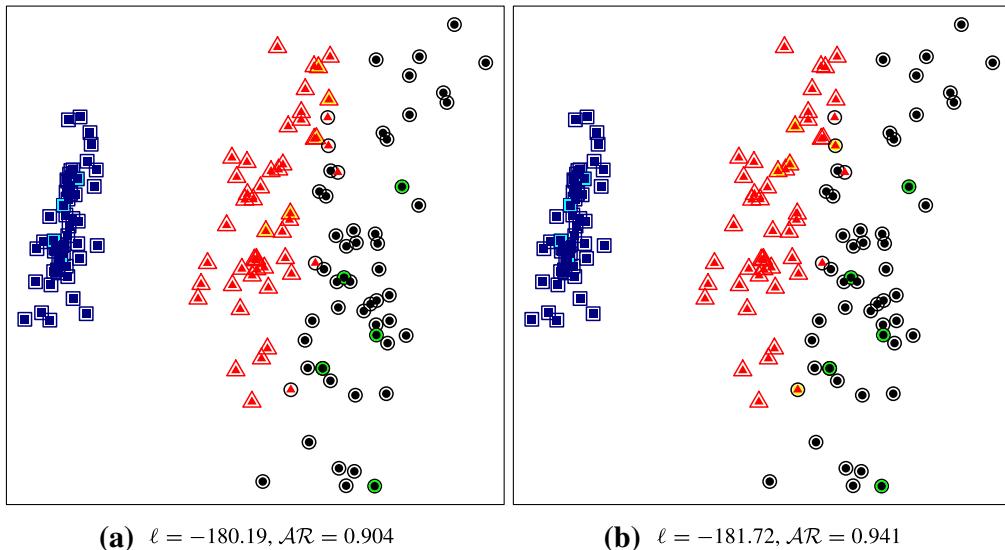


Fig. 5 Plots of principal components for the solutions obtained for the *Iris* dataset based on **a** unsupervised clustering and **b** semi-supervised clustering with positive and negative constraints. *I. Setosa*, *I. Versicolor*, and *I. Virginica* are shown as *navy squares*, *red triangles*, and *black circles*, respectively. Blocks with available membership information are provided in *cyan*, *yellow*, and *green* colors (color figure online)

Virginica are combined in one block and *I. Setosa* represents a different block. This arrangement produces the same BIC value of 574.02 as found by the unsupervised method with $K = 2$. However, for $K = 3$, mixture modeling with constraints yields the BIC value of 586.31. Thus, the actual difference in BIC values is not 6.82 but rather 12.29. This almost doubles the original difference in BIC and provides more substantial grounds for favoring the two-cluster solution.

Figure 5 plots principal components to illustrate the solutions obtained by unsupervised model-based clustering [plot (a)] and semi-supervised clustering with different sets of positive and negative constraints [plots (a) and (b)]. The description of the plots is similar to that given for Fig. 4. Blocks of observations with positive constraints consist of 10 % of data points (5 for each cluster) and are shown in cyan, yellow, and green colors. The negative constraint is also set for these blocks, thus not allowing them to belong to the same components. As we can see from Fig. 5, different choices of constraints can lead to the same [plot (a)] or different [plot (b)] solutions as in the unsupervised clustering case.

3.3.2 Crabs dataset

Similar analysis is provided now for another famous dataset called *Crabs*. 200 5-variate measurements on orange and blue crabs are considered. The species are equally represented and, in turn, have equal numbers of males and females. Thus, four classes under consideration are *B. Male*, *B. Female*, *O. Male*, and *O. Female*, each consisting of 50 data points. The five measurements taken on each subject are the frontal lobe size, rear width, carapace length, carapace width, and body depth. The original goal of the study involving this dataset was to check whether blue and orange crabs form well-separated groups. Table 4 represents the results obtained by unsupervised and

Table 4 Unsupervised and semi-supervised clustering solutions obtained for different K values for the dataset *Crabs*

	<i>B. Male</i>	<i>B. Female</i>	<i>O. Male</i>	<i>O. Female</i>	BIC	\mathcal{AR}
Unsupervised						
$K = 1$	50	50	50	50	3069.72	0.000
$K = 2$	50	50	0	0	2925.54	0.496
	0	0	50	50		
$K = 3$	50	50	0	0	2891.06	0.678
	0	0	50	3		
	0	0	0	47		
$K = 4$	39	0	0	0	2887.15	0.818
	11	49	0	0		
	0	0	50	3		
	0	1	0	47		
Semi-supervised						
$K = 2$	1	1	1	2	3036.07	0.330
	1	1	2	1	3010.18	0.330
	1	2	1	1	3044.60	0.330
	1	1	2	2	2925.55	0.496
	1	2	1	2	2951.72	0.496
	1	2	2	1	3090.93	0.496
	1	2	2	2	3029.79	0.330
$K = 3$	1	1	2	3	2894.36	0.711
	1	2	1	3	2931.33	0.711
	1	2	3	1	2974.97	0.711
	1	2	3	2	2913.37	0.711
	1	2	3	3	2930.12	0.711
	1	2	2	3	2990.57	0.711
$K = 4$	1	2	3	4	2898.09	1.000

Bold font in the BIC column highlights the smallest BIC value under unsupervised or semi-supervised setting

semi-supervised model-based clustering. *Crabs* is a complicated dataset that is not easy to analyze. The solution obtained by the popular package MCLUST (Fraley and Raftery 2006) recommends a Gaussian mixture with 9 spherical components. Melnykov (2013) remarks that this solution is difficult to interpret biologically. Hennig (2010) discusses that the choice of Gaussian components itself can be incorrect in this case. Bouveyron and Brunet (2014) explain the unsatisfactory clustering performance by the dimensionality of the dataset and recommend reducing the number of variables along with searching for an optimal partition. While both points of view are viable and probably true to some extent, we believe that the biggest challenge is to initialize the dataset properly. We employed the *emEM* algorithm with 10,000 *short EM* iterations in order to find a four-component mixture of Gaussian components with BIC equal to 2887.15 and $\mathcal{AR} = 0.818$, the best solution.

It is worth mentioning that there are only 15 misclassifications (7.5 %) associated with this solution. This result is the best in terms of BIC and misclassification proportion among the findings reported in literature. As we discussed before, in the unsupervised clustering setting, we completely ignore the fact that classification of all observations is, in fact, available. Therefore, while useful for checking the performance of various clustering algorithms, unsupervised clustering applied to classification datasets ignores known labels and can be misleading in reporting the actual number of components and clusters. As we can see from Table 4, the best detected model in the semi-supervised clustering framework has three components with the corresponding BIC value of 2894.36 and $\mathcal{AR} = 0.711$. *B. Male* and *B. Female* crabs are not distinct enough to recommend their separation. This finding is not very surprising as the overlap between these two groups is quite high: 0.087. The 4-component solution has a slightly higher BIC value of 2898.09. This example illustrates that accounting for all information available can lead to a conclusion different from that in the unsupervised clustering framework when no membership information is used.

Figure 6 displays principal components of the dataset under unsupervised clustering [plot (a)] and semi-supervised clustering with positive and negative constraints with block sizes chosen again at the 10 % of a cluster size for each group of crabs [plot (b)]. As the primary goal of the analysis for this dataset is to decide whether orange and blue crabs form separate groups, the negative constraints were placed only for gender groups within each species, i.e., in plot (b), it is assumed that blocks representing *B. Male* and *B. Female* cannot be merged as well as those representing *O. Male* and *O. Female*. The description of the figure is similar to the one of Fig. 5. Navy squares and steel blue circles represent *B. Male* and *B. Female*, while red triangles and orange rhombi illustrate *O. Male* and *O. Female* groups, respectively. As we can see from Fig. 6, the provision of restrictions for the highlighted points does not change the solution substantially: there is only one navy square (besides those included in the cyan block) that switched its classification to the correct one.

4 Application

In this section, we conduct semi-supervised clustering of data using mixtures of regression models. This type of data can arise when information on covariate variables is available in addition to the response. The inclusion of covariates can help account for correlation among observations being clustered. Some applications of mixtures based on regression models include clustering of trajectories (Gaffney and Smyth 1999) and clustering of time series data (Chen and Maitra 2011). The latter arises, for example, in the field of dendrochronology.

Dendrochronology is the area of science that analyzes the sequences of tree rings. The width of the rings indicates the growth of a tree over a period of time. If the period is equal to one year, tree rings are commonly called annual. Their growth is dictated by environmental conditions and biological agents. Tree rings have been used to reconstruct past climate changes and environmental circumstances (Hughes et al. 2009). Other applications of dendrochronology can be found in archaeology where the analysis of tree ring sequences helps track ancient wooden artifacts. For a comprehensive review of these applications, we refer the reader to the work of

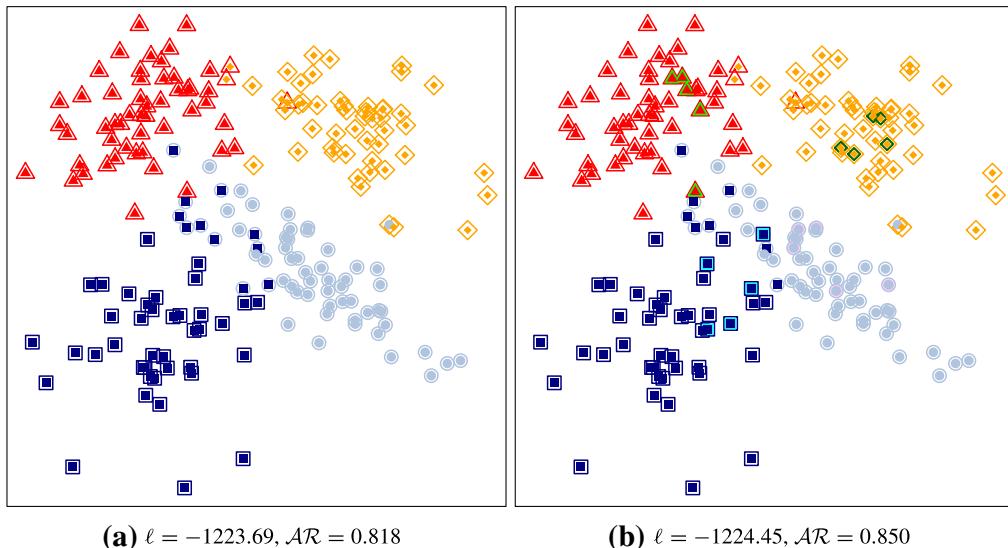


Fig. 6 Plots of principal components for the solutions obtained for the *Crabs* dataset based on **a** unsupervised clustering and **b** semi-supervised clustering with positive and negative constraints. *B. Male*, *B. Female*, *O. Male*, and *O. Female* are shown as navy squares, light blue circles, red triangles, and orange rhombi, respectively. Blocks with available membership information are provided in cyan, pink, light and dark green colors (color figure online)

Bridge (2012). Previous attempts of clustering tree ring sequences using traditional cluster analysis tools have been unsuccessful producing results with no geographical meaning (Haneca et al. 2005; Bridge 2012). In this paper, we tackle the problem using the developed semi-supervised clustering methodology.

The analyzed dataset is extracted from the International Tree-Ring Data Bank (ITRDB) (Grissino-Mayeri and Fritts 1997). The measurements are taken from three states in the western part of the US (California, Nevada, and Oregon) at nine different locations. The three similar species of pines considered are called *Pinus Monticola*, *Pinus Jeffreyii*, and *Pinus Ponderosa*. There are 160 trees that represent these species and have records in the time interval between 1700 and 1980. Hence, the dataset includes 160 observations in 281 dimensions. Generally, it can be observed that the width of annual rings decreases as the tree gets older. As a result, log-transformed measurements are considered in our analysis. The information about the exact location, species, site elevation, and the number of trees is summarized in Table 5. Our goal is to group trees according to the similarity in annual ring behavior to study climate zones and patterns. Intuitively, we can expect that trees from the same location face nearly identical environmental conditions and thus can be modeled by a common mixture component. Therefore, semi-supervised clustering with positive constraints, requiring all trees within each site to be grouped together, can be readily applied. The maximum number of mixture model components is 9 as there are 9 distinct sites.

Melnykov (2012) considered unsupervised model-based clustering of Gaussian regression autoregressive (AR) time series based on the finite mixture model

$$f(\mathbf{y}_i | \Psi) = \sum_{k=1}^K \tau_k \phi_{Y_i}(\mathbf{y}_i | \mathbf{X}_i \boldsymbol{\beta}_k, \boldsymbol{\Sigma}_k),$$

Table 5 Species, site names, site coordinates and elevation, and the number of trees at each of the nine sites from which tree ring data are included in the dataset analyzed in Sect. 4

	Site	Species	# of trees	Elevation	Coordinates
1	Little Aspen Butte, OR	<i>Pinus Ponderosa</i>	18	1650	(42.27, -122.08)
2	Likely Mountain, CA	<i>Pinus Jeffreyii</i>	2	1811	(41.15, -120.57)
3	Lemon Canyon, CA	<i>Pinus Jeffreyii</i>	36	1859	(39.57, -120.25)
4	Bryant Creek, CA	<i>Pinus Monticola</i>	9	2073	(38.75, -119.67)
5	Kennedy Meadows, CA	<i>Pinus Jeffreyii</i>	36	2024	(36.03, -118.18)
6	Wall Canyon, NV	<i>Pinus Monticola</i>	33	2316	(38.68, -117.23)
7	Moody Mountain, NV	<i>Pinus Monticola</i>	17	2004	(39.10, -115.80)
8	Pony Express, NV	<i>Pinus Monticola</i>	4	2210	(39.82, -114.62)
9	Panaca Summit, NV	<i>Pinus Monticola</i>	5	2103	(37.77, -114.18)

where \mathbf{y}_i and \mathbf{X}_i represent the i th time series of length T and corresponding matrix of predictor variables, respectively, $\boldsymbol{\beta}_k$ is the vector of predictor variable coefficients, and $\phi_{\mathbf{Y}_i}(\cdot | \mathbf{X}_i \boldsymbol{\beta}_k, \Sigma_k)$ is the density function of the T -variate Gaussian distribution with mean vector $\mathbf{X}_i \boldsymbol{\beta}_k$ and covariance matrix Σ_k . In this setting, the estimation of parameters can be approached by the traditional EM algorithm (Chen and Maitra 2011). However, the closed form expression for the estimate of the $T \times T$ matrix Σ_k is not readily available and the numerical approach to the estimation of covariance matrices can fail if the length of time series is large. To account for this in mixture modeling setting, Melnykov (2012) proposed employing the conditional maximum likelihood estimates (cMLE) obtained by conditioning on the first few observations in time series. Using this approach, the problem with the dimensionality of covariance matrices can be overcome. Moreover, cMLE and MLE share the same asymptotic distribution. For more details, we refer the reader to the original paper.

Here, we modify the EM algorithm provided in Melnykov (2012) according to the developed methodology and perform semi-supervised clustering of regression time series. Annual tree ring sequences represent the response variable while tree age serves as the predictor variable. We run the EM algorithm described in Sect. 2.4 for $K = 1, 2, \dots, 9$ and various AR orders. Using BIC as a model selection criterion, we find the best solution at $K = 4$. While BIC tended to prefer models with higher AR orders, clustering solutions for such models remained unchanged. The best solutions detected for $K = 2, 3$, and 4 are provided in Fig. 7.

Each circle represents a particular site location while the area of the circle is proportional to the number of trees at that site. Colors mark estimated clusters. An interesting observation can be made that all the solutions for $K = 2, 3, 4$ are nested and reflect the geographical location of sites very well. The clusters formed are reasonable as they consistently emerge along the north-south axes. More specifically, the best detected solution at $K = 4$ [plot (c)] follows the north-south gradient which makes good environmental sense. Taking into account already reported challenges with clustering annual tree ring sequences, we can conclude that the proposed technique shows good and promising performance for these data. If some additional environmental infor-

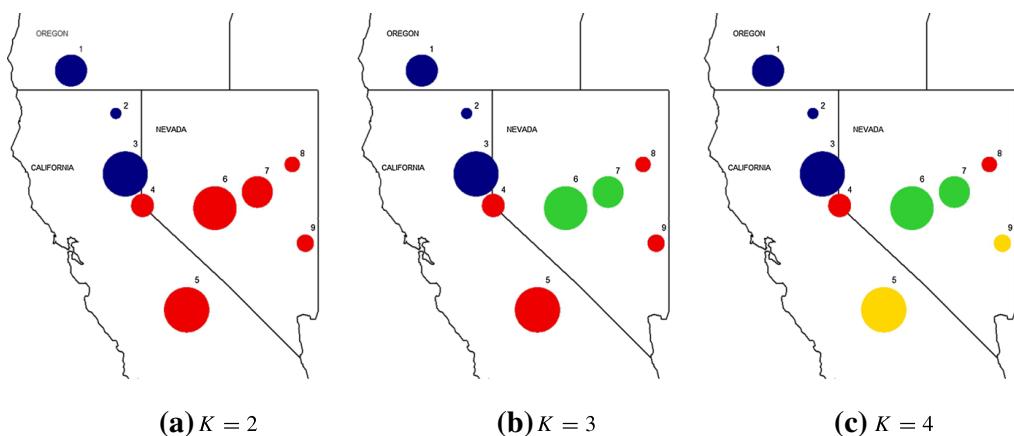


Fig. 7 Semi-supervised clustering solutions. The area of each *circle* is proportional to the number of trees at a particular site. *Colors* represent estimated clusters (color figure online)

mation that trees at specific sites should belong to different clusters is available, the incorporation of negative constraints is quite straightforward for this application.

5 Discussion

We proposed an attractive and intuitive way to conduct semi-supervised clustering with positive and negative constraints and considered the effect that such constraints have on the solutions obtained in the framework of model-based clustering. While previous research in this field produced many useful results, a unified approach for incorporating both kinds of constraints in the EM algorithm has not been previously developed. Of the two kinds of constraints, positive restrictions are more tractable and yield the modifications of the EM algorithm that can accommodate any number of blocks while not leading to a substantial increase in the amount of computations involved. Negative relations among blocks can result in considerably more complicated structures within the dataset, making the computation rather time-consuming. We described all situations that can arise while dealing with negative constraints in the two- and three-block structures and also showed how our approach to obtaining posterior probabilities π_{bk}^{\pm} can be extended to a larger number of blocks.

Semi-supervised clustering can be used effectively in the circumstances where the classification vector is partially known or where two or more competing classifications have to be evaluated to decide in favor of the most appropriate one. We considered two well-known classification datasets, *Iris* and *Crabs*, and compared proposed classifications in terms of BIC. Thus, in the *Iris* dataset, the solution with two clusters appeared more accurate than the three-cluster one after taking into account the known classification of data points. This result is more pronounced than the outcome obtained by means of unsupervised clustering, where the difference in BIC values is almost twice smaller.

A simulation study was conducted to compare the outcomes of the *emEM* initialization algorithm in the unsupervised and semi-supervised settings. Semi-supervised

clustering performed better than its unsupervised counterpart in the situations with moderate and high overlap, but slightly worse with the low overlap. This emphasizes the need for a careful initialization in the semi-supervised setting, as block hulls may overlap with each other, which can cause one of the components to degenerate in the absence of points from other clusters nearby. The suggested approach was also applied to conduct an analysis of dendrochronological data from the western region of the United States. The results obtained by our method are readily interpreted from the geographical viewpoint.

Overall, the proposed method of incorporating both positive and negative constraints in the EM algorithm appears to work well, producing intuitive results consistent with the nature of the data being analyzed.

References

- Anderson E (1935) The Irises of the Gaspe Peninsula. *Bull Am Iris Soc* 59:2–5
- Basu S, Banerjee A, Mooney R (2002) Semi-supervised clustering by seeding. In: Proceedings of the 19th International Conference on Machine Learning, pp 19–26
- Basu S, Bilenko M, Mooney RJ (2004) A Probabilistic Framework for Semi-Supervised Clustering. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 59–68
- Basu S, Davidson I, Wagstaff K (2008) Constrained clustering: advances in algorithms, theory, and application. Chapman and Hall/CRC
- Bouveyron C, Brunet C (2014) Model-based clustering of high-dimensional data: a review. *Comput Stat Data Anal* 71:52–78
- Bridge M (2012) Locating the origins of wood resources: a review of dendroprovenancing. *J Archaeol Sci* 39:2828–2834
- Campbell NA, Mahon RJ (1974) A multivariate study of variation in two species of rock crab of genus *Leptograsus*. *Aust J Zool* 22:417–425
- Chen W-C, Maitra R (2011) Model-based clustering of regression time series data via APECM-An AECM Algorithm Sung to an even faster beat. *Stat Anal Data Min* 4:567–578
- Côme E, Oukhellou L, Deneux T, Aknin P (2009) Learning from partially supervised data using mixture models and belief functions. *Pattern Recognit* 42:334–348
- Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood for incomplete data via the EM algorithm (with discussion). *J Royal Stat Soc, Ser B* 39:1–38
- Digalakis VV, Ritschev D, Neumeyer LG (1995) Speaker adaptation using constrained estimation of Gaussian mixtures. *IEEE Trans Speech Audio Process* 3:357–366
- Fisher RA (1936) The use of multiple measurements in taxonomic problems. *Ann Eugen* 7:179–188
- Forgy E (1965) Cluster analysis of multivariate data: efficiency vs. interpretability of classifications. *Biometrics* 21:768–780
- Fraley C, Raftery AE (1998) How many clusters? Which cluster method? Answers via model-based cluster analysis. *Comput J* 41:578–588
- Fraley C, Raftery AE (2002) Model-based clustering and density estimation. *J Am Stat Assoc* 97:611–631
- Fraley C, Raftery AE (2006) MCLUST Version 3 for R: normal mixture modeling and model-based clustering, Tech. Rep. 504, University of Washington, Department of Statistics, Seattle, WA
- Gaffney SJ, Smyth P (1999) Trajectory clustering with mixture of regression model. Proceedings of Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San diego. CA. ACM, USA, pp 63–72
- Grissino-Mayeri HD, Fritts H (1997) The international tree-ring data bank: an enhanced global database serving the Global Scientific Community. *Holocene* 7:235–238
- Haneca K, Wazny T, Van Acker J, Beeckman H (2005) Provenancing Baltic timber from art historical objects: success and limitations. *J Archaeol Sci* 32:261–271
- Hennig C (2010) Methods for merging Gaussian mixture components. *Adv Data Anal Classif* 4:3–34

- Huang J-T, Hasegawa-Johnson M (2009) On semi-supervised learning of Gaussian mixture models for phonetic classification. In: NAACL HLT workshop on semi-supervised learning
- Hughes MK, Swetnam TW, Diaz HF (2009) Dendroclimatology: progress and prospects, vol 11. Princeton, Developments in Paleoenvironmental Research Springer
- Johnson S (1967) Hierarchical clustering schemes. *Psychometrika* 32(3):241–254
- Law MHC, Topchy A, Jain AK (2005) Model-based clustering with probabilistic constraints. In: 2005 SIAM International Conference on Data Mining, pp 641–645
- Liu B, Shen X, Pan W (2013) Semi-supervised spectral clustering with application to detect population stratification. *Front Genet* 4:1–5
- Lu Z, Leen TK (2007) Penalized probabilistic clustering. *Neural Comput* 19:1528–1567
- MacQueen J (1967) Some methods for classification and analysis of multivariate observations. Proc Fifth Berkeley Symp 1:281–297
- Maitra R, Melnykov V (2010) Simulating data to study performance of finite mixture modeling and clustering algorithms. *J Comput Graph Stat* 19:354–376
- Martinez-Uso A, Pla F, Sotoca J (2010) A semi-supervised Gaussian mixture model for image segmentation. In: International Conference on Pattern Recognition, pp 2941–2944
- McLachlan G, Peel D (2000) Finite Mixture Models. Wiley, New York
- Melnykov V (2012) Efficient estimation in model-based clustering of Gaussian regression time series. *Stat Anal Data Min* 5:95–99
- Melnykov V (2013) On the distribution of posterior probabilities in finite mixture models with application in clustering. *J Multivar Anal* 122:175–189
- Melnykov V, Chen W-C, Maitra R (2012) MixSim: R package for simulating datasets with pre-specified clustering complexity. *J Stat Softw* 51:1–25
- Melnykov V, Maitra R (2010) Finite mixture models and model-based clustering. *Stat Surv* 4:80–116
- Nigam K, McCallum AK, Thrun S, Mitchell T (2000) Text classification from labeled and unlabeled documents using EM. *Mach Learn* 39:103–134
- Pan W, Shen X, Jiang A, Hebbel R (2006) Semisupervised learning via penalized mixture model with application to microarray sample classification. *Bioinformatics* 22(19):2388–2395
- Schwarz G (1978) Estimating the dimensions of a model. *Ann Stat* 6:461–464
- Shental N, Bar-Hillel A, Hertz T, Weinshall D (2003) Computing Gaussian mixture models with EM using equivalence constraints. In: Advances in NIPS, vol. 15
- Sloane NJA (2014) The online encyclopedia of integer sequences: A001349 Number of connected graphs with n nodes
- Wagstaff K, Cardie C, Rogers S, Schroedl S (2001) Constrained K -means Clustering with Background Knowledge. In: Proceedings of the Eighteenth International Conference on Machine Learning, pp 577–584
- Wang L, Zhu J, Zou H (2007) Hybrid Huberized Support Vector Machines for Microarray Classification. Proceedings of the 24th International Conference on Machine Learning, New York. NY. ACM, USA, pp 983–990
- Ward JH (1963) Hierarchical grouping to optimize an objective function. *J Am Stat Assoc* 58:236–244

A consensus-constrained parsimonious Gaussian mixture model for clustering hyperspectral images

Ganesh Babu^{*1}, Aoife Gowen^{†2}, Michael Fop^{‡1}, and Isobel Claire Gormley^{§1}

¹School of Mathematics and Statistics,
²School of Biosystems and Food Engineering
 University College Dublin, Ireland.

Abstract

The use of hyperspectral imaging to investigate food samples has grown due to the improved performance and lower cost of spectroscopy instrumentation. Food engineers use hyperspectral images to classify the type and quality of a food sample, typically using classification methods. In order to train these methods, every pixel in each training image needs to be labelled. Typically, computationally cheap threshold-based approaches are used to label the pixels, and classification methods are trained based on those labels. However, threshold-based approaches are subjective and cannot be generalized across hyperspectral images taken in different conditions and of different foods. Here a consensus-constrained parsimonious Gaussian mixture model (ccPGMM) is proposed to label pixels in hyperspectral images using a model-based clustering approach. The ccPGMM utilizes available information on the labels of a small number of pixels and the relationship between those pixels and neighbouring pixels as constraints when clustering the rest of the pixels in the image. A latent variable model is used to represent the high-dimensional data in terms of a small number of underlying latent factors. To ensure computational feasibility, a consensus clustering approach is employed, where the data are divided into multiple randomly selected subsets of variables and constrained clustering is applied to each data subset; the clustering results are then consolidated across all data subsets to provide a consensus clustering solution. The ccPGMM approach is applied to simulated datasets and real hyperspectral images of three types of puffed cereal, corn, rice, and wheat. Improved clustering performance and computational efficiency are demonstrated when compared to other current state-of-the-art approaches.

1 Introduction

Hyperspectral imaging is a spectroscopic method combining digital imaging with spectroscopy (Gowen et al., 2019). Hyperspectral imaging collects an image as a function of the light, in which each pixel is a vector of reflectance or fluorescence information of the sample under study in the continuous spectrum. The idea behind producing hyperspectral images lies in the interaction between the photons emitted by a light source and the physical and chemical properties of the sample (Amigo and Santos, 2020). The interaction allows hyperspectral images to capture critical information and traits of a sample not visible to the naked eye. The critical information captured on the continuous spectrum can be affected by spatial interference and spectral and redundant noise (Amigo, 2020). These issues are effectively handled with computationally intensive spectral preprocessing techniques (Amigo, 2010) like denoising and scatter correction (Rinnan et al., 2009; Bocklitz et al., 2011; Geladi et al., 1985). The preprocessed information is then used to detect chemical contamination, adulteration, and presence of foreign components in the food samples under study (Feng and Sun, 2012). Thanks to the increasing availability of statistical methods for handling complex data, food engineers are routinely employing classification

^{*}ganesh.babu@ucdconnect.ie

[†]aoife.gowen@ucd.ie

[‡]michael.fop@ucd.ie

[§]claire.gormley@ucd.ie

Isobel Claire Gormley and Michael Fop contributed equally to this work.

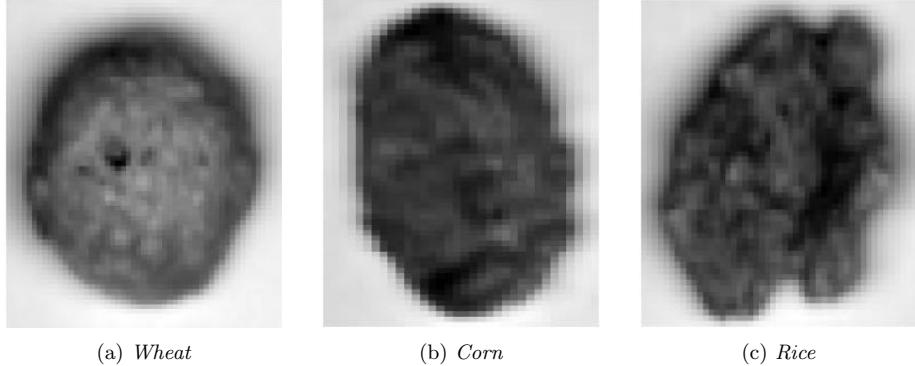
methods for segmentation, food sample detection, and analysis of large volumes of hyperspectral food images.

To train a classification method, each pixel of a hyperspectral image needs to be appropriately labelled as e.g., either the background or the food sample. Usually, the images are captured on appropriate backgrounds that facilitate segmentation. In such cases, to select the regions of interest or the food sample under study, a process called masking is performed to remove the background pixels from the image (Amigo et al., 2015). To create the mask, multiple techniques are employed (Pal and Pal, 1993), with the most widely used being k -means clustering (Amigo et al., 2015) and threshold-based approaches (Pal and Pal, 1993; Gowen et al., 2019; Xu et al., 2020). In Amigo et al. (2015), k -means clustering with 2 clusters is applied to a set of hyperspectral images to separate the background from the food sample. Abdi and Williams (2010), Gowen et al. (2019), and Xu et al. (2020) highlight some of the prominent threshold-based approaches used to label the different food samples and background pixels in a given hyperspectral image. In threshold-based approaches, the threshold for the segmentation of the pixels is selected by producing a histogram of some useful pixel-related summary of the entire spectrum, like the mean or the principal component scores of the wavelength intensities for each pixel. The selection of which summary to employ in the histogram depends on the type of background and the food sample under study. In the histogram, typically a bimodal distribution is observed and a threshold is subjectively chosen as the point that separates the two modes. The quality of the threshold inferred labels in each hyperspectral image depends on several decisions like the summary metric used and which variables are selected for computing the summary. Furthermore, with the threshold approach, only hard classification of pixels as background or food type are possible, ignoring the inherent uncertainty in the labelling process, particularly in certain areas of the image, like the pixels around the edges of the food sample (Amigo and Santos, 2020).

A major challenge in labelling the pixels of the hyperspectral images is that each image has a large number of pixels and for each pixel, a large amount of reflectance information is captured in a continuous p -dimensional spectrum, whose wavelengths are highly correlated. Further, often threshold and k -means clustering approaches are applied to each hyperspectral image individually, which is impractical as the number of images increases and does not allow for the borrowing of information across images. Also, the threshold approach and k -means clustering with 2 clusters work well only when there is exactly one type of food sample in a hyperspectral image. As the number of food samples in a given image increases, multiple thresholds need to be selected from the histogram, which is even more challenging. Overall, threshold based approaches are subjective and are not easily generalisable across hyperspectral images.

Approaches to clustering pixels in hyperspectral images are receiving increasing attention (Zhai et al., 2021). The density based DBSCAN clustering approach (Khan et al., 2014; Hahsler et al., 2019) has been widely used in imaging. Further, model-based clustering approaches have proved effective, for example via Gaussian mixture models (GMM, Scrucca et al., 2016; McNicholas, 2016; Bouveyron et al., 2019) and parsimonious versions (McNicholas and Murphy, 2008). Bouveyron and Brunet-Sauvage (2014) propose a dimension reduction-based approach to segmenting hyperspectral images of Mars. Further, Jacques and Ruckebusch (2016) consider co-clustering of pixels and wavelengths in the context of hyperspectral imaging of an oil-in-water emulsion using a latent block model.

Here, we propose a novel model-based clustering approach, a consensus-constrained parsimonious Gaussian mixture model (ccPGMM), to label the pixels of multiple hyperspectral images simultaneously in a computationally feasible manner while accounting for uncertainty. A key aspect of ccPGMM is that it involves a constrained parsimonious Gaussian mixture model (Melnikov et al., 2016) in which information regarding the types of food samples present in the hyperspectral images along with information about some pixels is incorporated as constraints to perform informed clustering. Further, as hyperspectral images have a large number of pixels, each of which has an associated high-dimensional spectrum of p strongly correlated variables, ccPGMM employs a parsimonious Gaussian mixture model (PGMM) which describes the high-dimensional data using a small number of latent variables. To enable computationally efficient implementation, ccPGMM adopts a consensus approach, whereby the high-dimensional spectrum is divided into randomly selected subsets of $d < p$ variables, and a constrained PGMM is fitted to each subset of variables on all pixels. The clustering solutions of the constrained PGMMs fitted to each subset are consolidated to provide a final consensus clustering allocation for each pixel. The performance of the proposed ccPGMM approach is demonstrated, and compared to other state-of-the-art approaches, through thorough simulation studies and application



(a) Wheat

(b) Corn

(c) Rice

Figure 1: Greyscale images of one hyperspectral image of each puffed cereal type. For each pixel, the intensity of the color corresponds to the average of its NIR spectrum.

to real hyperspectral images of puffed cereals (Gowen et al., 2019).

In what follows, Section 2 gives details of the motivating dataset containing hyperspectral images of puffed cereals. Section 3 outlines the ccPGMM approach and discusses its inference and implementation details. Section 4 delineates thorough simulation studies that explore the performance of ccPGMM under different settings. Section 5 discusses the application of ccPGMM to the motivating puffed cereal images and Section 6 summarises the contributions of ccPGMM and gives an overview of future research directions. The R (R Core Team, 2023) code to implement ccPGMM is available at this repository [GitHub](#).

2 Hyperspectral images of puffed cereals

The ccPGMM approach is motivated by the need to simultaneously label pixels in nine hyperspectral images of three types of puffed cereal, wheat, corn, and rice (Gowen et al., 2019). Each pixel in an image records the near-infrared (NIR) spectrum, with $p = 101$ equally spaced variables spanning the wavelengths in the 880-1720nm interval. Each image $l = 1, \dots, L = 9$ contains only one puffed cereal and is a three-dimensional tensor of dimension $S_l \times T_l \times p$ where S_l and T_l are the numbers of row and column pixels respectively, which vary across the images as they are of different dimensions (details are given in Appendix 8). Figure 1 shows a greyscale image of one hyperspectral image of each puffed cereal type, where, for each pixel, the intensity of the color is given by the average of the NIR spectrum.

For the purpose of ccPGMM, the pixels are assumed to be independent and so the three-dimensional tensor of each image is represented by a rectangular dataset of dimension $(S_l T_l) \times p$. These $L = 9$ rectangular datasets are then collated to form a single dataset of dimension $N \times p$, where $N = \sum_{l=1}^L S_l T_l$. Here, the final dataset of nine images results in $N = 28039$ pixels, each measured over $p = 101$ wavelengths which are highly correlated and contain spectral noise.

The objective is to simultaneously label all pixels in all nine images as either background or a cereal type, taking into account available information on the images (i.e., it is known which cereal type is in each image) and on some pixels (e.g., corner pixels are background), in a computationally feasible manner.

3 The ccPGMM and its inference

3.1 Parsimonious Gaussian mixture models (PGMM)

Factor analysis is a latent variable model that represents a large number p of correlated variables using a smaller number $q \ll p$ of underlying latent factors. In a single factor analysis model (McLachlan and Peel, 2000), each observation $\mathbf{x}_n \in \mathbb{R}^p$ for $n = 1, \dots, N$ in data $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ of dimension $N \times p$ is modelled as $\mathbf{x}_n = \boldsymbol{\mu} + \boldsymbol{\Lambda}\mathbf{u}_n + \boldsymbol{\varepsilon}_n$ where $\boldsymbol{\mu}$ is the mean, $\boldsymbol{\Lambda}$ is the $p \times q$ loadings matrix and $\mathbf{u}_n \sim \mathcal{N}_q(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{\varepsilon}_n \sim \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Psi})$ are the latent factor score and specific factor, respectively, for

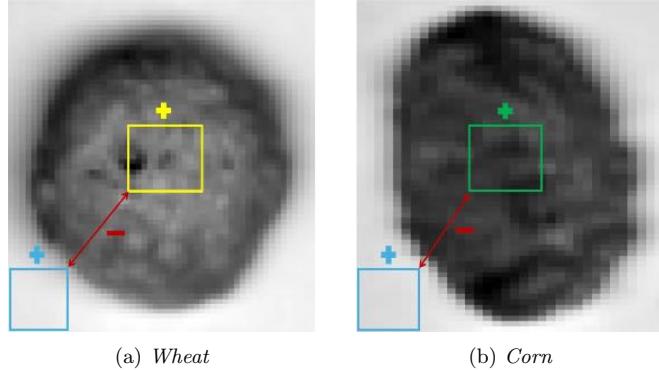


Figure 2: An illustration of positive (+) and negative (−) constraints. Four blocks of three types of pixels are highlighted. Pixels in the blue blocks (assumed to be the background) must be clustered together, as must the pixels in the yellow block (wheat) and the pixels in the green block (corn). However, pixels in the blue blocks must not be clustered with those in the yellow block or with those in the green block, and vice versa.

observation n . Therefore, $\mathbf{x}_n \sim \mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Psi}$ provides a parsimonious covariance structure.

The parsimonious Gaussian mixture model (PGMM) (McLachlan and Peel, 2004; McNicholas and Murphy, 2008; McNicholas et al., 2010) is a finite mixture of G factor analysis models such that each mixture component g has a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_g$ and covariance $\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g^\top + \boldsymbol{\Psi}_g$. Denoting the probability of membership of mixture component g as τ_g , under PGMM, we have $f(\mathbf{x}_n) = \sum_{g=1}^G \tau_g \mathcal{N}_p(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\Psi}_g)$. By constraining $\boldsymbol{\Lambda}_g$ and $\boldsymbol{\Psi}_g$ to be the same or different across components, a family of PGMMs have been developed (McNicholas and Murphy, 2008) with inference via an alternating expectation-conditional maximization(AECM, Ghahramani et al., 1996; McLachlan et al., 2003). These PGMMs facilitate model-based clustering of high-dimensional data.

3.2 Constrained parsimonious Gaussian mixture models

When labelling pixels in a set of hyperspectral images, there is often information about the type of food sample(s) present in some of the hyperspectral images, along with indirect information on the labels of some pixels in the images. For example, in Figure 2, four blocks of pixels are highlighted in three different colours in the two images. The pixels in the blue blocks located at the corner of the images are assumed to belong to the background. On the other hand, the pixels in the yellow and green blocks are located in the center of their images, where the corresponding cereal is observed to be situated. Pixels in the blue blocks should be clustered together, as must the pixels in the yellow block and the pixels in the green block; this information imposes what are known as positive constraints. Also, the pixels in the blue blocks should not be placed in the same cluster as pixels in the yellow or green blocks, and vice versa; this information imposes what are known as negative constraints.

Melnykov et al. (2016) proposed a constrained Gaussian mixture model that uses such additional information about some observations as constraints to perform informed model-based clustering. However, the high-dimensionality of the hyperspectral imaging data poses computational challenges for this approach. Here, we build on Melnykov et al. (2016) and allow for constraints to be incorporated when clustering using a PGMM. The resulting constrained parsimonious Gaussian mixture model (constrained-PGMM) facilitates clustering and therefore labelling of the pixels of the hyperspectral images.

Under the constrained-PGMM, the observed data log-likelihood is

$$\ell_o(\mathbf{X}) = \sum_{n=1}^N \log \sum_{g=1}^G \tau_g \mathcal{N}_p(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\Psi}_g). \quad (1)$$

As (1) is difficult to maximize directly, an AECM algorithm is used to fit the constrained-PGMM.

This requires the introduction of a latent component indicator $\mathbf{z}_n = (z_{n1}, \dots, z_{nG})'$, for $n = 1, \dots, N$, where $z_{ng} = 1$ if observation n belongs to cluster g or 0 otherwise. The AECM algorithm works in two cycles and allows for a different definition of the complete data in each cycle. In the first cycle, the component indicators $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$ are assumed to be the missing data. The complete data log-likelihood is then

$$\ell_c(\mathbf{X}, \mathbf{Z}) = \sum_{n=1}^N \sum_{g=1}^G z_{ng} [\log \tau_g + \log \mathcal{N}_p(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\Psi}_g)]. \quad (2)$$

In the E-step of the first cycle, the available constraints are incorporated into PGMM. Let B_1 be the set of J pixels (indexed by j) in the blue blocks shown in Figure 2 which are known to be background and must be clustered together. Let B_2 be the set of O pixels (indexed by o) in the yellow block of Figure 2 which are known to be wheat and should be clustered together. These are positive constraints (denoted $+$). However, the pixels in B_1 should not be clustered together with the pixels in B_2 . This is a negative constraint (denoted $-$). Thus the posterior probability $\hat{z}_{B_1, g}^-$ of the pixels in B_1 belonging to cluster g , given the positive and negative constraints $+$ and $-$ is

$$\hat{z}_{B_1, g}^- = \frac{\prod_{j=1}^J \hat{\tau}_g \mathcal{N}_p(\mathbf{x}_j; \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Lambda}}_g \hat{\boldsymbol{\Lambda}}_g^\top + \hat{\boldsymbol{\Psi}}_g) \sum_{f=1}^G \prod_{\substack{o \in B_2 \\ f \neq g \\ o=1}} \hat{\tau}_f \mathcal{N}_p(\mathbf{x}_o; \hat{\boldsymbol{\mu}}_f, \hat{\boldsymbol{\Lambda}}_f \hat{\boldsymbol{\Lambda}}_f^\top + \hat{\boldsymbol{\Psi}}_f)}{\sum_{g'=1}^G \prod_{j=1}^J \hat{\tau}_{g'} \mathcal{N}_p(\mathbf{x}_j; \hat{\boldsymbol{\mu}}_{g'}, \hat{\boldsymbol{\Lambda}}_{g'} \hat{\boldsymbol{\Lambda}}_{g'}^\top + \hat{\boldsymbol{\Psi}}_{g'}) \sum_{f=1}^G \prod_{\substack{o \in B_2 \\ f \neq g' \\ o=1}} \hat{\tau}_f \mathcal{N}_p(\mathbf{x}_o; \hat{\boldsymbol{\mu}}_f, \hat{\boldsymbol{\Lambda}}_f \hat{\boldsymbol{\Lambda}}_f^\top + \hat{\boldsymbol{\Psi}}_f)}, \quad (3)$$

where $\hat{\cdot}$ denotes an initial or current parameter value, as relevant. In (3), the posterior probabilities of pixels in B_1 are considered with only one set of pixels B_2 as a negative constraint, however (3) can easily be extended to accommodate multiple such negative constraints. The remaining pixels in Figure 2 which are not in the highlighted blocks are considered as blocks with only one pixel and no negative constraints. The estimated model parameters $\hat{\boldsymbol{\mu}}_g$ and $\hat{\tau}_g$ are updated in the M-step of the first cycle as in PGMM (see McNicholas and Murphy, 2008). Further details are in Appendix 9.

In the E-step of the second cycle, the expected value of $\hat{z}_{B_1, g}^-$ is computed as in (3), with $\hat{\boldsymbol{\mu}}_g$ and $\hat{\tau}_g$ updated to those from the M-step of the first cycle. The expected value of \mathbf{u}_n , $n = 1, \dots, N$, and the estimates of $\hat{\boldsymbol{\Lambda}}_g$ and $\hat{\boldsymbol{\Psi}}_g$, $g = 1, \dots, G$ are computed as in PGMM; details are reported in Appendix 9.

The AECM algorithm moves between these cycles, iteratively updating the model parameters, until convergence. The values of $\hat{\mathbf{Z}}$ at convergence are the posterior probabilities of cluster membership for each pixel; labels can be obtained based, for example, on the *maximum a posteriori* values of $\hat{\mathbf{Z}}$. However, constrained-PGMM exhibits very long run times when fitted to the hyperspectral image data and could not be fitted to the entire $p = 101$ variables of the motivating dataset in a computationally efficient manner.

3.3 Consensus-constrained parsimonious Gaussian mixture models

To improve the computational efficiency of constrained-PGMM, we propose a consensus-constrained parsimonious Gaussian mixture model (ccPGMM), inspired by Russell et al. (2015). Russell et al. (2015) proposed a consensus approach to avoid the selection of a single best model based on the Bayesian information criterion (BIC, Neath and Cavanaugh, 2012; Schwarz, 1978). In Russell et al. (2015), the clustering solutions of multiple models fitted on the same data are consolidated to provide the final consensus clustering solution. Previously, Strehl and Ghosh (2002) proposed three ensemble clustering methods namely a cluster-based similarity partitioning algorithm, a hyper-graph partitioning algorithm, and a meta-clustering algorithm to combine multiple clustering solutions into one, but the uncertainty in cluster membership was not accounted for. Fern and Brodley (2003) and Punera and Ghosh (2008) extended these approaches, incorporating the posterior probabilities into the ensemble.

Unlike Russell et al. (2015), the proposed ccPGMM takes a divide-and-conquer approach. In ccPGMM, a constrained-PGMM is fitted to M subsets (indexed by m) of $d < p$ randomly selected variables from the high dimensional NIR spectrum. The M posterior probabilities which account for

the uncertainty of the cluster memberships of the N pixels are then consolidated to provide the final consensus clustering solution as follows. An $N \times N$ similarity matrix \mathbf{S}^m for $m = 1, \dots, M$ is computed based on the estimated posterior probabilities $\hat{\mathbf{Z}}^m$ available on convergence of the AECM after fitting the constrained-PGMM to the subset m . Each entry of \mathbf{S}^m is a similarity score S_{ij}^m computed as

$$S_{ij}^m = \begin{cases} \hat{\mathbf{z}}_i^m (\hat{\mathbf{z}}_j^m)^\top & \text{if } i \neq j, \\ 1 & \text{if } i = j, \end{cases}$$

where $\hat{\mathbf{z}}_i^m$ and $\hat{\mathbf{z}}_j^m$ are the posterior probabilities of cluster membership for observations i and j respectively. The M similarity matrices are then averaged to compute the final similarity matrix \mathbf{S} , whose entries S_{ij} are given by

$$S_{ij} = \frac{1}{M} \sum_{m=1}^M \sum_{g=1}^G \hat{z}_{ig}^m \hat{z}_{jg}^m,$$

where \hat{z}_{ig}^m and \hat{z}_{jg}^m are the posterior probabilities of observation i and j belonging to group g , inferred from data subset m . A dissimilarity matrix \mathbf{D} is then computed as $\mathbf{D} = 1 - \mathbf{S}$ and hierarchical clustering with complete linkage (Sokal, 1963) is performed employing \mathbf{D} . As the number of clusters G is known based on the application at hand, the resulting dendrogram is cut to give G clusters and the final consensus clustering solution.

3.4 Technical details

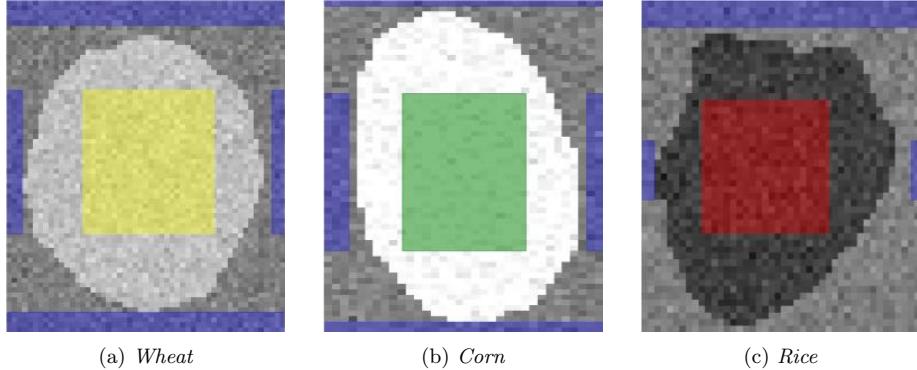
In the proposed ccPGMM, a single constrained-PGMM with a fixed number of clusters G and a fixed number of factors q is fitted to M data subsets with d variables per subset; the values of G , q , M , and d require specification. The number of clusters G is known and fixed in advance according to the particular application. For example, in the motivating application here (see Section 2) where there are three cereal types and the background, $G = 4$. The selection of the number of factors q is based on monitoring the proportion of variance explained by the first q principal components obtained from the application of principal component analysis (PCA) to the entire dataset. There is a trade-off between the values of M , the number of subsets of variables, and d , the number of variables per subset. Typically, d is smaller than p but too small values may provide little clustering information. Further, very large values of d hamper the computational feasibility. Also, the choice of M must be cognisant of the choice of d to ensure good coverage of all the variables in the M data subsets. These choices are explored further in sections 4 and 5.

In terms of specifying the constraints, there are technically no limits on the number of pixels that can be used. The size of the blocks of pixels (constraints) highlighted in Figure 2 can vary and the number of pixels in one block need not be the same as the number of pixels in the other blocks. Usually, the hyperspectral images of food samples are plotted as greyscale images, and the maximum region of pixels that are distinguishable as the background and as the relevant sample under study are selected as constraints. Increasing the number of positive constraints and the number of pixels within the positive constraints does not significantly increase the computational cost of fitting the ccPGMM, as in Melnykov et al. (2016). However, an increase in the number of negative constraints will result in more complex posterior probability calculations in (3), negatively impacting the computational cost. Investigations on the choice of constrained pixels are in sections 4 and 5.

The AECM algorithm, used to fit the constrained-PGMM to each data subset, is initialized with the cluster labels obtained from applying k -means clustering on the respective data subset. The initial values of Λ_g and Ψ_g are computed based on the eigenvalue decomposition of Σ_g (McNicholas and Murphy, 2008). The relative change in the log-likelihood is used to assess the convergence of the AECM with a tolerance of $1e - 6$.

4 Simulation studies

Simulation studies are conducted to assess the performance of ccPGMM and compare it to the performance of the threshold approach and existing state-of-the-art methods such as DBSCAN (Khan et al., 2014; Hahsler et al., 2019), Gaussian mixture models (GMM, Scrucca et al., 2016; McNicholas, 2016;



(a) Wheat

(b) Corn

(c) Rice

Figure 3: *Greyscale image of three simulated hyperspectral images for one of the five well-separated cluster datasets. The pixels in the blue blocks are background, the pixels in the yellow block are wheat, the pixels in the green block are corn and the pixels in the red block are rice.*

Bouveyron et al., 2019), PGMM (McNicholas and Murphy, 2008) and consensus PGMM (cPGMM) i.e., ccPGMM with no constraints. The convergence of each model-based clustering approach is assessed with the default parameter settings detailed in the respective R packages. Four scenarios are considered: well-separated clusters (scenario 1), heavily overlapping clusters (scenario 2), mildly overlapping clusters (scenario 3), and synthetic puffed cereal data (scenario 4). For each scenario, five datasets are simulated.

In scenarios 1–3, each simulated dataset contains three simulated hyperspectral images (one for each cereal type), giving a total of $N = 7825$ pixels, where each pixel has an associated $p = 101$ spectrum of strongly correlated variables. The three images are simulated according to labels resulting from applying the threshold approach to three of the hyperspectral images in the motivating dataset detailed in Section 2. The p variables for each of the three types of cereal and the background pixels are generated from different factor analysis models with $q = 3$ and parameters μ_g , Λ_g , and Ψ_g for $g = 1, \dots, G = 4$. To emulate the real data, a strong correlation is induced between the p variables at each pixel by, for every set of 5 consecutive variables, generating the associated values in the loadings matrix from a $\mathcal{N}_q(\mathcal{U}(0.3, 0.9), 0.03)$ and the values of $\text{diag}(\Psi_g)$ from a $\Gamma(\mathcal{U}(0, 0.1), 1)$. The values of the means μ_g for $g = 1, \dots, G$ are used to control the degree of overlap of the clusters in each scenario; details are given in the respective subsections of Section 4. To determine the constraints, the greyscale images of the simulated hyperspectral images are examined to decide on the sets of pixels to be used as constraints, as shown in Figure 3. Here, this resulted in 2,956 pixels of the $N = 7825$ pixels (37.7%) being used to inform the constraints. The 1,356 pixels in the blue blocks (background) must be clustered together, as must the 750 pixels in the yellow block (wheat), the 450 pixels in the green block (corn), and the 400 pixels in the red block (rice). These are the positive constraints. The pixels in one coloured block must not be clustered together with the pixels in another coloured block; these are the negative constraints.

In scenario 4, a synthetic cereal dataset is considered which mirrors the properties of the motivating hyperspectral images. Each dataset contains $L = 9$ simulated hyperspectral images (three for each cereal type) with $N = 28,039$ and where each pixel has an associated $p = 101$ length spectrum. The p variables for each cereal type and background pixels are generated from the relevant factor analyzer model, fitted to the real data; details are discussed in Section 4.4. In this scenario, 12,215 pixels of the $N = 28,039$ pixels (43.5%) are selected as constraints by plotting the greyscale images of the synthetic puffed cereals. Of the 12,215 pixels selected as constraints, 5,900 are background, 3,130 are wheat, 1,535 are corn, and 1,650 are rice. Further, to assess the impact of the proportion of pixels used as constraints on the clustering performance, a smaller set of 6,951 pixels (24.7%) are selected as constraints. Of the 6,951 pixels, 3,801 are background, 1,540 are wheat, 783 are corn and 827 are rice. The set of the 43.5% of pixels and the 24.7% of pixels used as constraints are illustrated in Figures 13 and 14 respectively in Appendix 10.

The simulation study is conducted on an Intel(R) core(TM) i7-10850H CPU @ 2.7 GHz processor with 16 GB RAM and 64-bit operating system. The code used to generate the datasets for each

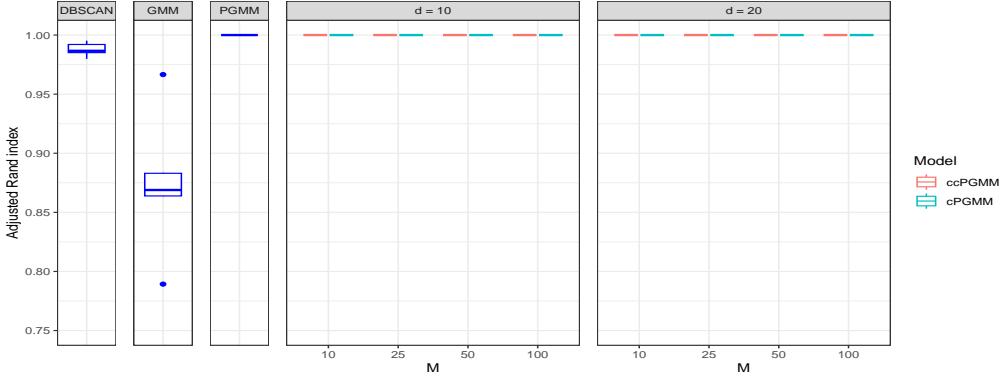


Figure 4: For each of the five simulated datasets with well-separated clusters, the ARI between the known labels and the clustering solutions of DBSCAN, GMM, and PGMM fitted on $p = 101$ variables (first three panels from the left) and cPGMM and ccPGMM (last two panels on the right) fitted with different settings of M and d .

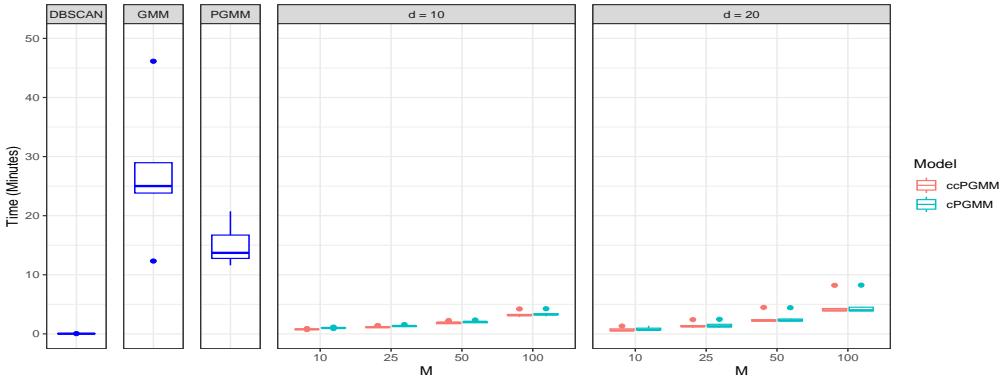


Figure 5: For each of the five simulated datasets with well-separated clusters, the time taken to fit DBSCAN, GMM, and PGMM on $p = 101$ variables (first three panels from the left) and cPGMM and ccPGMM (last two panels on the right) fitted with different settings of M and d .

scenario and to conduct the simulation studies is available on [GitHub](#).

4.1 Scenario 1: well-separated clusters

To generate the well-separated clusters, for $g = 1, \dots, 4$, the cluster mean parameters are generated as $\mu_g \sim \mathcal{N}_p(a, 0.5)$, where $a \in \{0, 5, 10, -5\}$; the p variables of the background, wheat, corn, and rice pixels in each dataset are generated from the four well-separated factor analyzer models, respectively. For DBSCAN, the minimum points required in the neighborhood for core points are set to $2p$, and for the GMM, $G = 4$ is determined by the setting. For the PGMM approaches, $G = 4$ and $q = 1$, where q is informed by fitting PCA to the entire dataset and monitoring the proportion of variance explained by the components. For consensus based approaches, $M = \{10, 25, 50, 100\}$ randomly selected subsets of variables are considered with the number of variables per subset $d = \{10, 20\}$. The positive and negative constraints for ccPGMM are selected as shown in Figure 3.

Figure 4 shows the adjusted Rand index(Hubert and Arabie, 1985) between the known labels and the clustering solutions for the five simulated datasets in scenario 1 under the DBSCAN, GMM, PGMM, cPGMM, and ccPGMM approaches. Figure 5 shows the time taken in minutes to fit the methods. All approaches demonstrate strong clustering performance. In terms of computational cost,

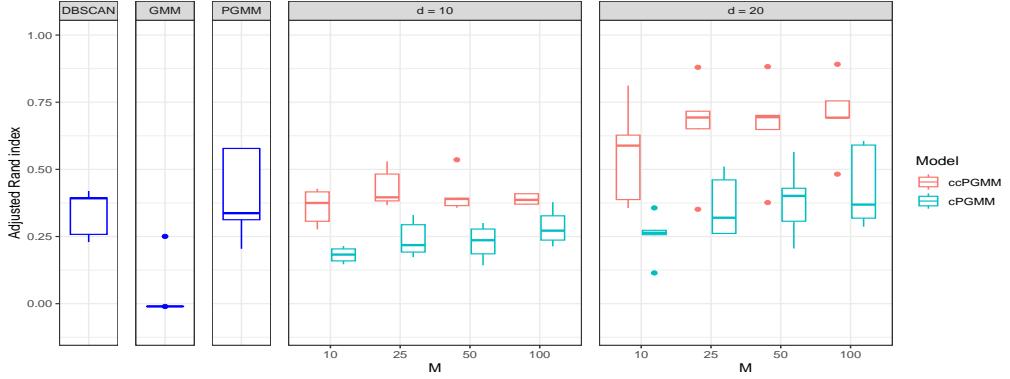


Figure 6: For each of the five datasets with heavily overlapping clusters, the ARI of DBSCAN, GMM, and PGMM fitted on $p = 101$ variables (first three panels from the left) and cPGMM and ccPGMM (last two panels on the right) with different settings of M and d .

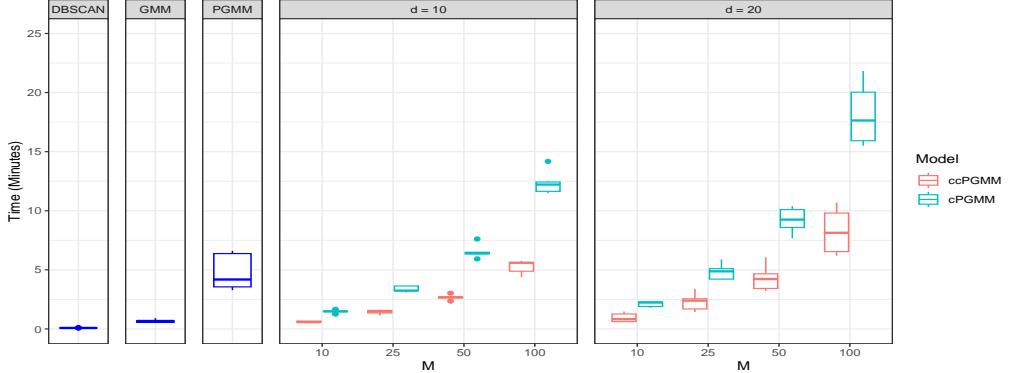


Figure 7: For each of the five datasets with heavily overlapping clusters, the time taken to fit DBSCAN, GMM, and PGMM on $p = 101$ variables (first three panels from the left) and cPGMM and ccPGMM (last two panels on the right) with different settings of M and d .

DBSCAN clusters each dataset in less than a minute while GMM and PGMM take an average of 27 minutes and 15 minutes respectively. The cPGMM and ccPGMM fitted with different settings of M and d have lower computational costs than GMM and PGMM, but similar to DBSCAN, with no loss of accuracy.

4.2 Scenario 2: heavily overlapping clusters

To generate data with heavily overlapping clusters, the cluster mean parameters are generated as $\mu_g \sim \mathcal{N}_p(a, 1.5)$, where $a \in \{1, 1.5, 3, -1.5\}$ for $g = 1, \dots, 4$ respectively. Figures 6 and 7 show the ARI between the known pixel labels and the clustering solutions of DBSCAN, GMM, PGMM, cPGMM, and ccPGMM and the time taken in minutes by each method, respectively. DBSCAN clusters the datasets with an average ARI of 0.34 in less than a minute on average. The GMM struggles to uncover the clustering structure. The PGMM approach performs competitively, achieving an average ARI of 0.40 in 5 minutes on average. The ccPGMM approach performs well in this scenario: with $M \geq 25$ and $d = 20$ an average ARI of 0.66 is achieved, which outperforms PGMM fitted to all p variables. In terms of clustering performance, ccPGMM performs better than cPGMM across all settings of M and d , in comparable time.

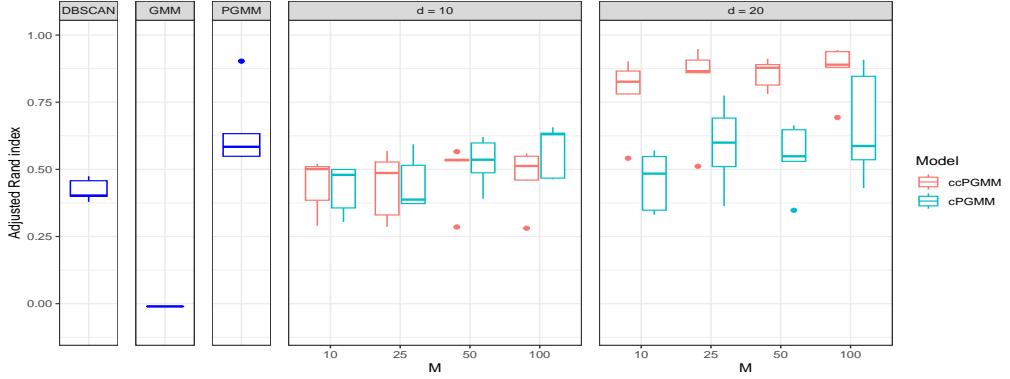


Figure 8: For each of the five datasets with mildly overlapping clusters, the ARI of DBSCAN, GMM, and PGMM fitted on $p = 101$ variables (first three panels from the left) and cPGMM and ccPGMM (last two panels on the right) fitted with different settings of M and d .

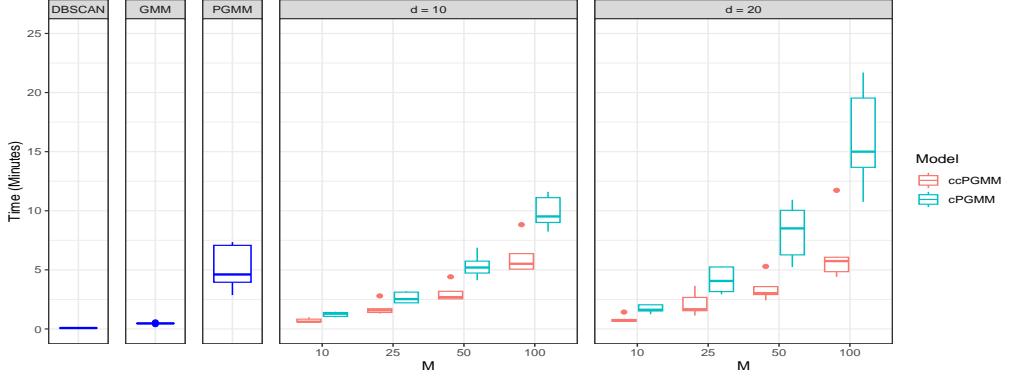


Figure 9: For each of the five datasets with mildly overlapping clusters, the time taken to fit DBSCAN, GMM, and PGMM on $p = 101$ variables (first three panels from the left) and cPGMM and ccPGMM (last two panels on the right) fitted with different settings of M and d .

4.3 Scenario 3: mildly overlapping clusters

The p variables of the background, wheat, corn, and rice pixels in each dataset are generated from four mildly overlapping factor analyzer models, with the cluster mean parameters generated as $\mu_g \sim \mathcal{N}(a, 1.5)$ where $a = \{0, 2, 5, -2\}$ for $g = 1, \dots, 4$, respectively. Figures 8 and 9 show the ARI of the cluster solutions of DBSCAN, GMM, PGMM, cPGMM, and ccPGMM fitted on the five datasets with mildly overlapping clusters and the time taken in minutes to fit them. While GMM again struggles to uncover the clustering structure, DBSCAN and PGMM fitted to all p variables clustered the pixels with an average ARI of 0.42 and 0.64 respectively. In terms of cPGMM and ccPGMM, Figure 8 shows that increasing d from 10 to 20 significantly improves the clustering performance of both methods, with the performance of ccPGMM being the best among the approaches considered. Run times are comparable overall, with cPGMM with the highest values of M and d intuitively taking the longest. The ccPGMM approach with $M \geq 10$ and $d = 20$ is competitively computationally efficient with an average ARI > 0.80 , which has again a much better performance than PGMM fitted to all p variables.

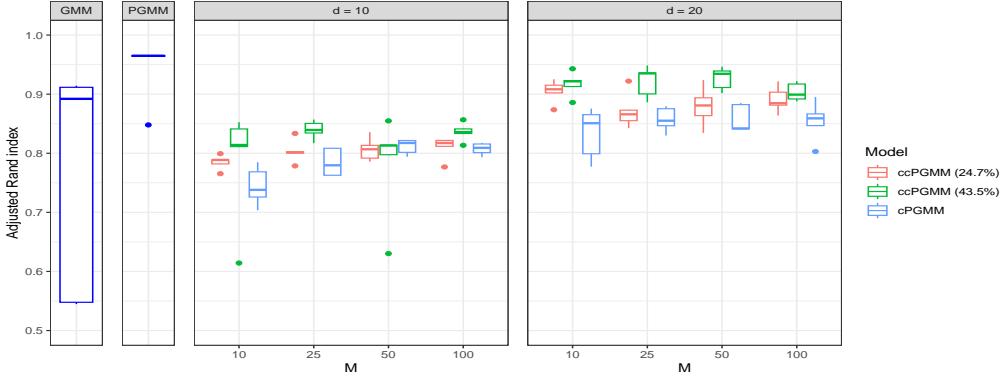


Figure 10: For each of the five synthetic cereal datasets, the ARI of GMM, and PGMM fitted on $p = 101$ variables (first two panels from the left) and cPGMM and ccPGMM (last two panels on the right) fitted with different settings of M and d . For ccPGMM, two settings with 24.7% and 43.5% of pixels used as constraints are shown.

4.4 Scenario 4: synthetic puffed cereal data

Here a synthetic cereal dataset is considered which mirrors the properties of the motivating hyperspectral images. To generate the data, PCA is fitted to the real puffed cereal images and, based on the cumulative proportion of variance explained (Table 2 in Appendix 11), $q = 2$ is selected. Threshold labels, generated as proposed by Xu et al. (2020), are used to identify the pixels belonging to the 3 cereal types and the background in the real puffed cereal images. Four individual factor analysis models with $q = 2$ are then fitted to pixels in the motivating dataset from each label type. Finally, to generate the synthetic cereal images, for each pixel, p variables are generated from the resulting factor analysis models fitted to the background, wheat, corn, and rice pixels in real data.

Figures 10 and 11 show, respectively, the ARI of the cluster solutions of the different clustering methods fitted to the five synthetic cereal datasets, and the time taken in minutes for estimation. Intuitively, since the synthetic cereal datasets are generated from a mixture of factor analyzers, PGMM fitted to all p variables shows strong clustering performance (average ARI of 0.94), however, it takes an average of 50 hours to cluster each synthetic cereal dataset. The GMM approach has a much lower computational cost than PGMM but the quality of the clustering solutions is inconsistent. For DBSCAN (not illustrated), while computationally cheap, clustering performance is poor with an average ARI of 0.32 with a variance of 0.11. The ccPGMM with 43.5% of pixels as constraints and ccPGMM with 24.7% of pixels as constraints fitted with $M \geq 10$ and $d = 20$ shows strong clustering performances. Intuitively, ccPGMM with larger set of constraints (43.5%) performs slightly better than ccPGMM with smaller set of constraints (24.7%) and cPGMM across different settings of M and d . For the ccPGMM approach with 43.5% pixels as constraints, fitted with $M \geq 25$ and $d = 20$ has an average ARI of 0.92. Overall, ccPGMM shows comparable clustering performance to PGMM fitted to all p variables, in less than one-tenth of the computational time.

5 Application to hyperspectral images of puffed cereals

To simultaneously cluster the pixels of the 9 hyperspectral images in the motivating dataset, we apply ccPGMM with $G = 4$, as there are 3 puffed cereal types (wheat, corn, and rice) and the background, and $q = 2$ based on Table 2 in Appendix 11. On the basis of the performance observed in simulation study scenario 4 (Section 4.4) which mirrored the real data, $M = 25$ subsets and $d = 20$ variables per subset were employed. As detailed in Section 4, the greyscale version of the hyperspectral images were considered and, based on pixels for which the labelling was clear, 12,215 or (43.5%) of the $N = 28,039$ pixels were selected; these pixels informed the constraints. Also, a smaller set of 6,951 pixels out of the $N = 28,039$ pixels (24.7%) are selected as constraints to assess the impact of the proportion of pixels used as constraints on the clustering performance. For comparison purposes, DBSCAN, GMM

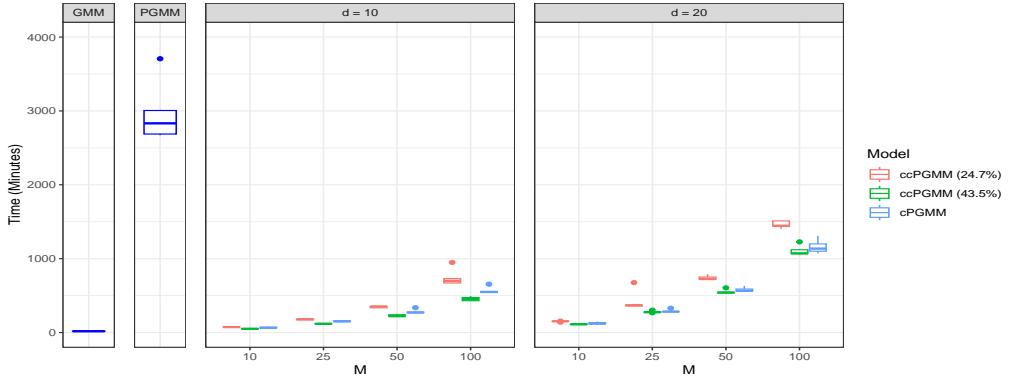


Figure 11: For each of the five synthetic cereal datasets, the time taken to fit GMM, and PGMM on $p = 101$ variables (first two panels from the left) and cPGMM and ccPGMM (last two panels on the right) fitted with different settings of M and d . For ccPGMM, two settings with 24.7% and 43.5% of pixels used as constraints are shown.

with $G = 4$, and PGMM with $G = 4$ and $q = 2$ are fitted to the motivating dataset with all p variables. The threshold approach of Xu et al. (2020) is also applied, individually, to each hyperspectral image to label the pixels.

Figure 12 shows the greyscale images for three of the nine hyperspectral images, and the pixel labels as generated using the threshold approach, DBSCAN, GMM, PGMM, ccPGMM with 43.5% pixels as constraints and ccPGMM with 24.7% pixels as constraints; Figures 15 and 16 in Appendix 12 illustrate the same for the remaining six hyperspectral images. Figures 12d, 12e and 12f show the labels generated for each hyperspectral image by the threshold approach. Each type of pixel is coloured in a different shade of blue according to its label; the resulting labels are very close to the respective greyscale images (12a, 12b, and 12c). However, the threshold approach is ad-hoc involving subjective choice, and does not provide any clustering uncertainties. Figures 12g, 12h and 12i show the cluster solutions of DBSCAN where each pixel is coloured in a different shade of green depending on its label. While DBSCAN correctly distinguishes the background pixels from the cereal pixels in each hyperspectral image, the pixels of the three different cereals are incorrectly allocated to the same cluster. Similarly, GMM clusters the pixels of the three types of cereals into a single cluster as shown in Figures 12j, 12k, and 12l. In addition, GMM allocates the uncertain edge pixels of the cereal grains to a separate cluster and allocates some pixels that likely correspond to spectral noise into another separate cluster. Figures 12m, 12n, and 12o illustrate the pixel labels based on the application of PGMM to all p variables. Each label obtained from the fitted PGMM is coloured in a different shade of purple. Though PGMM has correctly allocated several background and cereal pixels to their respective separate clusters, it also appears to have incorrectly clustered some together, giving non-contiguous images. In Figure 12n, which shows a corn cereal, many background pixels are allocated to the corn cluster. In Figure 12m and 12o, corresponding to wheat and rice cereals respectively, several edge pixels of the cereals are incorrectly allocated to the corn cluster. In addition, PGMM takes nearly 72 hours to cluster the motivating dataset.

Finally, Figures 12p, 12q and 12r show the labels of the pixels, in different shades of brown, obtained by applying ccPGMM with 43.5% of pixels as constraints. Applying ccPGMM to all nine hyperspectral images simultaneously took only 6 hours and the cluster labels are close to labels obtained by applying the threshold approach to each image individually. Compared to the other clustering solutions considered, ccPGMM distinguishes the cereals well from the background while, overall, pixels are allocated to different clusters corresponding to corn, wheat, and rice. However, some pixels are still incorrectly mixed across clusters, particularly around the edges of the wheat and rice cereals. Figures 12s, 12t and 12u show the cluster labels obtained by applying ccPGMM with 24.7% pixels as constraints in different shades of brown. The ccPGMM fitted with 24.7% of pixels as constraints cluster the pixels as well as the ccPGMM with 43.5% of pixels as constraints. However, some pixels around the edges of the wheat and corn cereal are incorrectly clustered as rice. Importantly, the uncertainties

associated with the cluster membership for each pixel are available given the model-based approach to clustering and the uncertainty plots are given in Figures 17, 18, and 19 in Appendix 13; intuitively the pixels around the edges of the cereals have the highest clustering uncertainties.

6 Discussion

A novel model-based approach to simultaneously labelling pixels of hyperspectral images while avail-ing of additional information about the relationship between groups of pixels within the images is proposed. The proposed ccPGMM demonstrates strong clustering performance when compared to existing approaches both in terms of quality of the clustering solution, and the computational time required to cluster such a large dataset.

While the ccPGMM approach is less subjective than the currently used threshold approach, user decisions must still be made. For example, the selection of the number of factors q , while well established, is subjective. Examining alternative approaches to choosing q either using model selection criteria or using shrinkage priors ([Bhattacharya and Dunson, 2011](#)) in a Bayesian context would address this, but increase the computational cost. Indeed, under the ccPGMM approach, allowing for different values of q when fitting to the M different subsets of variables could bring improved performance but would require further exploration. Further, the choice of M and d are currently data-driven and therefore subjective. Additionally, similar to [Jacques and Ruckebusch \(2016\)](#), considering co-clustering in the context of ccPGMM would facilitate both clustering of pixels and wavelengths.

Overall, the proposed ccPGMM model works well when clustering a large number of pixels in a set of hyperspectral images, and provides associated clustering uncertainty, in a computationally efficient manner. As such, the approach should have a broader application to other hyperspectral images where labelling of the pixels is required.

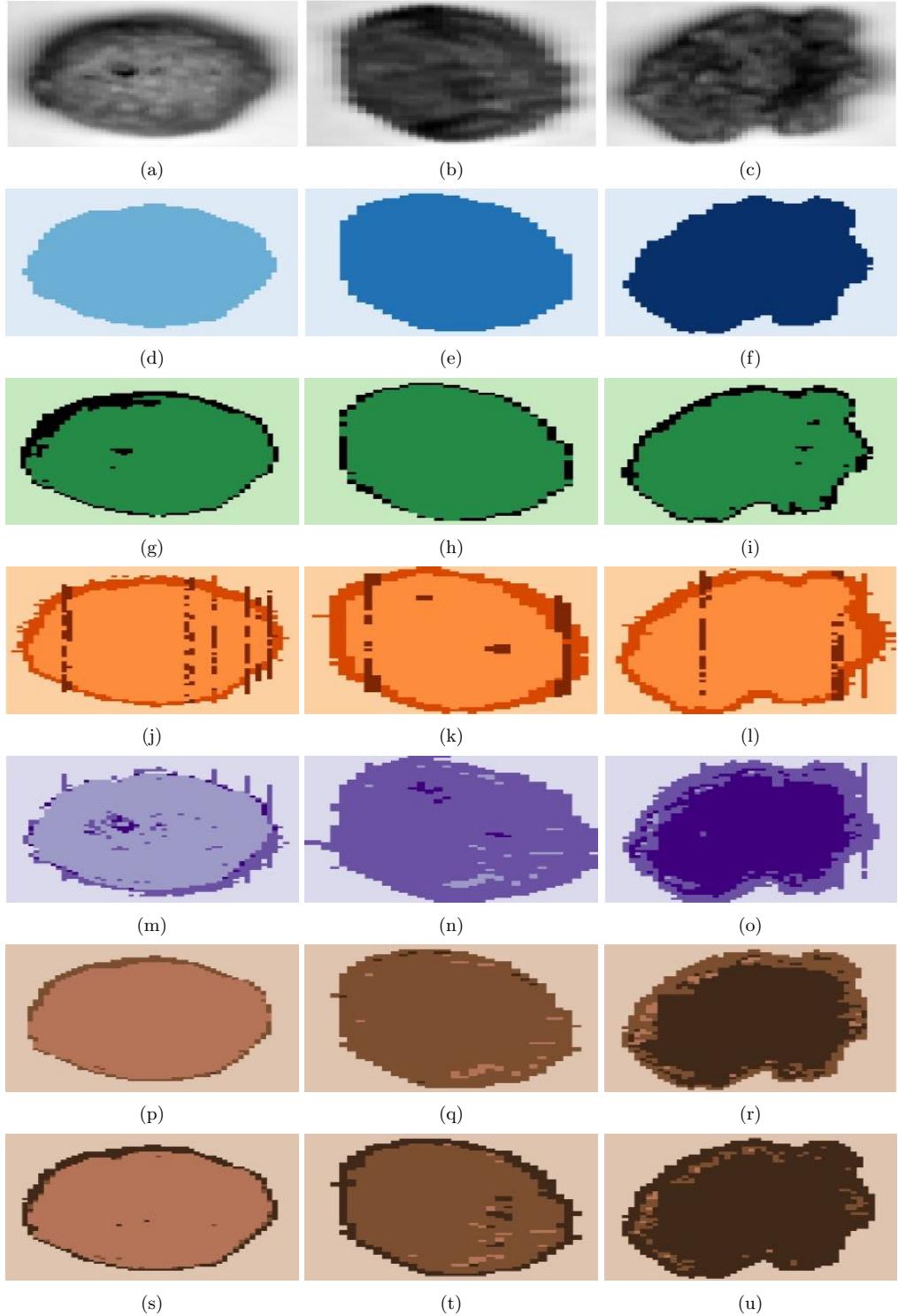


Figure 12: Greyscale images and pixel labels for three of the nine hyperspectral images ($l = \{1, 4, 7\}$) of the cereals using the threshold approach, DBSCAN, MBC, PGMM, ccPGMM with 43.5% of pixels as constraints, and ccPGMM with 24.7% of pixels as constraints respectively for wheat (12a, 12d, 12g, 12j, 12m, 12p, 12s), corn (12b, 12e, 12h, 12k, 12n, 12q, 12t) and rice (12c, 12f, 12i, 12l, 12o, 12r, 12u).

7 Acknowledgement

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6049. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

- Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.
- Amigo, J. M. (2010). Practical issues of hyperspectral imaging analysis of solid dosage forms. *Analytical and bioanalytical chemistry*, 398(1):93–109.
- Amigo, J. M. (2020). Hyperspectral and multispectral imaging: Setting the scene. In *Data Handling in Science and Technology*, volume 32, pages 3–16. Elsevier.
- Amigo, J. M., Babamoradi, H., and Elcoroaristizabal, S. (2015). Hyperspectral image analysis. a tutorial. *Analytica chimica acta*, 896:34–51.
- Amigo, J. M. and Santos, C. (2020). Preprocessing of hyperspectral and multispectral images. In *Data handling in science and technology*, volume 32, pages 37–53. Elsevier.
- Bhattacharya, A. and Dunson, D. B. (2011). Sparse Bayesian infinite factor models. *Biometrika*, 98(2):291–306.
- Bocklitz, T., Walter, A., Hartmann, K., Rösch, P., and Popp, J. (2011). How to pre-process Raman spectra for reliable and stable models? *Analytica chimica acta*, 704(1-2):47–56.
- Bouveyron, C. and Brunet-Sauvad, C. (2014). Discriminative variable selection for clustering with the sparse fisher-em algorithm. *Computational Statistics*, 29(3):489–513.
- Bouveyron, C., Celeux, G., Murphy, T. B., and Raftery, A. E. (2019). *Model-based clustering and classification for data science: with applications in R*, volume 50. Cambridge University Press.
- Feng, Y.-Z. and Sun, D.-W. (2012). Application of hyperspectral imaging in food safety inspection and control: a review. *Critical reviews in food science and nutrition*, 52(11):1039–1058.
- Fern, X. Z. and Brodley, C. E. (2003). Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 186–193.
- Geladi, P., MacDougall, D., and Martens, H. (1985). Linearization and scatter-correction for near-infrared reflectance spectra of meat. *Applied spectroscopy*, 39(3):491–500.
- Ghahramani, Z., Hinton, G. E., et al. (1996). The EM algorithm for mixtures of factor analyzers. Technical report, Technical Report CRG-TR-96-1, University of Toronto.
- Gowen, A. A., Xu, J.-L., and Herrero-Langreo, A. (2019). Comparison of spectral selection methods in the development of classification models from visible near infrared hyperspectral imaging data. *Journal of Spectral Imaging*, 8.
- Hahsler, M., Piekenbrock, M., and Doran, D. (2019). DBscan: Fast density-based clustering with R. *Journal of Statistical Software*, 91:1–30.
- Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2:193–218.
- Jacques, J. and Ruckebusch, C. (2016). Model-based co-clustering for hyperspectral images. *Journal of Spectral Imaging*.
- Khan, K., Rehman, S. U., Aziz, K., Fong, S., and Sarasvady, S. (2014). DBscan: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pages 232–238. IEEE.
- McLachlan, G. and Peel, D. (2000). Mixtures of factor analyzers. In *In Proceedings of the Seventeenth International Conference on Machine Learning*. Citeseer.

- McLachlan, G. J. and Peel, D. (2004). *Finite mixture models*. John Wiley & Sons.
- McLachlan, G. J., Peel, D., and Bean, R. W. (2003). Modelling high-dimensional data by mixtures of factor analyzers. *Computational Statistics & Data Analysis*, 41(3-4):379–388.
- McNicholas, P. D. (2016). Model-based clustering. *Journal of Classification*, 33(3):331–373.
- McNicholas, P. D. and Murphy, T. B. (2008). Parsimonious Gaussian mixture models. *Statistics and Computing*, 18(3):285–296.
- McNicholas, P. D., Murphy, T. B., McDaid, A. F., and Frost, D. (2010). Serial and parallel implementations of model-based clustering via parsimonious Gaussian mixture models. *Computational Statistics & Data Analysis*, 54(3):711–723.
- Melnikov, V., Melnykov, I., and Michael, S. (2016). Semi-supervised model-based clustering with positive and negative constraints. *Advances in data analysis and classification*, 10(3):327–349.
- Neath, A. A. and Cavanaugh, J. E. (2012). The Bayesian information criterion: background, derivation, and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(2):199–203.
- Pal, N. R. and Pal, S. K. (1993). A review on image segmentation techniques. *Pattern recognition*, 26(9):1277–1294.
- Punera, K. and Ghosh, J. (2008). Consensus-based ensembles of soft clusterings. *Applied Artificial Intelligence*, 22(7-8):780–810.
- R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Rinnan, Å., Van Den Berg, F., and Engelsen, S. B. (2009). Review of the most common pre-processing techniques for near-infrared spectra. *TrAC Trends in Analytical Chemistry*, 28(10):1201–1222.
- Russell, N., Murphy, T. B., and Raftery, A. E. (2015). Bayesian model averaging in model-based clustering and density estimation. *arXiv preprint arXiv:1506.09035*.
- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.
- Scrucca, L., Fop, M., Murphy, T. B., and Raftery, A. E. (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1):289–317.
- Sokal, R. R. (1963). The principles and practice of numerical taxonomy. *Taxon*, pages 190–199.
- Strehl, A. and Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617.
- Xu, J.-L., Riccioli, C., Herrero-Langreo, A., and Gowen, A. A. (2020). Deep learning classifiers for near infrared spectral imaging: a tutorial. *Journal of Spectral Imaging*, 9.
- Zhai, H., Zhang, H., Li, P., and Zhang, L. (2021). Hyperspectral image clustering: Current achievements and future lines. *IEEE Geoscience and Remote Sensing Magazine*, 9(4):35–67.

8 Appendix A

As discussed in Section 2, Table 1 details the dimensionality of the nine hyperspectral images of the three puffed cereals.

Table 1: *The dimensionality of motivating hyperspectral images of puffed cereals.*

Puffed cereal type	l	S_l	T_l	$S_l \times T_l$
Wheat	1	67	53	3551
Wheat	2	62	64	3968
Wheat	3	65	55	3575
Corn	4	61	34	2074
Corn	5	39	69	2691
Corn	6	56	54	3024
Rice	7	56	49	2744
Rice	8	78	54	4212
Rice	9	50	44	2200

9 Appendix B: AECM for constrained-PGMM

In constrained-PGMM, the observed data log-likelihood is

$$\ell_o(\mathbf{X}) = \sum_{n=1}^N \log \sum_{g=1}^G \tau_g f(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\Psi}_g). \quad (4)$$

As (4) is difficult to maximize directly, an AECM algorithm is used to fit the constrained-PGMM. AECM works in two cycles and allows for a different definition of the complete data in each cycle to estimate the parameters.

9.1 First cycle

In the first cycle, the component indicators $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$ are assumed to be the missing data, where \mathbf{z}_N is a binary vector of length G . The complete data log-likelihood is

$$\ell_c(\mathbf{X}, \mathbf{Z}) = \sum_{n=1}^N \sum_{g=1}^G z_{ng} [\log \tau_g + \log f(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\Psi}_g)], \quad (5)$$

where $f(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\Psi}_g) \sim \mathcal{N}_p(\boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g \boldsymbol{\Lambda}_g^\top + \boldsymbol{\Psi}_g)$.

9.1.1 E-step:

The expected complete data log-likelihood is,

$$Q_1(\boldsymbol{\mu}_g, \tau_g) = \sum_{n=1}^N \sum_{g=1}^G E[z_{ng}; \mathbf{x}_n] \left[\log \tau_g - \frac{p}{2} \log(2\pi) - \frac{1}{2} \log(|\boldsymbol{\Lambda}_g \boldsymbol{\Lambda}_g^\top + \boldsymbol{\Psi}_g|) - \frac{1}{2} [(\mathbf{x}_n - \boldsymbol{\mu}_g)^\top (\boldsymbol{\Lambda}_g \boldsymbol{\Lambda}_g^\top + \boldsymbol{\Psi}_g)^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_g)] \right].$$

The posterior probability \hat{z}_{ng} of observation n belonging to cluster g is

$$E[z_{ng} | \mathbf{x}_n] = \hat{z}_{ng} = \Pr(z_{ng} = 1; \mathbf{x}_n, \hat{\tau}_g, \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Lambda}}_g, \hat{\boldsymbol{\Psi}}_g),$$

$$\hat{z}_{ng} = \frac{\hat{\tau}_g \mathcal{N}_p(\mathbf{x}_n; \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Lambda}}_g \hat{\boldsymbol{\Lambda}}_g^\top + \hat{\boldsymbol{\Psi}}_g)}{\sum_{g'}^G \hat{\tau}_{g'} \mathcal{N}_p(\mathbf{x}_n; \hat{\boldsymbol{\mu}}_{g'}, \hat{\boldsymbol{\Lambda}}_{g'} \hat{\boldsymbol{\Lambda}}_{g'}^\top + \hat{\boldsymbol{\Psi}}_{g'})}.$$

where $\hat{\cdot}$ denotes an initial or current parameter estimate, as relevant.

Let B_1 be the set of J pixels (indexed by j) in the blue blocks shown in Figure 2 which are known to be background and must be clustered together. This is a positive constraint (denoted by +). The posterior probability $\hat{z}_{B_1,g}^+$ of the pixels in B_1 belonging to cluster g given the positive constraints (+) is

$$\hat{z}_{B_1,g}^+ = \frac{\prod_{\substack{j \in B_1 \\ j=1}}^J \hat{\tau}_g \mathcal{N}_p(\mathbf{x}_j; \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Lambda}}_g \hat{\boldsymbol{\Lambda}}_g^\top + \hat{\boldsymbol{\Psi}}_g)}{\sum_{g'=1}^G \prod_{\substack{j \in B_1 \\ j=1}}^J \hat{\tau}_{g'} \mathcal{N}_p(\mathbf{x}_j; \hat{\boldsymbol{\mu}}_{g'}, \hat{\boldsymbol{\Lambda}}_{g'} \hat{\boldsymbol{\Lambda}}_{g'}^\top + \hat{\boldsymbol{\Psi}}_{g'})}.$$

Let B_2 be the set of O pixels (indexed by o) in the yellow block of Figure 2 which are known to be wheat and should be clustered together. These are positive constraints (denoted +). However, the pixels in B_1 should not be clustered together with the pixels in B_2 . This is a negative constraint (denoted -). Let A be the event that pixels in B_1 belong to cluster g and B be the event that pixels in B_2 belong to any cluster f in G but $f \neq g$ and vice versa. Then the probability of event A and B occurring together is

$$\Pr(A \cap B) = \Pr(A) \Pr(B). \quad (6)$$

Based on equation (6), the posterior probability $\hat{z}_{B_1,g}^-$ of the pixels in B_1 belonging to cluster g , given the positive and negative constraints $_-^+$ is

$$\hat{z}_{B_1,g}^- = \frac{\prod_{\substack{j \in B_1 \\ j=1}}^J \hat{\tau}_g \mathcal{N}_p(\mathbf{x}_j; \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Lambda}}_g \hat{\boldsymbol{\Lambda}}_g^\top + \hat{\boldsymbol{\Psi}}_g) \sum_{\substack{f=1 \\ f \neq g}}^G \prod_{\substack{o \in B_2 \\ o=1}}^O \hat{\tau}_f \mathcal{N}_p(\mathbf{x}_o; \hat{\boldsymbol{\mu}}_f, \hat{\boldsymbol{\Lambda}}_f \hat{\boldsymbol{\Lambda}}_f^\top + \hat{\boldsymbol{\Psi}}_f)}{\sum_{g'=1}^G \prod_{\substack{j \in B_1 \\ j=1}}^J \hat{\tau}_{g'} \mathcal{N}_p(\mathbf{x}_j; \hat{\boldsymbol{\mu}}_{g'}, \hat{\boldsymbol{\Lambda}}_{g'} \hat{\boldsymbol{\Lambda}}_{g'}^\top + \hat{\boldsymbol{\Psi}}_{g'}) \sum_{\substack{f=1 \\ f \neq g}}^G \prod_{\substack{o \in B_2 \\ o=1}}^O \hat{\tau}_f \mathcal{N}_p(\mathbf{x}_o; \hat{\boldsymbol{\mu}}_f, \hat{\boldsymbol{\Lambda}}_f \hat{\boldsymbol{\Lambda}}_f^\top + \hat{\boldsymbol{\Psi}}_f)}. \quad (7)$$

Similarly, the posterior probability $\hat{z}_{B_2,g}^+$ of the pixels in B_2 belonging to cluster g , given the positive and negative constraints $_-^+$ is

$$\hat{z}_{B_2,g}^+ = \frac{\prod_{\substack{o \in B_2 \\ o=1}}^O \hat{\tau}_g \mathcal{N}_p(\mathbf{x}_o; \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Lambda}}_g \hat{\boldsymbol{\Lambda}}_g^\top + \hat{\boldsymbol{\Psi}}_g) \sum_{\substack{f=1 \\ f \neq g}}^G \prod_{\substack{j \in B_1 \\ j=1}}^J \hat{\tau}_f \mathcal{N}_p(\mathbf{x}_j; \hat{\boldsymbol{\mu}}_f, \hat{\boldsymbol{\Lambda}}_f \hat{\boldsymbol{\Lambda}}_f^\top + \hat{\boldsymbol{\Psi}}_f)}{\sum_{g'=1}^G \prod_{\substack{o \in B_2 \\ o=1}}^O \hat{\tau}_{g'} \mathcal{N}_p(\mathbf{x}_o; \hat{\boldsymbol{\mu}}_{g'}, \hat{\boldsymbol{\Lambda}}_{g'} \hat{\boldsymbol{\Lambda}}_{g'}^\top + \hat{\boldsymbol{\Psi}}_{g'}) \sum_{\substack{f=1 \\ f \neq g}}^G \prod_{\substack{j \in B_1 \\ j=1}}^J \hat{\tau}_f \mathcal{N}_p(\mathbf{x}_j; \hat{\boldsymbol{\mu}}_f, \hat{\boldsymbol{\Lambda}}_f \hat{\boldsymbol{\Lambda}}_f^\top + \hat{\boldsymbol{\Psi}}_f)}.$$

The remaining pixels in Figure 2 which are not in the highlighted blocks are considered as blocks with only one pixel and no negative constraints.

9.1.2 CM-step:

The expected complete data log-likelihood can be rewritten as

$$Q_1(\boldsymbol{\mu}_g, \tau_g) = \sum_{n=1}^N \sum_{g=1}^G \hat{z}_{ng}^+ \left[\log \tau_g - \frac{p}{2} \log(2\pi) - \frac{1}{2} \log(|\boldsymbol{\Lambda}_g \boldsymbol{\Lambda}_g^\top + \boldsymbol{\Psi}_g|) - \frac{1}{2} [(\mathbf{x}_n - \boldsymbol{\mu}_g)^\top (\boldsymbol{\Lambda}_g \boldsymbol{\Lambda}_g^\top + \boldsymbol{\Psi}_g)^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_g)] \right]. \quad (8)$$

Differentiating (8) with respect to μ_g ,

$$\hat{\mu}_g = \frac{\sum_{n=1}^N \hat{z}_{ng}^+ \mathbf{x}_n}{\sum_{n=1}^N \hat{z}_{ng}^+}.$$

Differentiating (8) with respect to τ_g ,

$$\hat{\tau}_g = \frac{\sum_{n=1}^N \hat{z}_{ng}^+}{N}.$$

9.2 Second cycle

In the second cycle, both component indicators $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$ and latent factors $\mathbf{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ are assumed to be the missing data. Therefore, the complete data log-likelihood is

$$\ell_c(\mathbf{X}, \mathbf{Z}, \mathbf{U}) = \sum_{n=1}^N \sum_{g=1}^G z_{ng} [\log \tau_g + \log f(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\Psi}_g, \mathbf{u}_n) + \log f(\mathbf{u}_n)],$$

where $f(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\Psi}_g, \mathbf{u}_n) \sim \mathcal{N}_p(\boldsymbol{\mu}_g + \boldsymbol{\Lambda}_g \mathbf{u}_n, \boldsymbol{\Psi}_g)$ and $f(\mathbf{u}_n) \sim \mathcal{N}_q(0, I)$.

The complete data log-likelihood can be rewritten as,

$$\begin{aligned} \ell_c(\mathbf{X}, \mathbf{Z}, \mathbf{U}) = C &+ \sum_{g=1}^G \left[\mathbf{n}_g \log \tau_g - \frac{\mathbf{n}_g}{2} \log(|\boldsymbol{\Psi}_g|) - \frac{1}{2} \left[\sum_{n=1}^N z_{ng} [(\mathbf{x}_n - \boldsymbol{\mu}_g)^\top \boldsymbol{\Psi}_g^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_g)] \right. \right. \\ &+ \sum_{n=1}^N z_{ng} [(\mathbf{x}_n - \boldsymbol{\mu}_g)^\top \boldsymbol{\Psi}_g^{-1} (\boldsymbol{\Lambda}_g \mathbf{u}_n)] \\ &\left. \left. - \frac{1}{2} (\boldsymbol{\Lambda}_g^\top \boldsymbol{\Psi}_g^{-1} \boldsymbol{\Lambda}_g \sum_{n=1}^N z_{ng} \mathbf{u}_n \mathbf{u}_n^\top)^\top \right] \right], \end{aligned}$$

where C is the constant and $\mathbf{n}_g = \sum_{n=1}^N z_{ng}$.

9.2.1 E-step:

The expected complete data log-likelihood is,

$$\begin{aligned} Q_2(\boldsymbol{\Lambda}_g, \boldsymbol{\Psi}_g) = C &+ \sum_{g=1}^G \left[\mathbf{n}_g \log \tau_g - \frac{\mathbf{n}_g}{2} \log(|\boldsymbol{\Psi}_g|) - \frac{\mathbf{n}_g}{2} (\boldsymbol{\Psi}_g^{-1} \mathbf{S}_g)^\top \right. \\ &+ \sum_{n=1}^N E[z_{ng}; \mathbf{x}_n] [(\mathbf{x}_n - \boldsymbol{\mu}_g)^\top \boldsymbol{\Psi}_g^{-1} (\boldsymbol{\Lambda}_g E[\mathbf{u}_n; \mathbf{x}_n])] \\ &\left. - \frac{1}{2} (\boldsymbol{\Lambda}_g^\top \boldsymbol{\Psi}_g^{-1} \boldsymbol{\Lambda}_g \sum_{n=1}^N E[z_{ng}; \mathbf{x}_n] E[\mathbf{u}_n \mathbf{u}_n^\top; \mathbf{x}_n])^\top \right], \end{aligned} \quad (9)$$

where $\mathbf{S}_g = \frac{1}{\mathbf{n}_g} \sum_{n=1}^N z_{ng} (\mathbf{x}_n - \boldsymbol{\mu}_g) (\mathbf{x}_n - \boldsymbol{\mu}_g)^\top$.

The expected value $E[z_{ng} | \mathbf{x}_n] = \hat{z}_{ng}$ is computed with $\boldsymbol{\mu}_g = \hat{\mu}_g$ and $\tau_g = \hat{\tau}_g$ updated in the CM-step of the first cycle with the known positive and negative constraints. The posterior probability $\hat{z}_{B_1,g}^+$ of the pixels in B_1 belonging to cluster g , given the positive and negative constraints $+$ is

$$\hat{z}_{B_1,g}^+ = \frac{\prod_{j=1}^J \hat{\tau}_g \mathcal{N}_p(\mathbf{x}_j; \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Lambda}}_g \hat{\boldsymbol{\Lambda}}_g^\top + \hat{\boldsymbol{\Psi}}_g) \sum_{f=1}^G \prod_{o \in B_2} \hat{\tau}_f \mathcal{N}_p(\mathbf{x}_o; \hat{\boldsymbol{\mu}}_f, \hat{\boldsymbol{\Lambda}}_f \hat{\boldsymbol{\Lambda}}_f^\top + \hat{\boldsymbol{\Psi}}_f)}{\sum_{g'=1}^G \prod_{j=1}^J \hat{\tau}_{g'} \mathcal{N}_p(\mathbf{x}_j; \hat{\boldsymbol{\mu}}_{g'}, \hat{\boldsymbol{\Lambda}}_{g'} \hat{\boldsymbol{\Lambda}}_{g'}^\top + \hat{\boldsymbol{\Psi}}_{g'}) \sum_{f=1}^G \prod_{o \in B_2} \hat{\tau}_f \mathcal{N}_p(\mathbf{x}_o; \hat{\boldsymbol{\mu}}_f, \hat{\boldsymbol{\Lambda}}_f \hat{\boldsymbol{\Lambda}}_f^\top + \hat{\boldsymbol{\Psi}}_f)}. \quad (10)$$

Similarly, the computation is repeated for estimating the posterior probabilities of pixels in B_2 belonging to cluster g with pixels in B_1 as negative constraints. The expected value $E[\mathbf{u}_n | \mathbf{x}_n]$ is computed by considering a joint Gaussian distribution between \mathbf{x}_n and \mathbf{u}_n . By Gaussian conditioning formulas,

$$E[\mathbf{u}_n | \mathbf{x}_n] = \hat{\Lambda}_g^\top (\hat{\Lambda}_g \hat{\Lambda}_g^\top + \hat{\Psi}_g)^{-1} (\mathbf{x}_n - \hat{\mu}_g).$$

Also,

$$E[\mathbf{u}_n \mathbf{u}_n^\top | \mathbf{x}_n] = \mathbf{I}_q - \hat{\beta}_g \hat{\Lambda}_g + \hat{\beta}_g (\mathbf{x}_n - \hat{\mu}_g) (\mathbf{x}_n - \hat{\mu}_g)^\top \hat{\beta}_g^\top,$$

$$\text{where } \hat{\beta}_g = \hat{\Lambda}_g^\top (\hat{\Lambda}_g \hat{\Lambda}_g^\top + \hat{\Psi}_g)^{-1}.$$

9.2.2 CM-step:

The expected complete log-likelihood can be rewritten as

$$\begin{aligned} Q_2(\Lambda_g, \Psi_g) = C + \sum_{g=1}^G \mathbf{n}_g \left[-\frac{1}{2} \log |\Psi_g| - \frac{1}{2} [\Psi_g^{-1} \mathbf{S}_g]^\top + [\Psi_g^{-1} \Lambda_g \hat{\beta}_g \mathbf{S}_g]^\top \right. \\ \left. - \frac{1}{2} [\Lambda_g^\top \Psi_g^{-1} \Lambda_g \Theta_g] \right], \end{aligned} \quad (11)$$

where,

$$\begin{aligned} \Theta_g &= \mathbf{I}_q - \hat{\beta}_g \Lambda_g + \hat{\beta}_g \mathbf{S}_g \hat{\beta}_g^\top, \\ \mathbf{S}_g &= \frac{1}{\mathbf{n}_g} \sum_{n=1}^N \hat{z}_{ng}^+ (\mathbf{x}_n - \hat{\mu}_g) (\mathbf{x}_n - \hat{\mu}_g)^\top, \\ \mathbf{n}_g &= \sum_{n=1}^N \hat{z}_{ng}^+. \end{aligned}$$

Differentiating 11 with respect to Λ_g

$$\hat{\Lambda}_g = \mathbf{S}_g \hat{\beta}_g^\top \Theta_g^{-1}.$$

Differentiating 11 with respect to Ψ_g^{-1}

$$\hat{\Psi}_g = \text{diag}[\mathbf{S}_g - \hat{\Lambda}_g \hat{\beta}_g \mathbf{S}_g].$$

The algorithm iteratively updates the parameters until convergence and the posterior probabilities z_{ng} at convergence are used to compute the similarity matrix in the consensus approach.

10 Appendix C

As discussed in Section 4, Figures 13 and 14 highlight the 43.5% of pixels and 24.7% of pixels used as constraints respectively.

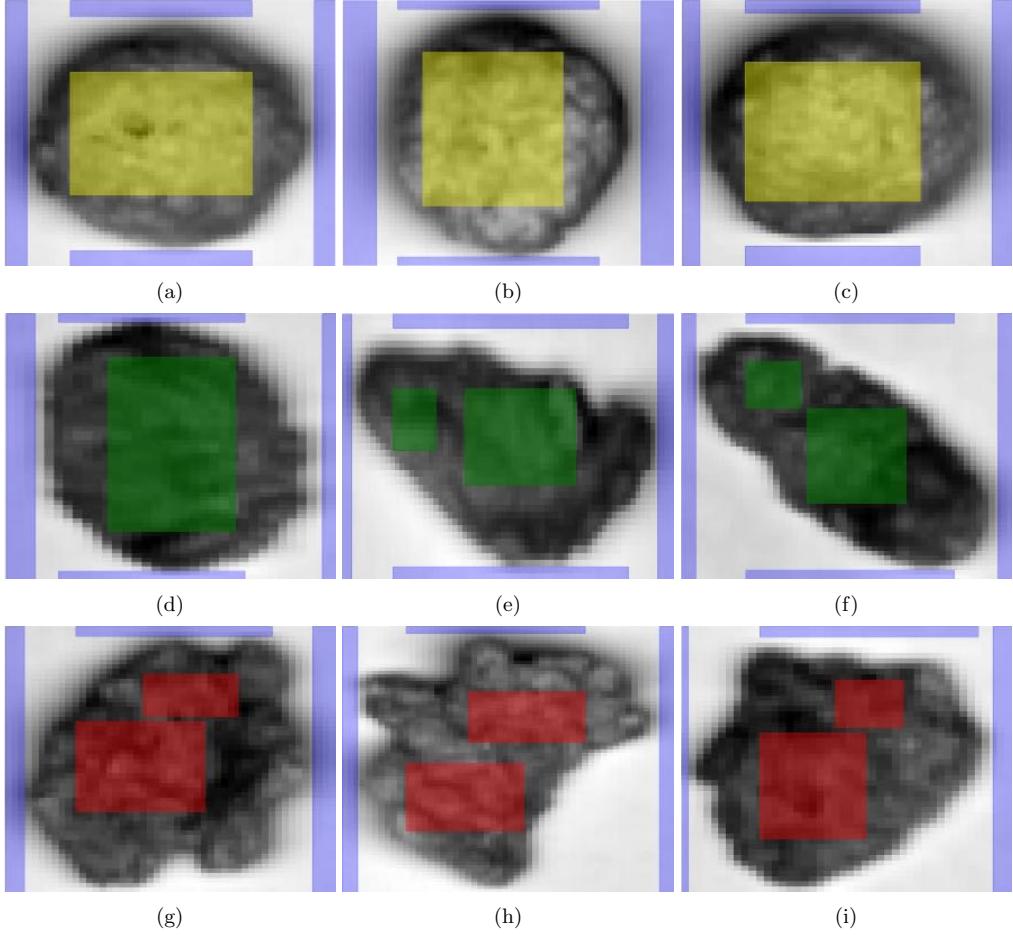


Figure 13: *Blocks of pixels (12,215 pixels of $N = 28,039$ pixels (43.5%)) selected as constraints in three greyscale images for each of the three types of synthetic puffed cereal, wheat (13a, 13b, 13c), corn (13d, 13e, 13f), and rice (13g, 13h, 13i).* The pixels in the blue blocks are background, the pixels in the yellow blocks are wheat, the pixels in the green blocks are corn and the pixels in the red blocks are rice.

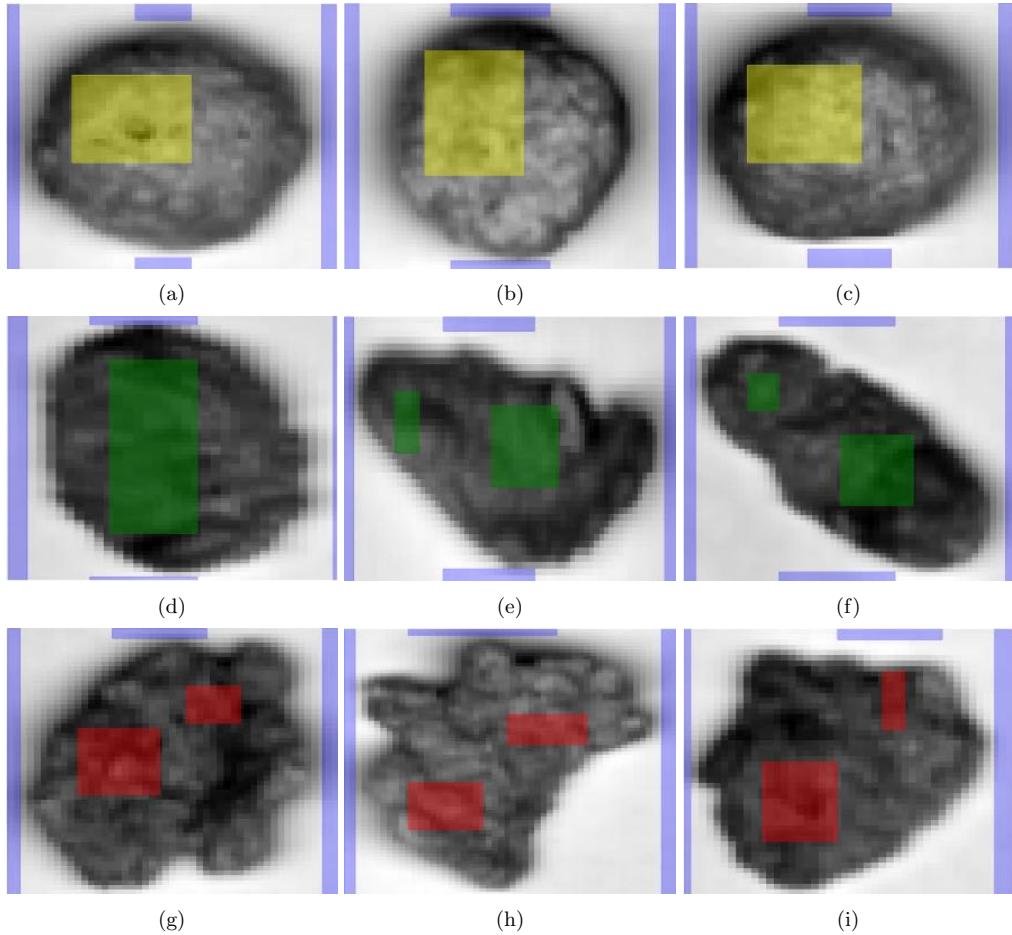


Figure 14: *Blocks of pixels (6,951 pixels of $N = 28,039$ pixels (24.7%)) selected as constraints in three greyscale images for each of the three types of synthetic puffed cereal, wheat (14a, 14b, 14c), corn (14d, 14e, 14f), and rice (14g, 14h, 14i). The pixels in the blue blocks are background, the pixels in the yellow blocks are wheat, the pixels in the green blocks are corn and the pixels in the red blocks are rice.*

11 Appendix D

As discussed in Section 4.4, Table 2 details the cumulative proportion of the variance explained by the first 5 principal components of the motivating hyperspectral image dataset.

Table 2: *Cumulative proportion of variance explained by the principal components of the motivating hyperspectral image dataset on puffed cereals.*

Number of principal components	1	2	3	4	5
Cumulative proportion of variance explained	94.2	99.7	99.92	99.94	99.95

12 Appendix E

As discussed in Section 5, Figures 15, and 16 shows the greyscale images for the remaining six hyperspectral images, and the respective pixel labels as generated using different clustering approaches.

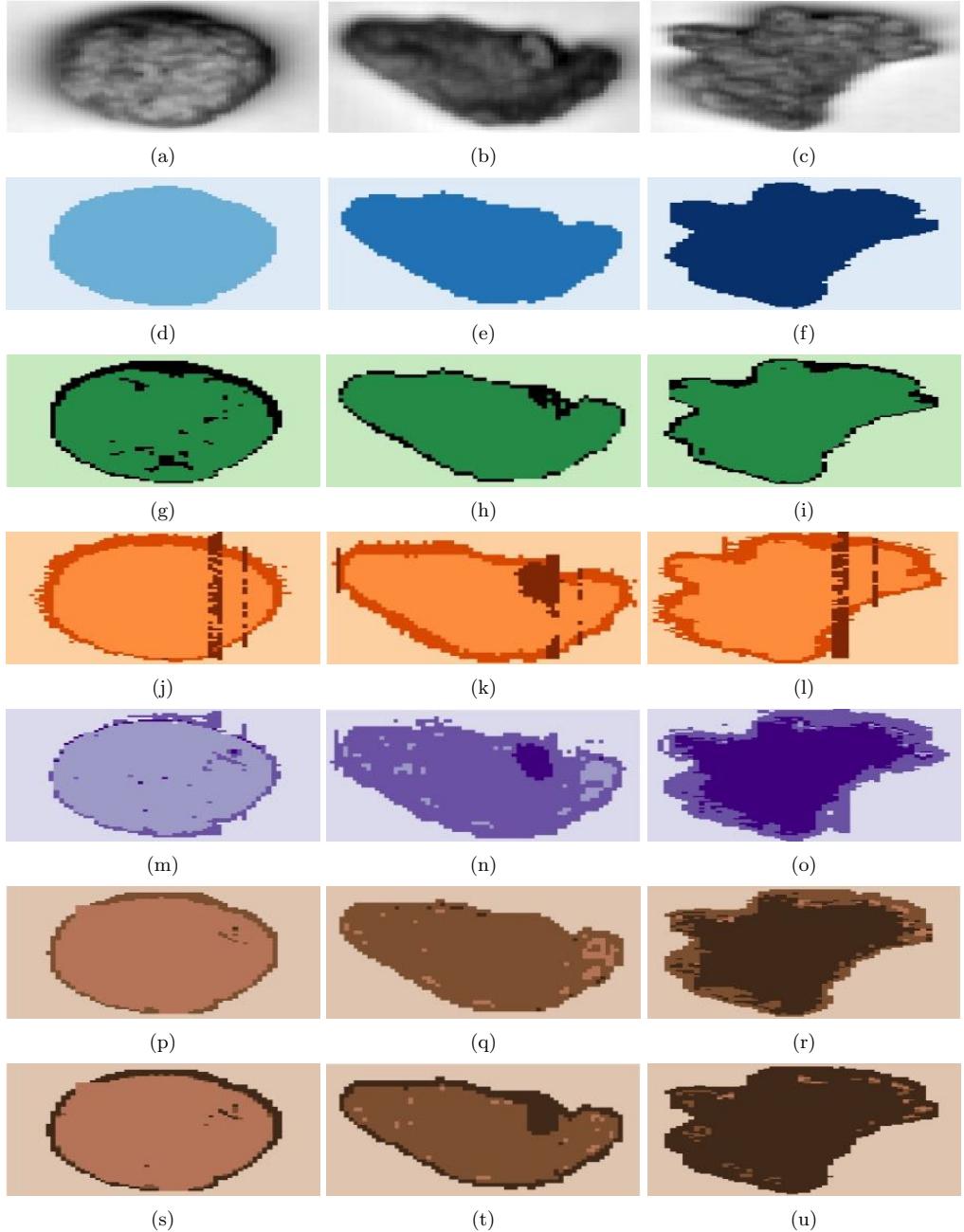


Figure 15: Greyscale images and pixel labels for three of the nine hyperspectral images ($l = \{2, 5, 8\}$) of the cereals using the threshold approach, DBSCAN, MBC, PGMM, ccPGMM with 43.5% of pixels as constraints, and ccPGMM with 24.7% of pixels as constraints respectively for wheat (15a, 15d, 15g, 15j, 15m, 15p, 15s), corn (15b, 15e, 15h, 15k, 15n, 15q, 15t) and rice (15c, 15f, 15i, 15l, 15o, 15r, 15u).

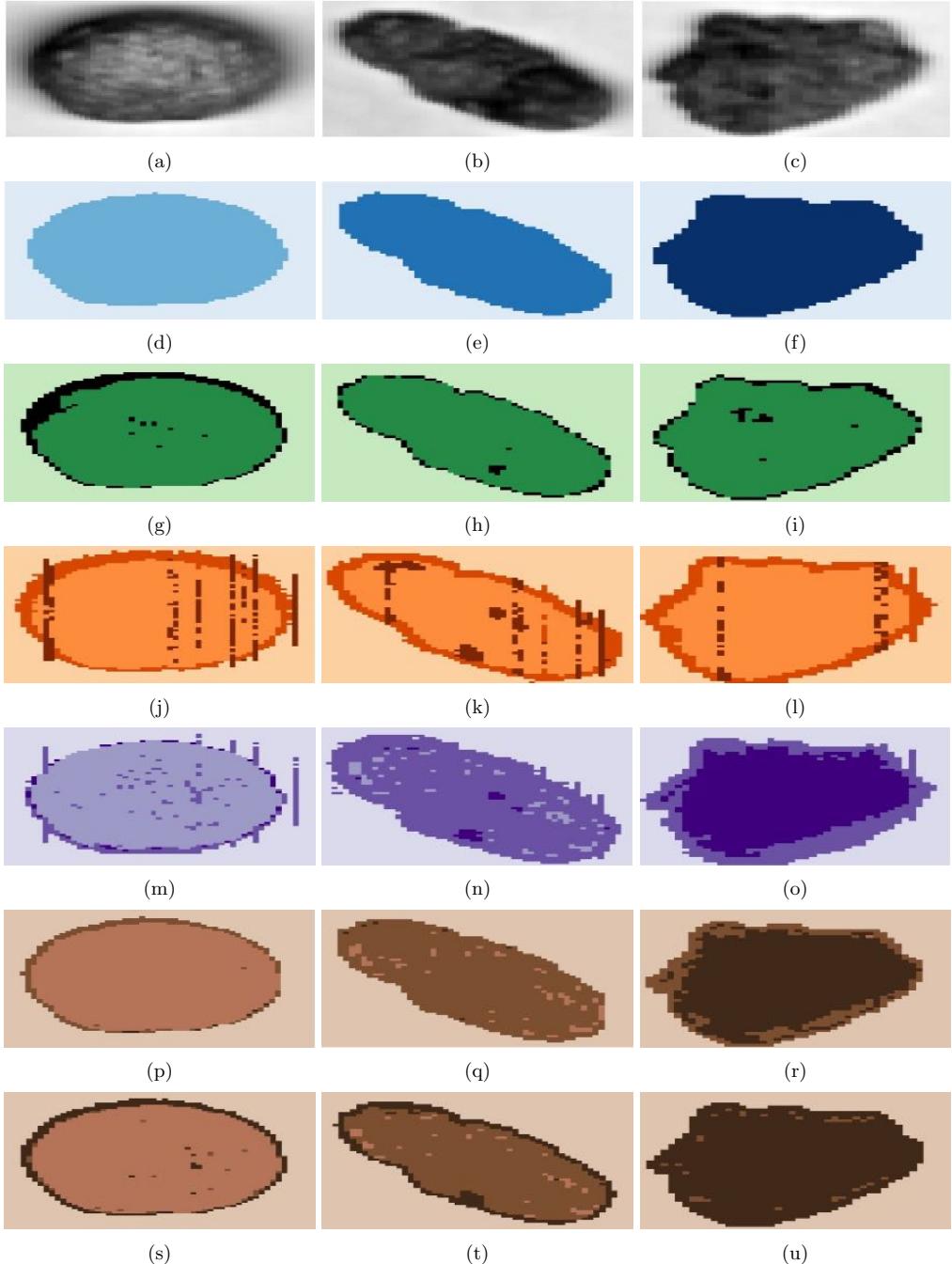


Figure 16: Greyscale images and pixel labels for three of the nine hyperspectral images ($l = \{3, 6, 9\}$) of the cereals using the threshold approach, DBSCAN, MBC, PGMM, ccPGMM with 43.5% pixels as constraints, and ccPGMM with 24.7% pixels as constraints respectively for wheat (16a, 16d, 16g, 16j, 16m, 16p, 16s), corn (16b, 16e, 16h, 16k, 16n, 16q, 16t) and rice (16c, 16f, 16i, 16l, 16o, 16r, 16u).

13 Appendix F

As discussed in Section 5, Figures 17, 18, and 19 illustrates the uncertainty associated with the pixel labels for the nine hyperspectral images of the puffed cereals based on the cluster solutions of MBC,

PGMM, ccPGMM with 43.5% pixels as constraints and ccPGMM with 24.7% pixels as constraints.

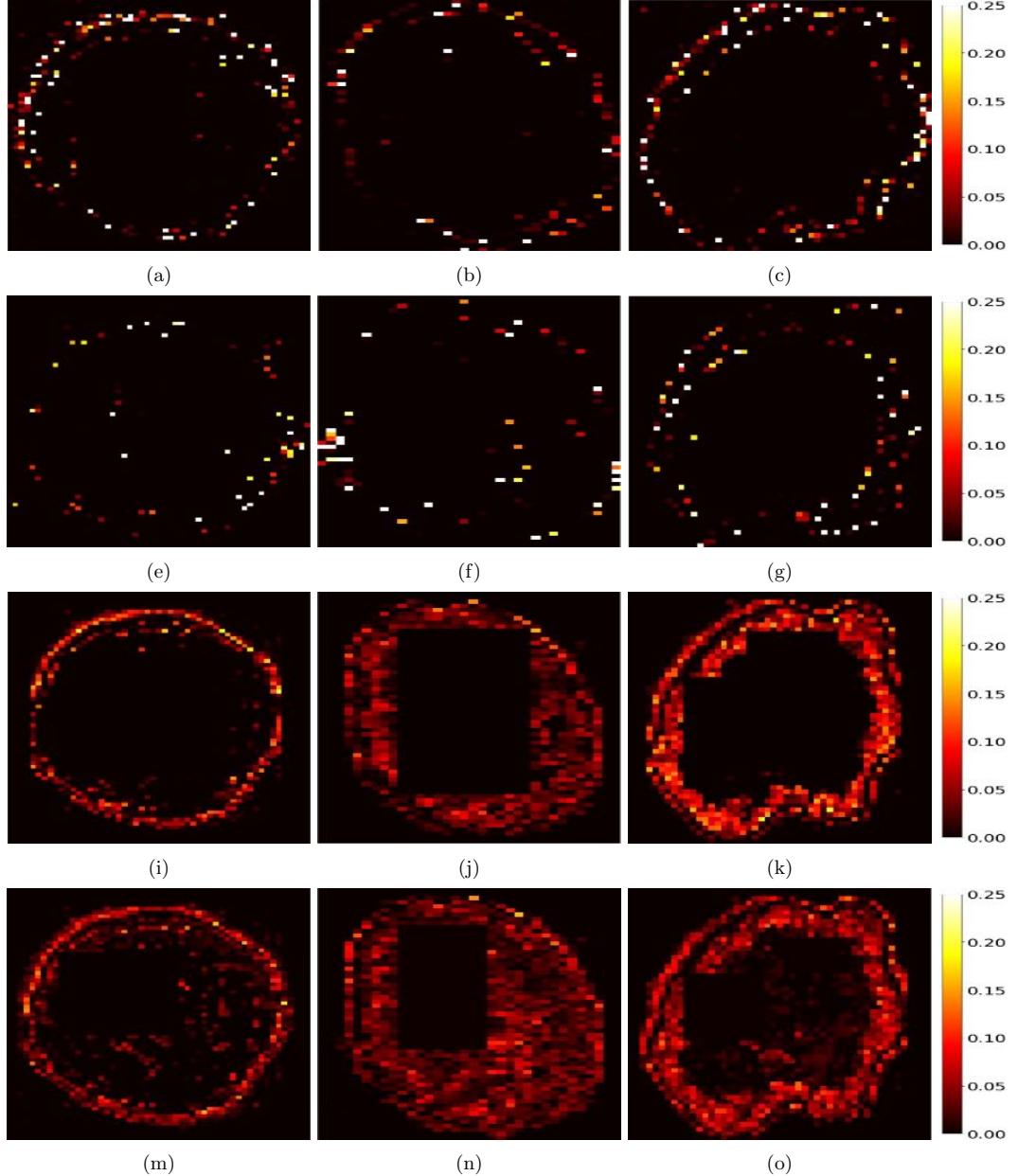


Figure 17: *Uncertainty associated with pixel labels for three of the nine hyperspectral images ($l = \{1, 4, 7\}$) of the puffed cereals based on the cluster solutions of MBC, PGMM, ccPGMM with 43.5% of pixels as constraints, and ccPGMM with 24.7% of pixels as constraints respectively for wheat (17a, 17e, 17i, 17m), corn (17b, 17f, 17j, 17n) and rice (17c, 17g, 17k, 17o).*

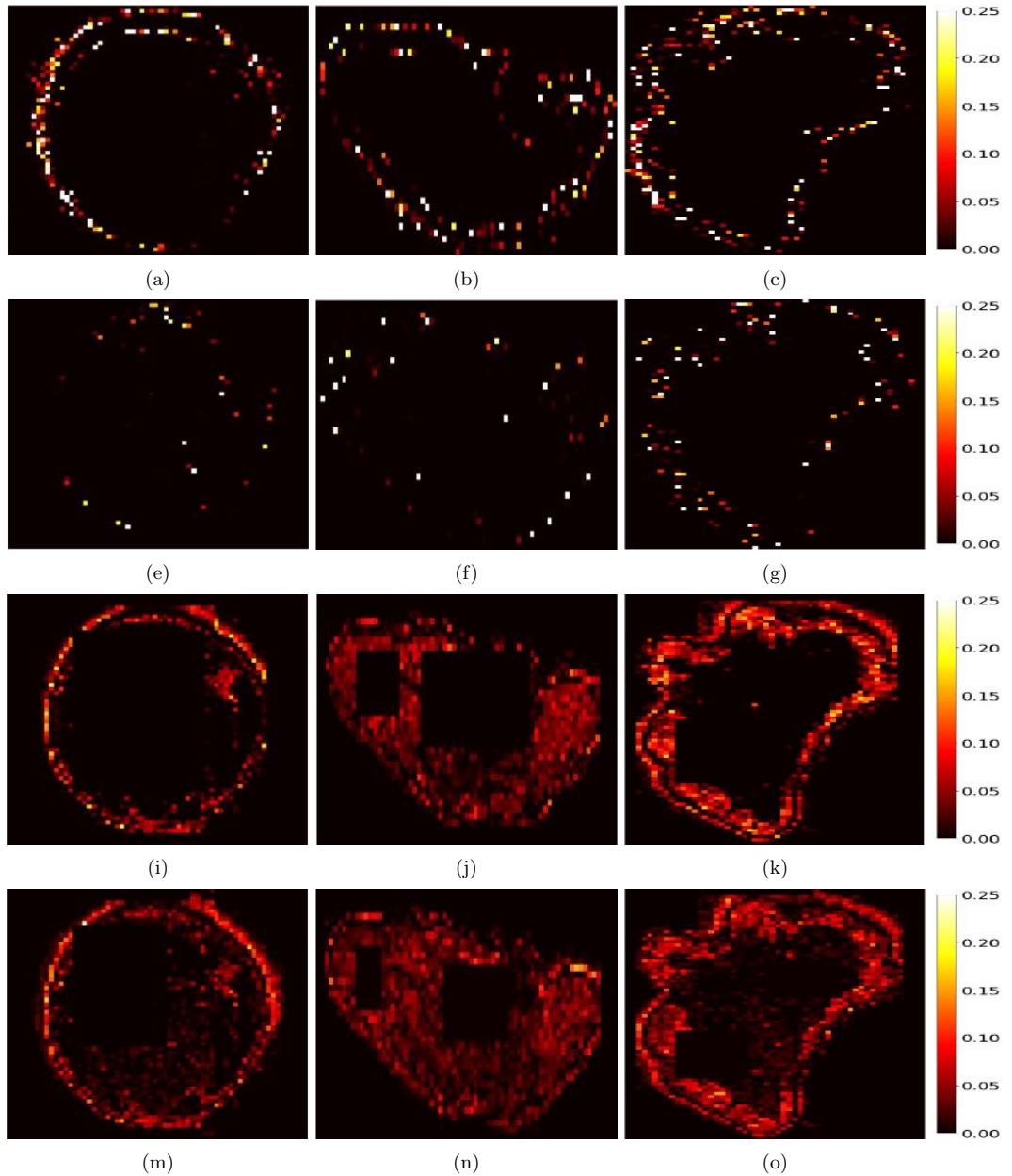


Figure 18: *Uncertainty associated with pixel labels for three of the nine hyperspectral images ($l = \{2, 5, 8\}$) of the cereals based on the cluster solutions of MBC, PGMM, ccPGMM with 43.5% of pixels as constraints, and ccPGMM with 24.7% of pixels as constraints respectively for wheat (18a, 18e, 18i, 18m), corn (18b, 18f, 18j, 18n) and rice (18c, 18g, 18k, 18o).*

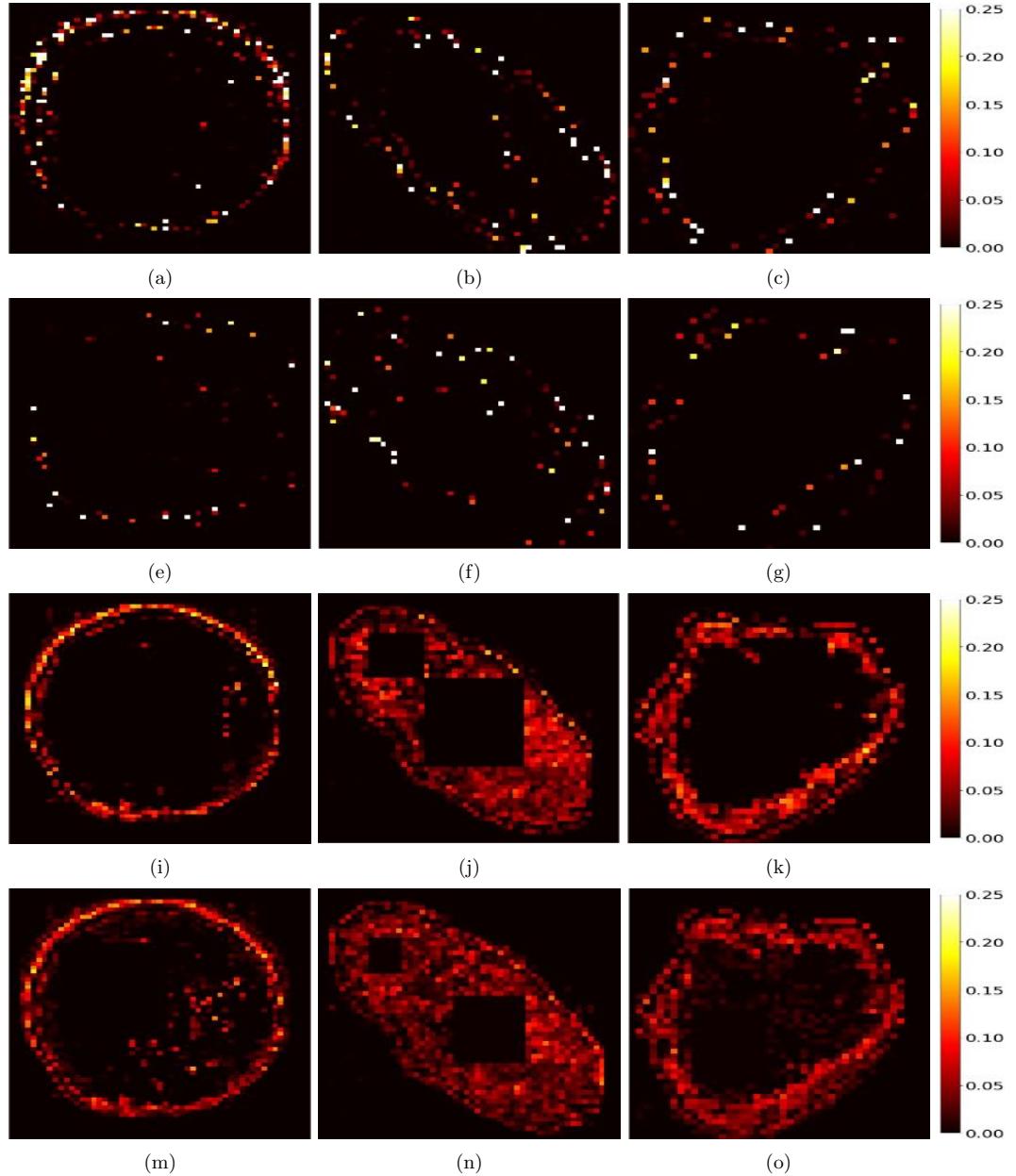


Figure 19: *Uncertainty associated with pixel labels for three of the nine hyperspectral images ($l = \{3, 6, 9\}$) of the cereals based on the cluster solutions of MBC, PGMM, ccPGMM with 43.5% of pixels as constraints, and ccPGMM with 24.7% of pixels as constraints respectively for wheat (19a, 19e, 19i, 19m), corn (19b, 19f, 19j, 19n) and rice (19c, 19g, 19k, 19o).*

Poster session

Andrea Cappozzo (Department of Economics, Management and Quantitative Methods, University of Milan) *Model-based clustering of right-censored lifetime data with frailties and random covariates*

Andrea Mecchini (Università degli Studi di Trieste) *Towards a Transition Probability Matrix Formulation of Advanced Density Peaks*

CJ Clarke (School of Mathematics and Statistics, University College Dublin, Ireland) *A Mixture of Latent Position Cluster Models for Model Based Clustering of Collections of Networks*

Daniel Suen (Department of Statistics, University of Washington) *Mixture of Binomial Product Experts*

Daniele Tancini (University of Perugia, Department of Economics) *Unimodal density-based clustering and merging algorithm using Gaussian mixtures*

Edoardo Redivo (University of Bologna) *Bayesian Inference for Mixtures of Quantile-based Factor Models*

Elena Buscaroli (Department of Mathematics, Informatics and Geosciences, University of Trieste) *Bayesian Modeling for Cancer Subclonal Deconvolution in Multisample and Longitudinal Data*

Francesco Amato (Université Lumière Lyon 2) *Clustering Longitudinal Mixed Data*

Giorgia Zaccaria (University of Milano-Bicocca) *Cellwise outlier detection in model-based clustering*

Giulia Marchello (Centre Inria d'Université Côte d'Azur, Antenne de Montpellier) *Deep dynamic co-clustering of count data streams: application to pharmacovigilance*

Lapo Santi (UCD) *Ranked Stochastic Block Models*

Marco Berrettini (University of Bologna) *Mean-restricted Matrix-variate Normals with an application to clustering*

Marta Nai Ruscone (University of Genoa) *Mixture-based clustering with covariates for ordinal responses*

Martin Metodiev (Université Clermont Auvergne, Laboratoire de Mathématiques Blaise Pascal) *Computationally efficient Bayesian model selection in mixture models from MCMC using the THAMES estimator*

Metodiev (Université Clermont Auvergne) *Computationally efficient Bayesian model selection in mixture models from MCMC using the THAMES estimator*

Noemi Corsini (Dipartimento di Scienze Statistiche - Università di Padova) *A Bayesian overlapping stochastic block model for biographical network*

Pedro Menezes de Araújo (University College Dublin) *Modelling mortality rates with beta latent variable models*

Sara Geremia (Department of Mathematics, Informatics and Geosciences, University of Trieste) *Community Detection in Attributed Networks based on their Density Landscape*

Silvia Dallari (University of Bologna) *Clustering microbiome data via diversity-based mixtures*

Thais Pacheco Menezes (University College Dublin) *Hausdorff Distance: A Powerful Tool for Matching Households and Individuals in Historical Censuses*

Software session

Cristina Tortora (San José State University) *MixtureMissing: An R package for model-based clustering with missing data*

Luca Scrucca (Università degli Studi di Perugia) *gmmclust ... The Shape of Graphs to Come*