An online Bayesian mixture labelling method by minimizing deviance of classification probabilities to reference labels

Weixin Yao and Longhai Li

Published Version Information

Citation: Yao, W., & Li, L. (2014). An online Bayesian mixture labelling method by minimizing deviance of classification probabilities to reference labels. Journal of Statistical Computation and Simulation, 84(2), 310-323.

Digital Object Identifier (DOI): doi:10.1080/00949655.2012.707201

Publisher's Link:
http://www.tandfonline.com/doi/full/10.1080/00949655.2012.707201#.Uxo0vj9dXL8

An Online Bayesian Mixture Labeling Method by Minimizing Deviance of Classification Probabilities to Reference Labels

Weixin Yao and Longhai Li

Abstract

Solving label switching is crucial for interpreting the results of fitting Bayesian mixture models. The label switching originates from the invariance of posterior distribution to permutation of component labels. As a result, the component labels in Markov chain simulation may switch to another equivalent permutation, and the marginal posterior distribution associated with all labels may be similar and useless for inferring quantities relating to each individual component. In this article, we propose a new simple labeling method by minimizing the deviance of the class probabilities to a fixed reference labels. The reference labels can be chosen before running MCMC using optimization methods, such as EM algorithms, and therefore the new labeling method can be implemented by an online algorithm, which can reduce the storage requirements and save much computation time. Using the Acid data set and Galaxy data set, we demonstrate the success of the proposed labeling method for removing the labeling switching in the raw MCMC samples.

[1]Weixin Yao is Assistant Professor, Department of Statistics, Kansas State University, Manhattan, Kansas 66506, U.S.A. Email: wxyao@ksu.edu. Longhai Li is Assistant Professor, Department of Mathematics and Statistics, University of Saskatchewan, Saskatoon, Saskatchewan, S7N5E6, Canada. Email: longhai@math.usask.ca. The research of Longhai Li is supported by fundings from Natural Sciences and Engineering Research Council of Canada, and Canadian Foundation of Innovations.

Key words: Bayesian mixtures; Label switching; Markov chain Monte Carlo; Mixture models; Relabeling.

1 Introduction

Label switching is one of the fundamental issues for Bayesian mixtures if our interests are quantities relating to each individual component. It occurs due to the invariance of the posterior distribution to the permuation of the component labels. Many methods have been proposed to solve the label switching problem. One simple way is to use an explicit parameter identifiability constraint so that only one permutation can satisfy it. See Diebolt and Robert (1994); Dellaportas et al. (1996); Richardson and Green (1997). One problem with the identifiability constraint labeling is that the results are sensitive to the choice of constraint, especially for multivariate problems. Celeux et al. (2000) demonstrated that different order constraints may generate markedly different results; it is difficult to anticipate the overall effect. Moreover, many choices of identifiability constraint do not completely remove the symmetry of the posterior distribution. As a result, label switching problem may remain after imposing an identifiability constraint, see the example by Stephens (2000). Celeux (1998) and Stephens (2000) proposed a relabeling algorithm, which is based on minimizing a Monte Carlo risk. Yao and Lindsay (2009) proposed to label the samples based on the posterior modes and an ascent algorithm (PM(ALG)). PM(ALG) uses each Markov chain Monte Carlo (MCMC) sample as the starting point in an ascending algorithm, and labels the sample based on the mode of the posterior to which it converges. Then PM(ALG) assumes that the samples converged to the same mode have the same labels. Sperrin, Jaki, and Wit (2010) developed several probabilistic relabeling algorithms by extending the probabilistic relabeling of Jasra (2005).

Papastamoulis and Iliopoulos (2010) proposed an artificial allocations based solution to the label switching problem. Yao (2012a) proposed to assign the probabilities for each

possible labels by fitting a mixture model to the permutation symmetric posterior. Other labeling methods include, for example, Celeux et al. (2000); Fruhwirth (2001); Hurn et al. (2003); Chung et al. (2004); Marin et al. (2005); Geweke (2007); Grun and Leisch (2009); Cron and West (2011); Yao (2012b). Jasra et al. (2005) provided a good review about the existing methods to solve the label switching problem in Bayesian mixture modeling.

In this article, we propose a new alternative labeling method by minimizing the deviance of the class probabilities to a fixed reference labels. The reference labels may be chosen before running MCMC using optimization methods, such as EM algorithms, and therefore the new labeling method can be implemented by an online algorithm, i.e., the output of MCMC samples will have been automatically relabeled along with simulating MCMC samples. Such online algorithms have advantages in storage and computation time. More specifically, our method can be implemented during MCMC simulation by making use of the classification probability matrices that are needed for MCMC simulation itself. As consequence, our method neither requires storing the classification probability matrices, nor requires recomputing them after MCMC simulation. The reference labels can also be chosen after MCMC sampling by alternating two steps of finding the reference labels and relabeling MCMC samples, as in method proposed by Stephens (2000).

The rest of the paper is organized as follows. Section 2 introduces our new labeling method. In Section 3, we use a simulation study and two real data applications to demonstrate the success of the proposed labeling method. We summarize our proposed labeling method in Section 4.

2 New Method

Generally, the mixture model has the density

$$p(x;\boldsymbol{\theta}) = \pi_1 f(x;\lambda_1) + \pi_2 f(x;\lambda_2) + \cdots + \pi_m f(x;\lambda_m), \tag{2.1}$$

where $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_m, \lambda_1, \ldots, \lambda_m)$, $f(\cdot)$ is the component density, λ_j is the component specific parameter, which can be scalar or vector and π_j is the proportion of the jth component in the whole population with $\sum_{i=1}^{m} \pi_i = 1$. If $\mathbf{x} = (x_1, \ldots, x_n)$ are independent observations from the $m-$component mixture model (2.1), the likelihood of $\boldsymbol{\theta}$ given $\mathbf{x}$ is

$$L(\boldsymbol{\theta}; \mathbf{x}) = \prod_{i=1}^{n} \{\pi_1 f(x_i; \lambda_1) + \pi_2 f(x_i; \lambda_2) + \cdots + \pi_m f(x_i; \lambda_m)\} . \tag{2.2}$$

A permutation $\boldsymbol{\omega} = (\boldsymbol{\omega}(1), \ldots, \boldsymbol{\omega}(m))$ of the component labels $\{1, \ldots, m\}$ defines a corresponding permutation of the parameter vector $\boldsymbol{\theta}$ by

$$\boldsymbol{\theta}^{\boldsymbol{\omega}} = (\pi^{\boldsymbol{\omega}}, \lambda^{\boldsymbol{\omega}}) = (\pi_{\boldsymbol{\omega}(1)}, \ldots, \pi_{\boldsymbol{\omega}(m)}, \lambda_{\boldsymbol{\omega}(1)}, \ldots, \lambda_{\boldsymbol{\omega}(m)}).$$

A special feature of mixture model is that the likelihood function $L(\boldsymbol{\theta}^{\boldsymbol{\omega}}; \mathbf{x})$ is exactly the same as $L(\boldsymbol{\theta}; \mathbf{x})$ for any permutation $\boldsymbol{\omega}$.

For Bayesian mixtures, if the prior distributions for model parameters are symmetric for all components then the posterior distribution for the parameters will be also symmetric and thus invariant to permutations in the labeling of the component parameters. The marginal posterior distributions for the parameters will be identical for all mixture components. Then the posterior means of each component are the same and are thus poor estimates of these parameters. Similar problem will occur when we try to estimate quantities relating to individual components of the mixture such as predictive component densities, marginal classification probabilities. So in Bayesian analysis, after we get a sequence of simulated values $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$ from the posterior distribution of $\boldsymbol{\theta}$ given $Y = y$ using MCMC sampling methods, we must first find permuations $\{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N\}$ such that $\boldsymbol{\theta}_1^{\boldsymbol{\omega}_1}, \ldots, \boldsymbol{\theta}_N^{\boldsymbol{\omega}_N}$ have the same label meaning, then we can use the labeled samples to do Bayesian analysis. Many methods (as reviewed in Section 1) have been proposed to find $\{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N\}$ for relabeling MCMC samples. In this article, we introduce a new method. Note that, even though we introduce and demonstrate our method in the post-MCMC context, the striking feature of our new method is that it

can be implemented during MCMC simulation for fitting Bayesian mixture model.

Given observations $\mathbf{x} = (x_1, \ldots, x_n)$, suppose we have found a set of reference component labels for each observation x_i represented by $\mathbf{Z} = \{Z_{ij}, i = 1, \ldots, n, j = 1, \ldots, m\}$, where

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

We will talk about how to find the reference label $\mathbf{Z}$ later. Let $\boldsymbol{\theta} = (\lambda_1, \ldots, \lambda_m, \pi_1, \ldots, \pi_m)$. Our new method for finding a permutation $\boldsymbol{\omega}$ for relabeling a Markov chain sample $\boldsymbol{\theta}$ (note that we drop MCMC index since our method will be implemented during MCMC simulation for each sample of parameters) is to minimize the sum of minus log classification probabilities of $\mathbf{Z}$ given by $\boldsymbol{\theta}^{\boldsymbol{\omega}}$ with respect to $\boldsymbol{\omega}$:

$$\ell(\boldsymbol{\omega}; \mathbf{Z}, \boldsymbol{\theta}) \;\; = \;\; -\sum_{i=1}^{n}\sum_{j=1}^{m} Z_{ij} \log p_{ij}(\boldsymbol{\theta}^{\boldsymbol{\omega}}), \tag{2.3}$$

where $p_{ij}(\boldsymbol{\theta}^{\boldsymbol{\omega}})$ is the classification probability that the ith observation belongs to jth component based on relabeled parameter $\boldsymbol{\theta}^{\boldsymbol{\omega}}$:

$$p_{ij}(\boldsymbol{\theta}^{\boldsymbol{\omega}}) = \frac{\pi_{\boldsymbol{\omega}}(j) f(x_i; \lambda_{\boldsymbol{\omega}}(j))}{p(x_i; \theta^{\boldsymbol{\omega}})} = p_{i,\boldsymbol{\omega}(j)}(\boldsymbol{\theta}). \tag{2.4}$$

The objective function $\ell(\boldsymbol{\omega}; \mathbf{Z}, \mathbf{x})$ in (2.3) can be also considered as the Kullback-Leibler divergence if we consider Z_{ij} as the true classification probability and $p_{ij}(\boldsymbol{\theta})$ as the estimated classification. One may notice that the loss function in (2.3) has some similarity to Kullback-Leibler divergence algorithm proposed by Stephens (2000), which basically switches the position of Z_{ij} and p_{ij} and thus considers $p_{ij}(\boldsymbol{\theta})$ as the true classification probability, in addition, Stephens (2000) used soft reference classification probabilities to replace Z_{ij}. The performance of using (2.3) or Kullback-Leibler divergence is therefore expected to be similar. However, we notice some advantages of using (2.3). First, computing (2.3) is faster than

KL divergence, since we can save computing the product of Z_{ij} and p_{ij} once we know $Z_{ij} = 0$. When m is large, the saving of computing time of using (2.3) compared to using KL divergence is substantial. Note, however, similar to the general relabeling algorithm (Celeux, 1998; Stephens, 2000), when m is large, we need to compare $m!$ permutations in order to minimize (2.3) for each MCMC sample.

Next, we will discuss some other interpretations of (2.3), which will make this loss function more easily understood. Note that $2\ell(\boldsymbol{\omega}; \mathbf{Z}, \mathbf{x})$ is often called deviance of classification probabilities $p_{ij}(\boldsymbol{\theta}^{\boldsymbol{\omega}}), i = 1, \ldots, n, j = 1, \ldots, m$ to the reference labels $\mathbf{Z}$ in the literature of generalized linear models, if $\mathbf{Z}$ is the true response values and $p_{ij}(\boldsymbol{\theta}^{\boldsymbol{\omega}})$ is the predictive probabilities based on a generalized linear model. In words, by minimizing $\ell(\boldsymbol{\omega}; \mathbf{Z}, \mathbf{x})$ with respect to $\boldsymbol{\omega}$ we will find the optimal permuation $\boldsymbol{\omega}$ for a Markov chain sample $\boldsymbol{\theta}$ such that the corresponding classification probabilities can best explain the reference label $\mathbf{Z}$. It is crucial to note that our method uses the differences of the whole probability density functions $f(x; \lambda_j)$ and mixture proportion π_j of all mixture components $j = 1, \ldots, m$ in relabeling $\boldsymbol{\theta}$ rather than the values of a single or an arbitrarily chosen subset of parameters in $\boldsymbol{\theta}$. Our method therefore works well in the situations where any single parameter in $\boldsymbol{\theta}$ cannot clearly distinguish all components but the density functions given the whole set of parameters are clearly different for components.

The proposed objective function (2.3) has another nice interpretation based on complete posterior distribution. Let $\pi(\boldsymbol{\theta})$ be the prior for $\boldsymbol{\theta}$. Then the posterior for complete data $(\mathbf{x}, \mathbf{Z})$ is

$$p_c(\boldsymbol{\theta}; \mathbf{x}, \mathbf{Z}) = \pi(\boldsymbol{\theta}) p(\mathbf{x}, \mathbf{Z} \mid \boldsymbol{\theta}) = \pi(\boldsymbol{\theta}) \prod_{i=1}^{n} \prod_{j=1}^{m} \{\pi_j f(x_i; \lambda_j)\}^{Z_{ij}} .$$

Note that the above complete posterior is *not* invariant to the component labels and thus can be used to do labeling. Given the reference label $\mathbf{Z}$, it is natural to do labeling for $\boldsymbol{\theta}$ by

maximizing the log complete posterior

$$\log p_c(\boldsymbol{\theta}^{\boldsymbol{\omega}}; \mathbf{x}, \mathbf{Z}) = \log \pi(\boldsymbol{\theta}^{\boldsymbol{\omega}}) + \sum_{i=1}^{n}\sum_{j=1}^{m} \left[Z_{ij} \log(\pi_j^{\boldsymbol{\omega}} f(x_i; \lambda_j^{\boldsymbol{\omega}})) \right] \tag{2.5}$$

with respect to $(\mathbf{Z}, \boldsymbol{\omega})$, where $\pi_j^{\boldsymbol{\omega}} = \pi_{\boldsymbol{\omega}(j)}$, and $\lambda_j^{\boldsymbol{\omega}} = \lambda_{\boldsymbol{\omega}(j)}$.

Note that

$$\begin{aligned}
&\log p_c(\boldsymbol{\theta}^{\boldsymbol{\omega}}; \mathbf{x}, \mathbf{Z}) \\
=& \log \pi(\boldsymbol{\theta}^{\boldsymbol{\omega}}) + \sum_{i=1}^{n}\sum_{j=1}^{m} \left[Z_{ij} \log\{\pi_j^{\boldsymbol{\omega}} f(x_i; \lambda_j^{\boldsymbol{\omega}}) / p(x_i; \theta^{\boldsymbol{\omega}})\} \right] + \sum_{i=1}^{n}\sum_{j=1}^{m} Z_{ij} \log p(x_i; \theta^{\boldsymbol{\omega}}) \\
=& \log \pi(\boldsymbol{\theta}^{\boldsymbol{\omega}}) + \sum_{i=1}^{n}\sum_{j=1}^{m} Z_{ij} \log p_{ij}(\boldsymbol{\theta}^{\boldsymbol{\omega}}) + \sum_{i=1}^{n} \log p(x_i; \theta^{\boldsymbol{\omega}}),
\end{aligned} \tag{2.6}$$

where $p(x_i; \boldsymbol{\theta}^{\boldsymbol{\omega}}) = \sum_{j=1}^{m} \pi_j^{\boldsymbol{\omega}} f(x_i; \lambda_j^{\boldsymbol{\omega}})$. Notice that the first and third terms of (2.6) are invariant to the permutation of $\boldsymbol{\omega}$. Therefore, maximizing (2.6) is equivalent to maximizing the second term of (2.6), which is equivalent to minimizing (2.3).

There are many methods for finding reference labels $\mathbf{Z} = (Z_{ij}, i = 1, \ldots, n, j = 1, \ldots, m)$. One simple method is to find the posterior mode, say $\hat{\boldsymbol{\theta}}$, and the corresponding classification probabilities, say $p_{ij}(\hat{\boldsymbol{\theta}})$. Then the *hard labels* Z_{ij} can be estimated by maximizing the classification probabilities over all components, i.e.,

$$Z_{ij} = \begin{cases} 1, & \text{if } p_{ij}(\hat{\boldsymbol{\theta}}) \geq p_{il}(\hat{\boldsymbol{\theta}}) \text{ for all } l = 1, \ldots, m; \\ 0, & \text{otherwise.} \end{cases} . \tag{2.7}$$

In addition, one might also directly use the *soft labels* $p_{ij}(\hat{\boldsymbol{\theta}})$ for Z_{ij} in (2.3). Based on our experience, the soft labels and the hard labels usually provide similar labeling results.

To find the posterior mode, one might simply calculate the posterior for each MCMC sample $\boldsymbol{\theta}_t, t = 1, \ldots, N$ and then use the sample that has the largest posterior to approximate the posterior mode. Note, however, this method can only be performed offline. In addition,

Yao and Lindsay (2009) also proposed the ECM algorithm for Bayesian mixtures to find the posterior mode. Suppose that there exists a partition of $\boldsymbol{\theta} = (\boldsymbol{\theta}_{(1)}, \ldots, \boldsymbol{\theta}_{(p)})$ such that all the conditional complete posterior distributions $\{p(\boldsymbol{\theta}_{(i)} \mid ...), 1 \leq i \leq p\}$ can be easily found, where $\boldsymbol{\theta}_{(i)}$ can be scalar or vector and |... denotes conditioning on all other parameters and the latent variable $\mathbf{Z}$. In the E step, the ECM algorithm calculates the classification probabilities, p_{ij}, for each observation. In the M step, the ECM algorithm maximizes the conditional complete posterior distribution $p(\boldsymbol{\theta}_{(i)} \mid ...), 1 \leq i \leq p$, sequentially with the latent variable Z_{ij} replaced by the classification probability p_{ij}. The ECM iterates the above E step and M step until convergence. Similar to the general optimization algorithm, ECM algorithm might find different modes from different starting values. Therefore, it is prudent to run the ECM algorithm from several starting values (say ten) and use the converged mode that has the largest posterior. One advantage of the ECM algorithm is that it can be implemented before the sampling process of MCMC algorithm. To report the results on the examples in Section 3, we used this ECM algorithm to find the posterior mode and the corresponding reference labels $\mathbf{Z}$.

Therefore, the above proposed labeling procedure can be summarized as follows.

Algorithm 2.1. *Step 1: Find the posterior mode and the corresponding reference labels* $\boldsymbol{Z} = (Z_{ij}, i = 1, \ldots, n, j = 1, \ldots, m)$.
Step 2: For each MCMC sample $\boldsymbol{\theta}_t$*, choose* $\boldsymbol{\omega}_t$ *to minimize* $\ell(\omega_t; \boldsymbol{Z}, \boldsymbol{\theta}_t)$ *of (2.3).*

One main advantage of the above algorithm is that it can be implemented along with MCMC simulation. The reference label $\mathbf{Z}$ is first found before the MCMC simulation and will then be used along with simulating MCMC, which saves storage. Therefore the above algorithm is an online algorithm — the output of MCMC samples will have been automatically relabeled. We will use the above online algorithm in Section 3, although the computation is implemented in the post-MCMC context

Following Stephens (2000), we can also find the reference label $\mathbf{Z}$ after simulating MCMC,

by simultaneously finding $\mathbf{Z}$ and $\mathbf{\Omega} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)$ that minimize a Monte Carlo risk:

$$R(\mathbf{Z}, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N) = \sum_{t=1}^{N} \ell(\boldsymbol{\omega}_t; \mathbf{Z}, \boldsymbol{\theta}), \tag{2.8}$$

where ℓ is given by (2.3). We propose the following algorithm to minimize (2.8):

Algorithm 2.2. *Starting with some initial values for $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N$ (set by order constraint labels for example), iterate the following two steps until a fixed point is reached.*
Step 1: Given $\boldsymbol{Z}$, for each t, choose $\boldsymbol{\omega}_t$ to minimize $\ell(\omega_t; \boldsymbol{Z}, \boldsymbol{\theta}_t)$. In other words, relabel all Markov chain iterations such that the relabeled samples have the same label meaning as $\boldsymbol{Z}$.
Step 2: Estimate $\boldsymbol{Z}$ by

$$Z_{ij} = \begin{cases} 1, & \text{if } \sum_{t=1}^{N} \log p_{ij}(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t}) > \sum_{t=1}^{N} \log p_{il}(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t}) \text{ for all } l \neq j; \\ 0, & o.w. \end{cases},$$

where $i = 1, \ldots, n, j = 1, \ldots, m$.

Note that, similar to Stephens (2000), the Algorithm 2.2 can only be implemented after saving all MCMC samples and thus is not an online algorithm. However, one advantage of Algorithm 2.2, compared to Algorithm 2.1, is that it doesn't require to find the posterior mode. Based on empirical experience, Algorithm 2.1 and 2.2 usually provide similar labeling results.

Theorem 2.1. *The Algorithm 2.2 must converge and monotonically decrease the objective function (2.8).*

Based on Theorem 2.1, the objective function (2.8) will decrease after each iteration of Algorithm 2.2. Therefore, the Algorithm 2.2 will converge. Note, however, the Algorithm 2.2 depends on the initial labels and is only guaranteed to converge locally. Therefore, it is prudent to run the Algorithm 2.2 from several choices of initial labels and to choose the

labeling results that correspond to the best local optimum found. One way to choose the initial labels is to set the permutations at random for each sample.

3 Examples

In this section, we use both simulation study and real data applications to demonstrate the success of the proposed labeling method for removing the labeling switching in the raw MCMC samples. In addition, we also add Stephens (2000)'s KL algorithm and Yao and Lindsay (2009)'s PM(ECM) and NORMLH for comparison. Note that the runtime for the NORMLH and KL algorithm depends on the number of starting points (i.e. the initial labels for all samples), we only report the runtime of NORMLH and KL when using the PM(ECM) labels as the initial labels. All the computations were done in Matlab 7.0 using a personal desktop with Intel Core 2 Quad CPU 2.40GHz.

Example 1: We generated 400 data points from 0.3N(0,1)+0.7N(2,1). Based on this data set, we generated 20,000 MCMC samples, after initial burn-in, of component means, component proportions, and the equal 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

$$\boldsymbol{\pi} \sim D(\delta, \delta, \delta),\ \mu_j \sim N(\xi, \kappa^{-1}),\ \sigma_j^{-2} \sim \Gamma(\alpha, \beta), \quad j = 1, 2, 3\,,$$

where $D(\cdot)$ is Dirichlet distribution and $\Gamma(\alpha, \beta)$ is gamma distribution with mean α/β and variance α/β^2, $\delta = 1$, ξ equal the sample mean of the observations, κ equal $1/R^2$, $\alpha = 2$, and $\beta = R^2/200$, where R is the range of the observations. Similar priors are used for other examples.

We post processed the 20,000 Gibbs samples by Stephens (2000)'s KL algorithm, Yao and Lindsay (2009)'s PM(ECM) and NORMLH, and the proposed new labeling method.

The runtime for KL, NORMLH, PM(ECM), and the new method were 43, 1, 53, and 29 seconds, respectively. Therefore, NORMLH is computationally much faster than the other three methods. In addition, the proposed new method is also faster than KL and PM(ECM).

Since there are only two components, similar to Yao and Lindsay (2009), we can use the parameter plots to check where the labeling differences occurred. Figure 1 gives the plots of $\mu_1 - \mu_2$ vs. π_1 for different labeling methods. The grey and black points represent the two permuted images of the labeled parameter values. The star points are the posterior modes. From these plots, we can see that all four methods correctly recover the two symmetric modal regions that are around two symmetric posterior modes. The labeling difference for the four methods only occurred to the samples corresponding to the near degenerate mixture models which have close component means. Note that when the mixture components are close, the component labels are not well defined and thus the found labels will be very sensitive to the labeling methods.

Example 2 (Galaxy Data): The galaxy data (Roeder, 1990) consists of the velocities (in thousands of kilometers per second) of 82 distant galaxies diverging from our own galaxy. They are sampled from six well-separated conic sections of the corona borealis. A histogram of the 82 data points is shown in Figure 2. This data set has been analyzed by many researchers, for example, Crawford (1994); Chib (1995); Carlin and Chib (1995); Escobar (1995); Phillips and Smith (1996); Richardson and Green (1997). Stephens (2000) also used this data set to explain the label switching problem. We fit this data by six-component normal mixture. The MCMC samples are generated by Gibbs sampler with the same priors used in Example 1.

We post processed the 20,000 Gibbs samples by Stephens (2000)'s KL algorithm, Yao and Lindsay (2009)'s PM(ECM) and NORMLH, and the proposed new labeling method. The runtime for KL, NORMLH, PM(ECM), and the new method were 2486, 1094, 48, and 186 seconds, respectively. Therefore, PM(ECM) is the fastest since it doesn't require to compare $m!$ permutations. In addition, the new method is also faster than KL and NORMLH since

our online algorithm avoids the iteration of two steps of relabeling MCMC samples and finding reference.

As Yao and Lindsay (2009) argued it is difficult to use the similar parameter plots in Example 1 to compare different labeling methods when the number of components is larger than two. Here, we provide the trace plots and the marginal density plots to illustrate the success of the new labeling method. Figure 3 and 4 are the trace plots and the estimated marginal posterior density plots, respectively, for the original samples and the labeled samples by new method. In this example, Stephens (2000)'s KL algorithm, Yao and Lindsay (2009)'s PM(ECM) and NORMLH provided similar visual results for those two plots. From Figure 3 and 4, we can see that the new labeling method successfully removed the label switching in the raw output of the Gibbs sampler.

Example 3 (Acidity Data): We consider the acidity data set (Crawford et al., 1992; Crawford, 1994). The observations are the logarithms of an acidity index measured in a sample of 155 lakes in north-central Wisconsin. The data are shown in Figure 5. Crawford et al. (1992), Crawford (1994), and Richardson and Green (1997) have used a mixture of Gaussian distributions to analyze this data set. Here, we fit this data set by a three-component normal mixture based on the result of Richardson and Green (1997). The MCMC samples are generated by Gibbs sampler with the same priors used in Example 1.

We post processed the 20,000 Gibbs samples by Stephens (2000)'s KL algorithm, Yao and Lindsay (2009)'s PM(ECM) and NORMLH, and the proposed new labeling method. The runtime for KL, NORMLH, PM(ECM), and new method were 58, 8, 49, and 13 seconds, respectively. Therefore, NORMLH and the new method are faster than KL and PM(ECM).

Figure 6 and 7 are the trace plots and the estimated marginal posterior density plots, respectively, for the original samples and the labeled samples by new method. Stephens (2000)'s KL algorithm, Yao and Lindsay (2009)'s PM(ECM) and NORMLH had similar visual results for those plots. From Figures 6(a) and 7(a), we can see that the label switching occurred in the raw samples and the marginal density plots display the multi-modality.

Based on Figures 6(b) and 7(b), we can see that the new labeling method successfully removed the label switching in the raw output of the Gibbs sampler.

Remarks: Note that KL algorithm and NORMLH are not online algorithms and require the iterations to minimize the corresponding criteria after all the samples are collected. In addition, in order to find the global minimum, KL algorithm and NORMLH require to start from different initial values, which also increase the computation time. The new method based on Algorithm 2.1 and PM(ECM) are online algorithms, which can reduce the storage space. In addition, neither the new method nor PM(ECM) require the iterations or starting from several initial labels, which can save much computation time. Note however PM(ECM) requires to run the ECM algorithm N times with each of the MCMC sample as the initial value. Based on three examples considered in this section, we can see that the proposed new method is always computationally faster than KL; and is faster than PM(ECM) when m is not large. However when m is large PM(ECM) is much faster than the other three methods, since it doesn't require to compare $m!$ permutations while all other three methods do.

4 Summary

Label switching has been a long standing problem for Bayesian mixtures. In this paper, we proposed a new alternative labeling method by minimizing deviance of classification probabilities to reference labels. The new labeling method also has a nice interpretation based on the complete posterior likelihood. After finding the reference labels, the new method can be implemented without saving all MCMC samples and classification probabilities, i.e, the output of MCMC samples will have been automatically relabeled along with simulating MCMC samples. Therefore, the new method is an on online algorithm, which can reduce much storage requirements and speed the computation. The examples in Section 3 demonstrate the success of the new method in removing the label switching in the raw MCMC

samples. Based on our empirical studies, the new method has similar labeling results to Stephens(2000)'s KL algorithm but run faster than KL. Note, however, given $\mathbf{Z}$, the Algorithm 2.1 and 2.2 require to compare $m!$ permutations in order to minimize (2.8). Therefore, similar to the relabeling algorithm (Celeux 1998 and Stephens 2000), the computation of the new method is expensive when m is very large. However, note that one may find a much faster optimization algorithm that avoids comparing all of these $m!$ permutations with risk of finding a local mode of the objective function (2.3). This is an area worth further research. In addition, note that in order to use the online Algorithm 2.1, we need to first find the posterior mode and the reference labels $\mathbf{Z}$ in advance. In some complicated models, it might be difficult to find the posterior mode. One way to solve such problem is to use the maximum likelihood estimate (MLE) to approximate the posterior mode, which is sensible when a relative noninformative priors are used.

References

Carlin, B. P. and Chib, S. (1995). *Bayesian model choice via Markov chain Monte Carlo methods. Journal of Royal Statistical Society*, Ser. B, 57, 473-484.

Celeux, G. (1998). Bayesian inference for mixtures: The label switching problem. In Compstat 98-*Proc. in Computational Statistics* (eds. R. Payne and P.J. Green), 227-232. Physica, Heidelberg.

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.

Chib, S. (1995). Marginal likelihood from the Gibbs output. *Journal of American Statistical Association*, 90, 1313-1321.

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 Statistican*, 58, 152-158.

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.

Crawford, S. L. (1994). An application of the Laplace method to finite mixture distributions. *Journal of the American Statistical Association*, 89, 259-267.

Cron A. J. and West, M. (2011). Efficient classification-based relabeling in mixture models. *The American Statistician*, 65, 16-20.

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) 601-616, Dekker, New York.

Diebolt, J. and Robert, C. P. (1994). Estimation of finite mixture distributions through Bayesian sampling. *Journal of Royal Statistical Society*, Ser. B, 56, 363-375.

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

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.

Grün, B. and Leisch, F. (2009). Dealing with label switching in mixture models under genuine multimodality. *Journal of Multivariate Analysis*, 100, 851-861.

Hurn, M., Justel, A., and Robert, C. P. (2003). Estimating mixtures of regressions. *Journal of Computational and Graphical Statistics*, 12, 55-79.

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.

Marin, J.-M., Mengersen, K. L. and Robert, C. P. (2005). Bayesian modelling and inference on mixtures of distributions. *Handbook of Statistics* 25 (eds. D. Dey and C.R. Rao), North-Holland, Amsterdam.

Phillips, D. B. and Smith, A. F. M. (1996). Bayesian model comparison via jump diffusion. *Makov Chain Monte Carlo in Practice*, ch. 13, 215-239, London: Chapman and Hall.

Papastamoulis, P. and Iliopoulos, G. (2010). An artificial allocations based solution to the label switching problem in Bayesian analysis of mixtures of distributions. *Journal of Computational and Graphical Statistics*, 19, 313-331.

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

Roeder, K. (1990). Density estimation with confidence sets exemplified by superclusters and voids in the galaxies. *Journal of American Statistical Association*, 85, 617-624.

Sperrin, M., Jaki, T., and Wit, E. (2010). Probabilistic relabeling strategies for the label switching problem in Bayesian mixture models. *Statistics and Computing*, 20, 357-366.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of Royal Statistical Society*, Ser. B, 62, 795-809.

Yao, W. (2012a). Model based labeling for mixture models. *Statistics and Computing*, 22, 337-347.

Yao, W. (2012b). Bayesian mixture labeling and clustering. *Communications in Statistics - Theory and Methods*, 41, 403-421.

Yao, W. and Lindsay, B. G. (2009). Bayesian mixture labeling by highest posterior density. *Journal of American Statistical Association*, 104, 758-767.

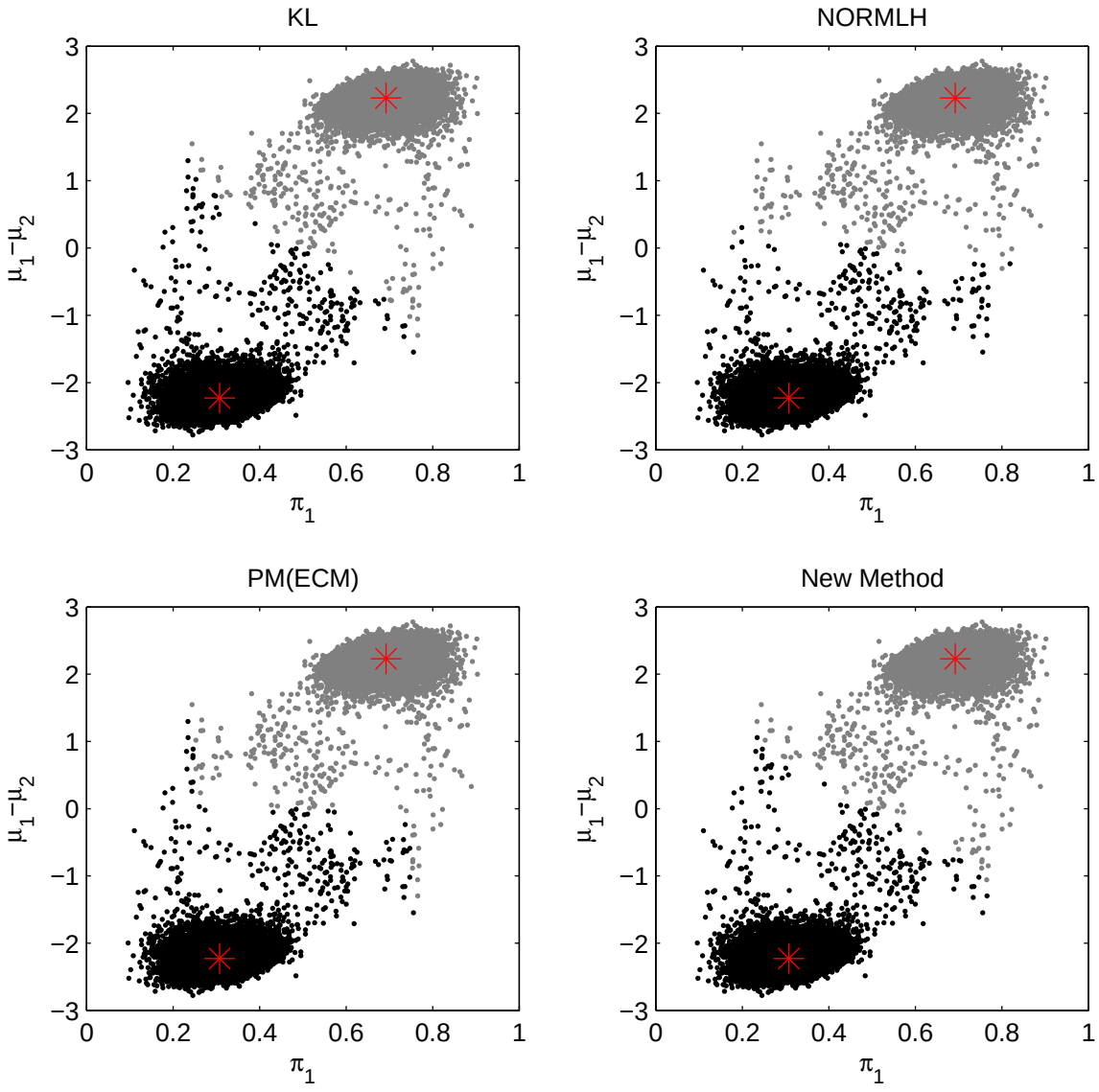


Figure 1: Plots of $\mu_1 - \mu_2$ vs. π_1 for the four labeling methods in Example 1. The black points represent one set of labels and the gray points are the permuted samples. The star points are the posterior modes.

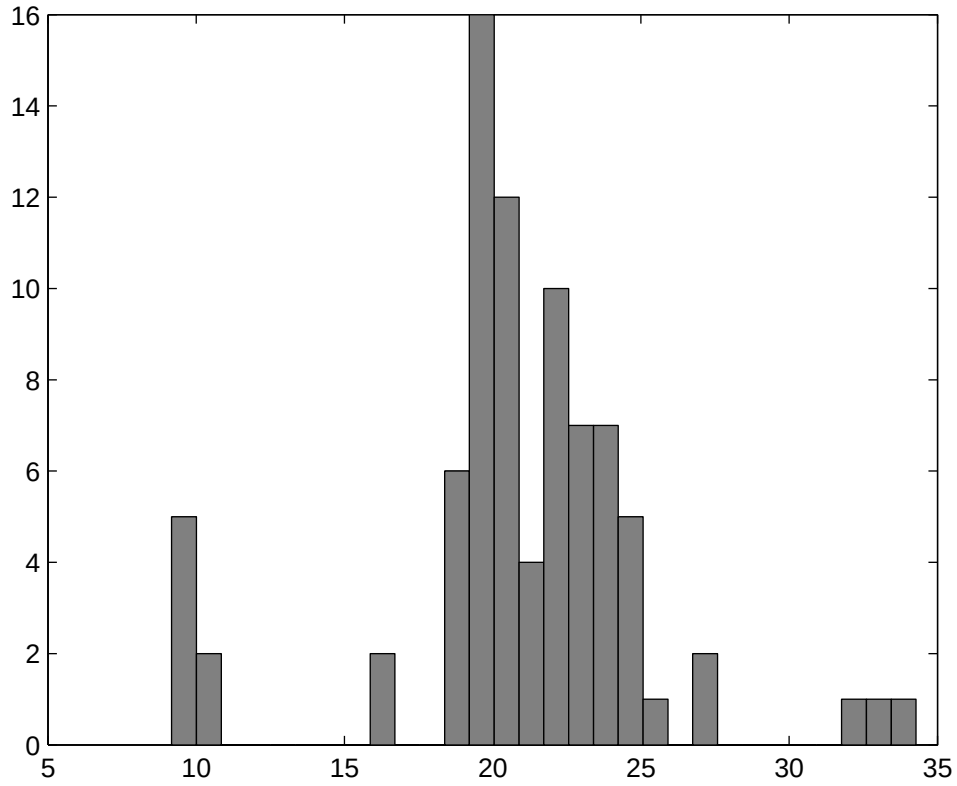
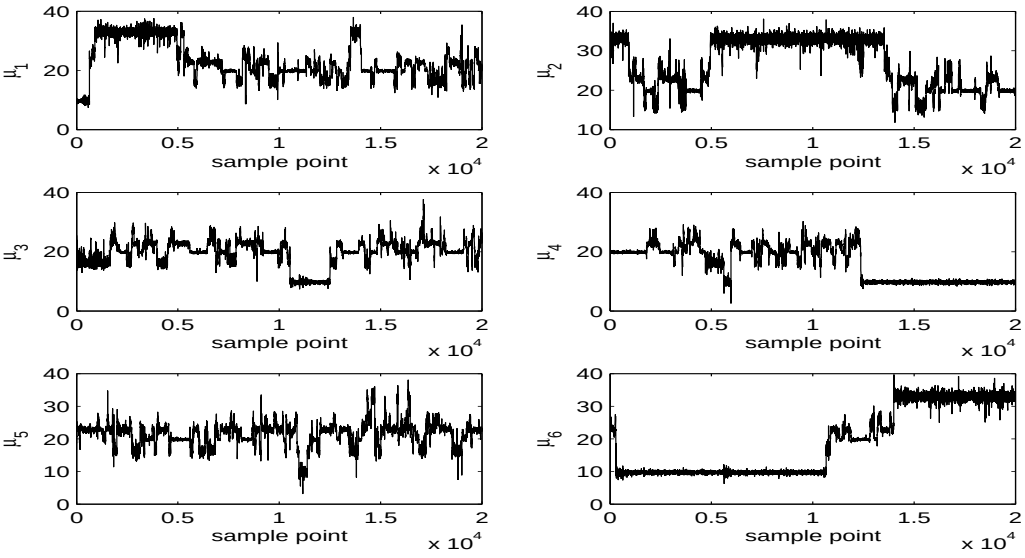
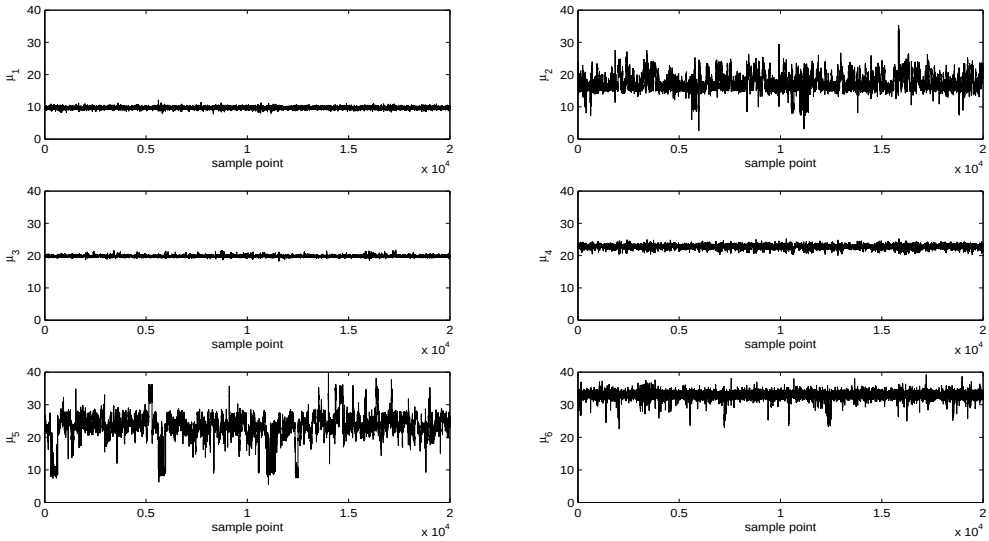


Figure 2: Histogram plot of galaxy data. The number of bins used is 30.

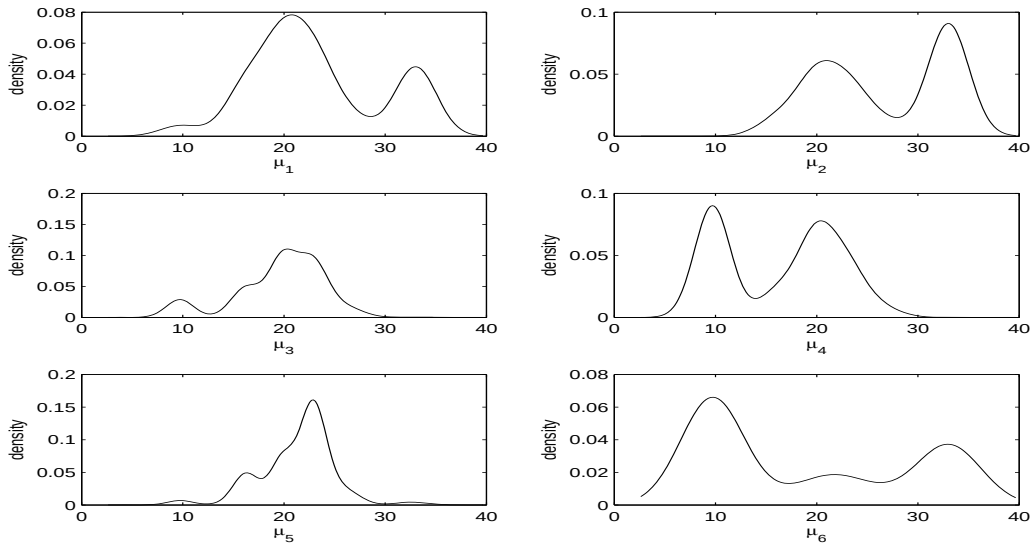


(a)

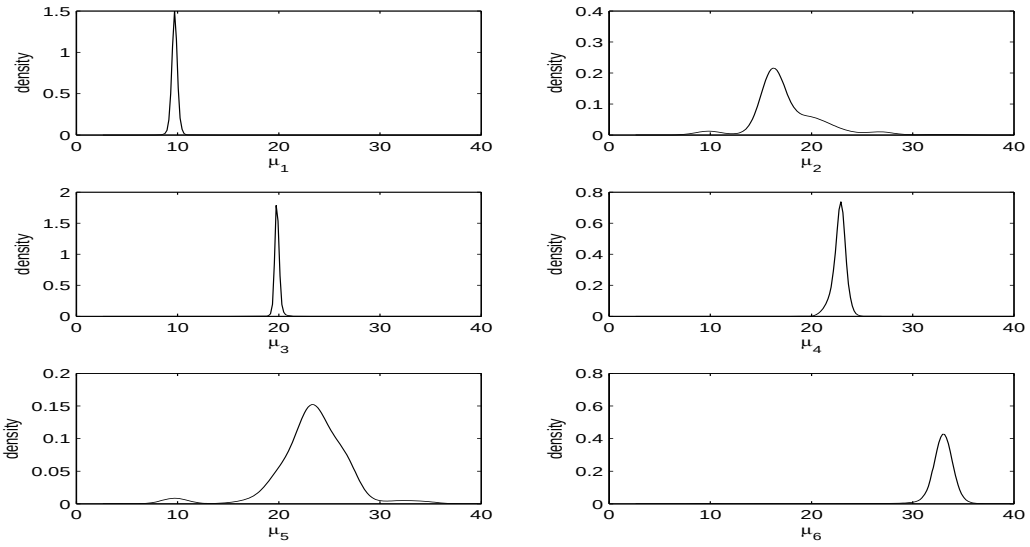


(b)

Figure 3: Trace plots of the Gibbs samples of component means for galaxy data: (a) original Gibbs samples; (b) labeled samples by the new method.



(a)



(b)

Figure 4: Plots of estimated marginal posterior densities of component means for galaxy data based on: (a) original Gibbs samples; (b) labeled samples by the new method.

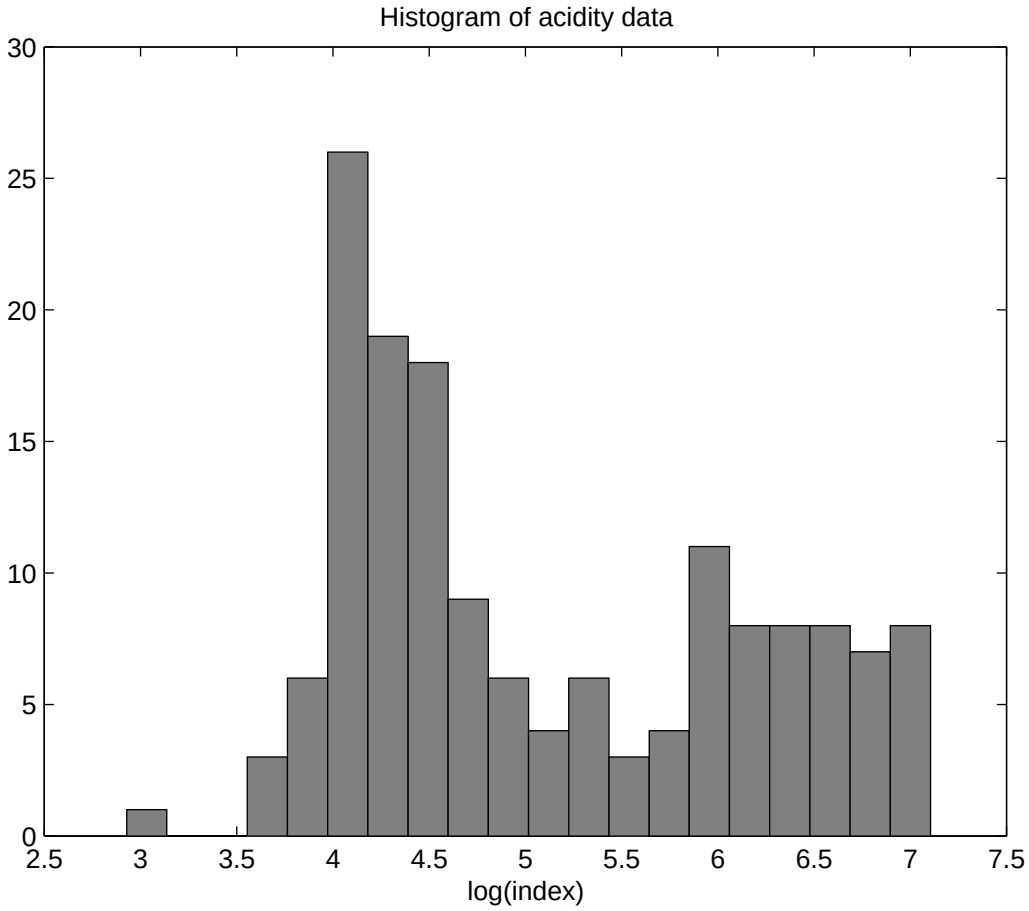
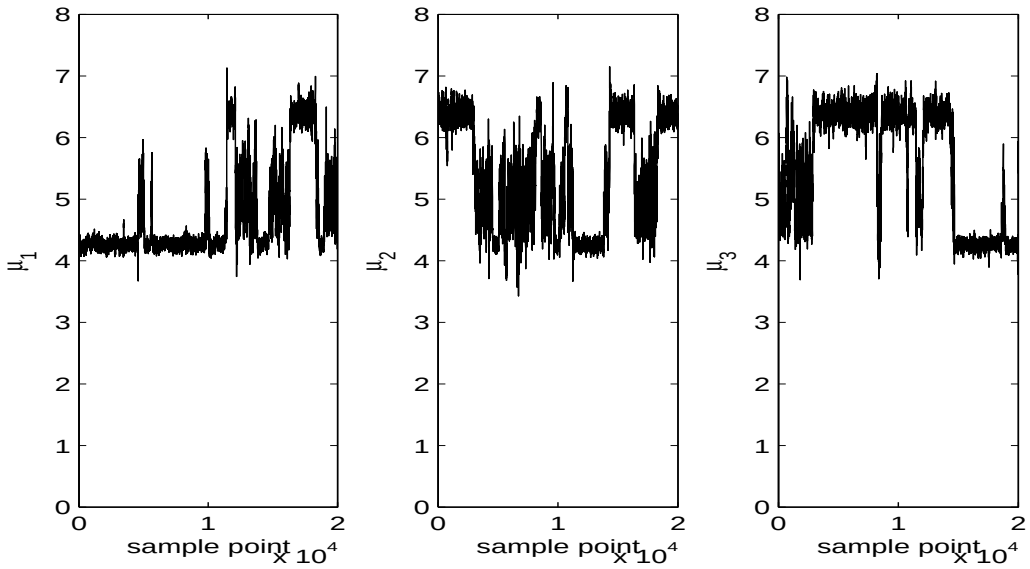
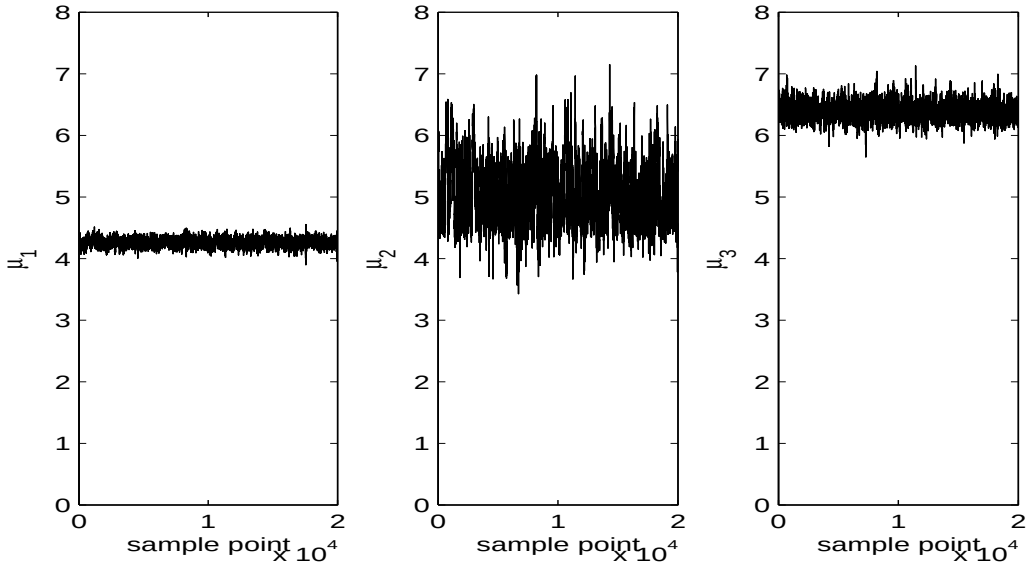


Figure 5: Histogram of acidity data. The number of bins used is 20.

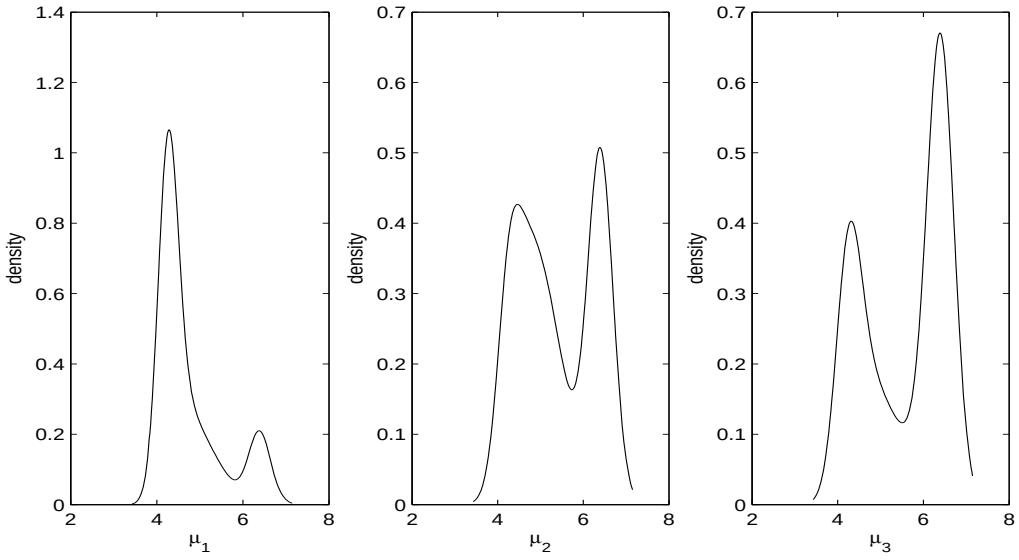


(a)

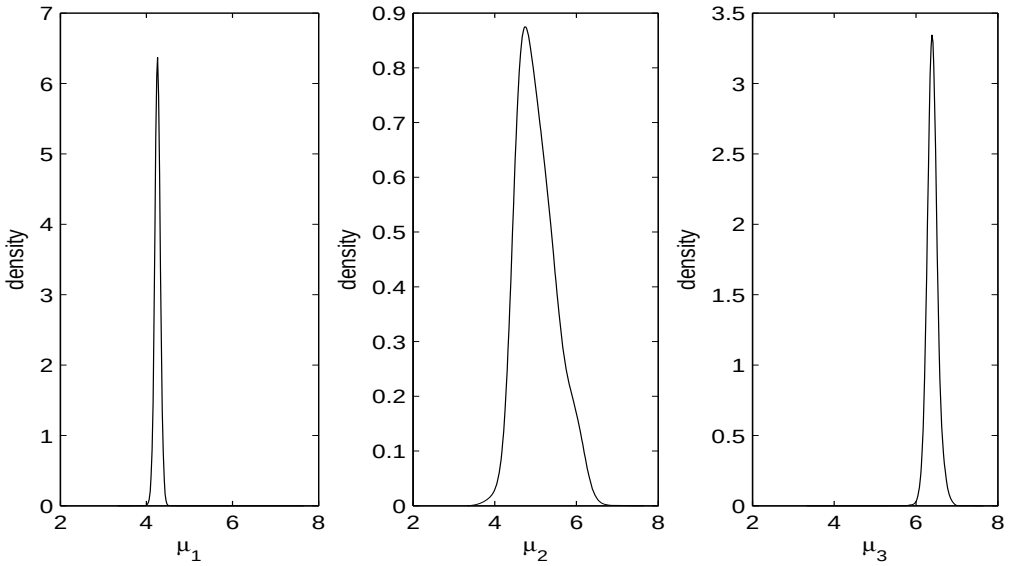


(b)

Figure 6: Trace plots of the Gibbs samples of component means for acidity data: (a) original Gibbs samples; (b) labeled samples by the new method.



(a)



(b)

Figure 7: Plots of estimated marginal posterior densities of component means for acidity data based on: (a) original Gibbs samples; (b) labeled samples by the new method.