Machine Learning with Applications 15 (2024) 100528

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Parsimonious Bayesian model-based clustering with dissimilarities

Samuel Morrissette *, Saman Muthukumarana, Maxime Turgeon

Department of Statistics, University of Manitoba, 66 Chancellors Cir, Winnipeg, MB R3T 2N2, Canada

ARTICLE INFO

Keywords:
Bayesian clustering
Model-based clustering
Markov chain Monte Carlo
Gibbs sampling
Measurement error

ABSTRACT

Clustering techniques are used to group observations and discover interesting patterns within data. Model-based clustering is one such method that is often an attractive choice due to the specification of a generative model for the given data and the ability to calculate model-selection criteria, which is in turn used to select the number of clusters. However, when only distances between observations are available, model-based clustering can no longer be used, and heuristic algorithms without the aforementioned advantages are usually used instead. As a solution, Oh and Raftery (2007) suggest a Bayesian model-based clustering method (named BMCD) that only requires a dissimilarity matrix as input, while also accounting for the measurement error that may be present within the observed data. In this paper, we extend the BMCD framework by proposing several additional models, alternative model selection criteria, and strategies for reducing computing time of the algorithm. These extensions ensure that the algorithm is effective even in high-dimensional spaces and provides a wide range of choices to the practitioner that can be used with a variety of data. Additionally, a publicly available software implementation of the algorithm is provided as a package in the R programming language.

1. Introduction

Clustering is a process in which observations of a data set are grouped together based on their similarity to one another. As a result of partitioning the data set into distinct clusters, interesting patterns and relationships can be discovered within the data set.

The Gaussian mixture model (GMM) is one such popular clustering method due to its foundation in probability theory. Fitting a GMM to data requires, in part, the computation of means (i.e. centroids) of the data set. Therefore, in circumstances where the data of interest consists of distances between objects, GMM clustering cannot be used. For instance, the challenge of working with this type of data commonly arises in the fields of psychology (Bimler et al. 2004, Fitzgerald and Hubert 1985, Jaworska and Chupetlovska-Anastasova 2009), genomics (Chen and Meltzer 2005, Kim et al. 2019), market research (Bijmolt et al., 2020), and sociology (McEntee, 2004). Consequently, alternative clustering techniques that only require distances between observations as input, such as k-medoids, hierarchical clustering, or density-based clustering methods like DBSCAN (Ester et al., 1996), are usually employed instead. However, in doing so, the advantages of GMM clustering are lost. These advantages include interpretability due to specification of a generative probability distribution, as well as the ability to calculate model selection criteria such as the Bayesian Information Criterion (BIC) (Schwarz, 1978) or Integrated Complete Likelihood (ICL) (Biernacki et al., 2000).

When only distances between observations are available, it is still possible to use GMM clustering by first applying multidimensional scaling (MDS) to the data set. MDS takes as its input a dissimilarity matrix and represents observations in p-dimensional space by assigning them in such a way that the distances between them are preserved according to the given distance matrix. MDS is often used in the context of clustering only for the purpose of visualizing clustering results in a lower-dimensional space. However, the merit of conducting clustering on the actual output of MDS is not well-studied. Furthermore, selecting an appropriate number of dimensions to retain from MDS represents another challenge. Since the result of clustering is dependent on the number of dimensions retained, choosing an appropriate dimension for MDS is vital to the success of the subsequent clustering procedure.

Oh and Raftery (2001) describe a Bayesian version of MDS (named BMDS) that includes a selection criteria called the Multidimensional Scaling Information Criterion (MDSIC). The MDSIC criterion is used to estimate the effective dimension of the data set and select an appropriate number of dimensions to be retained from the BMDS method.

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: morris50@myumanitoba.ca (S. Morrissette), saman.muthukumarana@umanitoba.ca (S. Muthukumarana), max.turgeon@umanitoba.ca (M. Turgeon).

https://doi.org/10.1016/j.mlwa.2024.100528
Received 10 October 2023; Received in revised form 16 January 2024; Accepted 16 January 2024
Available online 23 January 2024

Furthermore, Oh and Raftery (2007) describe a two-stage clustering method that first utilizes BMDS and then fits a GMM using Markov Chain Monte Carlo (MCMC) methods. In this manner, Bayesian methods can be used to fit a GMM, even when only distances between observations are observed. The authors named the resulting method Bayesian Model-based Clustering with Dissimilarities (BMCD).

A particular advantage of BMCD is that it can estimate and account for the measurement error that may be present in the distances between data. Measurement error is the difference between the recorded distances and their true, but unobserved, value. Measurement error can occur for a variety of reasons, including inaccuracies in the tools used to record observations (i.e. systematic error), or simply random error between measurements. By incorporating the estimation of measurement error within the data set, BMCD can result in more accurate clusters than other clustering algorithms that do not account for this source of error. Despite the advantages offered by BMCD, no software implementations of the method were ever released. Consequently, no further analysis into the merit of BMCD has been published.

In this paper, we extend the work of Oh and Raftery by proposing five additional Gaussian mixture models to the original one given in their 2007 paper. In particular, these models differ in the shape of clusters that they are able to capture, and they may also be more effective in high-dimensional problems by limiting the number of free parameters that have to be estimated. This approach to constraining the covariance matrices in Gaussian mixture models was first proposed by Banfield and Raftery (1993). For each of the given models, we provide the Gibbs sampling steps as well as suggest strategies for fitting and selecting an appropriate model. To fit these models to given data, a software implementation of BMCD has been developed for as a freely available package in the R software (R Core Team, 2023). The package is available in a GitHub repository located at https://github.com/SamMorrissette/BMCDcpp. This package is able to improve computational efficiency of the method by using parallel computing to fit several candidate models at the same time. Using this software, we are able to conduct simulation studies to investigate the performance of the BMCD models under a variety of conditions. These simulations show that, even in higher dimensions, the proposed BMCD models may be more effective than alternative clustering methods such as k-means or a GMM fit using the Expectation–Maximization (EM) method (Dempster et al., 1977). We then demonstrate that BMCD can be useful in real-world applications by performing clustering on the Wisconsin Breast Cancer data set.

2. Methodology

2.1. Review

This section provides a review of the BMCD method introduced by Oh and Raftery (2007). One of the main goals of BMCD is to provide an estimate of the configuration of the observations in p-dimensional Euclidean space. Let the observation configuration be denoted by a matrix, $\mathbf{X}$, in which each row, $\mathbf{x}_i$, is a p-dimensional vector that corresponds to the observation's position. Then, $\mathbf{x}_i$ is assumed to be generated from a mixture of p-dimensional multivariate normal distributions:

$$\mathbf{x}_i \sim \sum_{k=1}^{G} \epsilon_k \, \mathcal{N}_p\left(\mu_k, \mathbf{\Sigma_k}\right),$$

where $\sum_{k=1}^{G} \epsilon_k = 1$ and $\epsilon_k > 0$ for all k. However, $\mathbf{X}$ is an unobserved variable in the model, and only the distances between observations, d_{ij} are observed.

An advantage of the BMCD method is the ability to account for the measurement error that may occur when these distances are recorded. For instance, let δ_{ij} be the true, unobserved, distance between two distinct p-dimensional observations $\mathbf{x}_i$ and $\mathbf{x}_j$,

$$\delta_{ij} = \sqrt{\sum_{k=1}^{p} \left(x_{ik} - x_{jk}\right)^2}. \tag{1}$$

Then, the observed distance is assumed to be generated from a truncated normal distribution with mean δ_{ij} and variance σ^2, the latter of which controls the amount of measurement error present:

$$d_{ij} \sim \mathcal{N}\left(\delta_{ij}, \sigma^2\right) I\left(d_{ij} > 0\right), \tag{2}$$

where $I(\cdot)$ denotes the indicator function. A key advantage of BMCD is that it is able to estimate the object configuration, $\mathbf{X}$, while simultaneously estimating the measurement error, σ^2.

Another main objective of BMCD is to identify the probability that each component generated the observations. Latent variables are introduced, denoted z_i, which take on a value of k if observation i was generated by component k. The addition of these latent variables into the model greatly simplifies calculations since

$$\mathbf{x}_i | z_i = k \sim \mathcal{N}_p\left(\mu_k, \mathbf{\Sigma_k}\right).$$

Usually, the Expectation–Maximization (EM) algorithm is used to fit a GMM to the given data. However, the BMCD method takes an alternative approach by using Bayesian methods to fit the GMM, which allows for the specification of prior distributions over the model parameters. If we let $\theta_k = (\mu_k, \mathbf{\Sigma_k})$, $\boldsymbol{\epsilon}$ be a G-element vector of mixing weights, and assuming that the distances between observations are modelled according to Eq. (2), then BMCD uses the following prior distributions:

$$\begin{aligned}
\mathbf{x}_i | z_i = k, \theta_k &\sim \mathcal{N}_p\left(\mu_k, \mathbf{\Sigma_k}\right), \\
z_i | \boldsymbol{\epsilon} &\sim \text{Cat}(\epsilon_1, \ldots, \epsilon_G), \\
\sigma^2 &\sim IG(a, b), \\
(\epsilon_1, \ldots, \epsilon_G) &\sim \text{Dir}(1, \ldots, 1), \\
\theta_k &\sim NIW(\mu_{k0}, 1, \Psi_0, \nu_0) \quad \text{for } k = 1, \ldots, G,
\end{aligned}$$

where NIW is the Normal-Inverse-Wishart distribution, Cat is the categorical distribution, Dir is the Dirichlet distribution, and IG is the Inverse-Gamma distribution.

Next, Gibbs sampling (Geman & Geman, 1984) is used to infer the model parameters. To employ Gibbs sampling, the full conditional posterior distributions must be derived and are given by Oh and Raftery (2007):

$$\begin{aligned}
\pi\left(\mathbf{x}_i | \sigma^2, \mathbf{z}, \boldsymbol{\epsilon}, \boldsymbol{\theta}, d_{i\cdot}\right) &\propto \prod_{i>j} \left[\Phi\left(\frac{\delta_{ij}}{\sigma}\right)\right] \\
&\quad \times \exp\left[\frac{-1}{2}(\mathbf{x}_i - \mu_k)^T \mathbf{\Sigma_k}^{-1}(\mathbf{x}_i - \mu_k) - \frac{1}{2\sigma^2}\sum_{i>j}(\delta_{ij} - d_{ij})^2\right], \\
\pi\left(\sigma^2 | \mathbf{X}, \mathbf{z}, \boldsymbol{\epsilon}, \boldsymbol{\theta}, d_{i\cdot}\right) &\propto (\sigma^2)^{(-n(n-1)/4+a+1)} \\
&\quad \times \exp\left[-\frac{1}{2\sigma^2}\left(\sum_{i>j}(\delta_{ij} - d_{ij})^2 + 2b\right) - \sum_{i>j}\log\Phi\left(\frac{\delta_{ij}}{\sigma}\right)\right], \\
\pi\left(z_i | \mathbf{X}, \sigma^2, \boldsymbol{\epsilon}, \boldsymbol{\theta}, d_{i\cdot}\right) &\sim \text{Cat}\left(\gamma_1, \ldots, \gamma_G\right), \\
\pi\left(\boldsymbol{\epsilon} | \mathbf{X}, \sigma^2, \mathbf{z}, \boldsymbol{\theta}, d_{i\cdot}\right) &\sim \text{Dir}(n_1 + 1, \ldots, n_G + 1), \\
\pi\left(\theta_k | \mathbf{X}, \sigma^2, \mathbf{z}, \boldsymbol{\epsilon}, d_{i\cdot}\right) &\sim NIW\left(\frac{\mu_{k0} + n_k\bar{\mathbf{x}}_k}{n_k + 1}, n_k + 1, \nu_0 + n_k, \right. \\
&\quad \left. \Psi_0 + \mathbf{S_k} + \frac{n_k}{n_k + 1}(\bar{\mathbf{x}}_k - \mu_{k0})(\bar{\mathbf{x}}_k - \mu_{k0})^T\right),
\end{aligned}$$

where $\gamma_k = \dfrac{\epsilon_k \phi(\mathbf{x}_i | \theta_k)}{\sum_{j=1}^{G} \epsilon_j \phi(\mathbf{x}_i | \theta_j)}$, μ_{k0} is the prior mean of component k, and $\mathbf{S_k} = \sum_{i;z_i=k}(\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^T$. Within the sampling steps, δ_{ij} is calculated using Eq. (1). The derivation for the posterior conditional distribution of θ_k is given in Appendix.

However, it is clear that the conditional posteriors of $\mathbf{x}_i$ and σ^2 are not trivial to sample from, and so a random-walk Metropolis Hastings algorithm (Hastings, 1970) is used to sample from these distributions. Additionally, sampling from the NIW distribution above involves the following steps:

1. Sample $\mathbf{\Sigma_k}$ from $IW\left(\Psi_0 + \mathbf{S_k} + \frac{n_k}{n_k+1}(\bar{\mathbf{x}}_k - \mu_{k0})(\bar{\mathbf{x}}_k - \mu_{k0})^T, \nu_0 + n_k\right)$,
2. Sample $\mu_k | \mathbf{\Sigma_k}$ from $\mathcal{N}_p\left(\frac{\mu_{k0} + n_k\bar{\mathbf{x}}_k}{n_k+1}, \mathbf{\Sigma_k}\right)$.

Since the Gibbs algorithm requires initial values for the model parameters, the starting values for $\mathbf{x}_i$ are obtained through an initial run of the BMDS algorithm. Then, a GMM is fit to the estimated $\mathbf{X}$ using the Mclust software (Scrucca et al., 2016) for the R environment. Initial values of all other model parameters are obtained from the resulting fit of this GMM.

In their 2007 paper, Oh and Raftery consider a process in which the dimension, p, and number of clusters, G, are estimated simultaneously. Specifically, $p \times G$ models are fit to the data and a selection criteria called "MIC" is calculated for each candidate to select the optimal model among them. However, Oh and Raftery also suggest an alternative, two-step, procedure in which the dimension is first estimated through BMDS, and then models are fit to the data (each containing a different number of clusters). It should be noted that Oh and Raftery found that this two-stage approach may result in more extreme membership probabilities as opposed to simultaneous estimation of p and G, but it is more practical in terms of computing time. Additionally, they concluded that the two-stage approach is still greatly effective in selecting the correct value of p and G. Therefore, our software implementation of the BMCD method takes this two-step approach to estimating the object configuration and clustering of $\mathbf{X}$. Finally, instead of using the MIC selection criteria, we opt to use the BIC due to easy implementation and the fact that it is a standard criterion in mixture modelling.

The optimal number of dimensions is first selected by estimating the object configuration through BMDS for every dimension between 1 and a maximum value, denoted p_{max}, and selecting the dimension with the lowest MDSIC. The calculation of MDSIC is described below, but for a full treatment and derivation, see Section 4 of Oh and Raftery (2001). In one dimension, the MDSIC is calculated as follows:

$$\text{MDSIC}_1 = (m-2)\ \log SSR_1,$$

where $m = \frac{n(n-1)}{2}$ is the total number of dissimilarities and SSR_1 is the sum of squared residuals between the estimated distances between objects (calculated by applying Eq. (1) to the output of BMDS in one dimension) and the observed dissimilarity matrix (consisting of d_{ij}'s). That is,

$$SSR_1 = \sum_{i>j} (\delta_{ij} - d_{ij})^2. \tag{3}$$

For dimensions greater than 1,

$$\text{MDSIC}_p = \text{MDSIC}_1 + \sum_{j=1}^{p-1} LR_j,$$

where, if $s_j = \sum_{i=1}^{n} x_{ij}^2$ and $r_j^{p+1} = \frac{s_j^{p+1}}{s_j^p}$, then

$$LR_p = (m-2)\ \log\left(\frac{SSR_{p+1}}{SSR_p}\right) + \left[(n+1)\sum_{j=1}^{p}\log\left(\frac{r_j^{p+1}(n+1)}{n+r_j^{p+1}}\right) + (n+1)\log(n+1)\right]. \tag{4}$$

Here, SSR_p is calculated according to Eq. (3) by first calculating the δ_{ij}'s based on the output of BMDS in dimension p. Furthermore, s_j^p is calculated using the estimated p-dimensional object configuration. Note that the first term in (4) is approximately the log-likelihood ratio comparing the output of BMDS in dimension p and $(p+1)$. When dimension $(p+1)$ results in a smaller SSR than dimension p, this term is negative. The second term in (4) acts as a penalty when the dimension is increased. Oh and Raftery note that the penalty term is positive when there are significant changes in the output of BMDS between dimensions p and $(p+1)$ which ultimately results in $\prod_{j=1}^{p}\left(n/r_j^{p+1}+1\right) < (n+1)^{p+1}$. Conversely, when there is no significant changes between the output of BMDS in dimensions p and $(p+1)$, $r_j \approx 1$, and so the penalty term is approximately equal to $(n+1)\log(n+1)$. By calculating the MDSIC value in all dimensions between 1 and p_{max}, the optimal dimension can be selected by choosing the one resulting in the lowest MDSIC. The implementation of BMDS and the calculation of MDSIC in the BMCD software package is adapted from the "bayMDS" package (Oh & Lee, 2022).

In summary, the process from start to finish is given as follows. First, BMDS, as described in Oh and Raftery (2001), is performed on the observed distance matrix with entries d_{ij}. Next, the MDSIC is used to estimate the true dimension, p, of the object configuration. Lastly, the estimated object configuration, acquired through BMDS, is used to initialize the BMCD clustering method, which uses Gibbs sampling to simultaneously cluster the data, estimate the object configuration, and estimate the associated measurement error. This process is repeated for each candidate model, each of which will contain a different number of clusters. Finally, model-selection criterion is calculated for each candidate and then used to select a final model.

2.2. The label-switching problem

The label switching phenomenon arises in mixture models due to invariance of the likelihood function under permutation of the parameters. This challenge is explained below:

Suppose we have a mixture model with G components. The density can be written as

$$f(x|\epsilon,\theta) = \sum_{j=1}^{G} \epsilon_j f_j(x|\theta_j),$$

where $\epsilon = (\epsilon_1,\dots,\epsilon_G)$ with $\sum_{j=1}^{G}\epsilon_j = 1$ and $\epsilon_j > 0$ for all j, and $\theta = (\theta_1,\dots,\theta_G)$ denotes the parameter(s) of the mixture components. The likelihood of the mixture model can be written as

$$L(\theta,\epsilon|\mathbf{x}) = \prod_{i=1}^{n}\sum_{j=1}^{G}\epsilon_j f_j(x_i|\theta_j).$$

This likelihood function is the same for all permutations of the parameters. For example, let $\tau\epsilon$ and $\tau\theta$ denote any permutation of the parameters. Then,

$$L(\tau\theta,\tau\epsilon|\mathbf{x}) = L(\theta,\epsilon|\mathbf{x}).$$

In a Bayesian framework, the invariance of the likelihood function leads directly to invariance of the parameters' posterior distribution when the prior distribution of the parameters does not differentiate between components (i.e. the prior distribution is also invariant under permutations):

$$\begin{aligned}\pi(\theta,\epsilon|\mathbf{x}) &\propto L(\theta,\epsilon|\mathbf{x})\pi(\theta,\epsilon)\\ &= L(\tau\theta,\tau\epsilon|\mathbf{x})\pi(\tau\theta,\tau\epsilon) \quad \text{(by invariance)},\\ \pi(\tau\theta,\tau\epsilon|\mathbf{x}) &\propto L(\tau\theta,\tau\epsilon|\mathbf{x})\pi(\tau\theta,\tau\epsilon).\end{aligned}$$

The invariance of the posterior distribution under permutations means that estimation of component-specific parameters can become problematic. For example, the posterior distribution will exhibit $G!$ different modes. This multi-modality may render component-specific estimates obtained through MCMC sampling, such as posterior means and credible intervals, effectively meaningless, since the sampler may move between modes of the posterior distribution between iterations. This obstacle to obtaining meaningful posterior estimates due to the invariance of label permutations is known as the "label-switching problem".

Due to the rising popularity of Bayesian analyses, several methods to address the label-switching problem have been proposed. Many such methods rely on post-processing of the samples obtained through an employed MCMC algorithm. However, for problems in which each iteration relies on correct labelling of the previous iteration's parameters, post-processing is not viable. Instead, label switching must be done between each iteration. One such algorithm of "on-the-fly" label-switching is proposed in Celeux (1998) and is used within the BMCD method. The algorithm is given as follows.

Let ξ^t denote the p-dimensional parameter vector of the mixture model (generated by an MCMC algorithm or some other means) at iteration t. In a (univariate) Gaussian mixture model with 2 components, for example, $\xi^t = (\epsilon_1^t, \epsilon_2^t, \mu_1^t, \mu_2^t, \sigma_1^t, \sigma_2^t)$. The algorithm is initialized by using the first 100 iterations (using the first 100 iterations allows one to be fairly confident that a label switch has not occurred yet) and we denote this value m. Using these m iterations, the variance of each component, $i = 1, \dots, p$ of the parameter vector, ξ, is calculated:

$$(s_i^{[0]})^2 = \frac{1}{m}\sum_{t=1}^{m}(\xi_i^t - \bar{\xi}_i)^2,$$

where $\bar{\xi}_i = \frac{1}{m}\sum_{t=1}^{m}\xi_i^t$. Then, an initial reference centre, denoted $\bar{\xi}_1^{[0]}$, is defined by $\bar{\xi}_1^{[0]} = (\bar{\xi}_1, \dots, \bar{\xi}_p)$. Next, $G! - 1$ other centres, $\bar{\xi}_2^{[0]}, \dots, \bar{\xi}_{(G!-1)}^{[0]}$, are derived by permuting labelling of the mixture components. In the previous normal mixture model example, for example, one other centre is derived when the labels 1 and 2 are exchanged.

In each successive iteration, two steps are carried out: in the first step, the vector ξ^{m+r} is assigned to the permutation that minimizes the normalized squared distance to a particular reference centre

$$\left\|\xi^{m+r} - \bar{\xi}_j^{[r-1]}\right\|^2 = \sum_{i=1}^{p}\frac{(\xi_i^{m+r} - \bar{\xi}_{ij}^{[r-1]})^2}{(s_i^{[r-1]})^2}, \quad j = 1, \dots, G!$$

where $\bar{\xi}_{ij}^{[r-1]}$ denotes the centre of the ith component of ξ in the jth centre. If any $j \neq 1$ minimizes the distance, then the labels of ξ^{m+r} are permuted to ensure that $\bar{\xi}_1^{[r-1]}$ is the centre that minimizes the distance (i.e. the labels from the initial m iterations are restored).

In the second step, the variances and reference centre is updated. Once again, the other $G! - 1$ centres are derived through permuting the labels of the updated reference centre. The update formulas are given in the original paper.

Celeux's algorithm attempts to correct any label switching that occurs in each iteration of the sampler and does not rely on post-processing. However, the cost of this algorithm comes in the form of computational efficiency. Namely, in each iteration, $G!$ centres are derived and the distance between each centre and ξ^{m+r} must be calculated to find the minimum. For even moderately large numbers of clusters, this algorithm quickly becomes intractable since each iteration requires deriving an entirely new set of centres through permutation and recalculating distances. We propose a computationally-efficient approach to this obstacle in Section 3.3.

3. Extension of BMCD

In Section 2.1, we introduced the Bayesian GMM given in Oh and Raftery (2007). In high-dimensional data, this generic Gaussian mixture model with unconstrained covariances may provide disappointing results due to the large number of parameters that must be estimated. This problem is often referred to as the "curse of dimensionality". For instance, in the generic model (known as the "VVV" model in the Mclust software), there are $\kappa = (G-1)+(Gp)+Gp(p-1)/2$ free parameters to be estimated. In the calculation of κ, the first term is due to the mixing parameters (constrained to sum to unity), the second term is due to the means of the mixture components, and the last term is due to the covariance matrices of the components. As a result, the number of free parameters is a quadratic function of the dimension, p. Therefore, as p grows with a fixed number of observations, estimating the large number of parameters becomes difficult. Despite this challenge, the number of parameters can be reduced by constraining the number of free parameters. Namely, constraints can be introduced into the structure of the covariance matrices of the mixture components to greatly reduce the number of free parameters and thereby promote model parsimony (Celeux & Govaert, 1995). These constraints can be used to improve the quality of clustering and have been shown to be effective in real-world applications. For example, the parsimonious models have been used in the context of gene expression analysis (Yeung et al., 2001), flow cytometry (Gormley et al., 2023), and COVID-19 data (García-Escudero et al., 2022). The parsimonious models are given in Section 3.1, and model selection criteria for choosing a final model among these candidates is given in Section 3.2.

In Section 3.3, we propose an alternative approach to the naive implementation of the label-switching algorithm given in Section 2.2. Our approach aims to reduce the steep computational cost associated with the algorithm, allowing for an efficient software implementation of the BMCD method.

3.1. Parsimonious models

In this section, five different models are given, each of which place a different constraint upon the component covariance matrices, thereby changing the number of free parameters and shape of the clusters. In addition to the original model given in Oh and Raftery (2007) (which we will refer to as the unequal unrestricted model), this gives a total of six unique model types. Furthermore, the Gibbs sampling steps for the component parameters μ and $\mathbf{\Sigma}$ are given for each of the five additional models. For convenience, we use conjugate priors in each model so that the posteriors are simple to derive and easy to sample from.

The following notation is used in the models below:

$$\mathbf{W_k} = \sum_{i:z_i=k}(\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^T$$
$$n_k = \sum_i I(z_i = k).$$

3.1.1. Equal spherical

In the first model, there is a single free parameter in the covariance matrices of all components. That is, $\mathbf{\Sigma_k} = \lambda I_p$ for $k = 1, \dots, G$, where λ is a scalar quantity. This results in a spherical shape of the clusters and reduces the number of parameters to $\kappa = (G-1)+(Gp)+1$. Note that this model is the most restrictive of any model considered.

The priors used for μ_k and λ are:

$$\lambda \sim IG(\alpha, \beta)$$
$$\mu_k|\lambda \sim \mathcal{N}_p(\mu_{k0}, \lambda I_p)$$

These prior distributions result in the following Gibbs sampling steps (Bensmail, 1997):

1. Sample
$$\lambda \sim IG\left(\alpha + \frac{np}{2},\ \beta + \frac{1}{2}\sum_{k=1}^{G}\left[\mathrm{tr}(\mathbf{W_k}) + \frac{n_k}{n_k+1}(\bar{\mathbf{x}}_k - \mu_{k0})^T(\bar{\mathbf{x}}_k - \mu_{k0})\right]\right)$$
2. For $k = 1, \dots, G$, sample
$$\mu_k|\lambda \sim \mathcal{N}_p\left(\frac{n_k\bar{\mathbf{x}}_k + \mu_{k0}}{n_k+1}, \frac{\lambda I_p}{n_k+1}\right)$$

The clusters resulting from this model are equal in volume and spherical in shape.

3.1.2. Unequal spherical

Instead of using a single parameter for the scalar quantity of the covariance matrices, a different λ can be used for each component, resulting in a less restrictive model. Let $\mathbf{\Sigma_k} = \lambda_k I_p$ for $k = 1, \dots, G$. Then, the number of free parameters is $\kappa = (G-1)+(Gp)+G$. The priors used are:

$$\lambda_k \sim IG(\alpha_k, \beta_k),$$
$$\mu_k|\lambda_k \sim \mathcal{N}_p(\mu_{k0}, \lambda I_p)$$

and the Gibbs sampling steps are given as:

For $k = 1, \ldots, G$, sample

1. $\lambda_k \sim IG\left(\alpha_k + \frac{n_k p}{2},\ \beta_k + \frac{1}{2}\left[\text{tr}(\mathbf{W_k}) + \frac{n_k}{n_k+1}(\bar{\mathbf{x}}_k - \mu_{k0})^T(\bar{\mathbf{x}}_k - \mu_{k0})\right]\right)$
2. $\mu_k|\lambda_k \sim \mathcal{N}_p\left(\frac{n_k\bar{\mathbf{x}}_k+\mu_{k0}}{n_k+1}, \frac{\lambda_k I_p}{n_k+1}\right)$

With this model, the resulting clusters are still spherical in shape, but they are now allowed to vary in volume.

3.1.3. Equal diagonal

In the equal diagonal model, the covariance matrices are constrained to be diagonal and equal across components. That is, $\mathbf{\Sigma_k} = \mathbf{A}$, where $\mathbf{A}$ is a diagonal matrix. The resulting number of free parameters is $\kappa = (G-1) + (Gp) + p$. Let the diagonal elements of $\mathbf{A}$ be denoted a_q, $q = 1, \ldots, p$. Then, the priors used are

$$a_q \sim IG\left(\alpha_q, \beta_q\right)$$
$$\mu_k|\mathbf{\Sigma_k} \sim \mathcal{N}_p\left(\mu_{k0}, \mathbf{\Sigma_k}\right)$$

Let $D = \frac{1}{2}\sum_{k=1}^{G}\left[\mathbf{W_k} + \frac{n_k}{n_k+1}(\bar{\mathbf{x}}_k - \mu_{k0})(\bar{\mathbf{x}}_k - \mu_{k0})^T\right]$. Then, the Gibbs sampling steps are given as

1. For $q = 1, \ldots, p$, sample
$$a_q \sim IG\left(\alpha_q + \frac{n}{2},\ \beta_q + D_q\right),$$
where D_q is the qth diagonal element of D.
2. For $k = 1, \ldots, G$, sample
$$\mu_k|\mathbf{\Sigma_k} \sim \mathcal{N}_p\left(\frac{n_k\bar{\mathbf{x}}_k + \mu_{k0}}{n_k+1}, \frac{\mathbf{\Sigma_k}}{n_k+1}\right)$$

The resulting clusters are equal in volume and shape across clusters, while the orientation of the cluster is aligned with the coordinate axes.

3.1.4. Unequal diagonal

In the unequal diagonal model, the covariance matrices are still constrained to be diagonal but can differ between components. That is, $\mathbf{\Sigma_k} = \mathbf{A}_k$, where $\mathbf{A}_k$ is a diagonal matrix. The resulting number of free parameters is $\kappa = (G-1) + (Gp) + (Gp)$. Let the diagonal elements of $\mathbf{A}_k$ be denoted a_{kq}, $q = 1, \ldots, p$. Then, the priors used are

$$a_{kq} \sim IG\left(\alpha_{kq}, \beta_{kq}\right)$$
$$\mu_k|\mathbf{\Sigma_k} \sim \mathcal{N}_p\left(\mu_{k0}, \mathbf{\Sigma_k}\right)$$

For simplicity, let $D_k = \frac{1}{2}\left(\mathbf{W_k} + \frac{n_k}{n_k+1}(\bar{\mathbf{x}}_k - \mu_{k0})(\bar{\mathbf{x}}_k - \mu_{k0})^T\right)$. Then, the Gibbs sampling steps, for $k = 1, \ldots, G$, are:

1. For $q = 1, \ldots, p$, sample
$$a_{kq} \sim IG\left(\alpha_{kq} + \frac{n_k}{2},\ \beta_{kq} + D_{kq}\right),$$
where D_{kq} is the qth diagonal element of D_k.
2. For $k = 1, \ldots, G$, sample
$$\mu_k|\mathbf{\Sigma_k} \sim \mathcal{N}_p\left(\frac{n_k\bar{\mathbf{x}}_k + \mu_{k0}}{n_k+1}, \frac{\mathbf{\Sigma_k}}{n_k+1}\right)$$

3.1.5. Equal unrestricted

The equal unrestricted model does not constrain the parameters of any particular component's covariance matrix, but ensures that the matrix is equal across components. That is, $\mathbf{\Sigma_k} = \mathbf{\Sigma}$ for each k, resulting in $\kappa = (G-1) + (Gp) + p(p-1/2)$ free parameters. The priors used are:

$$\mathbf{\Sigma} \sim IW(\alpha, B)$$
$$\mu_k|\mathbf{\Sigma} \sim \mathcal{N}_p\left(\mu_{k0}, \mathbf{\Sigma}\right).$$

The Gibbs sampling steps are:

1. Sample
$$\mathbf{\Sigma} \sim IW\left(\alpha + n, B + \sum_{k=1}^{G}\left[\mathbf{W_k} + \frac{n_k}{n_k+1}(\bar{\mathbf{x}}_k - \mu_{k0})(\bar{\mathbf{x}}_k - \mu_{k0})^T\right]\right)$$
2. For $k = 1, \ldots, G$, sample
$$\mu_k|\mathbf{\Sigma} \sim \mathcal{N}_p\left(\frac{n_k\bar{\mathbf{x}}_k + \mu_{k0}}{n_k+1}, \frac{\mathbf{\Sigma}}{n_k+1}\right)$$

As a result of the equality of covariance matrices, the volume, shape, and orientation of the clusters are all equal.

3.2. Model selection

A key advantage in using model-based clustering is the ability to calculate model selection criteria such as BIC and ICL. Here, we use BIC to select a final model among a set of candidate models, although the ICL could also be easily implemented. The formula for calculating the BIC is given as:

$$BIC = \kappa\ln(n) - 2\ln(\hat{L}),$$

where κ is the number of estimated parameters, n is the number of observations, and $\hat{L}$ is the maximum likelihood of the model, given a set of estimated parameter values. Since the likelihood of a model can be increased by adding parameters, the BIC introduces a penalty term to avoid overfitting.

Model selection in the BMCD software proceeds as follows. First, the BMCD clustering method fits several candidate models to the data. Each of the six implemented models given above are fit to the data, with a user-specified number of clusters. Then, using samples drawn from the posterior distribution, model parameters are estimated for each model. For each model, the BIC is calculated and the selected (final) model is the candidate model that results in the lowest BIC value.

3.3. Efficient label-switching algorithm

This section aims to address the steep computational cost associated with the label-switching algorithm introduced in Section 2.2. To reduce this cost, the following modified approach is proposed. Firstly, let $\theta^t_{k_d}$ denote parameter d of component k at iteration t. Using the same notation given in Section 2.2, calculate

$$S_{jk} = \sum_{d=1}^{D}\frac{(\theta^{m+r}_{j_d} - \bar{\theta}^{[r-1]}_{k_d})^2}{(s^{[r-1]}_{j_d})^2} \quad \text{for } j = 1, \ldots, G \text{ and } k = 1, \ldots, G.$$

Then store each resulting S_{jk} in a $G \times G$ cost matrix, denoted A. Note that $\bar{\theta}^{[r-1]}_{k_d}$ and $s^{[r-1]}_{k_d}$ are both calculated analogously to the previous algorithm.

With the cost matrix A, the objective is to find a row permutation so as to minimize the trace of A. This is a well-studied problem in combinatorics known as the "Assignment Problem". Fortunately, there exists several algorithms to solve or approximately solve the assignment problem. One such solution is known as the "Hungarian Method" (Kuhn, 1955) and scales significantly better than the naive solution. The software implementation of BMCD uses the Hungarian method to efficiently find a solution to the assignment problem. Once a solution is found through the Hungarian method, the labels are permuted accordingly (if necessary) and the algorithm continues as before. In addition to efficient run-time, the proposed method also has the benefit of avoiding the storage of large permutation vectors/lists in memory.

4. Simulation studies

Two simulation studies are conducted to assess the performance of the BMCD method. In both studies, BMCD and other clustering methods are applied to data simulated under a variety of conditions. Both studies use a unique combination of the following parameters to simulate data: sample size (n = 100 or 200), true number of clusters (G = 2, 4, 6, or 8), true (unobserved) dimension of the data (p = 3, 6, or 12), and three different values for the measurement error present in the data

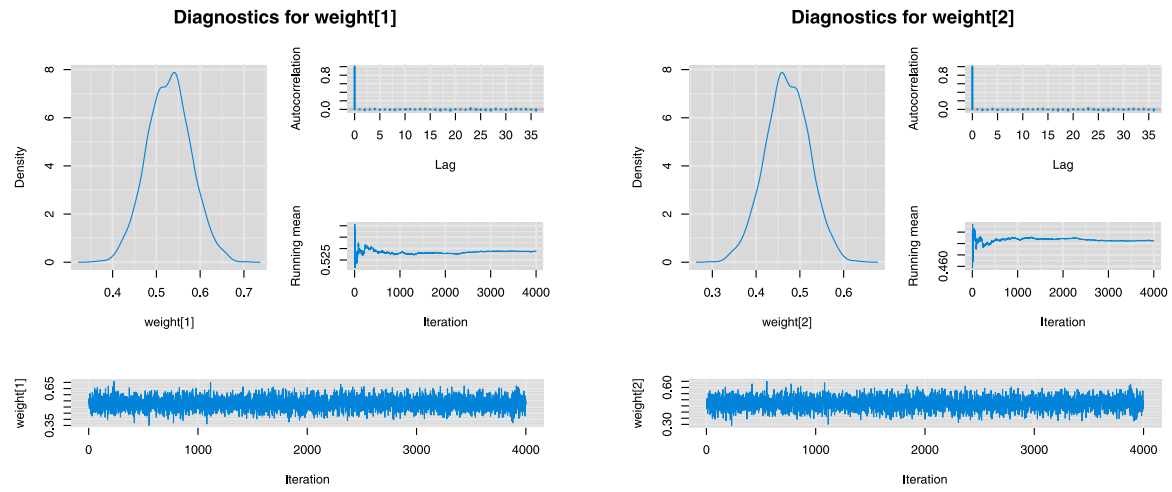

Fig. 1. Diagnostic plots produced by the function `diagnostic` for the weights of a two-component mixture model. Both trace plots indicate convergence of the MCMC chain.

($\sigma^2 = 0, 5,$ or 10). Therefore, 72 unique data sets are generated in each simulation study. We use the following process to generate this data:

1. Data is generated from a multivariate normal mixture distribution in p dimensions with G components. We use the *MixSim* package (Melnykov et al., 2012) to generate component means and covariance matrices and subsequently simulate the data.
2. Euclidean distances between observations are calculated (denoted δ_{ij}).
3. To add measurement error to the data, we generate the observed distances (d_{ij}) from a truncated normal distribution with mean δ_{ij} and variance σ^2.

As the accuracy of our simulations rely on convergence of the MCMC algorithm to the target distribution, it is vital to ensure that the samples drawn for each parameter have adequately stabilized within the specified number of iterations. For example, convergence can be assessed by computing the Gelman–Rubin statistic (Gelman & Rubin, 1992), which involves running multiple MCMC chains and comparing the variability within each chain to the variability between chains. However, this approach may involve increased computing time due to the necessity of multiple chains, in addition to the expertise necessary to interpret the statistic. Instead, we opt to assess convergence visually by inspecting parameter trace plots. These plots show the parameter value sampled (plotted on the y-axis) through each iteration of the MCMC algorithm (plotted on the x-axis). By ensuring that no persistent trends or patterns or present in the plot and that the parameters values are stable, we can evaluate convergence of the MCMC algorithm.

A potential complication in assessing MCMC convergence through trace plots is that a large number of parameters may cause manual inspection to be cumbersome. To reduce this difficulty, our software package includes a function (`diagnostic`) that produces a file complete with trace plots for each parameter value in the model. Furthermore, this file enables quick inspection and is easily navigable even for many parameters. In addition to the trace plot, the kernel density plot, autocorrelation plot, and running mean are also produced for each parameter in the model. Fig. 1 shows examples of the diagnostic plots produced for the weights of a mixture model with two components. After 4000 iterations, the trace plots (placed at the bottom of each panel) indicate stable parameter values with no evident trends. Thus, this may help indicate convergence of the MCMC algorithm, although inspection of other parameter values in the model is necessary to make a final conclusion. By default, our software package uses 5000 iterations with a burn-in period of 1000 iterations. However, in our simulations, we found that convergence is usually attained within the first few hundred iterations of the algorithm. Thus, the number of iterations can (and should) be manually controlled and catered toward each specific application.

Table 1
Mean and Standard Deviation of Non-Parallel and Parallel Computing Times for Sample Sizes 100 and 200. All computing times are given in minutes.

Sample size	Non-parallel time		Parallel time	
	Mean	Std Dev	Mean	Std Dev
100	32.208	0.918	20.914	0.119
200	103.995	6.538	31.743	3.392

Simulations were performed on the Digital Research Alliance of Canada's Cedar service (Digital Research Alliance of Canada, 2023), a CentOS v7 cluster featuring Intel E5-2683 v4 Broadwell processors running at 2.10 GHz and equipped with 125G of memory. Table 1 provides computing times (in minutes) for the simulations featuring the highest dimension and highest number of clusters considered in our experiments (12 and 8, respectively). For both sample sizes ($n = 100$ and $n = 200$), we present the average computing time required to execute the BMCD algorithm over ten independent runs with 5000 iterations and a burn-in period of 1000. Additionally, we include the standard deviation of these ten time measurements. In each run of the algorithm, we first estimate the dimension of the dataset using BMDS (with $p_{max} = 15$), and then we fit each of the six models for every cluster between 2 and 10. Therefore, each algorithm run consists of a total of 15 MCMC chains for the multi-dimensional scaling portion of the algorithm (75,000 total iterations), and 54 MCMC chains for the model-fitting portion of the algorithm (270,000 total iterations.)

When applied to increased sample sizes such as $n = 500$, we have found that the computational time of the BMCD algorithm scales inefficiently and quickly becomes intractable. This finding underscores one of the main challenges with the use of the BMCD algorithm. Therefore, application of the algorithm warrants a careful consideration of the associated computational cost against its inherent advantages. The BMCD algorithm's scalability is an area of consideration, and ongoing efforts are directed towards exploring avenues for optimization in order to address this aspect.

4.1. First simulation study

The first simulation study assumes both p and G are known *a priori*. For each of the 72 data sets, we apply four different clustering procedures:

1. BMCD, in which p dimensions are retained from BMDS and the 6 different model types described in Section 3.1 are fit to the

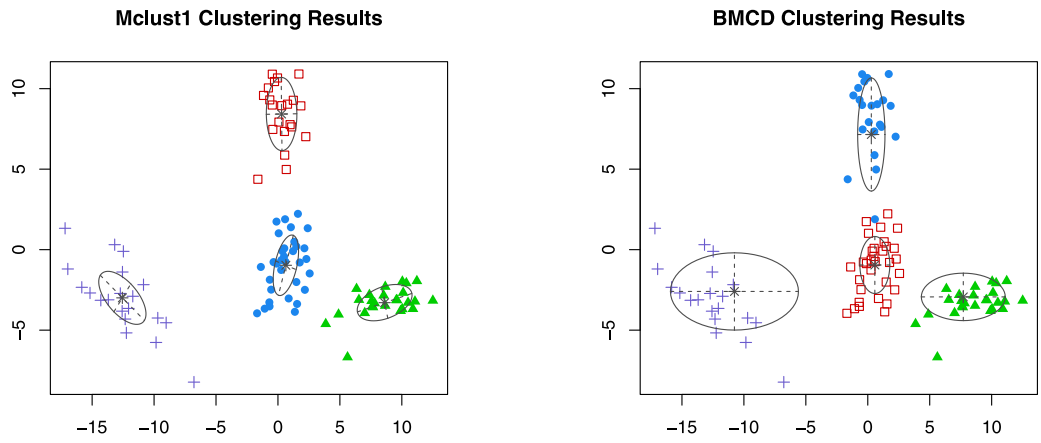

Fig. 2. The clustering partitions resulting from the Mclust1 method (left) and the BMCD method (right) for a simulated dataset with $n = 100$, true dimension equal to 4, and measurement error equal to 5.

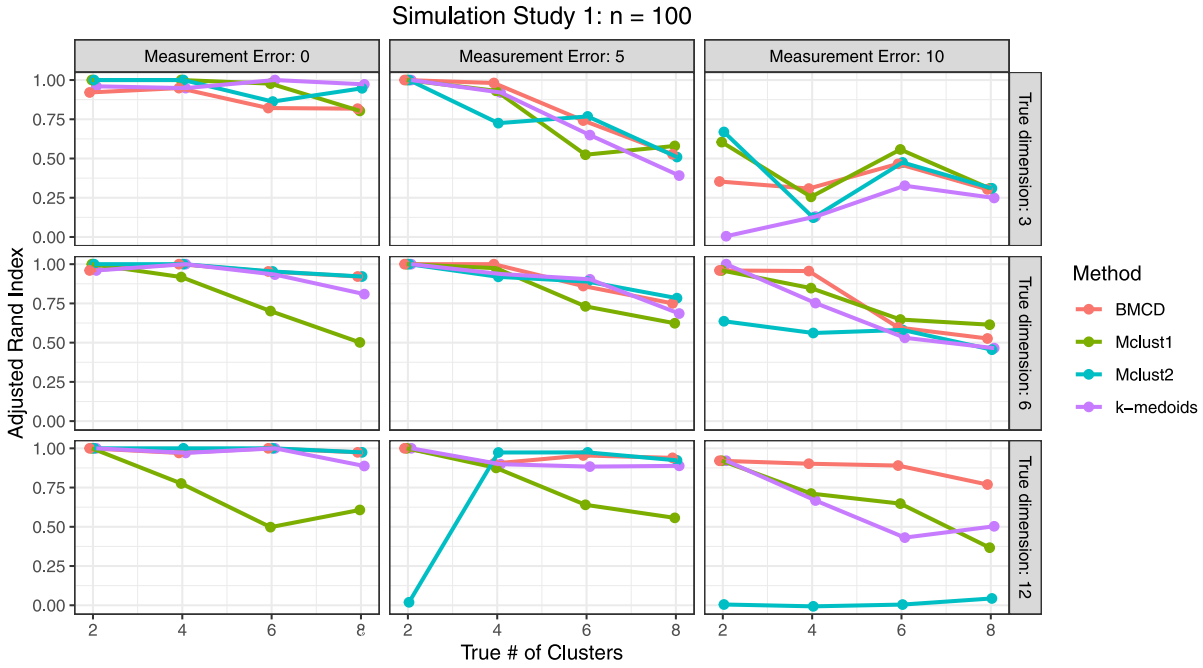

Fig. 3. Results of Simulation Study 1 with sample size $n = 100$.

data using Gibbs sampling. Each candidate model contains G components,

2. A two-stage approach, which uses the output of MDS (with p dimensions retained) to fit 14 unique model types (Scrucca et al., 2016) using the Expectation–Maximization (EM) algorithm. Each candidate model once again contains G components.
3. An alternative two-stage approach, which uses the output of BMDS (again with p dimensions retained) with the EM algorithm to fit the 14 candidate models with G components.
4. K-medoids with G clusters is applied directly to the observed distances.

The second and third approach are respectively denoted $Mclust1$ and $Mclust2$, since the Mclust package in R is used to implement both procedures. Note that the difference between the second and third strategy is the type of multi-dimensional scaling that is used (MDS vs. BMDS), while the actual clustering method remains the same. Since the first three procedures fit multiple candidate models to the data, the model with the lowest BIC is selected as the representative model for each approach.

After the four approaches have been applied to the data, the Adjusted Rand Index (ARI) (Hubert & Arabie, 1985) is calculated for each method in order to compare the resulting partition to the true clustering partition. A higher ARI value indicates a closer agreement between the fitted partition and the true partition. For instance, the left-hand panel of Fig. 2 shows the clustering partition resulting from the *Mclust1* method for one particular simulated dataset with $n = 100$, the true dimension equal to 4, and measurement error equal to 5. For visualization purposes, only the first two dimensions are shown. Similarly, the right-hand panel of Fig. 2 shows the clustering partition resulting from the Unequal Diagonal model of the BMCD method. While the Mclust1 method resulted in an ARI of 0.9716, the BMCD method resulted in a perfect ARI of 1.0 (despite a seemingly misclassified blue observation near the centre of the plot).

The results of the simulation study are shown graphically in Fig. 3 and Fig. 4 for sample sizes $n = 100$ and $n = 200$, respectively. Both figures show the ARI values for the partitions resulting from applying the different clustering methods to the generated data (36 data sets of size $n = 100$ and 36 data sets of size $n = 200$). From the plots, it is

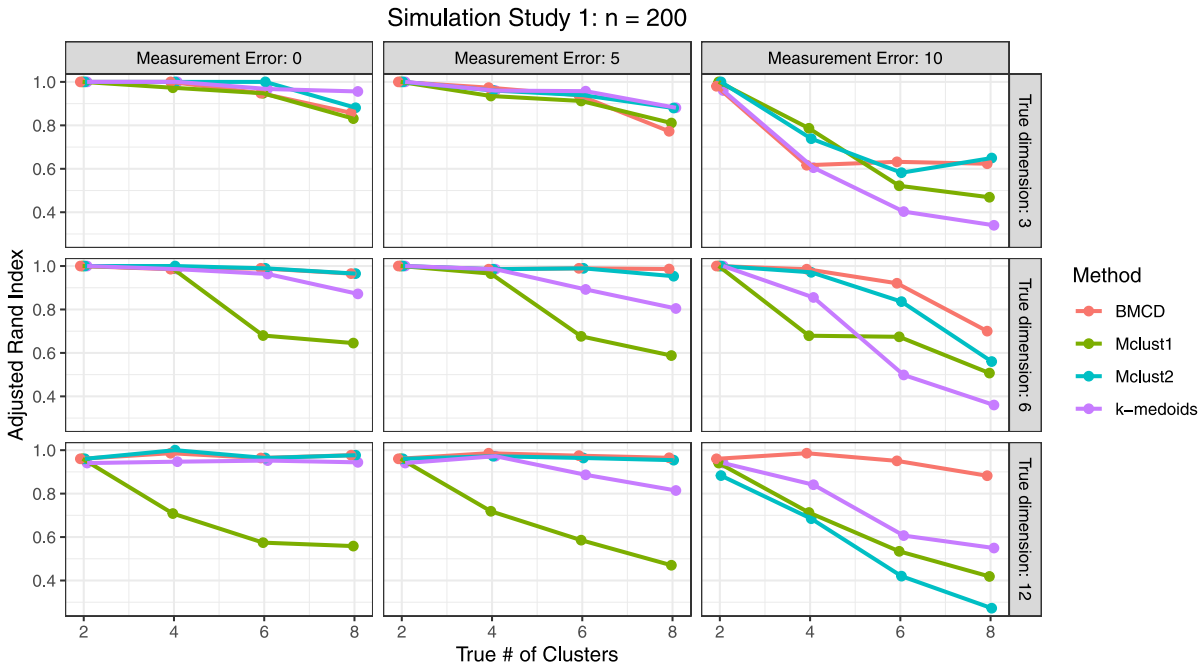

Fig. 4. Results of Simulation Study 1 with sample size $n = 200$.

clear that BMCD performs exceptionally well when the true dimension is high and there is high amount of measurement error present. Still, even in low dimensions and with little measurement error, BMCD often outperforms the other clustering approaches.

4.2. Second simulation study

In the second simulation study, we do not assume that the number of clusters or true dimension of the data set is known. Thus, this study assesses the performance of BMCD under more realistic conditions. The following clustering procedures are applied to each of the 72 data sets:

1. BMCD, in which the dimension of the data set is estimated by applying BMDS and selecting the dimension resulting in the lowest MDSIC value. For each of the six model types in Section 3.1, candidate models are fit to the data with the number of components ranging from 2 to 10. Therefore, 54 total candidate models are fit to each data set.
2. An two-stage approach, which retains only the first 2 dimensions from MDS and then fits the 14 unique model types with the EM algorithm. Once again, the number of components range from 2 to 10 for each model type, and so the total number of candidate models is 126.
3. A alternative two-stage approach, in which the dimension is first estimated using BMDS. Next, 14 unique model types are fit to the output of MDS using EM algorithm and with the number of components ranging from 2 to 10. As a result, 126 total candidate models are fit to each data set.

Since the number of clusters is unknown, we omit the k-medoids clustering from this study. We once again denote the second and third approach as *Mclust1* and *Mclust2*, respectively. However, it is important to note that the approaches do differ between the two simulation studies in their approach to estimating the dimension as well as the total number of resulting candidate models. For all three approaches, the representative model is chosen by selecting the candidate model with the lowest BIC. Finally, the ARI is once again calculated for each representative model to determine the degree of agreement between the fitted clustering partition and the true partition. The results of the simulation study are shown graphically in Fig. 5 for sample size $n = 100$, and in Fig. 6 for sample size $n = 200$. It is evident that BMCD usually outperforms the *Mclust1* approach and performs similarly to *Mclust2*.

5. Application to breast cancer data

The BMCD method is applied to the Wisconsin Diagnostic Breast Cancer (WDBC) data set (Street et al., 1993). The data set is available for download from the University of California Irvine machine learning repository (Dua & Graff, 2017). This data set contains different features computed from a fine needle aspirate of a breast mass in 569 patients. Ten different attributes are computed on the cell nuclei within the mass, including the nuclei radius, perimeter, area, and symmetry. For each of these ten attributes, the mean, standard error, and mean of the three largest values (known as the "extreme" features) are reported, resulting in a total of 30 different features. Along with these features, an outcome variable is included that indicates whether the mass is malignant or benign.

We aim to use the features of the WDBC data set to cluster the breast masses into distinct groups using BMCD. We use the 10 extreme features of the data set to conduct the clustering and, for the purposes of lower computing time, we use a random sample of 100 patients masses from the original 569 patients. We also tested the BMCD method on a random sample of 200 patients with similar results, but with a larger computational cost. First, the features are standardized and the Euclidean distances between breast masses are computed. Next, the BMDS algorithm is employed to estimate the true dimension of the data set. Although the true dimension is known to be 10 in this case, this preliminary step allows us to ensure that the BMDS algorithm works as intended in scenarios where the true dimension is unknown. The BMDS algorithm is run for dimensions between 1 and 20 and, subsequently, the MDSIC is computed. The MDSIC values for each dimension are shown in Fig. 7. From this plot, it is evident that there is a large drop in the MDSIC at $p = 10$ and, after that, the MDSIC does not change significantly. Thus, we use $p = 10$ as the estimated dimension and note that BMDS has correctly identified the true dimension of the data set.

Next, the BMCD algorithm is used to fit the original, unrestricted model and the five additional model types given in Section 3.1. For each model, between 2 and 8 clusters are fit to the data, resulting in

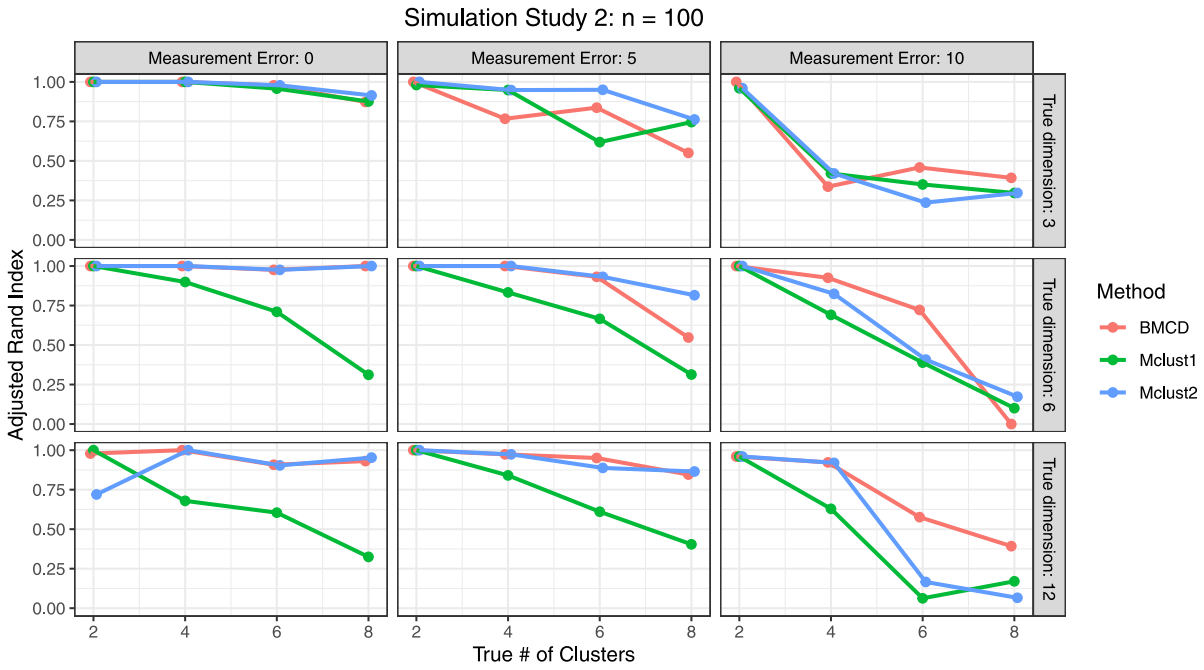

Fig. 5. Results of Simulation Study 2 with sample size $n = 100$.

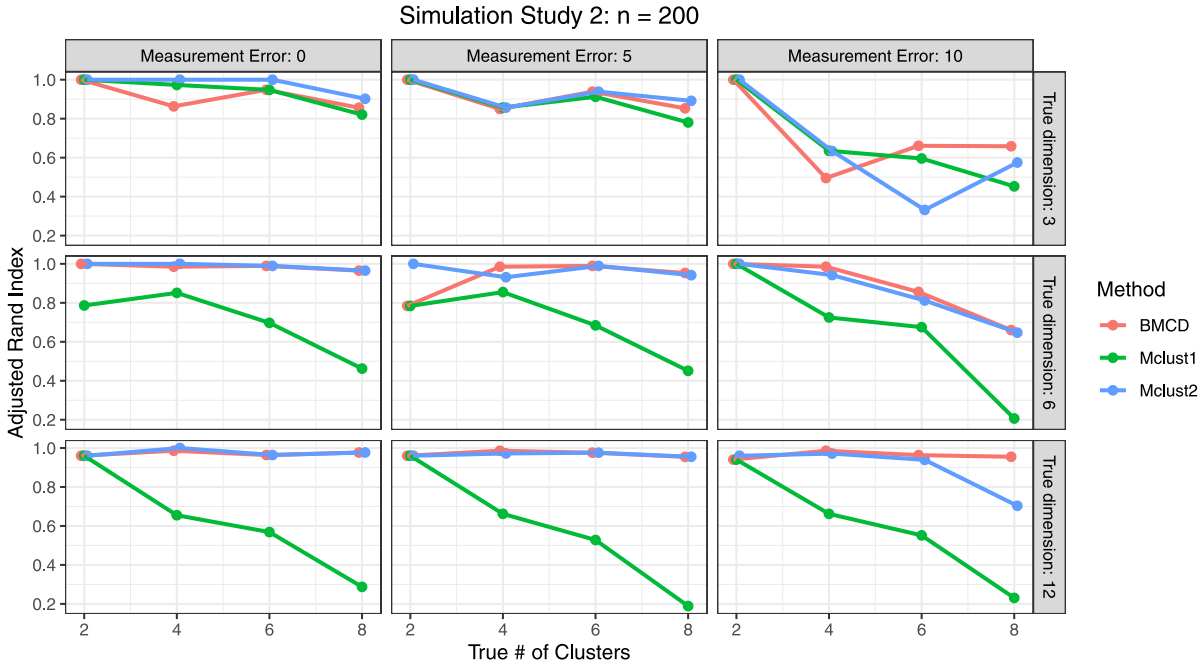

Fig. 6. Results of Simulation Study 2 with sample size $n = 200$.

a total of 42 candidate models. Parallel computing is utilized so that more than one model can be fit at a time, hence decreasing the required computing time. After each candidate model has been fit, the BIC is calculated and is shown in Fig. 8. Among this set of candidate models, we prefer the one with the lowest BIC. Consequently, we select the model in the unequal unrestricted family with 2 clusters. Thus, BMCD has correctly identified the true number of clusters within the data (malignant vs benign).

To assess the model fit, we may examine the final BMCD model compared to the known clustering partition. Although $p = 10$, we limit our attention to the first 2 dimensions for visualization purposes. This comparison is shown in Fig. 9. Since clustering is an unsupervised process, it is unknown which cluster consists of malignant or benign masses. We may also compute the ARI to evaluate model accuracy. We compare the ARI of the BMCD model to the ARI of a GMM fit through the Mclust package on the original data. Using the 10 extreme features, Mclust also selected a final model in the unequal unrestricted family with 2 clusters. An additional Mclust model is fit using a two-stage approach in which the estimated object configuration from the BMDS algorithm (with $p = 10$) is used as input to the Mclust software. Once again, Mclust selected a final model with 2 clusters. The final ARI results are given in Table 2. From the values in the table, it is evident that BMCD results in the highest ARI, indicating that BMCD provides

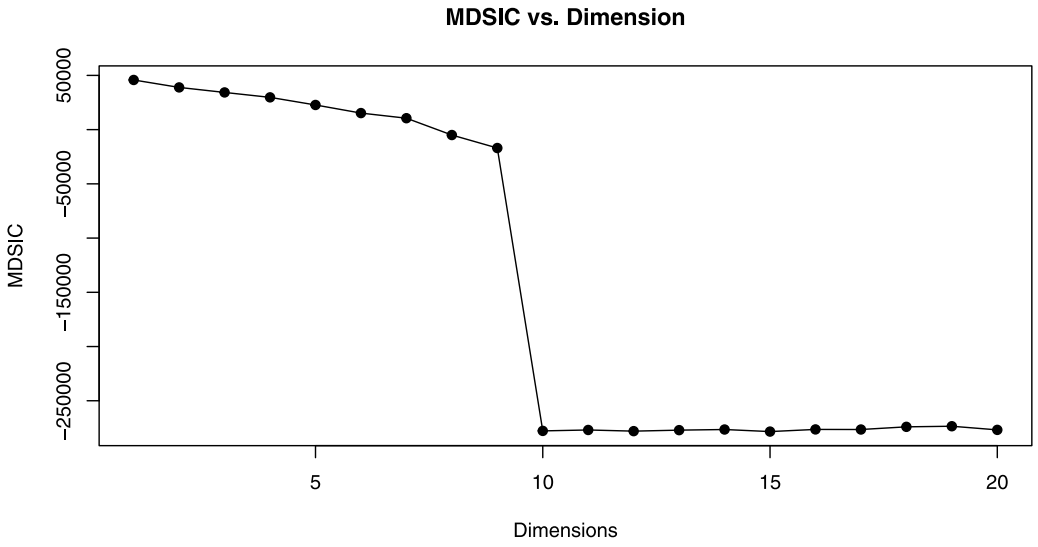

Fig. 7. MDSIC of WDBC data for $p = 1$ to $p = 20$.

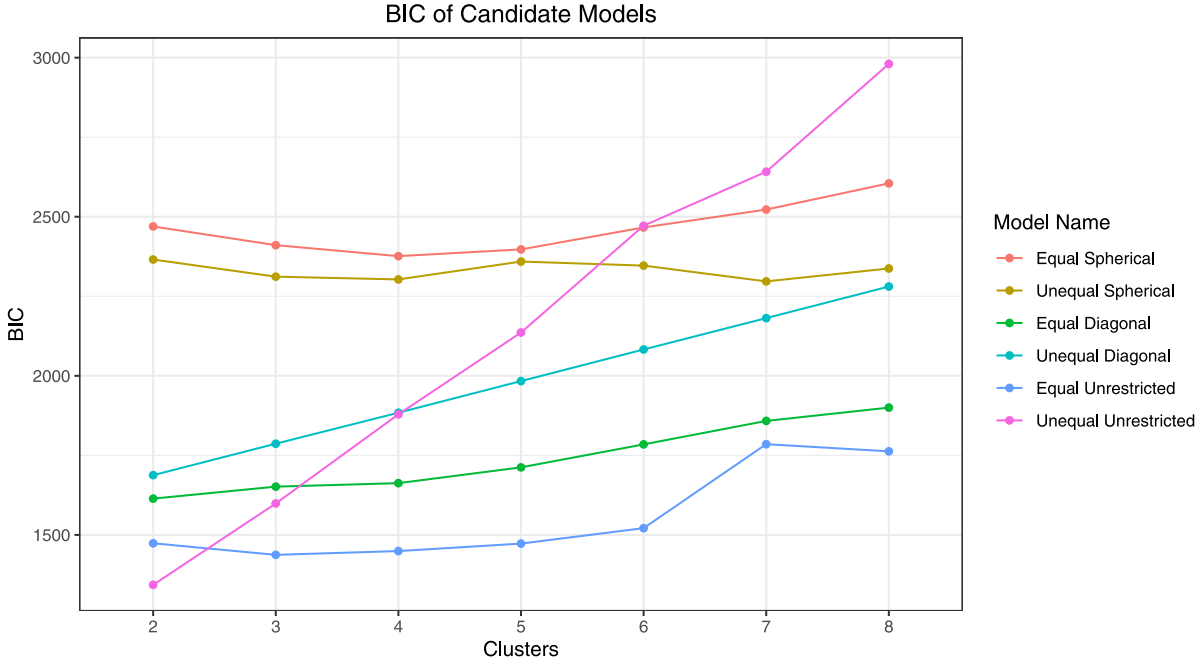

Fig. 8. BIC of the 42 candidate models.

Table 2
ARI of clustering partitions fit using three different methods.

Method	BMCD	Mclust	BMDS + Mclust
ARI	0.7668	0.6622	0.5609

the most accurate clustering partition, even though all methods were able to identify the true number of clusters.

6. Conclusion

Several modifications to the BMCD clustering algorithm were presented and implemented. Firstly, five additional models were included with the intention of capturing different shapes of clusters as well as promoting parsimony when the true dimension of the data is high. The Gibbs sampling steps for these models were derived and implemented in the R package "BMCD" (available on GitHub at https://github.com/SamMorrissette/BMCDcpp). Secondly, the BMCD software reduces computing time of the algorithm through parallel computing and other strategies. For instance, the Hungarian Method is used as a solution to the assignment problem, which arises from the issue of label-switching in Bayesian inference. This method avoids the storage of overly large objects in memory. Additionally, in accordance with the suggestion of Oh and Raftery, the dimension of the data is first estimated using BMDS, and then the number of clusters is selected using model selection criteria. This reduces computing time by requiring the BMCD algorithm to be run G times, instead of $p \times G$ times if the dimension and number of clusters were to be estimated simultaneously.

Although steps were taken to reduce computation time in the implementation of the algorithm, the required time may still be a deterrent to using BMCD. For data sets with many observations or a high number of dimensions, the computing time does not scale well. Although the software package is implemented in the C++ language, due to the

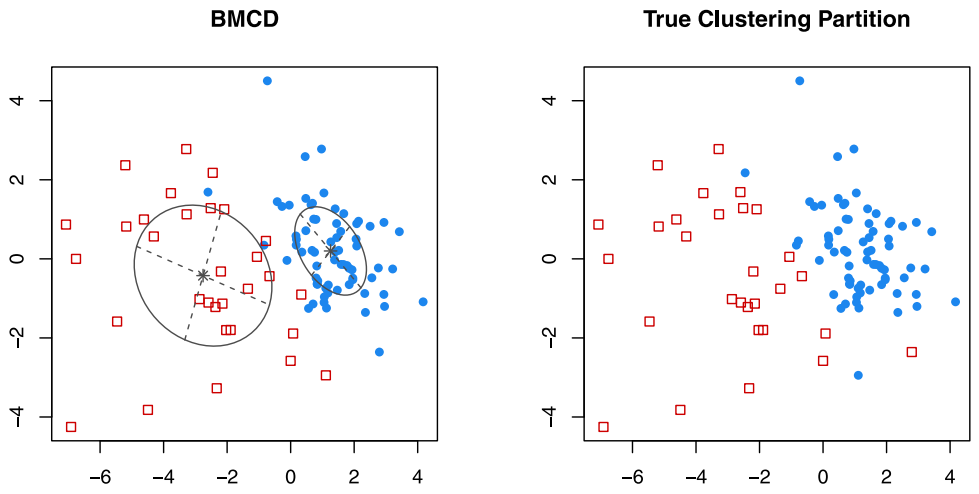

Fig. 9. BMCD fit of the unequal unrestricted model (left) versus the true clustering partition (right). The first two dimensions are shown.

nature of the algorithm, the computing time will likely still be longer than other clustering methods such as those implemented in Mclust. However, although the computational time is a deterrent to using the algorithm, there are benefits associated with using Bayesian inference to fit the mixture models. For instance, the incorporation of prior information, the ability to quantify uncertainty in parameter estimates, and increased flexibility in model specification are just some of the advantages offered by the use of Bayesian methods in the BMCD algorithm. As a result, the BMCD offers both advantages and drawbacks and an evaluation of these aspects is important to determine the suitability of employing the algorithm. Additionally, these disadvantages provide avenues for future exploration by investigating the reduction of computational time by using alternative methodologies such as Bayesian variational inference.

In both simulation studies, BMCD performed well when measurement error was present in the observed data and the dimension was high. Even when measurement error was absent, BMCD clustering was still able to perform relatively well, even outperforming popular clustering methods in many of the simulations. Furthermore, when applied to the Wisconsin Diagnostic Breast Cancer data set, BMCD identified both the true dimension and correct number of clusters within the data. Moreover, compared to models fit using the Mclust software, BMCD resulted in the highest ARI, indicating an accurate clustering partition. Therefore, BMCD may be useful in both simulations and real-world scenarios.

The BMCD algorithm provides a unique and effective way of fitting a GMM to dissimilarity data using Bayesian methods. Even when measurement error is present within the dissimilarity data, we showed that BMCD is effective at accurately capturing clusters. Our extension of the algorithm through the implementation of five additional models further increases the effectiveness of BMCD in high dimensions by reducing the number of parameters that must be estimated. Furthermore, these models may also be effective in capturing clusters of different shapes and sizes. Therefore, the BMCD algorithm can be used with a variety of data and is a viable alternative to heuristic clustering methods that are often used with dissimilarity data.

CRediT authorship contribution statement

Samuel Morrissette: Methodology, Software, Investigation, Formal analysis, Data curation, Writing – original draft, Visualization. **Saman Muthukumarana:** Conceptualization, Methodology, Writing – review & editing, Supervision, Funding acquisition, project administration. **Maxime Turgeon:** Conceptualization, Methodology, Writing – review & editing, Supervision, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix. Derivation of full conditionals for the unequal unrestricted model

The posterior conditional distribution of the component means and the covariance matrix are derived for the unequal unrestricted model presented in Section 2.1. For the other five models, the derivation of the posterior conditional distribution proceeds in a similar manner. Two relations will be used throughout the following derivation:

$$\sum_{i;z_i=k}(\mathbf{x}_i-\mu_k)(\mathbf{x}_i-\mu_k)^T = n_k(\mu_k-\bar{\mathbf{x}}_k)(\mu_k-\bar{\mathbf{x}}_k)^T + \sum_{i;z_i=k}(\mathbf{x}_i-\bar{\mathbf{x}}_k)(\mathbf{x}_i-\bar{\mathbf{x}}_k)^T \quad (5)$$

$$\begin{aligned}&\lambda_0(\mu_k-\mu_{k0})(\mu_k-\mu_{k0})^T + n_k(\mu_k-\bar{\mathbf{x}}_k)(\mu_k-\bar{\mathbf{x}}_k)^T = \\ &(\lambda_0+n_k)(\mu_k-\mu_k^*)(\mu_k-\mu_k^*)^T + \frac{\lambda_0 n_k}{\lambda_0+n_k}(\bar{\mathbf{x}}_k-\mu_{k0})(\bar{\mathbf{x}}_k-\mu_{k0})^T\end{aligned} \quad (6)$$

where $\mu_k^* = \frac{\lambda_0\mu_{k0}+n_k\bar{\mathbf{x}}_k}{\lambda_0+n_k}$. The proof of these relationships can be found in Appendix A of Fraley and Raftery (2005).

Using the unequal unrestricted model priors, the conditional joint posterior distribution is:

$$\begin{aligned}p(\mathbf{X},\mathbf{z},\boldsymbol{\theta},\boldsymbol{\pi}) &\propto p(\mathbf{X}|\mathbf{z},\boldsymbol{\theta})p(\mathbf{z}|\boldsymbol{\pi})p(\boldsymbol{\pi})p(\boldsymbol{\theta}) \\ &= \prod_{i=1}^{n}\prod_{j=1}^{G}\left[\mathcal{N}\left(\mathbf{x}_i|\mu_j,\boldsymbol{\Sigma}_j\right)^{I(z_i=j)}\pi_j^{I(z_i=j)}\right]\times \mathrm{Dir}\left(\boldsymbol{\pi}\right)\times\prod_{j=1}^{G}NIW(\theta_j),\end{aligned}$$

where the hyperparameters are omitted for brevity. The full conditional posterior distribution of $\theta_k = (\mu_k, \boldsymbol{\Sigma}_\mathbf{k})$ is given below:

$$\begin{aligned}p(\theta_k|\boldsymbol{\theta}_{-k},\mathbf{z},\mathbf{X},\boldsymbol{\pi}) &\propto \prod_{i;z_i=k}\left[\mathcal{N}\left(\mathbf{x}_i|\mu_k,\boldsymbol{\Sigma}_\mathbf{k}\right)\right]\times NIW\left(\mu_k,\boldsymbol{\Sigma}_\mathbf{k}\right) \\ &\propto |\boldsymbol{\Sigma}_\mathbf{k}|^{-\frac{n_k}{2}}\exp\left\{-\frac{1}{2}\sum_{i;z_i=k}(\mathbf{x}_i-\mu_k)^T\boldsymbol{\Sigma}_\mathbf{k}^{-1}(\mathbf{x}_i-\mu_k)\right\}\times \\ &|\boldsymbol{\Sigma}_\mathbf{k}|^{-\frac{\nu_0+p+2}{2}}\exp\left\{-\frac{1}{2}\mathrm{tr}(\Psi_0\boldsymbol{\Sigma}_\mathbf{k}^{-1})-\frac{\lambda_0}{2}(\mu_k-\mu_{k0})^T\boldsymbol{\Sigma}_\mathbf{k}^{-1}(\mu_k-\mu_{k0})\right\}\end{aligned}$$

$$= |\boldsymbol{\Sigma}_{\mathbf{k}}|^{-\frac{\nu_0+n_k+p+2}{2}} \times$$

$$\exp\left\{-\frac{1}{2}\mathrm{tr}\left[\boldsymbol{\Sigma}_{\mathbf{k}}^{-1}\left(\boldsymbol{\Psi}_0 + \sum_{i;z_i=k}(\mathbf{x}_i-\bar{\mathbf{x}}_k)(\mathbf{x}_i-\bar{\mathbf{x}}_k)^T\right)\right]\right\}\times$$

$$\exp\left\{-\frac{1}{2}\mathrm{tr}\left[\boldsymbol{\Sigma}_{\mathbf{k}}^{-1}\lambda_0(\mu_k-\mu_{k0})(\mu_k-\mu_{k0})^T + n_k(\mu_k-\bar{\mathbf{x}}_k)(\mu_k-\bar{\mathbf{x}}_k)^T\right]\right\} \quad \text{(by (5))}$$

$$= |\boldsymbol{\Sigma}_{\mathbf{k}}|^{-\frac{\nu_0+n_k+p+2}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\left[\boldsymbol{\Sigma}_{\mathbf{k}}^{-1}\left(\boldsymbol{\Psi}_0 + \sum_{i;z_i=k}(\mathbf{x}_i-\bar{\mathbf{x}}_k)(\mathbf{x}_i-\bar{\mathbf{x}}_k)^T + \right.\right.\right.$$

$$\left.\left.\left.\frac{\lambda_0 n_k}{\lambda_0+n_k}(\bar{\mathbf{x}}_k-\mu_{k0})(\bar{\mathbf{x}}_k-\mu_{k0})^T\right)\right]\right\} \times$$

$$\exp\left\{-\frac{\lambda_0+n_k}{2}\ \mathrm{tr}\left[\boldsymbol{\Sigma}_{\mathbf{k}}^{-1}(\mu_k-\mu_k^*)(\mu_k-\mu_k^*)^T\right]\right\} \quad \text{(by (6))}$$

This can be recognized as the kernel of a Normal Inverse Wishart distribution with parameters $\mu_k^*, \lambda_0+n_k, \nu_0+n_k$, and $\boldsymbol{\Psi}_0 + \sum_{i;z_i=k}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i-\bar{\mathbf{x}})^T + \frac{\lambda_0 n_k}{\lambda_0+n_k}$.

References

Banfield, J. D., & Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, *49*, 803.

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

Bijmolt, T. H., Wedel, M., & DeSarbo, W. S. (2020). Adaptive multidimensional scaling: Brand positioning based on decision sets and dissimilarity judgments. *Customer Needs and Solutions 2020 8:1*, *8*, 1–15.

Bimler, D. L., Kirkland, J., & Jameson, K. A. (2004). Quantifying variations in personal color spaces: Are there sex differences in color vision? *Color Research and Application*, *29*, 128–134.

Celeux, G. (1998). Bayesian inference for mixture: The label switching problem. *COMPSTAT*, 227–232.

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

Chen, Y., & Meltzer, P. S. (2005). Gene expression analysis via multidimensional scaling. *Current Protocols in Bioinformatics*, *10*(1), 7.11.1–7.11.9.

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

Digital Research Alliance of Canada (2023). Cedar. URL https://docs.alliancecan.ca/wiki/Cedar.

Dua, D., & Graff, C. (2017). *UCI machine learning repository*. University of California, Irvine, School of Information and Computer Sciences.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). *A density-based algorithm for discovering clusters in large spatial databases with noise* (pp. 226–231). AAAI Press.

Fitzgerald, L. F., & Hubert, L. J. (1985). Multidimensional scaling: Some possibilities for counseling psychology. *Journal of Counseling Psychology*, *34*, 469–480.

Fraley, C., & Raftery, A. E. (2005). *Bayesian Regularization for Normal Mixture Estimation and Model-Based Clustering: Technical Report 24, 486*, (pp. 155–181). Springer, Department of Statistics, Univerisity of Washington.

García-Escudero, L. A., Mayo-Iscar, A., & Riani, M. (2022). Constrained parsimonious model-based clustering. *Statistics and Computing*, *32*.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, *7*, 457–472.

Geman, S., & Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-6*, 721–741.

Gormley, I. C., Murphy, T. B., & Raftery, A. E. (2023). Model-based clustering. *Annual Review of Statistics and Its Application*, *10*, 573–595.

Hastings, W. (1970). Monte Carlo sampling methods sing Markov chains and their applications. *Biometrika*, *57*, 97.

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

Jaworska, N., & Chupetlovska-Anastasova, A. (2009). A review of multidimensional scaling (MDS) and its utility in various psychological domains. *Tutorials in Quantitative Methods for Psychology*, *5*, 1–10.

Kim, K., Baik, H., Jang, C. S., Roh, J. K., Eskin, E., & Han, B. (2019). Genomic GPS: Using genetic distance from individuals to public data for genomic analysis without disclosing personal genomes. *Genome Biology*, *20*.

Kuhn, H. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*, 83–97.

McEntee, S. (2004). World order and welfare provision : A multidimensional scaling analysis. *International Journal of Sociology*, *34*, 52–70.

Melnykov, V., Chen, W. C., & Maitra, R. (2012). MixSim: An R package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software*, *51*.

Oh, M. S., & Lee, E. K. (2022). BayMDS: An r package for Bayesian multidimensional scaling and choice of dimension. *Applied Psychological Measurement*, *46*, 250.

Oh, M.-S., & Raftery, A. E. (2001). Bayesian multidimensional scaling and choice of dimension. *Journal of the American Statistical Association*, *96*.

Oh, M. S., & Raftery, A. E. (2007). Model-based clustering with dissimilarities: A Bayesian approach. *Journal of Computational and Graphical Statistics*, *16*, 559–585.

R Core Team (2023). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

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

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

Street, W. N., Wolberg, W. H., & Mangasarian, O. L. (1993). Nuclear feature extraction for breast tumor diagnosis. *Biomedical Image Processing and Biomedical Visualization*, *1905*, 861–870.

Yeung, K. Y., Fraley, C., Murua, A., Raftery, A. E., & Ruzzo, W. L. (2001). Model-based clustering and data transformations for gene expression data. *Bioinformatics*, *17*, 977–987.