Advances in Data Analysis and Classification
https://doi.org/10.1007/s11634-023-00547-5

REGULAR ARTICLE

Model based clustering of multinomial count data

Panagiotis Papastamoulis[1]

Received: 3 August 2022 / Revised: 30 May 2023 / Accepted: 9 June 2023

Abstract

We consider the problem of inferring an unknown number of clusters in multinomial count data, by estimating finite mixtures of multinomial distributions with or without covariates. Both Maximum Likelihood (ML) as well as Bayesian estimation are taken into account. Under a Maximum Likelihood approach, we provide an Expectation–Maximization (EM) algorithm which exploits a careful initialization procedure combined with a ridge-stabilized implementation of the Newton-Raphson method in the M-step. Under a Bayesian setup, a stochastic gradient Markov chain Monte Carlo (MCMC) algorithm embedded within a prior parallel tempering scheme is devised. The number of clusters is selected according to the Integrated Completed Likelihood criterion in the ML approach and estimating the number of non-empty components in overfitting mixture models in the Bayesian case. Our method is illustrated in simulated data and applied to two real datasets. The proposed methods are implemented in a contributed `R` package, available online.

Keywords Mixture model · Multinomial logistic regression · Count data · Clustering

Mathematics Subject Classification 62H30 · 62F15 · 62-08

1 Introduction

Multinomial count data arise in various applications (see e.g. Yu and Shaw (2014), Nowicka and Robinson (2016)) and clustering them is a task of particular interest (Jorgensen 2004; Govaert and Nadif 2007; Portela 2008; Bouguila 2008; Zamzami and Bouguila 2020; Chen et al. 2020). Finite mixture models (McLachlan and Peel 2004; Marin et al. 2005; Frühwirth-Schnatter 2006; Frühwirth-Schnatter et al. 2019) are widely used for clustering heterogeneous datasets. Their applicability is

✉ Panagiotis Papastamoulis
papastamoulis@aueb.gr

[1] Department of Statistics, Athens University of Economics and Business, 76, Patission Str., 10434 Athens, Greece

Published online: 05 July 2023

extended beyond the model-based clustering framework, by also providing a means for semiparametric inference, see e.g. Morel and Nagaraj (1993), where mixtures of multinomial distributions model extra multinomial variation in count data.

In many instances, the resulting inference can be improved by taking into account the presence of covariates, when available. Naturally, the framework of mixtures of multinomial logistic regressions (Grün and Leisch 2008b) can be used for dealing with such data, under a model-based clustering point of view. These models belong to the broader family of mixtures of generalized linear models (Leisch 2004; Grün and Leisch 2007, 2008a), which are estimated either under a maximum likelihood approach via the EM algorithm (Dempster et al. 1977), or in a Bayesian fashion using MCMC sampling (Albert and Chib 1993; Hurn et al. 2003).

Various latent class models are based on mixtures of multinomial distributions. Durante et al. (2019) cluster multivariate categorical data by estimating mixtures of products of multinomial distributions, under the presence of covariates in the mixing proportions. Galindo Garre and Vermunt (2006) estimate latent class models using Bayesian Maximum A Posteriori estimation and illustrate via simulations that the Bayesian approach is more accurate than maximum likelihood estimation. More general latent class models based on multinomial distributions include hidden Markov models (Zuanetti and Milan 2017) and Markov random fields (Li et al. 2011).

In this paper, our goal is to cluster multinomial count data using finite mixtures of multinomial logistic regression models. Before proceeding we introduce some notation. Let $\boldsymbol{Y} = (Y_1, \ldots, Y_J; Y_{J+1})^\top$ denote a random vector distributed according to a multinomial distribution

$$\boldsymbol{Y} \sim \mathcal{M}_{J+1}(S, \boldsymbol{\theta}).$$

$S \in \mathbb{Z}_+$ corresponds to the number of independent replicates of the multinomial experiment, while the vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_J; \theta_{J+1})$, with $0 < \theta_j < 1$ and $\sum_{j=1}^{J+1} \theta_j = 1$ contains the probabilities of observing each category.

Under the presence of K heterogeneous sub-populations in the multinomial experiment, we typically model the outcome using a finite mixture model as follows. Let $\boldsymbol{Z} = (Z_1, \ldots, Z_K)^\top \sim \mathcal{M}_K(1, \boldsymbol{\pi})$ denote a latent multinomial random variable with K categories, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{K-1}; \pi_K)$ is such that $0 < \pi_k < 1$ and $\sum_{k=1}^{K} \pi_k = 1$. Conditional on $Z_k = 1$ we assume that

$$\boldsymbol{Y} | Z_k = 1 \sim \mathcal{M}_{J+1}(S, \boldsymbol{\theta}_k)$$

where $\boldsymbol{\theta}_k = (\theta_{k1}, \ldots, \theta_{kJ}; \theta_{k(J+1)})$, with $0 < \theta_{kj} < 1$ and $\sum_{j=1}^{J+1} \theta_{kj} = 1$ contains the probabilities of observing each category for the corresponding multinomial experiment. It follows that $\boldsymbol{Y}$ is drawn from a finite mixture of K multinomial distributions, so the probability mass function of $\boldsymbol{Y}$ can be written as

$$\sum_{k=1}^{K} \pi_k f(\boldsymbol{y} | \boldsymbol{\theta}_k). \tag{1}$$

The weights $\pi_1, \ldots, \pi_K$ correspond to the mixing proportions. Finally, $f(\cdot|\boldsymbol{\theta}_k)$ denotes the probability mass function of the $(J+1)$-dimensional multinomial distribution $\mathcal{M}_{J+1}(S, \boldsymbol{\theta})$, that is,

$$f(\boldsymbol{y}|\boldsymbol{\theta}_k) = \frac{S!}{\prod_{j=1}^{J+1} y_j!} \prod_{j=1}^{J+1} y_j^{\theta_{kj}} \mathrm{I}_{\mathcal{Y}_{S,J}}(\boldsymbol{y}), \tag{2}$$

where

$$\mathcal{Y}_{S,J} = \left\{ y_1, \ldots, y_J \in \mathbb{Z}_+ : 0 \le \sum_{j \le J} y_j \le S;\, y_{J+1} := S - \sum_{j \le J} y_j \right\}$$

and $S \in \mathbb{Z}_+$. A necessary and sufficient condition for the generic identifiability of finite mixtures of multinomial distributions is the restriction $S \ge 2K - 1$ (Teicher 1963; Blischke 1964; Titterington et al. 1985; Grün and Leisch 2008b).

Given a vector of P covariates $\boldsymbol{x} = (x_1, \ldots, x_P)$ and assuming that category $J+1$ is the baseline (in general, this can be any of the $J+1$ categories), we express the log-odds as

$$\text{logit}\theta_j = \log \frac{\theta_j}{\theta_{J+1}} = \boldsymbol{\beta}_j^\top \boldsymbol{x}, \tag{3}$$

for $j = 1, \ldots, J$. The vector $\boldsymbol{\beta}_j = (\beta_{j1}, \ldots, \beta_{jP})^\top \in \mathbb{R}^P$ contains the regression coefficients for category j. It follows from (3) that

$$\theta_j = \frac{\exp\{\boldsymbol{\beta}_j^\top \boldsymbol{x}\}}{1 + \sum_{\ell \le J} \exp\{\boldsymbol{\beta}_\ell^\top \boldsymbol{x}\}}, \tag{4}$$

for $j = 1, \ldots, J$.

Extending the previous model to the case of K latent groups, Equation (3) becomes

$$\text{logit}\theta_{kj} = \boldsymbol{\beta}_{kj}^\top \boldsymbol{x}, \tag{5}$$

for category $j = 1, \ldots, J$ and group-specific parameters $\boldsymbol{\beta}_{kj} = (\beta_{kj1}, \ldots, \beta_{kjP})^\top$, $k = 1, \ldots, K$. In analogy to (4), define

$$\theta_{kj} = \begin{cases} \dfrac{\exp\{\boldsymbol{\beta}_{kj}^\top \boldsymbol{x}\}}{1 + \sum_{\ell \le J} \exp\{\boldsymbol{\beta}_{k\ell}^\top \boldsymbol{x}\}}, & j \le J \\ \dfrac{1}{1 + \sum_{\ell \le J} \exp\{\boldsymbol{\beta}_{k\ell}^\top \boldsymbol{x}\}}, & j = J + 1 \end{cases} \tag{6}$$

for $k = 1, \ldots, K$.

We assume that we observe n independent pairs $(\boldsymbol{y}_i, \boldsymbol{x}_i)$; $i = 1, \ldots, n$, where the joint-probability mass function of $\boldsymbol{Y} = (\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n)^\top$ given $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ is written as

$$\begin{aligned} f(\boldsymbol{y}|\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{x}) &= \prod_{i=1}^{n} f(\boldsymbol{y}_i|\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{x}_i) \\ &= \prod_{i=1}^{n}\sum_{k=1}^{K}\pi_k \frac{S_i!}{\prod_{j=1}^{J+1} y_{ij}!}\prod_{j=1}^{J+1} y_{ij}^{g_{ikj}} \mathrm{I}_{\mathcal{Y}_{S_i,J}}(\boldsymbol{y}_i). \end{aligned} \tag{7}$$

where

$$g_{ikj} = \begin{cases} \dfrac{\exp\{\boldsymbol{\beta}_{kj}^\top \boldsymbol{x}_i\}}{1 + \sum_{\ell \le J} \exp\{\boldsymbol{\beta}_{k\ell}^\top \boldsymbol{x}_i\}}, & j \le J \\ \dfrac{1}{1 + \sum_{\ell \le J} \exp\{\boldsymbol{\beta}_{k\ell}^\top \boldsymbol{x}_i\}}, & j = J + 1 \end{cases} \tag{8}$$

for $i = 1, \ldots, n$; $k = 1, \ldots, K$. In practice, S_i is derived given $\boldsymbol{y}_i$, $i = 1, \ldots, n$.

The R package `mixtools` (Benaglia et al. 2009) can be used to estimate mixtures of multinomial distributions (among numerous other functionalities), under a maximum likelihood approach using the EM algorithm. However, the usage of covariates is not considered. On the other hand, the `flexmix` package (Leisch 2004; Grün and Leisch 2007, 2008a) can estimate mixtures of multinomial logistic regression models using the `FLXMRmultinom()` function, which also implements the EM algorithm. The package allows the user to run the EM algorithm repeatedly for different numbers of components and returns the maximum likelihood solution for each. However, alternative –and perhaps more efficient– initialization schemes are not considered. Finally, a fully Bayesian implementation is not available in both packages.

The contribution of the present study it to offer an integrated approach to the problem of clustering multinomial count data using mixtures of multinomial logit models. For this purpose we use frequentist as well as Bayesian methods. Both the EM algorithm (for the frequentist approach) as well as the MCMC sampler (for the Bayesian approach) are carefully implemented in order to deal with various computational and inferential challenges imposed by the complex nature of mixture likelihoods/posterior distributions (Celeux et al. 2000). At first, it is well known that the EM algorithm may converge to local modes of the likelihood surface. We tackle this problem by extending the initialization of the EM algorithm for mixture of Poisson regression models as suggested in Papastamoulis et al. (2016). Second, we implement a ridge-stabilized version of the Newton-Raphson algorithm in the M-step. This adjustment is based on a quadratic approximation of the function of interest on a suitably chosen spherical region and effectively avoids many of the pitfalls of standard Newton-Raphson iterations (Goldfeld et al. 1966). In the presented applications and simulation studies, our interest lies in cases where the multinomial data consists of a large number of replications for each multinomial observation. When the number of replicates is small, identifiability of the model is not guaranteed (see Grün and Leisch (2008b)).

Under a Bayesian approach, traditional Bayesian methods estimate the number of clusters using the reversible jump MCMC (Green 1995; Richardson and Green 1997) or the birth-death MCMC technique (Stephens 2000). In multivariate settings, however, the practical application of these methods is limited. More recently, alternative Bayesian methods for estimating the number of clusters focus on the use of overfitting mixture models (Rousseau and Mengersen 2011), information theoretic techniques which allow to post-process MCMC samples of partitions to summarize the posterior in Bayesian clustering models (Wade and Ghahramani 2018), and generalized mixtures of finite mixtures (Frühwirth-Schnatter et al. 2021). Our Bayesian model combines recent advances on overfitting mixture models (Rousseau and Mengersen 2011; van Havre et al. 2015; Papastamoulis 2020) with stochastic gradient MCMC sampling (Roberts and Tweedie 1996; Nemeth and Fearnhead 2021) and running parallel MCMC chains which can exchange states. Moreover, we efficiently deal with the label switching problem, using the Equivalence Classes Representatives (ECR) algorithm (Papastamoulis and Iliopoulos 2010). In such a way, the returned MCMC output is directly interpretable and provides various summaries of marginal posterior distributions such as posterior means and Bayesian credible intervals of the parameters of interest.

The combination of Maximum Likelihood and Bayesian estimation provides additional insights: it is demonstrated that the best-performing approach is to initialize the MCMC algorithm using information from the solution obtained in the EM implementation. Therefore, our proposed method provides a powerful and practical approach that allows to easily estimate the unknown number of clusters and related parameters in multinomial count datasets.

The rest of the paper is organized as follows. Maximum likelihood estimation of finite mixtures of multinomial distributions with or without covariates via the EM algorithm is described in Sect. 2. The careful treatment of the M-step is extensively described in Sect. 2.2. Section 2.3 discusses initialization issues in the EM implementation. Section 2.4 describes the selection of the number of clusters under the EM algorithm. The Bayesian formulation is described in Sect. 3. The proposed MCMC sampler is introduced in Sect. 3.1. Section 3.2 describes the estimation of the number of clusters using overfitting mixture models as well as how we deal with the label switching problem. Applications are illustrated in Sect. 4. The paper concludes with a Discussion in Sect. 5. An Appendix contains further implementation details, additional simulation results and comparisons with alternative approaches (`flexmix`).

2 Maximum Likelihood estimation via the EM algorithm

In this section we describe the Expectation–Maximization (EM) algorithm (Dempster et al. 1977) for estimating mixtures of multinomial logistic regressions. For the case of covariates, the complete log-likelihood is written as

$$\log f(\boldsymbol{y}|\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{z}) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\left\{\log \pi_k + \log c_i + \sum_{j=1}^{J+1} y_{ij}\log g_{ikj}\right\}, \tag{9}$$

where $c_i = S_i!/\prod_{j=1}^{J+1} y_{ij}!$.

The EM algorithm proceeds by computing the expectation of the complete log-likelihood (see Sect. 2.1 with respect to the latent allocation variables $\boldsymbol{Z}$ (given $\boldsymbol{y}$ and $\boldsymbol{x}$). Then, the expected complete log-likelihood is maximized with respect to the parameters $\boldsymbol{\pi}$, $\boldsymbol{\beta}$ (see Sect. 2.2), given the current expected values of missing data. In the case of mixtures of multinomial logistic regressions this task can become quite challenging, since typical numerical implementations (such as the standard Newton-Raphson algorithm) may fail. For this reason, it is crucial to apply more robust numerical implementations (Goldfeld et al. 1966) as discussed in Sect. 2.2. In Sect. 2.3 special attention is given to the important issue of initialization of the EM algorithm. Section 2.4 describes the selection of the number of clusters under the EM algorithm.

2.1 Expectation step

The expectation step (E-step) consists of evaluating the expected complete log-likelihood, with respect to the conditional distribution of $\boldsymbol{Z}$ given the observed data $\boldsymbol{y}$ (and $\boldsymbol{x}$ in the covariates case), as well as a current estimate of the parameters $(\boldsymbol{\pi}^{(t)}, \boldsymbol{\beta}^{(t)})$. Define the posterior membership probabilities w_{ik} as

$$w_{ik} = \mathrm{P}(Z_{ik} = 1|\boldsymbol{y}_i, \boldsymbol{x}_i, \boldsymbol{\pi}, \boldsymbol{\beta}) = \frac{\pi_k f(\boldsymbol{y}_i|\boldsymbol{g}_{ik})}{\sum_{\ell=1}^{K} \pi_\ell f(\boldsymbol{y}_i|\boldsymbol{g}_{i\ell})}, \quad i = 1, \ldots, n; k = 1, \ldots, K.$$

Note that, according to the Maximum A Posteriori rule, the estimated clusters are obtained as

$$c_i = \mathrm{argmax}_{k\in\{1,\ldots,K\}}\{w_{ik}; k = 1, \ldots, K\}, \quad i = 1, \ldots, n.$$

The expected complete log-likelihood is equal to

$$\begin{aligned} Q(\boldsymbol{\pi}, \boldsymbol{\beta}|\boldsymbol{\pi}^{(t)}, \boldsymbol{\beta}^{(t)}) &:= \mathrm{E}_{\boldsymbol{Z}|\boldsymbol{y},\boldsymbol{x},\boldsymbol{\pi}^{(t)},\boldsymbol{\beta}^{(t)}}\{\log f(\boldsymbol{y}|\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{Z})\} \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} w_{ik}\left\{\log \pi_k + \log c_i + \sum_{j=1}^{J+1} y_{ij} \log g_{ikj}\right\} \end{aligned} \tag{10}$$

where the current parameter values $(\boldsymbol{\pi}^{(t)}, \boldsymbol{\beta}^{(t)})$ are used to compute w_{ik}.

2.2 Maximization step

In the maximization step (M-step), (10) is maximized with respect to the parameters $\boldsymbol{\pi}$, $\boldsymbol{\beta}$, that is,

$$\left(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\beta}^{(t+1)}\right) = \underset{\boldsymbol{\pi},\boldsymbol{\beta}}{\mathrm{argmax}}\ Q(\boldsymbol{\pi}, \boldsymbol{\beta}|\boldsymbol{\pi}^{(t)}, \boldsymbol{\beta}^{(t)})$$

The maximization of the expected complete log-likelihood with respect to the mixing proportions leads to

$$\pi_k^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} w_{ik}, \quad k = 1, \ldots, K.$$

The maximization with respect to $\boldsymbol{\beta}$ is analytically tractable only when $P = 1$ (that is, a model with just a constant term). Recall that when no covariates are present then the model is reparameterized with respect to the multinomial probabilities, that is,

$$\theta_{kj} = \frac{e^{\beta_{kj}}}{1 + e^{\beta_{kj}}}.$$

The expected complete log-likelihood is maximized with respect to θ_{kj}. The analytical solution of the M-step in this case is

$$\theta_{kj}^{(t+1)} = \frac{\sum_{i=1}^{n} w_{ik} y_{ij}}{\sum_{i=1}^{n} w_{ik} S_i}, \tag{11}$$

for $k = 1, \ldots, K$ and $j = 1, \ldots, J + 1$.

In case where $P \geq 2$ numerical methods are implemented. We have used two optimization techniques: the typical Newton-Raphson algorithm, as well as a ridge-stabilized version introduced by Goldfeld et al. (1966). It is easy to show that the partial derivative of (10) with respect to β_{kjp} is

$$\frac{\partial Q}{\partial \beta_{kjp}} = \sum_{i=1}^{n} w_{ik}\big\{y_{ij} - S_i g_{ikj}\big\} x_{ip}, \tag{12}$$

$k = 1, \ldots, K$, $j = 1, \ldots, J$ and $p = 1, \ldots, P$. Thus, the gradient vector can be expressed as

$$\nabla Q(\boldsymbol{\beta}) := \left(\sum_{i=1}^{n} w_{i1}\{\boldsymbol{y}_i - S_i \boldsymbol{g}_{i1}\} \otimes \boldsymbol{x}_i, \ldots, \sum_{i=1}^{n} w_{iK}\{\boldsymbol{y}_i - S_i \boldsymbol{g}_{iK}\} \otimes \boldsymbol{x}_i, \right)^{\top}, \tag{13}$$

where $\otimes$ denotes the Kronecker product and we have also defined $\boldsymbol{g}_{ik} := (g_{ik1}, \ldots, g_{ikJ})^{\top}$, $k = 1, \ldots, K$.

The second partial derivative of the log-likelihood function (10) with respect to β_{kjp} and $\beta_{k'j'p'}$ is

$$\frac{\partial^2 Q}{\partial \beta_{kjp} \partial \beta_{k'j'p'}} = -\delta_{kk'} \sum_{i=1}^{n} S_i w_{ik} x_{ip} x_{ip'} g_{ikj} (\delta_{jj'} - g_{ikj'}),$$

where δ_{ij} denotes the Kronecker delta, for $k, k' = 1, \ldots, K$, $j, j' = 1, \ldots, J$ and $p, p' = 1, \ldots, P$. Note that the corresponding Hessian

$$H(\boldsymbol{\beta}) = \begin{pmatrix} H_1(\boldsymbol{\beta}_1) & 0 & \dots & 0 \\ 0 & H_2(\boldsymbol{\beta}_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & H_K(\boldsymbol{\beta}_K) \end{pmatrix}$$

is a block diagonal matrix consisting of K blocks H_k, where each one of them being a $JP \times JP$-dimensional matrix, with

$$H_k = \left\{ \frac{\partial^2 Q}{\partial \beta_{kjp} \partial \beta_{kj'p'}} \right\}_{j=1,\ldots,J;\, p=1,\ldots,P}.$$

This is particularly useful because the inverse of this $KJP \times KJP$-dimensional matrix is the corresponding block diagonal matrix of the inverse matrices, that is,

$$H^{-1}(\boldsymbol{\beta}) = \begin{pmatrix} H_1^{-1}(\boldsymbol{\beta}_1) & 0 & \dots & 0 \\ 0 & H_2^{-1}(\boldsymbol{\beta}_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & H_K^{-1}(\boldsymbol{\beta}_K) \end{pmatrix}.$$

Consequently, the Newton-Raphson update can be performed independently for each $k = 1, \ldots, K$, as described in the sequel.

In order to maximize the expected complete log-likelihood with respect to $\boldsymbol{\beta}$ we used a ridge-stabilized version (Goldfeld et al. 1966) of the Newton-Raphson algorithm. Denote by $\boldsymbol{\beta}^{(t,1)}$ the initial value of $\boldsymbol{\beta}$ at the M-step of iteration t of the EM algorithm. Then, the typical Newton-Raphson update at the $m + 1$-th iteration takes the form

$$\boldsymbol{\beta}^{(t,m+1)} = \boldsymbol{\beta}^{(t,m)} - H^{-1}(\boldsymbol{\beta}^{(t,m)}) \nabla Q(\boldsymbol{\beta}^{(t,m)}), \quad m = 1, 2, \ldots. \tag{14}$$

Let M denote the last iteration of the sequence of Newton-Raphson updates. The updated value of $\boldsymbol{\beta}$ for iteration t of the EM algorithm is then equal to

$$\boldsymbol{\beta}^{(t)} := \boldsymbol{\beta}^{(t,M)}.$$

In case that a second-order Taylor expansion is a good approximation of the underlying function around a maximum, the Newton-Raphson method will converge rapidly (Crockett and Chernoff 1955). However, in general settings, it may happen that the step of the basic update in Eq. (14) will be too large, or $-H$ will be negative definite, in which case the quadratic approximation has no validity.

The following technique addresses these issues by maximizing a quadratic approximation to the function on a suitably chosen spherical region. The algorithm of Goldfeld et al. (1966) is based on the updates

$$\boldsymbol{\beta}^{(t,m+1)} = \boldsymbol{\beta}^{(t,m)} - H_\alpha^{-1}(\boldsymbol{\beta}^{(t,m)}) \nabla Q(\boldsymbol{\beta}^{(t,m)}), \quad m = 1, 2, \ldots, \tag{15}$$

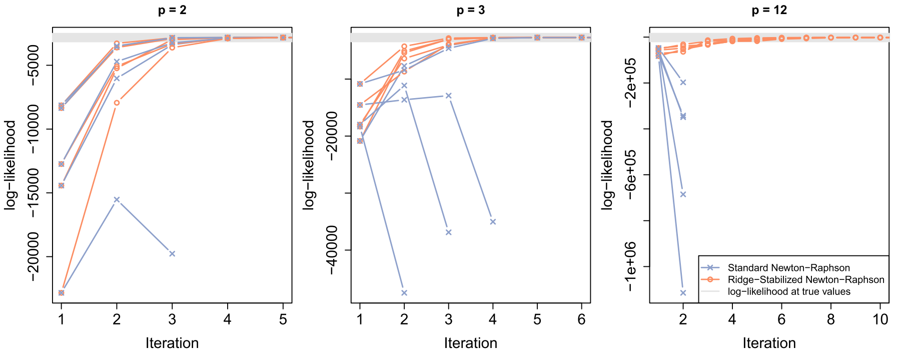


Fig. 1 Estimation of a typical ($K = 1$) multinomial logit model with $D = 6$ categories and p covariates (including constant term): Log-likelihood values per iteration of the standard Newton-Raphson algorithm and the ridge-stabilized version, based on 5 random starting values

where,

$$\alpha = \lambda_1 + R||\nabla Q(\boldsymbol{\beta}^{t-1})|| \tag{16}$$

$$H_\alpha(\boldsymbol{\beta}) = \begin{cases} H(\boldsymbol{\beta}) - \alpha I, & \text{if } \alpha > 0 \\ H(\boldsymbol{\beta}), & \text{if } \alpha \leq 0, \end{cases} \tag{17}$$

while λ_1 and $||x||$ denote the largest eigenvalue of H and the length of vector x, respectively. The parameter R controls the step size of the update: smaller values result to larger step sizes. This parameter is adjusted according to the procedure described Goldfeld et al. (1966): the step size tends to increase when the quadratic approximation appears to be satisfactory.

Figure 1 illustrates the two algorithms using simulated data of $n = 250$ observations from a typical ($K = 1$) multinomial logit model with $D = 6$ categories and varying number of explanatory variables p. In each case, the same random starting value was used for both the standard Newton-Raphson as well as the modified version. Observe that, especially as the number of parameters increases, the standard Newton-Raphson updates may decrease the log-likelihood function. On the other hand, the ridge stabilized version produces a sequence of updates which converge to the mode of the log-likelihood function (as indicated by the gray line).

2.3 EM initialization

Careful selection of initial values for the EM algorithm is crucial (Biernacki et al. 2003; Karlis and Xekalaki 2003; Baudry and Celeux 2015; Papastamoulis et al. 2016) in order to avoid convergence to minor modes of the likelihood surface. Following Papastamoulis et al. (2016), a small-EM (Biernacki et al. 2003) procedure is used. A small-EM initialization refers to the strategy of starting the main EM algorithm from values arising by a series of short runs of the EM. Each run consists of a small number of iterations (say 5 or 10), under different starting values. The selected values that will be used to initialize the main EM algorithm are the ones that correspond to the largest log-likelihood across all small-EM runs. The starting values of each small-EM are selected according to three alternative strategies, namely: *random*, *split* and *shake* small-EM schemes, described in detail in the sequel.

In what follows, we will use the notation

$$\widehat{w}_{ik}^{(K)} = \mathrm{P}(Z_{ik} = 1 | \boldsymbol{y}, \boldsymbol{x}, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\beta}}), \quad k = 1, \ldots, K; i = 1, \ldots, n. \tag{18}$$

in order to explicitly refer to the estimated membership probabilities arising from a mixture model with K components, where $\widehat{\boldsymbol{\pi}}$ and $\widehat{\boldsymbol{\beta}}$ are the parameter estimates obtained at the last iteration of the EM algorithm.

Random small-EM

This strategy corresponds to the random selection of M_{random} starting values and running the EM for a small number (say $T = 5$ or 10) iterations. The parameters of the run which results to the largest log-likelihood value in the last (T-th) iteration are used to initialize the main EM algorithm. The random selection can refer to either choosing random values for the coefficients of the multinomial logit model or for the posterior membership probabilities. The latter scheme is followed in our approach, in particular each row of the $n \times K$ matrix of posterior probabilities is generated according to the $\mathcal{U}(0, 1)$ distribution. Each row is then normalized according to the unity sum constraint.

Split small-EM

Fraley et al. (2005); Papastamoulis et al. (2016) proposed to begin the EM algorithm from a model that underestimates the number of clusters and consecutively adding one component using a splitting procedure among the previously estimated clusters. In our setup, this procedure begins with estimating the one-component ($K = 1$) mixture model. Then, for $g = 2, \ldots, K$, we estimate a g-component mixture by proposing to randomly split clusters obtained by the estimated model corresponding to $g - 1$ components. The way that clusters are split is determined by a random transformation of the estimated posterior classification probabilities $\widehat{w}_{ik}^{(g)}$, defined in Equation (18). Given $\widehat{w}_{ik}^{(g-1)}$, denote by $I_1, \ldots, I_{g-1}$ the clusters obtained applying the Maximum A Posteriori rule on the estimated model with $g - 1$ components. First, a non-empty component $I_{g^\star}$ is chosen at random among $\{I_1, \ldots, I_{g-1}\}$. Second, a new component labelled as g is formed by splitting the selected cluster $I_{g^\star}$ into two new ones, via a random transformation of the estimated posterior probabilities:

$$\begin{aligned}
\widehat{w}_{ik}^{(g)} &= \widehat{w}_{ik}^{(g-1)}, \quad k \notin \{g^\star, g\} \\
\widehat{w}_{ig^\star}^{(g)} &= u_i \widehat{w}_{ig^\star}^{(g-1)} \\
\widehat{w}_{ig}^{(g)} &= (1 - u_i) \widehat{w}_{ig^\star}^{(g-1)},
\end{aligned}$$

where $u_i \sim \text{Beta}(a, b)$, for $i = 1, \ldots, n$, with $\text{Beta}(a, b)$ denoting the Beta distribution with parameters $a > 0$ and $b > 0$. In our examples we have used $a = b = 1$, that

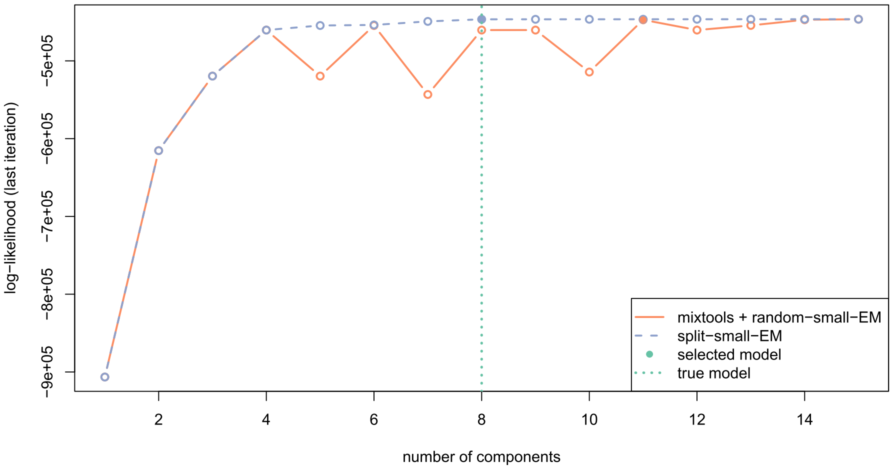


Fig. 2 Estimation of a multinomial mixture (without covariates) with $K = 8$ components: Log-likelihood values obtained at the last iteration of the EM algorithm for each value of K. The information criterion used to select K is the ICL. The small-EM scheme details are: $M_{\text{split}} = M_{\text{random}} = 10$ repetitions, each one consisting of $T = 5$ iterations

is, a Uniform distribution in (0, 1). Another valid option would be to set $a = b < 1$ in order to enforce greater cluster separation. Finally, the EM algorithm for a mixture with g components starts by plugging in $\{w_{ik}^{(g)}, i = 1, \ldots, n; k = 1, \ldots, g\}$ as starting values for the posterior membership probabilities. This procedure is repeated M_{split} times by running small EM algorithms and the one resulting to largest log-likelihood value is chosen to start the main EM for model g. We will refer to this strategy as a split-small-EM initialization scheme. A comparison between the random-small-EM strategy using `mixtools` (Benaglia et al. 2009) and the split-small-EM scheme for a mixture of 8 multinomial distributions is shown in Fig. 2. More detailed comparisons between the random small-EM initializations are reported in the simulation study of Sect. 4.1 and in Appendix C.

Shake small-EM

Assume that there are at least $K \geq 2$ clusters in the fitted model and that the estimated posterior membership probabilities are equal to $\widehat{w}_{ik}^{(K)}$, $i = 1, \ldots, n$, $k = 1, \ldots, K$. We randomly select 2 of them (say k_1 and k_2) and propose to randomly re-allocate the assigned observations within those 2 clusters. More specifically, let I_{k_1} and I_{k_2} denote the observations assigned (according to the MAP rule) to clusters k_1 and k_2, respectively. A small-EM algorithm is initialized by a state which uses a matrix $(\widehat{w}_{ik}^{\prime(K)})$ obtained by a random perturbation of the posterior probabilities as follows

$$\begin{aligned}
\widehat{w}_{ik}^{\prime(K)} &= \widehat{w}_{ik}^{(K)}, \quad k \notin \{k_1, k_2\} \\
\widehat{w}_{ik_1}^{\prime(K)} &= u_i(w_{ik_1} + w_{ik_2}) \\
\widehat{w}_{ik_2}^{\prime(K)} &= (1 - u_i)(w_{ik_1} + w_{ik_2}).
\end{aligned}$$

Note that in this way only the posterior probabilities of those observations assigned to components k_1 and k_2 are affected. This procedure is repeated M_{shake} times and the one leading to the highest log-likelihood value after T EM iterations is selected in order to initialize the algorithm. We will refer to this strategy as a shake small-EM initialization.

The aforementioned initialization schemes will be compared in our simulation study in Sect. 4.1 (see also Appendix C). We will use the notation

$$\text{EM}\left(M_{\text{split}}, M_{\text{shake}}, M_{\text{random}}\right)$$

to refer to a small-EM algorithm initialization consisting of M_{split} split-small-EM rounds, which are then followed by a sequence of M_{shake} shake-small-EM rounds and M_{random} random-small-EM rounds.

2.4 Estimation of the number of clusters under the EM algorithm

There is a plethora of techniques in order to select the number of components in a mixture model, see e.g. Chapter 6 in McLachlan and Peel (2004). One of the most popular choices is the Bayesian Information Criterion (Schwarz 1978), defined as

$$\text{BIC}(K) = -2\log f(\boldsymbol{y}|\boldsymbol{x}, \widehat{\boldsymbol{\theta}}_K) + d_K \log n,$$

where $\widehat{\boldsymbol{\theta}}_K$ and d_K denote the Maximum Likelihood estimate and the number of parameters of the mixture model with K components, respectively. Another criterion which is particularly suited to the task of model-based clustering is the Integrated Complete Likelihood (ICL) criterion (Biernacki et al. 2000).

$$\text{ICL}(K) = -2\log f(\boldsymbol{y}|\boldsymbol{x}, \widehat{\boldsymbol{\theta}}_K) + d_K \log n - 2\sum_{i=1}^{n}\sum_{k=1}^{K}\widehat{w}_{ik}\log\widehat{w}_{ik}.$$

It has been demonstrated that BIC may overestimate the number of clusters (see e.g. Rau et al. (2015); Papastamoulis et al. (2016)). In what follows, the number of clusters in the EM approach is selected according to the ICL criterion.

3 Bayesian formulation

We assume that the mixing proportions of the mixture model (7) follow a Dirichlet prior distribution, that is,

$$\boldsymbol{\pi} \sim \mathcal{D}(\alpha_1, \ldots, \alpha_K) \tag{19}$$

for some fixed hyper-parameters $\alpha_k > 0$, $k = 1, \ldots, K$. Usually, there is no prior information which separates the components between each other so typically (Marin et al. 2005) we set $\alpha_1 = \ldots = \alpha_K = \alpha > 0$ (see also Sect. 3.2).

The prior distribution of the coefficients β_{kjp} is normal centered on zero, that is,

$$\beta_{kjp} \sim \mathcal{N}(0, \nu^2), \quad \text{independent for} \quad k = 1, \ldots, K;\, j = 1, \ldots, J,\, p = 1, \ldots, P. \tag{20}$$

The prior variance ν^2 is assumed constant. A default value of $\nu^2 = 100$ was used in all of all our examples presented in subsequent sections, which corresponds essentialy to a vague[1] prior distribution, however we will also consider more informative choices ($\nu = 1$), in order to penalize large values of the coefficients. We furthermore assume that $\boldsymbol{\beta}$, $\boldsymbol{\pi}$ are a-priori independent random variables, thus the joint prior distribution of the parameters and latent allocation variables is written as

$$f(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\beta}|K, \boldsymbol{\alpha}, \nu) = f(\boldsymbol{z}|\boldsymbol{\pi}, K) f(\boldsymbol{\pi}|K, \boldsymbol{\alpha}) f(\boldsymbol{\beta}|K, \nu).$$

The joint posterior distribution of $\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{x}, K$ is written as

$$f(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{x}, K, \boldsymbol{\alpha}, \tau) \propto f(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta}, K) f(\boldsymbol{z}|\boldsymbol{\pi}, K) f(\boldsymbol{\pi}|K, \boldsymbol{\alpha}) f(\boldsymbol{\beta}|K, \nu)$$

3.1 A hybrid Metropolis-Adjusted-Langevin within Gibbs MCMC algorithm

From Equation (9) follows that the full conditional posterior distribution of the latent allocation vector for observation i is

$$\boldsymbol{Z}_i | \cdots \sim \mathcal{M}(1; w_{i1}, \ldots, w_{iK}), \tag{21}$$

independent for $i = 1, \ldots, n$.

The full conditional posterior distribution of mixing proportions is a Dirichlet distribution with parameters

$$\boldsymbol{\pi} | \cdots \sim \mathcal{D}(\alpha_1 + n_1, \ldots, \alpha_K + n_K), \tag{22}$$

where $n_k = \sum_{i=1}^{n} z_{ik}$.

For the regression coefficients we use a Metropolis–Hastings step, although other approaches which are based on the Gibbs sampler have been proposed (Dellaportas and Smith 1993; Holmes and Held 2006; Gramacy and Polson 2012). Note however that these approaches impose additional augmentation steps in the hierarchical model and have been applied only for simple (that is, $K = 1$) logistic regression models.

One could use a random walk for proposing updates to $\boldsymbol{\beta}$, but it is well known that the large number of parameters would lead to slow-mixing and poor convergence of the MCMC sampler. In order to overcome this issue, we used a proposal distribution which is based on the gradient information of the full conditional distribution. The Metropolis Adjusted Langevin Algorithm (MALA) (Roberts and Tweedie 1996; Roberts and

[1] This depends on the scale of the covariates, but in our simulations we are using standardized values in all cases.

Rosenthal 1998; Girolami and Calderhead 2011) is based on the following proposal mechanism

$$\widetilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(t)} + \tau \nabla \log f(\boldsymbol{\beta}^{(t)} | \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi}) + \sqrt{2\tau}\boldsymbol{\varepsilon}, \tag{23}$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $\nabla \log f(\boldsymbol{\beta}^{(t)} | \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi})$ denotes the gradient vector of the logarithm of the full conditional of $\boldsymbol{\beta}$, evaluated at $\boldsymbol{\beta}^{(t)}$. In order to select a value of τ with a reasonable acceptance rate betweeen proposed moves the MCMC sampler runs for an initial warm-up period. During this period τ is adaptively tuned as the MCMC sampler progresses in order to achieve acceptance rates of the proposed updates between user-specified limits (see Appendix A for details). The final value of τ is then selected as the one that will be used in the subsequent main MCMC sampler.

The derivative of the logarithm of the joint posterior distribution of $\boldsymbol{\beta}$, conditional on $\boldsymbol{z}$ and $\boldsymbol{\pi}$ is equal to

$$\frac{\partial \log f(\boldsymbol{\beta} | \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi})}{\partial \beta_{kjp}} = \sum_{i=1}^{n} z_{ik}(y_{ij} - S_i g_{ikj}) x_{ip} - \frac{\beta_{kjp}}{\nu^2} \tag{24}$$

Note that the first term on the right-hand side of the previous expression corresponds to the log-derivative of the complete log-likelihood (that is, given z), while the second term corresponds to the derivative of the prior distribution in (20).

The proposal in (23) is accepted according to the usual Metropolis-Hastings probability, that is,

$$\alpha(\boldsymbol{\beta}^{(t)}, \widetilde{\boldsymbol{\beta}} | \boldsymbol{z}^{(t)}, \boldsymbol{\pi}^{(t)}) = \min\left\{1, \frac{f(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}^{(t)}, \widetilde{\boldsymbol{\beta}}, \boldsymbol{\pi}^{(t)})\pi(\widetilde{\boldsymbol{\beta}})}{f(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\pi}^{(t)})\pi(\boldsymbol{\beta}^{(t)})} \frac{\mathrm{P}\left(\widetilde{\boldsymbol{\beta}} \to \boldsymbol{\beta}^{(t)}\right)}{\mathrm{P}\left(\boldsymbol{\beta}^{(t)} \to \widetilde{\boldsymbol{\beta}}\right)}\right\}, \tag{25}$$

where $\mathrm{P}(a \to b)$ denotes the probability density of proposing state b while in a. From (23) we have that $\mathrm{P}\left(\boldsymbol{\beta}^{(t)} \to \widetilde{\boldsymbol{\beta}}\right)$ is the density of the

$$\mathcal{N}_{KJP}\left(\boldsymbol{\beta}^{(t)} + \tau \nabla \log f(\boldsymbol{\beta}^{(t)} | \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}^{(t)}, \boldsymbol{\pi}^{(t)}), 2\tau I_{KJP}\right)$$

distribution, evaluated at $\widetilde{\boldsymbol{\beta}}$. The density of the reverse transition $\left(\widetilde{\boldsymbol{\beta}} \to \boldsymbol{\beta}^{(t)}\right)$ is equal to the density of the distribution

$$\mathcal{N}_{KJP}\left(\widetilde{\boldsymbol{\beta}} + \tau \nabla \log f(\widetilde{\boldsymbol{\beta}} | \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}^{(t)}, \boldsymbol{\pi}^{(t)}), 2\tau I_{KJP}\right)$$

evaluated at $\boldsymbol{\beta}^{(t)}$.

The overall procedure is summarized at Algorithm 1.

```
Input  : data $\boldsymbol{y}, \boldsymbol{x}$
         number of components $K$
         prior hyper-parameters $\nu^2, \boldsymbol{\alpha}$
         scale of the MALA proposal $\tau > 0$
         number of MCMC iterations $M$
         optional vector of starting values $(\boldsymbol{\pi}^{(0)}, \boldsymbol{z}^{(0)}, \boldsymbol{\beta}^{(0)})$
Output: MCMC sample $\{\boldsymbol{z}^{(t)}, \boldsymbol{\pi}^{(t)}, \boldsymbol{\beta}^{(t)};\ t = 1, \ldots, M\}$
and acceptance rate $r$ of the MALA proposal.
Step 0: Initialization
    Initialize acceptance rate counter: $r = 0$.
    if Starting values not supplied then
        Obtain random starting values $(\boldsymbol{\pi}^{(0)}, \boldsymbol{z}^{(0)}, \boldsymbol{\beta}^{(0)})$

for $t = 1$ to $M$
    Step 1: Gibbs sampling for $\boldsymbol{Z}$
        for $i = 1$ to $n$
            Simulate $\boldsymbol{z}_i^{(t)}$ from $\boldsymbol{Z}_i|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\beta}^{(t-1)}, \boldsymbol{\pi}^{(t-1)}$ in Equation (21)
        endfor

    Step 2: Gibbs sampling for $\boldsymbol{\pi}$
        Simulate $\boldsymbol{\pi}^{(t)}$ from $\boldsymbol{\pi}|\boldsymbol{z}^{(t)}$ in Equation (22)

    Step 3: MALA proposal for $\boldsymbol{\beta}$
        3.1 compute $\nabla \log f(\boldsymbol{\beta}^{(t-1)}|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}^{(t)}, \boldsymbol{\pi}^{(t)})$ according to (24)
        3.2 propose $\widetilde{\boldsymbol{\beta}}$ according to (23)
        3.3 Compute $\alpha(\boldsymbol{\beta}^{(t-1)}, \widetilde{\boldsymbol{\beta}}|\boldsymbol{z}^{(t)}, \boldsymbol{\pi}^{(t)})$ in (25)
        3.4 generate $u \sim \mathcal{U}(0, 1)$

        if $u < \alpha(\boldsymbol{\beta}^{(t-1)}, \widetilde{\boldsymbol{\beta}}|\boldsymbol{z}^{(t)}, \boldsymbol{\pi}^{(t)})$ then
            set $\boldsymbol{\beta}^{(t)} = \widetilde{\boldsymbol{\beta}}$ and $r \leftarrow r + 1$
        else
            set $\boldsymbol{\beta}^{(t)} = \boldsymbol{\beta}^{(t-1)}$

endfor
Set $r \leftarrow r/M$.

END of algorithm
```

Algorithm 1: Metropolis-Adjusted Langevin Within Gibbs MCMC

3.2 Estimation of the number of clusters using overfitting Bayesian mixtures

The Bayesian setup allows to estimate the number of clusters by using overfitting mixture models, that is, models where the number of mixture components is much larger than the number of clusters. Let $K_{\max} > K$ denote an upper bound on the number of clusters and define the overfitting mixture model

$$f(\boldsymbol{y}|\boldsymbol{\theta}, K_{\max}) = \sum_{k=1}^{K_{\max}} \pi_k f_k(\boldsymbol{y}|\boldsymbol{\theta}_k)$$

where $f_k \in \mathcal{F}_\Theta = \{f(\cdot|\boldsymbol{\theta}); \theta \in \Theta\}$ denotes a member of a parametric family of distributions. Let also d denote the dimension of free parameters in the distribution $f_k(\cdot)$. For instance, in the case of a mixture of multinomial logistic regression models with $J + 1$ categories and P covariates (including constant term) $d = JP$.

Rousseau and Mengersen (2011) show that the asymptotic behavior of the $K_{\max} - K$ extra components depends on the prior distributions of the mixing proportions $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{K_{\max}})$. For the case of a Dirichlet prior as in Equation (19), if $\max\{\alpha_k; k = 1, \ldots, K_{\max}\} < d/2$, the posterior weight of the extra components will tend to zero as $n \to \infty$ and force the posterior distribution to put all its mass in the sparsest way to approximate the true density.

Following Papastamoulis (2018), we set $\alpha_1 = \ldots = \alpha_K = \alpha$, thus the distribution of mixing proportions in Equation (19) becomes

$$\boldsymbol{\pi} \sim \mathcal{D}(\alpha, \ldots, \alpha) \tag{26}$$

where $0 < \alpha < d/2$ denotes a pre-specified positive number.

Therefore, the inference on the number of mixture components can be based on the posterior distribution of the "alive" components of the overfitted model, that is, the components which contain at least one allocated observation. In order to estimate the number of clusters we only have to keep track of the number of components with at least one allocated observation, across the MCMC run. This reduces to record the variable $K_0^{(t)} = ||\{k = 1, \ldots, K_{\max} : \sum_{i=1}^{n} z_{ik}^{(t)} > 0\}||$, where $z_i^{(t)}$ denotes the simulated allocation vector for observation i at MCMC iteration $t = 1, 2, \ldots$.

In order to produce a MCMC sample from the joint posterior distribution of the parameters of the overfitting mixture model (including the number of clusters), we embed the scheme described in Sect. 3.1 within a prior parallel tempering scheme (Geyer 1991; Geyer and Thompson 1995; Altekar et al. 2004). Each heated chain ($c = 1, \ldots, C$) corresponds to a model with identical likelihood as the original, but with a different prior distribution. Although the prior tempering can be imposed on any subset of parameters, it is only applied to the Dirichlet prior distribution of mixing proportions (van Havre et al. 2015; Papastamoulis 2018, 2020). The inference is based on the output of the first chain ($c = 1$) of the prior parallel tempering scheme (van Havre et al. 2015).

Let us denote by $f_c(\boldsymbol{\varphi}|\boldsymbol{x})$ and $f_c(\boldsymbol{\varphi})$; $c = 1, \ldots, C$, the posterior and prior distribution of the c-th chain, respectively. Obviously, $f_c(\boldsymbol{\varphi}|\boldsymbol{x}) \propto f(\boldsymbol{x}|\boldsymbol{\varphi}) f_c(\boldsymbol{\varphi})$. Let $\boldsymbol{\varphi}_c^{(t)}$

denote the state of chain c at iteration t and assume that a swap between chains c_1 and c_2 is proposed. The proposed move is accepted with probability $\min\{1, A(\boldsymbol{\pi}_{c_1}, \boldsymbol{\pi}_{c_2})\}$ where

$$A(\boldsymbol{\pi}_{c_1}, \boldsymbol{\pi}_{c_2}) = \frac{f_{c_1}(\boldsymbol{\varphi}_{c_2}^{(t)}|\boldsymbol{x}) f_{c_2}(\boldsymbol{\varphi}_{c_1}^{(t)}|\boldsymbol{x})}{f_{c_1}(\boldsymbol{\varphi}_{c_1}^{(t)}|\boldsymbol{x}) f_{c_2}(\boldsymbol{\varphi}_{c_2}^{(t)}|\boldsymbol{x})} = \frac{f_{c_1}(\boldsymbol{\varphi}_{c_2}^{(t)}) f_{c_2}(\boldsymbol{\varphi}_{c_1}^{(t)})}{f_{c_1}(\boldsymbol{\varphi}_{c_1}^{(t)}) f_{c_2}(\boldsymbol{\varphi}_{c_2}^{(t)})} = \frac{\widetilde{f}_{c_1}(\boldsymbol{\pi}_{c_2}^{(t)}) \widetilde{f}_j(\boldsymbol{\pi}_{c_1}^{(t)})}{\widetilde{f}_{c_1}(\boldsymbol{\pi}_{c_1}^{(t)}) \widetilde{f}_{c_2}(\boldsymbol{\pi}_{c_2}^{(t)})}, \tag{27}$$

and $\widetilde{f}_c(\cdot)$ corresponds to the probability density function of the Dirichlet prior distribution related to chain $c = 1, \ldots, C$. According to Equation (26), this is

$$\boldsymbol{\pi} \sim D\big(\alpha_{(c)}, \ldots, \alpha_{(c)}\big), \tag{28}$$

for a pre-specified set of parameters $\alpha_{(c)} > 0$ for chain $c = 1, \ldots, C$.

When estimating a Bayesian mixture model, a well known problem stems from the label switching phenomenon (Jasra et al. 2005), which arises from the fact that both the likelihood and prior distribution are invariant to permutation of the labels of mixture componets. The posterior distribution of the parameters will also be invariant, thus the parameters are not marginally identifiable. We deal with this problem by post-processing the MCMC output of the overfitting mixture via the ECR algorithm (Papastamoulis and Iliopoulos 2010; Papastamoulis 2016). Note that after post-processing the MCMC output for correcting label switching, the estimated classification for observation i is obtained as the mode of the (reordered) simulated values of $\boldsymbol{Z}_i$ in Eq. (21) across the MCMC run (after discarding the draws corresponding to the burn-in period of the sampler), $i = 1, \ldots, n$. For more details the reader is referred to the `label.switching` package (Papastamoulis 2016). The overall procedure is summarized in Algorithm 2.

```
Input  : data $\boldsymbol{y}, \boldsymbol{x}$
         upper bound on the number of clusters $K_{\max}$
         number of parallel chains $C$
         prior hyper-parameters $\nu^2, \alpha_{(1)}, \ldots, \alpha_{(C)}$
         number of MCMC cycles $T$
         MCMC iterations per cycle $m_1$
         number of iterations that will be used for warm-up $m_0$
         initial value of the scale of the MALA proposal $\tau$
Output: identified MCMC sample $\{\boldsymbol{z}^{(t)}, \boldsymbol{\pi}^{(t)}_{\text{alive}}, \boldsymbol{\beta}^{(t)}_{\text{alive}};\ t = 1, \ldots, T\}$
Step 0: Warm up the parallel chains
    for chain $c = 1$ to $C$
        0.0 Set $\tau_{(c)} = \tau$ for all $c = 1, \ldots, C$ and run Algorithm 1 with $M = m_0$
            and $\alpha_k = \alpha_{(c)}$; $k = 1, \ldots, K_{\text{max}}$.
        0.1 Adjust the scale of the MALA proposal $(\tau_{(c)})$ every $m_{check}$ iterations
        0.2 Set $\boldsymbol{\theta}^{(0)}_{(c)} \leftarrow \left(\boldsymbol{z}^{(m_0)}, \boldsymbol{\pi}^{(m_0)}, \boldsymbol{\beta}^{(m_0)}\right)$
    endfor

for MCMC cycle $t = 1$ to $T$
    Step 1: perform a cycle of MALA within Gibbs per chain
        for chain $c = 1$ to $C$
            1.1 Run Algorithm 1 with $M = m_1$, $\alpha_k = \alpha_{(c)}$; $k = 1, \ldots, K_{\text{max}}$
                starting value $\boldsymbol{\theta}^{(t-1)}_{(c)}$ and scale of the MALA proposal equal to $\tau =$
                $\tau_{(c)}$.
            1.2 Set $\boldsymbol{\theta}^{(t)}_{(c)} \leftarrow \left(\boldsymbol{z}^{(m_1)}, \boldsymbol{\pi}^{(m_1)}, \boldsymbol{\beta}^{(m_1)}\right)$
        endfor

    Step 2: Chain swaping
        2.1 Randomly choose $1 \leqslant c \leqslant C - 1$ and set $c_1 = c$, $c_2 = c + 1$
        2.2 Generate $u \sim \mathcal{U}(0, 1)$
        if $u < A(\boldsymbol{\pi}_{c_1}, \boldsymbol{\pi}_{c_2})$ in Equation (27) then
            set $\boldsymbol{\theta}^{(t)}_{c_1} \Longleftrightarrow \boldsymbol{\theta}^{(t)}_{c_2}$

endfor
Step 3: Undo label switching
    Apply the ECR algorithm on the output of chain $c = 1$ and return
      an identifiable MCMC sample $\{\boldsymbol{z}^{(t)}, \boldsymbol{\pi}^{(t)}_{\text{alive}}, \boldsymbol{\beta}^{(t)}_{\text{alive}};\ t = 1, \ldots, T\}$

END of algorithm
```

Algorithm 2: Prior Parallel Tempering MALA-within-Gibbs MCMC for Overfitting Mixtures of Multinomial Logistic Regressions

Regarding the initialization of the overfitting mixture model (see Step 0) in Algorithm 2, we use two alternative approaches. The first initialization is based on random starting values ("MCMC-RANDOM" scheme) and the second initialization scheme uses a more elaborate scheme, by exploiting the output of the EM algorithm under the split-small-EM scheme ("MCMC-EM" scheme). As expected, the latter scheme performs better as illustrated in the simulation studies. The overall procedure of the MCMC sampler is summarized in Algorithm 1 and Algorithm 2. The typical choices of the parameters as well as further details on the prior parallel tempering scheme and initialization schemes are given in Section A of the Appendix.

4 Applications

In Sect. 4.1 we use a simulation study in order to evaluate and rank the proposed methods in terms of their ability in clustering multinomial data. Next, we present two applications on real data: in Sect. 4.2 our method is used to identify clusters of age profiles within a regional unit in Greece and in Sect. 4.3 we study clusters of Facebook engagement metrics in Thailand. Further simulation results and comparisons with `flexmix` are reported in Appendix C.

4.1 Simulation study

In order to evaluate the ability of the proposed methods in clustering multinomial count data, we considered synthetic datasets generated from a mixture of multinomial logistic regression models (7). The number of multinomial replicates (S_i) per observation is drawn from a negative binomial distribution: $S_i \sim \mathcal{NB}(r, p)$ with number of successful trials $r = 20$ and probability of success $p = 0.025$. We simulated 500 datasets in total where the values of n, K, P, J are uniformly drawn in the range of values shown in Table 1. Given K, the weight of each cluster was equal to $\pi_k \propto k$, $k = 1, \ldots, K$. Notice that this setup gives rise to mixture models with total number of free parameters ranging from 10 up to 535.

The true values of the regression coefficients were simulated according to

$$\begin{aligned}\sigma &\sim \mathcal{U}(1, 5)\\ \beta_{kjp}|\sigma &\sim 0.5\mathrm{I}_{\{0\}}(\beta) + 0.5\phi(\beta; 0, \sigma^2)\end{aligned}$$

(conditionally) independent for $k = 1, \ldots, K$; $j = 1, \ldots, J$; $p = 1, \ldots, P$, where $\mathrm{I}_{\{0\}}(\beta)$ denotes a discrete distribution degenerate at 0 and $\phi(\beta; \mu, \sigma^2)$ denotes the density function of the normal distribution with mean μ and variance σ^2.

We applied the proposed methodology in order to estimate mixtures of multinomial logit models. In particular we compared the EM algorithm under three initialization schemes, as well the MCMC sampling scheme under random initialization and an

Table 1 Values for sample size (n), number of clusters (K), covariates (P) and number of categories ($J + 1$) in the simulation study

n	K	P	$J + 1$
$\{125, 250, 500, 1000\}$	$\{1, \ldots, 8\}$	$\{2, 4, 6\}$	$\{6, 9, 12\}$

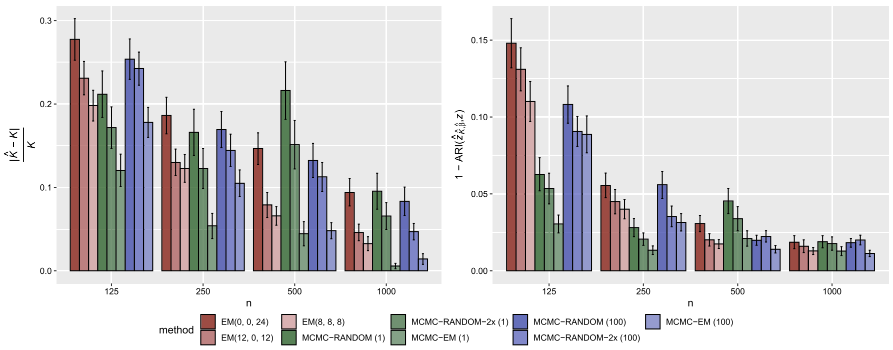
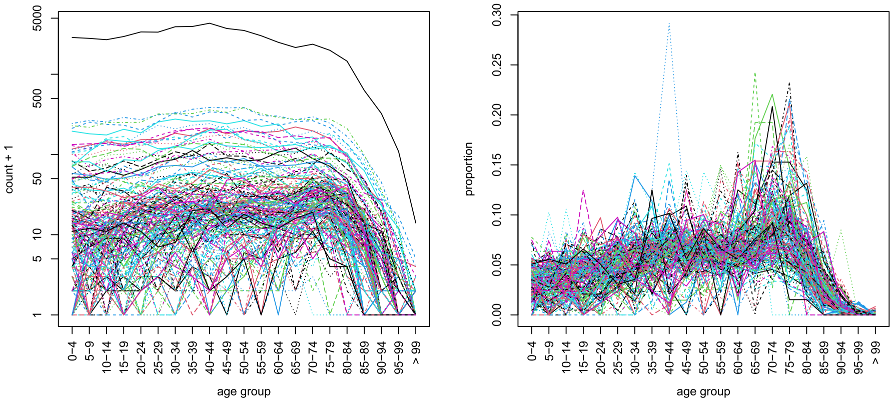


Fig. 3 Simulation study summary. See Table 1 for the simulation study set-up. For the EM implementations: the three numbers in parenthesis refer to the number of split, shake and random small-EM runs. For the MCMC implementations, the number in parenthesis denotes the prior variance (ν^2) of the coefficients β_{kjp} in Equation (20)

initialization based on the output of the EM algorithm (under the split-shake-random small-EM scheme). In total we considered 24 different starts in the small-EM schemes: a random small-EM with 24 starts: EM(0, 0, 24), a combination of split and random small-EM with 12 starts each: EM(12, 0, 12) and finally, a combination of split, random and shake small-EM with 8 starts each: EM(8, 8, 8). The total number of MCMC iterations is held fixed at 100,000. We also present results when considering the double amount of iterations (both in the warm-up period as well as the main MCMC sampler) under the MCMC-RANDOM scheme, which we will denote by MCMC-RANDOM-2x. Finally, we considered two different values of prior variance of the coefficients in Eq. (20): $\nu = 1$ and $\nu^2 = 100$. The first choice corresponds to an informative prior distribution, heavily penalizing large values of $|\beta_{kjp}|$. The second choice corresponds to a vague prior distribution. The chosen value of ν will be denoted in a parenthesis, that is, MCMC-RANDOM (ν^2) and MCMC-EM (ν^2) will indicate the output of MCMC algorithm with random and EM initialization schemes (respectively) and prior variance equal to ν^2. See Appendix A for further details of various other parameters for the EM and MCMC algorithms.

Figure 3 illustrates a graphic summary of the simulation study findings, based on our 500 synthetic datasets. The metrics we are focusing are the following: The left graph shows the mean of relative absolute error $\frac{|\hat{K}-K|}{K}$ between the estimated number of clusters ($\hat{K}$) and the correspoding true value (K). The right graph displays the mean of the adjusted Rand index (with respect to the ground-truth classification) subtracted from 1. In all cases we conclude that the EM algorithm with a random small-EM initialization (denoted as EM(0,0,24)) is worse compared to the split-shake-random small EM initialization (denoted as EM(8,8,8)). Regarding the MCMC sampler we see that the random initialization scheme (MCMC-RANDOM) is worse than the EM-based initialization (MCMC-EM), when both MCMC-RANDOM and MCMC-EM run for the same number of iterations. However, as the number of iterations increases in the randomly initialized MCMC sampler (MCMC-RANDOM-2x), the results are improved, particularly for the mean relative absolute error of the estimation of the



(a) (b)

Fig. 4 Age profiles for $n = 187$ settlements in the Phthiotis regional unit according to the 2011 census of Eurostat. **a** Population counts (increased by 1) displayed in log-scale in the y axis and **b** relative frequency of population counts

number of clusters. Overall, the MCMC algorithm initialized by the (split-small) EM solution is the best performing method, closely followed by the EM algorithm under the split-small EM scheme. Naturally, the informative prior distribution ($\nu = 1$, corresponding to the green-coloured bars in Fig. 3) outperforms the vague prior distribution ($\nu^2 = 100$, corresponding to the red-coloured bars). More detailed summaries of the resulting estimates are given in Appendix C.

4.2 Phthiotis population dataset

In this example we present an application of our methodology in clustering areas within a certain region with respect to the age profiles of their population, taking also into account geographical covariate information. For this purpose we considered population data based on the 2011 census of Eurostat[2]. We considered the Phthiotis area, a regional unit located in central Greece. Our extracted dataset consists of number of people per age group (21 groups: $0-4, 5-9, \ldots, 95-99, > 99$ years old) for a total of $n = 187$ settlements (such as villages, towns, and the central city of the regional unit, *Lamia*), as displayed in Fig. 4. The separated line in the upper part of Fig. 4a corresponds to *Lamia*. Observe that there are various regions where there is a peak in the older population groups (between 65 and 85), as vividly displayed when looking at the plot of relative frequencies per age group at Fig. 4b. A different behaviour is obvious for *Lamia* where we see that the dominating age groups are between $30 - 50$. The research question is to cluster these settlements based on the age profiles of their population.

If we cluster the raw dataset of age counts within each group using a mixture of multinomial distributions without any covariate information, then a large number

[2] https://ec.europa.eu/eurostat/web/main/data/database.

of clusters is found. In particular, when using `mixtools` (Benaglia et al. 2009), a relatively large number of clusters is found ($\hat{K} = 8$ using ICL). Therefore, we opt to apply our method using the following covariate information for settlement $i = 1, \ldots, n$:

x_{i1}: distance (in Km) from *Lamia* (capital city of the regional unit)
x_{i2}: logarithm of the altitude (elevation, in m).

Both covariates were scaled to zero mean and unit variance. Let us denote by y_{ij} the number of people in age group $j = 1, \ldots, 21$ for settlement $i = 1, \ldots, n$. The probability $\theta_{kj}^{(i)}$ denotes the proportion of population being in age group j conditional on the event that settlement i belongs to cluster k, where $\sum_{j=1}^{21} \theta_{kj} = 1$ for all k. Conditional on cluster $k = 1, \ldots, K$, the random vector $\boldsymbol{Y}_i = (Y_{i1}, \ldots, Y_{i,21})^\top$ is distributed according to a multinomial distribution

$$\begin{aligned} \boldsymbol{Y}_i | Z_{ik} = 1 &\sim \mathcal{M}_{21}(S_i, \boldsymbol{\theta}_k^{(i)}) \\ \log \frac{\theta_{kj}^{(i)}}{\theta_{k,1}^{(i)}} &= \beta_{kj0} + \beta_{kj1} x_{i1} + \beta_{kj} x_{i2}, \quad j = 2, \ldots, 21 \end{aligned}$$

where $\boldsymbol{\theta}_k^{(i)} = (\theta_{k1}^{(i)}, \ldots, \theta_{k,21}^{(i)})$ for $k = 1, \ldots, K$ and S_i denotes the total population for settlement i, for $i = 1, \ldots, n$. Note that we have used the 1st category (ages between 0 and 4) as baseline in order to express the log-odds of the remaining groups. The distribution of counts per age group is written as a mixture of multinomial distributions

$$\boldsymbol{Y}_i \sim \sum_{k=1}^{K} \pi_k \mathcal{M}_{21}(S_i, \boldsymbol{\theta}_k^{(i)}), \quad \text{independent for} \quad i = 1, , \ldots, n$$

where π_k denotes the weight of cluster k. Hence, each cluster represents areas with different age profile as reflected by the corresponding vector of multinomial probabilities. The total number of clusters (K) is unknown.

At first we used the EM algorithm under the proposed initialization scheme to estimate mixtures of multinomial logistic regression models for a series of $K = 1, 2, \ldots, K_{\max} = 10$ components. According to the ICL criterion, the selected number of clusters is equal to $K = 3$. Next we estimated an overfitting Bayesian mixture of $K_{\max} = 10$ components, using a prior parallel tempering scheme based on 12 chains. The MCMC algorithm was initialized from the EM solution, while all remaining parameters were initialized from a zero value. The MCMC sampler ran for a warm-up period of 100,000 iterations, followed by 400,000 iterations. A thinned MCMC sample of 20,000 iterations was retained for inference. In almost all MCMC draws the number of non-empty mixture components was equal to $K_0 = 3$ (estimated posterior probability equal to 99.5%). The retained MCMC sample was then post-processed according to the ECR algorithm (Papastamoulis and Iliopoulos 2010; Papastamoulis 2016) in order to undo label switching. The confusion matrix of the single best clusterings between the two methods (EM and MCMC) is displayed in Table 2. The correspond-

Table 2 Confusion matrix between the single best clustering of the Phthiotis Population Dataset arising from the EM and MCMC algorithms (after post-processing the MCMC output for correcting label switching)

	MCMC		
	1	2	3
EM			
1	29	2	0
2	1	73	5
3	0	12	65

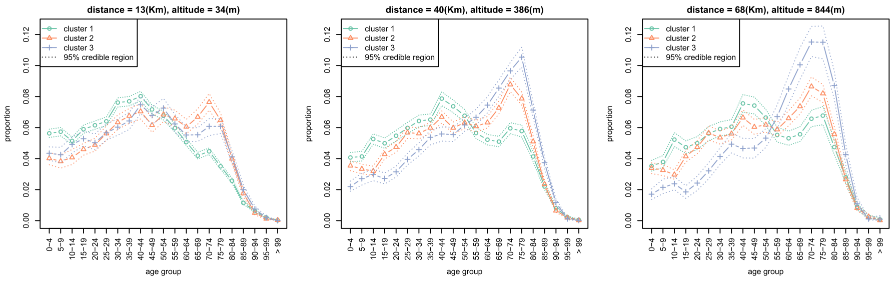
ing adjusted Rand index is equal to 0.67 indicating that the two resulting partitions have strong agreement.

Next we focus on the results according to the MCMC algorithm (after post-processing). Figure 5 illustrates the posterior mean (and 95% credible region) of the probability $\theta_{kj}^{(i)}$ for age group j in cluster $k = 1, \ldots, 3$. Three characteristic configurations of covariate levels were used, that is, the 0.1, 0.5 and 0.9 percentiles of the two covariates. In all cases we see distinct age group characteristics and what is evident is the presence of a group which contains places with younger age profiles (cluster 1). In cluster 3, notice a strong peak at the group of ages between 76 to 84, which emerges in cases of moderate to large values of the two covariates. In cluster 2, the peak is also located at the older age groups however it is less pronounced compared to cluster 3. Figure 6 visualizes the three clusters on the map of the regional unit. We may conclude that cluster 1 (the "younger" cluster) mainly consists of settlements that are either located close to *Lamia* (gray spot on the map), including *Lamia* itself, or their total population is larger than 1000 (towns such as *Makrakomi*, *Malessina*, *Sperchiada*, *Atalanti*, *Domokos*, *Stavros* and the central city of *Lamia*). However this younger group of age profiles is also present in some of the most distant and mountainous southwestern areas (*Dafni*, *Neochori Ypatis*, *Kastanea*, *Pavliani* and *Anatoli*: the altitude of these small villages is larger than 1000 m). In general, however, as we move further away from *Lamia* the "older" and "eldest" clusters dominate, particularly for areas with a small number of population. See also the histogram of settlement populations per cluster in Fig. 7. Note that the majority of smaller villages (population of 100 citizens, approximately) are mainly assigned to the third cluster (the eldest group).

Finally, we have to mention that this specific application involves an ordinal and not a nominal response. Therefore, one could use alternative techniques to model the data, such as proportional odds models, or smoothing the changes between adjacent categories.

4.3 Facebook live sellers in Thailand data set

The dataset of Dehouche (2020) (see also Wongkitrungrueng et al. (2020)) contains engagement metrics of Facebook pages for Thai fashion and cosmetics retail sellers. We consider the number of emoji reactions for each Facebook post, which are known as "like", "love", "wow", "haha", "sad" and "angry". The aim of our analysis is to cluster posts based on the reaction profiles, using additional covariate information. Each post



(a) (b) (c)

Fig. 5 Posterior mean and 95% credible region of age profiles per cluster for the Phthiotis population data. The two covariates (distance from Lamia and altitude) are set equal to the corresponding 0.1 (**a**), 0.5 (**b**) and 0.9 (**c**) percentiles

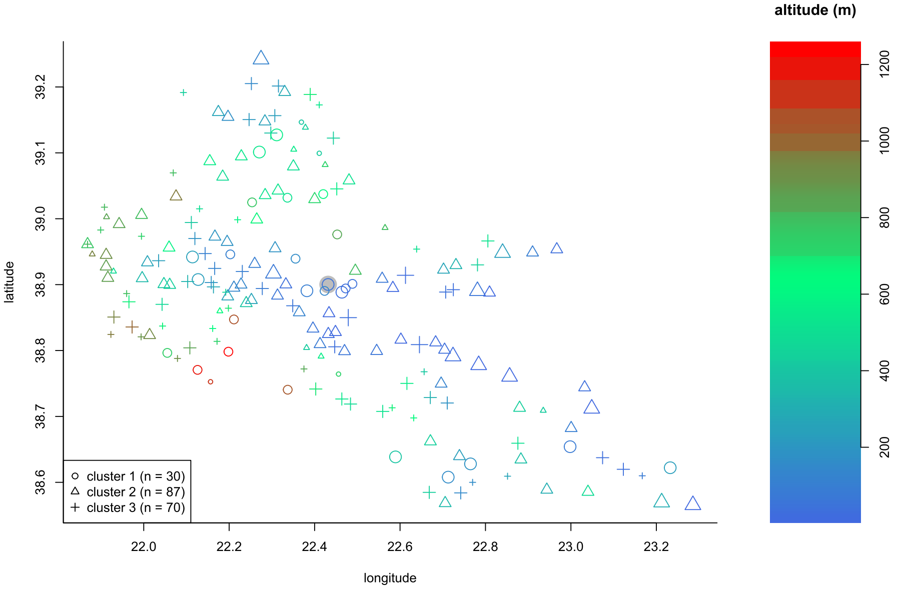


Fig. 6 Geographical coordinates of the settlements and inferred cluster membership according to the Maximum A Posteriori rule on the output of the MCMC sampler. The gray circle indicates Lamia, that is, the central city of the Phthiotis region. Different point sizes are used according to the total population of each settlement: small ($S_i < 150$), medium $150 \leq S_i \leq 999$ and larger ($S_i > 999$)

can be of a different nature (“video”, “photo”, “status”), a categorical variable which we are taking into account as a categorical predictor. In addition, we also use as covariate the number of shares per post (in log-scale). The dataset is available at the UCI machine learning repository[3].

[3] https://archive.ics.uci.edu/ml/datasets/Facebook+Live+Sellers+in+Thailand.

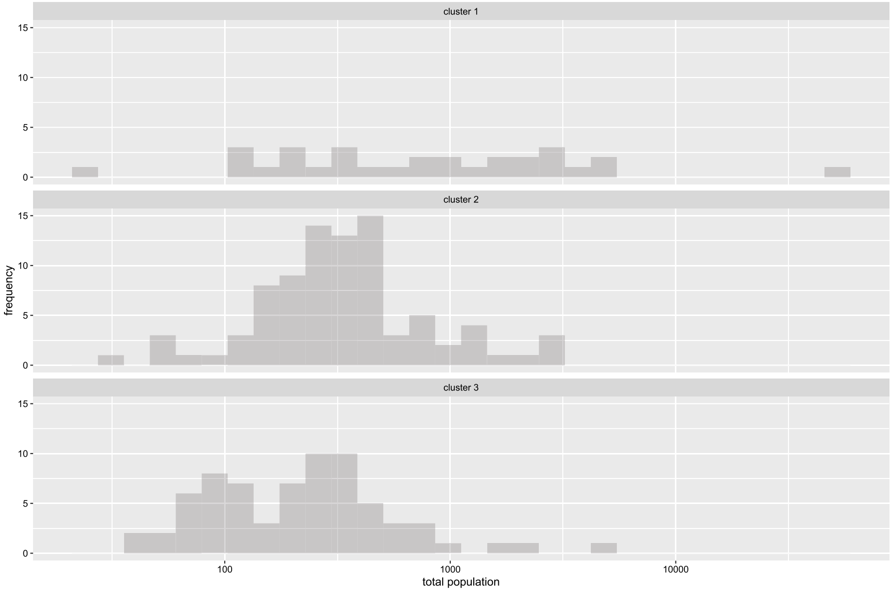


Fig. 7 Total population counts per cluster of the Phthiotis dataset

We considered the period between 2017-01-01 and 2018-12-31 taking into account posts with a minimum overall number of reactions equal to 40. We then randomly selected 100 posts per type (100 videos, 100 photos and 100 statuses), that is, 300 posts in total. The observed data is displayed in Fig. 8 (note that for the sole purpose of visualization in the log-scale, each count is increased by 1). It is evident that most

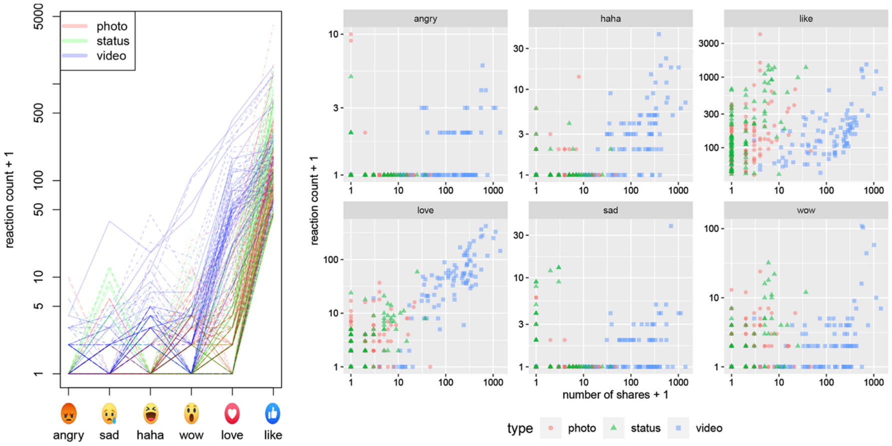


Fig. 8 Reaction counts for 300 posts of the Facebook Live Sellers Dataset. A different colour displays the type of each post (100 video, 100 photos and 100 statuses). Note that the y axis on both graphs as well as the x axis of the right graph are displayed in log-scale after increasing each observed count by one

reactions correspond to “loves” and “likes”. There is also some visual evidence that videos may result to a larger number of “loves” compared to photos or statuses. On the other hand, many posts result to zero counts for any kind of reaction other than “like”. So we might expect that such a dataset exhibits heterogeneity, due to zero inflation in the first five categories. Thus, it makes sense to cluster posts according to the reaction profiles, i.e. reaction probability.

Let us denote by $\boldsymbol{y}_i = (y_1, y_2, y_3, y_4, y_5, y_6)^\top$ the observed vector of reaction counts for post $i = 1, \ldots, n$ ($n = 300$). We assume that $\boldsymbol{y}_i$, conditional on post type and number of shares (as well as the total number of reactions for that particular post), is distributed according to a mixture of multinomial distributions with $J + 1 = 6$ categories, where y_j denotes the number of reactions of type j for post i, $i = 1, \ldots, n$. The type of each post serves as a categorical predictor with three levels (“video”, “photo” and “status”). Selecting the probability of “like” as the reference category and conditional on cluster $k = 1, \ldots, K$, the multinomial logit model is written as

$$\log \frac{\theta_{kj}^{(i)}}{\theta_{k6}^{(i)}} = \beta_{kj0} + \beta_{kj1} x_i^{\text{status}} + \beta_{kj2} x_i^{\text{photo}} + \beta_{kj3} \log(1 + x_i^{\text{shares}}), \quad j = 1, 2, 3, 4, 5$$

where $\theta_{kj}^{(i)}$ denotes the probability of reaction j corresponding to “angry” ($j = 1$), “sad” ($j = 2$), “haha” ($j = 3$), “wow” ($j = 4$), “love” ($j = 5$) and “like” ($j = 6$). Note that the categorical predictor consists of three levels, thus, we created the two dummy variables

$$x_i^{\text{status}} = \begin{cases} 1, \text{ if post } i \text{ is “status”} \\ 0, \text{ otherwise} \end{cases}, x_i^{\text{photo}} = \begin{cases} 1, \text{ if post } i \text{ is “photo”} \\ 0, \text{ otherwise} \end{cases}$$

after selecting the “video” type as the baseline. In addition, x_i^{shares} denotes the number of shares for post i.

We applied our method using the EM algorithm with the proposed initialization scheme as well as the MCMC sampler using an overfitting mixture model with $K_{\max} = 10$ components. A total of 12 chains under the prior parallel tempering scheme were considered. The MCMC sampler ran for an initial warm-up period of 100,000 iterations, followed by 400,000 iterations. A thinned MCMC sample of 20,000 iterations was retained for inference. Both methods select $K = 4$ clusters. In the MCMC sampler we have considered two different levels of the prior variance of the regression coefficients, that is, $\nu^2 = 100$ (vague prior distribution) and $\nu^2 = 1$ (informative prior).

More specifically, for the EM algorithm the minimum value of ICL is equal to 4610.89 (corresponding to a model with $K = 4$ clusters) while for the MCMC sampler the mode of the posterior distribution of the number of non-empty mixture components corresponds to $K_0 = 4$, with $\hat{P}(K_0 = 4|\text{data}) = 0.67$ for the sampler with $\nu^2 = 100$. The same number of components is also selected when considering the prior distribution with $\nu^2 = 1$, where $\hat{P}(K_0 = 4|\text{data}) = 0.80$. The confusion matrix of the single best clusterings arising after applying the Maximum A Posteriori rule is displayed in Table 3. The corresponding adjusted Rand Indices between the two

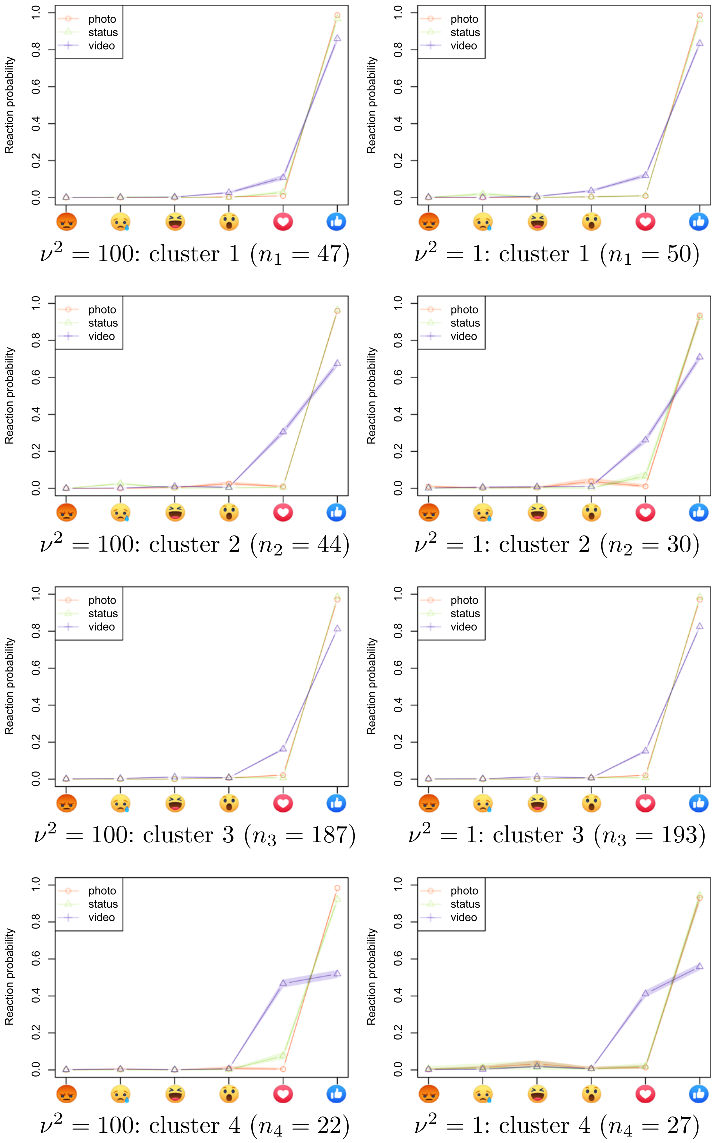
Table 3 Confusion matrix between the single best clustering of the Facebook Live Sellers Dataset arising from the EM and MCMC algorithms with a prior variance of regression coefficients equal to $\nu^2 = 100$ and $\nu^2 = 1$ (after post-processing the MCMC output for correcting label switching)

	MCMC ($\nu^2 = 100$)				MCMC ($\nu^2 = 1$)			
	1	2	3	4	1	2	3	4
EM								
1	47	0	0	0	37	2	8	0
2	0	44	3	0	10	20	6	11
3	0	0	183	0	2	3	178	0
4	0	0	1	22	1	5	1	16

partitions are equal to ARI(EM, MCMC(100)) ≈ 0.96, ARI(EM, MCMC(1)) ≈ 0.74 and ARI(EM, MCMC(100)) ≈ 0.76, indicating a high level of agreement between the three approaches.

Next we are concerned with the identifiability of the selected model with $K = 4$ components. Recall that in our extracted dataset the minimum number of reactions is equal to 40, thus, condition 1.(a) in Theorem 2 of Grün and Leisch (2008b) (see also Hennig (2000)) is satisfied. If we were considering only the categorical predictor (video type) in our model, the number of distinct hyperplanes (lines in this case) needed to cover the covariates of each cluster would be equal to 2: one line covering the points (0, 0) (origin) and $(x_i^{\text{status}}, x_i^{\text{photo}}) = (1, 0)$ and another line covering the points (0, 0) and $(x_i^{\text{status}}, x_i^{\text{photo}}) = (0, 1)$. This number is less than the selected number of clusters and the coverage condition (see condition 1.(b) in Theorem 2 of Grün and Leisch (2008b)) is violated. However, this condition is satisfied after including a continuous covariate with cluster-specific effect (number of shares). Finally, the generated MCMC sample has been post-processed according to the ECR algorithm (Papastamoulis and Iliopoulos 2010; Papastamoulis 2016) in order to deal with the label switching issue.

Figure 9 displays the posterior mean of reaction probability per cluster and the corresponding (equally tailed) 95% credible interval, for both prior setups. The continuous covariate (number of shares) is set equal to the observed mean per post type. In all cases, there is an increased probability of a "love" reaction when the post is a video, compared to a photo or a status. However, the average probability of such a reaction is different between the clusters, with the most notable difference obtained in cluster "4". Finally, notice the similarity of cluster profiles for both prior distributions.



$\nu^2 = 100$: cluster 1 $(n_1 = 47)$

$\nu^2 = 1$: cluster 1 $(n_1 = 50)$

$\nu^2 = 100$: cluster 2 $(n_2 = 44)$

$\nu^2 = 1$: cluster 2 $(n_2 = 30)$

$\nu^2 = 100$: cluster 3 $(n_3 = 187)$

$\nu^2 = 1$: cluster 3 $(n_3 = 193)$

$\nu^2 = 100$: cluster 4 $(n_4 = 22)$

$\nu^2 = 1$: cluster 4 $(n_4 = 27)$

Fig. 9 Posterior mean and 95% Credible Region of the reaction probabilities $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ per cluster for the Facebook Sellers data, when setting the continuous covariate (number of shares) equal to its mean per post type. The left and right columns correspond to the MCMC samplers with the large $(\nu^2 = 100)$ and small $(\nu^2 = 1)$ prior variance of the regression coefficients in Equation (20), respectively

5 Discussion

The problem of clustering multinomial count data under the presence of covariates has been treated using a frequentist as well as a Bayesian approach. Our simulations showed that our proposed models perform well, provided that the suggested estimation and initialization schemes are selected. The application of our method in clustering real count datasets reveal the interpretability of our approach in real-world data. Our contributed package in R makes our method directly available to the research community.

Under a frequentist approach we have demonstrated that an efficient initialization (i.e. the split-shake-random small-EM scheme in Sect. 2.3) yields improved results, when compared to a more standard random small-EM initialization scheme. Furthermore, a crucial point for the implementation of the maximization step of the EM algorithm is the control of the step size of the Newton-Raphson iterations, something that was achieved using the ridge-stabilized version in Sect. 2.2.

We did not address the issue of estimating standard errors in our EM implementation. However, these can be obtained by approximating the covariance matrix of the estimates by the inverse of the observed information matrix (Louis 1982; Meng and Rubin 1991; Jamshidian and Jennrich 2000) or using bootstrap approaches (Basford et al. 1997; McLachlan et al. 1999; Grün and Leisch 2004; Galindo Garre and Vermunt 2006). Maximum likelihood estimation with the EM algorithm can be modified in order to provide Maximum A Posteriori estimates under a regularized likelihood approach, as implemented in Galindo Garre and Vermunt (2006).

The Bayesian framework of Sect. 3 has clear benefits over the frequentist approach, but of course, under the cost of increased computing time. As demonstrated in our simulations, the proposed MCMC scheme outperforms the EM algorithm in terms of estimation of the number of clusters as well the clustering of the observed data in terms of the Adjusted Rand Index. Moreover, the Bayesian setup allows for even greater flexibility in the resulting inference, such as the calculation of Bayesian credible intervals from the MCMC output which provide a direct assessment of the uncertainty in our point-estimates. For this purpose we used state-of-the-art algorithms that deal with the label switching problem in mixture, suitably adjusted to the special framework of overfitting mixture models.

A natural and interesting extension of our research is to consider the problem of variable selection in model based clustering (Maugis et al. 2009; Dean and Raftery 2010; Yau and Holmes 2011; Fop and Murphy 2018). In the Bayesian setting, one could take into account alternative prior distributions of the multinomial logit coefficients per cluster, e.g. spike and slab or shrinkage prior distributions (Malsiner-Walli et al. 2016; Vávra et al. 2022) that encourage sparsity in the model. Another direction for future research is to combine our mixture model with alternative Bayesian logistic regression models that exploit data augmentation schemes (Held and Holmes 2006; Frühwirth-Schnatter and Frühwirth 2010; Polson et al. 2013; Choi and Hobert 2013) and assess whether MCMC inference is improved.

Acknowledgements The author wishes to thank three anonymous reviewers for their valuable comments and suggestions. This study was funded by the program "DRASI 1/Research project: 11338201", coordinated by the Research Center of the Athens University of Economics and Business (RC/AUEB). The author wishes to thank Prof. Dim- itris Karlis (Athens University of Economics and Business) for helpful discussions.

Funding Open access funding provided by HEAL-Link Greece. The author was funded by the program "DRASI 1/Research project: 11338201", coordinated by the Research Center of the Athens University of Economics and Business (RC/AUEB).

Data availability All datasets and software used in this paper are available online at https://github.com/mqbssppe/multinomialLogitMix.

Declaration

Conflict of interest The authors declare that they have no conflict of interest.

Appendix A: Details of the MCMC sampler

This section describes the values of the parameters for the EM and MCMC sampler, which were used in order to produce the results reported in our simulation study in Sect. 4.1.

Prior parameters and number of parallel chains

We implemented the prior parallel tempering MALA-within-Gibbs algorithm described in the previous section with the following set-up. We used 8 parallel chains and overfitted mixtures with a total of $K_{\max} = 20$ components. For chain $c = 1, \ldots, C$, the parameters of the Dirichlet prior $\mathcal{D}(\alpha_c, \ldots, \alpha_c)$ were set equal to

$$\alpha_1 = \frac{1}{200} \quad \text{(target chain)} \tag{29}$$

$$\alpha_c = \frac{1}{200} + \frac{1}{4000}\exp\left\{2 + 12\frac{c-2}{C-2}\right\}, \quad \text{for chain} \quad c = 2, \ldots, C. \tag{30}$$

The specific set of values worked reasonably well in our simulations and applications on real datasets, however, we do not claim that they are in any way "optimal" choices. Our guide for choosing this set of values is to achieve average acceptance rate between chain swaps around 10%–30%. The variance of the normal prior distribution of the coefficients in (20) is set to $\nu^2 = 100$. Notice that in our applications we have also presented results with a much smaller prior variance ($\nu^2 = 1$), a choice which can be seen as a regularized estimation approach.

Warm-up period

The warm-up period of the MCMC sampler consists of $m_0 = 48000$ iterations. As the sampler progresses, it keeps track of the proposal acceptance ratio within the last 500 iterations. The parameter τ (which controls the scale of the MALA proposal) is adaptively tuned in order the proposal acceptance rate stays within the range $15\% - 25\%$. In case that a sequence of 500 iterations the acceptance rate is less than 15% then $\tau \leftarrow 0.9\tau$. On the other hand, if the acceptance rate is too large (larger than 25%), then $\tau \leftarrow \tau/0.9$. The value of τ obtained at the last iteration of this initial phase is used at the main MCMC sampler. The initial value is set equal to $\tau = 0.00035$.

Main MCMC sampler

After the warm-up period, the MCMC sampler runs for a total of $T = 2600$ cycles. Each cycle consists of $m_1 = 20$ iterations. A chain swap is attempted at the end of each MCMC cycle. The results are obtained retaining the last 2500 cycles (and discarding the first 100 cycles as burn-in period) of the MCMC sampler.

Note that the initial warm-up period and the main MCMC sampler consist of $48{,}000 + 2600 \times 20 = 10{,}0000$ MCMC iterations in total.

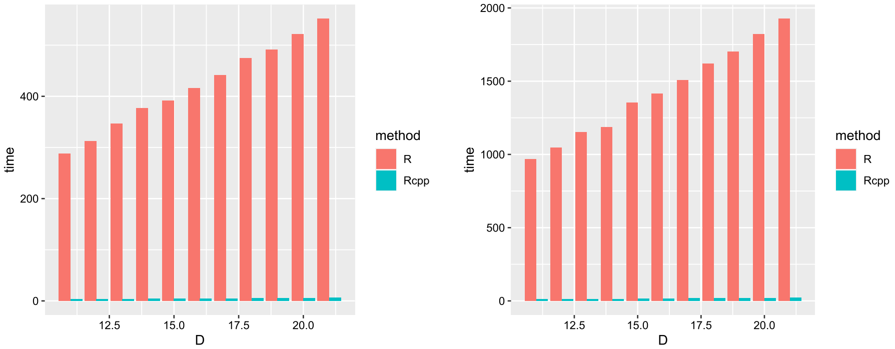
MCMC sampler initialization

Each chain was initialized under two different schemes: a scheme based on randomly selected started values (which we will be referring to as "MCMC-RANDOM") and a more elaborate initialization scheme based on the output of the EM algorithm ("MCMC-EM" starting scheme). More specifically, in "MCMC-RANDOM" all parameters are initialized by simulating from the prior distributions. In "MCMC-EM" we first estimate the number of clusters as well as the model parameters according to the EM algorithm under our split-small-EM scheme. Let us denote by $\hat{K}^{(EM)}$ the selected number of clusters according to the EM algorithm (using ICL), under the split-small-EM initialization. Next, consider a Bayesian overfitting mixture with $K_{\max} > K^{(EM)}$ components. The parameters of the first $K^{(EM)}$ components are all set equal to the values of the corresponding parameters obtained at the last iteration of the EM algorithm for that particular model. The parameters of the remaining $K_{\max} - K$ components are all initialized by a zero value. Finally, a random permutation is drawn among the initial parameters of the $K_{\max}$ components for each of the different chains in order to encourage the presence of the label switching phenomenon in the MCMC sampler.

Appendix B: Computational details: package in R

The computational pipeline for the proposed methodology has been implemented in R. It is furthermore publicly available as a contributed R package named `multinomialLogitMix`, which is available at https://CRAN.R-project.org/package=multinomialLogitMix. The proposal scheme of the MALA sampler is implemented in the `Rcpp` (Eddelbuettel and François 2011; Eddelbuettel 2013; Eddelbuettel and Balamuta 2018) and `RcppArmadillo` (Eddelbuettel and Sanderson 2014) packages, which integrate R and C++. Figure 10 illustrates that the gain in computing time is tremendous when replacing the R code with Rcpp.

The basic pipeline is illustrated next. For this purpose we use a simulated dataset as the ones in Sect. 4.1.



$K = 3$ $K = 10$

Fig. 10 Benchmarking runtime for the MALA proposal (Step 3 of Algorithm 1) between `R` and `Rcpp` for different values of number of mixture components (K) and multinomial categories (D)

```
library("multinomialLogitMix")
set.seed(727)
# sample size
n <- 250
# number of covariates (constant term included)
p <- 3
# number of multinomial categories
D <- 6
# number of clusters
K <- 2
# maximum number of clusters
Kmax <- 10
# generate synthetic as in Section 4.1 of the paper
simData <- simulate_multinomial_data(K = K, p= p, D = D, n = n , size = 20,
      prob = 0.025)
# response (multinomial counts)
y <- simData$count_data
# design matrix
X <- simData$design_matrix
# preview multivariate response counts
> y
       [,1] [,2] [,3] [,4] [,5] [,6]
  [1,]  209  118    0    0  536  124
  [2,]  703    0    0    0    0    0
<..........(+ 246 rows of counts)..>
[249,]  819    2   47    4    3    3
[250,]  710    0    0    0    0    1
# preview design matrix
> X
       [,1]          [,2]         [,3]
  [1,]    1 -0.8469699933  0.088500232
  [2,]    1  0.5114674818  2.136154036
<..........(+ 246 rows)................>
[249,]    1  0.1812801447  0.583363993
[250,]    1  0.5725885602  1.734839847
# set the number of cores to 8
nCores <- 8
# Run both EM and MCMC with default setttings (based on the split small-EM
     initialization scheme)
mlm_split <- multinomialLogitMix(response = y,
          design_matrix = X, method = "MCMC",
          Kmax = Kmax, nCores = nCores, splitSmallEM = TRUE)
```

Now let's explore the output based on the EM algorithm only. At first we retrieve the selected number of clusters according to ICL. Then we display the estimated clustering conditional on the selected value. Finally, we retrieve the estimated parameters (mixing proportions and coefficients of the mixture of multinomial logits) of the model.

```
# estimated number of clusters
> mlm_split$EM$estimated_K
[1] 2
> # estimated number of assigned observations per cluster
> table(mlm_split$EM$estimated_clustering)
  1   2
 68 182
> # estimated mixing proportions
> # estimated coefficients
> round(mlm_split$EM$all_runs[[2]]$beta,2)
, , 1

      [,1]  [,2]  [,3]
[1,] -0.07  0.05  9.78
[2,] -1.80  4.26  2.21
[3,]  0.01 -3.93  5.51
[4,] -0.03  3.78 -0.57
[5,] -0.04 -0.59 -0.01

, , 2

       [,1]  [,2]  [,3]
[1,]   0.16  0.01  4.71
[2,]  -0.02  0.00 -0.02
[3,] -22.73 -1.35  1.43
[4,]  -9.92  1.83 -0.13
[5,]   0.00 -1.73 -0.05
```

Note that in the last chunk of the output the rows correspond to multinomial categories (there are 6 categories in total so $j = 1, \ldots, 5$) and the columns correspond to covariates ($p = 1, \ldots, 3$), per cluster. For example, the estimate of the coefficient of the second multinomial category ($j = 2$) for the second covariate (so $p = 3$ because the model includes a constant term) for cluster 1 ($k = 1$) is equal to $\hat{\beta}_{kjp} = \hat{\beta}_{1,2,3} = 2.21$. The corresponding estimate for cluster 2 is equal to $\hat{\beta}_{kjp} = \hat{\beta}_{2,2,3} = -0.02$.

Let's explore the MCMC output now. We stress once again that the raw MCMC sample of the overfitting mixture is not directly interpretable due to label switching. Moreover, bear in mind that it also consists of the values of the empty mixture components which are not relevant for all practical purposes. These points are illustrated in Fig. 11, which displays the raw MCMC output for $\beta_{k,2,3}$ for all components $k = 1, 2, \ldots, 10$ of the overfitting mixture model. A careful inspection of this graph reveals that up to a switching of the labels the sampled values of 2 (among 10 components) are concentrated around the two horizontal dotted lines which correspond the estimates of the corresponding parameters according to the EM algorithm. The remaining values which are further away from the dotted lines correspond to the sampled values of this parameter for the remaining 8 empty mixture components (again up to switching of the labels).

Let's proceed now by inspecting the post-processed output according to the ECR algorithm. At first we can retrieve the estimated posterior distribution of the number of clusters (which correspond to the number of non-empty mixture components across the MCMC run) as well as the number of assigned observation per cluster, conditional on the value of the most probable number of clusters.

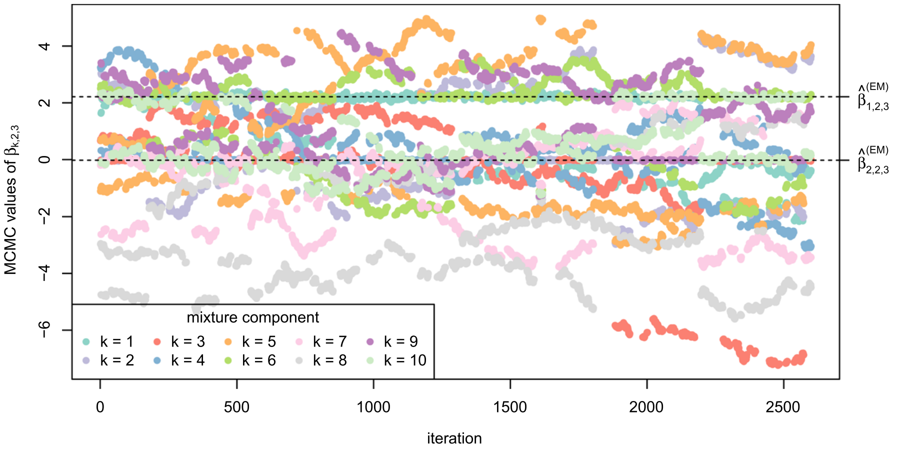


Fig. 11 Raw MCMC output (after an initial warm-up period) for the coefficients $\beta_{k,2,3}$ for component $k = 1, 2, \ldots, 10$ of the overfitting mixture model. The dotted lines correspond to the estimate of $\beta_{k,2,3}$ for $k = 1, 2$ when using a mixture model with $K = 2$ components

```
#estimated posterior distribution of the number of clusters
> mlm_split$MCMC_post_processed$nClusters_posterior
      2      3
0.9992 0.0008
# estimated number of assigned observations per cluster according to the MCMC sampler
> table(mlm_split$MCMC_post_processed$cluster)
  1   2
 68 182
```

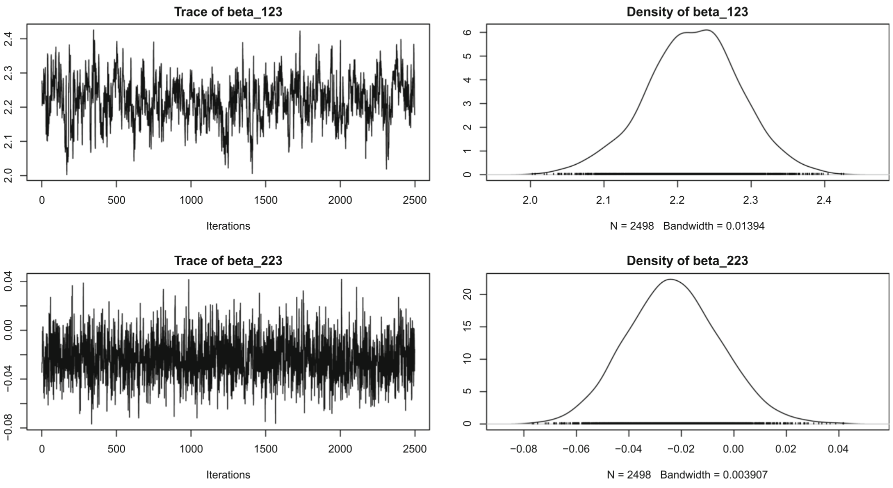


Fig. 12 Post-processed MCMC outputs for $\beta_{1,2,3}$ (top) and $\beta_{2,2,3}$ (bottom)

Now we concentrate on the post-processed output of the non-empty mixture components, so we essentially discard the sampled values of the remaining 8 empty mixture components. We can retrieve basic MCMC summaries using the `coda` package. For example let us retrieve MCMC summaries for the coefficients $\beta_{1,2,3}$ and $\beta_{2,2,3}$.

```
> beta_123 <- mlm_split$MCMC_post_processed$mcmc[[2]][[3]][,1,2]
> beta_223 <- mlm_split$MCMC_post_processed$mcmc[[2]][[3]][,2,2]
> betas <- as.mcmc(cbind(beta_123, beta_223))
> summary(betas)

Iterations = 1:2498
Thinning interval = 1
Number of chains = 1
Sample size per chain = 2498

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

            Mean     SD Naive SE Time-series SE
beta_123  2.2207 0.0643 0.001286       0.004478
beta_223 -0.0229 0.0178 0.000357       0.000434

2. Quantiles for each variable:

            2.5%    25%     50%     75%  97.5%
beta_123  2.0872  2.179  2.2221  2.2634 2.3436
beta_223 -0.0566 -0.035 -0.0235 -0.0114 0.0129
> plot(betas)
```

The last command produces Fig. 12, which displays the trace of the post-processed values of $\beta_{1,2,3}$ (left) along with the corresponding estimate of the marginal posterior distribution (right).

Of particular interest is also the matrix of posterior membership probabilities for each observation. The next chunk of code shows how one can retrieve these estimates according to output from the EM and the MCMC algorithm, respectively.

```
# estimated posterior membership probabilities according to the EM
    algorithm
> round(mlm_split$EM$all_runs[[mlm_split$EM$estimated_K]]$
    posteriorProbabilities, 2)
       [,1] [,2]
  [1,] 0.00 1.00
  [2,] 0.38 0.62
<...(+246 rows)...>
[249,] 1.00 0.00
[250,] 0.00 1.00
# estimated posterior membership probabilities according to the MCMC
    algorithm
> round(mlm_split$MCMC_post_processed$posteriorProbabilities, 2)
       [,1] [,2]
  [1,] 0.00 1.00
  [2,] 0.37 0.63
<...(+246 rows)...>
[249,] 1.00 0.00
[250,] 0.00 1.00
```

The previous example uses the parameter setup detailed in Section A. The user can modify these arguments by passing the desired input to the optional arguments `em_parameters` and `mcmc_parameters` of the `multinomialLogitMix()`

function. Both `em_parameters` and `mcmc_parameters` should be lists, consisting of the following entries.

Arguments of the `em_parameters` *list.*

`maxIter`	Maximum number of EM iterations. Default: 100
`emthreshold`	Positive real threshold for terminating the EM algorithm. The algorithm stops when the difference between two successive evaluations of the observed log-likelihood is less than this threshold. Default: 10^{-8}
`maxNR`	Maximum number of Newton-Raphson iterations. Default: 10
`tsplit`	Number of different starts that will be used within the small-EM scheme (this quantity refers to all schemes: random, split and shake). Default: 16
`msplit`	Number of iterations for each small-EM start. Default: 10
`split`	Boolean denoting whether the EM algorithm will use the split-small EM scheme. Default: TRUE. In the opposite case, the small-EM scheme will use only randomly selected initial values
`R0`	The initial value for the parameter R that controls the step-size of the ridge-stabilized Newton-Raphson scheme (see Equation (16)). Default: 0.1

Arguments of the `mcmc_parameters` *list.*

`tau`	initial value for the scale of the MALA proposal (positive, it corresponds to the parameter ν in Equation (23)). Default: 0.00035. This parameter is adjusted in the initial (warm-up) period of the sampler in order to achieve the desirable acceptance rate of the MALA proposal
`nu2`	the variance of the normal prior distribution of the logit coefficients (the parameter τ^2 in Equation (20)). Default: 100
`mcmc_cycles`	Total number of MCMC cycles (after the initial warm-up) period of the sampler. Default: 2600. At the end of each MCMC cycle a swap between chains is attempted
`iter_per_cycle`	Number of MCMC iterations per cycle. Default: 20
`nChains`	Number of MCMC chains that run in parallel. Each chain uses a different prior distribution of the mixing proportions. The inference is based on the first chain

`dirPriorAlphas`	The concentration parameter of the Dirichlet prior distributions per chain (see Equation (28)). It should be a vector with length equal to `nChains`. The default is: `5 * exp((seq(2, 14, length = nChains - 1)))/100)/(200)`, see Equations (29) and (30)
`warm_up`	Initial warm-up period of the sampler, in order to adaptively tune the scale of the MALA proposal. Default: 48000
`checkAR`	Number of iterations required in order to adjust the scale of the proposal in MALA mechanism during the initial warm-up phase of the sampler. Default: 500
`ar_low`	Lowest threshold for the acceptance rate of the MALA proposal. Default: 0.15
`ar_up`	Highest threshold for the acceptance rate of the MALA proposal. Default: 0.25
`burn`	Number of MCMC cycles that will be discarded as burn-in. Default: 100
`withRandom`	Boolean value indicating whether or not to apply a random permutation in the supplied starting values for each chain. Default: true

Appendix C: Further simulation results

Main simulation study

Figure 13 displays a summary regarding the clustering accuracy versus the number of covariates (including constant) in the simulated datasets of Sect. 4.1, that is, $P \in \{2, 3, 4\}$ for EM(8, 8, 8), MCMC-EM (1) and MCMC-EM c(100). We observe that for both indices (estimation of the number of clusters and partition agreement as measured by the adjusted Rand index), the clustering accuracy tends to decrease as the number of covariates increase. This effect is clearly illustrated in the smaller sample sizes (n). However, as n gets larger the impact of the number of covariates becomes smaller.

Figure 14 displays the difference between the estimated and true value of the numbers of clusters ($\hat{K} - K$) (left) and the adjusted Rand index (right) for each value of K (horizontal axis), stratified for all different values of sample size ($n \in \{125, 250, 500, 1000\}$). It is evident that as the number of clusters increases, overestimates of the number of clusters occur more often and this effect is more severe for small sample sizes. Once again we note that the MCMC sampler with small prior variance (MCMC-EM (1)) outperforms the remaining implementations.

Next we are concerned with the accuracy of point estimates $\widehat{\beta}_{kjp}$, that is, the coefficient value at the last iteration for the EM implementation and the estimate of the

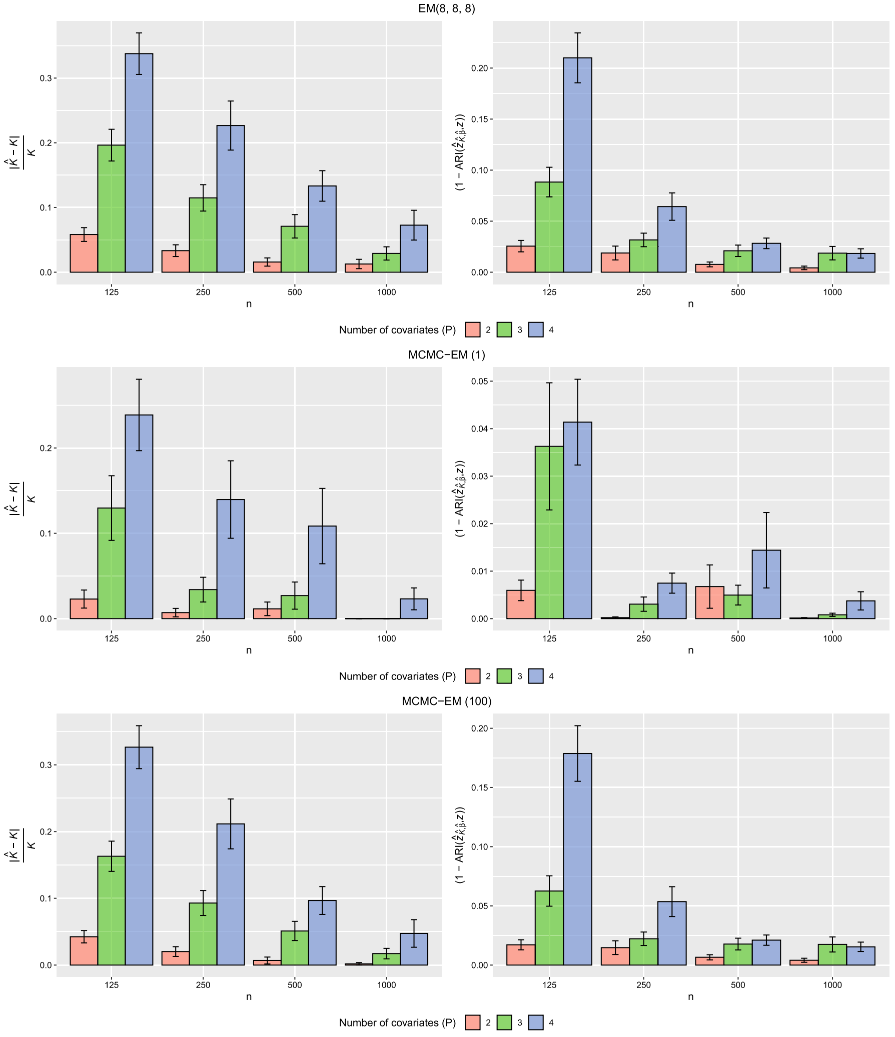


Fig. 13 Clustering accuracy for EM (up) and MCMC (middle and bottom) when taking into account sample size (n) and the number of covariates P (including constant term) for the simulation study presented in Sect. 4.1 of the manuscript

posterior mean after reordering the MCMC output in order to deal with label switching for the MCMC implementation. We generated 100 synthetic datasets with $n = 250$ observations and 100 datasets with $n = 500$, considering $K = 4$ clusters, $J + 1 = 6$ multinomial categories and $P = 3$ covariates (including constant term). In each case, the number of components K was set equal to the true number of clusters, that is, $K = 4$. All other parameters were generated as described in Sect. 4.1. In order to meaningfully compare β_{kjp} with the corresponding point estimates $\widehat{\beta}_{kjp}$ (arising either from EM(8, 8, 8) or from MCMC-EM (100)), we relabelled the resulting point estimates

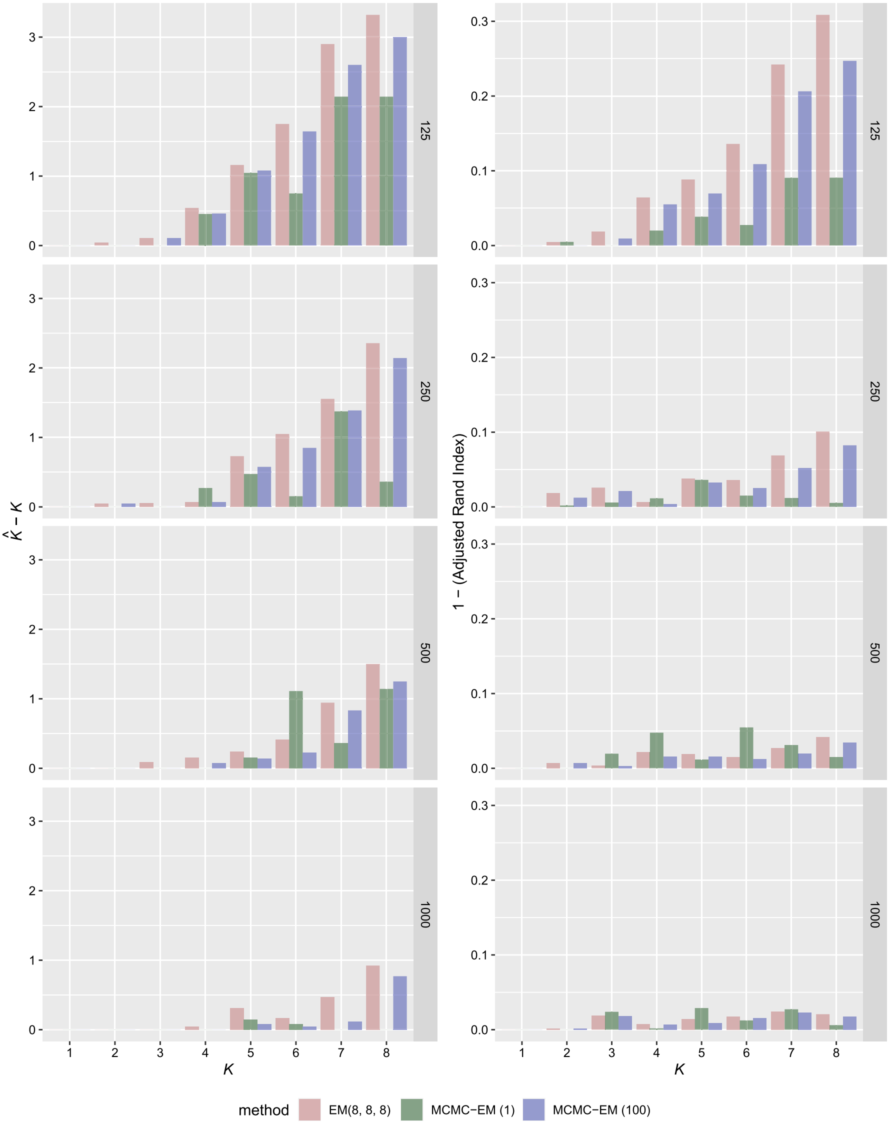


Fig. 14 Detailed results for our three best performing methods in the simulation study of Sect. 4.1. Left: mean difference of the estimated number of clusters ($\widehat{K}$) from the corresponding true value (K). Right: mean values of $1 - \text{ARI}$ between the model corresponding to the estimated number of clusters and the true partition

by applying the ECR algorithm (Papastamoulis and Iliopoulos 2010), by considering that the pivot allocation vector (required in the ECR algorithm) is set to the true partition. This procedure ensures the labelling between β_{kjp} and $\widehat{\beta}_{kjp}$ (for $j = 1, \ldots, J$; $p = 1, \ldots, P$) is consistent for all $k = 1, \ldots, K$, and it has no impact at the quality of the estimates themselves.

Figure 15 displays the estimated Mean Absolute Error (MAE) between the point estimate $\widehat{\beta}_{kjp}$ and the corresponding true value β_{kjp}, for $k = 1, \ldots, K$, $j = 1, \ldots, J$ and $p = 1, \ldots, P$. As expected, the estimated MAEs are smaller on average as the sample (n) increases. Observe that when $n = 250$ the MAEs become smaller as k increases. This is of course due to the fact that in our simulation study (see Sect. 4.1) the mixing proportions are generated according to $\pi_k \propto k$, for $k = 1, \ldots, K$. Thus, in our

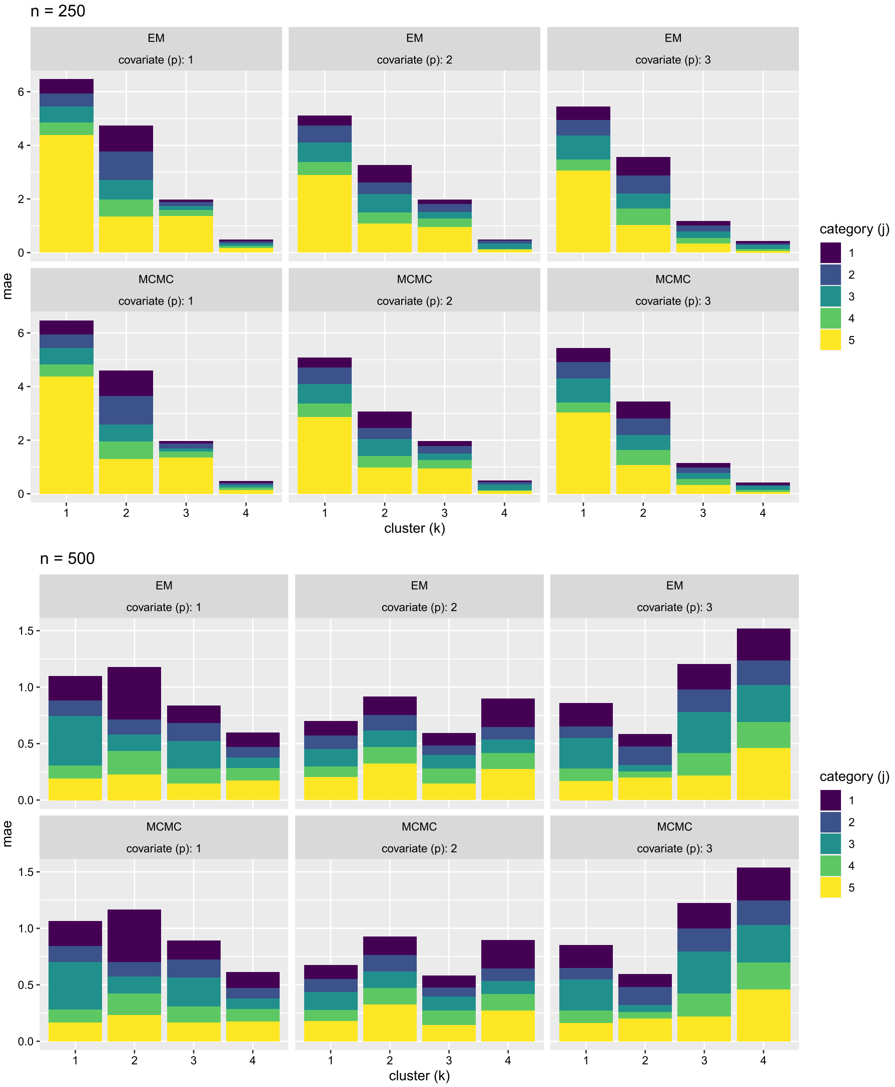


Fig. 15 Estimated mean absolute error (mae) of parameter estimates $\widehat{\beta}_{kjp}$ considering 100 synthetic datasets with $K = 4$ clusters, $P = 3$ covariates (including constant term) and $J + 1 = 6$ multinomial categories, when the sample size is equal to $n = 250$ (top) and $n = 500$ (bottom)

4-cluster scenario we have that $\pi_1 = 0.1, \pi_2 = 0.2, \pi_3 = 0.3$ and $\pi_4 = 0.4$. Naturally, the point estimates are less accurate in cases of very small clusters, a behaviour which is vividly illustrated when $n = 250$. However, when $n = 500$ we do not spot any systematic pattern in the resulting MAEs across clusters.

Further simulations and comparison with `flexmix`

In this Section we compare the proposed methods with the popular `R` package `flexmix` (Leisch 2004; Grün and Leisch 2007, 2008a) and we also explore the impact of the average number of multinomial replicates $\bar{s} = \sum_{i=1}^{n} S_i/n$. In the main simulation study of Sect. 4.1, the number of multinomial replicates (S_i) per observation is drawn from a negative binomial distribution $\mathcal{NB}(r, p)$ with number of trials equal to $r = 20$ and $p = 0.025$. This yields a potentially large of multinomial replicates: the average number is equal to 781. In order to assess the sensitivity of our results to the number of multinomial replicates we consider that r varies in the set $\{2.5, 5, 10, 20\}$ and we use the $\mathcal{NB}(r, 0.025)$ distribution to simulate S_i (in case that the generated number is 0 we set it to 1). It follows that the average value of S_i is approximately 100, 200, 400 and 800, respectively (more precisely, the average values in our simulated datasets are 97.5, 195, 390 and 781, respectively).

For this task we considered the following simulation scenario. In all cases the number of multinomial categories was set equal to $J+1 = 6$. The number of covariates (including constant term) was set equal to $P = 3$. The true values of the regression coefficients are generated as in Sect. 4.1. The number of clusters varied between $1 \leq K \leq 5$. We considered sample sizes of $n = 250$ and $n = 500$ observations. We generated 5 synthetic datasets for each unique combination of K (number of clusters), level of the average number of multinomial replicates $\bar{s}$ and sample size (n), resulting to 200 datasets in total.

Next, we fitted mixtures of multinomial logistic regressions considering the proposed methodology, as well as the EM implementation in the `R` package `flexmix`. The configuration of the EM algorithm for our proposed method was EM(8, 8, 8), that is, an EM algorithm with 8 splits, 8 shakes and 8 random starts of small EM for each possible value of the number of components. Then we have used the selected model in order to initialize a run of a Bayesian overfitting mixture model. The prior variance of the regression coefficients is equal to $\nu = 100$ (vague prior distribution) using 8 chains in total. Finally, we have run `flexmix` repeatedly considering 24 random starts, for each possible value of the number of components, using the `stepFlexmix` function with options: `k=1:Kmax` (where `Kmax` denotes the maximum number of components—see next paragraph), `nrep = 24`, `control = list(minprior = 0)` and `model = flexmix::FLXMRmultinom()`.

In all cases, the maximum number of mixture components was set equal to $K_{\max} = K+2$, where K denotes the true value of the number of clusters for each case. This upper bound of the number of components is much more informative regarding the number of clusters than the one in our main simulation study of Sect. 4.1 (where $K_{\max} = 20$). The reason for choosing such a value was mainly to speed-up the computing time in

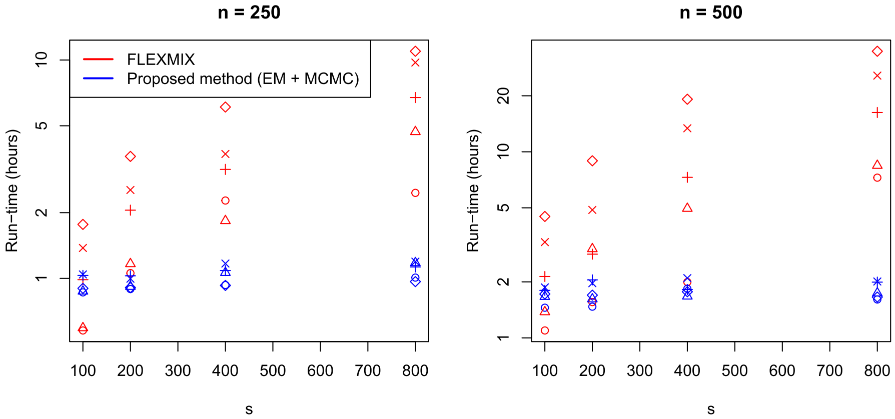


Fig. 16 Run-time (averaged across 5 runs) comparison of the proposed methods (EM(8, 8, 8) and MCMC) against the EM algorithm of `flexmix` under 24 randomly selected starting values. The *x*-axis indicates the average value of multinomial replicates (rounded to the nearest multiple of 100). The *y*-axis is on log-scale. All times correspond to computations in one single CPU core. A different symbold is used to denote the maximum number of fitted mixture components: ∘(3), △(4), +(5), ×(6), ⋄(7)

`flexmix`, where in most cases is significantly elevated (see Fig. 16) compared to the proposed implementation.

The resulting estimates are summarized in Fig. 17. We conclude that in all cases the estimation becomes more challenging when the average number of multinomial replicates is smaller. Note also that our proposed methods (EM and MCMC) are better than `flexmix` in terms of estimation of the number of clusters as well as classification accuracy under the adjusted Rand index.

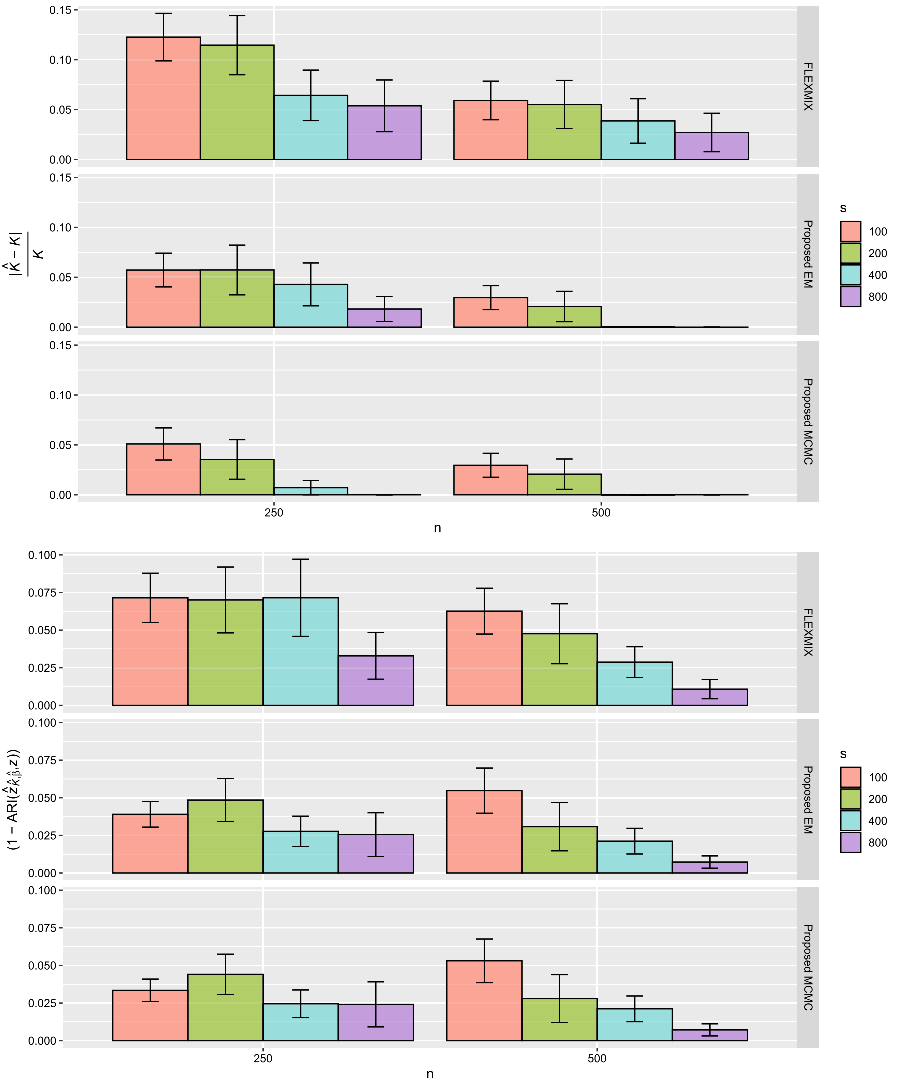


Fig. 17 Comparison of the proposed methods (EM and MCMC) against `flexmix`. The average value of multinomial replicates is shown in the legend (s), after rounding to the nearest multiple of 100

References

Albert JH, Chib S (1993) Bayesian analysis of binary and polychotomous response data. J Am Stat Assoc 88(422):669–679. https://doi.org/10.1080/01621459.1993.10476321

Altekar G, Dwarkadas S, Huelsenbeck JP et al (2004) Parallel Metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. Bioinformatics 20(3):407–415. https://doi.org/10.1093/bioinformatics/btg427

Basford K, Greenway D, McLachlan G et al (1997) Standard errors of fitted component means of normal mixtures. Comput Stat 12(1):1–18

Baudry JP, Celeux G (2015) EM for mixtures. Stat Comput 25(4):713–726

Benaglia T, Chauveau D, Hunter DR et al (2009) mixtools: an R package for analyzing finite mixture models. J Stat Softw 32(6):1–29

Biernacki C, Celeux G, Govaert G (2000) Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Trans Pattern Anal Mach Intell 22(7):719–725

Biernacki C, Celeux G, Govaert G (2003) Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. Comput Stat Data Anal 41(3–4):561–575

Blischke WR (1964) Estimating the parameters of mixtures of binomial distributions. J Am Stat Assoc 59(306):510–528. https://doi.org/10.1080/01621459.1964.10482176

Bouguila N (2008) Clustering of count data using generalized Dirichlet multinomial distributions. IEEE Trans Knowl Data Eng 20(4):462–474. https://doi.org/10.1109/TKDE.2007.190726

Celeux G, Hurn M, Robert CP (2000) Computational and inferential difficulties with mixture posterior distributions. J Am Stat Assoc 95(451):957–970

Chen L, Wang W, Zhai Y et al (2020) Single-cell transcriptome data clustering via multinomial modeling and adaptive fuzzy k-means algorithm. Front Genet 11:295

Choi HM, Hobert JP (2013) The Polya-Gamma Gibbs sampler for Bayesian logistic regression is uniformly ergodic. Electron J Stat 7:2054–2064

Crockett JB, Chernoff H et al (1955) Gradient methods of maximization. Pac J Math 5(1):33–50

Dean N, Raftery AE (2010) Latent class analysis variable selection. Ann Inst Stat Math 62:11–35

Dehouche N (2020) Dataset on usage and engagement patterns for Facebook live sellers in Thailand. Data Brief 30:105,661. https://doi.org/10.1016/j.dib.2020.105661

Dellaportas P, Smith AF (1993) Bayesian inference for generalized linear and proportional hazards models via Gibbs sampling. J R Stat Soc Ser C (Appl Stat) 42(3):443–459

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B (Methodol) 39(1):1–22

Durante D, Canale A, Rigon T (2019) A nested expectation--maximization algorithm for latent class models with covariates. Stat Probab Lett 146:97–103

Eddelbuettel D (2013) Seamless R and C++ integration with Rcpp. Springer, New York. https://doi.org/10.1007/978-1-4614-6868-4 (**iSBN 978-1-4614-6867-7**)

Eddelbuettel D, Balamuta JJ (2018) Extending extitR with extitC++: A Brief Introduction to extitRcpp. Am Stat 72(1):28–36. https://doi.org/10.1080/00031305.2017.1375990

Eddelbuettel D, François R (2011) Rcpp: seamless R and C++ integration. J Stat Softw 40(8):1–18. https://doi.org/10.18637/jss.v040.i08

Eddelbuettel D, Sanderson C (2014) Rcpparmadillo: accelerating r with high-performance C++ linear algebra. Comput Stat Data Anal 71:1054–1063. https://doi.org/10.1016/j.csda.2013.02.005

Fop M, Murphy TB (2018) Variable selection methods for model-based clustering. Stat Surv 12(none):18–65. https://doi.org/10.1214/18-SS119

Fraley C, Raftery A, Wehrens R (2005) Incremental model-based clustering for large datasets with small clusters. J Comput Graph Stat 14(3):529–546

Frühwirth-Schnatter S (2006) Finite mixture and Markov switching models, vol 425. Springer, Berlin

Frühwirth-Schnatter S, Celeux G, Robert CP (2019) Handbook of mixture analysis. CRC Press, Boca Raton

Frühwirth-Schnatter S, Malsiner-Walli G, Grün B (2021) Generalized mixtures of finite mixtures and telescoping sampling. Bayesian Anal 16(4):1279–1307

Frühwirth-Schnatter S, Frühwirth R (2010) Data augmentation and mcmc for binary and multinomial logit models. In: Statistical modelling and regression structures. Springer, pp 111–132

Galindo Garre F, Vermunt JK (2006) Avoiding boundary estimates in latent class analysis by Bayesian posterior mode estimation. Behaviormetrika 33:43–59

Geyer CJ (1991) Markov chain Monte Carlo maximum likelihood. In: Proceedings of the 23rd symposium on the interface, interface foundation, Fairfax Station, Va, pp 156–163

Geyer CJ, Thompson EA (1995) Annealing Markov chain Monte Carlo with applications to ancestral inference. J Am Stat Assoc 90(431):909–920. https://doi.org/10.1080/01621459.1995.10476590

Girolami M, Calderhead B (2011) Riemann manifold Langevin and Hamiltonian Monte Carlo methods. J R Stat Soc Ser B (Stat Methodol) 73(2):123–214

Goldfeld SM, Quandt RE, Trotter HF (1966) Maximization by quadratic hill-climbing. Econom J Econom Soc 34:541–551

Govaert G, Nadif M (2007) Clustering of contingency table and mixture model. Eur J Oper Res 183(3):1055–1066

Gramacy RB, Polson NG (2012) Simulation-based regularized logistic regression. Bayesian Anal 7(3):567–590. https://doi.org/10.1214/12-BA719

Green PJ (1995) Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika 82(4):711–732

Grün B, Leisch F (2007) Fitting finite mixtures of generalized linear regressions in R. Comput Stat Data Anal 51(11):5247–5252. https://doi.org/10.1016/j.csda.2006.08.014

Grün B, Leisch F (2008) FlexMix version 2: finite mixtures with concomitant variables and varying and constant parameters. J Stat Softw 28(4):1–35. https://doi.org/10.18637/jss.v028.i04

Grün B, Leisch F (2008) Identifiability of finite mixtures of multinomial logit models with varying and fixed effects. J Classif 25(2):225–247

Grün B, Leisch F (2004) Bootstrapping finite mixture models. NA

Held L, Holmes CC (2006) Bayesian auxiliary variable models for binary and multinomial regression. Bayesian Anal 1(1):145–168

Hennig C (2000) Identifiablity of models for clusterwise linear regression. J Classif 17(2):273

Holmes CC, Held L et al (2006) Bayesian auxiliary variable models for binary and multinomial regression. Bayesian Anal 1(1):145–168

Hurn M, Justel A, Robert CP (2003) Estimating mixtures of regressions. J Comput Graph Stat 12(1):55–79

Jamshidian M, Jennrich RI (2000) Standard errors for EM estimation. J R Stat Soc Ser B (Stat Methodol) 62(2):257–270

Jasra A, Holmes CC, Stephens DA (2005) Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. Statist Sci 20(1):50–67. https://doi.org/10.1214/088342305000000016

Jorgensen M (2004) Using multinomial mixture models to cluster internet traffic. Aust N Z J Stat 46(2):205–218. https://doi.org/10.1111/j.1467-842X.2004.00325.x

Karlis D, Xekalaki E (2003) Choosing initial values for the EM algorithm for finite mixtures. Comput Stat Data Anal 41(3–4):577–590

Leisch F (2004) FlexMix: a general framework for finite mixture models and latent class regression in R. J Stat Softw 11(8):1–18. https://doi.org/10.18637/jss.v011.i08

Li J, Bioucas-Dias JM, Plaza A (2011) Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields. IEEE Trans Geosci Remote Sens 50(3):809–823

Louis TA (1982) Finding the observed information matrix when using the EM algorithm. J R Stat Soc Ser B (Methodol) 44(2):226–233

Malsiner-Walli G, Frühwirth-Schnatter S, Grün B (2016) Model-based clustering based on sparse finite gaussian mixtures. Stat Comput 26(1–2):303–324

Marin JM, Mengersen K, Robert C (2005) Bayesian modelling and inference on mixtures of distributions. Handb Stat 25:459–507. https://doi.org/10.1016/S0169-7161(05)25016-2

Maugis C, Celeux G, Martin-Magniette ML (2009) Variable selection for clustering with gaussian mixture models. Biometrics 65(3):701–709

McLachlan GJ, Peel D, Basford KE, et al (1999) The EMMIX software for the fitting of mixtures of normal and t-components. J Stat Softw 4(2)

McLachlan GJ, Peel D (2004) Finite mixture models. Wiley, New York

Meng XL, Rubin DB (1991) Using EM to obtain asymptotic variance-covariance matrices: the SEM algorithm. J Am Stat Assoc 86(416):899–909

Morel JG, Nagaraj NK (1993) A finite mixture distribution for modelling multinomial extra variation. Biometrika 80(2):363–371

Nemeth C, Fearnhead P (2021) Stochastic gradient Markov chain monte Carlo. J Am Stat Assoc 116(533):433–450. https://doi.org/10.1080/01621459.2020.1847120

Nowicka M, Robinson MD (2016) DRIMSeq: a Dirichlet-multinomial framework for multivariate count outcomes in genomics. F1000Research 5

Papastamoulis P (2016) label. switching: an R package for dealing with the label switching problem in MCMC outputs. J Stat Softw 69(1):1–24

Papastamoulis P (2018) Overfitting Bayesian mixtures of factor analyzers with an unknown number of components. Comput Stat Data Anal 124:220–234. https://doi.org/10.1016/j.csda.2018.03.007

Papastamoulis P (2020) Clustering multivariate data using factor analytic Bayesian mixtures with an unknown number of components. Stat Comput 30(3):485–506

Papastamoulis P, Iliopoulos G (2010) An artificial allocations based solution to the label switching problem in Bayesian analysis of mixtures of distributions. J Comput Graph Stat 19:313–331

Papastamoulis P, Martin-Magniette ML, Maugis-Rabusseau C (2016) On the estimation of mixtures of Poisson regression models with large number of components. Comput Stat Data Anal 93:97–106

Polson NG, Scott JG, Windle J (2013) Bayesian inference for logistic models using Pólya-gamma latent variables. J Am Stat Assoc 108(504):1339–1349

Portela J (2008) Clustering discrete data through the multinomial mixture model. Commun Stat-Theory Methods 37(20):3250–3263

Rau A, Maugis-Rabusseau C, Martin-Magniette ML et al (2015) Co-expression analysis of high-throughput transcriptome sequencing data with Poisson mixture models. Bioinformatics 31(9):1420–1427. https://doi.org/10.1093/bioinformatics/btu845

Richardson S, Green PJ (1997) On Bayesian analysis of mixtures with an unknown number of components (with discussion). J R Stat Soc Ser B (Stat Methodol) 59(4):731–792

Roberts GO, Rosenthal JS (1998) Optimal scaling of discrete approximations to Langevin diffusions. J R Stat Soc Se B (Stat Methodol) 60(1):255–268. https://doi.org/10.1111/1467-9868.00123

Roberts GO, Tweedie RL (1996) Exponential convergence of Langevin distributions and their discrete approximations. Bernoulli 2(4):341–363

Rousseau J, Mengersen K (2011) Asymptotic Behaviour of the posterior distribution in overfitted mixture models. J R Stat Soc Ser B (Stat Methodol) 73(5):689–710

Schwarz G et al (1978) Estimating the dimension of a model. Ann Stat 6(2):461–464

Stephens M (2000) Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods. Ann Stat 28:40–74

Teicher H (1963) Identifiability of finite mixtures. Ann Math Stat 34(4):1265–1269. https://doi.org/10.1214/aoms/1177703862

Titterington DM, Smith AF, Makov UE (1985) Statistical analysis of finite mixture distributions. Wiley, New York

van Havre Z, White N, Rousseau J et al (2015) Overfitting Bayesian mixture models with an unknown number of components. PLoS ONE 10(7):1–27

Vávra J, Komárek A, Grün B, et al (2022) Clusterwise multivariate regression of mixed-type panel data. Technical Report

Wade S, Ghahramani Z (2018) Bayesian cluster analysis: point estimation and credible balls (with discussion). Bayesian Anal 13(2):559–626. https://doi.org/10.1214/17-BA1073

Wongkitrungrueng A, Dehouche N, Assarut N (2020) Live streaming commerce from the sellers' perspective: implications for online relationship marketing. J Market Manag 36(5–6):488–518

Yau C, Holmes C (2011) Hierarchical Bayesian nonparametric mixture models for clustering with variable relevance determination. Bayesian Anal (Online) 6(2):329

Yu P, Shaw CA (2014) An efficient algorithm for accurate computation of the Dirichlet-multinomial log-likelihood function. Bioinformatics 30(11):1547–1554

Zamzami N, Bouguila N (2020) Sparse count data clustering using an exponential approximation to generalized Dirichlet multinomial distributions. IEEE Trans Neural Netw Learn Syst 33(1):89–102

Zuanetti DA, Milan LA (2017) A generalized mixture model applied to diabetes incidence data. Biom J 59(4):826–842