entropy

MDPI

Article

An Integrated Approach for Making Inference on the Number of Clusters in a Mixture Model

Erlandson Ferreira Saraiva [1,*], Adriano Kamimura Suzuki [2], Luis Aparecido Milan [3] and Carlos Alberto de Bragança Pereira [1,4]

1 Instituto de Matemática, Universidade Federal de Mato Grosso do Sul, Campo Grande 79070-900, Brazil; cpereira@ime.usp.br
2 Departamento de Matemática Aplicada e Estatística, Universidade de São Paulo, São Carlos 13566-590, Brazil; suzuki@icmc.usp.br
3 Departamento de Estatística, Universidade Federal de São Carlos, São Carlos 13565-905, Brazil; dlam@ufscar.br
4 Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo 05508-090, Brazil
* Correspondence: erlandson.saraiva@ufms.br; Tel.: +55-67-3345-7511

Received: 23 September 2019; Accepted: 26 October 2019; Published: 30 October 2019

Abstract: This paper presents an integrated approach for the estimation of the parameters of a mixture model in the context of data clustering. The method is designed to estimate the unknown number of clusters from observed data. For this, we marginalize out the weights for getting allocation probabilities that depend on the number of clusters but not on the number of components of the mixture model. As an alternative to the stochastic expectation maximization (SEM) algorithm, we propose the integrated stochastic expectation maximization (ISEM) algorithm, which in contrast to SEM, does not need the specification, a priori, of the number of components of the mixture. Using this algorithm, one estimates the parameters associated with the clusters, with at least two observations, via local maximization of the likelihood function. In addition, at each iteration of the algorithm, there exists a positive probability of a new cluster being created by a single observation. Using simulated datasets, we compare the performance of the ISEM algorithm against both SEM and reversible jump (RJ) algorithms. The obtained results show that ISEM outperforms SEM and RJ algorithms. We also provide the performance of the three algorithms in two real datasets.

Keywords: model-based clustering; mixture model; EM algorithm; integrated approach

1. Introduction

Recently, there has been increasing interest in modeling using mixture models. This is mainly due to the flexibility for treating heterogeneous populations. Under a data-clustering framework, this model has the advantage of being probabilistic, and then the obtained clusters can have a better interpretation from a statistical point of view [1]. This contrasts with usual methods, such as k-means or hierarchical clustering, in which clusters are not statistically based, as discussed by [2].

From a frequentist viewpoint, the standard method to get the maximum likelihood estimates for the parameters of a mixture model is based on the use of the Expectation Maximization (EM) algorithm [3]. However, for the use of this algorithm, the number of components k of the mixture model needs to be known a priori. As the resulting model is highly dependent on the choice of this value, the main question is how to set the k value. Selecting an erroneous k value may cause the non-convergence of the algorithm and/or a low exploration of the clusterings. In addition, depending on the k value chosen we may have empty components, and therefore, there are no maximum likelihood estimates for these components.

An approach frequently used to determine the best k value among a fixed set of values is the use of the stochastic version of the EM algorithm (SEM) with some model selection criterion, such as the Akaike information criterion (AIC) [4,5] or the BIC [6]. In this approach, models are fitted for a set of predefined k values, and the best model is the one that has the smallest AIC or BIC value.

However, as discussed by [7], to adjust several models for a predefined set of values for the number of the cluster and compare them using some model selection criterion is not a practical and efficient procedure. Therefore, it is desirable to have an efficient algorithm to calculate the optimal number of clusters together with the estimation of the parameters of each mixture component. In this scenario, the Bayesian approach was successfully performed considering the Markov chain Monte Carlo (MCMC) algorithm with reversible jumps, described by [8] in the context of univariate normal mixture models. On the other hand, a difficulty often encountered in implementing a reversible jump algorithm (RJ) is the construction of efficient transition proposals that lead to a reasonable acceptance rate.

Following in the line of MCMC algorithms, [9] proposes a split–merge MCMC procedure for the conjugated Dirichlet process mixture model using a restricted Gibbs sampling scan to determine a split proposal, where the number of scans (tuning parameter) must be previously fixed by the user, and [10] extend their method to a nonconjugated Dirichlet process mixture model. [11] proposes a data-driven split-and-merge approach. In this proposal, the number of clusters is updated according to the creation of a new component based on a single observation and using a split–merge strategy, developed based on the use of the Kullback–Leibler divergence. A difficulty encountered for implementing this algorithm is the obtaining of the mathematical expression for the Kullback–Leibler divergence, which does not always have known analytical expression. In addition, the sequential allocation used in the split–merge strategy of these three works may make the algorithm slow when the sample size is great, and the computation implementation of these methods is not so simple.

The present work proposes an integrated approach that, in a joint way, selects the number of clusters and estimates the parameters of interest. With this approach, the mixture weights are integrated out to obtain allocation probabilities that depend on the number of clusters (nonempty components) but do not depend on the number of components k. In addition, considering k tending to infinity, this procedure introduces a positive probability of a new cluster being created by a single observation. When a new cluster is created, the parameters associated with it are generated from its posterior distribution. We then developed the ISEM (integrated stochastic expectation maximization) algorithm to estimate the parameters of interest. This algorithm configures a setting for latent allocation variables $\mathbf{c}$ according to allocation probabilities, and then the cluster parameters are updated conditionally on $\mathbf{c}$ as follows: for clusters with at least two observations, the parameter values are the maximum likelihood estimates; for the clusters with only one observation, the parameter values are generated from their posterior distribution.

In order to illustrate the computation implementation of the method and verify its performance, we have considered a specific model in which data are generated from mixtures of univariate normal distributions. This model allows us to avoid the label switching problem by considering the labeling of the components according to the increasing order of the component averages, as done by [8,11–13], among others. But we emphasize that our algorithm is not restricted to this particular model. For instance, for the multivariate case, we may consider the labeling of the components according to the eigenvalues of the current covariance matrix, as done by [14]. However, a detailed discussion of the multivariate case will be done in a future paper.

We also compare the performance of the ISEM with both SEM and RJ algorithms. The criteria used to compare the methods are the estimated probability of the number of clusters, convergence of the sampled values, mixing, autocorrelation, and computation time. We also applied the three algorithms to two real datasets. The first is the well-known Galaxy data, and the second is a dataset on Acidity.

The remainder of the paper is as follows. Section 2 describes the mixture model and the estimation process based on the SEM algorithm. Section 3 develops the integrated approach and describes the ISEM algorithm. Section 4 shows how we applied the algorithm to simulated datasets in order to assess its performance. Section 5 describes the application of the three algorithms to two real datasets. Section 6 is about our final remarks. Additional details are in the Supplementary Material, which is referred to as "SM" in this paper. Table 1 presents the main notations used throughout the article.

Table 1. Main mathematical notation used throughout the paper.

Notation	Description
k	Number of components
$k_{\mathbf{c}}$	Number of clusters
θ_j	Parameter of the j-th component, for $j = 1, \ldots, k$
$\boldsymbol{\theta}_k = (\theta_1, \ldots, \theta_k)$	The whole vector of parameters
w_j	Weight of the j-th component, for $j = 1, \ldots, k$
Y_i	The i-th sampled value, for $i = 1, \ldots, n$
c_i	The i-th indicator variable, for $i = 1, \ldots, n$
$\mathbf{y} = (y_1, \ldots, y_n)$	The vector of independent observations
$\mathbf{c} = (c_1, \ldots, c_n)$	The vector of latent indicator variables
$k_{\mathbf{c}_{-i}}$	Number of clusters excluding the i-th observation
$n_{j,-i}$	Number of observations assigned to the j-th component, excluding the i-th observation

2. Mixture Model and SEM Algorithm

Let $\mathbf{y} = (y_1, \ldots, y_n)$ be a vector of independent observations from a mixture model with k components, i.e.,

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

where $f(y_i|\theta_j)$ is the density of a family of parametric distributions with parameters θ_j (scalar or vector), $\boldsymbol{\theta}_k = (\theta_1, \ldots, \theta_k)$ are the parameters of the components, and $\mathbf{w} = (w_1, \ldots, w_k)$, $w_j > 0$ and $\sum_{j=1}^{k} w_j = 1$ are component weights.

The log-likelihood function for $(\boldsymbol{\theta}_k, \mathbf{w})$ is given by

$$l(\boldsymbol{\theta}_k, \mathbf{w}|\mathbf{y}, k) = \log\left\{\prod_{i=1}^{n}\left[\sum_{j=1}^{k} w_j f(y_i|\theta_j)\right]\right\} = \sum_{i=1}^{n} \log\left\{\left[\sum_{j=1}^{k} w_j f(y_i|\theta_j)\right]\right\}.$$

The mathematical notation $l(\boldsymbol{\theta}_k, \mathbf{w}|\mathbf{y}, k)$ is given as in the book of Casella and Berger (2002).

The usual procedure to obtain the maximum likelihood estimators consists of getting partial derivatives of $l(\boldsymbol{\theta}_k, \mathbf{w}|\mathbf{y})$ in relation to θ_j and then equalizing the result to zero, i.e.,

$$\frac{dl(\boldsymbol{\theta}_k, \mathbf{w}|\mathbf{y})}{d\theta_j} = \sum_{i=1}^{n} \frac{w_j f(y_i|\theta_j)}{\sum_{j=1}^{k} w_j f(y_i|\theta_j)} \frac{d\log\left[f(y_i|\theta_j)\right]}{d\theta_j} = 0, \tag{2}$$

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

But, note that in (2), the maximization procedure consists of a weighted maximization process of the log-likelihood function with each observation y_i having a weight associated to component j given by

$$w_{ij}^{*} = \frac{w_j f(y_i|\theta_j)}{\sum_{j=1}^{k} w_j f(y_i|\theta_j)}, \tag{3}$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, k$. However, these weights depend on the parameters that we are trying to estimate. In this way, we cannot obtain a "closed" mathematical expression that allows the direct maximization of the log-likelihood function. Due to this, the mixture problem is reformulated as a complete-data problem [12,15].

Complete-Data Formulation

Consider associated to each observation y_i a latent indicator variable c_i not known, so that if $c_i = j$, then y_i is from component j, for $i = 1, \ldots, n$ and $j = 1, \ldots, k$. The probability of $c_i = j$ is w_j, $P(c_i = j|\mathbf{w}, k) = w_j$, for $i = 1, \ldots, n$ and $j = 1, \ldots, k$. Letting n_j be the number of observations from component j (i.e., the number of c_is equals to j), the joint probability for $\mathbf{c} = (c_1, \ldots, c_n)$ given $\mathbf{w}$ and k is

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

The distribution of the number of observations assigned to each component, $n_1, \ldots, n_k$, called the occupation number, is multinomial, $(n_1, \ldots, n_k|n, \mathbf{w}) \sim Multinomial(n, \mathbf{w})$, where $n = n_1 + \ldots + n_k$.

Thus, under this augmented framework, we have that

(1) the probability of $c_i = j$, conditional on observation y_i and on component parameters $\boldsymbol{\theta}_k$, is w_{ij}^{*}, i.e., $P(c_i = j|y_i, \boldsymbol{\theta}_k, k) = w_{ij}^{*}$, for w_{ij}^{*} given in Equation (3), for $i = 1, \ldots, n$ and $j = 1, \ldots, k$. That is, although the indicator variables are nonobservable, they are implicitly present in the estimation procedure given in Equation (2).

(2) the log-likelihood function for $(\boldsymbol{\theta}_k, \mathbf{w})$, conditional on complete data $(\mathbf{y}, \mathbf{c})$, is given by

$$l(\boldsymbol{\theta}_k, \mathbf{w}|\mathbf{y}, \mathbf{c}) = \log\left\{\prod_{j=1}^{k} w_j^{n_j} L(\theta_j|\mathbf{y})\right\} = \sum_{j=1}^{k} \left[n_j \log(w_j) + l(\theta_j|\mathbf{y})\right],$$

where $l(\theta_j|\mathbf{y}) = \log\left[L(\theta_j|\mathbf{y})\right]$ is the log-likelihood function for component j, for $j = 1, \ldots, k$. Thus, the estimation procedure of the k component parameters reduce to k independent problems of estimation. For example, for a normal mixture model, the maximum likelihood estimates for component parameters $\theta_j = (\mu_j, \sigma_j^2)$ is $\hat{\theta}_j = (\hat{\mu}_j, \hat{\sigma}_j^2) = (\overline{y}_j, s_j^2)$, where $\overline{y}_j$ and s_j^2 are, respectively, the average and variance of the observations allocated to component j, for $j = 1, \ldots, k$.

From this complete-data formulation, the estimation procedure is given by an iterative process with two steps. In the first one, the allocation indicator variables are updated conditional on component parameters, and in the subsequent step, the component parameters are updated conditional on configuration of the allocation indicator variables.

The usual algorithm used to implement these two steps is the EM algorithm [3]. The stochastic version of the EM algorithm (SEM) can be implemented according to Algorithm 1.

Algorithm 1 SEM Algorithm

1: Initialize the algorithm with a configuration $\mathbf{c}^{(0)} = \left(c_1^{(0)}, \ldots, c_n^{(0)}\right)$ for allocation indicator variables.
2: **procedure** For the s-th iteration of the algorithm, $s = 1, \ldots$
3: get the maximum likelihood estimates $\hat{\boldsymbol{\theta}}_k^{(s)} = \left(\hat{\theta}_1^{(s)}, \ldots, \hat{\theta}_k^{(s)}\right)$ and $\hat{\mathbf{w}}^{(s)} = \left(\hat{w}_1^{(s)}, \ldots, \hat{w}_k^{(s)}\right)$ conditional on configuration $\mathbf{c}^{(s-1)}$;
4: **if** $\left| \frac{l\left(\hat{\boldsymbol{\theta}}_k^{(s)}, \hat{\mathbf{w}}^{(s)} | \mathbf{y}\right) - l\left(\hat{\boldsymbol{\theta}}_k^{(s-1)}, \hat{\mathbf{w}}^{(s-1)} | \mathbf{y}\right)}{l\left(\hat{\boldsymbol{\theta}}_k^{(s-1)}, \hat{\mathbf{w}}^{(s-1)} | \mathbf{y}\right)} \right| < \epsilon$, where ϵ is a threshold value previously fixed, **then** stop the algorithm. Otherwise, go to item (iii);
5: conditional on $\hat{\boldsymbol{\theta}}_k^{(s)}$ and $\hat{\mathbf{w}}^{(s)}$, update $\mathbf{c} = (c_1, \ldots, c_n)$ as follows. For $i = 1, \ldots, n$ and $j = 1, \ldots, k$ do the following:
6: Let $\mathbf{z}_i = (z_{i1}, \ldots, z_{ik})$ be a indicator vector, so that $z_{ij} = 0$ or $z_{ij} = 1$;
7: Generate $\mathbf{z}_i \sim Multinomial(1, \mathbf{w}_i^*)$, where $\mathbf{w}_i^* = (w_{i1}^*, \ldots, w_{ik}^*)$ and w_{ij}^* is obtained from Equation (3) doing $\theta_j = \hat{\theta}_j$ and $w_j = \hat{w}_j$. If $z_{ij} = 1$, then do $c_i = j$;
8: Do $s = s + 1$ and return to step (3).

Although it is simple to implement computationally, the SEM algorithm may present some practical problems. As discussed by [16], the algorithm may present a slow convergence. Due to this, some authors, such as [17,18], discuss how to set up the start values in order to increase the convergence. In addition, [15] discusses the non-existence of the global maximum estimator.

Moreover, in this algorithm, the k value must be known previously. For the cases in which k is an unknown quantity, the best k value is chosen by fitting a set of models associated with a set of predefined k values and comparing them according to AIC [4,5] or BIC [6] criteria. Furthermore, given a sample of size n and fixed a k value, there exists a positive probability, given by $(1 - w_j)^n \neq 0$, of the j-th component not having observations allocated in an iteration of the algorithm. In this case, we have an empty component, and the maximum likelihood estimates cannot be calculated for these components. Thus, in order to avoid the practical problems presented by the EM algorithm, we propose an integrated approach.

3. Integrated Approach

We start our integrated approach linking data clustering to a mixture model. For this, consider a sampling process from a heterogeneous population that is subdivided into k sub-populations. Thus, it is natural to assume that the sampling process consists of the realization of the following steps:

(i) choose a sub-population j with probability w_j, where w_j is the relative frequency of the j-th sub-population in relation to the overall population;
(ii) sample a Y_i value of this sub-population,

for $i = 1, \ldots, n$ and $j = 1, \ldots, k$, where n is the sample size.

Let (Y_i, c_i) be a sample unit, where c_i is an indicator allocation variable that assumes a value of the set $\{1, \ldots, k\}$ with probabilities $\{w_1, \ldots, w_k\}$, respectively. Thus, assuming that subpopulation j is modeled by a probability distribution $F(\theta_j)$ indexed by parameter θ_j (scalar or vector), we have that

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

for $i = 1, \ldots, n$ and $j = 1, \ldots, k$.

However, in clustering problems, the c_i's values are non-observable. Thus, the probability of $c_i = j$ is w_j, and the marginal probability density function for $Y_i = y_i$ is given by Equation (1).

In addition, as the model in Equation (1) is a population model; so there exists a non-null probability $(1 - w_j)^n$ that the j-th component is an empty component. Thus, the number of clusters

(i.e., non-empty components) is smaller than the number of components k. As viewed in the description of the EM algorithm, the number of clusters is defined by the configuration of the latent allocation variables $\mathbf{c}$; thus hereafter, we will denote the number of clusters by $k_{\mathbf{c}}$, for $k_{\mathbf{c}} \leq k$.

Since the interest lies in the configuration of $\mathbf{c}$, let us marginalize out the weights of the mixture model. Thus, integrating density (4) with respect to the prior *Dirichlet* $\left(\frac{\gamma}{k}, \ldots, \frac{\gamma}{k}\right)$ distribution of the weights, denoted by $(w_1, \ldots, w_k)|k, \gamma \sim$ *Dirichlet* $\left(\frac{\gamma}{k}, \ldots, \frac{\gamma}{k}\right)$, the joint probability for $\mathbf{c}$ is given by (see Appendix 3 of the SM)

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

Similarly, the conditional probability for $c_i = j$ given $\mathbf{c}_{-i} = (c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n)$, 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 allocated to the j-th component, excluding the i-th observation, for $i = 1, \ldots, n$ and $j = 1, \ldots, k$.

As the main interest is in $k_{\mathbf{c}}$ and not k, we remove k from Equation (6) by letting k tend to infinity. Under this assumption, the probability reaches the following limit:

$$\pi(c_i = j|\mathbf{c}_{-i}, \gamma) = \frac{n_{j,-i}}{n + \gamma - 1}, \tag{7}$$

when $n_{j,-i} > 0$, for $i = 1, \ldots, n$ and $j = 1, \ldots, k_{\mathbf{c}}$, where $k_{\mathbf{c}}$ is the number of clusters defined by configuration $\mathbf{c}$. In addition, we now have a probability of the i-th observation being allocated to one of the other infinite components, which is given by

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

for $j^* \notin \{1, \ldots, k_{\mathbf{c}}\}$. This is the probability of the observation y_i creating a new cluster, for $i = 1, \ldots, n$. The probabilities in (7) and (8) are equivalent to the update probabilities of a Dirichlet process mixture model. See, for example, [19–21].

Given y_i, the conditional probability for $c_i = j$, such that $n_{j,-i} > 0$, is

$$\pi_{ij} = \pi(c_i = j|y_i, \theta_j, \mathbf{c}_{-i}, \gamma) = \frac{n_{j,-i}}{n + \gamma - 1} f(y_i|\theta_j), \tag{9}$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, k_{\mathbf{c}_{-i}}$, where $k_{\mathbf{c}_{-i}}$ is the number of clusters excluding the i-th observation. At this point, it is important to note that if an observation y_i is allocated to 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 $k_{\mathbf{c}_{-i}} = k_{\mathbf{c}} - 1$.

In order to define the conditional probability of the i-th observation creating a new cluster j^*, we integrate parameters out for this case, for $j^* = k_{\mathbf{c}_{-i}} + 1$. This was done because that probability does not depend on the parameter value θ_{j^*}. Thus, the conditional posterior probability for $C_i = j^*$ is

$$\pi_{ij^*} = \pi(c_i = j^*|y_i, \mathbf{c}_{-i}, \gamma) = \frac{\gamma}{n + \gamma - 1} \mathbf{I}(y_i), \tag{10}$$

where $\mathbf{I}(y_i) = \int f(y_i|\theta_{j^*}) \pi(\theta_{j^*}) d\theta_{j^*}$ and $\pi(\theta_{j^*})$ is the density of the prior distribution for θ_{j^*}, for $i = 1, \ldots, n$.

As is known from the literature, the likelihood function for a mixture model is non-identifiable, i.e., any permutation of the components' labels lead to the same likelihood function (see, for example, [8,11,22–24]). Thus, in order to get identifiability, we assume that $\mu_1, \ldots, \mu_{k_{\mathbf{c}}}$ are the component means for clusters and that $\mu_1 < \ldots < \mu_{k_{\mathbf{c}}}$. However, it does not prevent the algorithm

described in the next Section from being applicable to another labeling criterion. Additional discussion about label switching can be found in [22,23].

3.1. Integrated SEM Algorithm

Using probabilities given in Equations (9) and (10), we update the allocation indicator variables according to Algorithm 2.

Conditional on a configuration $\mathbf{c}$, we have $k_{\mathbf{c}}$ clusters. So, we update parameters of interest according to Algorithm 3. We then join Algorithms 2 and 3 to get the Algorithm 4.

After the S iterations, we discard the first B iterations as a burn-in. In the following, we also consider "jumps" of size h, i.e., only one draw every h is extracted from the original sequence in order to obtain a sub-sequence of size $H = (S - B)/h$ to make inferences. Denote this sub-sequence by $\mathbb{S}(H)$.

Consider $\mathbb{N}_{k_{\mathbf{c}}}(j)$ to be the number of times that $k_{\mathbf{c}} = j$ in $\mathbb{S}(H)$, for $j \in \{1, \ldots, K_{max}\}$, where K_{max} is the maximum $k_{\mathbf{c}}$ value sampled in the course of iterations. Thus, $\tilde{P}(k_{\mathbf{c}} = j) = \frac{\mathbb{N}_{k_{\mathbf{c}}}(j)}{H}$ is the estimated probability for $k_{\mathbf{c}} = j$. We then consider

$$\tilde{k}_{\mathbf{c}} = \underset{1 \leq j \leq K_{max}}{argmax} \left(\tilde{P}(k_{\mathbf{c}} = j) \right)$$

as being the estimates for the number of components $k_{\mathbf{c}}$.

Appendix 1 of the SM presents the mathematical expression used to determine a configuration for $\mathbf{c}$ and estimates for the parameters of the clusters, conditional on the estimate $\tilde{k}_{\mathbf{c}}$.

Algorithm 2 Updating $\mathbf{c}$

1: Let $\mathbf{c} = (c_1, \ldots, c_n)$ be the current configuration for latent allocation variables. Then, update $\mathbf{c}$ as follows.
2: **procedure** For $i = 1, \ldots, n$:
3: Remove c_i from the current state $\mathbf{c}$, obtaining $\mathbf{c}_{-i}$ and $k_{\mathbf{c}_{-i}}$;
4: Generate a 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 π_{ij} given in (9) and π_{ij^*} given in (10), for $j = 1, \ldots, k_{\mathbf{c}_{-i}}$ and $j^* = k_{\mathbf{c}_{-i}} + 1$;
5: If $Z_{ij} = 1$, for $j \in \{1, \ldots, k_{\mathbf{c}_{-i}}\}$, set up $c_i = j$ and do $n_j = n_{j,-i} + 1$;
6: If $Z_{ij^*} = 1$, do $n_{j^*} = 1$ and $k_{\mathbf{c}} = k_{\mathbf{c}_{-i}} + 1$. As this new cluster has just one observation allocated, set as component parameter $\theta_{j^*} = \theta_j^g$, where θ_j^g is a value generated from posterior distribution $\pi(\theta_{j^*}|y_i) \propto f(y_i|\theta_{j^*})\pi(\theta_{j^*})$, where $f(y_i|\theta_{j^*})$ is the probability density function for y_i conditional on θ_{j^*} and $\pi(\theta_{j^*})$ is the density of the prior distribution for θ_{j^*}. Relabel the $k_{\mathbf{c}}$ clusters in order to maintain the adjacency condition. If the component mean μ_{j^*} of the new cluster is such that:
7: $\mu_{j^*} = \underset{1 \leq j \leq k_{\mathbf{c}}}{min} \mu_j$, then do $j^* = 1$ and relabel all other clusters doing $j + 1$;
8: $\mu_{j^*} = \underset{1 \leq j \leq k_{\mathbf{c}}}{max} \mu_j$, then do $j^* = k_{\mathbf{c}}$ and keep all other clusters labels;
9: $\mu_j < \mu_{j^*} < \mu_{j+1}$, for $j \neq \{1, k_{\mathbf{c}}\}$, then do $j^* = j + 1$ and relabel all other clusters $j' \geq j + 1$ doing $j' = j' + 1$.

Algorithm 3 Updating cluster parameters

1: Let $\boldsymbol{\theta}_{k_{\mathbf{c}}} = (\theta_1, \ldots, \theta_{k_{\mathbf{c}}})$ be the current parameter values of the clusters. Conditional on configuration $\mathbf{c}$, get $\boldsymbol{\theta}_{k_{\mathbf{c}}}^{updated} = (\theta_1^{updated}, \ldots, \theta_{k_{\mathbf{c}}}^{updated})$ as follows:
2: **if** cluster j is such that $n_j > 1$, **then** do $\theta_j^{updated} = \hat{\theta}_j$, where $\hat{\theta}_j$ are the maximum likelihood estimates of the j-th cluster;
3: **if** cluster j is such that $n_j = 1$, **then** generate θ_j^g from conditional posterior distribution $\pi(\theta|y_i)$ and set $\theta_j^{updated} = \theta_j^g$;
4: Do $\boldsymbol{\theta}_k = \boldsymbol{\theta}_k^{updated}$ only if the adjacency condition $\mu_1^{updated} < \ldots < \mu_{k_{\mathbf{c}}}^{updated}$ is met. Otherwise, keep $\boldsymbol{\theta}_{k_{\mathbf{c}}}$ as the current value.

Algorithm 4 ISEM Algorithm

1: Initialize the algorithm with a configuration $\mathbf{c}^{(0)} = \left(c_1^{(0)}, \ldots, c_n^{(0)}\right)$ for allocation indicator variables.
2: **procedure** For the s-th iteration of the algorithm, $s = 1, \ldots, S$, do the following.
3: Conditional on $\mathbf{c}^{(s-1)}$, update the parameters of the clusters according to algorithm 3 ;
4: Obtain a new configuration $\mathbf{c}^{(s)}$ for the allocation of indicator variables using algorithm 2.

4. Simulation Study

In this section, we describe the results from a simulation study carried out to verify the performance of the proposed algorithm. To generate the artificial datasets, we considered univariate normal mixture models. We set up the number of clusters and parameter values according to the specified values in Table 2. We also fixed the sample size equal to $n = 200$.

Table 2. Number of clusters and parameter values used for simulating the datasets.

Artificial Dataset	Number of Clusters	Parameter Values
A_1	$k_{\mathbf{c}} = 2$	$\mu_1 = 0, \quad \mu_2 = 3,$ $\sigma_1^2 = 1, \quad \sigma_2^2 = 1,$ $w_1 = 0.80, w_2 = 0.20,$
A_2	$k_{\mathbf{c}} = 3$	$\mu_1 = -6, \quad \mu_2 = 0, \quad \mu_3 = 4$ $\sigma_1^2 = 3, \quad \sigma_2^2 = 2, \quad \sigma_3^2 = 1$ $w_1 = 0.50, w_2 = 0.30, w_3 = 0.20$
A_3	$k_{\mathbf{c}} = 4$	$\mu_1 = -6, \quad \mu_2 = 0, \quad \mu_3 = 7, \quad \mu_4 = 14$ $\sigma_1^2 = 1, \quad \sigma_2^2 = 2, \quad \sigma_3^2 = 2, \quad \sigma_4^2 = 1$ $w_1 = 0.10, w_2 = 0.40, w_3 = 0.40, w_4 = 0.10$
A_4	$k_{\mathbf{c}} = 5$	$\mu_1 = -13, \quad \mu_2 = 7, \quad \mu_3 = 0, \quad \mu_4 = 6, \quad \mu_5 = 11$ $\sigma_1 = 1, \quad \sigma_2 = 2, \quad \sigma_3 = 3, \quad \sigma_4 = 2, \quad \sigma_5 = 1$ $w_1 = 0.15, w_2 = 0.20, w_3 = 0.30, w_4 = 0.20, w_5 = 0.15,$

The procedure for generating the datasets is given by the following 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 $Y_i \sim \mathcal{N}\left(\mu_j, \sigma_j^2\right)$, with fixed parameter values according to Table 2, for $w_0 = 0$ and $j = 1, \ldots, k_{\mathbf{c}}$.

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

Having generated the datasets, we need to define the the probability of creating a new cluster and the posterior distribution for $\theta_{j^*} = \left(\mu_{j^*}, \sigma_{j^*}^2\right)$ given y_i, for $i = 1, \ldots, n$. For this, consider the following conjugated prior distributions for component parameters $\theta_j = \left(\mu_j, \sigma_j^2\right)$:

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

where μ_0, λ, α, and β are hyperparameters. The parametrization of the gamma distribution is such that the mean is α/β and the variance is α/β^2.

Following [11,24], we consider the following procedure to define the values for the hyperparameters. Let R be the observed variation interval of the data and ε its midpoint. Then, we set up $\mu_0 = \varepsilon$ and $E(\sigma_j^{-2}) = R^{-2}$. Thus, we obtain $\beta = \alpha R^2$, and we fix $\alpha = 1$. In addition, to obtain a prior distribution with a large variance, we fixed $\lambda = 10^{-2}$, and for the hyperparameter γ, we consider the value 0.1, $\gamma = 0.1$.

Thus, the probability of creating a new cluster is given by Equation (10), in which

$$\mathbf{I}(y_i) = \left[\frac{\lambda}{2\beta\pi(1+\lambda)}\right]^{\frac{1}{2}} \frac{\Gamma(\alpha+1)}{\Gamma(\alpha)} \left[1 + \frac{y_i^2 + \lambda\mu_0^2}{2\beta} - \frac{(y_i + \lambda\mu_0)^2}{2\beta(1+\lambda)}\right]^{-(\alpha+\frac{1}{2})}, \tag{11}$$

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

When a new cluster is created, the new parameter values $\theta_{j^*} = (\mu_{j^*}, \sigma_{j^*}^2)$ are generated from the following conditional posterior distributions,

$$\mu_{j^*} | \sigma_{j^*}^2, y_i, \mathbf{c}, \mu_0, \lambda \sim \mathcal{N}\left(\frac{y_i + \lambda\mu_0}{1+\lambda}, \frac{\sigma_j^2}{1+\lambda}\right) \tag{12}$$

and

$$\sigma_{j^*}^{-2} | y_i, \mathbf{c}, \tau, \beta \sim \Gamma\left(\alpha + 1, \beta + \frac{y_i^2 + \lambda\mu_0^2}{2} - \frac{(y_i + \lambda\mu_0)^2}{2(1+\lambda)}\right), \tag{13}$$

for $j^* = k_{\mathbf{c}_{-i}} + 1$.

We run the ISEM algorithm for S = 55,000, B = 5000, and h = 10. From these values, we got a sub-sequence $\mathbb{S}(H)$ of size 5000 to make inferences. The algorithm was initialized with $k_{\mathbf{c}} = 1$ and parameter values $\mu_1 = \overline{y}$ and $\sigma_1^2 = s^2$, the sample mean and variance of the generated dataset, respectively.

We also apply to the generated datasets the SEM algorithm, as describe in Section 2, and the RJ algorithm as proposed by [8]. In order to choose the number of clusters using the SEM algorithm, we consider the AIC and BIC model selection criteria. In addition, the algorithm was initialized using a configuration $\mathbf{c}^{(0)}$ obtained via the k-means algorithm [25]. As stop criterion, we set up the threshold $\varepsilon = 0.001$. For the RJ algorithm, we consider the same number of iterations, burn-in, and thin value used in the ISEM algorithm.

In order to compare the three algorithms in terms of the estimation of the number of clusters, we consider $M = 500$ simulated datasets. Table 3 shows the proportion of times that the ISEM and RJ algorithms put the highest estimated probability on the $k_{\mathbf{c}}$ values presented. This table also show the proportion of times that the AIC and BIC indicated the $k_{\mathbf{c}}$ value as the best among the tested values. The values highlighted in bold are the proportions on the $k_{\mathbf{c}}$ true value. As one can note, the ISEM

shows a better performance, i.e., higher proportion of the k_c true value than the other two algorithms, especially in relation to the SEM algorithm with the selection of k_c via the AIC and BIC. The results also show that the AIC and BIC model selection criteria have a low success ratio, with a proportion of the k_c true value smaller than 0.50.

Table 3. Proportion of times the algorithms chose the k_c values as the number of clusters.

Data Set	k_c^{true}	k_c	$\tilde{P}(k_c = j\cdot)$		*AIC*	*BIC*	Data Set	k_c^{true}	k_c	$\tilde{P}(k_c = j\cdot)$		*AIC*	*BIC*
			ISEM	**RJ**						**ISEM**	**RJ**		
A_1	2	1	0.014	0.002	0.050	0.210	A_2	3	1	0.000	0.000	0.000	0.004
		2	**0.976**	**0.972**	**0.294**	**0.448**			2	0.276	0.094	0.104	0.438
		3	0.010	0.026	0.238	0.224			3	**0.720**	**0.672**	**0.304**	**0.384**
		4	0.000	0.000	0.152	0.082			4	0.004	0.232	0.262	0.138
		5	0.000	0.000	0.148	0.028			5	0.000	0.002	0.184	0.028
		6	0.000	0.000	0.118	0.008			6	0.000	0.000	0.146	0.008
A_3	4	1	0.000	0.000	0.000	0.000	A_4	5	1	0.000	0.000	0.000	0.000
		2	0.000	0.004	0.000	0.000			2	0.006	0.000	0.000	0.006
		3	0.000	0.000	0.010	0.066			3	0.006	0.000	0.000	0.018
		4	**0.956**	**0.476**	**0.226**	**0.450**			4	0.218	0.010	0.038	0.210
		5	0.044	0.474	0.252	0.296			5	**0.682**	**0.509**	**0.322**	**0.446**
		6	0.000	0.044	0.214	0.122			6	0.028	0.442	0.246	0.222
		7	0.000	0.000	0.184	0.056			7	0.000	0.039	0.210	0.072
		8	0.000	0.002	0.114	0.010			8	0.000	0.000	0.184	0.026

4.1. Results from a Single Simulated Data Set

We also analyze the results from a single dataset selected at random from the $M = 500$ generated datasets in each situation A_1 to A_4. Then, we discuss the convergence of the ISEM and RJ algorithms based on the sample generated across iterations, using graphical tools. In general, the graphical tools show whether the simulated chain stabilizes in some sense and provide useful feedback about the convergence [26].

Table 4 shows the estimated probabilities of k_c obtained with ISEM and RJ and the AIC and BIC values from the SEM algorithm for the selected dataset. In this table, the values highlighted in bold are the highest estimated probabilities and the smallest AIC and BIC values. As we can note, the ISEM algorithm set up a maximum probability for the k_c true value for the four simulated cases.

The RJ algorithm puts a higher probability on the k_c true value for datasets A_1 and A_2. However, the probability on the k_c true value is smaller than that estimated by ISEM. This indicates a higher precision for the ISEM algorithm. For datasets A_3 and A_4, the RJ attributes maximum probability to the wrong values, $k_c = 5$ and $k_c = 6$, respectively. Moreover, the probabilities estimated by RJ do not evidence a single value for k_c as being the best value since there are different values for k_c with similar probabilities. For example, for dataset A_2, the maximum is at $k_c = 3$ with $P(k_c = 3|\cdot) = 0.3836$, but one can argue that the estimated probabilities favor $k_c = 3$ or $k_c = 4$. For dataset A_3, there is similar support for k_c between 4 and 7, and for A_4 between 5 and 7.

Analogously to ISEM and RJ, the AIC and BIC model selection criteria indicate the k_c true value as the best value for datasets A_1 and A_2. For dataset A_3, similar to the RJ, the AIC indicates the wrong value $k_c = 5$ as the best value, while the BIC indicates the k_c true value as the best value. For dataset A_4, the AIC and BIC indicate the wrong value $k_c = 6$ as the best model.

4.2. An Empirical Check of the Convergence

We now empirically check the convergence of the sequence of the probability for k_c across iterations, the capacity to move for different values of k_c in the course of the iterations, and the estimated autocorrelation function the (acf) for the ISEM and RJ algorithms.

Table 4. Estimated probability for k_c.

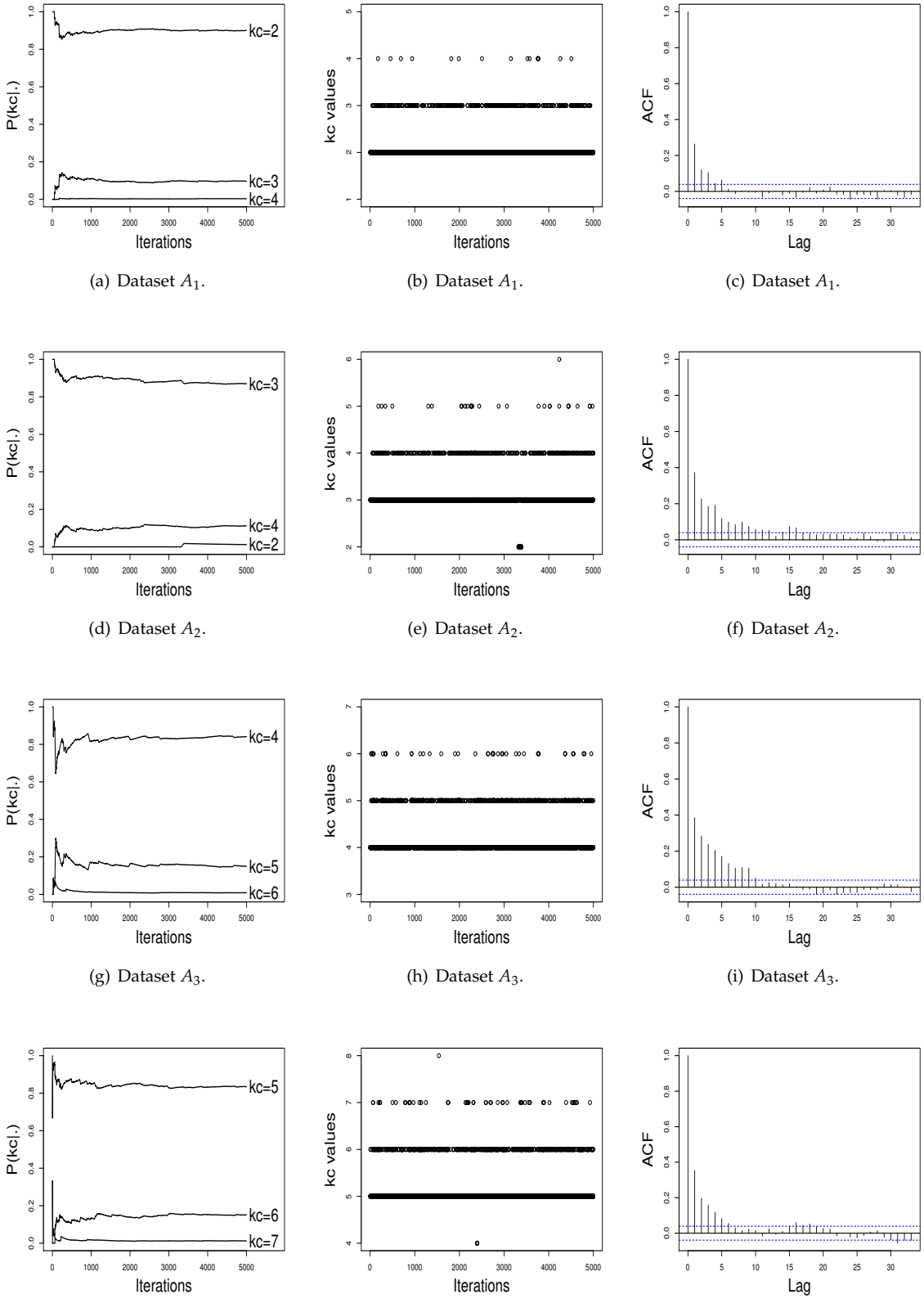
Data Set	k_c^{true}	k_c	$\tilde{P}(k_c = j\cdot)$ ISEM	$\tilde{P}(k_c = j\cdot)$ RJ	*AIC*	*BIC*	Data Set	k_c^{true}	k_c	$\tilde{P}(k_c = j\cdot)$ ISEM	$\tilde{P}(k_c = j\cdot)$ RJ	*AIC*	*BIC*
A_1	2	1	0.0000	0.0000	786.7166	793.3133	A_2	3	1	0.0000	0.0004	1160.758	1167.355
		2	**0.9006**	**0.5252**	**762.5204**	**779.0120**			2	0.0122	0.0136	1129.981	1146.472
		3	0.0962	0.2862	764.1440	790.5305			3	**0.8694**	**0.3836**	**1114.024**	**1140.411**
		4	0.0032	0.1138	769.2648	805.5463			4	0.1124	0.3140	1118.789	1155.070
		5	0.0000	0.0466	768.0492	814.2256			5	0.0058	0.1716	1120.108	1166.284
		6	0.0000	0.0160	775.1082	831.1796			6	0.0002	0.0744	1130.558	1186.630
		≥ 7	0.0000	0.0122	-	-			≥ 7	0.0000	0.0424	-	-
A_3	4	1	0.0000	0.0000	1273.886	1280.482	A_4	5	1	0.0000	0.0002	1416.124	1422.721
		2	0.0000	0.0000	1276.281	1292.773			2	0.0000	0.0004	1388.738	1405.230
		3	0.0000	0.0002	1251.357	1277.743			3	0.0000	0.0028	1358.474	1384.861
		4	**0.8412**	0.1696	1188.470	**1224.751**			4	0.0014	0.0114	1357.037	1393.318
		5	0.1500	**0.3014**	**1186.075**	1232.252			5	**0.8340**	0.2788	1355.922	1402.098
		6	0.0088	0.2400	1191.747	1247.818			6	0.1520	**0.3004**	**1325.927**	**1381.998**
		7	0.0000	0.1632	1197.028	1262.995			7	0.0124	0.2224	1331.940	1397.907
		8	0.0000	0.0816	1200.337	1276.199			8	0.0002	0.0186	1331.352	1407.213
		≥ 9	0.0000	0.0440	-	-			≥ 9	0.0000	0.0750	-	-

Figure 1a,d,g,j presents the graphics of the probability for k_c in the course of the iterations, for the four simulated datasets. To maintain a better visualization, we plot in these graphics only the three higher $P(k_c|\cdot)$ estimates. Observing at these figures, it can be seen that the L iterations and the burn-in value B used were adequate to achieve stability for $P(k_c|\cdot)$. In addition, Figure 1b,e,h,k shows that the ISEM algorithm mixes well over k_c, i.e., "visits" mixture models with different values of k_c across iterations. As shown by Figure 1c,f,i,l, the sampled k_c values also do not have significant autocorrelation function (ACF). Thus, based on these graphical tools, there is no evidence against the convergence of the generated values by the ISEM algorithm.

Figure 2 shows the performance of the RJ algorithm. The probabilities of k_c present a satisfactory stability. The sampled k_c values have a satisfactory mix, and the estimated autocorrelation is non-significant. In addition, as can be noted in Figure 2, probabilities for the number of clusters do not differentiate a value of k_c in order to be chosen as the better value, as done by ISEM. This may happen due the fact that the performance of the RJ depends on the choice of the transition functions to do "good" jumping, meaning that a transition function that is adequate for one dataset may be not for another one. As the ISEM algorithm does not need the specification of transition functions to propose a change of the k_c value, these results shows us that ISEM may be an effective alternative in relation to RJ and SEM algorithms for the joint estimation of k_c and the cluster parameters of a mixture model.

Figure 1 in Appendix 2 of the SM shows the generated values for datasets A_1 to A_4. This Figure also shows the clusters identified by the ISEM algorithm. As can be seen, clusters are satisfactorily identified by the proposed algorithm.

We also compare ISEM and RJ algorithms in terms of CPU computation time. The simulations were realized on a MacBook Pro, 2.5 GHz Intel Core i5 dual core, 4 Gb MHz DDR3. Table 5 shows a summary of the times of iterations for the ISEM and RJ algorithms. The column denoted by s.d. presents the standard deviation values. For dataset A_1, the average time that RJ takes to run one iteration is 1.8491 times greater than the average time that ISEM takes to run an iteration. For datasets A_2, A_3, and A_4, the average time that RJ needs to run one iteration is 1.8175, 2.3239, and 1.8932 times greater than the average time that ISEM takes to run an iteration, respectively. These results show a better performance of the ISEM algorithm. The higher iteration times of the RJ algorithm are mainly due to the split–merge step used to increase the mixing of the Markov chain in relation to the number of clusters.



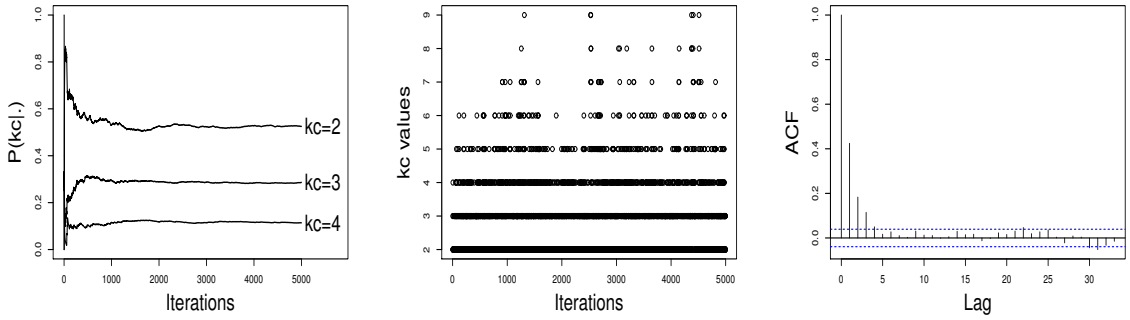
(a) Dataset A_1. (b) Dataset A_1. (c) Dataset A_1.

(d) Dataset A_2. (e) Dataset A_2. (f) Dataset A_2.

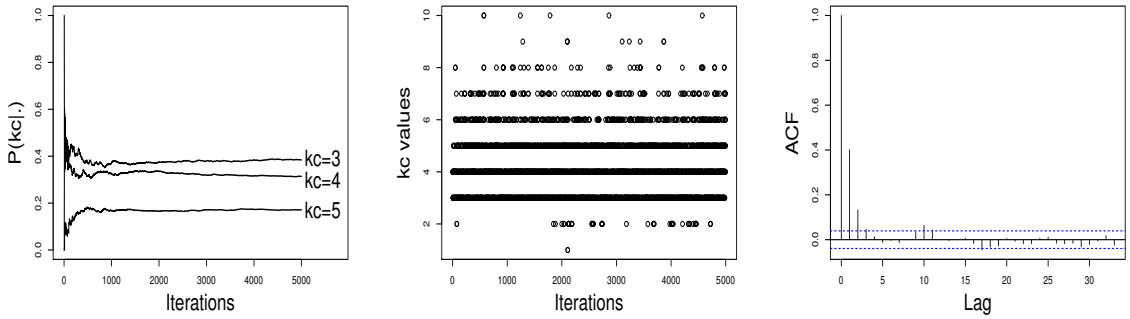
(g) Dataset A_3. (h) Dataset A_3. (i) Dataset A_3.

(j) Dataset A_4. (k) Dataset A_4. (l) Dataset A_5.

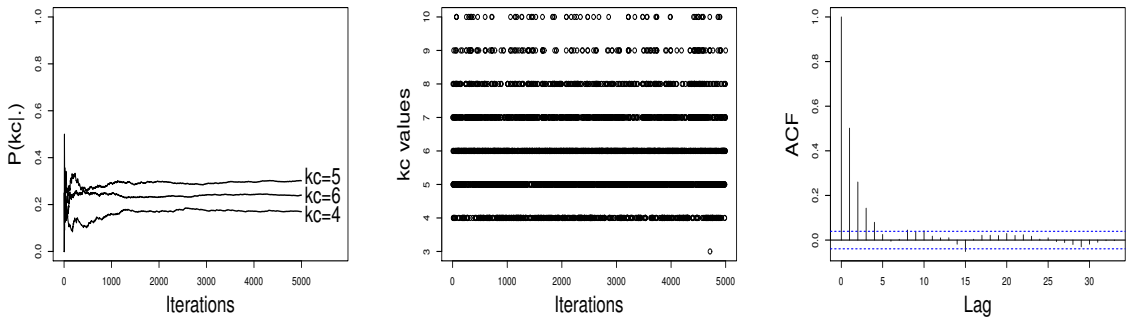
Figure 1. Performance of the ISEM algorithm across iterations.



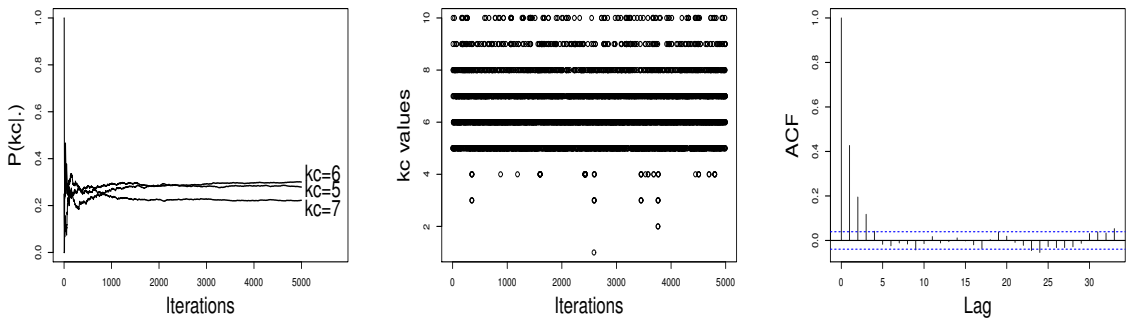
(a) Dataset A_1. (b) Dataset A_1. (c) Dataset A_1.



(d) Dataset A_2. (e) Dataset A_2. (f) Dataset A_2.



(g) Dataset A_3. (h) Dataset A_3. (i) Dataset A_3.



(j) Dataset A_4. (k) Dataset A_4. (l) Dataset A_4.

Figure 2. Performance of the RJ algorithm across iterations.

The results from these simulated datasets show that the ISEM algorithm may be an effective alternative to the RJ and SEM algorithms for data clustering in situations where the number of clusters is a unknown quantity.

Table 5. Times of the iterations, in seconds.

Artificial Dataset	Algorithm	Summary						
		Min	1^o Q.	Med.	Mean	3^o Q.	Max.	s.d.
A_1	ISEM	0.0064	0.0082	0.0091	0.0109	0.0105	0.4987	0.0107
	RJ	0.0032	0.0137	0.0158	0.0208	0.0202	0.3855	0.0174
A_2	ISEM	0.0055	0.0100	0.0114	0.0137	0.0146	0.3806	0.0108
	RJ	0.0032	0.0169	0.0196	0.0249	0.0243	0.7709	0.0181
A_3	ISEM	0.0059	0.0112	0.0123	0.0142	0.0139	0.4951	0.0100
	RJ	0.0020	0.0218	0.0255	0.0330	0.0320	0.4785	0.0239
A_4	ISEM	0.0059	0.0130	0.0146	0.0179	0.0187	0.5149	0.0108
	RJ	0.0026	0.0232	0.0266	0.0339	0.0323	0.5490	0.0231

5. Application

The three algorithms are now applied to two real datasets. The first real dataset refers to velocity in km/s of $n = 82$ galaxies from 6 well-separated conic sections of an unfilled survey of the Corona Borealis region. This dataset is known in the literature as the Galaxy data and has already been analyzed by [8,13,22,27], among others. This dataset is available in the R software. The second real dataset refers to an acidity index measured in a sample of $n = 155$ lakes in central-north Wisconsin. This dataset was downloaded from the website https://people.maths.bris.ac.uk/$\sim$mapjg/mixdata.

For application of ISEM and RJ algorithms, we consider the same number $L = 5500$, $B = 5000$, and $h = 10$. Table 6 shows the estimated probabilities for k_c obtained with ISEM and RJ and the AIC and BIC values from EM algorithm for each dataset. The maximum probability from ISEM and RJ and the minimum AIC and BIC values are highlighted in bold.

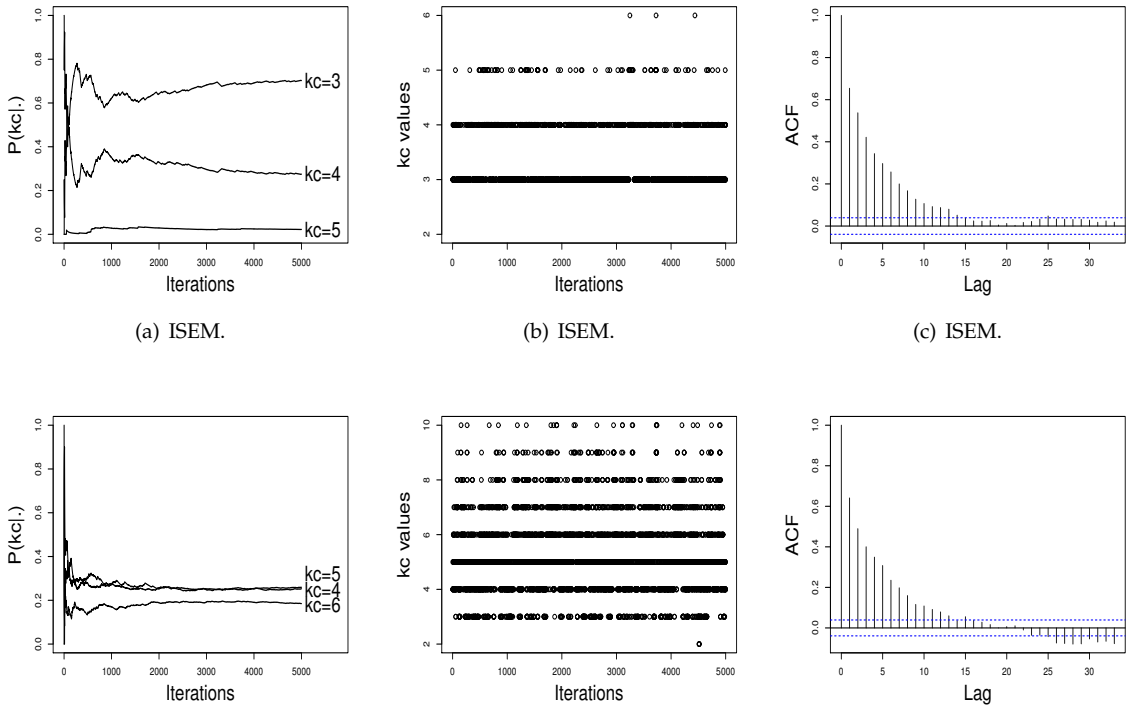
Table 6. Estimated probabilities for k_c, real datasets.

Data Set	k_c	$\tilde{P}(k_c = j\cdot)$		*AIC*	*BIC*	Data Set	k_c	$\tilde{P}(k_c = j\cdot)$		*AIC*	*BIC*
		ISEM	RJ					ISEM	RJ		
Galaxy	1	0.0000	0.0000	484.6819	489.4954	Acidity	1	0.0000	0.0000	455.5740	461.6608
	2	0.0000	0.0008	451.0018	463.0354		2	**0.7194**	0.0502	**380.3449**	**395.5620**
	3	**0.7024**	0.1200	426.7421	445.09959		3	0.2638	**0.3164**	382.7395	407.0869
	4	0.2748	0.2530	427.4915	453.9654		4	0.0152	0.3040	382.3660	415.8437
	5	0.0222	**0.2592**	**410.3666**	**444.0607**		5	0.0016	0.1724	391.7630	434.3709
	6	0.0006	0.1848	413.7755	454.6897		6	0.0000	0.0832	386.1420	437.8802
	7	0.0000	0.1084	422.1793	470.3137		7	0.0000	0.0452	388.1296	448.9981
	8	0.0000	0.0472	423.5542	478.9088		8	0.0000	0.0186	395.3957	465.3945
	$\geq$9	0.0000	00226	-	-		$\geq$9	0.0000	0.0010	-	-

For the Galaxy dataset, the ISEM and RJ algorithms put highest probability on $k_c = 3$ and $k_c = 5$, respectively. However, analogously to the simulation study, the probabilities estimated by RJ do not evidence a single value for k_c as being the best value. For this dataset, the estimated probabilities indicate a k_c value between 3 and 7. The AIC and BIC also indicate $k_c = 5$ as the best value. For the Acidity dataset, ISEM, AIC, and BIC indicate $k_c = 2$ as the best value. The probabilities estimated by RJ attribute similar values for $k_c = 3$ and $k_c = 4$.

Figures 3 and 4 show the performance of the ISEM and RJ algorithms across iterations for the Galaxy and Acidity datasets. The values sampled by the ISEM algorithm present satisfactory stability for estimated probability across iterations, mix well among different k_c values, and present no significant autocorrelation. That is, we do not have evidence against the convergence of the generated chain by the ISEM algorithm. In relation to the RJ, the sampled values mix well and do not present significant autocorrelation. However, although the values sampled by RJ present stability for $P(k_c)$, the estimated probabilities do not differentiate a value of k_c in order to be chosen as the better value, as done by ISEM. This result shows the need to run RJ for a greater number of iterations. With this, we have that for both real datasets, ISEM presents faster convergence than the RJ algorithm.

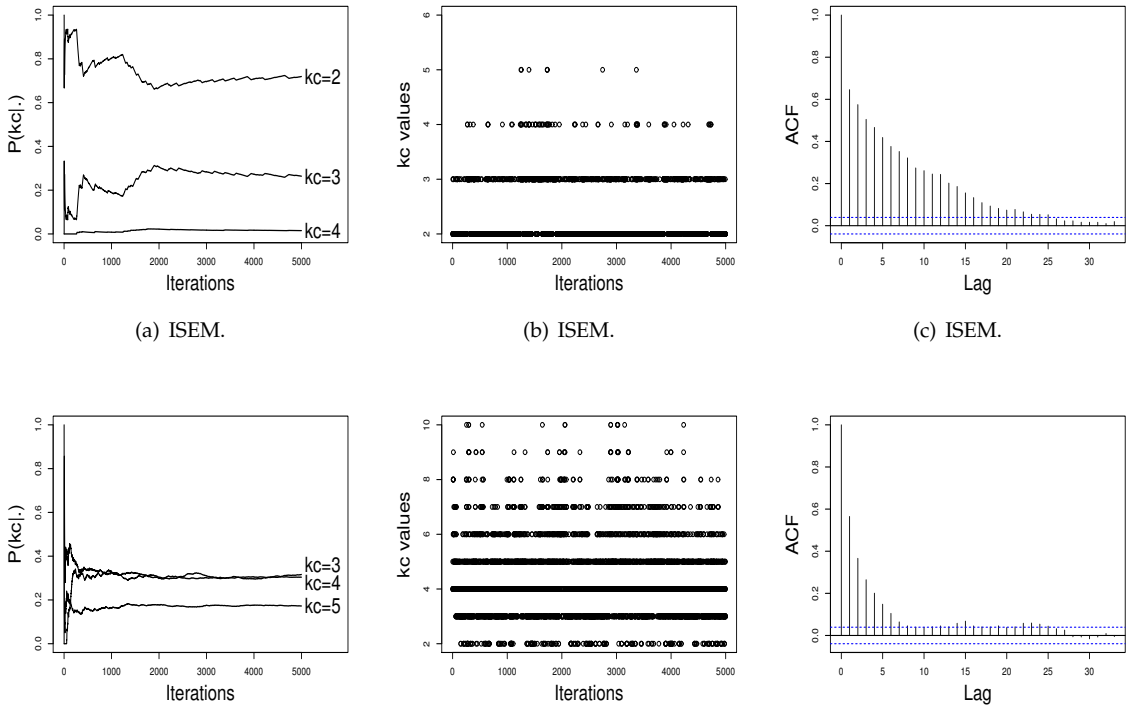
Table 7 shows a summary of the iteration times for the ISEM and RJ algorithms. For the Galaxy data, the average time that ISEM takes to run an iteration is 0.0053 s; while the average time for RJ is 0.0098 s. That is, the average time that RJ takes to run one iteration is 1.8491 times greater than the average time that ISEM takes to run an iteration. For the Acidity data, the average times that the ISEM and RJ algorithms take to run an iteration are 0.0085 and 0.0180 s, respectively. For this dataset, the average time that RJ needs to run an iteration is 2.2118 times greater than the average time that ISEM runs. Similarly to results from the simulation study, ISEM presents better results, i.e., a shorter time to run the iterations.



(a) ISEM. (b) ISEM. (c) ISEM.

(d) RJ. (e) RJ. (f) RJ.

Figure 3. Performance of the ISEM and RJ algorithms for the Galaxy data.



(a) ISEM. (b) ISEM. (c) ISEM.

(d) RJ. (e) RJ. (f) RJ.

Figure 4. Performance of the RJ algorithm across iterations for the Acidity data.

Table 7. Iteration times in seconds.

Artificial Dataset	Algorithm	Summary						
		Min	1^o Q.	Med.	Mean	3^o Q.	Max.	s.d.
Galaxy	ISEM	0.0023	0.0038	0.0045	0.0053	0.0054	0.2468	0.0062
	RJ	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Acidity	ISEM	0.0055	0.0100	0.0114	0.0137	0.0146	0.3806	0.0108
	RJ	0.0046	0.0128	0.0149	0.0188	0.0180	0.4588	0.0160

6. Final Remarks

This article presents a discussion of how to estimate the parameters of a mixture model in the context of data clustering. We propose an alternative algorithm to the EM algorithm called ISEM. This algorithm was developed through an integrated approach in order to allow $k_{\mathbf{c}}$ to be estimated jointly with the other parameters of interest. In the ISEM algorithm, the allocation probabilities depend on the number of clusters $k_{\mathbf{c}}$ and are independent of the number of components k of the mixture model.

In addition, there exists a positive probability of a new cluster being created by a single observation. This is an advantage of the algorithm because it creates a new cluster without the need to specify transition functions. In addition, the cluster parameters are updated according to the number of allocated observations. For the clusters with at least two of these observations, the values of the parameters are taken by the maximum likelihood estimates. For a cluster with just one observation, the parameter values are generated from the posterior distribution.

In order to illustrate the performance of the ISEM algorithm, we developed a simulation study. In this simulation study, we considered four scenarios with artificial data generated from Gaussian mixture models. In addition, each one of the four scenarios was replicated $M = 500$ times, and the proportion of times that ISEM put a higher probability on the $k_{\mathbf{c}}$ true value was recorded. We applied

this same procedure to the EM algorithm, choosing the number of clusters $k_{\mathbf{c}}$ via the AIC and BIC, and to the RJ algorithm. Then, the three algorithms were compared in terms of proportion of times that the $k_{\mathbf{c}}$ true value was selected as the best value. The results obtained show that the ISEM algorithm outperforms the RJ and SEM algorithms. Moreover, the results also show that the AIC and BIC model selection criteria should not be used to determine the number of clusters in a mixture model due to a low success rate.

We also compared the performance of ISEM and RJ in terms of empirical convergence of the sequence of values generated using graphical tools. For this, we selected at random an artificial dataset from each scenery, and then we plotted the probability estimates for $k_{\mathbf{c}}$ across iterations, the generated $k_{\mathbf{c}}$ values, and the estimated autocorrelation of the sampled values (see Figures 1 and 2). Again, the results show a better performance for the ISEM algorithm. While ISEM presents satisfactory stability for the probability of $k_{\mathbf{c}}$ and differentiates the true $k_{\mathbf{c}}$ as the best value, the probabilities estimated by RJ do not differentiate a value of $k_{\mathbf{c}}$ in order to be chosen as the better value.

In order to illustrate the practical use of the proposed algorithm and compare its performance with the SEM and RJ algorithms, we applied the three algorithms to two real datasets: the Galaxy and Acidity datasets. For the Galaxy dataset, ISEM indicates $k_{\mathbf{c}} = 3$ with probability $P(k_{\mathbf{c}} = 3|\cdot) = 0.7024$, while the RJ algorithm, the AIC, and the BIC indicate $k_{\mathbf{c}} = 5$. However, as shown in Figure 3d, the RJ algorithm again does not differentiate a value of $k_{\mathbf{c}}$, while ISEM differentiates the $k_{\mathbf{c}} = 3$ value, and the generated values across iterations present satisfactory stability. For the Acidity dataset, the ISEM, AIC, and BIC indicate $k_{\mathbf{c}} = 2$ as the best value, while RJ attributes similar probabilities for $k_{\mathbf{c}} = 3$ and $k_{\mathbf{c}} = 4$.

As mentioned in the Introduction, the generalization of the proposed algorithm for the multivariate case is the next step of our research. The simulation study and the application were done in R software, and the computational codes can be obtained by emailing the authors.

Supplementary Materials: The following are available online at http://www.mdpi.com/1099-4300/21/11/1063/s1.

Author Contributions: E.F.S. and C.A.d.B.P. developed the whole theoretical part of the research. E.F.S., A.K.S. and L.A.M. developed the simulation studies and real data application.

Funding: This research was funded by Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, grant number 308776/2014-3.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bouveyron, C.; Brunet, C. Model-based clustering of high-dimensional data: A review. *Comput. Stat. Data Anal.* **2014**, *71*, 52–78. [CrossRef]
2. Oh, M.S.; Raftery, A.E. Model-based clustering with dissimilarities: A bayesian approach. *J. Comput. Graph. Stat.* **2007**, *16*, 559–585. [CrossRef]
3. Dempster, A.; Laird, N.; Rubin, D. Maximum likelihood from incomplete data via EM algorithm. *J. Roy. Statist. Soc. B* **1977**, *39*, 1–38. [CrossRef]
4. Akaike, H.A. New look at the statistical model identification. *IEEE Trans. Autom. Control* **1974**, *19*, 716–723. [CrossRef]
5. Bozdogan, H. Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. *Psychometrica* **1987**, *52*, 345–370. [CrossRef]
6. Schwarz, G.E. Estimating the dimension of a model. *Ann. Stat.* **1978**, *6*, 461–464. [CrossRef]
7. Saraiva, E.F.; Suzuki, A.K.; Milan, L.A. A Bayesian sparse finite mixture model for clustering data from a heterogeneous population. *Braz. J. Probab. Stat.* **2019**.
8. Richardson, S.; Green, P.J. On Bayesian analysis of mixtures with an unknown number of components. *J. R. Stat. Soc. Ser. B* **1997**, *59*, 731–792; errata, *J. R. Stat. Soc. Ser. B* **1998**, *60*, U3. [CrossRef]
9. Jain, S.; Neal, R. A split–merge markov chain monte carlo procedure for the Dirichlet process mixture models. *J. Comput. Graph. Stat.* **2004**, *13*, 158–182. [CrossRef]

10. Jain, S.; Neal, R. Splitting and merging components of a nonconjugated Dirichlet process mixture model. *Bayesian Anal.* **2007**, *3*, 445–472. [CrossRef]
11. Saraiva, E.F.; Pereira, C.A.B.; Suzuki, A.K. A data-driven selection of the number of clusters in the Dirichlet allocation model via Bayesian mixture modelling. *J. Stat. Comput. Simul.* **2019**, *89*, 2848–2870. [CrossRef]
12. Diebolt, J.; Robert, C. Estimation of finite mixture distributions by Bayesian sampling. *J. R. Stat. Soc. B* **1994**, *56*, 363–375. [CrossRef]
13. Escobar, M.D.; West, M. Bayesian Density Estimation and Inference Using Mixtures. *J. Am. Stat. Assoc.* **1995**, *90*, 577–588. [CrossRef]
14. Dellapotas, P.; Papgeorgiou, I. Multivariate mixtures of normals with unknown number of components. *Stat. Comput.* **2006**, *16*, 57–68. [CrossRef]
15. McLachlan, G.; Peel, D. *Finite Mixture Models*; Wiley Interscience: New York, NY, USA, 2000.
16. Finch, S.; Mendell, N.; Thode, H. Probabilistic measures of adequacy of a numerical search for a global maximum. *J. Am. Stat. Assoc.* **1989**, *84*, 1020–1023. [CrossRef]
17. Karlis, D.; Xekalaki, W.D. Choosing initial values for the EM algorithm for Finite mixtures. *Comput. Stat. Data Anal.* **2003**, *41*, 577–590. [CrossRef]
18. Biernacki, C.; Celeux, G.; Govaert, G. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Comput. Stat. Data Anal.* **2003**, *41*, 561–575. [CrossRef]
19. Ferguson, S.T. A bayesian analysis of some nonparametric problems. *Ann. Stat.* **1973**, *2*, 209–230. [CrossRef]
20. Blackwell, D.; MacQueen, J.B. Ferguson distributions via Polya urn scheme. *Ann. Stat.* **1973**, *1*, 353–355. [CrossRef]
21. Antoniak, C.E. Mixture of processes dirichlet with applications to bayesian nonparametric problems. *Ann. Stat.* **1974**, *2*, 1142–1174. [CrossRef]
22. Stephens, M. Dealing with label switching in mixture models. *J. R. Stat. Soc. B* **2000**, *62*, 795–809. [CrossRef]
23. Jasra, A.; Holmes, C.C.; Stephens, D.A. Markov Chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Stat. Sci.* **2005**, *20*, 50–67. [CrossRef]
24. Saraiva, E.F.; Suzuki, A.K.; Louzada, F.; Milan, L.A. Partitioning gene expression data by data-driven Markov chain Monte Carlo. *J. Appl. Stat.* **2016**, *43*, 1155–1173. [CrossRef]
25. MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*; University of California Press: Berkeley, CA, USA, 1967; pp. 281–297.
26. Sinharay, S. Assessing Convergence of the Markov Chain Monte Carlo Algorithms: A Review. *ETS Res. Rep. Ser.* **2003**, *2003*, i-52. Available online: http://www.ets.org/Media/Research/pdf/RR-03-07-Sinharay.pdf (accessed on 19 August 2019). [CrossRef]
27. Roeder, K.; Wasserman, L. Practical Bayesian Density Estimation Using Mixture of Normals. *J. Am. Stat. Assoc.* **1997**, *92*, 894–902. [CrossRef]