Technical Report No. 0507, Department of Statistics, University of Toronto

Splitting and Merging Components of a Nonconjugate Dirichlet Process Mixture Model

Sonia Jain [*] Radford M. Neal [†]

5 August 2005

Abstract. The inferential problem of associating data to mixture components is difficult when components are nearby or overlapping. We introduce a new split-merge Markov chain Monte Carlo technique that efficiently classifies observations by splitting and merging mixture components of a nonconjugate Dirichlet process mixture model. Our method, which is a Metropolis-Hastings procedure with split-merge proposals, samples clusters of observations simultaneously rather than incrementally assigning observations to mixture components. Split-merge moves are produced by exploiting properties of a restricted Gibbs sampling scan. A simulation study compares the new split-merge technique to a nonconjugate version of Gibbs sampling and an incremental Metropolis-Hastings technique. The results demonstrate the improved performance of the new sampler. We illustrate the utility of our technique as an unsupervised clustering method using real data.

Key words: Dirichlet process, Markov chain Monte Carlo, split-merge moves, nonconjugate prior

1 Introduction

Bayesian mixture models have gained in popularity as an alternative to traditional density estimation and clustering techniques (see, for example, Escobar and West 1995, Neal 2000, Richardson and Green 1997). In particular, Bayesian mixture models in which a Dirichlet process prior defines the mixing distribution are of interest due to their flexibility in fitting a countably infinite number of components (Ferguson 1983). Much of the recent research related to the Dirichlet process mixture model has been devoted to developing computational techniques, usually Markov chain Monte Carlo methods, to sample from its posterior distribution (Escobar 1994, Bush and MacEachern 1996, Green and Richardson 2001, Neal 2000). Other techniques to estimate the Dirichlet process model include sequential importance sampling (MacEachern, Clyde, and Liu 1999) and variational methods (Blei and Jordan 2004). The practical utility of these methods is illustrated by their recent use for complex biological and genetics problems, such as haplotype reconstruction (Xing, Sharan, and Jordan 2004), estimation of rates of non-synonymous and synonymous nucleotide substitutions

[*] Division of Biostatistics and Bioinformatics, Department of Family and Preventive Medicine, University of California at San Diego, La Jolla, California 92093-0717. Email: sojain@ucsd.edu ; Internet: http://biostat.ucsd.edu

[†] Department of Statistics and Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 3G3. Email: radford@utstat.toronto.edu ; Internet: http://www.cs.toronto.edu/∼radford/

as evidence for natural selection in evolutionary biology problems (Huelsenbeck, Jain, Frost, and Pond 2005), and determination of differential gene expression (Do, Müller, and Tang 2005).

The focus of this article is on Markov chain sampling for nonconjugate Dirichlet process mixture models, building on our previous work for conjugate models (Jain and Neal 2004). Conjugate models are appropriate for some problems, which is convenient due to the analytical tractability of these priors. However, in many situations, conjugate priors can be too restrictive. Forcing conjugacy on the model can lead to undesirable or even nonsensical priors. A classic example is a simple model for normally distributed data, where conjugacy requires an assumption that the mean and variance are *a priori* dependent, which is often unrealistic in actual problems.

Computationally, Markov chain sampling procedures can operate differently depending on whether conjugacy is assumed. In the conjugate case, we can analytically integrate away the mixing proportions for the components and the parameters for each component. This leads to Markov chain Monte Carlo procedures that update only the latent indicator variable associating mixture components with data observations (MacEachern 1994, Neal 1992). However, in the nonconjugate case, the parameters of the model cannot be integrated away and must be included in the Markov chain update. Further, since we lose the advantage of analytic tractability, computational difficulties arise, which makes it more difficult, but not impossible, to construct valid Markov chain Monte Carlo procedures.

Nonconjugate Markov chain sampling methods based on the Gibbs sampler have been proposed previously; see, for instance, MacEachern and Müller (1998) and Neal (2000). When the mixture components are nearby or overlapping, these incremental samplers (as well as those for conjugate models) suffer from computational difficulties, such as remaining stuck in isolated modes and poor mixing between components.

Alternative nonincremental Markov chain samplers for the Dirichlet process mixture model based on split-merge moves have been proposed by Green and Richardson (2001) and by ourselves (Jain and Neal 2004). In a single iteration, these methods can split a mixture component moving all observations to an appropriate new component, or merge two distinct components together. The Green and Richardson (2001) method is based on the reversible-jump procedure, in which numerous ways to propose a split move are possible. Since specific moment conditions must be preserved, the split-merge proposals are model-dependent. Jain and Neal (2004) introduce a Metropolis-Hastings technique with split-merge proposals for conjugate Dirichlet process mixture models. The innovation in this work is exploiting properties of a Gibbs sampling scan to construct split-merge moves, such that their Metropolis-Hastings proposals are model-independent. In this article, we extend the conjugate split-merge technique to a class of nonconjugate Dirichlet process mixture models.

This article is organized as follows. Section 2 defines the nonconjugate Dirichlet process mixture model under consideration. Section 3 briefly describes the Metropolis-Hastings split-merge technique based on Gibbs sampling proposals. Our new split-merge technique for a class of nonconjugate models is proposed in Section 4. Next, in Section 5, we illustrate the utility of our method in high-dimensional problems by comparing it to an auxiliary Gibbs sampling method (Neal 2000, Algorithm 8). In Section 6, we apply the new algorithm to a real data set and demonstrate its performance as an unsupervised clustering method. Section 7 is a general discussion and concluding remarks.

2 The model

The Dirichlet process mixture model takes the following hierarchical model form for observed data $\boldsymbol{y} = (y_1, \ldots, y_n)$ that is considered exchangeable:

$$\begin{aligned} y_i \mid \theta_i &\sim F(\theta_i) \\ \theta_i \mid G &\sim G \\ G &\sim DP(G_0, \alpha) \end{aligned} \tag{1}$$

Here, $F(\theta_i)$ is a component density from a parametric distribution parameterized by θ_i, whose density will be written as $F(y; \theta)$. G is the mixing distribution. G_0 defines a base distribution for the Dirichlet process (DP) prior. Finally, α is a concentration parameter that takes values greater than zero. The usual conditional independence assumptions for a hierarchical model apply, so that the only dependencies are those that are explicitly shown.

Realizations of the Dirichlet process are discrete with probability one. A consequence of this is that the mixture model in equation (1) can be viewed as a countably infinite mixture model (Ferguson 1983). This is evident when we simplify the model in equation (1) by integrating G over its prior distribution. The θ_i follow a generalized Polya urn scheme (Blackwell and MacQueen 1973) and the prior distribution for the θ_i may be represented by the following conditional distributions:

$$\begin{aligned} \theta_1 &\sim G_0 \\ \theta_i \mid \theta_1, \ldots, \theta_{i-1} &\sim \frac{1}{i-1+\alpha} \sum_{j=1}^{i-1} \delta(\theta_j) \; + \; \frac{\alpha}{i-1+\alpha} G_0 \end{aligned} \tag{2}$$

where $\delta(\theta_j)$ is the distribution which is a point mass at θ_j.

We can represent the fact that (2) results in some of the θ_i being identical by setting $\theta_i = \phi_{c_i}$, where c_i represents the latent class associated with observation i, and all ϕ_c are independently drawn from G_0. The Polya urn scheme for sampling the θ_i is equivalent to the following scheme for sampling the latent variables, c_i, and associated ϕ_c:

$$\begin{aligned} P(c_i = c \mid c_1, \ldots, c_{i-1}) &= \frac{n_{i,c}}{i-1+\alpha}, \quad \text{for } c \in \{c_j\}_{j<i} \\ P(c_i \neq c_j \text{ for all } j<i \mid c_1, \ldots, c_{i-1}) &= \frac{\alpha}{i-1+\alpha} \end{aligned} \tag{3}$$

where $n_{i,c}$ is the number of c_k for $k < i$ that are equal to c. The labeling of the indicator c_i is irrelevant in the above probabilities; all that matters is which c_i are equal to each other.

The probabilities shown in (3) define the Dirichlet process model. This notation will be employed in subsequent sections.

3 Jain and Neal's conjugate split-merge procedure

We have previously introduced a split-merge Metropolis-Hastings procedure for conjugate Dirichlet process mixture models (Jain and Neal 2004; Jain 2002). In this version of the algorithm, we assume that F is conjugate to G_0 in equation (1), so the model parameters, ϕ_c, in addition to the mixing distribution, G, can be integrated away. The state of the Markov chain consists only of the mixture component indicators, c_i.

This sampler proposes nonincremental moves that can produce major changes to the configuration of observations to mixture components in a single iteration. The split-merge proposals are evaluated by a Metropolis-Hastings procedure, in which split proposals are constructed by exploiting properties of a *restricted* Gibbs sampling scan on the component indicators, c_i. The Gibbs sampling scan is restricted in that it is only performed on a subset of the data (the observations associated with the merged component that is proposed to be split) and will only allocate observations between two mixture components.

To achieve more reasonable split proposals, several intermediate restricted Gibbs sampling scans are conducted prior to the final restricted Gibbs sampling scan, which is used to calculate the Metropolis-Hastings acceptance probability. The result of the last intermediate Gibbs sampling scan is denoted as the random *launch* state, from which the restricted Gibbs sampling transition probability is explicitly calculated. The number of intermediate restricted Gibbs sampling scans is considered a tuning parameter of this algorithm.

Note that for a merge proposal, there is only one way to combine items in two components to one component. However, deciding whether to accept or reject a merge proposal requires hypothetical consideration of the reverse split, which requires computations similar to those done for an actual split. A description of the steps involved in this algorithm, details to compute the Metropolis-Hastings acceptance probability, and a discussion of the validity of the conjugate version of the split-merge Metropolis-Hastings algorithm are provided in Jain and Neal (2004).

4 The nonconjugate split-merge procedure

Jain and Neal's conjugate split-merge Markov chain procedure described in Section 3 can be generalized to accommodate models with nonconjugate priors. As mentioned earlier, because conjugate priors are not appropriate for all modeling situations, much of the recent Bayesian mixture modeling literature has been dedicated to nonconjugate algorithms (for instance, MacEachern and Müller 1998, Green and Richardson 2001, and Neal 2000). A major impediment in designing nonconjugate procedures is the computational difficulty that arises when the model is no longer analytically tractable.

We say the model is nonconjugate when G_0 is not conjugate to F in the mixture model (equation 1). Aside from being unable to simplify the state of the Markov chain by integrating away the model parameters, ϕ, the main obstacle occurs when trying to sample for a new mixture component. When a c_i is updated, it can be set either to one of the other components currently associated with some observation or to a new mixture component. The probability of setting c_i to a new component involves the integral, $\int F(y_i; \phi)\, dG_0(\phi)$, which is analytically intractable in most nonconjugate situations. Allowances that some previous nonconjugate methods have made when dealing with this integral include approximating the true posterior distribution by another stationary distribution (which can be extremely detrimental) or creating model-specific *ad hoc* algorithms (which fail to generalize well).

Neal (2000) proposed two incremental Markov chain sampling procedures: Gibbs sampling with auxiliary parameters (Algorithm 8), and an incremental Metropolis-Hastings technique (Algorithm 5). These are exact Markov chain Monte Carlo methods that sample the correct posterior distribution and are straightforward to implement. However, in situations where the mixture components are nearby or similar in structure, these incremental methods' performance is analogous to the incremental methods for conjugate models (see Jain and Neal 2004). To overcome their problems, such as remaining stuck in isolated modes and poor mixing between mixture components, we have developed a nonincremental split-merge alternative. In the next section, we compare empirically the performance of the new sampler to Neal's two incremental algorithms.

In this article, we show how such a nonincremental split-merge procedure can be applied when the model uses a particular type of nonconjugate prior, the conditionally conjugate family of priors. In conditionally conjugate models, it is still impossible to efficiently compute the integral, $\int F(y_i;\phi)\,dG_0(\phi)$. However, the pair F and G_0 are conditionally conjugate in one model parameter if the remaining parameters are held fixed. A well-known instance of this is the following Normal model. Suppose the observations, $y_1,\ldots,y_n$, are distributed as $F(y_i;\mu,\sigma^2) = \text{Normal}(y_i;\mu,\sigma^2)$, and the prior is $G_0(\mu,\sigma^{-2}) = \text{Normal}(\mu;w,B^{-1}) \cdot \text{Gamma}(\sigma^{-2};r,R)$. The distributions, $F(y_i;\mu,\sigma^2)$ and $G_0(\mu,\sigma^{-2})$, are conjugate in μ when σ^2 is fixed, and conjugate in σ^2 if μ is fixed. But, the joint posterior distribution is not analytically tractable. For the sake of brevity, when this nonconjugate Normal-Gamma prior is applied to a Normal mixture model, we will refer to it as the Normal-Gamma mixture model. Note, however, that this model using a conjugate prior, in which the mean and variance are *a priori* dependent, is sometime referred to similarly.

Section 4.1 outlines the basic differences between the nonconjugate and conjugate versions of the split-merge procedure. A detailed description of the nonconjugate algorithm is provided in Section 4.2, while Section 4.3 gives the Metropolis-Hastings acceptance probability for the nonconjugate case. We suggest ways to improve the efficiency and performance of the algorithm in Section 4.5.

4.1 Restricted Gibbs sampling split-merge proposals

The conjugate split-merge algorithm of Section 3 cannot be applied directly to the conditionally conjugate case, but the basic mechanism of creating restricted Gibbs sampling split-merge proposals can still be applied. Since the model parameters, ϕ_c, cannot be integrated away, the state of the Markov chain for the split-merge sampler consists of both the component indicators and model parameters, denoted by $\boldsymbol{\gamma} = (\boldsymbol{c},\boldsymbol{\phi})$, where $\boldsymbol{c} = (c_1,\ldots,c_n)$ and $\boldsymbol{\phi} = (\phi_c : c \in \{c_1,\ldots,c_n\})$.

Conditional conjugacy in the model is required so that restricted Gibbs sampling scans can be performed to allocate observations reasonably between two mixture components. During these scans, we do not need to compute the integral, $\int F(y_i;\phi)\,dG_0(\phi)$, since we are only allocating observations between two known components that have at least one observation already assigned to them. For a nonconjugate model, a restricted Gibbs sampling scan also updates the parameters for the affected mixture components, while holding the parameters of the other components fixed. Note that use of a restricted Gibbs sampling scan (and consequently, conditional conjugacy) is only crucial for the final Gibbs sampling scan from the launch state, since it allows the Metropolis-Hastings proposal density can be calculated. The intermediate scans could be replaced by some other type of Markov chain update.

Due to the inclusion of the model parameters, when two separate components are being merged to a single component, there is no longer only one possible component to merge into. The merged component is now defined by component parameters, which must be accounted for in the Metropolis-Hastings acceptance probability (in Section 4.3). The algorithm addresses this problem by conducting intermediate restricted Gibbs sampling for the merged component's parameters to arrive at a *launch state* (in a similar fashion as the "split" intermediate Gibbs sampling). From this launch state, **one** final restricted Gibbs sampling scan is performed to obtain the model parameters of the proposed merged component. The number of intermediate Gibbs sampling scans for the merged component's parameters is an additional tuning parameter in this algorithm. In this generalized version of the split-merge algorithm, there are therefore two launch states, $\boldsymbol{\gamma}^{L_{split}}$ and $\boldsymbol{\gamma}^{L_{merge}}$, that are necessary in order to calculate Gibbs sampling transition kernels for the split and merge proposal distributions.

4.2 Restricted Gibbs sampling split-merge procedure for the nonconjugate case

Let the state of the Markov chain consist of $\boldsymbol{\gamma} = (\boldsymbol{c}, \boldsymbol{\phi})$ where $\boldsymbol{c} = (c_1, \ldots, c_n)$ and $\boldsymbol{\phi} = (\phi_c : c \in \{c_1, \ldots, c_n\})$.

1. Select two distinct observations, i and j, at random uniformly.
2. Let S denote the set of observations, $k \in \{1, \ldots, n\}$, for which $k \neq i$ and $k \neq j$, and $c_k = c_i$ or $c_k = c_j$.
3. Define **launch** states, $\boldsymbol{\gamma}^{L_{split}}$ and $\boldsymbol{\gamma}^{L_{merge}}$, that will be used to define Gibbs sampling distributions required for the split and merge proposals.
   - Obtain launch state $\boldsymbol{\gamma}^{L_{split}} = (\boldsymbol{c}^{L_{split}}, \boldsymbol{\phi}^{L_{split}})$ as follows:
     - If $c_i = c_j$, then let $c_i^{L_{split}}$ be set to a new component such that $c_i^{L_{split}} \notin \{c_1, \ldots, c_n\}$ and let $c_j^{L_{split}} = c_j$. Otherwise, when $c_i \neq c_j$, let $c_i^{L_{split}} = c_i$ and $c_j^{L_{split}} = c_j$. For every $k \in S$, randomly set $c_k^{L_{split}}$, independently with equal probability, to either of the distinct components, $c_i^{L_{split}}$ or $c_j^{L_{split}}$. Initialize model parameters, $\phi^{L_{split}}_{c_i^{L_{split}}}$ and $\phi^{L_{split}}_{c_j^{L_{split}}}$, associated with the two distinct components by drawing new values from their prior distribution.
     - Modify $\boldsymbol{\gamma}^{L_{split}}$ by performing t intermediate restricted Gibbs sampling scans to update $\boldsymbol{c}^{L_{split}}$, $\phi^{L_{split}}_{c_i^{L_{split}}}$, and $\phi^{L_{split}}_{c_j^{L_{split}}}$.
   - Obtain launch state $\boldsymbol{\gamma}^{L_{merge}} = (\boldsymbol{c}^{L_{merge}}, \boldsymbol{\phi}^{L_{merge}})$ as follows:
     - If $c_i = c_j$, then let $c_i^{L_{merge}} = c_j^{L_{merge}} = c_j$ (which is the same as c_i). Similarly, if $c_i \neq c_j$, then set $c_i^{L_{merge}} = c_j^{L_{merge}} = c_j$. For every $k \in S$, set $c_k^{L_{merge}} = c_j$. Initialize model parameter, $\phi^{L_{merge}}_{c_j^{L_{merge}}}$, associated with the merged component by drawing a new value from its prior distribution.
     - Modify $\boldsymbol{\gamma}^{L_{merge}}$ by performing r intermediate restricted Gibbs sampling scans to update $\phi^{L_{merge}}_{c_j^{L_{merge}}}$.
4. If items i and j are in the same mixture component, i.e. $c_i = c_j$, then:

   (a) Propose a new assignment of data items to mixture components, denoted as $\boldsymbol{c}^{split}$, in which component $c_i = c_j$ is split into two separate components, c_i^{split} and c_j^{split}, and propose new values for the corresponding components' parameters, $\phi^{split}_{c_i^{split}}$ and $\phi^{split}_{c_j^{split}}$. Define each element of the candidate state, $\boldsymbol{\gamma}^{split} = (\boldsymbol{c}^{split}, \boldsymbol{\phi}^{split})$, as follows:

- Let $c_i^{split} = c_i^{L_{split}}$ (note that $c_i^{L_{split}} \notin \{c_1, \ldots, c_n\}$)
- Let $c_j^{split} = c_j^{L_{split}}$ (which is the same as c_j)
- By conducting **one** final Gibbs sampling scan from the launch state, $\boldsymbol{\gamma}^{L_{split}}$, for every observation $k \in S$, let c_k^{split} be set to either component c_i^{split} or c_j^{split} and draw values for the model parameters, $\phi^{split}_{c_i^{split}}$ and $\phi^{split}_{c_j^{split}}$.
- For observations $k \notin S \cup \{i, j\}$, let $c_k^{split} = c_k$, and for $c \notin \{c_i^{split}, c_j^{split}\}$, let $\phi^{split}_{c^{split}} = \phi_c$.

(b) Compute the proposal densities, $q(\boldsymbol{\gamma}^{split}|\boldsymbol{\gamma})$ and $q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^{split})$, that will be used to calculate the Metropolis-Hastings acceptance probability.

- Calculate the split proposal density, $q(\boldsymbol{\gamma}^{split}|\boldsymbol{\gamma})$, by computing the Gibbs sampling transition kernel from the split launch state, $\boldsymbol{\gamma}^{L_{split}}$, to the final proposed state, $\boldsymbol{\gamma}^{split}$. The Gibbs sampling transition kernel is the product of the individual probabilities of setting each element in the launch state to its final proposed value during the final Gibbs sampling scan.
- Calculate the corresponding proposal density, $q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^{split})$, by computing the Gibbs sampling transition kernel from the merge launch state, $\boldsymbol{\gamma}^{L_{merge}}$, to the original merged configuration, $\boldsymbol{\gamma}$. The Gibbs sampling transition kernel is the product of the probability of setting each element in the original merge state (in this case, elements of ϕ_{c_j}) to its original value in a (hypothetical) Gibbs sampling scan from the merge launch state.

(c) Evaluate the proposal by the Metropolis-Hastings acceptance probability $a(\boldsymbol{\gamma}^{split}, \boldsymbol{\gamma})$. If the proposal is accepted, $\boldsymbol{\gamma}^{split}$ becomes the next state in the Markov chain. If the proposal is rejected, the original configuration and model parameter, $\boldsymbol{\gamma}$, remain as the next state.

5. Otherwise, if i and j are in different mixture components, i.e. $c_i \neq c_j$, then:

(a) Propose a new assignment of data items to mixture components, denoted as $\boldsymbol{c}^{merge}$, in which distinct components, c_i and c_j, are combined into a single component, and propose a new value for the corresponding merged component's model parameter, $\phi^{merge}_{c_j^{merge}}$. Define each element of the candidate state, $\boldsymbol{\gamma}^{merge} = (\boldsymbol{c}^{merge}, \boldsymbol{\phi}^{merge})$, as follows:

- Let $c_i^{merge} = c_i^{L_{merge}}$ (which is the same as c_j)
- Let $c_j^{merge} = c_j^{L_{merge}}$ (which is the same as c_j)
- For every observation $k \in S$, let $c_k^{merge} = c_j^{L_{merge}}$ (which is the same as c_j)
- For observations $k \notin S \cup \{i, j\}$, let $c_k^{merge} = c_k$, and for $c \neq c^{merge}$, let $\phi^{merge}_{c^{merge}} = \phi_c$.
- Conduct **one** final restricted Gibbs sampling scan from the launch state, $\boldsymbol{\gamma}^{L_{merge}}$, in order to draw a new value for the model parameter, $\phi^{merge}_{c_j^{merge}}$.

(b) Compute the proposal densities, $q(\boldsymbol{\gamma}^{merge}|\boldsymbol{\gamma})$ and $q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^{merge})$, that will be used to calculate the Metropolis-Hastings acceptance probability.

- Calculate the merge proposal density, $q(\boldsymbol{\gamma}^{merge}|\boldsymbol{\gamma})$, by computing the Gibbs sampling transition kernel from the merge launch state, $\boldsymbol{\gamma}^{L_{merge}}$, to the final proposed state, $\boldsymbol{\gamma}^{merge}$. The Gibbs sampling transition kernel is the probability of setting $\phi^{L_{merge}}_{c_j^{L_{merge}}}$ to its final proposed value, $\phi^{merge}_{c_j^{merge}}$, via one Gibbs sampling scan.
- Calculate the corresponding proposal density, $q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^{merge})$, by computing the Gibbs sampling transition kernel from the split launch state, $\boldsymbol{\gamma}^{L_{split}}$, to the original split configuration, $\boldsymbol{\gamma}$. The Gibbs sampling transition kernel is the product of the probabilities of setting each element in the original split state to its original value in a (hypothetical) Gibbs sampling scan from the split launch state.

(c) Evaluate the proposal by the Metropolis-Hastings acceptance probability $a(\boldsymbol{\gamma}^{merge}, \boldsymbol{\gamma})$. If the proposal is accepted, $\boldsymbol{\gamma}^{merge}$ becomes the next state. If the merge proposal is rejected, the original configuration and model parameters, $\boldsymbol{\gamma}$, remain as the next state.

The component labels only serve to distinguish which items are grouped in the same component; the actual numerical values do not matter. A component's label should of course correspond to the label for that component's parameters.

4.3 The Metropolis-Hastings acceptance probability

The Metropolis-Hastings acceptance probability (Metropolis *et al* 1953, Hastings 1970) takes the following form when updating $\boldsymbol{\gamma} = (\boldsymbol{c}, \boldsymbol{\phi})$:

$$a(\boldsymbol{\gamma}^*, \boldsymbol{\gamma}) \quad = \quad \min\left[1, \; \frac{q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^*)}{q(\boldsymbol{\gamma}^*|\boldsymbol{\gamma})} \, \frac{P(\boldsymbol{\gamma}^*)}{P(\boldsymbol{\gamma})} \, \frac{L(\boldsymbol{\gamma}^*|\boldsymbol{y})}{L(\boldsymbol{\gamma}|\boldsymbol{y})}\right] \tag{4}$$

where $\boldsymbol{\gamma}^*$ is either $\boldsymbol{\gamma}^{split}$ or $\boldsymbol{\gamma}^{merge}$ depending on the type of proposal.

The prior distribution, $P(\boldsymbol{\gamma})$, will be a product of the individual prior distributions for $\boldsymbol{c}$ and $\boldsymbol{\phi}$, since they are *a priori* independent. As before, the prior distribution for $P(\boldsymbol{c})$ will be a product of factors in equation (3). The ϕ_c for different mixture components are independent. Therefore, the prior distribution for $P(\boldsymbol{\gamma})$ is:

$$P(\boldsymbol{\gamma}) \quad = \quad P(\boldsymbol{c}) \prod_{c \in \boldsymbol{c}} P(\phi_c) \tag{5}$$

$$\quad = \quad \alpha^D \, \frac{\prod_{c \in \boldsymbol{c}} (n_c - 1)!}{\prod_{k=1}^{n} (\alpha + k - 1)} \prod_{c \in \boldsymbol{c}} f(\phi_c) \tag{6}$$

where D is the number of distinct mixture components, n_c is the count of items belonging to mixture component $c \in \boldsymbol{c}$, and $f(\phi_c)$ is the prior probability density function for ϕ_c for mixture component $c \in \boldsymbol{c}$.

For the split proposal, the appropriate ratio of prior distributions is:

$$\frac{P(\boldsymbol{\gamma}^{split})}{P(\boldsymbol{\gamma})} \quad = \quad \alpha \, \frac{(n^{split}_{c_i^{split}} - 1)! \, (n^{split}_{c_j^{split}} - 1)! \, f(\phi^{split}_{c_i^{split}}) \, f(\phi^{split}_{c_j^{split}})}{(n_{c_i} - 1)! \, f(\phi_{c_i})} \tag{7}$$

where $\boldsymbol{\gamma}$ is the original state in which i and j belong to the same mixture component, $n^{split}_{c_i^{split}}$ and $n^{split}_{c_j^{split}}$ are the number of observations associated with each split component. The ratio of the prior distributions simplifies because the denominator in equation (6) and factors not associated with components that are directly involved in the Metropolis-Hastings update cancel.

For the merge proposal, the prior ratio simplifies to:

$$\frac{P(\boldsymbol{\gamma}^{merge})}{P(\boldsymbol{\gamma})} \quad = \quad \frac{1}{\alpha} \, \frac{(n^{merge}_{c_i^{merge}} - 1)! \, f(\phi^{merge}_{c_i^{merge}})}{(n_{c_i} - 1)! \, (n_{c_j} - 1)! \, f(\phi_{c_i}) \, f(\phi_{c_j})} \tag{8}$$

where $\boldsymbol{\gamma}$ represents the original state in which items i and j belong to separate components.

The likelihood, $L(\boldsymbol{\gamma}|\boldsymbol{y})$, will be a product over n observations:

$$L(\boldsymbol{\gamma}|\boldsymbol{y}) \quad = \quad \prod_{k=1}^{n} F(y_k; \phi_{c_k}) \tag{9}$$

$L(\boldsymbol{\gamma}|\boldsymbol{y})$ can be expressed as a double product over components, c, and items, $k \in \{1,\ldots,n\}$, associated with each component:

$$L(\boldsymbol{\gamma}|\boldsymbol{y}) \;=\; \prod_{c=1}^{D} \prod_{k\,:\,c_k=c} F(y_k;\phi_c) \tag{10}$$

where D is the number of distinct components. This expression to calculate the likelihood is often easier to use in real examples.

Likelihood factors involving items associated with components not directly involved in the split proposal cancel. The ratio of likelihoods in equation (4) reduces to the following:

$$\frac{L(\boldsymbol{\gamma}^{split}|\boldsymbol{y})}{L(\boldsymbol{\gamma}|\boldsymbol{y})} = \frac{\displaystyle\prod_{k\,:\,c_k^{split}=c_i^{split}} F(y_k;\phi^{split}_{c_i^{split}}) \prod_{k\,:\,c_k^{split}=c_j^{split}} F(y_k;\phi^{split}_{c_j^{split}})}{\displaystyle\prod_{k\,:\,c_k=c_i} F(y_k;\phi_{c_i})} \tag{11}$$

Likewise, for the merge proposal, the ratio of likelihoods is:

$$\frac{L(\boldsymbol{\gamma}^{merge}|\boldsymbol{y})}{L(\boldsymbol{\gamma}|\boldsymbol{y})} = \frac{\displaystyle\prod_{k\,:\,c_k^{merge}=c_i^{merge}} F(y_k;\phi^{merge}_{c_i^{merge}})}{\displaystyle\prod_{k\,:\,c_k=c_i} F(y_k;\phi_{c_i}) \prod_{k\,:\,c_k=c_j} F(y_k;\phi_{c_j})} \tag{12}$$

The Metropolis-Hastings proposal density, $q(\boldsymbol{\gamma}^*|\boldsymbol{\gamma})$, is the restricted Gibbs sampling transition kernel from launch state $\boldsymbol{\gamma}^L$ to final state $\boldsymbol{\gamma}^*$. This is a product of the conditional probabilities of each individual update of the vector $\boldsymbol{c}^*$ from $\boldsymbol{c}^L$ and the conditional densities of assigning successive components of $\boldsymbol{\phi}^L$ to their final values, $\boldsymbol{\phi}^*$.

Typically, for each mixture component, ϕ is composed of more than one model parameter, i.e. each ϕ_c can be a vector of parameters. For example, in the normal model, there are two parameters per component, $\phi_c = (\mu_c, \sigma_c^2)$. In a Gibbs sampling scan, each element of parameter ϕ_c is updated individually, while holding the other elements of ϕ_c fixed. A single element of ϕ_c is updated in a restricted Gibbs sampling scan by drawing a new value from its full conditional distribution.

We will denote the product of conditional probabilities obtained from **one full scan** of restricted Gibbs sampling as P_{GS}. Since $\boldsymbol{\gamma}$ is comprised of both $\boldsymbol{c}$ and $\boldsymbol{\phi}$, for clarity, we can split the Gibbs sampling transition kernel into its factors. The order of updating the variables does not affect the validity of the method, but for presentation purposes, we assume that Gibbs sampling updates $\boldsymbol{\phi}$ first (as is done in the later examples):

$$q(\boldsymbol{\gamma}^*|\boldsymbol{\gamma}) \;=\; P_{GS}(\boldsymbol{\phi}^*\,|\,\boldsymbol{\phi}^L,\,\boldsymbol{c}^L,\,\boldsymbol{y})\cdot P_{GS}(\boldsymbol{c}^*\,|\,\boldsymbol{c}^L,\boldsymbol{\phi}^*,\,\boldsymbol{y}) \tag{13}$$

An individual update of a particular c_k is as follows:

$$P(c_k\,|\,c_{-k},\,\phi_{c_k},\,y_k) = \frac{n_{-k,c_k}\,F(y_k;\phi_{c_k})}{n_{-k,c_i}\,F(y_k;\phi_{c_i}) \,+\, n_{-k,c_j}\,F(y_k;\phi_{c_j})} \tag{14}$$

where c_{-k} represents the c_l for $l \neq k$ in $S \cup \{i, j\}$, $n_{-k,c}$ is the number of c_l for $l \neq k$ in $S \cup \{i, j\}$ that are equal to c, and $F(y_k; \phi_c)$ is the likelihood. Here, c_k is restricted to being either c_i or c_j. Each time a c_k or ϕ_{c_k} is incrementally modified during a restricted Gibbs sampling scan, it is immediately used in the subsequent Gibbs sampling computation.

The required ratios for the split and merge proposals are shown below in equations (15) and (16), respectively. For the merge proposal, there is still only one way to combine items in two components into one component, so $P_{GS}(\boldsymbol{c}|\boldsymbol{c}^{L_{merge}}, \boldsymbol{\phi}, \boldsymbol{y}) = 1$ in equation (15). The same is true for $P(\boldsymbol{c}^{merge}|\boldsymbol{c}^{L_{merge}}, \boldsymbol{\phi}^{merge}, \boldsymbol{y})$ in equation (16). However, since specific parameters now define the mixture components, there are numerous possibilities for choosing a particular mixture component. We address this, in a similar method as the split scenario, by conducting intermediate Gibbs sampling scans to decide the value of the merged component's parameters. One final Gibbs sampling scan is conducted from the launch state to calculate the Gibbs sampling transition kernel.

The ratio of transition densities for the split proposal is:

$$\begin{aligned}
&\frac{q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^{split})}{q(\boldsymbol{\gamma}^{split}|\boldsymbol{\gamma})} \\
&\quad = \frac{P_{GS}(\phi_{c_i}|\phi_{c_i}^{L_{merge}}, \boldsymbol{c}^{L_{merge}}, \boldsymbol{y})\, P_{GS}(\boldsymbol{c}|\boldsymbol{c}^{L_{merge}}, \boldsymbol{\phi}, \boldsymbol{y})}{P_{GS}(\phi^{split}_{c_i^{split}}|\phi^{L_{split}}_{c_i^{split}}, \boldsymbol{c}^{L_{split}}, \boldsymbol{y})\, P_{GS}(\phi^{split}_{c_j^{split}}|\, \phi^{L_{split}}_{c_j^{split}}, \boldsymbol{c}^{L_{split}}, \boldsymbol{y})\, P_{GS}(\boldsymbol{c}^{split}|\boldsymbol{c}^{L_{split}}, \boldsymbol{\phi}^{split}, \boldsymbol{y})} \\
&\quad = \frac{P_{GS}(\phi_{c_i}|\phi_{c_i}^{L_{merge}}, \boldsymbol{c}^{L_{merge}}, \boldsymbol{y})}{P_{GS}(\phi^{split}_{c_i^{split}}|\phi^{L_{split}}_{c_i^{split}}, \boldsymbol{c}^{L_{split}}, \boldsymbol{y})\, P_{GS}(\phi^{split}_{c_j^{split}}|\phi^{L_{split}}_{c_j^{split}}, \boldsymbol{c}^{L_{split}}, \boldsymbol{y})\, P_{GS}(\boldsymbol{c}^{split}|\boldsymbol{c}^{L_{split}}, \boldsymbol{\phi}^{split}, \boldsymbol{y})} \qquad (15)
\end{aligned}$$

To calculate $q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^{split})$, the same intermediate Gibbs sampling operations that are performed when proposing a merge must be conducted here to arrive at a suitable merge launch state, even though no actual merge is performed. The Gibbs sampling transition probability is calculated from the launch state (which is the last intermediate Gibbs sampling state) to the original merged state. These operations are necessary to produce the correct proposal ratios.

For the merge proposal, the ratio of transition densities is:

$$\begin{aligned}
\frac{q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^{merge})}{q(\boldsymbol{\gamma}^{merge}|\boldsymbol{\gamma})} &= \frac{P_{GS}(\phi_{c_i}|\phi_{c_i}^{L_{split}}, \boldsymbol{c}^{L_{split}}, \boldsymbol{y})\, P_{GS}(\phi_{c_j}|\phi_{c_j}^{L_{split}}, \boldsymbol{c}^{L_{split}}, \boldsymbol{y})\, P_{GS}(\boldsymbol{c}|\boldsymbol{c}^{L_{split}}, \boldsymbol{\phi}, \boldsymbol{y})}{P_{GS}(\phi^{merge}_{c_i^{merge}}|\phi^{L_{merge}}_{c_i^{merge}}, \boldsymbol{c}^{L_{merge}}, \boldsymbol{y})\, P_{GS}(\boldsymbol{c}^{merge}|\boldsymbol{c}^{L_{merge}}, \boldsymbol{\phi}^{merge}, \boldsymbol{y})} \\
&= \frac{P_{GS}(\phi_{c_i}|\phi_{c_i}^{L_{split}}, \boldsymbol{c}^{L_{split}}, \boldsymbol{y})\, P_{GS}(\phi_{c_j}|\phi_{c_j}^{L_{split}}, \boldsymbol{c}^{L_{split}}, \boldsymbol{y})\, P_{GS}(\boldsymbol{c}|\boldsymbol{c}^{L_{split}}, \boldsymbol{\phi}, \boldsymbol{y})}{P_{GS}(\phi^{merge}_{c_i^{merge}}|\phi^{L_{merge}}_{c_i^{merge}}, \boldsymbol{c}^{L_{merge}}, \boldsymbol{y})} \qquad (16)
\end{aligned}$$

To obtain $q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^{merge})$, we similarly perform the same intermediate Gibbs sampling moves when proposing a split, even though no actual split is proposed (since it is already known). This time the Gibbs sampling transition probability is calculated from the launch state to the original split state. This ensures correct proposal ratios.

The number of intermediate Gibbs sampling scans used to arrive at suitable launch states for both split and merge proposals are tuning parameters of this algorithm. There is an additional tuning parameter for the nonconjugate split-merge procedure that is not present in the conjugate version, which did not require a merge launch state.

4.4 Validity of the algorithm

The nonconjugate split-merge procedure described here is justified as a valid two-stage random Metropolis-Hastings procedure. In the first stage, we randomly select of observations i and j to decide which subset of Metropolis-Hastings proposals will be considered. In the second stage, we randomly select a launch state from among all possible launch states (given the selection of observations i and j), by means of intermediate Gibbs sampling scans. We then perform a standard Metropolis-Hastings update with a proposal distribution that depends on the selection of i and j and on the launch state. As discussed by Tierney (1994), a random selection among transitions (in this case, via random selection of a proposal distribution) is a valid way of constructing Markov chain Monte Carlo algorithms, as long as all the transitions that might be selected are valid on their own.

A subtle clarification should be pointed out regarding the construction of the Metropolis-Hastings acceptance probability for the nonconjugate procedure. When a split is proposed from a merged state, only one ϕ_c is included in the equations, since the merged component has only one set of parameters associated with it now. We happen to initially pick ϕ_{c_j} to be associated with the observations in the merged component, but this is equivalent to initially selecting ϕ_{c_i} since the labels are irrelevant. To avoid changing dimensions when we compute the Metropolis-Hastings acceptance probability, we could include the appropriate ϕ_{c_i} terms in the computations. Since ϕ_{c_i} is an extra parameter for the merged component that is no longer associated with the data, we choose to propose a new value for it during the restricted Gibbs sampling scan by drawing from its prior distribution. This choice conveniently allows the prior density for this term to implicitly cancel with the corresponding term in the proposal density of the acceptance probability, showing that the change in dimensionality is not a problem. Consider the following set-up for the prior and proposal ratios for a split proposal which include the ϕ_{c_i} terms. We intentionally omit the likelihoods and indicator terms for simplicity and space considerations:

$$\frac{P(\phi^{split}_{c_i^{split}})\, P(\phi^{split}_{c_j^{split}})}{P(\phi_{c_i})\, P(\phi_{c_j})} \; \frac{P_{GS}(\phi_{c_i}|\phi_{c_i}^{L_{merge}},\, \boldsymbol{c}^{L_{merge}})\, P_{GS}(\phi_{c_j}|\phi_{c_j}^{L_{merge}},\, \boldsymbol{c}^{L_{merge}},\, \boldsymbol{y})}{P_{GS}(\phi^{split}_{c_i^{split}}|\phi^{L_{split}}_{c_i^{split}},\, \boldsymbol{c}^{L_{split}},\, \boldsymbol{y}) P_{GS}(\phi^{split}_{c_j^{split}}|\phi^{L_{split}}_{c_j^{split}},\, \boldsymbol{c}^{L_{split}},\, \boldsymbol{y})}$$

The proposal factor, $P_{GS}(\phi_{c_i}|\phi_{c_i}^{L_{merge}},\, \boldsymbol{c}^{L_{merge}})$ does not depend on the data, since the ϕ_{c_j} factor has been selected earlier to be the merged component's parameter. Therefore, a new draw from ϕ_{c_i}'s conditional distribution will be equivalent to drawing a new value from its prior distribution, and this will cancel with the prior term, $P(\phi_{c_i})$. As a result, the ratios described earlier do not need to include these terms. The identical situation occurs in the case when a merge is proposed from an original split state and is handled similarly.

Note that it is possible to propose any configuration of observations from any initial state via a sequence of split and then merge proposals. However, to ensure ϕ-irreducibility on a continuous state space, it must be possible to propose any set of parameter values for each component. This will be true if each individual restricted Gibbs sampling conditional distribution for parameters of components that are involved in a particular split or merge update has a positive probability density of proposing any value.

4.5 Improving efficiency of the algorithm

Methods to improve efficiency and performance are presented in this section. Adding a final Gibbs sampling scan to the split-merge procedure is discussed in Section 4.5.1. In Section 4.5.2, a swap update is proposed to correct a labeling problem.

4.5.1 Cycling Metropolis-Hastings updates with Gibbs sampling

The nonconjugate method split-merge procedure is best used for making major, sweeping changes to the configuration of items in a single iteration. These split-merge moves dramatically reduce the length of burn-in time. However, for the minor shuffling of a single observation between components, incorporating an incremental procedure after a split-merge operation is worthwhile.

We have tried both incremental updates based on Gibbs sampling with auxiliary variables (Neal 2000, Algorithm 8) and incremental Metropolis-Hastings updates (Neal 2000, Algorithm 5), and found that both are effective in supplementing the split-merge updates. In the remainder of this paper, we use only the auxiliary variable method. The number of final incremental scans in each full iteration is a tuning parameter for the algorithm.

4.5.2 The swap proposal

There is is a potential labeling problem stemming from the split-merge procedure's random selection and treatment of the observations, i and j. This problem is for the conjugate models by Jain and Neal (2004), is also present in this version of the algorithm. The initial random split of the other observations in a merged component could assign labels biased towards a split that is opposite to the fixed labels of i and j. Therefore, i and j end up in the "wrong" mixture components.

This nuisance labeling problem can be remedied by proposing a direct swap of the labels of the items associated with each split component (equivalent to switching the component indicators of i and j) prior to reaching the launch state. A simple Metropolis update that evaluates a swap proposal is recommended immediately after conducting the intermediate restricted Gibbs sampling scans. In this case, the swap proposal includes a direct switch of the parameters of the two components associated with items i and j (as well as swapping the labels of the items associated with each split component). This swap proposal should result in i and j being in the "correct" mixture component and should improve the overall Metropolis-Hastings acceptance rate by up to a factor of 2. After performing the swap proposal, it is not necessary to recalculate the proposal prior and likelihood for the split-merge Metropolis-Hastings acceptance probability, since this was already computed during the swap proposal's Metropolis-Hastings update.

5 Performance of the nonconjugate split-merge procedure

In this section, a simulation study is conducted to examine the performance of the nonconjugate split-merge procedure. Our model of choice is the Normal mixture, in which the data, $\boldsymbol{y} = (y_1, \ldots, y_n)$, are independent and identically distributed, such that each observation, y_i, given the class, c_i, has m Normally distributed attributes, $(y_{i1}, \ldots, y_{im})$. An observation's attributes are independent given the class, c_i. The Normal mixture model is commonly used in Bayesian mixture analysis because of its simplicity in constructing conditional distributions and flexibility in modeling a number of heterogeneous populations simultaneously.

Section 5.1 presents a version of the Normal mixture model under consideration. Three data sets, discussed in Section 5.2, are used to empirically compare the split-merge technique to two incremental samplers in Section 5.3. Section 5.4 examines the performance of the nonconjugate technique when each of the four tuning parameters is varied while holding the others fixed.

5.1 The mixture model with Normal-Gamma prior

We model data from a mixture of Normal distributions using a Dirichlet process mixture model with Normal-Gamma prior, as follows:

$$\begin{aligned} y_i \mid \mu_i,\, \tau_i &\sim F(y_i;\, \mu_i, \tau_i) \;=\; N(y_i;\, \mu_i, \tau_i^{-1}\,\boldsymbol{I}) \\ (\mu_i,\, \tau_i) \mid G &\sim G \\ G &\sim DP(G_0, \alpha) \\ G_0(\mu, \tau) &= N(\mu;\, w, B^{-1}) \,\cdot\, \text{Gamma}\,(\tau;\, r, R) \end{aligned} \tag{17}$$

where τ, the *precision* parameter, is σ^{-2}. Hyperpriors could be placed on w, B, r, and R to add another stage to this hierarchy if desired. Here, we consider these parameters to be known.

The probability density function for the prior distribution of μ given in (17) is:

$$f(\mu \,|\, w, B) = \left(\frac{B}{2\pi}\right)^{\frac{1}{2}} \exp\left(\frac{-B}{2}(\mu - w)^2\right) \tag{18}$$

where B is a precision parameter.

The probability density function for the prior for τ is:

$$f(\tau \,|\, r, R) = \frac{1}{R^r\,\Gamma(r)}\, \tau^{r-1} \exp\left(\frac{-\tau}{R}\right) \tag{19}$$

This parameterization of the Gamma density is adopted throughout this section.

These priors, equations (18) and (19), are necessary to compute the priors for the parameters in the Metropolis-Hastings acceptance probability of equation (4).

It is straightforward to set up the conditional distributions required for the restricted Gibbs sampling in the split-merge procedure used in the Metropolis-Hastings proposal densities. For the model parameters, this amounts to sampling from the marginal posterior distributions for a particular parameter of component c. The conditional posterior distribution for μ_{ch} (when τ_{ch} is known) for a specific attribute h is:

$$\mu_{ch} \,|\, \boldsymbol{c}, \boldsymbol{y}, \tau_{ch}, w, B \;\;\sim\;\; N\left(\frac{w\,B + \bar{y}_{ch}\, n_c\, \tau_{ch}}{B + n_c\, \tau_{ch}},\; \frac{1}{B + n_c\, \tau_{ch}}\right) \tag{20}$$

where n_c is the number of observations belonging to component c and $\bar{y}_{ch}$ of attribute h for these these observations.

Similarly, if μ_{ch} is fixed, the conditional posterior distribution for τ_{ch} for a particular attribute h is:

$$\tau_{ch} \,|\, \boldsymbol{c}, \boldsymbol{y}, \mu_{ch}, r, R \quad \sim \quad \text{Gamma}\left(r + \frac{n_c}{2},\; \frac{1}{R^{-1} + \frac{1}{2} \sum\limits_{k:c_k=c} (y_{kh} - \mu_{ch})^2} \right) \tag{21}$$

The conditional posterior distribution for an indicator variable, c_i, is obtained by combining the probability of the data (given in equation 17) given a value for c_i with the prior for indicators, $P(\boldsymbol{c})$. This yields for $c \in \{c_j\}_{j \neq i}$:

$$\begin{aligned} P(c_i = c \mid c_{-i},\, \mu_c, \tau_c, y_i) \quad &\propto \quad P(c_i = c \mid c_{-i}) \cdot P(y_i \mid \mu_c, \tau_c, c_{-i}) \qquad (22) \\ &\propto \quad n_{-i,c} \prod_{h=1}^{m} \tau_{ch}^{\frac{1}{2}} \exp\left(\frac{-\tau_{ch}}{2} \, (y_{ih} - \mu_{ch})^2 \right) \end{aligned}$$

These conditional distributions are also employed in computations required for Gibbs sampling with auxiliary parameters and incremental Metropolis-Hastings updates that will be used as comparisons to the nonconjugate split-merge technique later in this article.

The likelihood used in computing acceptance probabilities for split-merge updates is much simpler to obtain than in the conjugate case, since the parameters are not integrated away. For the mixture of Normals, the likelihood (given component indicators) is

$$L(\boldsymbol{\gamma}|\boldsymbol{y}) \quad = \quad \prod_{c=1}^{D} \prod_{k\,:\,c_k=c} \prod_{h=1}^{m} \left(\frac{\tau_{ch}}{2\pi} \right)^{\frac{1}{2}} \exp\left(\frac{-\tau_{ch}}{2} \, (y_{kh} - \mu_{ch})^2 \right) \tag{23}$$

Interchanging the products over k and h of equation (23) yields the following:

$$L(\boldsymbol{\gamma}|\boldsymbol{y}) \quad = \quad \prod_{c=1}^{D} \prod_{h=1}^{m} \left(\frac{\tau_{ch}}{2\pi} \right)^{\frac{n_c}{2}} \exp\left(\frac{-\tau_{ch}}{2} \sum_{k:c_k=c} (y_{kh} - \mu_{ch})^2 \right) \tag{24}$$

Efficiency can be improved by incrementally updating the sufficient statistics for the model. For this particular model, we maintained counts for the sum of the observations and the sum of the squares of the observations for each component, and used these counts when computing the likelihood and doing Gibbs sampling. The savings in these operations more than offsets the cost of incrementing or decrementing these counts when the component indicators change.

5.2 The synthetic data

The purpose of this study is to classify observations into appropriate latent classes using the Normal-Gamma Dirichlet process mixture model. We can make this problem computationally more difficult by increasing the dimensionality of the data and by moving the components closer together. Various combinations of these factors were tested on all procedures. We found that the split-merge procedures outperformed the incremental procedures even in very low-dimensional problems, in which distinct components were visible by eye, showing the difficulty that incremental samplers have in reaching equilibrium even in simple problems when the components are similar.

We will consider three simulated data sets with a finite number of components. We expect that the Dirichlet process mixture model will model the finite situation perfectly well without problems such as overfitting, even though the model allows an infinite number of components. For each of the three examples, the data are composed of five equally-probable mixture components, in which each component is a distribution over m dimensions. To maintain uniformity amongst the examples, we generated $n = 100$ observations, stratified so that 20 observations came from each of the five mixture components.

Data for the three examples were randomly generated from the mixture distributions shown in Tables 1–3. Scatterplots of the data are shown in Figures 1 - 3. A standard deviation of 0.2 was selected for all Normal distributions, so that only the means would vary. The first two examples differ in that one of the components is moved closer to two of the other components in Example 2, while holding the dimensionality at two. The third example differs from the others in that the dimensionality is increased to three, and the components are closer together. Intentional asymmetry is introduced so that three components are more similar than the other two. This is intended to test whether the nonconjugate split-merge techniques can split in three ways.

The Dirichlet process parameter, α, is set to one for all demonstrations. Recall that a small value of α places stronger belief that the number of mixture components in the data is likely to be small. The parameters of the priors for the parameters on the component distributions have been set to the same values over all dimensions as follows: $w = 5$, $B = 1/12$, $r = 1$, and $R = 5$. Here, B is a precision parameter. For consistency, these parameters are fixed at these values for all simulations. In actual problems, these parameters could be set either by prior knowledge or given higher-level priors.

To verify the correctness of the implementations of these procedures, a small, two-dimensional example was constructed that allowed us to theoretically compare the actual (true) posterior quantities to the simulated results. Results obtained on the three data sets using the various algorithms were also compared.

5.3 Performance

For each of the three examples, two incremental procedures, Gibbs sampling with $v = 3$ auxiliary variables, and an incremental Metropolis-Hastings method, are compared to four versions of the nonconjugate split-merge procedure. We use four numbers to describe the various split-merge procedures.

1. Number of intermediate Gibbs sampling scans to reach the launch state for a split proposal

2. Number of split-merge updates done in a single overall iteration

3. Number of complete incremental Gibbs sampling scans after the final split-merge update

4. Number of intermediate Gibbs sampling scans to reach the launch state for a merge proposal

Table 1: True mixture distribution for Example 1.

c	$P(c_i = c)$	$P(y_{ih} \mid c_i = c), h = 1, 2$	
1	0.2	N(2.0, 0.04)	N(3.0, 0.04)
2	0.2	N(3.0, 0.04)	N(2.0, 0.04)
3	0.2	N(3.3, 0.04)	N(3.3, 0.04)
4	0.2	N(8.0, 0.04)	N(9.0, 0.04)
5	0.2	N(9.0, 0.04)	N(8.5, 0.04)

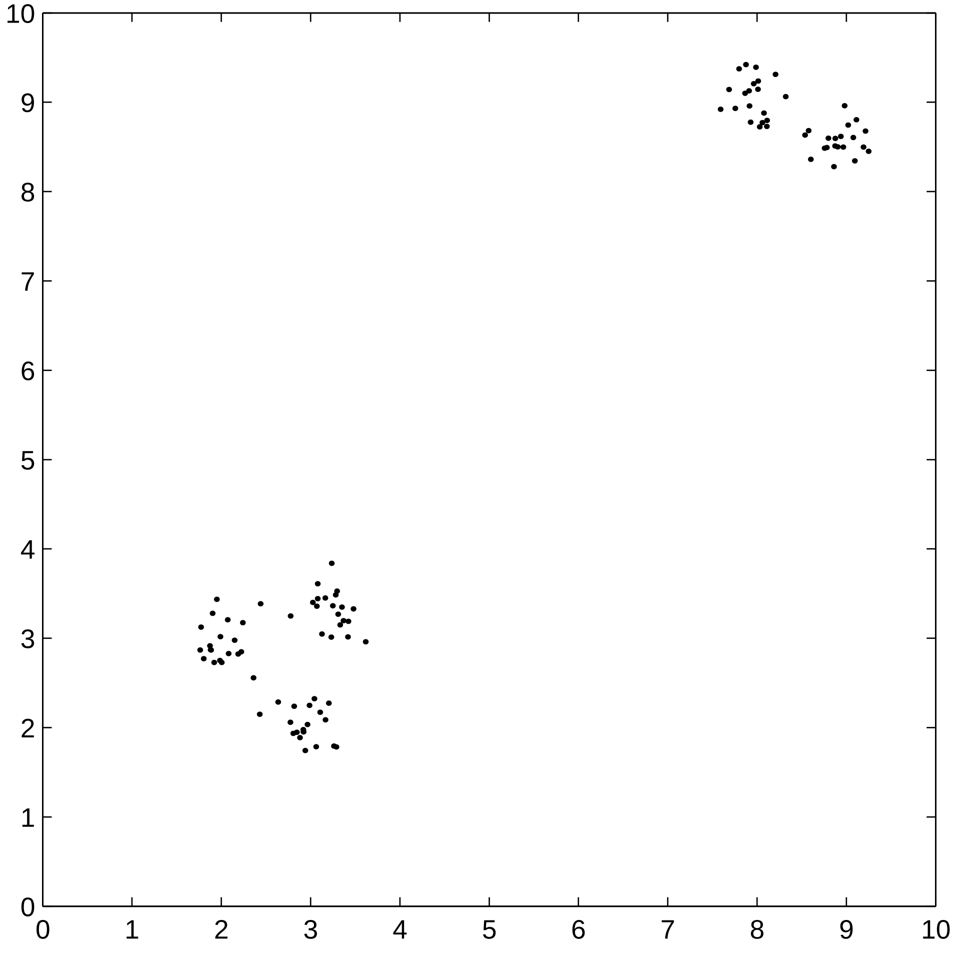


Figure 1: Scatterplot of the data in Example 1

Table 2: True mixture distribution for Example 2.

c	$P(c_i = c)$	$P(y_{ih} \mid c_i = c), h = 1, 2$	
1	0.2	N(2.0, 0.04)	N(3.0, 0.04)
2	0.2	N(3.0, 0.04)	N(2.0, 0.04)
3	0.2	N(3.0, 0.04)	N(3.0, 0.04)
4	0.2	N(8.0, 0.04)	N(9.0, 0.04)
5	0.2	N(9.0, 0.04)	N(8.5, 0.04)

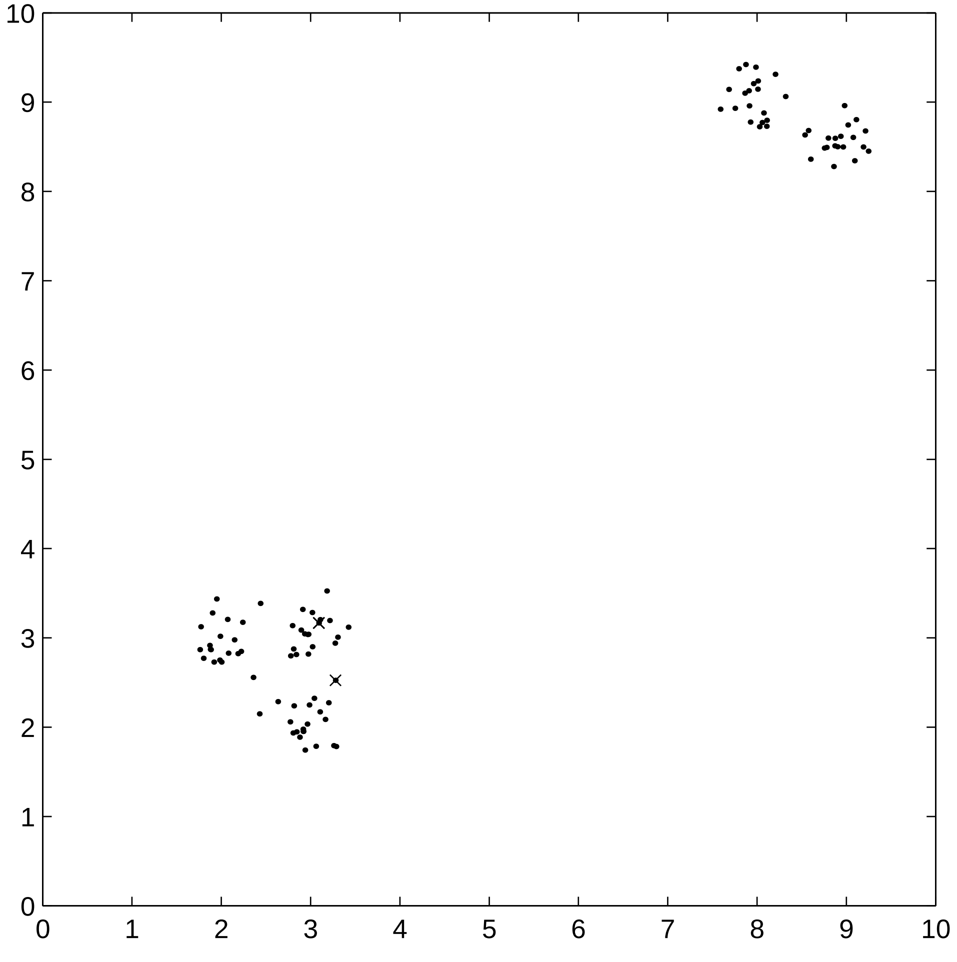


Figure 2: Scatterplot of the data in Example 2. The two x's represent observations 41 and 62 used in autocorrelation calculations for an indicator variable.

Table 3: True mixture distribution for Example 3.

c	$P(c_i = c)$	$P(y_{ih}\|c_i = c), h = 1, 2, 3$		
1	0.2	N(2.0, 0.04)	N(2.0, 0.04)	N(3.0, 0.04)
2	0.2	N(2.0, 0.04)	N(3.0, 0.04)	N(2.0, 0.04)
3	0.2	N(2.0, 0.04)	N(2.5, 0.04)	N(2.5, 0.04)
4	0.2	N(8.0, 0.04)	N(8.0, 0.04)	N(8.0, 0.04)
5	0.2	N(8.0, 0.04)	N(9.0, 0.04)	N(9.0, 0.04)

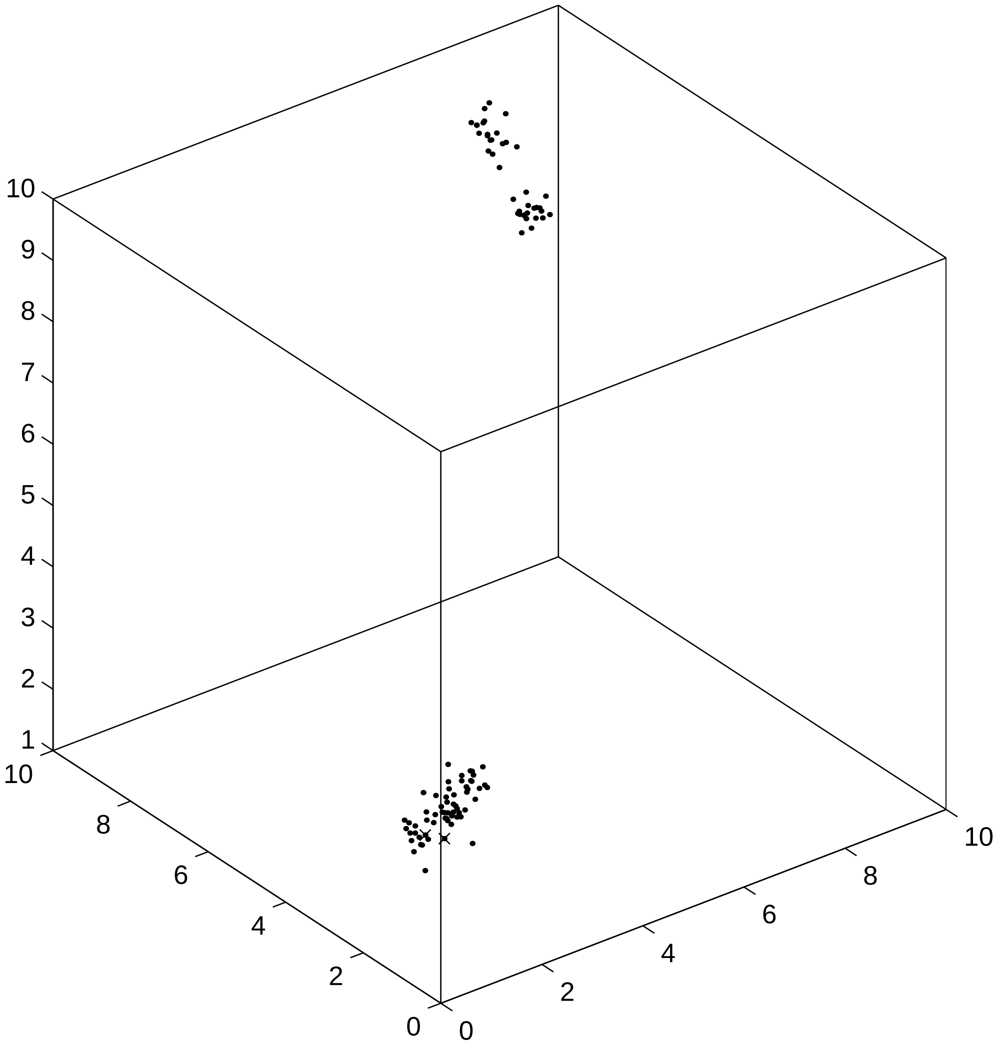


Figure 3: Scatterplot of the data in Example 3. The two **x**'s represent observations 26 and 57 used in autocorrelation calculations for an indicator variable.

Table 4: Time per iteration (in seconds) for the algorithms tested.

Algorithm	*Example 1*	*Example 2*	*Example 3*
Incremental M-H	0.08	0.08	0.09
Gibbs Sampling	0.45	0.42	0.60
Split-Merge (0,1,0,0)	0.05	0.06	0.10
Split-Merge (0,1,1,0)	0.27	0.27	0.35
Split-Merge (5,1,0,5)	0.16	0.20	0.24
Split-Merge (5,1,1,5)	0.40	0.42	0.53

For all split-merge procedures, the swap procedure described in Section 4.5.2 was conducted immediately after the intermediate Gibbs sampling scans for the split proposal to arrive at a launch state. The four split-merge procedures we tested are described using these numbers as Split-Merge (0,1,0,0), Split-Merge (5,1,0,5), Split-Merge (0,1,1,0), and Split-Merge (5,1,1,5).

We compared the split-merge procedures with both the auxiliary variable and Metropolis-Hastings incremental samplers because we did not know beforehand which incremental method would perform better situations where splits and merges might be necessary. Performance of the auxiliary variable Gibbs sampling is expected to improve as we increase the number of auxiliary components, except that it also takes longer per iteration (Neal 2000). We did vary this parameter, but will report findings for $v = 3$ for all examples, since this version is comparable to the best version of split-merge in terms of computation time per iteration. As the incremental final scan for the split-merge procedure, Gibbs sampling with one auxiliary variable is used for all examples.

Performance measures that were considered include trace plots over time (Figures 4–6) and computation time per iteration (Table 4). The trace plots show five values which represent the fractions of observations associated with the most common, two most common, three most common, four most common, and five most common mixture components. Since each of the five components appear equally in the samples, if the true situation were captured exactly, the five traces would occur at values of 0.2, 0.4, 0.6, 0.8, and 1.0.

For each algorithm, all observations were assigned to the same mixture component for the initial state, and each algorithm was run for 5000 iterations. All simulations were performed on Matlab, Version 6.1, on a Dell Precision 530 workstation (which has a 1.7 GHz Pentium 4 processor). Note that the computation times reported include the extra time spent due to Matlab's inefficiencies when copying and incrementally updating arrays, which are not inherent in the algorithm.

5.3.1 Example 1

The three types of procedures, incremental Metropolis-Hastings, incremental Gibbs sampling with auxiliary variables, and split-merge, correctly classify the data in Figure 1 into five distinct clusters. The main difference in performance is the number of burn-in iterations that must be discarded.

The trace plots in Figure 4 show that Gibbs sampling with three auxiliary parameters has fewer burn-in iterations than the incremental Metropolis-Hastings method (compare 1000 to 3200 burn-in iterations). However, since the incremental Metropolis-Hastings method is approximately 5.5 times faster per iteration than the auxiliary Gibbs sampling method, it actually converges sooner

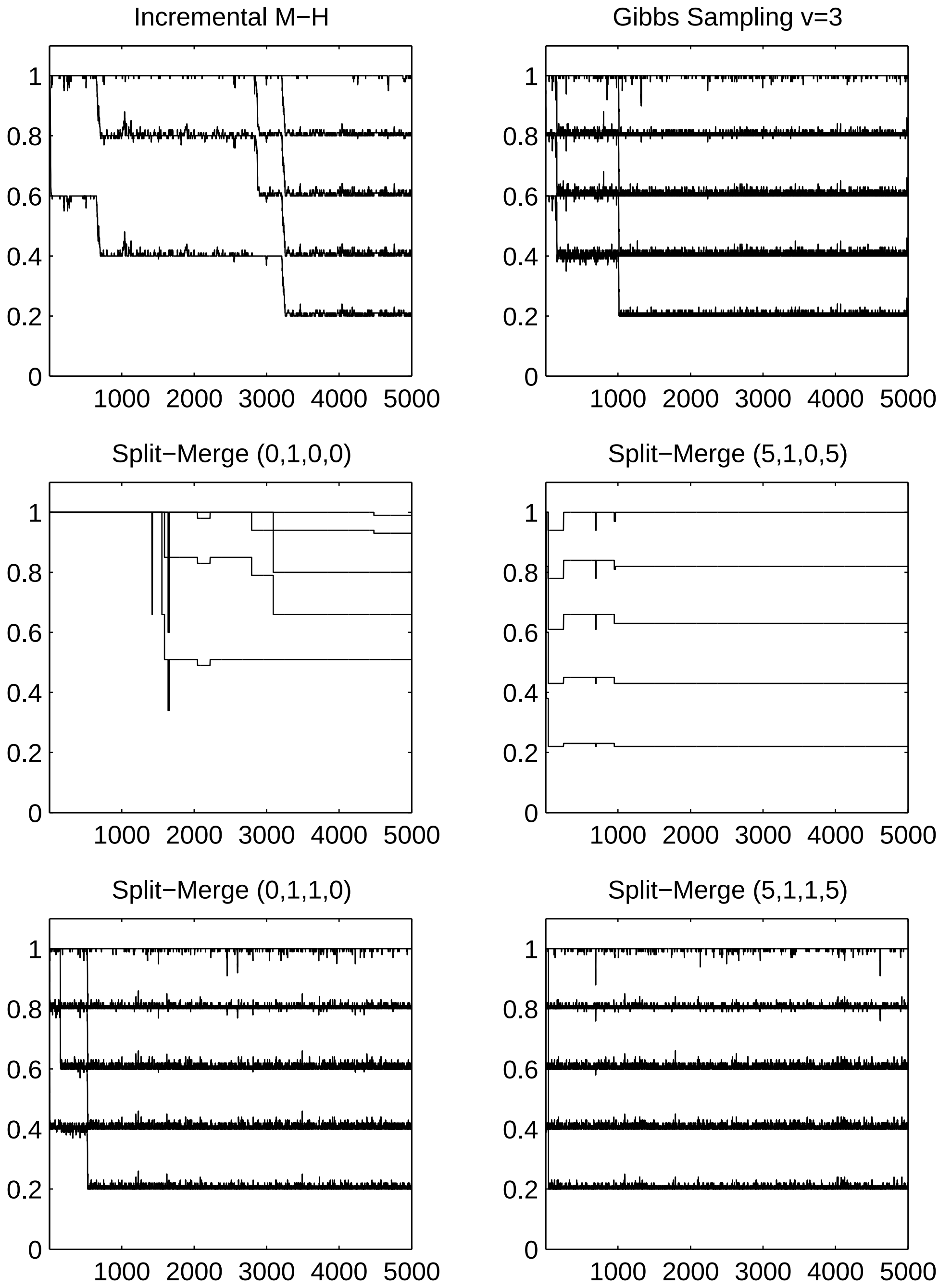


Figure 4: Trace plots of the six algorithms in Example 1.

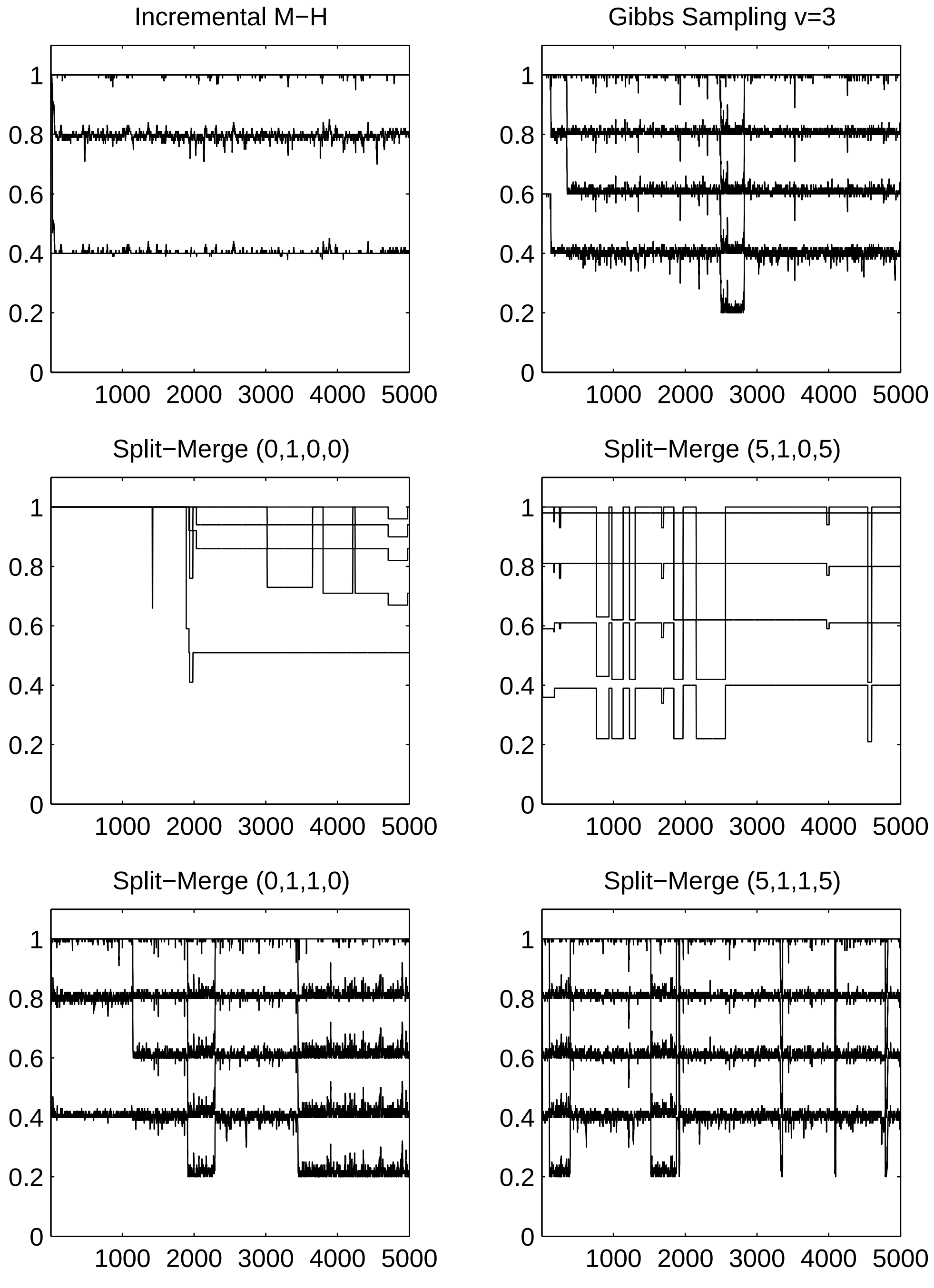


Figure 5: Trace plots of the six algorithms in Example 2.

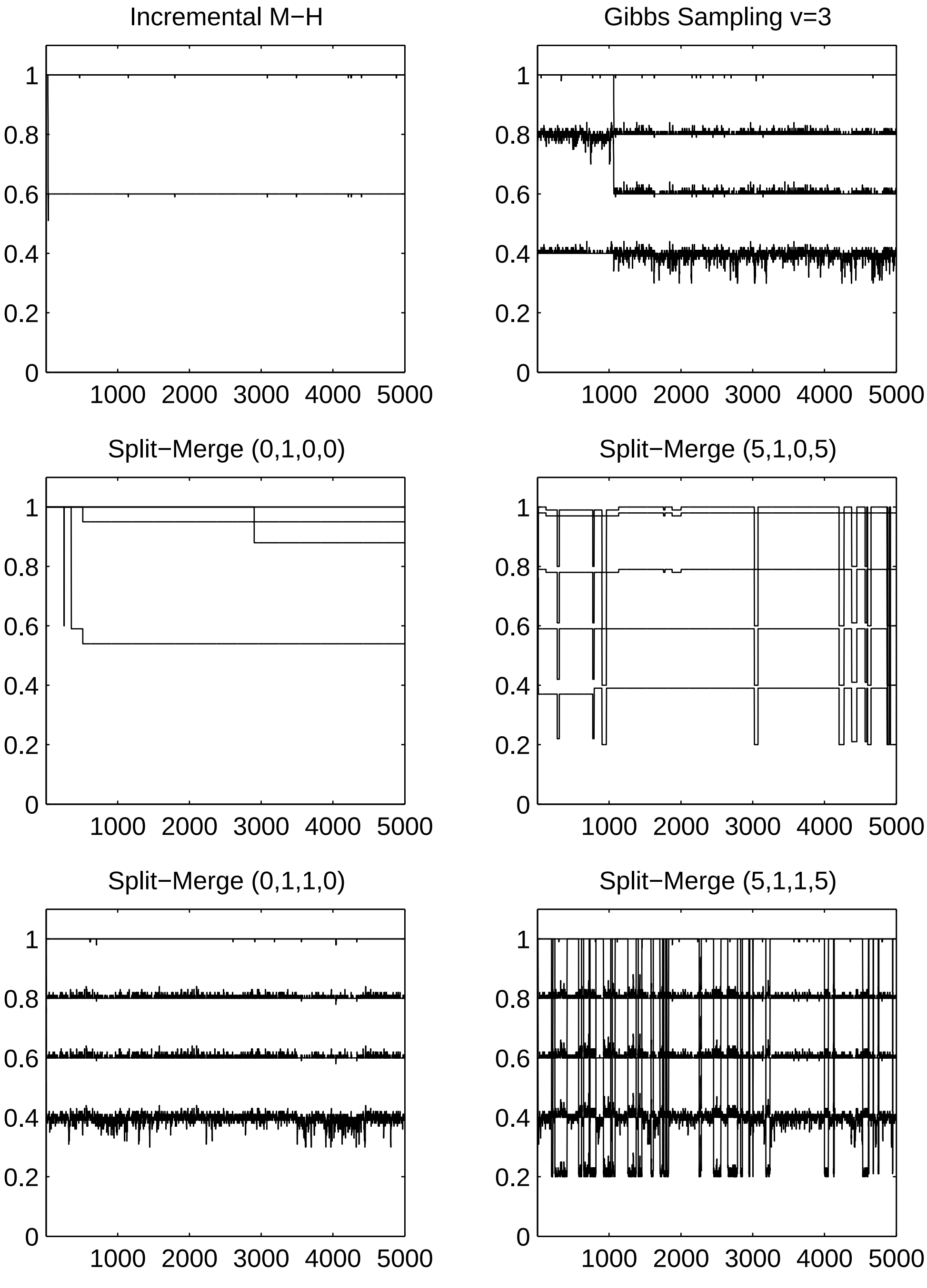


Figure 6: Trace plots of the six algorithms in Example 3.

with respect to computation time. Split-Merge (5,1,0,5) almost immediately splits the data into five components, but notice that the proportions do not occur at exactly 0.2 intervals until after the first thousand iterations. It takes this procedure longer to move a few singleton observations between components, since there is no final incremental update to make these minor adjustments. In five thousand iterations, it is not clear if Split-Merge (5,1,0,5) has actually reached the equilibrium distribution. Split-Merge (0,1,0,0) does not reach the equilibrium distribution in the five thousand iterations shown. Because the split and merge proposals have no intermediate Gibbs sampling scans, the proposals are not expected to be realistic. Split-Merge (0,1,0,0) is essentially a simple random split procedure, except that one restricted Gibbs sampling scan is conducted to reach the final state, which of course will not lead to reasonable split and merge proposals.

However, either by adding intermediate Gibbs sampling scans (as in the case of Split-Merge (5,1,0,5)) or adding a final full incremental scan (as in Split-Merge (0,1,1,0)), the correct proportion of items in each cluster is established. Split-Merge (0,1,1,0) eventually reaches the five component configuration after 500 burn-in iterations. The final procedure of Figure 4, Split-Merge (5,1,1,5), finds the five components immediately, and it appears that there is negligible burn-in (four iterations). The computation time per iteration is higher for Split-Merge (5,1,1,5) versus Split-Merge (0,1,1,0) and (5,1,0,5), but the computation time to equilibrium is much lower.

5.3.2 Example 2

Example 2 holds the dimensionality at two, but from the scatterplot of the data (Figure 2), we see that one of the clusters (cluster 3 from Table 2) has been moved closer to two other clusters and its simulated data values change. The four other clusters' data values have not changed from Example 1. Given the priors, the posterior assigns significant probability to both the four and five component configurations, but the four component configuration dominates. The trace plots in Figure 5 indicate that both incremental samplers are not mixing well between four and five components. The incremental Metropolis-Hastings is stuck in a three component configuration, in which the proportion of observations in each component is 0.4, 0.4, and 0.2. The auxiliary Gibbs sampling is slightly better in that it reaches the dominating four component configuration after 300 iterations, but it takes roughly 2500 iterations to move to five components. The mixing between the four and five component configurations is much slower compared to some of the split-merge methods. This simulation was repeated with different pseudo-random seed numbers, and similar results were obtained.

Split-Merge (5,1,0,5) and Split-Merge (5,1,1,5), on the other hand, mix relatively well between the four and five components, which is due to the intermediate Gibbs sampling proposals. Non-incremental moves allow the transfer of a group of observations to a new component in a single iteration, which is necessary to overcome low probability (single observation per cluster) intermediate states. Split-Merge (0,1,1,0) also reaches the equilibrium distribution, but has long mixing times between the two high-probability configurations due to less appropriate split-merge proposals. Again, Split-Merge (0,1,0,0) performs the worst compared to the other methods and is nowhere near the equilibrium distribution by the end of the specified time interval.

We further examine performance in terms of autocorrelations and the Metropolis-Hastings acceptance probability for the split-merge procedures in Table 5. The autocorrelation time is computed for two quantities: the first trace on the plots (corresponding to the fraction of observations asso-

Table 5: Autocorrelation times and M-H acceptance rates for some algorithms in Example 2.

Algorithm	*Autocorrelation time for Trace 1*	*Autocorrelation time for $I_{41,62}$*	*Acceptance rate in percent*
Gibb Sampling v=3	8834	3405	NA
Split-Merge (0,1,1,0)	2405	206	0.03
Split-Merge (5,1,0,5)	413	348	0.38
Split-Merge (5,1,1,5)	324	204	0.38

ciated with the most common mixture component) and an indicator variable, $I_{41,62}$, which codes if observations 41 and 62 are assigned to the same mixture component. These observations are marked by an x on the scatterplot of the data in Figure 2. One item was clearly generated from the distribution for mixture component 3 in Table 2, and the other observations was actually generated from component 2. However, due to random variation, it appears that the second observation could have easily been generated from component 3. We expect that these two items should have non-zero posterior probability of being assigned to the same mixture component. A low autocorrelation time for this indicator variable implies that the sampler is successful in moving a single observation between components. As before, the autocorrelation time for trace 1 (at 0.2) indicates that the sampler is mixing well between four and five components, so the major allocation moves in a single iteration are successful. The autocorrelation values in Table 5 are based on 20,000 iterations.

Split-Merge (5,1,1,5) has the smallest autocorrelation times of the three methods, while Gibbs sampling has the largest times. It is interesting to compare the autocorrelations of Split-Merge (0,1,1,0) and Split-Merge (5,1,0,5) though. The autocorrelation time of trace 1 for Split-Merge (0,1,1,0) is clearly larger than for Split-Merge (5,1,0,5) (compare 2405 vs. 413), whereas the behaviour of autocorrelation time of $I_{41,62}$ is the opposite for these two procedures. This supports the proposition that an incremental final scan is necessary for small-scale changes, which Split-Merge (5,1,0,5) is unable to do well. However, for major changes to the configuration, proposals based on several intermediate Gibbs sampling scans are required.

5.3.3 Example 3

The most difficult example considered is Example 3 involving three dimensions and mixture components that are close together. A perspective scatterplot of the data is given in Figure 3, and it shows that the components are more difficult to distinguish. Again, given the priors selected, there is significant posterior probability for both the four and five mixture component configurations. Only Split-Merge (5,1,0,5) and Split-Merge (5,1,1,5) mix between these configurations, as observed in Figure 6. The incremental samplers and the split-merge procedures with zero intermediate restricted Gibbs sampling scans do not find the five components over the 5000 iterations, but are stuck in either two or four components. If each item is initially assigned to a different mixture component (plots not included), these samplers do split the data into five components, but take a long time to move to four components, indicating poor mixing. Here, the problem is that the deletion of a component is rare under both incremental updates and poor split-merge proposals.

Comparing further the two procedures that appear to converge, the autocorrelation time for trace 1 is much lower for Split-Merge (5,1,1,5) than Split-Merge (5,1,0,5) (126 vs. 718). For the autocorrelation time of an indicator variable, $I_{26,57}$, coding if observations 26 and 57 are in the same component, the time is again much lower for Split-Merge (5,1,1,5) (38 vs. 417). Even though both algorithms do mix between the two configurations and Split-Merge (5,1,0,5) is faster per iteration, the improvement in autocorrelation time for Split-Merge (5,1,1,5) cannot be ignored. The extra full scan of incremental sampling for minor adjustments is worth the computational effort.

5.3.4 Summary of findings

It appears that split-merge moves are necessary in nonconjugate problems of this sort. Incremental samplers perform adequately when the components are distinct clusters in low dimensions, but as the components become more difficult to distinguish, these samplers take much longer to reach equilibrium. It is important to note that the incremental samplers that we considered begin to break down even in low dimensions. The split-merge procedures are able to handle three-way splits without any problems, although this is done by two two-way splits.

The split-merge procedure with several intermediate Gibbs sampling scans followed by an incremental full scan is the best version of the split-merge procedure. The split-merge method relies on proposing appropriate new clusters, which is accomplished by conducting several intermediate scans to reach the split and merge launch states.

The presence of an additional tuning parameter for the number of intermediate Gibbs sampling scans for a merge proposal does not cause any additional difficulty, in comparison to the conjugate split-merge procedure, for which it is not needed.

The split-merge methods generally have a longer computation time per iteration. However, in the case of the Gibbs sampling procedure with $v = 3$ auxiliary parameters, the best version of the split-merge procedure, Split-Merge (5,1,1,5), is slightly faster in our implementation (see Table 4). Therefore, there does not appear to be any advantage in using only incremental procedures for these types of problems.

In higher dimensions, split-merge procedures continue to work well as the components are moved closer together. Convergence to the equilibrium distribution is relatively quick. We believe that the split-merge procedure may break down for very high dimensional problems, because appropriate splits will be rejected, since it will become unlikely that a merge operation from the split state would produce the same merged parameter values as the current state. However, we have not encountered an example of this. Perhaps this issue arises only in situations where the dimensionality is in the hundreds.

5.4 Tuning parameters

This section investigates the effect of varying the tuning parameters of the nonconjugate split-merge procedure. As discussed at the start of Section 5.3, the split-merge method has four adjustable tuning parameters: the number of intermediate Gibbs sampling scans to reach the split launch state, the number of split-merge updates conducted in a single iteration, and the number of incremental Gibbs sampling scans conducted after the split-merge updates, and the number of intermediate Gibbs sampling scans to reach the merge launch state. The data from Example 3 is used to examine each tuning parameter. Computation time per iteration and autocorrelation times for trace 1 and

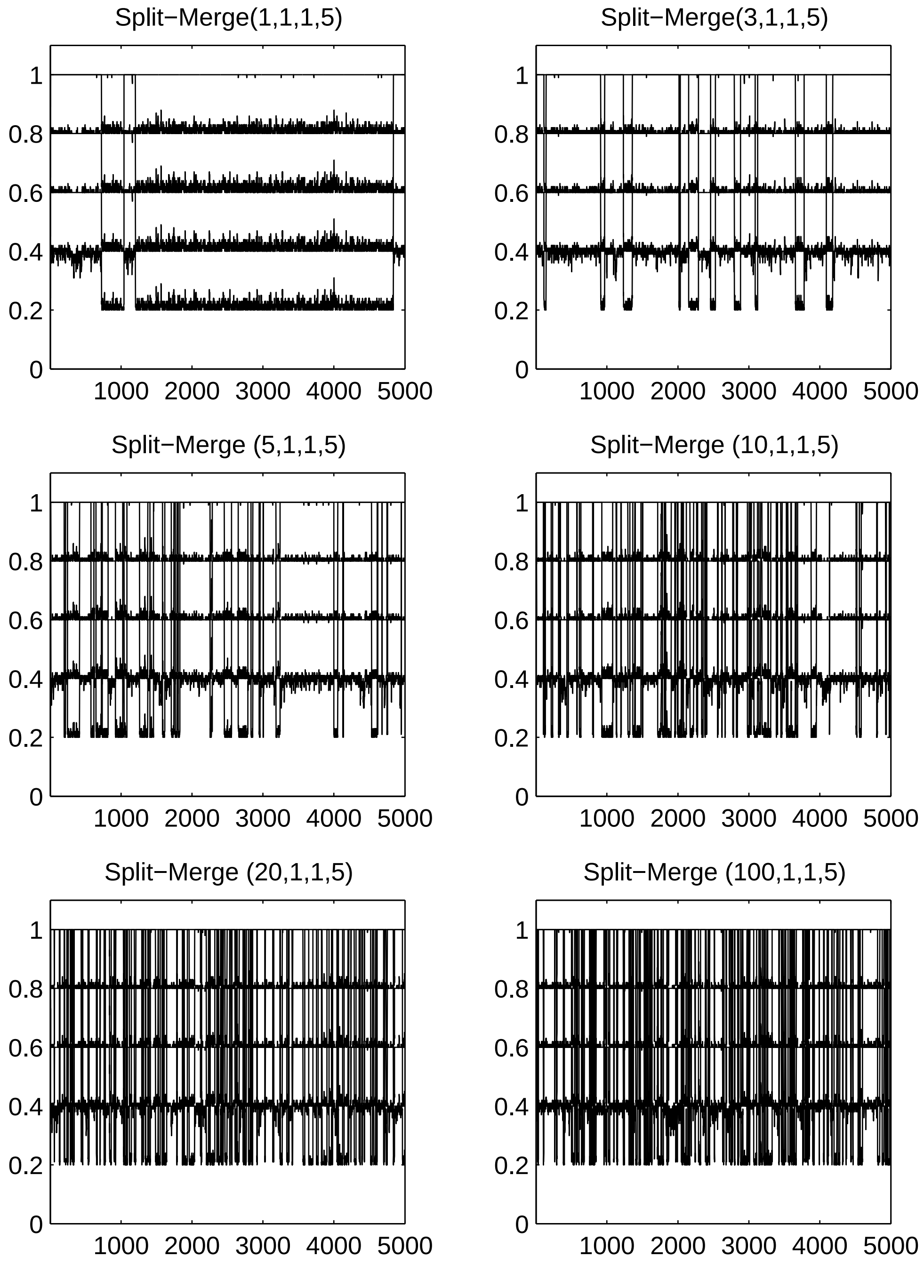
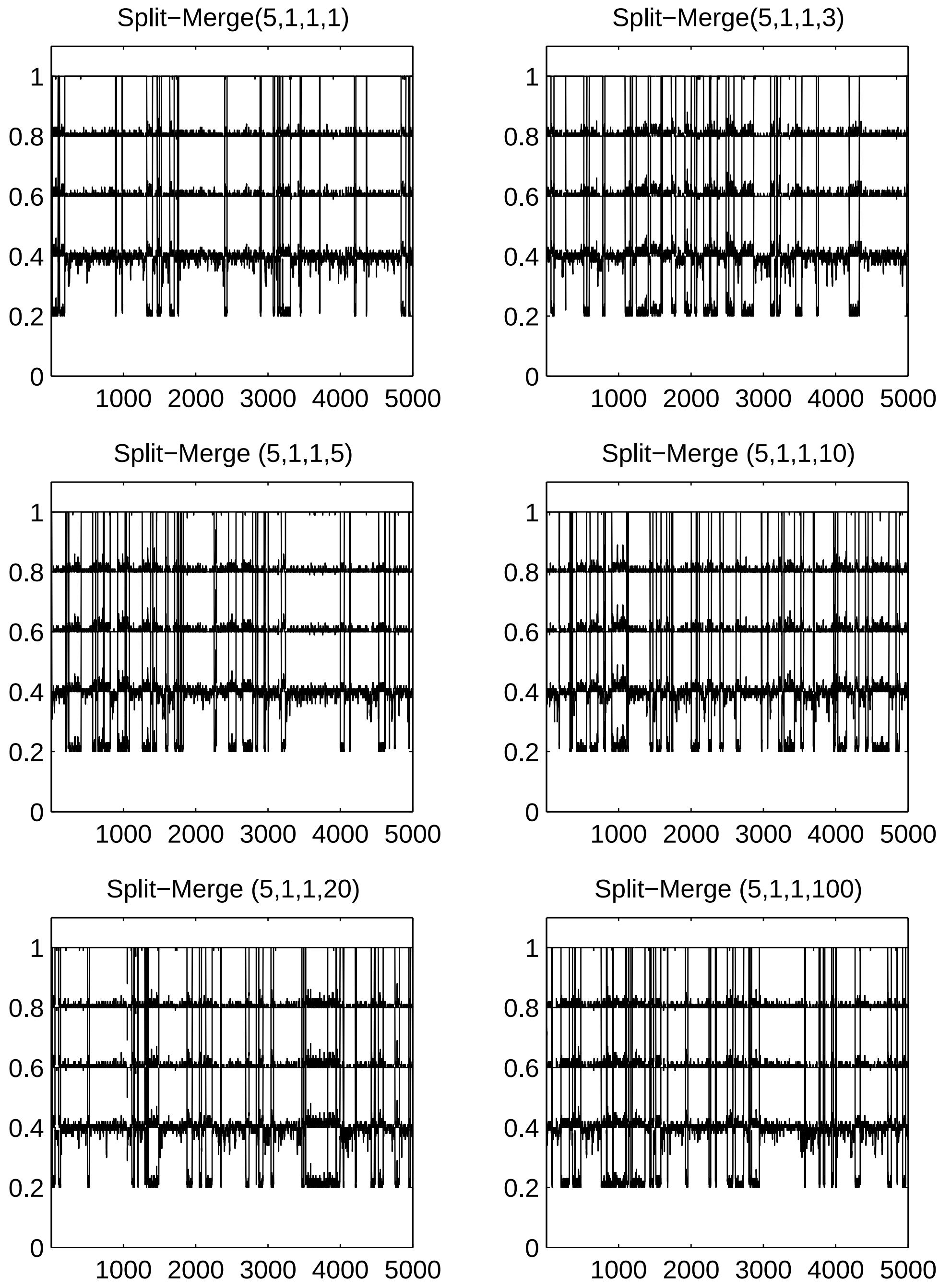


Figure 7: Trace plots showing the effect of the number of intermediate Gibbs sampling scans (split proposal) tuning parameter.



Split−Merge(5,1,1,1)

Split−Merge(5,1,1,3)

Split−Merge (5,1,1,5)

Split−Merge (5,1,1,10)

Split−Merge (5,1,1,20)

Split−Merge (5,1,1,100)

Figure 8: Trace plots showing the effect of the number of intermediate Gibbs sampling scans (merge proposal) tuning parameter.

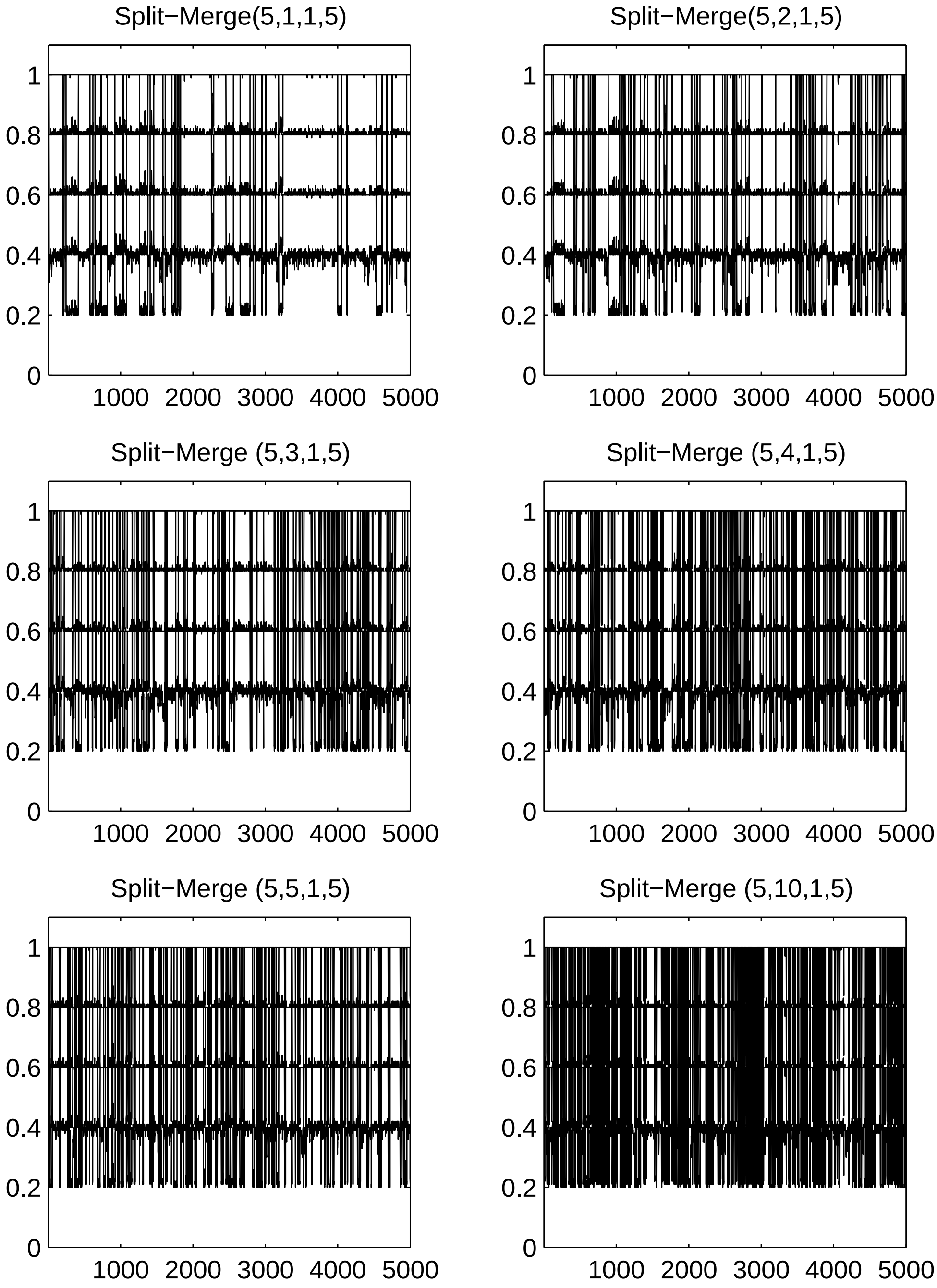


Figure 9: Trace plots showing the effect of the number of Metropolis-Hastings updates in a single iteration.

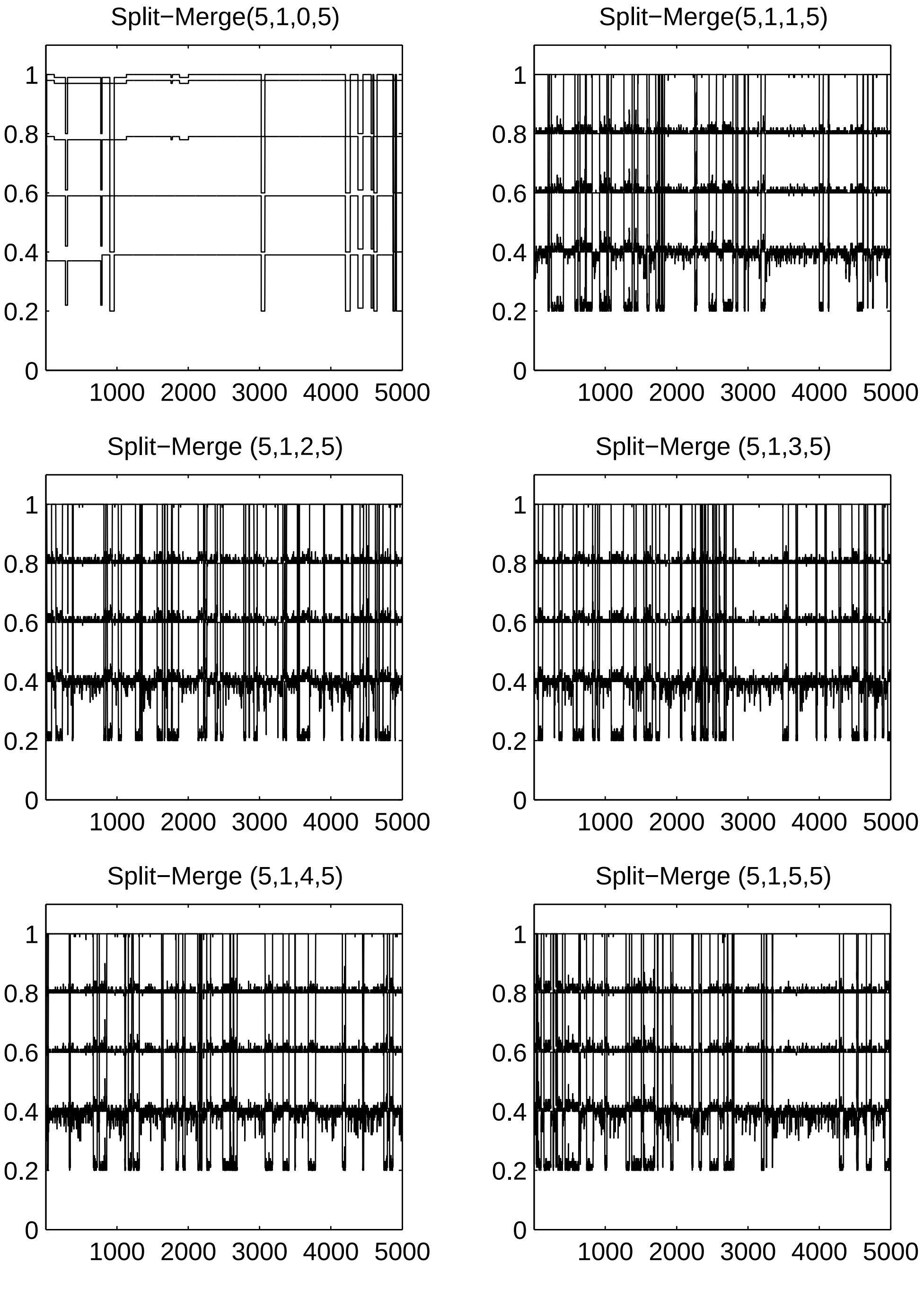


Figure 10: Trace plots showing the effect of the number of final complete Gibbs sampling scans.

indicator $I_{26,57}$ are performance measures considered for various settings of this algorithm shown in Table 6. Trace plots are given in Figures 7–10. The plots show the first 5000 iterations, but the simulations were run for 10,000 iterations in order to obtain better autocorrelation time estimates.

5.4.1 Number of intermediate Gibbs sampling scans for the split proposal

Increasing the number of intermediate Gibbs sampling scans will produce better split proposals since the restricted equilibrium distribution will be better approximated. It is not necessary to reach equilibrium to produce valid proposals. Therefore, the question is how many scans are necessary to achieve a reasonable allocation of observations between two components while keeping computation costs at a minimum.

From the trace plots in Figure 7, it is clear that as the number of scans is increased, the mixing dramatically improves. The sampler's performance for 100 intermediate scans is undeniably better than one intermediate scan. In terms of autocorrelations and Metropolis-Hastings acceptance rate, there are obvious improvements when scans are increased (Tables 6 and 7), but at the cost of computation time per iteration. Notice that 100 scans requires over five times the amount of time compared to ten scans.

This clear improvement by increasing the number of scans differs from the conjugate method, since the improvements quickly taper off as the scans increase in the conjugate case. This may be explained by the addition of the model parameters to the state of the Markov chain. Prior to restricted Gibbs sampling, values are drawn from the prior distribution of the parameters of the two split components. Depending on the choice of priors and size of the problem, this could take the restricted scans longer to converge or even reach reasonable splits. Improvements in performance could be made by selecting these values from the sample mean and variance, but this, of course, would make the procedure model-dependent, which we wish to avoid. However, in real data problems, for this type of Normal mixture model, choosing reasonable initial states would be useful.

It is difficult to say what the optimum number of intermediate scans for the split proposal should be, since this depends on the complexity of the problem and computational resources at one's disposal. For the comparisons considered in Section 5.3.2, it appears that for this data, five to ten scans did well in splitting amongst the five components, and additional scans would have been unnecessary.

5.4.2 Number of intermediate Gibbs sampling scans for the merge proposal

The intermediate Gibbs sampling scans to reach the launch state for the merge proposal differ from the scans for the split proposal because Gibbs sampling is only performed on the parameters for the single merged component. Indicators are not included, since the only way to merge two components is to group all observations together. This reduces the amount of work performed in one scan of restricted Gibbs sampling. These scans are also expected to converge faster than intermediate Gibbs sampling to reach the split launch state.

From the trace plots in Figure 8, it appears that the benefit of additional scans levels off after three

Table 6: Effects of the four tuning parameters.

	Time per iteration	*Autocorrelation*	*Autocorrelation*
Algorithm	*in seconds*	*time for Trace 1*	*time for Indicator* $I_{26,57}$
Split-Merge (1,1,1,5)	0.40	1725	593
Split-Merge (3,1,1,5)	0.47	359	182
Split-Merge (5,1,1,5)	0.54	126	38
Split-Merge (10,1,1,5)	0.71	66	23
Split-Merge (20,1,1,5)	1.04	45	16
Split-Merge (100,1,1,5)	3.67	28	14
Split-Merge (5,1,1,1)	0.52	354	108
Split-Merge (5,1,1,3)	0.52	87	36
Split-Merge (5,1,1,5)	0.54	126	38
Split-Merge (5,1,1,10)	0.54	80	29
Split-Merge (5,1,1,20)	0.56	85	31
Split-Merge (5,1,1,100)	0.75	91	40
Split-Merge (5,1,1,5)	0.54	126	38
Split-Merge (5,2,1,5)	0.78	60	22
Split-Merge (5,3,1,5)	1.00	49	16
Split-Merge (5,4,1,5)	1.25	38	9
Split-Merge (5,5,1,5)	1.50	52	12
Split-Merge (5,10,1,5)	2.70	31	6
Split-Merge (5,1,0,5)	0.25	718	417
Split-Merge (5,1,1,5)	0.54	126	38
Split-Merge (5,1,2,5)	0.81	70	28
Split-Merge (5,1,3,5)	1.09	75	31
Split-Merge (5,1,4,5)	1.49	74	29
Split-Merge (5,1,5,5)	1.67	85	31

Table 7: Acceptance rate for different numbers of intermediate Gibbs sampling scans for the split proposal.

Algorithm	*Acceptance rate in percent*
Split-Merge (1,1,1,5)	0.1
Split-Merge (3,1,1,5)	0.4
Split-Merge (5,1,1,5)	1.2
Split-Merge (10,1,1,5)	2.3
Split-Merge (20,1,1,5)	3.4
Split-Merge (100,1,1,5)	4.4

Table 8: Acceptance rate for different numbers of intermediate Gibbs sampling scans for the merge proposal.

Algorithm	*Acceptance rate in percent*
Split-Merge (5,1,1,1)	1.0
Split-Merge (5,1,1,3)	1.2
Split-Merge (5,1,1,5)	1.2
Split-Merge (5,1,1,10)	1.2
Split-Merge (5,1,1,20)	1.2
Split-Merge (5,1,1,100)	1.3

to five scans. Improvements in autocorrelation times (Table 6) and acceptance rate (Table 8) are not statistically significant. The standard error for trace 1 autocorrelation times based on dividing the ten thousand iterations into five equal samples is approximately twelve. The computation time per iteration is not much of a factor for these scans, since one to twenty scans take approximately the same time. These scans are much faster than the corresponding intermediate scans for the split proposal.

5.4.3 Number of split-merge updates per iteration

The trace plots for varying the number of split-merge updates per iteration are shown in Figure 9. Increasing the number of such updates has the effect of putting more emphasis on split-merge updates in comparison with incremental Gibbs sampling scans. As for the conjugate version, we see that the improvement that the improvement in autocorrelation time gradually diminishes for more than a few split-merge updates. In this example, no more than three per iteration seems desirable. A final incremental Gibbs sampling scan may not be necessary after every split-merge update. This is desirable, since such Gibbs sampling scans require more computational effort than a single split-merge update.

5.4.4 Number of final complete Gibbs sampling scans

As shown in Section 5.3, the split-merge Metropolis-Hastings updates need to be cycled with an incremental scan of the data. This is evident in the trace plots shown in Figure 10 and autocorrelations dropping from 718 to 126 after one final scan was added. The final incremental scans make the minor configuration adjustments for single observations that the split-merge procedure alone does not handle well (compare 0 vs. 1 scan in autocorrelation time for the indicator variable). Although improvements in autocorrelation time continue as the number of scans increase, it does not seem critical to perform more than one scan for most problems.

These full incremental scans are computationally expensive, so we prefer to use an incremental sampler that is computationally cheap. We recommend either the incremental Metropolis-Hastings or Gibbs sampling with $v = 1$ auxiliary parameters. Additional auxiliary parameters in our implementation are quite expensive, so no more than one will be used.

5.4.5 Suggestions for selecting tuning parameters values

The number of intermediate Gibbs sampling scans to reach the split launch state controls the performance of the procedure, since this decides the quality of the split proposal. We have shown empirically that a number of scans is necessary, and many should be performed if possible. It may be helpful to consider a more judicious approach to selecting an initial state than simply drawing from the prior to avoid performing a large number of these intermediate scans.

On the other hand, the number of intermediate scans to reach the merge launch state is less of an issue. The scans are computationally cheap, so several could be performed if desired. However, we observed that benefits taper off after only a few scans.

The number of Metropolis-Hastings updates per iteration and final full incremental scans of the data in the nonconjugate case behave similarly to the conjugate method. We prefer to keep these tuning parameters as low as possible and usually set them both to one to reduce computation time.

6 Illustration

The Dirichlet process mixture model is a useful tool in model-based, unsupervised cluster analysis. We illustrate the practical utility of our split-merge algorithm with a six-dimensional data set from Lubischew (1962) that has been previously used by West *et al* (1994). The data consists of six measurements of physical characteristics of three species of male beetles for a total of $n = 74$ beetles. The three species are *chactocnema concina, chactocnema heikertinger,* and *chactocnema heptapotamica*, in which $n_{conc} = 21$, $n_{heik} = 31$, and $n_{hept} = 22$.

The measurements for the i^{th} beetle are denoted as: $y_{ij} = (y_{i1}, \ldots, y_{i6})$ for $i = (1, \ldots, 74)$. The six measurements are:

$y_{.1} =$ width of the first joint	$\mu_1 = 177.3$	$\sigma_1 = 865.1$
$y_{.2} =$ width of the second joint	$\mu_2 = 124.0$	$\sigma_2 = 71.9$
$y_{.3} =$ maximal width of the aedeagus	$\mu_3 = 50.4$	$\sigma_3 = 7.6$
$y_{.4} =$ front angle of the aedeagus	$\mu_4 = 134.8$	$\sigma_4 = 107.1$
$y_{.5} =$ maximal width of the head	$\mu_5 = 13.0$	$\sigma_5 = 4.6$
$y_{.6} =$ aedeagus side-width	$\mu_6 = 95.4$	$\sigma_6 = 204.6$

The objective of our analysis is to recover the three latent classes corresponding to the three dif-

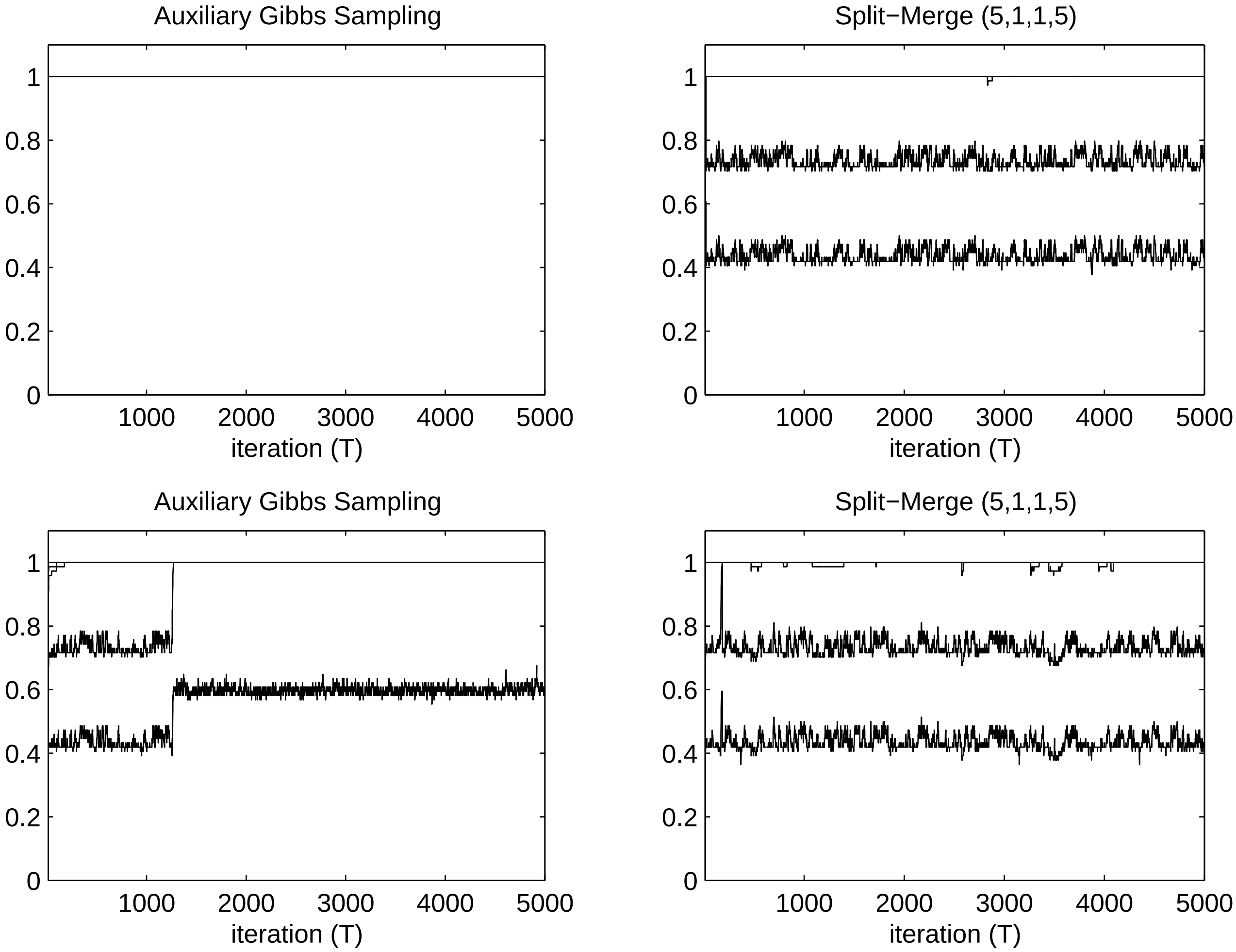


Figure 11: Trace plots comparing Auxiliary Gibbs Sampling to Split-Merge (5,1,1,5) for the beetle data using vague priors (top) and realistic priors (bottom).

ferent species of beetles **without** using the species information in the analysis. We apply the Normal-Gamma Dirichlet process mixture model to this data, identical to equation 17. The Dirichlet process parameter, α, is set to one. The values for the priors of the parameters have been set for each dimension as follows: $w_j = (w_1, \ldots, w_6) = (100, 100, 50, 100, 25, 100)$, $B_j^{-1} = (B_1^{-1}, \ldots, B_6^{-1}) = (500, 100, 25, 100, 25, 150)$ where B is a precision parameter, $r = 1$ across all six dimensions, and $R = 5$ across all six dimensions.

We applied the nonconjugate split-merge algorithm (5,1,1,5) and Neal's Gibbs sampling technique (2000) with $v = 3$ auxiliary components to this data. Computation time per iteration is similar for both algorithms. For each algorithm, results are provided for the case in which all observations are initially assigned to the same mixture component, and each algorithm is run for 5000 iterations.

From the two top trace plots given in Figure 11, it is evident that Gibbs sampling is unable to separate the data and leaves all observations in the same mixture component. It is clear that Gibbs sampling will take longer to reach equilibrium. On the other hand, split-merge splits the data into three major clusters (corresponding to the correction proportion of observations to species, i.e. 42%, 30% and 28%.) within the first twenty iterations.

To generate the two bottom trace plots in Figure 11, we set the prior values of w_j and B^{-1} to be more reflective of the data. The values used are: $w_j = (w_1, \ldots, w_6) = (100, 100, 50, 100, 10, 100)$ and $B_j^{-1} = (B_1^{-1}, \ldots, B_6^{-1}) = (800, 100, 10, 100, 10, 200)$. While Gibbs sampling does recover the three different species groups almost immediately, it is important to note that it becomes stuck in a low probability two-component configuration and mixes poorly. However, split-merge continues to mix well in a three-component configuration.

As a final check, the simulations were repeated by starting the simulation from a typical state of the competing method's apparent equilibrium distribution. Gibbs sampling stayed in the three-component state that it was started from, confirming that the three-component state has high posterior probability, and that the difference seen is not the result of some bug in the split-merge procedure. When the simulations were repeated using an initial state in which each observation is in a different component, the Gibbs sampler is able to reach equilibrium sooner and performs better.

The results from the beetle data illustration show that Gibbs sampling experiences a long burn-in time compared to the nonconjugate split-merge technique and is not always suitable for high-dimensional analysis. While it is true that the values of the priors for the parameters may not be ideal and that more realistic values may yield better sampling, often in real data analysis, there is no *a priori* information to suggest reasonable priors. A Markov chain Monte Carlo technique that can overcome poor choices in priors is preferred, as illustrated here, since this leads to shorter burn-in times and full exploration of the posterior distribution.

7 Discussion

The nonincremental split-merge procedure for nonconjugate models introduced in this article avoids the problem of being trapped in local modes, allowing the posterior distribution to be fully explored. In general, the nonconjugate split-merge procedure can become computationally expensive, but when Gibbs sampling or some other incremental procedure fails to reach equilibrium in a sensible amount of time, this procedure becomes necessary.

Another related issue is burn-in time. Even if an incremental procedure reaches stationarity within a desired time limit, one must often discard a large number of early iterations, which can lead to poor estimates. In split-merge type situations, the computational burden of using a nonincremental procedure is offset by its quick burn-in and dramatic improvement in performance.

A possible extension of the split-merge technique is to employ Dahl's (2003) sequentially allocated split-merge sampler as a method to initialize the intermediate Gibbs sampling step. This method could potentially provide a better starting state than our present method of performing a random split of items and selecting values for the parameters from the prior.

Acknowledgements

Sonia Jain's research is supported by a UCSD Division of Biostatistics and Bioinformatics start-up faculty grant. Radford Neal's research is supported by the Natural Sciences and Engineering Research Council of Canada. He holds a Canada Research Chair in statistics and machine learning.

References

Blackwell, D. and MacQueen, J. B. (1973) "Ferguson distributions via Pólya urn schemes", *Annals of Statistics*, vol. 1, pp. 353-355.

Blei, D. M. and Jordan, M. I. (2004) "Variational methods for the Dirichlet process", *ACM International Conference Proceeding Series: Proceedings of the twenty-first international conference on machine learning*, vol. 69, article no. 12.

Bush, C. A. and MacEachern, S. N. (1996) "A semiparametric Bayesian model for randomised block designs", *Biometrika*, vol. 83, pp. 275-285.

Dahl, D. B. (2003) "An improved merge-split sampler for conjugate Dirichlet process mixture models", Technical Report 1086, Department of Statistics, University of Wisconsin.

Do, K-A., Müller, P., Tang, F. "A Bayesian mixture model for differential gene expression", *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 54, pp. 627-644.

Escobar, M. D. (1994) "Estimating normal means with a Dirichlet process prior", *Journal of the American Statistical Association*, vol. 89, pp. 268-277.

Escobar, M. D. and West, M. (1995) "Bayesian density estimation and inference using mixtures", *Journal of the American Statistical Association*, vol. 90, pp.577-588.

Ferguson, T. S. (1983) "Bayesian density estimation by mixtures of normal distributions", in H. Rizvi and J. Rustagi (editors) *Recent Advances in Statistics*, pp. 287-303, New York: Academic Press.

Green, P. J. and Richardson, S. (2001) "Modelling heterogeneity with and without the Dirichlet process", *Scandinavian Journal of Statistics*, vol. 28, pp. 355-375.

Hastings, W. K. (1970) "Monte Carlo sampling methods using Markov chains and their applications", *Biometrika*, vol. 57, pp. 97-109.

Huelsenbeck, J. P., Jain, S., Frost, S. W. D., Pond, S. L. K. (2005) "A Dirichlet process model for detecting positive selection in protein-coding DNA sequences", submitted.

Jain, S. and Neal, R. M. (2004) "A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model", *Journal of Computational and Graphical Statistics*, vol. 13, pp. 158-182.

Jain, S. (2002) "Split-Merge Techniques for Bayesian Mixture Models", unpublished Ph.D. dissertation, University of Toronto, Department of Statistics.

Lubischew, A. (1962) "On the use of discriminant functions in taxonomy", *Biometrics*, volume 18, pp. 455-477.

MacEachern, S. N. (1994) “Estimating normal means with a conjugate style Dirichlet process prior”, *Communications in Statistics: Simulation and Computation*, vol. 23, pp. 727-741.

MacEachern, S. N. and Müller, P. (1998) “Estimating mixture of Dirichlet process models”, *Journal of Computational and Graphical Statistics*, vol. 7, pp. 223-238.

MacEachern, S. N., Clyde, M., Liu, J. (1999) “Sequential importance sampling for nonparametric Bayes models: the next generation”, *The Canadian Journal of Statistics*, vol. 27, pp. 251-267.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953) “Equation of state calculations by fast computing machines”, *Journal of Chemical Physics*, vol. 21, pp. 1087-1092.

Neal, R. M. (1992) “Bayesian mixture modeling”, in C. R. Smith, G. J. Erickson, and P. O. Neudorfer (editors) *Maximum Entropy and Bayesian Methods: Proceedings of the 11th International Workshop on Maximum Entropy and Bayesian Methods of Statistical Analysis*, Seattle, 1991, pp. 197-211, Dordrecht: Kluwer Academic Publishers.

Neal, R. M. (2000) “Markov chain sampling methods for Dirichlet process mixture models”, *Journal of Computational and Graphical Statistics*, vol. 9, pp. 249-265.

Richardson, S. and Green, P. J. (1997) “On Bayesian analysis of mixtures with an unknown number of components” (with discussion), *Journal of the Royal Statistical Society, Series B*, vol. 59, pp. 731-792.

Tierney, L. (1994) “Markov chains for exploring posterior distributions” (with discussion), *Annals of Statistics*, vol. 22, pp. 1701-1762.

West, M., Müller, P., and Escobar, M. D. (1994) “Hierarchical priors and mixture models, with application in regression and density estimation”, in P. R. Freeman and A. F. M. Smith (editors) *Aspects of Uncertainty*, John Wiley, pp. 363-386.

Xing, E., Sharan, R., Jordan, M. I. (2004) “Bayesian Haplotype Inference via the Dirichlet Process”, *ACM International Conference Proceeding Series: Proceedings of the twenty-first international conference on machine learning*, vol. 69, article no. 111.