Bayesian Mixture Labeling by Highest Posterior Density

Weixin YAO and Bruce G. LINDSAY

A fundamental problem for Bayesian mixture model analysis is label switching, which occurs as a result of the nonidentifiability of the mixture components under symmetric priors. We propose two labeling methods to solve this problem. The first method, denoted by PM(ALG), is based on the posterior modes and an ascending algorithm generically denoted ALG. We use each Markov chain Monte Carlo sample as the starting point in an ascending algorithm, and label the sample based on the mode of the posterior to which it converges. Our natural assumption here is that the samples converged to the same mode should have the same labels. The PM(ALG) labeling method has some computational advantages over other popular labeling methods. Additionally, it automatically matches the ''ideal'' labels in the highest posterior density credible regions. The second method does labeling by maximizing the normal likelihood of the labeled Gibbs samples. Using a Monte Carlo simulation study and a real dataset, we demonstrate the success of our new methods in dealing with the label switching problem.

KEY WORDS: Bayesian approach; Label switching; Markov chain Monte Carlo; Mixture model; Posterior modes.

1. INTRODUCTION

The m-component mixture models we consider here have densities of the form

$$p(x;\boldsymbol{\theta}) = \pi_1 f(x;\lambda_1) + \pi_2 f(x;\lambda_2) + \cdots + \pi_m f(x;\lambda_m),$$

where $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_m, \lambda_1, \ldots, \lambda_m)^T$, $f(\cdot)$ is the density of a discrete or continuous random vector called the ''component density,'' λ_j is the component specific parameter, which can be scalar or vector, and π_j is the proportion of jth subpopulation in the whole population with $\sum_{j=1}^{m} \pi_j = 1$. For a general introduction to mixture models, see Lindsay (1995), Böhning (1999), McLachlan and Peel (2000), and Mengersen (2009).

For any permutation $\boldsymbol{\sigma} = (\boldsymbol{\sigma}(1), \ldots, \boldsymbol{\sigma}(m))$ of the identity permutation $(1, \ldots, m)$, define the corresponding permutation of the parameter vector $\boldsymbol{\theta}$ by

$$\boldsymbol{\theta}^{\sigma} = (\pi_{\sigma(1)}, \ldots, \pi_{\sigma(m)}, \lambda_{\sigma(1)}, \ldots, \lambda_{\sigma(m)})^T.$$

Supposing that $\mathbf{x} = (x_1, \ldots, x_n)$ is a random sample from the m-component mixture density, the likelihood for $\mathbf{x}$ is

$$L(\boldsymbol{\theta};\mathbf{x}) = \prod_{i=1}^{n} \{\pi_1 f(x_i;\lambda_1) + \pi_2 f(x_i;\lambda_2) + \cdots + \pi_m f(x_i;\lambda_m)\}. \quad (1)$$

For any permutation $\boldsymbol{\sigma}$, $L(\boldsymbol{\theta}^{\boldsymbol{\sigma}}; \mathbf{x})$ will be numerically the same as $L(\boldsymbol{\theta}; \mathbf{x})$. Hence, if $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimator (MLE), $\hat{\boldsymbol{\theta}}^{\boldsymbol{\sigma}}$ is the MLE for any permutation $\boldsymbol{\sigma}$. In a technical sense, this means that the subscripts we assign to the π's and λ's are not identifiable unless we put additional restrictions on the model. This is the so-called ''label switching'' problem.

The label switching problem also occurs in Bayesian mixtures. Bayesian mixture analysis requires a prior distribution $\pi(\boldsymbol{\theta})$ for the parameters of the mixture model. If we do not have prior information that distinguishes between the components of a mixture model (i.e., $\pi(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}^{\boldsymbol{\sigma}})$) for any permutation $\boldsymbol{\sigma}$, the posterior distribution will be similarly symmetric and thus invariant to all the permutations of the component parameters, and the marginal posterior distributions for the parameters will also be identical for each mixture component. It is then meaningless to draw inference, relating to individual components, directly from Markov chain Monte Carlo (MCMC) samples using ergodic averaging before solving the label switching problem. For the illustrative examples of label switching, see Stephens (2000) and Jasra, Holmes, and Stephens (2005), among others.

Many methods have been proposed to deal with the labeling problem in Bayesian analysis. The easiest way to solve the label switching is to use an explicit parameter constraint so that only one permutation can satisfy it. This method was initially used by Diebolt and Robert (1994), Dellaportas et al. (1996), and Richardson and Green (1997). However, Celeux (1997); Celeux, Hurn, and Robert (2000); and Stephens (1997a, b, 2000) have all expressed their concerns about imposing an identifiability constraint. Another popular labeling method is to use a relabeling algorithm (Celeux 1998; Stephens 2000) that is designed to minimize a selected Monte Carlo risk. Stephens (2000) suggested a particular choice of loss function based on the Kullback-Liebler (KL) divergence. We will refer to this particular relabeling algorithm as the ''KL algorithm.'' Such risk-based relabeling algorithms have two liabilities: They give results that can depend on the choice of starting labels and they require one to compare $m!$ permutations in each iteration. In addition, relabeling algorithms require batch processing, which can be computationally demanding on storage. Celeux (1998) and Stephens (2000) did provide some alternative online versions, designed to reduce the storage requirements.

There are many other labeling methods in the literature. See, for example, Celeux et al. (2000); Frühwirth-Schnatter (2001); Hurn, Justel, and Robert (2003); Chung, Loken, and Schafer (2004); and Marin, Mengersen, and Robert (2005). Jasra, Holmes, and Stephens (2005) provided a good review about the existing methods to solve the label switching problem in Bayesian mixture modeling.

Our main proposed method, PM(ALG), uses each MCMC sample as the starting value for an ascending algorithm

Weixin Yao is Assistant Professor, Department of Statistics, 101 Dickens Hall, Kansas State University, Manhattan, KS 66506 (E-mail: *wxyao@ksu.edu*). Bruce G. Lindsay is Willaman Professor, Department of Statistics, The Pennsylvania State University, University Park, PA 16802 (E-mail: *bgl@psu.edu*). This research was supported by National Science Foundation grant DMS-0405637. The authors are grateful to the associate editor, three referees, and the joint editor for insightful comments on the article.

Journal of the American Statistical Association
June 2009, Vol. 104, No. 486, Theory and Methods
DOI 10.1198/jasa.2009.0037

generically denoted ALG. In our examples we will use the ECM (Meng and Rubin 1993) as the ascending algorithm. The samples are then relabeled according to the posterior modes to which they converge. We will show that the PM(ALG) method is superior to other existing proposals in capturing the credible regions of highest posterior density (HPD). We will also show by example that it is computationally much faster than many other existing proposals when the number of components is larger. In addition, PM(ALG) is an online algorithm, which can reduce the storage requirements. Furthermore, risk-based labeling methods have results that can depend on the choice of the initial labels for the samples. The PM(ALG) method does not require the initial labels, which can save considerable computation time.

The structure of the article is as follows. Section 2 introduces our new labeling methods. In Section 3, we use two simulation examples and a real dataset to compare the new labeling methods with two popular existing methods. We summarize our proposed labeling methods and discuss some future research work in Section 4.

2. INTRODUCTION OF NEW LABELING METHODS

Given a smooth objective function, such as the posterior density $p(\boldsymbol{\theta})$, one can cluster points in $\boldsymbol{\theta}$ space by using an ascent algorithm (ALG) that monotonically increases the objective function. Each point $\boldsymbol{\theta}$ is then assigned to the critical point to which the algorithm converges when $\boldsymbol{\theta}$ is used as an initial value. Although there is the possibility of converging to a saddle point, in a typical posterior, the posterior modes will be the points of attraction for almost all starting values for the ascent algorithm, and so we are creating ''modal clusters.'' See Li, Ray, and Lindsay (2007) for the use of this idea in density-based clustering.

2.1 Labeling Using Modal Clusters

The mixture labeling problem can be viewed as a clustering problem with a special structure. If we let the dataset be all the MCMC samples $\boldsymbol{\theta}$ together with all their possible permutations $\boldsymbol{\theta}^{\boldsymbol{\sigma}}$, then the objective is to find $m!$ tight clusters, each containing exactly one permutation of each sample element $\boldsymbol{\theta}$. One can then choose any one of these tight clusters to be the newly labeled dataset.

This relates to modal clustering as follows. If $\tilde{\boldsymbol{\theta}}$ is a mode, then so is $\tilde{\boldsymbol{\theta}}^{\boldsymbol{\sigma}}$ for any permutation $\boldsymbol{\sigma}$. If the chosen algorithm ascends from $\boldsymbol{\theta}$ to $\tilde{\boldsymbol{\theta}}^{\boldsymbol{\sigma}}$,we will say $\boldsymbol{\theta}$ has the same labeling as $\tilde{\boldsymbol{\theta}}^{\boldsymbol{\sigma}}$. If the algorithm is permutation symmetric, we will also know that $\boldsymbol{\theta}^{\boldsymbol{\sigma}^{-1}}$, where $\boldsymbol{\sigma}^{-1}$ is the inverse permutation of $\boldsymbol{\sigma}$ such that $(\boldsymbol{\theta}^{\boldsymbol{\sigma}})^{\boldsymbol{\sigma}^{-1}} = \boldsymbol{\theta}$ for any $\boldsymbol{\theta}$ and $\boldsymbol{\sigma}$, will be given the same labeling as $\tilde{\boldsymbol{\theta}}$.

If the posterior density has a maximal mode at $\tilde{\boldsymbol{\theta}}$ it also has modes at all permutations of $\tilde{\boldsymbol{\theta}}$ and they are all maximal. We can pick one such mode to be our reference mode (hence, the reference label)—say, by order constraint (OC) labeling on some parameter. Denote by $\hat{\boldsymbol{\theta}}$ the chosen reference maximal mode. If a sampled $\boldsymbol{\theta}$ converges to a maximal mode—say, $\hat{\boldsymbol{\theta}}^{\boldsymbol{\sigma}}$—then the natural label of $\boldsymbol{\theta}$ is $\boldsymbol{\sigma}^{-1}$ because $\boldsymbol{\theta}^{\boldsymbol{\sigma}^{-1}}$ would ascend to $\hat{\boldsymbol{\theta}}$. If the $\boldsymbol{\theta}$ converges to a minor mode—say, $\boldsymbol{\theta}_*$—we could create a labeling system for all the samples $\boldsymbol{\theta}$ that are attracted to $\boldsymbol{\theta}_*$ (or its permutations) by creating a secondary reference mode $\hat{\boldsymbol{\theta}}_2$. If the reference mode $\hat{\boldsymbol{\theta}}_2$ was chosen so that it matched the label with the major mode $\hat{\boldsymbol{\theta}}$ using a risk-based criterion that makes $\hat{\boldsymbol{\theta}}_2 = \boldsymbol{\theta}_*^{\boldsymbol{\sigma}}$ most similar to $\hat{\boldsymbol{\theta}}$ for some $\boldsymbol{\sigma}$, then we have a system that labels all points attracted to both the maximal and minor modes. One can extend this idea to any number of minor modes.

If one wishes to use this algorithm in a way that does not require storage of all the MCMC samples, one needs to find the reference maximal mode $\hat{\boldsymbol{\theta}}$ in advance of processing. Ascending algorithms are guaranteed only to find local modes, not global ones. To find one of the $m!$ maximal modes, we need to start from different initial values and choose the converged mode that has the largest posterior. Practically, the initial values can be chosen equally spaced from the burn-in samples of the MCMC sampling, such as choosing one from every 1,000 (or more) burn-in samples. If one uses a burn-in of length 10,000 to 20,000, then, based on our experience, the resulting 10 to 20 initial values will have every good chances of finding the maximal mode. (Suppose that the maximal mode garners 50% of the samples in posterior probability. If one were to take independent samples, then the chance it does not show up in 20 trials is about 0.000001 in probability.) If the MLE is not difficult to find, we can also include it as one of the initial values. (Although it is possible that one will find a higher mode later in the sampling, it is unlikely to attract many of the samples, and so it might not be a wise choice to be the mode of reference.) Using this strategy, we successfully found all the maximal modes in the examples in Section 3. As an additional precaution, a general global search optimization technique, such as genetic algorithms (Holland 1975; Goldberg 1989; Davis 1991) and adaptive simulated annealing (Corana, Marchesi, Martini, and Ridella 1987; Ingber and Rosen 1992), can also be used to find the maximal mode. For an off-line version of our algorithm, one could also find the maximal mode at the end of the sampling. In our experience, the maximal mode is the one to which most of the samples converge when each MCMC sample is used as the starting value for the ascending algorithm.

Take the previously found reference maximal mode $\hat{\boldsymbol{\theta}}$ and its associated minor modes as the reference modes (hence, the reference labels). The aim of labeling is to find the labels $(\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_N)$ such that $\{\boldsymbol{\theta}_1^{\boldsymbol{\sigma}_1}, \ldots, \boldsymbol{\theta}_N^{\boldsymbol{\sigma}_N}\}$ have the same label meaning as $\hat{\boldsymbol{\theta}}$. Roughly speaking, this means that we would like this labeling to create a tight cluster around $\hat{\boldsymbol{\theta}}$. The algorithm of our proposed labeling method is as follows.

Algorithm 1: Labeling Based on Posterior Modes and an Ascent Algorithm (PM(ALG))

Step 1. Taking each MCMC sample $\{\boldsymbol{\theta}_t, t = 1, \ldots, N\}$ as the initial value, find the corresponding converged mode $\{\mathbf{m}_t, t = 1, \ldots, N\}$ using the given ascent algorithm ALG.

Step 2. Apply to m_t the OC labeling used to define $\hat{\boldsymbol{\theta}}$, denoted by $\boldsymbol{\sigma}_t^*$ (hence, $m_t^{\sigma_t^*}$ has the same OC as $\hat{\boldsymbol{\theta}}$) and find the label $\boldsymbol{\sigma}_t$ of $\boldsymbol{\theta}_t$ based on the following situations:

(a) If $\mathbf{m}_t^{\sigma_t^*}$ is $\hat{\boldsymbol{\theta}}$, up to numerical error, then $\boldsymbol{\sigma}_t = \boldsymbol{\sigma}_t^*$.

(b) If $\mathbf{m}_t^{\sigma_t^*}$ is not $\hat{\boldsymbol{\theta}}$, but it is equivalent (up to a permutation) to a known reference minor mode—say, $\hat{\boldsymbol{\theta}}_2$—assign the label $\boldsymbol{\sigma}_t$ such that $\mathbf{m}_t^{\sigma_t} = \hat{\boldsymbol{\theta}}_2$.

(c) If $\mathbf{m}_t^{\sigma_t^*}$ is not $\hat{\boldsymbol{\theta}}$ and is not equivalent to a preexisting reference minor mode, create a new reference minor mode $\mathbf{m}_t^{\sigma_t}$, where $\boldsymbol{\sigma}_t$ is based on a risk-based criterion such as least squares:

$$\boldsymbol{\sigma}_t = \arg\min_{\sigma} (\mathbf{m}_t^{\sigma} - \hat{\boldsymbol{\theta}})^T (\mathbf{m}_t^{\sigma} - \hat{\boldsymbol{\theta}}) \ . \tag{2}$$

■

The main idea of PM(ALG) is to explore the geometry of the mixture posterior by using each MCMC draw as a starting point for the ascent algorithm ALG and labeling the samples based on the modes of the posterior density to which they converge. The natural assumption here is that the samples converged to the same mode should have the same labels.

2.2 The ECM Algorithm

The EM class of algorithms provides a natural ascent methodology for clustering because these algorithms are easy to use, requiring no choice of tuning parameters to maintain their ascent property. We can extend the modal clustering idea to the posterior density to label samples by constructing a Bayesian EM algorithm suitable in many mixture models. If the algorithm presented here is not suitable in a given Bayesian mixture problem, it could be replaced with a gradient ascent algorithm that is suitably tuned to provide monotonic increases in the posterior.

Let us start by introducing an ascending algorithm to find the local posterior mode of Bayesian mixtures. Define the latent variable

$$Z_{ij} = \begin{cases} 1, & \text{if the } i \text{ th observation is from } j\text{th component;} \\ 0, & \text{otherwise.} \end{cases}$$

Then the complete likelihood for $(\mathbf{x}, \mathbf{Z})$ is

$$L(\boldsymbol{\theta}; \mathbf{x}, \mathbf{Z}) = \prod_{i=1}^{n} \prod_{j=1}^{m} [\pi_j f(x_i; \lambda_j)]^{Z_{ij}},$$

where $\mathbf{Z} = \{Z_{ij},\ 1 \le i \le n,\ 1 \le j \le m\}$, and the complete posterior distribution is

$$p(\boldsymbol{\theta}, \mathbf{Z}|\mathbf{x}) = \frac{1}{p(\mathbf{x})} \pi(\boldsymbol{\theta}) L(\boldsymbol{\theta}; \mathbf{x}, \mathbf{Z}),$$

where $p(\mathbf{x})$ is the marginal density for $\mathbf{x} = \{x_1, \ldots, x_n\}$.

Suppose that all the prior parameters are fixed and we are in a setting such that we can use a Gibbs sampler to get the MCMC samples (i.e., there exists a partition of $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{(1)}, \ldots, \boldsymbol{\theta}_{(p)}\}$ such that all the conditional complete posterior distributions $\{p(\boldsymbol{\theta}_{(i)} \,|\, \ldots), 1 \le i \le p\}$ are easily found), where $\boldsymbol{\theta}_{(i)}$ can be scalar or vector and $|\ \ldots$ denotes conditioning on all other parameters and the latent variable $\mathbf{Z}$. By combining the ideas of ECM (Meng and Rubin 1993), a class of GEM algorithm (Dempster, Laird, and Rubin 1977), and properties of a Gibbs sampler, we propose the following algorithm to find the posterior modes of Bayesian mixtures.

Algorithm 2: ECM Algorithm for Bayesian Mixtures (ECM(BM)). Starting with the initial value of $\boldsymbol{\theta}$, iterate the following two steps until a fixed point is reached.

E-Step. Find the conditional expectation of the latent variable $\mathbf{Z}$—in other words, the classification probability for each observation

$$p_{ij} = E(Z_{ij} \,|\, \mathbf{x}, \boldsymbol{\theta}) = \frac{\pi_j f(x_i; \lambda_j)}{\sum_{l=1}^{m} \pi_l f(x_i; \lambda_l)}.$$

M-Step. Update θ by maximizing the conditional complete posterior distribution $p(\theta_{(i)} \,|\, \ldots)$, $1 \le i \le p$ sequentially with the latent variable Z_{ij} replaced by the classification probability p_{ij}. ■

From the theory of ECM (Meng and Rubin 1993) and GEM (Dempster, Laird, and Rubin 1977), we know that the posterior distribution $p(\boldsymbol{\theta})$ will increase after each iteration. Moreover, it is clear that the algorithm has a natural equivalence property. If $\boldsymbol{\theta}$ converges to $\boldsymbol{\theta}_*$, then $\boldsymbol{\theta}^{\boldsymbol{\sigma}}$ converges to $\boldsymbol{\theta}_*^{\boldsymbol{\sigma}}$. This will mean that if a modal cluster is formed by the algorithm, a fixed permutation of its elements will also be a cluster that ascends to the permuted mode.

2.3 Implementation Issues

One nice feature of PM(ALG) is that the algorithm does not depend on any initial choice of labels, which can save much computation time compared with other relabeling algorithms. In addition, PM(ALG) is an online algorithm, which does not require batch processing and thus reduces the amounts of storage.

Notice that the PM(ALG) method does not require one to compare $m!$ permutations to find each $\boldsymbol{\sigma}_t$ except for the initial discovery of a minor mode. In our experience, most of the samples will converge to one of the $m!$ maximal modes. If $\mathbf{m}_t$ is one of the maximal modes (i.e., there exists $\boldsymbol{\sigma}_t$ such that $\mathbf{m}_t^{\sigma_t} = \hat{\boldsymbol{\theta}}$), the natural label of $\mathbf{m}_t$ is $\boldsymbol{\sigma}_t$ and it can be directly found by ordering $\mathbf{m}_t$ (based on any one-dimensional component parameter such as a component mean) the same as the reference mode $\hat{\boldsymbol{\theta}}$.

For example, for a univariate normal mixture, suppose the reference maximal mode $\hat{\boldsymbol{\theta}}$ is ordered by the component means, so

$$\hat{\boldsymbol{\theta}} = (\hat{\pi}_1, \ldots, \hat{\pi}_m, \hat{\mu}_1, \ldots, \hat{\mu}_m, \hat{\sigma}_1, \ldots, \hat{\sigma}_m),$$

where $\hat{\mu}_1 < \hat{\mu}_2 < \cdots < \hat{\mu}_m$. Suppose $\mathbf{m}_t$ is one of the $m!$ maximal modes and we want to find $\boldsymbol{\sigma}_t$ such that $\mathbf{m}_t^{\sigma_t} = \hat{\boldsymbol{\theta}}$. If the label $\boldsymbol{\sigma}^*$ is the one such that $\mathbf{m}_t^{\sigma^*}$ is also ordered by the component means μ's, then we have $\boldsymbol{\sigma}^* = \boldsymbol{\sigma}_t$. Hence, if $\mathbf{m}_t$ is one of the maximal modes, the labeling of $\mathbf{m}_t$ will be as easy as the OC labeling. This property makes PM(ALG) much faster, when m is large, than other risk-based relabeling algorithms, which require $m!$ comparison in each iteration.

If $\mathbf{m}_t$ is a minor mode, we use the distance criteria (2) to find $\boldsymbol{\sigma}_t$ such that the distance between $\mathbf{m}_t^{\sigma_t}$ and $\hat{\boldsymbol{\theta}}$ is minimized. Many other existing labeling methods can be also used to label the minor modes. For example, similar to the KL algorithm, we can also use the Kullback-Leibler divergence from the distribution on clusters based on the reference mode $\hat{\boldsymbol{\theta}}$, to the distribution on clusters based on $\mathbf{m}_t^{\sigma}$. Hence, the criteria (2) can be replaced by

$$\boldsymbol{\sigma}_t = \arg\max_{\sigma} \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}(\hat{\boldsymbol{\theta}}) \log(p_{ij}(m_t^{\sigma})) \,, \tag{3}$$

where $p_{ij}(\boldsymbol{\theta}) = \pi_j f(x_i;\lambda_j)/\boldsymbol{\Sigma}_{l=1}^{m}\{\pi_l f(x_i;\lambda_l)\}$ is the classification probability of x_i from the jth component based on parameter $\boldsymbol{\theta}$. One nice feature of this criteria is its invariance to the scale effect of parameters. Notice that both of the previous two criteria ((2) and (3)) require $m!$ comparisons to get the label $\boldsymbol{\sigma}_t$ for the minor mode $\mathbf{m}_t$.

2.4 HPD Labels and Labeling Credibility

In this section, we will describe one very attractive feature of PM(ALG) based on the new concept of ''HPD label.'' This leads to a new method to assess the quality of the labels that have been assigned. To simplify the explanation, let us assume that the number of components is two (i.e., $m = 2$) and there is only one permutation class of modes (i.e., maximal modes).

Suppose that the parameter space is the full product space Ω (π's in the simplex, λ's in cross-product space). Let us say that a subset S of Ω is an identifiable subset if there are no degenerate points in S and for every $\boldsymbol{\theta} \in \boldsymbol{S}$, we have $\boldsymbol{\theta}^{\boldsymbol{\sigma}} \notin S$, where $\boldsymbol{\sigma} = (2, 1)$. If we restrict the parameters to lie in an identifiable subset S, then all the parameters have unique labels. For any identifiable subset S, we can create an image set by permutation: $S^{\boldsymbol{\sigma}} = \{\boldsymbol{\theta}^{\boldsymbol{\sigma}} : \boldsymbol{\theta} \in S\}$. The image set is also identifiable.

Let us suppose that our goal is to build credible regions for the parameters, for any fixed credibility level $1 - \alpha$, using regions of HPD. Such credible regions have the theoretical justification of being the smallest volume credible regions at a fixed level. To be specific, let the regions have the form $\psi_c = \{\boldsymbol{\theta} : p(\boldsymbol{\theta}) \geq c\}$, where $c = c_\alpha$ is chosen to give the target credibility level. For a given mode $\hat{\boldsymbol{\theta}}$, we define $S_c(\hat{\boldsymbol{\theta}})$ to be the maximal connected subset of the HPD region ψ_c that contains $\hat{\boldsymbol{\theta}}$. We will call $S_c(\hat{\boldsymbol{\theta}})$ the ''modal region'' defined by c and $\hat{\boldsymbol{\theta}}$. When $c = p(\hat{\boldsymbol{\theta}})$, $S_c(\hat{\boldsymbol{\theta}})$ is the single point $\{\hat{\boldsymbol{\theta}}\}$. As c decreases, the size of $S_c(\hat{\boldsymbol{\theta}})$ increases. Note also that $S_c^{\sigma}(\hat{\boldsymbol{\theta}})$, the permutation image of $S_c(\hat{\boldsymbol{\theta}})$, is automatically the maximal connected subset that contains $\hat{\boldsymbol{\theta}}^{\boldsymbol{\sigma}}$—in other words, $S_c^{\sigma}(\hat{\boldsymbol{\theta}}) = S_c(\hat{\boldsymbol{\theta}}^{\boldsymbol{\sigma}})$. As long as c is sufficiently large, the set ψ_c will be the union of disjoint identifiable sets $S_c(\hat{\boldsymbol{\theta}})$ and $S_c(\hat{\boldsymbol{\theta}}^{\boldsymbol{\sigma}})$. Assume that we have specified such a value of c. Then it is natural to use the identifiable (and hence well-labeled) set $S_c(\hat{\boldsymbol{\theta}})$ to describe the HPD region, because any other $S_c(\hat{\boldsymbol{\theta}}^{\boldsymbol{\sigma}})$ is just the permuted (relabeled) image of $S_c(\hat{\boldsymbol{\theta}})$. In fact, if we view the problem asymptotically in n, these identifiable sets will eventually be disjoint for any c in accordance with the asymptotic identifiability of the labels.

Because the parameters have unique labels in $S_c(\hat{\boldsymbol{\theta}})$, the HPD region $S_c(\hat{\boldsymbol{\theta}})$ gives a natural labeling to all $\boldsymbol{\theta}$ values it contains. We will call these labels the ''HPD labels,'' and will consider them to be the ideal labels. Note that not all points can be given HPD labels, because at some value of c—say, c_0—the modal regions for $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\theta}}^{\boldsymbol{\sigma}}$ intersect or they contain some degenerate points. For c larger than c_0, however, we can define unique HPD labels. We will let $\alpha_0 = \Pr(p(\boldsymbol{\theta}) > c_0)$ be the posterior probability of the points with HPD labels, and will call it the ''labeling credibility.''

Assuming that the HPD region $S_c(\hat{\boldsymbol{\theta}})$ contains the single mode $\hat{\boldsymbol{\theta}}$, if we start an ascending algorithm at $\boldsymbol{\theta}$ within this HPD region, it necessarily climbs the posterior to $\hat{\boldsymbol{\theta}}$, and is so labeled. (The only way to leave the set is for the algorithm to decrease the posterior.) Hence, the PM(ALG) method will assign the same labels to all the points of $S_c(\hat{\boldsymbol{\theta}})$ and thus PM(ALG) recovers all the ideal HPD labels, which is a primary motivation and essentially unique benefit of labeling based on an ascending algorithm. Specifically, if $\hat{\boldsymbol{\theta}}$ is the reference mode, then any point of HPD region $S_c(\hat{\boldsymbol{\theta}})$ has the label with identity permutation (1, 2) and any point of region $S_c(\hat{\boldsymbol{\theta}}^{\boldsymbol{\sigma}})$ has the label $\boldsymbol{\sigma}^{-1} = (2, 1)$.

If there are minor modes, the situation is somewhat more complex. Now each minor mode also creates a locally identifiable set that grows with index c shrinking. As c becomes small enough, the HPD region around one minor mode might begin to intersect with HPD regions from other minor or major modes. If we always take c to be sufficiently large that there is a single mode in the major modal regions, then the ALG always identifies the ideal labels. If c is set low enough that there are one or more minor modes in $S_c(\hat{\boldsymbol{\theta}})$, then it is possible that our assignment method using a risk-based criterion does not agree with the HPD region, which might cluster the minor modes differently. (Although we would have liked for PM(ALG) to agree with HPD even for minor modal clusters, doing so would add considerable computational complexity to the problem.)

Let c^* be the maximum posterior value among all the degenerate points. We define the ''upper labeling credibility'' to be $\alpha^* = \Pr(p(\boldsymbol{\theta}) > c^*)$. We will argue next that α^* provides an upper bound to, and a good approximation to, the labeling credibility α_0. As such, it is a measure of how difficult the labeling problem is. It also indicates to us the level of arbitrariness involved in assigning labels to all sample points. (Small α^* implies that very few sample points will have HPD labels.)

When $c < c^*$, the modal region $S_c(\hat{\boldsymbol{\theta}})$ will contain one or more degenerate points and thus it is not identifiable. Hence, $c_0 \geq c^*$ and the upper credibility level α^* is an ''upper bound'' for α_0, the proportion of points with ideal HPD labels. This upper bound becomes the actual labeling credibility if $S_c(\hat{\boldsymbol{\theta}})$ and $S_c(\hat{\boldsymbol{\theta}}^{\boldsymbol{\sigma}})$ first connect at a degenerate point, because when $c > c^*$, $S_c(\hat{\boldsymbol{\theta}})$ and $S_c(\hat{\boldsymbol{\theta}}^{\boldsymbol{\sigma}})$ are not connected and they do not contain any degenerate points. Unfortunately, it is difficult to verify whether this property holds in general, or even in a specific data analysis. Yao (2007) provided some graphical checking methods, and the empirical evidence was that the upper bound α^* was indeed the labeling credibility α_0. We will therefore say that sample points with posterior greater than c^* are ''likely'' HPD labeled.

The value of c^* and hence the upper credibility level α^* can be easily estimated based on the ECM(BM) algorithm. When using ECM(BM), the updated point after each iteration from the degenerate point will be also the degenerate point. So, the c^* value can be found by running the ECM(BM) algorithm starting from several degenerate points and choosing the converged degenerate mode with the largest posterior. In practice, one can make use of the MLE of $(m - 1)$-component mixture when choosing the starting points. For example, suppose $m = 3$ and $((\hat{\pi}, 1 - \hat{\pi}), (\hat{\lambda}_1, \hat{\lambda}_2))$ is the MLE of a two-component

mixture. The parameter sets $((\hat{\pi}, 1-\hat{\pi}, 0), (\hat{\lambda}_1, \hat{\lambda}_2, \lambda_3))$, where λ_3 can be any real value such as the one maximizing the prior for λ_3 can be included as one of the initial values for the ECM(BM) algorithm. Denote the estimate of c^* by $\hat{c}^*$. Then α^* can be estimated by the proportion of MCMC samples with posterior larger than $\hat{c}^*$.

2.5 The Classification MLE Method

From the asymptotic theory for the posterior distribution (see Walker (1969) and Frühwirth-Schnatter (2006, Secs. 1.3, 2.4.3, 3.3), we know that when sample size is large, the ''correctly'' labeled MCMC samples should, approximately, follow the normal distribution. Based on this property, we propose another method to do labeling based on minimizing the following negative lognormal likelihood over $(\bar{\boldsymbol{\theta}}, \boldsymbol{\Sigma}, \sigma)$:

$$L(\bar{\boldsymbol{\theta}}, \boldsymbol{\Sigma}, \sigma) = N \log(|\boldsymbol{\Sigma}|) + \sum_{t=1}^{N} (\boldsymbol{\theta}_t^{\sigma_t} - \bar{\boldsymbol{\theta}})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\theta}_t^{\sigma_t} - \bar{\boldsymbol{\theta}}) \quad (4)$$

where $\bar{\boldsymbol{\theta}}$ is the center value for the normal distribution, $\boldsymbol{\Sigma}$ is the covariance structure, and $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_N)$. This corresponds to applying the classification MLE clustering method to the full set of permuted $\boldsymbol{\theta}$-values (see Symons (1981), McLachlan (1982), and McLachlan and Basford (1988)). As we shall see, this is a batch processing algorithm that comes close to matching the likely HPD labels.

If we assume $\boldsymbol{\Sigma}$ is diagonal (i.e., all the parameters are orthogonal), this labeling method is exactly the same as that of Celeux (1998). We know that for the standard parameterization, the parameters are not orthogonal. So here we use the general covariance matrix $\boldsymbol{\Sigma}$.

The algorithm to find labels by minimizing (4) is as follows.

Algorithm 3: Labeling by Normal Likelihood (NORMLH) Starting with some initial values for $\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_N$ (setting them based on an OC, for example), iterate the following two steps until a fixed point is reached.

Step 1. Update $\bar{\boldsymbol{\theta}}$ and $\boldsymbol{\Sigma}$ by minimizing (4)

$$\bar{\boldsymbol{\theta}} = \frac{1}{N} \sum_{t=1}^{N} \boldsymbol{\theta}_t^{\sigma_t},$$

$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{t=1}^{N} (\boldsymbol{\theta}_t^{\sigma_t} - \bar{\boldsymbol{\theta}})(\boldsymbol{\theta}_t^{\sigma_t} - \bar{\boldsymbol{\theta}})^T.$$

Step 2. For $t = 1, \ldots, N$, choose $\boldsymbol{\sigma}_t$ by

$$\boldsymbol{\sigma}_t = \arg\min_{\sigma} (\boldsymbol{\theta}_t^{\sigma} - \bar{\boldsymbol{\theta}})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\theta}_t^{\sigma} - \bar{\boldsymbol{\theta}}) \ .$$

In Step 2, after any change of $\boldsymbol{\sigma}_t$, we could also update $\bar{\boldsymbol{\theta}}$ and $\boldsymbol{\Sigma}$, thereby increasing the speed of convergence but increasing complexity. Because in each step of the previous algorithm the objective function (4) decreases, this algorithm must converge. However, like other general relabeling algorithms, this algorithm is only guaranteed to converge to a local minimum that depends on the initial labels. To get better results, we might choose a number of different starting labels.

The NORMLH method has a simple and nice explanation, and runs much faster than the PM(ALG) method if m is not large. As one referee pointed out, if m is too large or the dimension of the data is large, this method could have numerical problems resulting from the calculation of $\boldsymbol{\Sigma}^{-1}$. If this problem occurs, one could add a penalty function to the objective function. A penalty of the form $\lambda \times \text{Trace}(\boldsymbol{\Sigma}^{-1})$ creates a ridge-type estimator for $\boldsymbol{\Sigma}$.

The dissertation of Yao (2007) described two other related labeling methods. Yao (2007) proposed to find the labels of the MCMC samples along with the mean $\bar{\boldsymbol{\theta}}$ by minimizing the determinant of the sample covariance matrix

$$L(\bar{\boldsymbol{\theta}}, \boldsymbol{\sigma}) = \det\left(\frac{1}{N} \sum_{t=1}^{N} (\boldsymbol{\theta}_t^{\sigma_t} - \bar{\boldsymbol{\theta}})(\boldsymbol{\theta}_t^{\sigma_t} - \bar{\boldsymbol{\theta}})^T \right), \quad (5)$$

where $\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_N$ and $\det(A)$ is the determinant of matrix A. The main idea of this method is to find the labels by minimizing the ellipsoidal volume of the labeled sample clusters. Yao (2007) argued that NORMLH produces similar results to the previous method but is much faster.

Without using the covariance $\boldsymbol{\Sigma}$ in Step 2, the $\boldsymbol{\sigma}$ and $\bar{\boldsymbol{\theta}}$ found by Algorithm 3 in fact minimize $L(\bar{\boldsymbol{\theta}}, \sigma) = \boldsymbol{\Sigma}_{t=1}^{N} (\boldsymbol{\theta}_t^{\sigma_t} - \bar{\boldsymbol{\theta}})^T (\boldsymbol{\theta}_t^{\sigma_t} - \bar{\boldsymbol{\theta}})$. This method is the K-means-type labeling method introduced in the dissertation of Yao (2007). When $\boldsymbol{\theta}$ only contains m parameters (one for each component)—say, the m component means for one dimension data—this labeling method will be exactly the same as the OC labeling. However, unlike the OC labeling, this method can incorporate different component parameters together and can be easily extended to the multivariate case.

3. EXAMPLES

In this section, we will use two simulation examples and one real dataset to compare our proposed two labeling methods (PM(ALG) and NORMLH) with OC labeling and Stephens' KL algorithm (KL). The OC method refers to ordering on the mean parameters. For PM(ALG), we used ECM(BM) for the ascent algorithm and we will refer to this particular modal cluster labeling method as PM(ECM). We used the MLE and 20 equally spaced samples from the 20,000 burn-in samples as the initial values to find the reference maximal mode. In all our examples, we successfully found the maximal modes.

For comparison, we report the number of different labels for each method that differed from PM(ECM). We also report the newly defined upper labeling credibility level, which can approximate the proportion of the HPD labels and measure how difficult the labeling problem is.

All the computations were done in Matlab 7.0 using a personal desktop computer with Intel Core 2 Quad CPU 2.40 GHz. It is known that the OC method is the fastest one and it takes no more than several seconds in our examples. Hence, we only report the runtimes for KL, NORMLH, and PM(ECM). Here we have used PM(ECM) in batch mode so that we can determine its runtime in direct comparison with the others. Because the runtime for the NORMLH and KL algorithms depends on the number of starting points (i.e., the initial labels for all samples), we only report the runtimes of NORMLH and KL when using the PM(ECM) labels as the initial labels. (The real runtimes for NORMLH and KL could be much longer. If one

used 10 different initializations for the algorithm, it might take about 10 times as long (generally longer than that because the runtimes of NORMLH and KL depend on the quality of start values).) Using these starts also ensures that the other methods are as similar to PM(ECM) as possible.

3.1 Simulation Studies

Example 3.1. We generated 400 data points from 0.3N(0,1) + 0.7N(0.5,2). Based on this dataset, we generated 20,000 MCMC samples (after initial burn-in) of component means, component proportions, and the unequal component variance. The MCMC samples are generated by Gibbs sampler with the priors given by Phillips and Smith (1996) and Richardson and Green (1997). That is to assume

$$\boldsymbol{\pi} \sim D(\delta, \delta), \mu_j \sim N(\xi, \kappa^{-1}), \sigma_j^{-2} \sim \Gamma(\alpha, \beta), j = 1, 2 \ ,$$

where $D(\cdot)$ is the Dirichlet distribution and $\Gamma(\alpha, \beta)$ is the gamma distribution with mean α/β and variance α/β^2. Following the suggestion of Richardson and Green (1997), we let $\delta = 1$, ξ equal the sample mean of the observations, $\kappa = 1/R^2$, and $\alpha = 2$, where R is the range of the observations. Richardson and Green (1997) introduced an additional hierarchical model by allowing β to follow a gamma distribution to reduce the influence of β on the posterior distribution of the number of components. Here we fix all the parameters in the prior distribution like Phillips and Smith (1996) and set $\beta = R^2/200$. Similar priors are used for the other two examples.

The upper labeling credibility level α^* was 98.5%, and so almost all the samples likely have the ideal HPD labels. In this example, all the 20,000 samples, except for six, converged to the maximal modes. The other six samples converged to the same minor mode. Hence, almost all the samples can be labeled directly by the converged maximal modes. The minor mode was labeled by the distance criterion (2).

The runtimes for KL, NORMLH, and PM(ECM) were 66, 0.2, and 25 seconds, respectively. The total numbers of different labels between (OC, KL, NORMLH) and PM(ECM) were 757, 212, and 0, respectively. On the subset above the labeling credibility c^*, the number of disagreements was 663, 203, and 0, respectively. (Note that NORMLH and PM(ECM) had the same labels in this example. Using the PM(ECM) labels as the initial values, NORMLH converged with just one iteration. If using the OC labels as the initial values, NORMLH converged in three iterations and the runtime was 3 seconds.)

Because there are only two components, we can easily use some parameter plots to check where the labeling differences occurred. Figure 1 gives the plots of $\sigma_1 - \sigma_2$ versus π_1 for different labeling methods. Figure 2 gives the plots of $\sigma_1 - \sigma_2$ versus $\mu_1 - \mu_2$. Note that the gray and black points represent the two permuted images of the labeled parameter values. From these plots, one can see that there are indeed relatively tight clusters around each posterior mode, and that OC and KL did not accurately recover these labels. The NORMLH and PM(ECM) methods clustered the two groups more naturally.

Example 3.2. We generated 400 data points from the eight-component normal mixture $\sum_{j=1}^{8} 0.125N(\mu_j, 1)$, where $\mu_j = 3(j-1)$. This is an example where, because of the constant weight parameters and variance parameters, we would expect the OC method to be very effective. The large number of components, however, will make labeling computationally difficult for relabeling algorithms. Based on this dataset, we generated 5,000 MCMC samples of component means, component proportions, and the equal component variance. (The personal computer used for the simulation did not have enough memory for the KL algorithm when we tried to label a large set of 10,000 samples, largely because of the storage of classification probabilities. Stephens (2000) did provide some alternative online versions for the KL algorithm.)

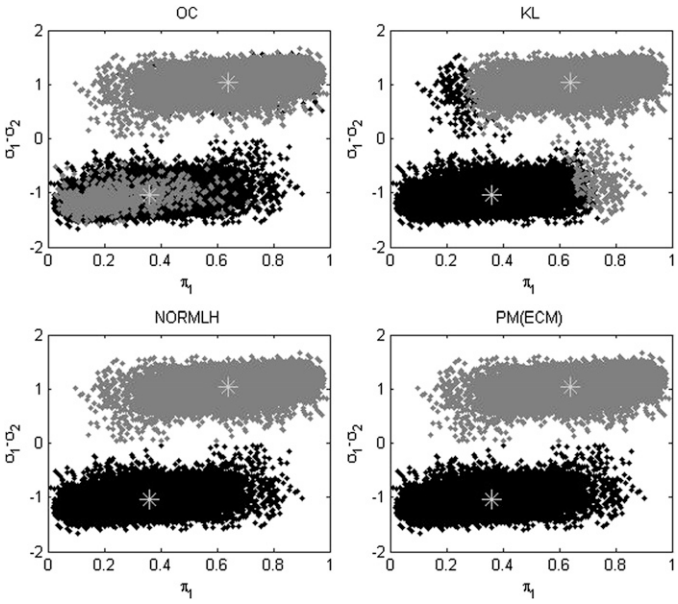

Figure 1. Plots of $\sigma_1 - \sigma_2$ versus π_1 for the four labeling methods in Example 3.1. The black points represent one set of labels and the gray points are the permuted samples. The star points are the posterior modes.

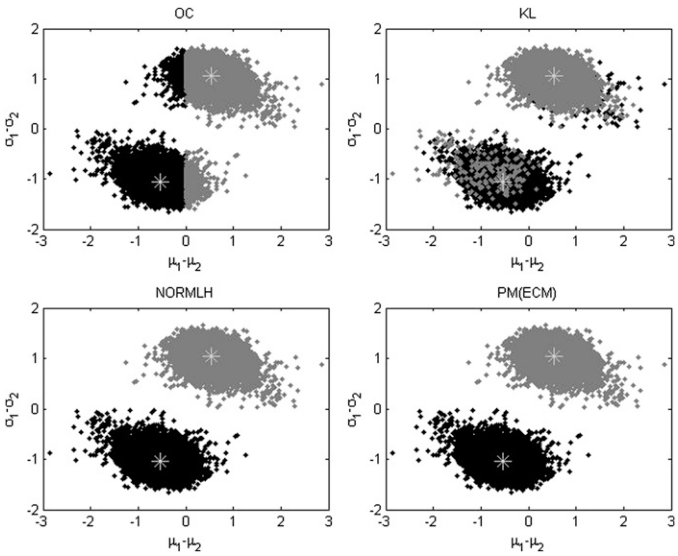

Figure 2. Plots of $\sigma_1 - \sigma_2$ versus $\mu_1 - \mu_2$ for the four labeling methods in Example 3.1.

The upper labeling credibility level was 58%, so at least 42% of the samples do not have ideal HPD labels. By our standards, the labels on these points are somewhat arbitrary (i.e., there is no natural/ideal way to label them). In this example, 95% of samples converged to the maximal modes. The other 5% of samples converged to four minor modes, three of which were degenerate modes.

The total number of different labels between (OC, KL, NORMLH) and PM(ECM) was 109, 365, and 142, respectively. In this example, all four methods had identical labels on the subset above the labeling credibility c^*. Hence, all four methods recovered the likely HPD labels well, and the labeling differences occurred for non-HPD labels.

The runtimes for KL, NORMLH, and PM(ECM) were 7.8629×10^4, 3.4394×10^4, and 79 seconds, respectively. (The runtime for KL and NORMLH is based on one initialization.) We can see that PM(ECM) was much faster than the other two methods because KL and NORMLH methods required one to compare $8! = 40{,}320$ permutations in each iteration. From this example, we can see that if the number of components is large, PM(ECM) will be much faster than KL and NORMLH.

It is difficult to compare different labeling methods graphically when the number of components is large. Instead, we provide the trace plots and the marginal density plots to illustrate the success of PM(ECM). (The OC, KL, and NORMLH methods had similar visual results for those plots.) Figure 3 F3 provides the trace plots for the original Gibbs samples and the labeled samples by PM(ECM). Figure 4 provides the estimated F4 marginal posterior density plots for the original samples and the labeled samples by PM(ECM). From these figures, we can see that PM(ECM) successfully removed the label switching in the raw output of the Gibbs sampler at a considerably lower computational expense than all but OC.

3.2 Real Data Application

We consider the acidity dataset (Crawford et al. 1992; Crawford 1994). The data are shown in Figure 5. The obser- F5 vations are the logarithms of an acidity index measured in a sample of 155 lakes in north–central Wisconsin. This dataset has been analyzed as a mixture of Gaussian distributions by Crawford et al. (1992), Crawford (1994), and Richardson and

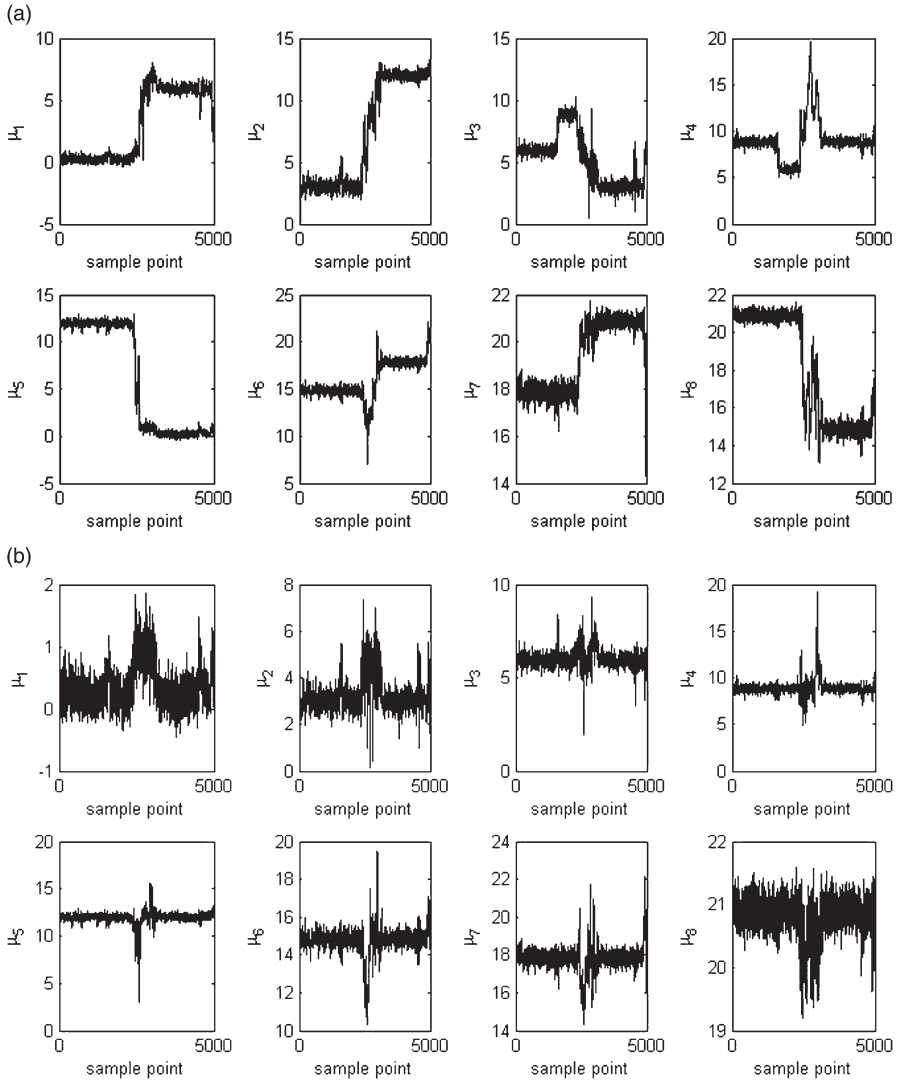

Figure 3. Trace plots of the Gibbs samples of component means for Example 3.2. (a) Original Gibbs samples. (b) Labeled samples by PM(ECM).

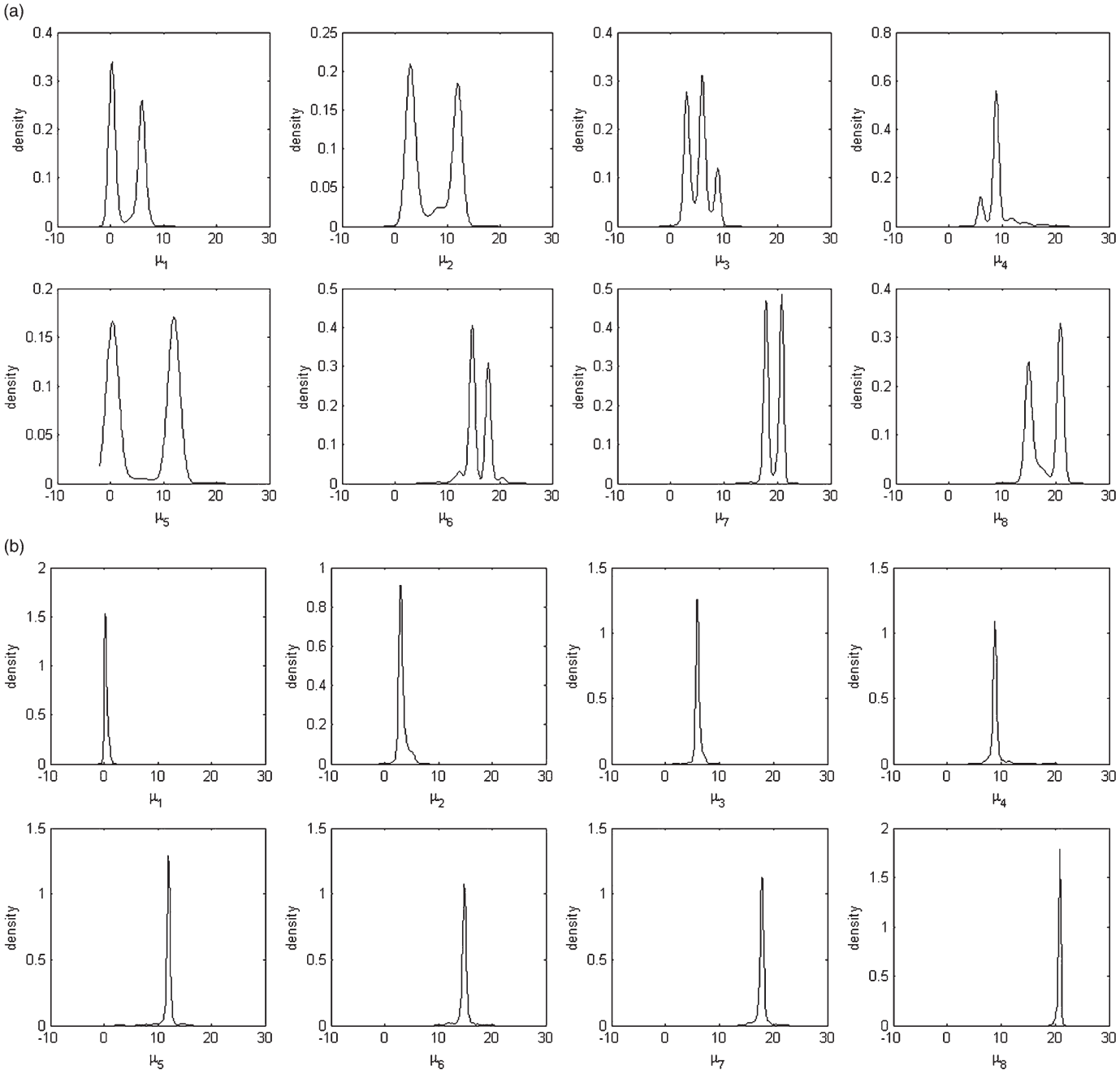

Figure 4. Plots of estimated marginal posterior densities of component means for Example 3.2 based on original Gibbs samples (a) and labeled samples by PM(ECM) (b).

Green (1997). Based on the result of Richardson and Green (1997), the posterior for three components is largest. Hence, we fit this dataset by a three-component normal mixture. We postprocessed the 20,000 Gibbs samples by the OC, KL, NORMLH, and PM(ECM) labeling methods.

The upper labeling credibility level was 71%. In this example, around 91% of the 20,000 samples converged to the maximal modes. The other 9% of the samples converged to four minor modes. The runtimes for KL, NORMLH, and PM(ECM) were 41, 5, and 60 seconds, respectively. The total numbers of different labels between (OC, KL, NORMLH) and PM(ECM) were 103, 527, and 127, respectively. On the set of samples with posterior probability more than c^*, the number of disagreements was 4, 105, and 9, respectively. Hence, both OC and NORMLH, but not the KL algorithm, recovered the likely HPD labels almost as well as PM(ECM) in this example.

Figure 6 shows the plots of $\sigma_2 - \sigma_3$ versus $\mu_2 - \mu_3$ and its permutation image, between the second and third components, for all the labeled samples. Figure 7 shows the similar plots but only for the labeled samples with posterior larger than c^*. Note that, unlike the two-component case, the points in the plots are not the same for all the methods. Whenever the labeling difference for one sample involves the label of the first component, the two permuted points, between the second and third components, in the plots will be different for different methods. (For example, supposing $(\mu_1^*, \mu_2^*, \mu_3^*, \sigma_1^*, \sigma_2^*, \sigma_3^*)$ is the labeled

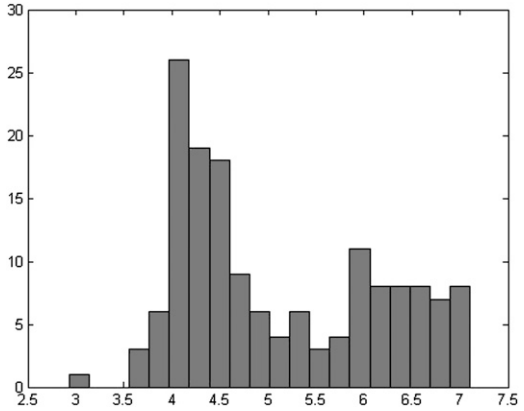

Figure 5. Histogram of acidity data. The number of bins used is 20.

sample used by one method and $(\mu_3^*, \mu_2^*, \mu_1^*, \sigma_3^*, \sigma_2^*, \sigma_1^*)$ is the corresponding labeled sample used by another method, the two permuted points, between the second and third components, in the plots will be $(\mu_2^* - \mu_3^*, \sigma_2^* - \sigma_3^*)$ and $(\mu_3^* - \mu_2^*, \sigma_3^* - \sigma_2^*)$ for the first method, and $(\mu_2^* - \mu_1^*, \sigma_2^* - \sigma_1^*)$ and $(\mu_1^* - \mu_2^*, \sigma_1^* - \sigma_2^*)$ for the second method.) From Figures 6 and 7, one can see that KL did not cluster the parameter points as well as the other three methods. Based on Figure 7, one can also see that all the methods, except for the KL algorithm, recovered the likely HPD labels pretty well.

4. DISCUSSION

In this article, we proposed two labeling methods: PM(ALG) and NORMLH. The PM(ALG) method uses each MCMC sample as the starting point for an ascending algorithm (such as the ECM(BM) algorithm introduced in Section 2.2) and assigns the label based on the mode to which the algorithm converges. Using one of the maximal modes as the reference mode, all other permuted maximal modes have clear labels. For the minor modes, we proposed to label them by comparing the

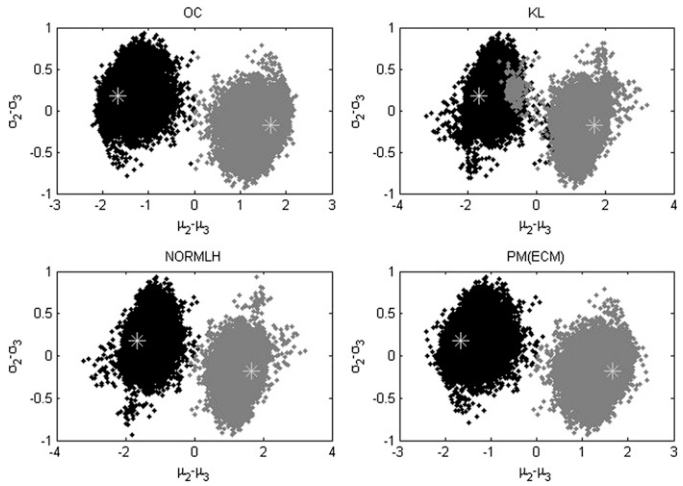

Figure 6. Plots of $\sigma_2 - \sigma_3$ versus $\mu_2 - \mu_3$ for the acidity data. The black points represent one set of labels and the gray points are the permuted samples between the second and the third components. The star points are the posterior modes.

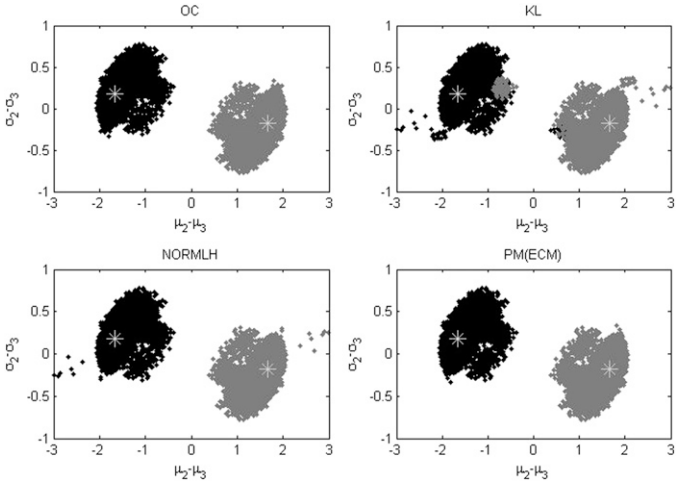

Figure 7. Plots of $\sigma_2 - \sigma_3$ versus $\mu_2 - \mu_3$ for the samples with posterior higher than c^* for the acidity data.

minor modes with the reference mode based on the Euclidean distance (2) or the Kullback-Leibler divergence criteria (3).

If the converged mode is a degenerate mode, meaning it corresponds to a mixture with at least one component less than the fitted model, then, as a referee pointed out, there really is no sensible labeling by PM(ALG) (or any other labeling method). We do not find this disturbing, because all sample points that converge to a degenerate mode do not have HPD labels, and so there is no single natural way to label them.

Because of the ascending property of ALG, the PM(ALG) method will reproduce the HPD labels in major modal groups. Hence, the PM(ALG) method creates a natural and intuitive partition of the parameter space into labeled regions.

There are several other nice properties of the PM(ALG) method. First, unlike a typical relabeling algorithm, the PM(ALG) method gives an answer that does not depend on a set of initial labels, the choice of which can change the labeling. Second, the PM(ALG) method is an online algorithm and it can do labeling along with the MCMC sampling process. Hence, the storage requirements are reduced. Finally, the PM(ALG) method does not require one to compare $m!$ permutations when doing labeling except for the minor modes. This property can make PM(ALG) much faster than some other labeling methods when m is large, as shown in Example 2 in Section 3.

There are also some possible ways to improve further the computation speed of PM(ALG). One way is to find a faster ascending algorithm to find the local posterior mode. Another possibility, when used in batch mode, is first to cluster the samples by a method like K-means with large number of clusters K. Then, by assuming that the samples within each cluster have the same labels, we only need to find one converged mode for each cluster.

If a hierarchical Bayesian model is used, the marginal prior and the posterior distribution of $\boldsymbol{\theta}$ contains the integration with respect to the random prior parameters. If there is a closed form for the marginal prior and hence the posterior distribution, we can still use the ECM(BM) to find the posterior modes. However, if there is no closed form for the posterior distribution, the ECM(BM) cannot be used directly. One could, however, use the ECM(BM) on the full posterior including hyperparameters.

A second possibility, provided that the likelihood function dominates the prior distribution (the prior is relatively flat or the sample size is large), is to use the likelihood function to approximate the posterior. Then one could use the usual mixture EM algorithm to assign the labels based on the modes of the likelihood itself.

Our second proposed labeling method NORMLH is often computationally easy and fast when the number of components is not large. However this method might be nearly as slow as the KL algorithm when the number of components is large. In our examples, it performed somewhat better than the alternatives at recreating the PM(ECM) labels.

Finally, we introduced a new reliability measure called the ''labeling credibility level'' and an easy-to-compute approximation called the ''upper credibility level.'' This approximates the proportion of the samples that will have ideal HPD labels and measures how difficult the labeling problem is. It is estimated by the proportion of the samples with posterior larger than the maximum posterior of the degenerate modes. It can be used, as in the examples, to examine the clustering of the HPD regions.

[Received December 2007. Revised December 2008.]

REFERENCES

Böhning, D. (1999), *Computer-Assisted Analysis of Mixtures and Applications*, Boca Raton, FL: Chapman and Hall/CRC.

Celeux, G. (1997), ''Discussion of 'On Bayesian Analysis of Mixtures with an Unknown Number of Components,' by S. Richardson and P. J. Green,'' *Journal of the Royal Statistical Society,* Ser. A, B59, 775–776.

———. (1998), ''Bayesian Inference for Mixtures: The Label Switching Problem,'' in *Compstat 98—Proceedings in Computational Statistics*, eds. R. Payne and P. J. Green, Heidelberg: Physica, pp. 227–232.

Celeux, G., Hurn, M., and Robert, C. P. (2000), ''Computational and Inferential Difficulties with Mixture Posterior Distributions,'' *Journal of the American Statistical Association,* 95, 957–970.

Chung, H., Loken, E., and Schafer, J. L. (2004), ''Difficulties in Drawing Inferences with Finite-Mixture Models: A Simple Example with a Simple Solution,'' *The American Statistician,* 58, 152–158.

Corana, A., Marchesi, M., Martini, C., and Ridella, S. (1987), ''Minimizing Multimodal Functions of Continuous Variables with the Simulated Annealing Algorithm,'' *ACM Transactions on Mathematical Software,* 13, 262–280.

Crawford, S. L. (1994), ''An Application of the Laplace Method to Finite Mixture Distributions,'' *Journal of the American Statistical Association,* 89, 259–267.

Crawford, S. L., Degroot, M. H., Kadane, J. B., and Small, M. J. (1992), ''Modeling Lake-Chemistry Distributions: Approximate Bayesian Methods for Estimating a Finite-Mixture Model,'' *Technometrics,* 34, 441–453.

Davis, L. (ed.) (1991), *Handbook of Genetic Algorithms*, New York: Van Nostrand Reinhold.

Dellaportas, P., Stephens, D. A., Smith, A. F. M., and Guttman, I. (1996), ''A Comparative Study of Perinatal Mortality Using a Two-Component Mixture Model,'' in *Bayesian Biostatistics*, eds. D. A. Berry and D. K. Stangl, New York: Dekker, pp. 601–616.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), ''Maximum Likelihood from Incomplete Data via the EM Algorithm'' (with discussion), *Journal of the Royal Statistical Society,* Ser. A, B39, 1–38.

Diebolt, J., and Robert, C. P. (1994), ''Estimation of Finite Mixture Distributions through Bayesian Sampling,'' *Journal of the Royal Statistical Society,* Ser. A, B56, 363–375.

Frühwirth-Schnatter, S. (2001), ''Markov Chain Monte Carlo Estimation of Classical and Dynamic Switching and Mixture Models,'' *Journal of the American Statistical Association,* 96, 194–209.

———. (2006), *Finite Mixture and Markov Switching Models*, Berlin: Springer.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, MA: Addison Wesley.

Holland, J. H. (1975), *Adaption in Natural and Artifical Systems*, Ann Arbor, MI: University of Michigan Press.

Hurn, M., Justel, A., and Robert, C. P. (2003), ''Estimating Mixtures of Regressions,'' *Journal of Computational and Graphical Statistics,* 12, 55–79.

Ingber, L., and Rosen, B. (1992), ''Genetic Algorithms and Very Fast Simulated Reannealing: A Comparison,'' *Mathematical and Computer Modelling,* 16, 87–100.

Jasra, A., Holmes, C. C., and Stephens, D. A. (2005), ''Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modeling,'' *Statistical Science,* 20, 50–67.

Li, J., Ray, S., and Lindsay, B. G. (2007), ''A Nonparametric Statistical Approach to Clustering via Mode Identification,'' *Journal of Machine Learning Research,* 8, 1687–1723.

Lindsay, B. G. (1995), *Mixture Models: Theory, Geometry, and Applications*, NSF-CBMS Regional Conference Series in Probability and Statistics v 5, Hayward, CA: Institute of Mathematical Statistics.

Marin, J.-M., Mengersen, K. L., and Robert, C. P. (2005), ''Bayesian Modelling and Inference on Mixtures of Distributions,'' in *Handbook of Statistics 25*, eds. D. Dey and C. R. Rao, Amsterdam: North-Holland, pp. 459–507.

McLachlan, G. (1982). ''The Classification and Mixture Maximum Likelihood Approaches to Cluster Analysis,'' in *Handbook of Statistics* (vol. 2), eds. P. R. Krishnaiah and L. N. Kanal, Amsterdam: North-Holland, pp. 199–208.

McLachlan, G. J., and Basford, K. E. (1988), *Mixture Models: Inference and Applications to Clustering*, New York: Marcel Dekker.

McLachlan, G. J., and Peel, D. (2000), *Finite Mixture Models*, New York: Wiley.

Meng, X.-L., and Rubin, D. B. (1993), ''Maximum Likelihood Estimation via the ECM Algorithm: A General Framework,'' *Biometrika,* 80, 267–278.

Mengersen, K. (2009), *Bayesian Analysis of Mixtures: Foundations and Applications*, New York: Wiley.

Phillips, D. B., and Smith, A. F. M. (1996), ''Bayesian Model Comparison via Jump Diffusion,'' in *Markov Chain Monte Carlo in Practice*, eds., W. R. Gilks, S. T. Richardson, and D. J. Spiegelhalter, London: Chapman and Hall, pp. 215–239.

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,* Ser. A, B59, 731–792.

Stephens, M. (1997a), ''Bayesian Methods for Mixtures of Normal Distributions,'' Ph.D. dissertation, University of Oxford, Dept. of Statistics.

———. (1997b), ''Discussion of 'On Bayesian Analysis of Mixtures with an Unknown Number of Components,' by S. Richardson and P. J. Green,'' *Journal of the Royal Statistical Society,* Ser. A, B59, 768–769.

———. (2000), ''Dealing with Label Switching in Mixture Models,'' *Journal of the Royal Statistical Society,* Ser. A, B62, 795–809.

Symons, M. (1981), ''Clustering Criteria and Multivariate Normal Mixtures,'' *Biometrics,* 37, 35–43.

Walker, A. M. (1969), ''On the Asymptotic Behaviour of Posterior Distributions,'' *Journal of the Royal Statistical Society,* Ser. A, B31, 80–88.

Yao, W. (2007), ''On Using Mixtures and Modes of Mixtures in Data Analysis,'' Ph.D. dissertation, The Pennsylvania State University, Dept. of Statistics.