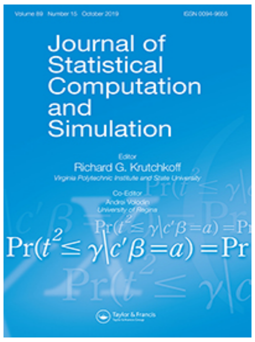


Journal of Statistical Computation and Simulation

ISSN: 0094-9655 (Print) 1563-5163 (Online) Journal homepage: https://www.tandfonline.com/loi/gscs20

A data-driven selection of the number of clusters in the Dirichlet allocation model via Bayesian mixture modelling

E. F. Saraiva, C. A. B. Pereira & A. K. Suzuki

To cite this article: E. F. Saraiva, C. A. B. Pereira & A. K. Suzuki (2019) A data-driven selection of the number of clusters in the Dirichlet allocation model via Bayesian mixture modelling, Journal of Statistical Computation and Simulation, 89:15, 2848-2870, DOI: 10.1080/00949655.2019.1643345

To link to this article: https://doi.org/10.1080/00949655.2019.1643345

JOURNAL OF STATISTICAL COMPUTATION AND SIMULATION
2019, VOL. 89, NO. 15, 2848–2870
https://doi.org/10.1080/00949655.2019.1643345

Taylor & Francis
Taylor & Francis Group

Check for updates

A data-driven selection of the number of clusters in the Dirichlet allocation model via Bayesian mixture modelling

E. F. Saraiva[a], C. A. B. Pereira[a,b] and A. K. Suzuki[c]

[a]Mathematics Institute, Federal University of Mato Grosso do Sul, Campo Grande, Brazil; [b]Institute of Mathematics and Statistics, University of São Paulo, São Paulo, Brazil; [c]Sciences Institute of Mathematics and Computers, University of São Paulo, São Carlos, Brazil

ABSTRACT

In this paper, we consider a Bayesian mixture model that allows us to integrate out the weights of the mixture in order to obtain a procedure in which the number of clusters is an unknown quantity. To determine clusters and estimate parameters of interest, we develop an MCMC algorithm denominated by sequential data-driven allocation sampler. In this algorithm, a single observation has a non-null probability to create a new cluster and a set of observations may create a new cluster through the split-merge movements. The split-merge movements are developed using a sequential allocation procedure based in allocation probabilities that are calculated according to the Kullback–Leibler divergence between the posterior distribution using the observations previously allocated and the posterior distribution including a 'new' observation. We verified the performance of the proposed algorithm on the simulated data and then we illustrate its use on three publicly available real data sets.

ARTICLE HISTORY
Received 2 February 2019
Accepted 10 July 2019

KEYWORDS
Mixture model; Bayesian approach; Gibbs sampling; Metropolis–Hastings; split-merge update; Kullback–Leibler divergence

2010 MATHEMATICS SUBJECT CLASSIFICATION
62M05

1. Introduction

Clustering problems occur in many real-world phenomena where the main objective is to group the observed data into disjoint groups (called clusters). In general, in the clustering methods such as k-means [1] or hierarchical cluster [2], the clusters are determined according to some predefined distance measure such as Euclidean distance or Mahalanobis distance. Besides, these methods require that the number of clusters is known a priori.

Due to its simplicity, these methods are used in many applications. For instance, Sturn et al. [3] consider the k-means and the hierarchical clustering for analysis of microarray data, Wride et al. [4] consider the k-means to investigate genes differentially expressed, Oyelade et al. [5] consider the k-means algorithm for the prediction of Students' Academic Performance, Peterson et al. [6] present an hybrid non-parametric clustering approach based on k-means and hierarchical clustering to identify general-shaped clusters, among others.

CONTACT E. F. Saraiva ✉ erlandson.saraiva@ufms.br

Supplemental data for this article can be accessed here. https://doi.org/10.1080/00949655.2019.1643345

However, as discussed by Oh and Raftery [7] 'these methods are not based on standard principles of statistical inference and they do not provide a statistically based method for choosing the number of clusters'. Thus, an alternative is to consider a clustering procedure using a probabilistic model. In this way, the obtained clusters can be interpreted from a statistical point of view [8].

Under a probabilistic approach, the main clustering methods are based on the use of a mixture model, see for example [9–12]. In this model, each component of the mixture represents a cluster and, in general, clusters are determined by the EM algorithm [13]. However, for the use of the EM algorithm, the number of cluster also need to be known a priori. For the cases where the number of cluster is unknown, the number of cluster is determined comparing fitted models with different number of clusters using some model selection criterion, such as AIC [14,15] or BIC [16]. A similar strategy is adopted in the Bayesian approach, considering the DIC [17] as a model selection criterion, see Celeux et al. [18].

This can be seen as a drawback to be overcome, since in practice it may be very tedious to fit several models and afterwards compare them according to a model selection criterion. Also, in these cases, the estimation depends on iterative methods which may not converge imposing additional difficulties to the process. Therefore, a practical and efficient computational algorithm to estimate the number of cluster jointly with the component-specific parameters is desirable. Under this scenario, the Bayesian approach has been successful, in special, due the reversible-jump Markov Chain Monte Carlo (MCMC) algorithm proposed by Richardson and Green [19] in the context of Gaussian mixture models. However, one difficulty frequently encountered for implementing a reversible-jump algorithm is the construction of efficient transitions proposals that lead to a reasonable acceptance rate.

In this paper, we consider a Bayesian mixture model with a Dirichlet prior distribution for the weights of the mixture that allows us to integrate them out in order to obtain a procedure in which the number of cluster is a random quantity. In order to determine clusters and estimate parameters of interest jointly, we develop an MCMC algorithm denominated by sequential data-driven allocation sampler (SDAS). In this algorithm, the latent allocation variables are updated using two steps. In the first one, each observation is allocated to a cluster via Gibbs sampling algorithm and there is a non-null probability of a single observation to define a new cluster; and in the second one, a split-merge step is used to create a new cluster using a set of observations. In this way, increasing the mixing of the Markov chain in relation to the number of cluster.

The split-merge movements are developed using a sequential allocation procedure based on allocation probabilities that are calculated according to the Kullback–Leibler divergence [20] between the posterior distribution of a specific parameter using the observations previously allocated and this posterior distribution including a 'new' observation. The advantage of using the Kullback–Leibler divergence is that it allows to calculate the allocation probabilities using the effect of the 'new' observation in the posterior distribution for the component parameter. Using an augmented parametrization through the introduction of densities linking [21,22], the acceptance probability for these both movements are given by the Metropolis–Hastings acceptance probability. Conditional on the allocation of the observations, the parameters of the clusters are updated from their conditional posterior distributions.

In order to verify the performance of SDAS, we developed a simulation study considering that clusters are generated from a mixture of univariate normal distributions. Following Saraiva et al. [23], we present the performance of SDAS in terms of posterior probability for the number of clusters, convergence, mixing and autocorrelation.

To illustrate the use of the SDAS algorithm we apply it to three real datasets. The first real data set is the benchmark Galaxy data, while second and third are the publicly available data set on Enzyme and Acidity, respectively.

The remainder of the paper is structured as follows. In Section 2, we describe the Bayesian mixture model for data clustering. In Section 3, we develop the SDAS algorithm. In Section 4, the proposed sampler is applied to simulated data sets to access its performance and to real data sets to illustrate its use. Section 5 concludes the paper with final remarks. Additional details are provided in the supplementary material, denoted by prefix ‘SM’ when referred to in this paper.

2. Bayesian mixture modelling for data clustering

Consider a population composed by k subpopulations, such that, the sampling units are homogeneous with respect to the characteristic under study within the subpopulation and heterogeneous among the subpopulations. Let $w_1, \ldots, w_k$ be the relative frequencies of each subpopulation in relation to the overall population, for $0 \leq w_j \leq 1$ and $\sum_{j=1}^{k} w_j = 1$. Assume that each subpopulation j is modelled by a probability distribution $F(\theta_j)$ indexed by parameter θ_j (scalar or vector), for $j = 1, \ldots, k$.

Suppose that the sampling process from this population consists of choosing a subpopulation j with probability w_j and then sample a Y_i value of this subpopulation, for $j = 1, \ldots, k$ and $i = 1, \ldots, n$ where n is the sample size. Then we can represent each sample unit by the pair (Y_i, c_i), where c_i is an indicator variable that assume a value of the set $\{1, \ldots, k\}$ with probabilities $\{w_1, \ldots, w_k\}$, respectively. Therefore, we have that

$$(Y_i | c_i = j, \theta_j) \sim F(\theta_j) \quad \text{and} \quad P(C_i = j | \mathbf{w}) = w_j,$$

where $\mathbf{w} = (w_1, \ldots, w_k)$, for $i = 1, \ldots, n$ and $j = 1, \ldots, k$.

However, in many practical problems such as clustering problems, the indicator variables are non-observable (also denominated by latent variables). Thus, the probability of i-th observation coming from subpopulation j is w_j and the marginal probability density function for $Y_i = y_i$ is given by

$$f(y_i | \boldsymbol{\theta}_k, \mathbf{w}) = \sum_{j=1}^{k} w_j f(y_i | \theta_j), \tag{1}$$

where $f(y_i|\theta_j)$ is the probability density function of $F(\theta_j)$, $\boldsymbol{\theta}_k = (\theta_1, \ldots, \theta_k)$ is the whole vector of parameters and $\mathbf{w} = (w_1, \ldots, w_k)$ are the weights, for $i = 1, \ldots, n$ and $j = 1, \ldots, k$. Model (1) is denominated in the literature by finite mixture model, see for example McLachlan and Basford [9], McLachlan and Peel [11] and Fruhwirth-Schnatter [24].

As one can note, the finite mixture model is a natural probabilistic approach for data clustering. However, as the model in (1) is a population model, then given an observed sample $\mathbf{y} = (y_1, \ldots, y_n)$ not all k components may have observations in the sample and we may have empty components. In this case, we have that the number of clusters (i.e. non-empty components) is smaller than the number of components k. Besides, under the data clustering framework, the main interest is in the configuration $\mathbf{c}$ due this define the clusters and the number of clusters; hereafter denoted by $k_{\mathbf{c}}$.

Without loss of generality, consider that clusters are labelled from 1 to $k_{\mathbf{c}}$ and that $\boldsymbol{\theta}_{k_{\mathbf{c}}} = (\theta_1, \dots, \theta_{k_{\mathbf{c}}})$ are the component parameters associated to the $k_{\mathbf{c}}$ clusters. Our interest is to estimate $k_{\mathbf{c}}$ and $\boldsymbol{\theta}_{k_{\mathbf{c}}}$ in a joint way.

Before we proceed, some remarks about the label switching are necessary. Note that, the cluster labels $j = 1, \dots, k_{\mathbf{c}}$ are not uniquely determined and a permutation of the labels would lead to the same model. Since our interest lies in inferences on clusters, the non-identifiability of labels would cause a problem in posterior computation and allocation probabilities are useless for partitioning the observations [25]. Thus, following Richardson and Green [19] and Saraiva et al. [23], we impose restrictions on the class of component means of the clusters to get identifiability, i.e. we assume that $\mu_1, \dots, \mu_{k_{\mathbf{c}}}$ are the component means for clusters and that $\mu_1 < \dots < \mu_{k_{\mathbf{c}}}$. However, it does not prevent the MCMC algorithm described in the next Section for being applicable to another labelling criterion. For further discussion and additional references about label switching, see Stephens [25], Jasra [26] and their references.

2.1. Bayesian approach

In order to estimate $k_{\mathbf{c}}$ and $\mathbf{c}$ jointly with component parameters $\boldsymbol{\theta}_{k_{\mathbf{c}}}$, we assume a Bayesian approach. For this, let $(\mathbf{y}, \mathbf{c})$ be the complete data, where $\mathbf{y} = (y_1, \dots, y_n)$ is the vector of independent observations and $\mathbf{c} = (c_1, \dots, c_n)$ is the vector of latent indicator variables, with $\mathbf{y}$ and $\mathbf{c}$ being paired. We then model the complete data $(\mathbf{y}, \mathbf{c})$ using the following hierarchical Bayesian model

$$\begin{aligned} Y_i|c_i = j, \boldsymbol{\theta}, k &\sim F(\theta_j), \\ c_i|\mathbf{w}, k &\sim Discrete(w_1, \dots, w_k), \\ \theta_j &\sim G(\eta_j), \\ \mathbf{w}|\gamma, k &\sim Dirichlet\left(\frac{\gamma}{k}, \dots, \frac{\gamma}{k}\right), \end{aligned} \tag{2}$$

where $G(\eta_j)$ is the prior distribution for component parameters θ_j, η_j (scalar or vector) are the hyperparameters, for $j = 1, \dots, k$, and $Dirichlet(\gamma/k, \dots, \gamma/k)$ represents the Dirichlet distribution with parameter γ/k, $\gamma > 0$, and probability density function

$$\pi(\mathbf{w}|\gamma, k) = \frac{\Gamma(\gamma)}{[\Gamma(\gamma)]^k} \prod_{j=1}^{k} w_j^{\gamma - 1}. \tag{3}$$

From the second line of model (2), we have that

$$\pi(\mathbf{C} = \mathbf{c}|\mathbf{w}, k) = \prod_{j=1}^{k} w_j^{n_j}, \tag{4}$$

where n_j is the number of observations assigned to component j, for $j = 1, \dots, k$.

Taking the product of densities in Equations (3) and (4) and integrating out the mixing proportions, we can write the joint probability of $\mathbf{c}$ as

$$\pi(\mathbf{C}=\mathbf{c}|\gamma,k)=\frac{\Gamma(\gamma)}{\Gamma(n+\gamma)}\prod_{j=1}^{k}\frac{\Gamma\left(n_j+\frac{\gamma}{k}\right)}{\Gamma\left(\frac{\gamma}{k}\right)}. \tag{5}$$

Besides, using the Dirichlet integral, the conditional probability for a single indicator variable c_i given all others, denoted by $\mathbf{c}_{-i}=(c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_n)$, is given by

$$\pi(C_i=j|\mathbf{c}_{-i},\gamma,k)=\frac{n_{j,-i}+\frac{\gamma}{k}}{n+\gamma-1}, \tag{6}$$

where $n_{j,-i}$ is the number of observations assigned to component j excluding the i-th observation, for $i=1,\ldots,n$ and $j=1,\ldots,k$.

As our main interest lies in the number of clusters $k_{\mathbf{c}}$, we then eliminate the number of components k from analysis by considering $k\rightarrow\infty$. Thus, the Equations in (5) and (6) are now given by

$$\pi(\mathbf{C}=\mathbf{c}|\gamma)=\frac{\Gamma(\gamma)}{\Gamma(n+\gamma)}\prod_{j=1}^{k_{\mathbf{c}}}\Gamma(n_j) \tag{7}$$

and

$$\pi(C_i=j|\mathbf{c}_{-i},\gamma)=\frac{n_{j,-i}}{n+\gamma-1} \tag{8}$$

for $i=1,\ldots,n$ and $j=1,\ldots,k_{\mathbf{c}}$, where $k_{\mathbf{c}}$ is the number of clusters defined by configuration $\mathbf{c}$. Under this approach, there exist the probability of the i-th observation to be allocated to one of the others infinite components, which is given by

$$\pi(C_i=j|\mathbf{c}_{-i},\gamma)=\frac{\gamma}{n+\gamma-1}. \tag{9}$$

This equation is the probability of the i-th observation to define a new cluster, for $i=1,\ldots,n$. Equations in (8) and (9) define a model equivalent to the Dirichlet process mixture model, see for example, Ferguson [27], Antoniak [28] and Jain and Neal [29].

From the first line of model (2) and Equation in (7), the joint probability of the complete data $(\mathbf{y},\mathbf{c})$ is given by

$$P(\mathbf{Y}=\mathbf{y},\mathbf{C}=\mathbf{c}|\boldsymbol{\theta}_{k_{\mathbf{c}}},\gamma)=\prod_{j=1}^{k_{\mathbf{c}}}\left(\prod_{D_j}f(y_i|\theta_j)\right)\pi(\mathbf{C}=\mathbf{c}|\gamma), \tag{9a}$$

where $D_j=\{y_i;c_i=j\}$ is the set of observations allocated to component j, for $i=1,\ldots,n$ and $j=1,\ldots,k_{\mathbf{c}}$.

From the third line of model (2), the joint prior distribution for component parameters of the $k_{\mathbf{c}}$ clusters is given by $\pi(\boldsymbol{\theta}_{k_{\mathbf{c}}}|\boldsymbol{\eta})=\prod_{j=1}^{k_{\mathbf{c}}}\pi_G(\theta_j|\eta_j)$, where $\pi_G(\theta_j|\eta_j)$ is the probability density function of the prior distribution $G(\eta_j)$ and $\boldsymbol{\eta}=(\eta_1,\ldots,\eta_{k_{\mathbf{c}}})$, for $j=1,\ldots,k_{\mathbf{c}}$.

Using the Bayes theorem, the joint posterior distribution upon which inference is based is given by

$$\pi(\boldsymbol{\theta}_{k_{\mathbf{c}}}, \mathbf{c}|\mathbf{y}, \gamma) \propto P(\mathbf{y}|\mathbf{c}, \boldsymbol{\theta}_{k_{\mathbf{c}}})\pi(\mathbf{c}|\gamma)\pi(\boldsymbol{\theta}_{k_{\mathbf{c}}}|\boldsymbol{\eta}),$$

where $P(\mathbf{y}|\mathbf{c}, \boldsymbol{\theta}_{k_{\mathbf{c}}})\pi(\mathbf{c}|\gamma) = L(\boldsymbol{\theta}_{k_{\mathbf{c}}}|\mathbf{y}, \mathbf{c})$ is the complete-data likelihood function for component parameters $\boldsymbol{\theta}_{k_{\mathbf{c}}}$, which is equal to the sampling distribution given in (9a) , regarded as a function of the unknown parameters $\boldsymbol{\theta}_{k_{\mathbf{c}}}$.

2.2. Conditional posterior distributions

Updating Equation (8) by $f(y_i|\theta_j)$, the conditional probability for $C_i = j$, for some component j, so that $n_{j,-i} > 0$, is

$$\pi_{ij} = \pi(C_i = j|y_i, \theta_j, \mathbf{c}_{-i}, \gamma) = \frac{n_{j,-i}}{n + \gamma - 1} f(y_i|\theta_j), \tag{10}$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, k_{\mathbf{c}_{-\mathbf{i}}}$, where $k_{\mathbf{c}_{-i}}$ is the number of cluster excluding the observation y_i. At this point, it is important to note that, if an observation y_i is allocated in a component j, $c_i = j$, and $n_j > 1$, then $n_{j,-i} \geq 1$ and $k_{\mathbf{c}_{-i}} = k_{\mathbf{c}}$. But, if $c_i = j$ and $n_j = 1$, then $n_{j,-i} = 0$ and $k_{\mathbf{c}_{-i}} = k_{\mathbf{c}} - 1$.

We need now to define the probability of an observation y_i to create a new cluster j^*, for $j^* = k_{\mathbf{c}_{-i}} + 1$. For this, we integrate parameters out. Thus, the conditional posterior probability for $C_i = j^*$ is

$$\pi_{ij^*} = \pi(C_i = j^*|y_i, \mathbf{c}_{-i}, \gamma, \eta_{j^*}) = \frac{\gamma}{n + \gamma - 1} \int f(y_i|\theta_{j^*})\pi_G(\theta_{j^*}|\eta_{j^*})d\theta_{j^*} \tag{11}$$

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

Conditional on a configuration $\mathbf{c}$, we have $k_{\mathbf{c}}$ clusters. The conditional posterior distribution for θ_j is given by

$$\pi(\theta_j|\mathbf{y}, \mathbf{c}, k) \propto L(\theta_j|D_j)\pi_G(\theta_j|\eta_j), \tag{12}$$

where $L(\theta_j|D_j) = \prod_{D_j} f(y_i|\theta_j)$ is the likelihood function for component j, for $i = 1, \ldots, n$ and $j = 1, \ldots, k_{\mathbf{c}}$.

Thus, we update parameters of interest according to Algorithm 1.

Although Algorithm 1 is visually attractive it may be inefficient in situations where clusters have near means. This happens because the algorithm updates only one latent indicator variable at a time and a new cluster may be created based on only one observation, according to Equation in (11). Consequently, it may lead to a poor exploration of observation clusters and the algorithm may be trapped in local modes. Therefore, in order to avoid these problems and increase the mixing of the Markov chain in relation to the number of clusters, we introduce a data-driven split-merge step within this algorithm.

Algorithm 1 Let the state of the Markov chain consist of $\mathbf{c} = (c_1, \ldots, c_n)$ and $\boldsymbol{\theta}_{k_\mathbf{c}} = (\theta_1, \ldots, \theta_{k_\mathbf{c}})$. For l-th iteration of the algorithm do as follows. For $i = 1, \ldots, n$:

```
(1) remove $c_i$ from current state $\mathbf{c}$, obtaining $\mathbf{c}_{-i}$ and $k_{\mathbf{c}_{-i}}$;
(2) generate an auxiliary variable $\mathbf{Z}_i = (Z_{i1}, \ldots, Z_{ik_\mathbf{c}}) \sim Multinomial(1, \mathbf{P}_i)$, where $\mathbf{P}_i = (\pi_{i1}, \ldots, \pi_{ik_{\mathbf{c}_{-i}}}, \pi_{ij^*})$ for $\pi_{ij}$ given in (10) and $\pi_{ij^*}$ given in (11), for $j = 1, \ldots, k_{\mathbf{c}_{-i}}$ and $j^* = k_{\mathbf{c}_{-1}} + 1$;
(3) If $Z_{ij} = 1$, for $j \in \{1, \ldots, k_{\mathbf{c}_{-i}}\}$, set up $c_i = j$ and do $n_j = n_{j,-i} + 1$;
(4) If $Z_{ij^*} = 1$ do $n_{j^*} = 1$ and $k_\mathbf{c} = k_{\mathbf{c}_{-i}} + 1$. Generate a value for the component parameter $\theta_{j^*}$ of the new cluster from the posterior distribution $\pi(\theta_{j^*}|y_i)$. Relabel the $k_\mathbf{c}$ clusters in order to maintain the adjacency condition. If the component mean $\mu_{j^*}$ of the new cluster is so that:
    (a) $\mu_{j^*} = \min_{1 \le j \le k_\mathbf{c}} \mu_j$, then do $j^* = 1$ and relabel all other clusters doing $j + 1$;
    (b) $\mu_{j^*} = \max_{1 \le j \le k_\mathbf{c}} \mu_j$, then do $j^* = k_\mathbf{c}$ and keep all other clusters labels;
    (c) $\mu_j < \mu_{j^*} < \mu_{j+1}$, for $j \neq \{1, k_\mathbf{c}\}$, then do $j^* = j + 1$ and relabel all other clusters $j' \ge j + 1$ doing $j' = j' + 1$.
(5) Conditional on configuration $\mathbf{c}$ update the cluster parameters $\boldsymbol{\theta}_{k_\mathbf{c}} = (\theta_1, \ldots, \theta_{k_\mathbf{c}})$. For this, generate $\theta_j$ from its posterior distribution, $\pi(\theta_j|\mathbf{y}, \mathbf{c}, \gamma)$ given in (12), for $j = 1, \ldots, k_\mathbf{c}$;
(6) Accept the updated values, $\boldsymbol{\theta}_{k_\mathbf{c}}^{updated}$, only if adjacency condition for component parameters of the clusters is met, i.e. if $\mu_1^{updated} < \ldots < \mu_{k_\mathbf{c}}^{updated}$. Otherwise, keep $\boldsymbol{\theta}_{k_\mathbf{c}}^{updated} = \boldsymbol{\theta}_{k_\mathbf{c}}$.
```

3. Data-driven split and merge movements

In this section, we describe as to insert within the MCMC Algorithm 1 a split-merge procedure. This procedure is data-driven and changes the number of clusters in the neighbourhood $k_\mathbf{c} + 1$ ou $k_\mathbf{c} - 1$.

As the maximum number of the cluster that we can have is n, $k_\mathbf{c} = n$, then to maintain the detailed balance equation when we propose the split-merge movements, consider the following alternative parametrization obtained by augmenting $\boldsymbol{\theta}_{k_\mathbf{c}}$ to

$$\boldsymbol{\theta}_n = (\underbrace{\theta_1, \ldots, \theta_{k_\mathbf{c}}}_{\boldsymbol{\theta}_{k_\mathbf{c}}}, \underbrace{\theta_{k_\mathbf{c}+1}, \ldots, \theta_n}_{\boldsymbol{\theta}_0}) = (\boldsymbol{\theta}_{k_\mathbf{c}}, \boldsymbol{\theta}_0),$$

where $\boldsymbol{\theta}_0$ are component parameters of $n - k_\mathbf{c}$ empty components, labelled $k_\mathbf{c} + 1, \ldots, n$, which can be viewed as 'potential' components to be used as cluster locations. Also, assume that $(\theta_{k_\mathbf{c}+1}, \ldots, \theta_n) \in \boldsymbol{\theta}_0$ are *a priori* independent of one another and also of $\boldsymbol{\theta}_{k_\mathbf{c}}$, i.e. $\pi(\boldsymbol{\theta}_0|k_\mathbf{c}) = \prod_{j=k_\mathbf{c}+1}^{n} \pi_G(\theta_j|\eta_j)$. This 'prior' for the parameters $\boldsymbol{\theta}_0$ is called by Carlin and Chib [30] as pseudo-priors and by Godsill [21] and Dellaportas et al. [22] as densities linking.

Under this augmented parametrization, the full posterior distribution is given by

$$\pi(\boldsymbol{\theta}_n, \mathbf{c}|\mathbf{y}, \gamma) \propto P(\mathbf{y}|\mathbf{c}, \boldsymbol{\theta}_n)\pi(\mathbf{c}|\gamma)\pi(\boldsymbol{\theta}_{k_\mathbf{c}}|\boldsymbol{\eta}) \left(\prod_{j=k_\mathbf{c}+1}^{n} \pi_G(\theta_j|\eta_j) \right),$$

where $L(\boldsymbol{\theta}_n|\mathbf{y},\mathbf{c},\gamma) = P(\mathbf{Y}=\mathbf{y}|\mathbf{c},\boldsymbol{\theta}_n)\pi(\mathbf{C}=\mathbf{c}|\gamma)$ is the complete-data likelihood function for $\boldsymbol{\theta}_n$, which remains equals to Equation (9a) , regarded as a function of the unknown parameters $\boldsymbol{\theta}_n$, due the components $k_\mathbf{c}+1,\dots,n$ to be empty components.

Thus, let $\boldsymbol{\Phi} = (\boldsymbol{\theta}_{k_\mathbf{c}},\boldsymbol{\theta}_0,\mathbf{c},k_\mathbf{c})$ be the current state of the MCMC algorithm and denote the split and merge proposals by $\boldsymbol{\Phi}^{sp} = (\boldsymbol{\theta}_{k_\mathbf{c}^{sp}},\boldsymbol{\theta}_0^{sp},\mathbf{c}^{sp},k_\mathbf{c}^{sp})$ and $\boldsymbol{\Phi}^{me} = (\boldsymbol{\theta}_{k_\mathbf{c}^{me}},\boldsymbol{\theta}_0^{me},\mathbf{c}^{me},k_\mathbf{c}^{me})$, respectively; where

$$\begin{aligned} k_\mathbf{c}^{sp} &= k_\mathbf{c}+1, & k_\mathbf{c}^{me} &= k_\mathbf{c}-1,\\ \boldsymbol{\theta}_{k_\mathbf{c}^{sp}} &= \left(\theta_1^{sp},\dots,\theta_{k_\mathbf{c}}^{sp},\theta_{k_\mathbf{c}+1}^{sp}\right), & \boldsymbol{\theta}_{k_\mathbf{c}^{me}} &= \left(\theta_1^{me},\dots,\theta_{k_\mathbf{c}-1}^{me}\right)\\ \boldsymbol{\theta}_0^{sp} &= \left(\theta_{k_\mathbf{c}+2}^{sp},\dots,\theta_n^{sp}\right), & \boldsymbol{\theta}_0^{me} &= \left(\theta_{k_\mathbf{c}}^{me},\dots,\theta_n^{me}\right). \end{aligned}$$

As the dimension of the parametric space of the component parameters $\boldsymbol{\theta}_n = (\boldsymbol{\theta}_{k_\mathbf{c}},\boldsymbol{\theta}_0)$ do not change when we propose a split or a merge movement, then the acceptance probability for both movements is given by the Metropolis–Hastings acceptance probability [31], i.e. $\Psi[\boldsymbol{\Phi}^*|\boldsymbol{\Phi}] = \min(1,A^*)$, where

$$A^* = \frac{P(\mathbf{y}|\mathbf{c}^*,\boldsymbol{\theta}_{k_\mathbf{c}^*},\boldsymbol{\theta}_0^*)}{P(\mathbf{y}|\mathbf{c},\boldsymbol{\theta}_{k_\mathbf{c}},\boldsymbol{\theta}_0)}\,\frac{\pi(\mathbf{c}^*|\gamma)}{\pi(\mathbf{c}|\gamma)}\,\frac{\pi(\boldsymbol{\theta}_{k_\mathbf{c}^*}|\boldsymbol{\eta})}{\pi(\boldsymbol{\theta}_{k_\mathbf{c}}|\boldsymbol{\eta})}\,\frac{\prod_{j=k_{\mathbf{c}^*}+1}^{n}\pi_G(\theta_j|\eta_j)}{\prod_{j=k_\mathbf{c}+1}^{n}\pi_G(\theta_j|\eta_j)}\,\frac{q[\boldsymbol{\Phi}|\boldsymbol{\Phi}^*]}{q[\boldsymbol{\Phi}^*|\boldsymbol{\Phi}]}, \tag{13}$$

where '$*$' means either a split or a merge, $q[\cdot]$ is the transition proposal which is obtained by a split or a merge depending on the type of proposal.

Once given the mathematical expression for the acceptance probability for the split-merge movements, consider

$$P_{sp|k_\mathbf{c}} = \begin{cases} 0, & \text{if } k_\mathbf{c} = n;\\ 0,5, & \text{para } 1 < k_\mathbf{c} < n\\ 1, & \text{if } k_\mathbf{c} = 1; \end{cases} \quad \text{and} \quad P_{me|k_\mathbf{c}} = \begin{cases} 1, & \text{if } k_\mathbf{c} = n;\\ 0,5, & \text{para } 1 < k_\mathbf{c} < n,\\ 0, & \text{if } k_\mathbf{c} = 1 \end{cases} \tag{14}$$

be the probabilities of proposing a split and a merge movement, respectively, with $P_{sp|k_\mathbf{c}} + P_{me|k_\mathbf{c}} = 1$.

3.1. Split movement

Given the choice of a split movement, select at random a cluster D_j composed by at least two observations. For this, generate an auxiliary vector $\mathbb{I} = (\mathbb{I}_1,\dots,\mathbb{I}_{k_\mathbf{c}}) \sim \mathit{Multinomial}(1,\mathbb{P})$, where $\mathbb{P} = (\mathbb{P}_1,\dots,\mathbb{P}_{k_\mathbf{c}})$ for $\mathbb{P}_j = 1/\kappa_2$ if $n_j \geq 2$ and $\mathbb{P}_j = 0$ otherwise, in which, κ_2 is the number of clusters with $n_j \geq 2$, for $j = 1,\dots,k_\mathbf{c}$. If $\mathbb{I}_j = 1$, then propose a split of the cluster D_j as follows:

(i) Do $D_{j_1} = \{\min(D_j)\}$, $D_{j_2} = \{\max(D_j)\}$ and $n_{j_1} = n_{j_2} = 1$;
(ii) In a sequential way, allocate all other observations $y_i \in D_j$ to D_{j_1} or D_{j_2} through the following steps:

(a) Calculate the allocation probability

$$P_{j_1}(y_i) = \frac{KL(D_{j2}^*, D_{j_2})}{KL(D_{j1}^*, D_{j_1}) + KL(D_{j2}^*, D_{j_2})},$$

where $D_m^* = D_m \cup \{y_i\}$ and $KL(D_m^*; D_m)$ is the Kullback–Leibler divergence between $\pi(\theta_m|D_m^*)$ and $\pi(\theta_m|D_m)$, for $m = j_1, j_2$. The KL divergence is used to quantify the effect of the observation y_i in the posterior distribution for θ_m. Thus, if $KL(D_{j2}^*; D_{j_2}) > KL(D_{j1}^*, D_{j_1})$ meaning that y_i is more similar to the observations allocated to D_{j_1}.

(b) Generate $\mathbb{I}_i \sim Bernoulli(P_{j_1}(y_i))$. If $\mathbb{I}_i = 1$, then do $D_{j_1} = \{D_{j_1}\} \cup \{y_i\}$, $n_{j_1} = n_{j_1} + 1$ and $c_i^{sp} = j_1$. Otherwise, do $D_{j_2} = \{D_{j_2}\} \cup \{y_i\}$, $n_{j_2} = n_{j_2} + 1$ and $c_i^{sp} = j_2$.

The probability of configuration D_{j_1} and D_{j_2} is

$$P_{alloc}^{sp} = \prod_{y_i \in D_{j_1}} P_{j_1}(y_i) \prod_{y_i \in D_{j_2}} P_{j_2}(y_i).$$

In order to obtain the configuration $\mathbf{c}^{sp} = (c_1^{sp}, \ldots, c_n^{sp})$, we consider the following relabelling procedure:

(i) if $c_i = j'$ for $j' < j$, then maintain $c_i^{sp} = c_i$;
(ii) if $c_i = j'$, for $j' > j$, then do $c_i^{sp} = c_i + 1$;
(iii) if $c_i^{sp} = j_1$, then do $j_1 = j$ and $c_i^{sp} = j$;
(iv) if $c_i^{sp} = j_2$, then do $j_2 = j + 1$ and $c_i^{sp} = j + 1$;

for $i = 1, \ldots, n$ and $j \in \{1, \ldots, k_{\mathbf{c}}\}$.

Conditional on configuration $\mathbf{c}^{sp}$ we have $k_{\mathbf{c}}^{sp} = k_{\mathbf{c}} + 1$ clusters. The new vector of parameters is given by $\boldsymbol{\theta}_n^{sp} = (\boldsymbol{\theta}_{k_{\mathbf{c}}^{sp}}, \boldsymbol{\theta}_0^{sp})$, where $\theta_{j'}^{sp} = \theta_{j'}\ \forall\, j' < j$ and $\theta_{j'}^{sp} = \theta_{j'-1}\ \forall\, j' > j$, for $\theta_{j'} \in \boldsymbol{\theta}$ and $j' \in \{1, \ldots, n\} \setminus \{j_1, j_2\}$ with $j_1 = j$ and $j_2 = j + 1$. For the two new clusters generate candidate-values θ_{j1}^{sp} and θ_{j2}^{sp} from posterior distributions $\pi(\theta_{j_1}|D_{j1}, \eta_{j1})$ and $\pi(\theta_{j_2}|D_{j_2}, \eta_{j_2})$, respectively.

At this point we must check if the adjacency condition is met, i.e. if $\mu_{j1-1}^{sp} < \mu_{j1}^{sp} < \mu_{j2}^{sp} < \mu_{j2+1}^{sp}$. In the case where it is not, the proposal is rejected because the movement may not be reversible by the merge proposal.

If the adjacency condition is met, the transition probability for the split proposal is given by

$$q[\boldsymbol{\Phi}^{sp}|\boldsymbol{\Phi}] = P_{sp|k_{\mathbf{c}}} P_{j|\kappa_2} P_{alloc}^{sp} \pi(\theta_{j1}^{sp}|D_{j_1}, \eta_{j_1}) \pi(\theta_{j2}^{sp}|D_{j_2}, \eta_{j_2}), \tag{15}$$

where $\pi(\theta_m|D_m, \eta_m)$ is the posterior density for θ_m, for $m \in \{j_1, j_2\}$.

3.2. Merge movement

Consider now the reverse move of the split movement, the merge movement. That is, let Φ^{sp} be the current state and we want to return to the initial state Φ. For this, we need to merge the adjacent clusters D_{j_1} and D_{j_2} in a single cluster D_j.

Firstly, a merge movement is chosen with probability $P_{me|k_{\mathbf{c}}^{sp}}$ given as in (14). The probability of selecting the adjacent clusters D_{j_1} and D_{j_2} for a merge is

$$P_{j_1,j_2|k_{\mathbf{c}}^{sp}} = P_{j_1}P_{j_2|j_1} + P_{j_2}P_{j_1|j_2} = \begin{cases} 1, & \text{if } k_{\mathbf{c}}^{sp} = 2; \\ \dfrac{3}{2k_{\mathbf{c}}^{sp}}, & \text{if } k_{\mathbf{c}}^{sp} > 2 \text{ and } j_1 = 1 \text{ or } j_2 = k_{\mathbf{c}}^{sp}; \\ \dfrac{1}{k_{\mathbf{c}}^{sp}}, & \text{if } k_{\mathbf{c}}^{sp} > 2 \text{ and } j_1 \neq 1 \text{ or } j_2 \neq k_{\mathbf{c}}^{sp}; \end{cases}$$

where P_{b_1} is the probability of choosing cluster b_1 and $P_{b_2|b_1}$ is the conditional probability of choosing cluster b_2 given the previous choice of b_1.

Given the choice if the clusters D_{j_1} and D_{j_2}, we join them in a single cluster D_j, i.e. we do $D_j = \{D_{j_1}\} \cup \{D_{j_2}\}$. The configuration $\mathbf{c} = (c_1, \dots, c_n)$ is obtained doing:

(a) $c_i = c_i^{sp}$ for all $c_i^{sp} = j'$ and $j' \leq j_1$;
(b) $c_i = c_i^{sp} - 1$ for all $c_i^{sp} = j'$ and $j' \geq j_2$;

for $i = 1, \dots, n, j' = 1, \dots, n$ and $j_1, j_2 \in \{1, \dots, k_{\mathbf{c}}\}$.

Conditional on configuration $\mathbf{c}$ we have $k_{\mathbf{c}} = k_{\mathbf{c}}^{sp} - 1$ clusters. The vector of parameters $\boldsymbol{\theta}_n = (\boldsymbol{\theta}_{k_{\mathbf{c}}}, \boldsymbol{\theta}_0)$ is obtained doing $\theta_{j'} = \theta_{j'}^{sp}$ $\forall j' < j_1$, $\theta_{j'} = \theta_{j'+1}^{sp}$ $\forall j' \geq j_2$ and generating θ_j from its posterior distribution $\pi(\theta_j|D_j)$. Besides, in order to complete $\boldsymbol{\theta}_n$, generate θ_n from its prior distribution, $\theta_n \sim \pi_G(\theta_n)$.

Here, we also must check if the adjacency condition for parameters of the clusters is met, i.e. $\mu_{j-1} < \mu_j < \mu_{j+1}$. If the adjacency condition is met, the transition probability for the merge proposal is given by

$$q[\boldsymbol{\Phi}|\boldsymbol{\Phi}^{sp}] = P_{me|k_{\mathbf{c}}^{sp}} P_{j_1,j_2|k_{\mathbf{c}}^{sp}} \pi(\theta_j^{me}|D_j, \eta_j)\pi_G(\theta_n|\eta_n). \tag{16}$$

From (15) and (16), the transition probability ratio for a split proposal is given by

$$\frac{q[\boldsymbol{\Phi}|\boldsymbol{\Phi}^{sp}]}{q[\boldsymbol{\Phi}^{sp}|\boldsymbol{\Phi}]} = \frac{P_{me|k_{\mathbf{c}}^{sp}}}{P_{sp|k_{\mathbf{c}}}} \frac{P_{j_1,j_2|k_{\mathbf{c}}^{sp}}}{P_{j|\kappa_2}} \frac{1}{P_{alloc}^{sp}} \frac{\pi(\theta_j|D_j, \eta_j)\pi(\theta_n|\eta_n)}{\pi(\theta_{j_1}^{sp}|D_{j_1}, \eta_{j_1})\pi(\theta_{j_2}^{sp}|D_{j_2}, \eta_{j_2})}. \tag{17}$$

From equation (13), the acceptance probability for a split movement is $\Psi[\boldsymbol{\Phi}^{sp}|\boldsymbol{\Phi}] = \min(1, A^{sp})$, where (see Appendix 1 of the SM)

$$A^{sp} = \frac{\mathbf{I}(D_{j_1})\mathbf{I}(D_{j_2})}{\mathbf{I}(D_j)} \frac{\Gamma(n_{j_1})\Gamma(n_{j_2})}{\Gamma(n_j)} \frac{Q^{sp}}{P_{alloc}},$$

$$Q^{sp} = \frac{P_{me|k_{\mathbf{c}}^{sp}}}{P_{sp|k_{\mathbf{c}}}} \frac{P_{j_1,j_2}}{P_{j|\mathbb{C}_2}} = \begin{cases} \dfrac{1}{2}, & \text{if } k_{\mathbf{c}} = 1; \\ \left(\dfrac{1}{2}\right)^{1-\mathbb{I}_{k_{\mathbf{c}}^{sp}}(n)} \dfrac{3\kappa_2}{k_{\mathbf{c}+1}}, & \text{if } k_{\mathbf{c}} \in \mathbb{K}_1; \\ 2^{\mathbb{I}_{k_{\mathbf{c}}^{sp}}(n)} \dfrac{\kappa_2}{k_{\mathbf{c}} + 1}, & \text{if } k_{\mathbf{c}} \in \mathbb{K}_2; \end{cases}$$

with $\mathbb{I}_{k_{\mathbf{c}}^{sp}}(n)$ being an indicator function and the sets $\mathbb{K}_1 = \{2 \leq k_{\mathbf{c}} \leq k - 1;$ and $j_1 = 1$ or $j_2 = k_{\mathbf{c}}\}$, $\mathbb{K}_2 = \{2 \leq k_{\mathbf{c}} \leq k - 1$ and $j_1 \neq 1$ or $j_2 \neq k_{\mathbf{c}}\}$. Similarly, the acceptance probability for a merge is $\Psi[\mathbf{\Phi}^{me}|\mathbf{\Phi}] = \min(1, A^{me}) = 1/A^{sp}$, but with some obvious differences in the substitutions.

3.3. Sequential data-driven allocation sampler

Now the split-merge procedure is inserted within the Algorithm 1 and described as an algorithm denominated by Sequential Data-driven allocation sampler (SDAS).

SDAS Algorithm: Initialize with a configuration $(\mathbf{c}^{(0)}, \boldsymbol{\theta}^{(0)}_{k_\mathbf{c}})$. For l-th iteration of the algorithm do:

(i) Update the indicator variables $\mathbf{c}$ using items (1)–(6) of the MCMC Algorithm 1;
(ii) Choose between split or merge with probabilities $P_{sp|k_\mathbf{c}}$ and $P_{me|k_\mathbf{c}}$;
(iii) Accept the proposal with probability $\Psi[\mathbf{\Phi}^*|\mathbf{\Phi}]$, where '$*$' is either a *sp* or a *me*;
  (a) If a split proposal is accepted, do $k^{(l)}_\mathbf{c} = k^{(l-1)}_\mathbf{c} + 1$;
  (b) If a merge proposal is accepted, do $k^{(l)}_\mathbf{c} = k^{(l-1)}_\mathbf{c} - 1$;
  (c) Otherwise, maintain $k^{(l)}_\mathbf{c} = k^{(l-1)}_\mathbf{c}$;
(i) Update the component parameters $\boldsymbol{\theta}_{k_\mathbf{c}}$ using item (7) of the Algorithm 1;

Run the SDAS algorithm for L iterations and discard the first B iterations as a burn-in. Consider $N_{k_\mathbf{c}}(j)$ be the number of times that $k_\mathbf{c} = j$ in the $L{-}B$ iterations, for $j \in \{1, \ldots, n\}$. Thus, $\tilde{P}(k_\mathbf{c} = j) = N_{k_\mathbf{c}}(j)/(L - B)$ is the estimated posterior probability for $k_\mathbf{c} = j$ and $\tilde{k}_\mathbf{c} = \underset{1\le j\le k}{\operatorname{argmax}}(\tilde{P}(k_\mathbf{c} = j))$ is the $k_\mathbf{c}$ value with highest estimated posterior probability.

Conditional on $\tilde{k}_\mathbf{c}$, we define a configuration for the latent allocation variables $\mathbf{c}$ and obtain the estimates for parameters of the clusters according to the procedure described in Appendix 2 of the SM.

4. Data analysis

In this section, we present a discussion on the performance of the proposed method by using simulated data sets and three real datasets. We model these datasets considering an univariate normal mixture model, i.e. in the model (1), $f(y_i|\theta_j)$ is the density of a normal distribution with mean μ_j and variance σ^2_j and $\theta_j = (\mu_j, \sigma^2_j)$, for $j = 1, \ldots, k$.

In order to explore the fully conjugation, we consider the following prior distributions for component parameters $\theta_j = (\mu_j, \sigma^2_j)$,

$$\mu_j|\sigma^2_j, \mu_0, \lambda \sim \mathcal{N}\left(\mu_0, \frac{\sigma^2_j}{\lambda}\right) \quad \text{and} \quad \sigma_j^{-2}|\alpha, \beta \sim \Gamma(\alpha, \beta),$$

where μ_0, λ, α and β are hyperparameters, $\mathcal{N}(\mu_0, \sigma^2_j/\lambda)$ represents the normal distribution with mean μ_0 and variance σ^2_j/λ and $\Gamma(\alpha, \beta)$ represents the Gamma distribution with location parameter α and scale parameter β. The parametrization of the Gamma distribution is so that the mean is α/β and the variance is α/β^2. These prior distributions are also used by Casella et al. [32], Nobile and Fearnside [33] and Saraiva et al. [34].

In order to obtain weakly prior distributions for component parameters $\theta_j = (\mu_j, \sigma_j^2)$, we specify the hyperparameters values in a way that $\mu_0 = \varepsilon$ and $E(\sigma_j^{-2}) = R^{-2}$, where ε is the midpoint of the observed variation interval of the data and R is the length of this interval. Thus, we obtain $\beta = \alpha R^2$ and we fix $\alpha = 1$. We also fix the hyperparameter $\lambda = 10^{-2}$ in order to get a prior distribution for component means with large variance and we fix $\gamma = 0.1$.

The conditional posterior distributions for parameters of the clusters are

$$\mu_j|\sigma_j^2, \mathbf{y}, \mathbf{c}, k, \mu_0, \lambda \sim \mathcal{N}\left(\mu_j^{post}, \frac{\sigma_j^2}{n_j + \lambda}\right) \quad \text{and} \quad \sigma_j^{-2}|\mathbf{y}, \mathbf{c}, k, \tau, \beta \sim \Gamma\left(\alpha_j^{post}, \beta_j^{post}\right),$$

where

$$\mu^{post} = \frac{\sum_{D_j} y_i + \lambda\mu_0}{n_j + \lambda}, \alpha^{post} = \alpha + \frac{n_j + 1}{2}, \beta^{post} = \beta + \frac{A}{2} - \frac{B^2}{2(n_j + \lambda)}, \tag{18}$$

for $A = \sum_{D_j} y_i^2 + \lambda\mu_0^2$, $B = \sum_{D_j} y_i + \lambda\mu_0$ and $j = 1, \ldots, k_{\mathbf{c}}$.

The normalizing constant $\mathbf{I}(D_m)$ present in the acceptance probability for the split-merge movements is given by

$$\mathbf{I}(D_m) = \left[\frac{1}{2\beta\pi}\right]^{n_m/2} \left[\frac{\lambda}{n_m + \lambda}\right]^{1/2} \frac{\Gamma\left(\alpha + \dfrac{n_m + 1}{2}\right)}{\Gamma(\alpha)} \times \left[1 + \frac{A}{2\beta} - \frac{B^2}{2\beta(n_m + \lambda)}\right]^{-(\alpha + (n_m/2))},$$

for $m \in \{j, j_1, j_2\}$.

The KL divergence between $\pi(\theta|D_m^*)$ and $\pi(\theta|D_m)$, used in the split movement, is

$$KL(D_m^*; D_m) = \sum_{r=1}^{6} K_r \tag{19}$$

where

$$K_1 = \frac{1}{2}log\left(\frac{n_m + 1 + \lambda}{n_m + \lambda}\right) - \frac{1}{2} + \frac{n_m + \lambda}{2(n_m + 1 + \lambda)}, \quad K_4 = log\left(\frac{\Gamma(\alpha_m^{post*})}{\Gamma(\alpha_m^{post})}\right),$$

$$K_2 = (n_m + \lambda)\frac{(\mu_m^{post*} - \mu_m^{post})^2}{2}\frac{\alpha_m^{post*}}{\beta_m^{post*}}, \quad K_5 = \left(\alpha_m^{post*} - \alpha_m^{post}\right)\psi(\alpha_m^{post*}),$$

$$K_3 = \alpha_m^{post} log\left(\frac{\beta_m^{post*}}{\beta_m^{post}}\right), \quad K_6 = \left(\beta_m^{post*} - \beta_m^{post}\right)\frac{\alpha_m^{post*}}{\beta_m^{post*}},$$

$D_m^* = D_m \cup \{y_i\}$, for $m \in \{j_1, j_2\}$, and μ_m^{post*}, α_m^{post*} and β_m^{post*} are calculated using D_m^* in Equation (18) and $\psi(a)$ is the digamma function.

Table 1. Number of clusters and parameter values used for simulating the datasets.

Artificial data set	Number of clusters	Parameter values				
A_1	$k_c = 2$	$\mu_1 = 0,$	$\mu_2 = 3,$			
		$\sigma_1^2 = 1,$	$\sigma_2^2 = 1,$			
		$w_1 = 0.40,$	$w_2 = 0.60,$			
A_2	$k_c = 3$	$\mu_1 = -4,$	$\mu_2 = 0,$	$\mu_3 = 4$		
		$\sigma_1^2 = 0.5,$	$\sigma_2^2 = 1,$	$\sigma_3^2 = 2$		
		$w_1 = 0.20,$	$w_2 = 0.30,$	$w_3 = 0.50$		
A_3	$k_c = 4$	$\mu_1 = 0,$	$\mu_2 = 8,$	$\mu_3 = 15,$	$\mu_4 = 22$	
		$\sigma_1^2 = 4,$	$\sigma_2^2 = 3,$	$\sigma_3^2 = 2,$	$\sigma_4^2 = 1$	
		$w_1 = 0.40,$	$w_2 = 0.30,$	$w_3 = 0.20,$	$w_4 = 0.10$	
A_4	$k_c = 5$	$\mu_1 = -6,$	$\mu_2 = 0,$	$\mu_3 = 8,$	$\mu_4 = 15,$	$\mu_5 = 21$
		$\sigma_1 = 1,$	$\sigma_2 = 2,$	$\sigma_3 = 3,$	$\sigma_4 = 2,$	$\sigma_5 = 1$
		$w_1 = 0.15,$	$w_2 = 0.20,$	$w_3 = 0.30,$	$w_4 = 0.20,$	$w_5 = 0.15,$

4.1. *Artificial datasets*

In order to generate the artificial datasets, we set up the number of clusters k_c and the component parameters for the k_c clusters according to the specified in Table 1.

In the set up A_1, the two clusters have equal variances while A_2 has three clusters with different variances and weights. In A_3, we consider four clusters with decreasing variances and weights values; and in A_4, we consider five clusters with different variances and weights being the two last clusters away from the first three clusters.

The procedure for generating the data sets is given by the following two steps:

(i) For $i = 1, \ldots, n$, generate $U_i \sim \mathcal{U}(0, 1)$; if $\sum_{j'=1}^{j-1} w_j < u_i \leq \sum_{j'=1}^{j} w_j$, generate $Y_i \sim \mathcal{N}(\mu_j, \sigma_j^2)$, with fixed parameter values according to Table 1, for $w_0 = 0$ and $j = 1, \ldots, k_c$.

(ii) In order to record from which component each observation is generated from we define $G = (G_1, \ldots, G_n)$ such that $G_i = j$ if $Y_i \sim \mathcal{N}(\mu_j, \sigma_j^2)$, for $i = 1, \ldots, n$ and $j = 1, \ldots, k_c$.

Figure 8 in Appendix 3 of the SM shows the values generated by cluster for datasets A_1 to A_4 for $n = 500$.

For each generated dataset, we apply the proposed SDAS algorithm fixing $L = 55{,}000$ iterations and a burn-in of $B = 5000$. We also consider a sample of one draw for every 20 in order to obtain a sequence of 5000 cases to make inferences. The algorithm was initialized with just one cluster, $k_c = 1$, and component parameters $\mu_1 = \bar{y}$ and $\sigma_1^2 = s^2$, where $\bar{y}$ and s^2 represent the sample mean and variance of the generated data.

The results for posterior probabilities of k_c are presented in Table 2. The estimated maximum posterior probability for each dataset is highlighted in bold. As we can note, the proposed SDAS algorithm attributes maximum posterior probability for the k_c true value.

Figure 1 shows the performance of the SDAS algorithm in relation to the sampled k_c values. Figure 1(a,d,g,j) show the plots of the $P(k_c|\cdot)$ estimates across the iterations for datasets A_1 to A_4, respectively. In order to maintain a good visualization, we display in each graphic only the two higher $P(k_c|\cdot)$ estimates. As we can note, the number

Table 2. Posterior probability for $k_{\mathbf{c}}$.

Data set	$k_{\mathbf{c}}^{true}$	$k_{\mathbf{c}}$	$P(k_{\mathbf{c}}\|\cdot)$	Data set	$k_{\mathbf{c}}^{true}$	$k_{\mathbf{c}}$	$P(k_{\mathbf{c}}\|\cdot)$
A_1	2	1	0.0000	A_2	3	1	0.0000
		2	**0.5782**			2	0.0000
		3	0.2900			3	**0.6068**
		4	0.0932			4	0.2968
		5	0.0318			5	0.0752
		≥ 6	0.0068			≥ 6	0.0212
A_3	4	≤ 3	0.0000	A_4	5	≤ 3	0.0000
		4	**0.7928**			4	0.0000
		5	0.1810			5	**0.6992**
		6	0.0234			6	0.2684
		7	0.0028			7	0.0312
		≥ 8	0.0000			≥ 8	0.0012

of iterations and burn-in seems to be adequate to achieve stability for the posterior probabilities of $k_{\mathbf{c}}$. Figure 1(b,e,h,k) shows the sampled $k_{\mathbf{c}}$ values in the course of iterations. Figure 1(c,f,i,l) shows the estimated autocorrelation functions (acf). These figures show us that the proposed algorithm mix well over $k_{\mathbf{c}}$ and does not present significant autocorrelation.

Figure 9 in Appendix 4 of the SM shows the generated values and identified clusters by the SDAS algorithm for datasets A_1 to A_4. The cluster were satisfactorily identified. This Appendix also present the parameter estimates for each cluster (Table 6 of the SM) and the histogram of the observed data and the estimated density function (Figure 10 of the SM). As one can note the results are satisfactory.

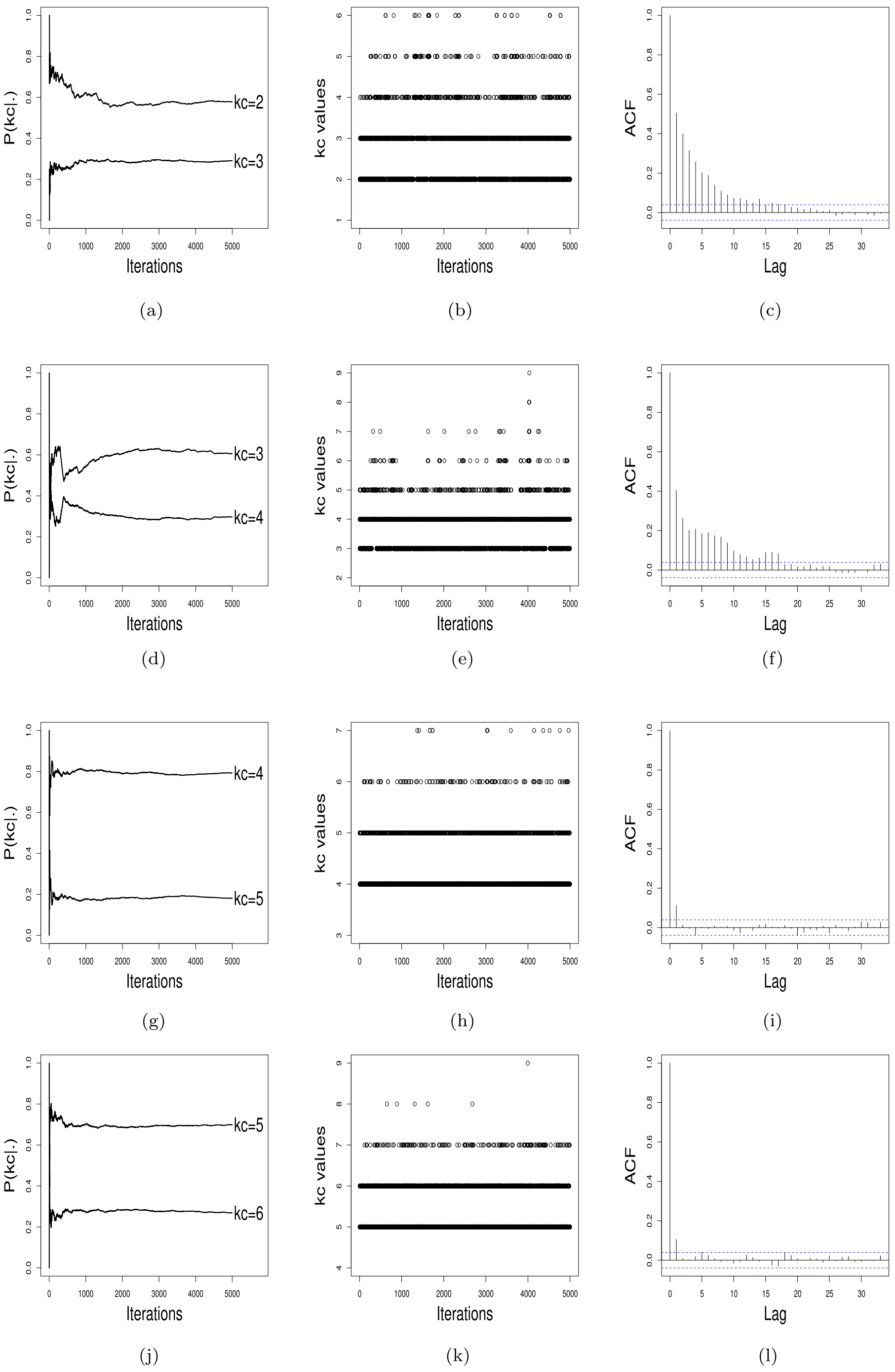
We also repeat the simulation procedure describe above for $M = 100$ different simulated datasets. For each one of the M simulated datasets the number of clusters $k_{\mathbf{c}}$ was estimated using the SDAS algorithm. The proportion of times that SDAS placed higher posterior probability on the $k_{\mathbf{c}}$ true value for datasets A_1, A_2, A_3 and A_4 were 0.92, 0.98, 0.98 and 0.98, respectively. This results show a satisfactory performance of the SDAS algorithm in estimation of the number of clusters $k_{\mathbf{c}}$.

Table 3 shows the average of the posterior probability for $k_{\mathbf{c}}$ obtained from the M simulated datasets. As we can note, the average of the posterior probability for the $k_{\mathbf{c}}$ true value is greater than others values of $k_{\mathbf{c}}$, for the four simulated datasets.

4.2. Real data sets

We now apply the proposed method to three benchmark datasets. The first one is the Galaxy dataset previously analysed by Richardson and Green [19], Stephens [25], Roeder and Wasserman [35], Escobar and West [36], among others. This dataset refers to velocity in km/sec of $n = 82$ galaxies from 6 well-separated conic sections of an unfilled survey of the Corona Borealis region. This dataset is available in the R software.

The second real dataset refers to enzymatic activity in the blood, for an enzyme involved in the metabolism of carcinogenic substances, among a group of $n = 245$ unrelated individuals; and the third dataset refers to an acidity index measured in a sample of $n = 155$ lakes in north-central Wisconsin. The Enzyme and Acidity datasets were downloaded from the website https://people.maths.bris.ac.uk/mapjg/mixdata.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 1. Performance of the SDAS algorithm across iterations for datasets A_1 and A_2. (a) $P(k_\mathbf{c}|\cdot)$ for dataset A_1, (b) sampled $k_\mathbf{c}$ values, A_1, (c) Est. acf for dataset A_1. (d) $P(k_\mathbf{c}|\cdot)$ for dataset A_2, (e) sampled $k_\mathbf{c}$ values, A_2, (f) Est. acf for dataset A_2. (g) $P(k_\mathbf{c}|\cdot)$ for dataset A_3, (h) sampled $k_\mathbf{c}$ values, A_3, (i) Est. acf for dataset A_3. (j) $P(k_\mathbf{c}|\cdot)$ for dataset A_4, (k) sampled $k_\mathbf{c}$ values, A_4 and (l) Est. acf for dataset A_4.

Table 3. Average of the posterior probability for $k_\mathbf{c}$ for $M = 100$ simulated datasets.

Data set	$k_\mathbf{c}^{true}$	$k_\mathbf{c}$	$P(k_\mathbf{c}\vert\cdot)$	Data set	$k_\mathbf{c}^{true}$	$k_\mathbf{c}$	$P(k_\mathbf{c}\vert\cdot)$
A_1	2	1	0.0001	A_2	3	1	0.0000
		2	**0.5233**			2	0.0045
		3	0.3078			3	**0.5856**
		4	0.1176			4	0.3018
		≥ 5	0.0511			≥ 5	0.1081
A_3	4	≤ 3	0.0085	A_4	5	≤ 3	0.0000
		4	**0.7369**			4	0.0000
		5	0.2149			5	**0.7679**
		6	0.0353			6	0.1939
		≥ 7	0.0044			≥ 7	0.0382

Note: The highlighted values in bold are the average of the estimated posterior probability for the $k_\mathbf{c}$ true value.

Table 4. Estimated probabilities for $k_\mathbf{c}$, real datasets.

	$k_\mathbf{c}$ values					
Data set	1	2	3	4	5	≥ 6
Galaxy	0.0000	0.0050	**0.8694**	0.1154	0.0096	0.0006
Enzyme	0.0000	0.0802	**0.6412**	0.2436	0.0314	0.0036
Acidity	0.0000	0.1896	0.2936	**0.2994**	0.1624	0.0550

Note: The estimated maximum posterior probability for each dataset is highlighted in bold.

These three real datasets have been analyzed with SDAS using the same hyperparameters specification, the number of iterations, burn in size and thin value used for simulated datasets.

Table 4 shows the estimates for posterior probability of $k_\mathbf{c}$ for each dataset. For Galaxy dataset, the maximum posterior is at $k_\mathbf{c} = 3$ with $P(k_\mathbf{c} = 3|\cdot) = 0.8694$. For Enzyme dataset the maximum posterior is also at $k_\mathbf{c} = 3$ with $P(k_\mathbf{c} = 3|\cdot) = 0.6412$; while for Acidity dataset the maximum posterior is at $k_\mathbf{c} = 4$ with $P(k_\mathbf{c} = 4|\cdot) = 0.2994$. But, for this dataset, the posterior probability for $k_\mathbf{c} = 3$ and $k_\mathbf{c} = 4$ are very near.

Figure 2 shows the performance of the SDAS for each dataset. As one can note, the SDAS sampler mixes well over $k_\mathbf{c}$ and shows a satisfactorily stability for probabilities of $k_\mathbf{c}$. Besides, the sampled $k_\mathbf{c}$ values do not present significant autocorrelation.

Figure 3 shows the identified clusters for each dataset, conditional on the estimate $\tilde{k}_\mathbf{c}$. At this point, we present the clusters identified with $\tilde{k}_\mathbf{c} = 4$ for acidity dataset. Below we make a discussion on the number of clusters $\tilde{k}_\mathbf{c} = 3$ and $\tilde{k}_\mathbf{c} = 4$ for this dataset. Table 5 shows the estimates for parameters of the identified clusters and the credibility intervals (95%). Figure 4 shows the histogram of the observed data and the estimated density function.

Consider now the estimated number of cluster for Acidity dataset. Note in Figure 3(c) that this dataset has a smaller observed value away from the others observations. Due this, we opt to verify whether this observation is an influential observation for the cluster 1.

In order to do it we consider the following procedure. Let $\mathbf{y}_1$ be the observations of the Acidity dataset allocated in cluster 1 with $\tilde{k}_\mathbf{c} = 4$ by the SDAS algorithm and $\mathbf{y}_1^* = \{\mathbf{y}_1\} \setminus \{\min(\mathbf{y}_1)\}$ be the observations of the cluster 1 excluding the smallest observation. Consider $\pi_1(\theta_1) = \pi(\mu_1, \sigma_1^2|\mathbf{y}_1)$ and $\pi_2(\theta_2) = \pi(\mu_1, \sigma_1^2|\mathbf{y}_1^*)$ be the posterior distributions for $\theta_1 = (\mu_1, \sigma_1^2)$ given the observed data $\mathbf{y}_1$ and $\mathbf{y}_1^*$, respectively. The KLD measure between $\pi_1(\theta_1)$ and $\pi_2(\theta_1)$ is $KL(\mathbf{y}_1, \mathbf{y}_1^*)$ that is calculated according to Equation (19).

However, the influence measure $KL(\mathbf{y}_1, \mathbf{y}_1^*)$ do not determine when an observation is influential. For this, we need to define a cutoff point in order to determine whether $\min(\mathbf{y}_1)$

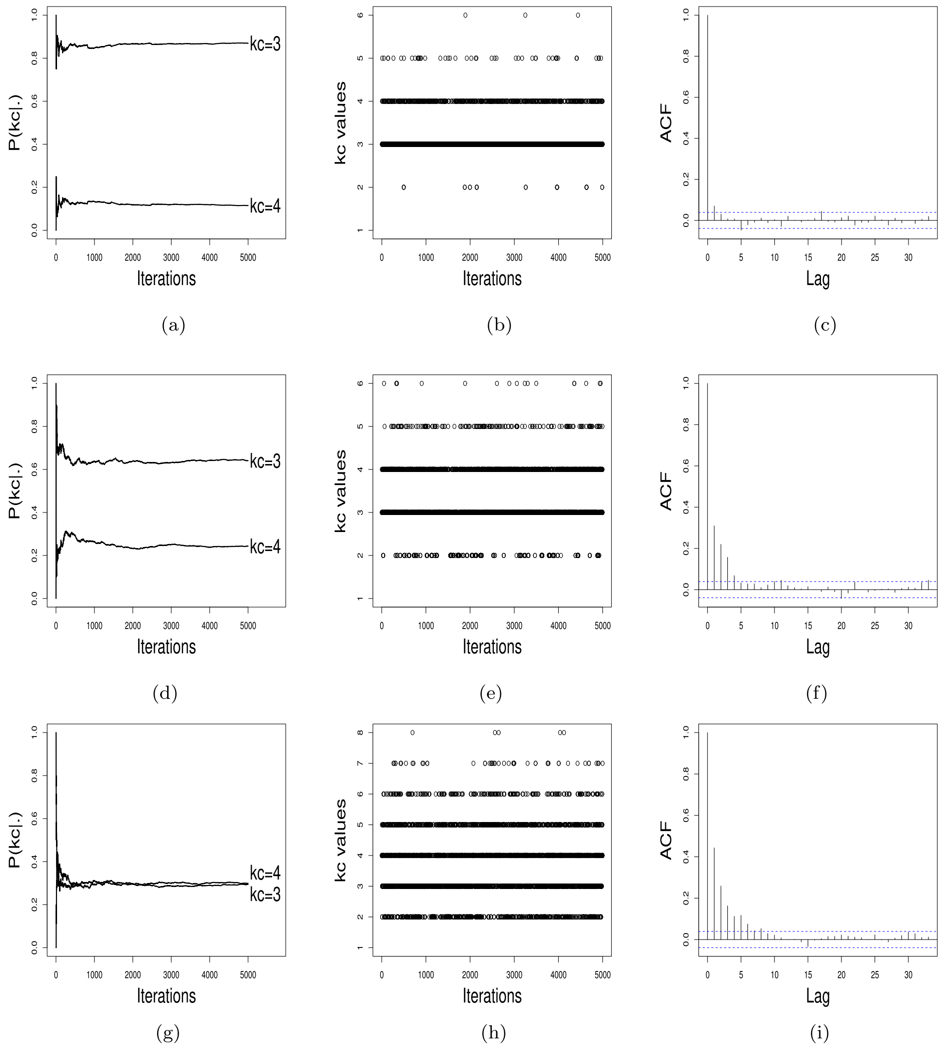


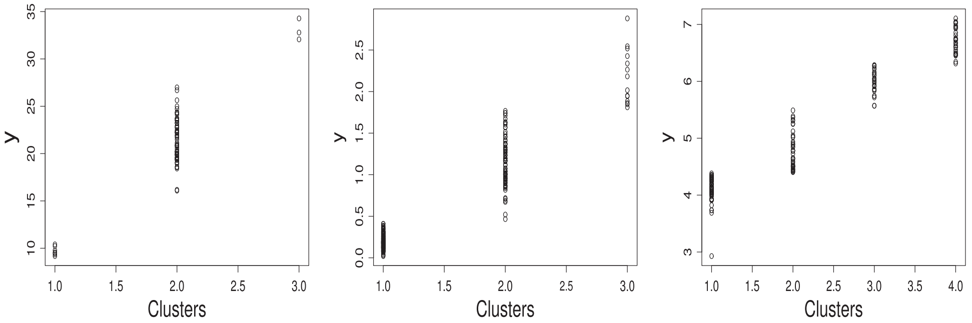
Figure 2. Performance of the SDAS algorithm across iterations for real datasets. (a) Galaxy data, (b) galaxy data, (c) galaxy data, (d) enzyme data, (e) enzyme data, (f) enzyme data, (g) acidity data, (h) acidity data and (i) acidity data.

is influential or not. In order to define the cutoff point we consider the proposal given by Peng and Dey [37]. The proposal is based on the divergence between distributions of a biased and unbiased coin, explained next.

The probability function of a biased coin is $\pi_1(x|p) = p^x(1-p)^{1-x}$ while for an unbiased coin is $\pi_2(x|p=0.5) = 0.5$, for $x = 0,1$ and $p \in (0,1)$. The KLD between a biased and an unbiased coin is given by

$$KLD(p) = \frac{-log(2p) - log(2(1-p))}{2}.$$

Figure 5 shows the graphic of $KLD(p)$. As one can note, $KLD(p)$ increases as p moves away from 0.5, is symmetric around $p = 0.5$ and achieves its minimum at $p = 0.5$. For

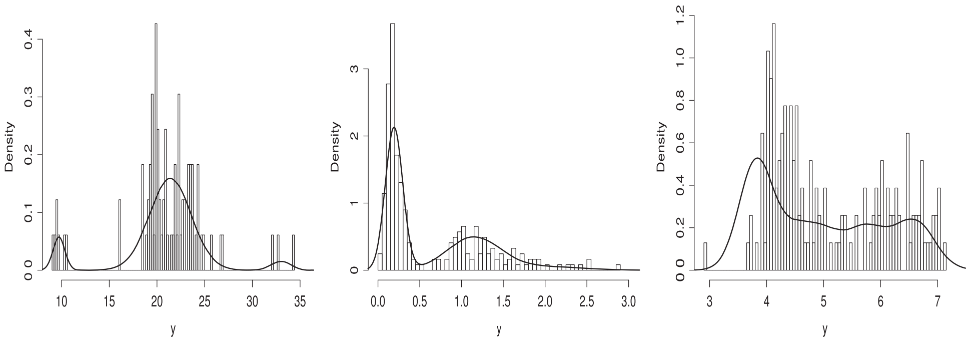


(a) (b) (c)

Figure 3. Identified clusters for real datasets. (a) Galaxy dataset, (b) enzyme dataset and (c) acidity dataset.

Table 5. Estimates for component parameters of the $\tilde{k}_c$ clusters.

	Data set		
Parameter	Galaxy	Enzyme	Acidity
μ_1	9.7286 (9.2905, 10.1615)	0.1921 (0.1771, 0.2075)	3.8025 (2.6188, 4.6391)
μ_2	21.4021 (20.8929, 21.0968)	1.1480 (1.0024, 1.2823)	4.6999 (4.1361, 5.9405)
μ_3	32.9864 (31.8243, 34.1145)	2.0202 (1.3610, 2.6398)	5.7798 (4.5106, 6.4661)
μ_4	–	–	6.6119) (6.1838, 6.9595)
σ_1^2	0.3383 (0.1308, 0.8443)	0.084 (0.0065, 0.0106)	0.774 (0.0158, 0.2399)
σ_2^2	4.7508 (3.4154, 6.6001)	0.0997 (0.0322, 0.2033)	0.2382 (0.0224, 0.7959)
σ_3^2	1.0094 (0.2630, 3.3673)	0.1803 (0.0377, 0.4264)	0.1303 (0.0170, 0.4903)
σ_4^2	–	–	0.1082 (0.0191, 0.3319)



(a) (b) (c)

Figure 4. Histogram of observed data and estimated density function. (a) Galaxy data, (b) enzyme data and (c) acidity data.

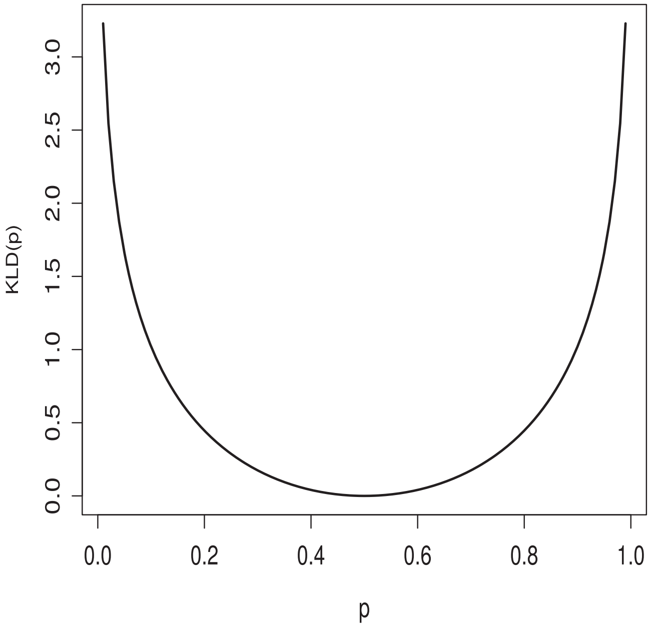


Figure 5. Kullback–Leibler divergence for *p*.

$p = 0.5$, $KLD(0.5) = 0$ and $\pi_1(x|p) = \pi_2(x|p)$. In this way, if we consider $p \geq 0.80$ (or $p \leq 0.20$) as a strong bias in a coin, then, since $KLD(0.80) = 0.2231$, we can indicate an influential observation when $DKL(p) > 0.2231$. Thus, we consider that $\min(\mathbf{y}_1)$ is an influential observation if $KL(\mathbf{y}, \mathbf{y}^*) > 0.2231$.

Applying Equation (19), we obtained $KL(\mathbf{y}_1, \mathbf{y}_1^*) = 8.3511$. This results indicates that $\min(\mathbf{y})$ is an influential observation. We then apply the SDAS algorithm for the acidity dataset excluding this influential observation. We call this dataset by Acidity* dataset. For this 'new' dataset the estimated posterior probability for $k_\mathbf{c} = 1, 2, 3, 4$ and ≥ 5 are 0, 0.2590, 0.3830, 0.2638 and 0.0942, respectively. The maximum posterior probability is at $k_\mathbf{c} = 3$, $P(k_\mathbf{c} = 3|\cdot) = 0.3830$. That is, the excluded observation is also influential for the estimation of the number of clusters.

Figure 6 shows the performance of the SDAS for this dataset. As one can note, the method again presented a satisfactory performance.

Figure 7 shows the histogram of the observed data and the estimated density function. Estimates for parameters of the three estimated clusters are $\mu_1 = 4.2771$, $\mu_2 = 5.3795$, $\mu_3 = 6.4675$, $\sigma_1^2 = 0.0859$, $\sigma_2^2 = 0.1867$ and $\sigma_3^2 = 0.1619$.

5. Final remarks

In this paper, we consider a Bayesian mixture model to estimate the number of cluster and the other parameters of interest in a joint way. In this approach, a Dirichlet prior distribution for the weights of the mixture with a convenient parametrization was assumed in order to allow us to integrate out the weights and to consider the number of components $k \to \infty$ to obtain a procedure to update the latent allocation variables. In this procedure, there is a non-null probability of a single observation to define a new cluster.

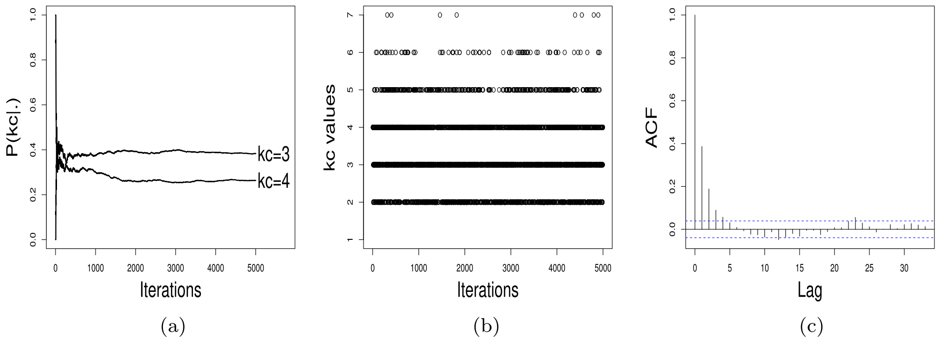


Figure 6. Performance of SDAS for acidity* dataset. (a) Galaxy data, (b) enzyme data and (c) acidity data.

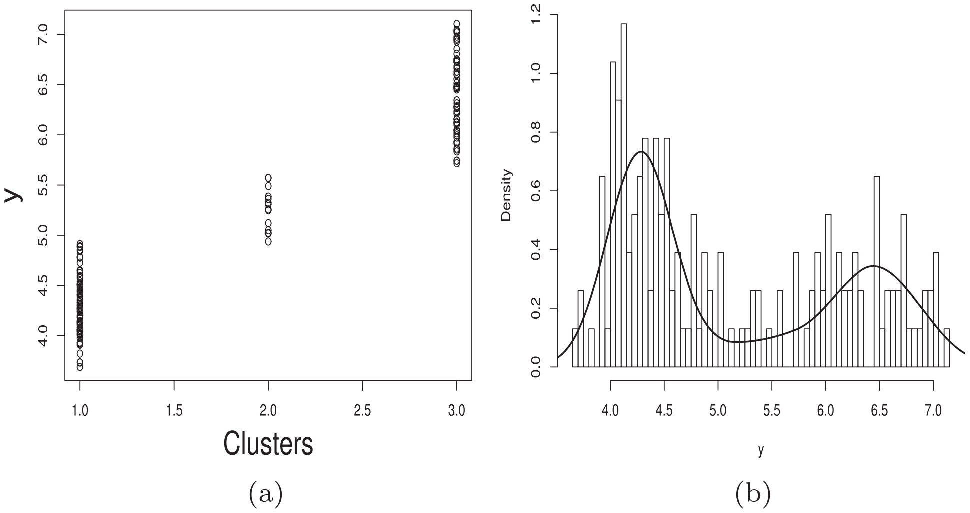


Figure 7. Histogram of observed data and estimated density function for acidity* dataset. (a) Galaxy data and (b) enzyme data.

In order to avoid local modes and increasing the mixing of the Markov chain in relation to the number of clusters we also consider a split-merge step to update the latent allocation variables. The split-merge step was constructed using a sequential allocation sampler based on allocations probabilities which are calculated according to the Kullback–Leibler divergence.

Defined the allocations probabilities we developed an MCMC algorithm called SDAS. In this algorithm, the procedure to update the latent allocation variables is given by a Metropolis–Hastings within a Gibbs sampling algorithm. The Metropolis–Hastings algorithm is used to update, in a joint way, a set of latent allocation variables according to a split-merge step; and the Gibbs sampling is used to update each indicator variable at time. Conditional on a configuration for the latent allocation variables, the parameters of the clusters are updated via Gibbs sampling algorithm generating values from their conditional posterior distribution.

Due to the way that we implement the split-merge strategy based on data, these proposals determine a new partition in the observed data set. This is one factor which improves the efficiency of the method in identifying clusters. Besides, these movements do not need of the specification of transition functions for being developed, simplifying its development and computational implementation.

In order to verify the performance of SDAS we developed a simulation study considering that clusters are generated from a mixture of univariate normal distributions. Results show a good performance of the SDAS. For all simulated cases, the SDAS placed higher posterior probability on the k_c true value and the identified clusters were a quite satisfactory.

We also apply the SDAS to three real datasets. For the Galaxy and Enzyme data sets, the SDAS placed higher posterior probability on $k_c = 3$, $P(k_c) = 0.8694$ and $P(k_c) = 0.6412$, respectively. For Acidity dataset, the SDAS placed higher posterior probability on $k_c = 3$ and $k_c = 4$ (please, see Table 4). This lack of definition of the method occurs due a influential observation present in this dataset. Due this, we apply the SDAS to this dataset excluding this influential value. For this case, the SDAS puts higher posterior probability on $k_c = 3$, $P(k_c = 3) = 0.3830$.

Results from simulated and real data sets show that SDAS may be an effective alternative for joint estimation of k_c, identification of clusters and estimation of parameters. A practical differential of the proposed algorithm is that it is essentially data-driven and it is simple to be implemented in softwares like *R* (the Comprehensive R Archive Network, http://cran.r-project.org). Besides, our approach does not need of the specification of transition functions to realize the split-merge movements and the candidate-values for parameters of the new cluster are generated from the posterior distribution. Due to the augmented parametrization considered the acceptance probability for split-merge movements are given by the Metropolis–Hastings acceptance probability. The source code used in data set analysis was developed in software *R* and is available upon request by emailing authors. In Appendix 4 of the SM, we provide the R codes used in the application of SDAS algorithm to the Galaxy dataset.

The SDAS algorithm was proposed here considering a Bayesian approach with conjugated prior distribution so that we could develop the split-merge movements using the Kullback–Leibler divergence and obtain the probability of a single observation define a new cluster in analytical way. Extending the SDAS for nonconjugated cases and the generalization for the multivariate case are possible future developments of the method.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

C. A. B. Pereira thanks the Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, for support [grant number 308776/2014-3].

References

[1] MacQueen J. Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability Vol. 1. Berkeley (CA): University of California Press; 1967. p. 281–297.

[2] Ward JH. Hierarchical groupings to optimize an objective function. J Am Stat Assoc. 1963;58:234–244.
[3] Sturn A, Quackenbush J, Trajanoski Z. Genesis: cluster analysis of microarray data. Bioinformatics. 2002;18:207–208.
[4] Wride M, Mansergh F, Adams S, et al. Expression profiling and gene discovery in the mouse lens. Mol Vis. 2003;9:360–396.
[5] Oyelade OJ, Oladipupo OO, Obagbuwa I. Application of k-means clustering algorithm for prediction of students' academic performance. Int J Comput Sci Inf Secur. 2010;7:292–295.
[6] Peterson AD, Ghosh AP, Maitra R. Merging k-means with hierarchical clustering for identifying general-shaped groups. Stats. 2018;7:e172.
[7] Oh MS, Raftery AE. Model-based clustering with dissimilarities: a Bayesian approach. J Comput Graph Stat. 2007;16:559–585.
[8] Bouveyron C, Brunet C. Model-based clustering of high-dimensional data: a review. Comput Stat Data Anal. 2014;71:52–78.
[9] McLachlan G, Basford KE. Mixture models: inference and applications to clustering. New York (NY): Marcel Dekker; 1988.
[10] Banfield JD, Raftery AE. Model-based gaussian and non-gaussian clustering. Biometrics. 1993;49:803–821.
[11] McLachlan G, Peel D. Finite mixture models. New York: Wiley Interscience; 2000.
[12] Fraley C, Raftery A. Model-based clustering, discriminant analysis, and density estimation. J Am Stat Assoc. 2002;97:611–631.
[13] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the em algorithm. J R Stat Soc, B. 1977;39(1):1–38.
[14] Akaike HA. New look at the statistical model identification. IEEE Trans Automat Contr. 1974;19:716–723.
[15] Bozdogan H. Model selection and Akaike's information criterion (AIC): the general theory and its analytical extensions. Psychometrica. 1987;52:345–370.
[16] Schwarz GE. Estimating the dimension of a model. Ann Stat. 1978;6:461–464.
[17] Spiegelhalter DJ, Best NG, Carlin BP, et al. Bayesian measures of model complexity and fit. J R Stat Soc Ser B. 2002;64:583–616.
[18] Celeux G, Hurn M, Robert CP. Computational and inferential difficulties with mixture posterior distributions. J Am Stat Assoc. 2000;95:957–970.
[19] Richardson S, Green PJ. On bayesian analysis of mixtures with an unknown number of components. J R Stat Soc Ser B. 1997;59:731–792.
[20] Kullback S, Leibler RA. On information and sufficiency. Ann Math Stat. 1951;22:79–86.
[21] Godsill SJ. On the relationship between markov chain monte carlo methods for model uncertainty. J Comput Graph Stat. 2001;10:230–248.
[22] Dellaportas P, Forster J, Ntzoufras I. On bayesian model and variable selection using MCMC. Stat Comput. 2002;12:27–36.
[23] Saraiva EF, Louzada F, Milan LA. Mixture models with an unknown number of components via a new posterior split-merge mcmc algorithm. Appl Math Comput. 2014;244:959–975.
[24] Fruhwirth-Schnatter S. Finite mixture and Markov switching models. New York (NY): Springer Science & Business Media; 2006.
[25] Stephens M. Dealing with label switching in mixture models. J R Stat Soc Ser B. 2000;62:795–809.
[26] Jasra A, Holmes CC, Stephens DA. Markov chain monte carlo methods and the label switching problem in bayesian mixture modeling. Stat Sci. 2005;20:50–67.
[27] Ferguson TS. A Bayesian analysis of some nonparametric problems. Ann Stat. 1973;1:209–230.
[28] Antoniak CE. Mixture of processes Dirichlet with applications to Bayesian nonparametric problems. Ann Stat. 1974;2:1152–1174.
[29] Jain S, Neal R. A split-merge markov chain monte carlo procedure for the dirichlet process mixture models. J Comput Graph Stat. 2004;13(1):158–182.
[30] Carlin BP, Chib S. Bayesian model choice via markov chain monte carlo methods. J R Stat Soc B. 1995;57:473–484.

[31] Chib S, Greenberg E. Understanding the Metropolis-Hastings algorithm. Am Stat. 1995;49: 327–335.

[32] Casella G, Robert C, Wells M. Mixture models, latent variables and partitioned importance sampling. Paris: CREST, INSEE; 2000. (Technical Report).

[33] Nobile A, Fearnside AT. Bayesian finite mixtures with an unknown number of components: the allocation sampler. Stat Comput. 2007;17:147–162.

[34] Saraiva EF, Suzuki AK, Louzada F, et al. Partitioning gene expression data by data-driven markov chain monte carlo. J Appl Stat. 2016;43:1155–1173.

[35] Roeder K, Wasserman L. Practical bayesian density estimation using mixture of normals. J Am Stat Assoc. 1997;92:894–902.

[36] Escobar MD, West M. Bayesian density estimation and inference using mixtures. J Am Stat Assoc. 1995;90:577–588.

[37] Peng F, Dey D. Bayesian analysis of outlier problems using divergence measures. Can J Stat. 1995;23:199–213.