UC Riverside
UC Riverside Previously Published Works

Title
Bayesian Mixture Labeling and Clustering

Permalink
https://escholarship.org/uc/item/1s24t36f

Journal
Communication in Statistics- Theory and Methods, 41(3)

ISSN
0361-0926

Author
Yao, Weixin

Publication Date
2012-02-01

DOI
10.1080/03610926.2010.526741

Peer reviewed

Bayesian Mixture Labeling and Clustering

Weixin Yao

Department of Statistics, Kansas State University, Manhattan, Kansas 66506, U.S.A.

wxyao@ksu.edu

Abstract

Label switching is one of the fundamental issues for Bayesian mixture modeling. It occurs due to the nonidentifiability of the components under symmetric priors. Without solving the label switching, the ergodic averages of component specific quantities will be identical and thus useless for inference relating to individual components, such as the posterior means, predictive component densities, and marginal classification probabilities. In this article, we establish the equivalence between the labeling and clustering and propose two simple clustering criteria to solve the label switching. The first method can be considered as an extension of K-means clustering. The second method is to find the labels by minimizing the volume of labeled samples and this method is invariant to the scale transformation of the parameters. Using a simulation example and two real data sets application, we demonstrate the success of our new methods in dealing with the label switching problem.

Key words: Bayesian mixtures; Clustering; K-means; Label switching; Markov chain Monte Carlo;

1 Introduction

Suppose $\mathbf{x} = (x_1, \ldots, x_n)$ are independent observations from a m-component mixture density

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

where $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_m, \lambda_1, \ldots, \lambda_m)$, $f(\cdot)$ is some parametric component density/mass function, $(\lambda_1, \ldots, \lambda_m)$ are the component specific parameters, which can be scalar or vector, and $(\pi_1, \ldots, \pi_m)$ are the mixture proportions with $\sum_{j=1}^{m} \pi_j = 1$. For a general introduction to mixture models, see **?**, **?**, **?**, and **?**. The likelihood for $\mathbf{x}$ is

$$L(\boldsymbol{\theta}; \mathbf{x}) = \prod_{i=1}^{n} \{\pi_1 f(x_i; \lambda_1) + \pi_2 f(x_i; \lambda_2) + \cdots + \pi_m f(x_i; \lambda_m)\}. \tag{1.1}$$

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

$$\boldsymbol{\theta}^{\boldsymbol{\omega}} = (\pi_{\boldsymbol{\omega}(1)}, \ldots, \pi_{\boldsymbol{\omega}(m)}, \lambda_{\boldsymbol{\omega}(1)}, \ldots, \lambda_{\boldsymbol{\omega}(m)}). \tag{1.2}$$

Then $L(\boldsymbol{\theta}^{\boldsymbol{\omega}}; \mathbf{x})$ will be numerically the same as $L(\boldsymbol{\theta}; \mathbf{x})$ for any permutation $\boldsymbol{\omega}$. Hence for Bayesian mixtures, if the prior is symmetric or permutation invariant for all components, the posterior distribution will be similarly symmetric and thus invariant to all the permutations of the component parameters. The marginal posterior distributions for the parameters will be also 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.

Many methods have been proposed to deal with the labeling problem in Bayesian analysis. The easiest way to solve the label switching is to impose constraints on the parameters. See **?**, **?**, and **?**. Another popular labeling method is relabeling algorithm (**??**), which is based on minimizing a Monte Carlo risk. **?** suggested a particular choice of loss function based on the Kullback-Liebler (KL) divergence. We will refer to this particular relabeling algorithm as *KL algorithm*. **?** imposed an asymmetric prior by fixing the label of a single observation. **?** proposed to label the samples based on the posterior modes they are associated with when they are used as the starting points for an ascending algorithm of the posterior. Other

labeling methods include, for example, **??????**. **?** provided a recent review of attempts to solve the label switching problem in mixture models.

In this article, we establish the equivalence between the labeling and clustering and propose two simple clustering criteria to solve the label switching. The first loss function is based on the euclidian distance between the sample and the center of the labeled samples. We will show that this labeling method is equivalent to applying the K-means clustering to all the permuted MCMC samples. If we only include one component specific parameter in the loss function (such as component means), then this method will be exactly the same as order constraint labeling on this component parameter. However, unlike the order constraint labeling, this method can simultaneously incorporate different component parameters together and can be easily extended to high dimensional case. In addition, this labeling method is computationally very fast as shown in our examples. The second method is to label the samples by minimizing the volume of the labeled samples. Here the volume is defined to be the determinant of covariance matrix. One nice property of this method is its invariance to the linear transformation of the parameters (changing both component means by a scale factor, both variances by a different one, for example). In addition, unlike some of the other labeling methods (such as the KL algorithm (**?**)), both of our proposed methods can be applied to solve label switching for frequentist mixtures. Using a simulation example and two real data sets application, we demonstrate the success of our new methods in dealing with the label switching problem.

The rest of the paper is organized as follows. Section **??** introduces our two new labeling methods. In Section **??**, we use one simulation example and two real data sets to compare our new labeling methods with two popular existing methods. We summarize our proposed labeling methods in Section **??**.

2 New labeling method

For Bayesian mixtures, after we get a sequence of MCMC samples, $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$, from the posterior distribution of $\boldsymbol{\theta}$, where N is the number of MCMC samples, the label switching problem is "solved" by finding the "right" labels $(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)$ for $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)$, i.e. relabeling the output of the sampler, such that $\boldsymbol{\theta}_1^{\boldsymbol{\omega}_1}, \ldots, \boldsymbol{\theta}_N^{\boldsymbol{\omega}_N}$ have the same label meaning. If we solve the label switching problem in this way, then we can use the labeled samples to estimate the quantities relating to individual components.

Due to the symmetry of the posterior, for a m component mixture model, the posterior distribution has $m!$ symmetric modal regions (each modal region is corresponding to one well labeled parameter space). Given the MCMC samples $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)$, the latent "true" labels $(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)$ are defined such that $\boldsymbol{\theta}_1^{\boldsymbol{\omega}_1}, \ldots, \boldsymbol{\theta}_N^{\boldsymbol{\omega}_N}$ are all in the same modal region and therefore have the same label meaning. The aim of labeling is to recover the latent labels $(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)$. Since each modal region defines a set of latent labels, there are essentially $m!$ sets of latent "true" labels and they are identifiable up to the same permutation. To do labeling, one only needs to recover one of the modal regions and the corresponding set of latent "true" labels.

From the asymptotic theory for the posterior distribution, see **?** and **?**[Sec 1.3, 3.3], we know that when sample size is large, the "correctly" labeled MCMC samples/the modal region should, approximately, follow the normal distribution. Therefore, it is reasonable to assume that the "right" labels $(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)$ will make the "size" of the cluster consisting of the labeled samples $(\boldsymbol{\theta}_1^{\boldsymbol{\omega}_1}, \ldots, \boldsymbol{\theta}_N^{\boldsymbol{\omega}_N})$ smaller than the "wrong" labels.

To explain the equivalence between labeling and clustering under a special setting, let $\boldsymbol{\Delta} = \{\boldsymbol{\theta}_t^{\boldsymbol{\omega}_{(j)}}, t = 1, \ldots, N\}$, where $\{\boldsymbol{\omega}_{(1)}, \ldots, \boldsymbol{\omega}_{(m!)}\}$ are the $m!$ permutations of $(1, \ldots, m)$. Note that $\boldsymbol{\Delta}$ includes both of the original samples and all of their permutations. Suppose one can find $m!$ tight clusters for $\boldsymbol{\Delta}$, 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

sample set and assume they are in the same modal region. So, the labeling problem is very similar to the clustering problem if only one permutation of each sample element $\boldsymbol{\theta}$ is allowed in each cluster.

Different clustering criteria lead to different labeling methods. In this section, we propose two simple clustering criteria to solve the label switching. The first method is to define the size of a cluster by the trace of their covariance matrix. The second method is to use the determinant of covariance matrix to define the size of a cluster.

2.1 The K-means method

We propose to find the labels $\boldsymbol{\Omega} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)$ together with their center $\boldsymbol{\theta}_c$ by minimizing the following loss functions

$$\ell(\boldsymbol{\theta}_c, \boldsymbol{\Omega}) = \mathrm{tr}\left(\sum_{t=1}^{N}(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)^T\right) = \sum_{t=1}^{N}(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)^T(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c), \qquad (2.1)$$

where $\boldsymbol{\Omega} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)$ and $\mathrm{tr}(A)$ is the trace of A. Since the loss function (**??**) is within cluster sum of squares, this labeling method can be also called *K-means method*.

When the labels $(\boldsymbol{\omega}_t, t = 1, \ldots, N)$ are fixed, the minimum of (**??**) over $\boldsymbol{\theta}_c$ occurs at the sample mean of $\{\boldsymbol{\theta}_1^{\boldsymbol{\omega}_1}, \ldots, \boldsymbol{\theta}_N^{\boldsymbol{\omega}_N}\}$. When $\boldsymbol{\theta}_c$ is fixed, the optimum over $\boldsymbol{\omega}_t, t = 1, \ldots, N$ can be done independently for all t. The algorithm to minimize (**??**) will be as follows.

Algorithm 2.1 *Labelling by Trace of Covariance (TRCOV)*

Starting with some initial values for $(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)$ (setting them based on the order constraint, for example), iterate the following steps until a fixed point is reached.

Step 1: Update $\boldsymbol{\theta}_c$ by the sample mean based on the current values $\{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N\}$,

$$\boldsymbol{\theta}_c = \frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t}.$$

Step 2: Given the current estimated center $\boldsymbol{\theta}_c$, $\{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N\}$ are updated by

$$\boldsymbol{\omega}_t = \arg \min_{\boldsymbol{\omega}} (\boldsymbol{\theta}_t^{\boldsymbol{\omega}} - \boldsymbol{\theta}_c)^T (\boldsymbol{\theta}_t^{\boldsymbol{\omega}} - \boldsymbol{\theta}_c) \, , t = 1, \ldots, N. \quad \square$$

The loss function $\ell(\boldsymbol{\theta}_c, \boldsymbol{\Omega})$ defined in (??) decreases after each of the above two steps. So this algorithm must converge.

Theorem 2.1 *The loss function $\ell(\boldsymbol{\theta}_c, \boldsymbol{\Omega})$ of (??) is decreased in each iteration of Algorithm 1 until a fixed point is reached.*

The proof of Theorem ?? is very simple and is omitted. Note that the minimum found by Algorithm ?? may only be a local minimum. To increase the chance of detecting the global minimum, one may run this algorithm starting from several initial values. In step 2, after each change of $\boldsymbol{\omega}_t$, one could also update $\boldsymbol{\theta}_c$, thereby increasing the speed of convergence but increasing complexity.

Notice that for $\boldsymbol{\theta}$ if the information from one component specific parameter dominates the other component parameters, then this labeling method will be very close to the order constraint labeling on this component specific parameter. Specially, if only m component specific parameters, say the m component means for one dimension data, are used in (??), then this labeling method will be exactly the same as the labeling by putting an order constraint on the component means. Because of this, the order constraint labeling can be considered as a special case of the TRCOV method. However, unlike the order constraint labeling, the TRCOV method can automatically make use of the most informative component parameters. In addition, the new method can simultaneous incorporate the information from different component parameters and can be easily extended to the high dimensional case.

2.2 The Determinant Based Loss

A drawback of TRCOV method is that the objective function (**??**) is not invariant to the scale transformation of the parameters. To solve the scale effect of the parameters, we propose another way to define the size of a cluster by the determinant of covariance matrix. We find the labels on the samples, along with the center, that minimize the determinant of covariance matrix

$$\mathcal{L}(\boldsymbol{\theta}_c, \boldsymbol{\Omega}) = \det \left(\sum_{t=1}^{N} (\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)^T \right) , \tag{2.2}$$

where $\boldsymbol{\Omega} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)$ and $\det(A)$ is the determinant of matrix A. The idea of determinant loss has also been used to create a robust estimator of the multivariate location and scatter (see the minimum covariance determinant (MCD) method of **?**) and do robust cluster analysis (see, for example, **?**).

One nice property about the determinant criterion (**??**) is that it is invariant to all permutation invariant linear transformations of the parameters (changing all component means by a linear transformation, all variances by a different one, for example). Therefore, the labels found by minimizing (**??**) will not be affected by such linear transformations.

Let $\tilde{\boldsymbol{\theta}}$ be the new parameter vector after a permutation invariant linear transformation of $\boldsymbol{\theta}$.

Theorem 2.2 *The determinant criteria of (**??**) is invariant, up to a multiplication constant, to all permutation invariant linear transformations of the parameters, i.e.*

$$\det \left(\sum_{t=1}^{N} (\tilde{\boldsymbol{\theta}}_t^{\boldsymbol{\omega}_t} - \tilde{\boldsymbol{\theta}}_c)(\tilde{\boldsymbol{\theta}}_t^{\boldsymbol{\omega}_t} - \tilde{\boldsymbol{\theta}}_c)^T \right) = k \det \left(\sum_{t=1}^{N} (\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)^T \right),$$

where the multiplication constant k does not depend on $\boldsymbol{\Omega} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)$. The proof of Theorem **??** and the constant k are given in the Appendix.

Based on the next theorem, we can know that when $(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)$ are fixed, the minimum of (**??**) over $\boldsymbol{\theta}_c$ occurs at the sample mean of $\{\boldsymbol{\theta}_1^{\boldsymbol{\omega}_1}, \ldots, \boldsymbol{\theta}_N^{\boldsymbol{\omega}_N}\}$.

Theorem 2.3 *Given* $(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)$, *let*

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

which is the sample mean of $\{\boldsymbol{\theta}_1^{\boldsymbol{\omega}_1}, \ldots, \boldsymbol{\theta}_N^{\boldsymbol{\omega}_N}\}$. *Then* $\bar{\boldsymbol{\theta}}$ *minimizes (??) over* $\boldsymbol{\theta}_c$.

The proof of Theorem **??** is given in the Appendix. Unlike the trace of covariance case, the minimum of (??) over $\boldsymbol{\Omega} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)$, given $\boldsymbol{\theta}_c$, can not truly be done independently for all t. Rather we need to optimize over $\boldsymbol{\omega}_t$ one t at a time while holding all others fixed.

Let

$$C_{<t>} = \sum_{l \neq t} (\boldsymbol{\theta}_l^{\boldsymbol{\omega}_l} - \boldsymbol{\theta}_c)(\boldsymbol{\theta}_l^{\boldsymbol{\omega}_l} - \boldsymbol{\theta}_c)^T. \tag{2.3}$$

Notice that the objection function $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\Omega})$ in (??) is

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\Omega}) &= \det\left(C_{<t>} + (\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)^T\right) \\
&= \det(C_{<t>}) \det\left[I + C_{<t>}^{-1/2}(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)^T C_{<t>}^{-1/2}\right] \\
&= \det(C_{<t>}) \left[1 + (\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)^T C_{<t>}^{-1}(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)\right] .
\end{aligned}
\tag{2.4}
$$

Thus to optimize over $\boldsymbol{\omega}_t$ for a particular t, other terms fixed, we just minimize

$$(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)^T C_{<t>}^{-1}(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c),$$

which is a weighted distance between $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_c$. *The leave-one out weight matrix* $C_{<t>}^{-1}$ *makes this labeling method invariant to the affine transformation of the component parameters.*

The algorithm to minimize (??) will be as follows.

Algorithm 2.2 *Labelling by Determinant of Covariance (DETCOV)*

Starting with some initial values for $(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)$ *(setting them based on the order constraint, for example), iterate the following two steps until a fixed point is reached.*

8

Step 1: Update $\boldsymbol{\theta}_c$ by the sample mean based on the current values $\{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N\}$,

$$\boldsymbol{\theta}_c = \frac{1}{N} \sum_{t=1}^{N} \boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} .$$

Step 2: For $t = 1, \ldots, N$, given the current estimated center $\boldsymbol{\theta}_c$, and $\{\boldsymbol{\omega}_l, l \neq t\}$, $\boldsymbol{\omega}_t$ are updated by

$$\boldsymbol{\omega}_t = \arg\min_{\boldsymbol{\omega}} (\boldsymbol{\theta}_t^{\boldsymbol{\omega}} - \boldsymbol{\theta}_c)^T C_{<t>}^{-1} (\boldsymbol{\theta}_t^{\boldsymbol{\omega}} - \boldsymbol{\theta}_c) . \quad \square \tag{2.5}$$

The above algorithm will monotonically decrease the objective function (??) after each step. So, the Algorithm 2 must converge.

Corollary 2.1 *The loss function $\mathcal{L}(\boldsymbol{\theta}_c, \boldsymbol{\Omega})$ of (??) will decrease after each iteration of Algorithm 2 until a fixed point is reached.*

The proof of Corrollary ?? follows directly from Theorem ?? and the result (??).

Note that in step 2, we need to calculate $C_{<t>}^{-1}$ for every t, which might be computationally expensive when the dimension of $\boldsymbol{\theta}$ is quite large. Based on the following result, we can greatly reduce the computation burden of $C_{<t>}^{-1}$. Its proof is given in the Appendix.

Theorem 2.4 *Let*

$$C = \sum_{t=1}^{N} (\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)^T, u_t = C^{-1/2}(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c), v_t = C_{<t>}^{-1/2}(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)C_{<t>}^{-1/2}.$$

Then

$$C_{<t>}^{-1} = C^{-1/2} \left(I + \frac{1}{1 - u_t^T u_t} u_t u_t^T \right) C^{-1/2} , \tag{2.6}$$

and

$$C^{-1} = C_{<t>}^{-1/2} \left(I - \frac{1}{1 + v_t^T v_t} v_t v_t^T \right) C_{<t>}^{-1/2} . \tag{2.7}$$

Based on (??) of Theorem ??, we can see that in order to calculate $C_{<t>}^{-1}$, we only need to find $C^{-1/2}$. Since $C^{-1/2}$ only needs to be updated after some label $\boldsymbol{\omega}_t$ changes, the computation

of (**??**) is much less than updating $C_{<t>}^{-1}$ for each t.

In addition, based on (**??**), we can also see that we only need to find the inverse of C once during the whole algorithm. Suppose that we have optimized over t, and we changed the permutation involved. Before we move to $t+1$, the C^{-1} can be updated using (**??**) based on $C_{<t>}^{-1/2}$ and $v_t = C_{<t>}^{-1/2}(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)C_{<t>}^{-1/2}$, where $\boldsymbol{\omega}_t$ is the updated new label for $\boldsymbol{\theta}_t$.

If $C_{<t>}^{-1}$ in (**??**) is replaced by C^{-1}, which does not depend on t, then the Algorithm 2 is the same as the normal likelihood labeling proposed by **?**. However, their method is based on the asymptotic normality assumption of the labeled samples. Our proposed DETCOV method does not require such assumption and is derived from different motivation.

3 Examples

In this section, we will use one simulation example and two real data sets to compare our proposed labeling methods TRCOV and DETCOV with order constraint (OC) labeling and Stephens' KL algorithm (KL). By default, the OC method refers to ordering on the mean parameters. For TRCOV, DETCOV, and KL methods, we use the OC labels as the initials. All the computations are done in Matlab 7.0 using a personal desktop with Intel Core 2 Quad CPU 2.40GHz.

Example 1: We generate 400 data points from $0.3N(0,1) + 0.7N(0.5, 2^2)$. Based on this data set, we generate 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 **?**. That is to assume

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

where $D(\cdot)$ is Dirichlet distribution and $\Gamma(\alpha, \beta)$ is gamma distribution with mean α/β and variance α/β^2. Following the suggestion of **?**, we let $\delta = 1$, ξ equal the sample mean of

the observations, $\kappa = 1/R^2$, $\alpha = 2$, $g = 0.2$, and $h = 10/R^2$, where R is the range of the observations. **?** introduced an additional hierarchical model by allowing β to follow a gamma distribution, in order to reduce the influence of β on the posterior distribution of the number of components. Similar priors are used for the other two examples.

The runtime for KL, TRCOV, and DETCOV were 63, 1, and 2 seconds, respectively. Note that TRCOV and DETCOV were much faster than KL. In this example, TRCOV and DETCOV had the same labeling results.

Since there are only two components, we can easily make use of some parameter plots to see where the labeling differences occurred. Figure 1 is the plot of $\sigma_1 - \sigma_2$ vs. $\mu_1 - \mu_2$ and Figure 2 is the plot of $\sigma_1 - \sigma_2$ vs. π_1. For better visual results, we also add the permuted samples to the plots. From these plots, one can see that there are indeed relatively two tight clusters. However, OC and KL did not accurately recover these two regions. (Based on Figure 2, it appears that KL used the component proportions more heavily than the other methods.) The TRCOV/DETCOV methods clustered the two groups more naturally.

For better comparison, in Table 1, we also report the average and root mean squared error (RMSE) of the parameter estimates for each re-labeling algorithm. For completeness, we include the order constraint labeling results based on three different component parameters μ, σ, and π and denote them by OC-μ, OC-σ, and OC-π, respectively. (Note that the OC-μ is the same as OC). Among three order constraint labeling methods, OC-σ performed well (based on both bias and RMSE), but OC-μ and OC-π did poorly. OC-μ had small bias for μ_1 but large bias for all other parameters. OC-π had small bias for π_1 but large bias for all other parameters. The OC-σ did work well in this simulation study. This example demonstrates both the power and danger of order constraint labeling. Similar discoveries had been found in **?**. In this example, KL didn't perform as well as OC-σ. It had large bias for the μ_1 and μ_2 and large RMSE for σ_1 and σ_2. From the Table 1, we can see that both of our proposed methods TRCOV and DETCOV worked well and produced closer results to

OC-σ than any other methods.

In Table 2, we report the performance of interval estimates, based on the percentage of intervals that covered their corresponding true parameters and the average interval width. From the table, we can see that all labeling methods had good coverage and exhibited higher-than-nominal rates of coverage in general. Based on Table 2, we can see that no single labeling method provided shorter intervals for all parameters. However, in general, OC-σ, $TRCOV$ and $DETCOV$ provided shorter interval width than OC-μ, OC-σ, and KL for most of parameters.
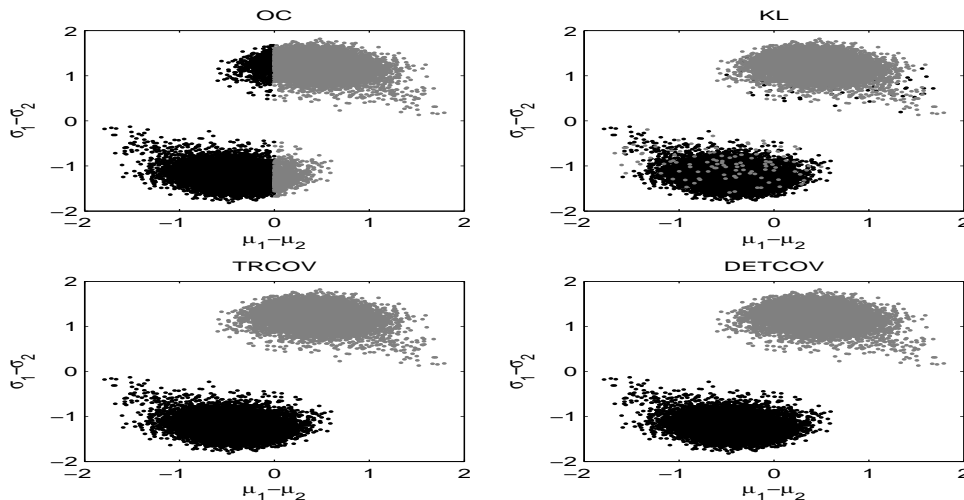
Figure 1: Plots of $\sigma_1 - \sigma_2$ vs. $\mu_1 - \mu_2$ for the four labeling methods in Example 1. The black points represent one set of labels and the gray points are the permuted samples.

Table 1: Average (RMSE) of Point Estimates Over 500 Repetitions

TRUE	OC-μ	OC-σ	OC-π	KL	TRCOV	DETCOV
$\mu_1 : 0$	-0.010(0.152)	0.057(0.200)	0.204(0.282)	0.119(0.247)	0.038(0.199)	0.059(0.204)
$\mu_2 : 0.5$	0.605(0.198)	0.538(0.202)	0.391(0.158)	0.476(0.195)	0.557(0.211)	0.536(0.201)
$\sigma_1 : 1$	1.232(0.327)	1.089(0.177)	1.281(0.396)	1.180(0.322)	1.095(0.184)	1.110(0.204)
$\sigma_2 : 2$	1.858(0.246)	2.001(0.139)	1.810(0.224)	1.910(0.181)	1.996(0.143)	1.981(0.156)
$\pi_1 : 0.3$	0.413(0.140)	0.360(0.108)	0.281(0.057)	0.305(0.089)	0.362(0.109)	0.342(0.113)
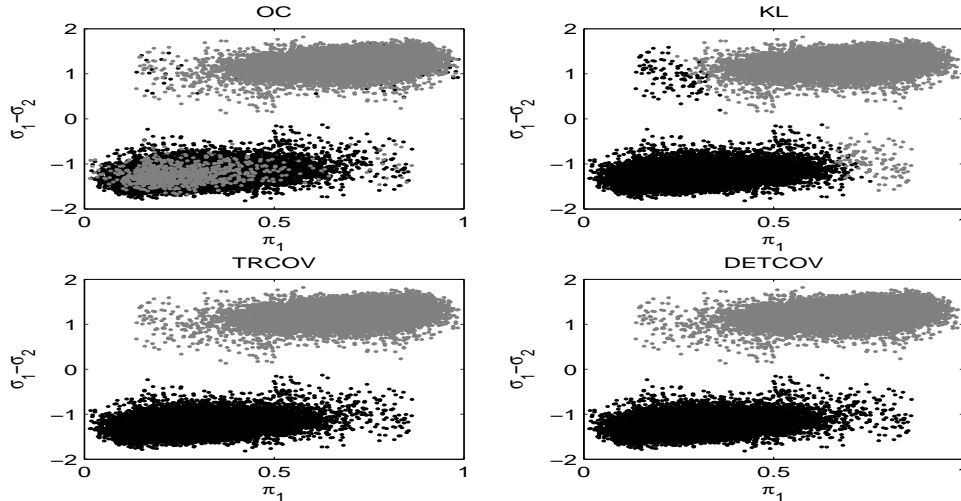
Figure 2: Plots of $\sigma_1 - \sigma_2$ vs. π_1 for the four labeling methods in Example 1.

Table 2: Percent Coverage (average width) of Nominal 95% Interval Estimates Over 500 Repetitions

TRUE	OC-μ	OC-σ	OC-π	KL	TRCOV	DETCOV
$\mu_1 : 0$	0.982(0.891)	0.976(1.172)	0.988(1.798)	0.984(1.517)	0.972(1.057)	0.974(1.208)
$\mu_2 : 0.5$	0.978(1.025)	0.982(1.100)	0.986(0.926)	0.976(0.911)	0.970(1.089)	0.974(1.002)
$\sigma_1 : 1$	0.988(1.405)	0.992(1.016)	0.992(1.803)	0.990(1.491)	0.992(1.043)	0.992(1.163)
$\sigma_2 : 2$	0.984(1.313)	0.990(0.703)	0.938(0.911)	0.942(0.738)	0.990(0.730)	0.958(0.696)
$\pi_1 : 0.3$	0.998(0.768)	0.992(0.699)	1.000(0.408)	0.992(0.494)	0.992(0.700)	0.992(0.615)

Example 2 (Acidity Data): We consider the acidity data set (**??**). The data are shown in Figure 3. The observations are the logarithms of an acidity index measured in a sample of 155 lakes in north-central Wisconsin. This data set has been analyzed as a mixture of Gaussian distributions by **???**. Based on the result of **?**, the posterior for three components was largest. Hence, we fit this data set by a three-component normal mixture. We post processed the 20,000 Gibbs samples by the OC, KL, TRCOV, and DETCOV labeling methods. The runtime for KL, TRCOV, and DETCOV were 45, 2, and 6 seconds, respectively.

It is difficult to use the similar graphic way in Example 1 to compare different labeling methods when the number of components is larger than two (**?**). Here, we mainly provided
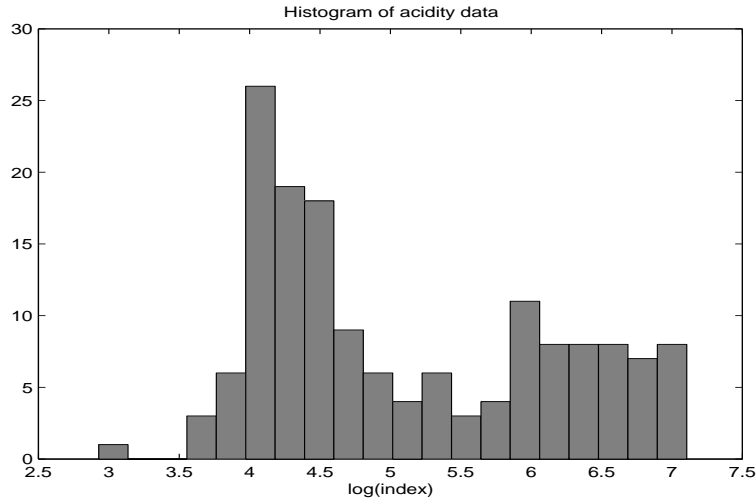
Figure 3: Histogram of acidity data. The number of bins used is 20.

the trace plots and the marginal density plots to illustrate the success of DETCOV. Figure 4 and 5 are the trace plots and the estimated marginal posterior density plots, respectively, for the original samples and the labeled samples by DETCOV. (The OC, KL, and TRCOV methods had similar visual results to DETCOV for those plots.) From these figures, one can see that the DETCOV method successfully removed the label switching in the raw output of the Gibbs sampler.

Example 3 (Galaxy Data): The galaxy data (**?**) 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 6. This data set has been analyzed by many researchers including, for example, **??????**. **?** also used this data set to explain the label switching problem. We fit this data by six-component normal mixture. We post processed the 20,000 Gibbs samples by the OC, KL, TRCOV, and DETCOV labeling methods.

The runtime for KL, TRCOV, and DETCOV were 2487, 47, and 1161 seconds, respectively. Hence, the TRCOV method is much faster than KL and DETCOV.

Figure 7 and 8 are the trace plots and the estimated marginal posterior density plots,

14

respectively, for the original samples and the labeled samples by DETCOV. For the marginal density plot, for comparison, we also add the OC labels. In this example, for trace plots, there is no big visual difference for the four labeling methods. For marginal density plots, the TRCOV methods had similar visual results to OC and the KL methods had similar visual results to DETCOV. From Figure 7 and 8, one can see that the DETCOV method successfully removed the label switching in the raw output of the Gibbs sampler. Based on Figure 8, one can see that the DETCOV method removed the multimodality of the marginal posterior densities of the means in the raw output, however the OC method did not remove the label switching very well. Therefore, in this example, the DETCOV and KL methods worked a little better than the OC and TRCOV methods.

Based on the above simulation study and the real data set application, one can see that DETCOV usually works better than the OC, KL, and TRCOV methods. In addition, TRCOV works better than the OC and comparable to KL but TRCOV runs much faster than KL and DETCOV.

4 Summary

In this article, we proposed two new clustering related labeling methods. The first method TRCOV uses the idea of K-means clustering to label the samples. This method can be considered as an extension of the OC method. However, unlike the OC method, this method can simultaneously incorporate different component parameters together and can be easily extended to the high dimension case. In addition, as shown in Section ??, this labeling method is computationally much faster than the KL and DETCOV methods. The second method DETCOV is to label the samples by minimizing the volume of the labeled samples. This method is invariant to the linear transformation of the parameters. Based on the examples in Section ??, we can see that the DETCOV method successfully removed the label switching in the raw output. Our simulations also have shown that the TRCOV and

15

DETCOV methods can produce good point and interval estimates.

In addition, our proposed methods TRCOV and DETCOV might also be able to solve label switching for frequentist mixtures since our methods only depend on the parameter samples. For frequentist mixtures, if one wants to use bootstrap method to estimate the variation of the parameter estimates, one needs to bootstrap the new data set and finds the corresponding parametric estimates for, say, N times. Let $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$ be the N bootstrap samples, which have meaningless labels. Similar to the Bayesian mixtures, one needs to solve the label switching problem for the N bootstrap samples before using them to estimate the variation. For bootstrap method, the data sets $\mathbf{x} = (x_1, \ldots, x_n)$ are different for different $\boldsymbol{\theta}_j$s and thus the classification probabilities are not well defined, since they require to use the same data set for all the parameter samples. Therefore, any labeling methods related to the classification probabilities, such as the KL algorithm, can not be applied. As far as we know, solving label switching for frequentist mixtures have not been well studied. This requires further research.

However, like other relabeling algorithms, our proposed methods TRCOV and DETCOV are not online algorithms. Users need to store all the samples before doing labeling. In addition, if the number of components m is too large or the dimension of the data is large, the DETCOV method might have numerical problems due to the calculation of $C_{<t>}^{-1}$ defined in (??). If this problem occurs, one could use a ridge type estimator for $C_{<t>}^{-1}$, say $(C_{<t>} + \lambda I)^{-1}$ for some constant λ.

Based on the equivalence between the labeling and clustering if only one permutation of each sample is allowed in each of the $m!$ clusters, one can also apply other clustering methods or criteria to do labeling.

5   Acknowledgements

APPENDIX: PROOFS

Proof of Theorem **??**: *For simplicity, suppose that there are only two unknown component parameters, such as component mean μ and component variance σ^2, for each component. Let $\boldsymbol{\theta}_t = (\boldsymbol{\xi}_{t1}^T, \boldsymbol{\xi}_{t2}^T)^T$, where $\boldsymbol{\xi}_{t1}$ is the m component means and $\boldsymbol{\xi}_{t2}$ is m component variance. Suppose*

$$M = \sum_{t=1}^{N}(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)^T = \begin{pmatrix} A & B \\ B & C \end{pmatrix},$$

where A, B, and C are all $m \times m$ matrix. Let $\tilde{\boldsymbol{\xi}}_{t1} = a_1 + b_1\boldsymbol{\xi}_{t1}$, $\tilde{\boldsymbol{\xi}}_{t2} = a_2 + b_2\boldsymbol{\xi}_{t1}$ for all t and $\tilde{\boldsymbol{\theta}}_c$ be the corresponding transformation of $\boldsymbol{\theta}_c$ (See Theorem **??**). *Then*

$$\tilde{M} = \sum_{t=1}^{N}(\tilde{\boldsymbol{\theta}}_t^{\boldsymbol{\omega}_t} - \tilde{\boldsymbol{\theta}}_c)(\tilde{\boldsymbol{\theta}}_t^{\boldsymbol{\omega}_t} - \tilde{\boldsymbol{\theta}}_c)^T = \begin{pmatrix} b_1^2 A & b_1 b_2 B \\ b_1 b_2 B & b_2^2 C \end{pmatrix}.$$

Based on some matrix algebra, we have

$$\det(M) = \det\begin{pmatrix} A & B \\ B & C \end{pmatrix} = \det(A)\det(C - BA^{-1}B).$$

So

$$\det(\tilde{M}) = \det \begin{pmatrix} b_1^2 A & b_1 b_2 B \\ b_1 b_2 B & b_2^2 C \end{pmatrix}$$

$$= \det(b_1^2 A) \det(b_2^2 C - b_2^2 B A^{-1} B)$$

$$= b_1^{2m} b_2^{2m} \det(A) \det(C - B A^{-1} B)$$

$$= b_1^{2m} b_2^{2m} \det(M).$$

So for the linear transformation, the determinant of covariance criteria will not change, up to a multiplication constant and thus the found labels will not change. □

Proof of Theorem ??: Let

$$\bar{\boldsymbol{\theta}} = \frac{1}{N} \sum_{t=1}^{N} \boldsymbol{\theta}_t^{\boldsymbol{\omega}_t}.$$

Note that

$$\det \left(\sum_{t=1}^{N} (\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)^T \right) = \det \left(\sum_{t=1}^{N} (\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \bar{\boldsymbol{\theta}})(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \bar{\boldsymbol{\theta}})^T + N(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_c)(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_c)^T \right)$$

$$\geq \det \left(\sum_{t=1}^{N} (\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \bar{\boldsymbol{\theta}})(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \bar{\boldsymbol{\theta}})^T \right),$$

since $(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_c)(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_c)^T) \geq 0$. *So the minimum of (??) over* $\boldsymbol{\theta}_c$ *occurs at the sample mean of* $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$. □

Proof of Theorem ??: Note that

$$C_{<t>} = C - (\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)^T = C^{1/2}(I - u_t u_t^T)C^{1/2},$$

where $u_t = C^{-1/2}(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c)$. By some calculation, we can verify

$$(I - u_t u_t^T)^{-1} = \left(I + \frac{1}{1 - u_t^T u_t} u_t u_t^T\right).$$

So

$$C_{<t>}^{-1} = C^{-1/2}\left(I + \frac{1}{1 - u_t^T u_t} u_t u_t^T\right) C^{-1/2}.$$

Let

$$v_t = C_{<t>}^{-1/2}(\boldsymbol{\theta}_t^{\boldsymbol{\omega}_t} - \boldsymbol{\theta}_c) C_{<t>}^{-1/2}.$$
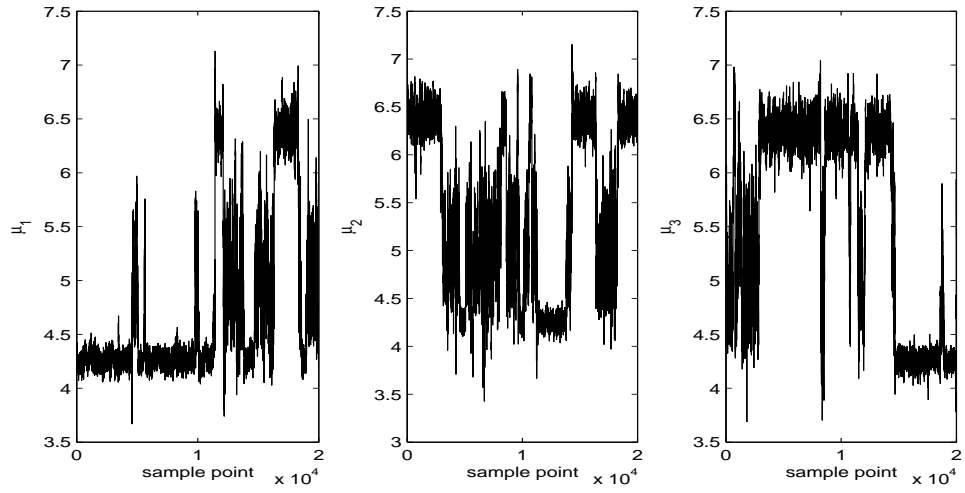
We have

$$C = C_{<t>} + (\boldsymbol{\theta}_t - \boldsymbol{\theta}_c)(\boldsymbol{\theta}_t - \boldsymbol{\theta}_c)^T = C_{<t>}^{1/2}(I + v_t v_t^T)C_{<t>}^{1/2}.$$

Note that

$$(I + v_t v_t^T)^{-1} = \left(I - \frac{1}{1 + v_t^T v_t} v_t v_t^T\right)$$

So

$$C^{-1} = C_{<t>}^{-1/2}\left(I - \frac{1}{1 + v_t^T v_t} v_t v_t^T\right) C_{<t>}^{-1/2}. \quad \square$$

(a)
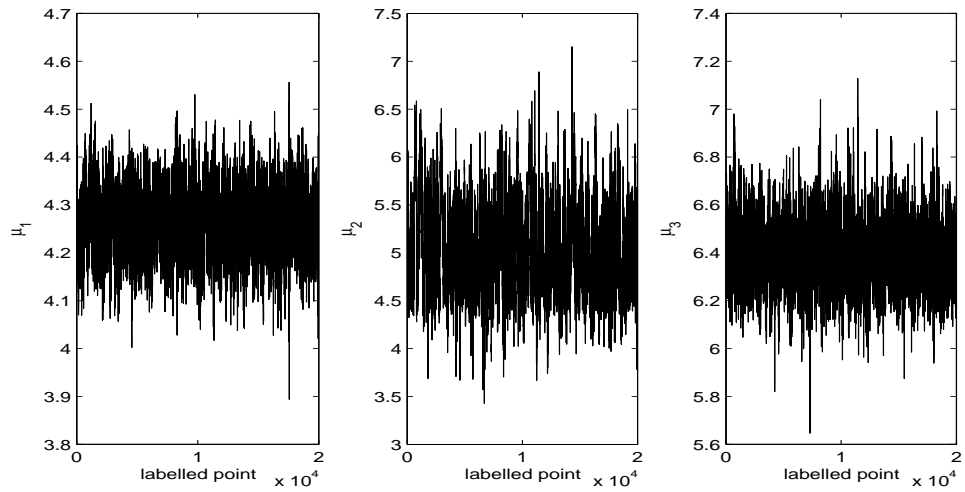
(b)

Figure 4: Trace plots of the Gibbs samples of component means for acidity data: (a) original Gibbs samples; (b) labeled samples by DETCOV.
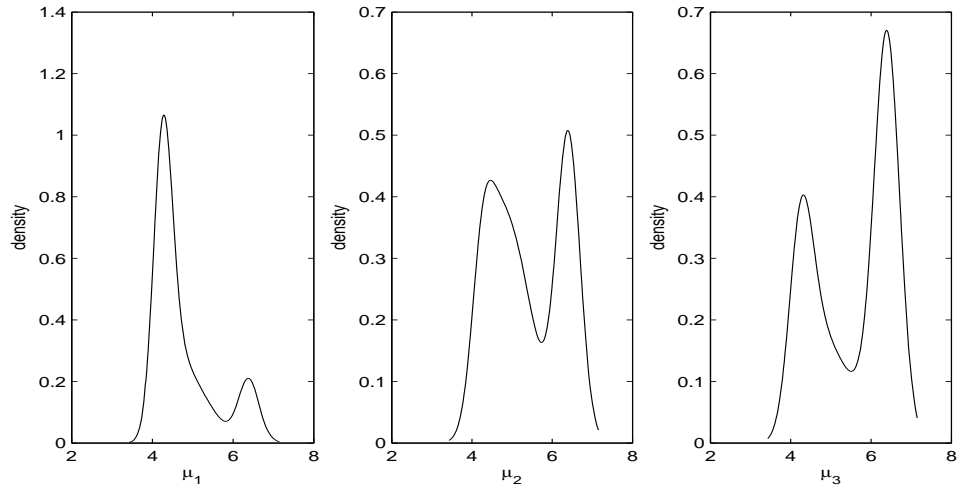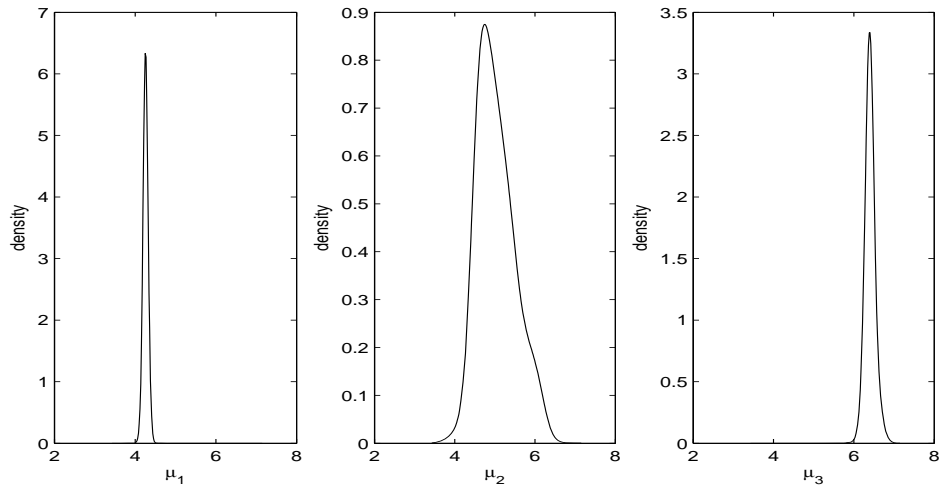
Figure 5: Plots of estimated marginal posterior densities of component means for acidity data based on: (a) original Gibbs samples; (b) labeled samples by DETCOV.
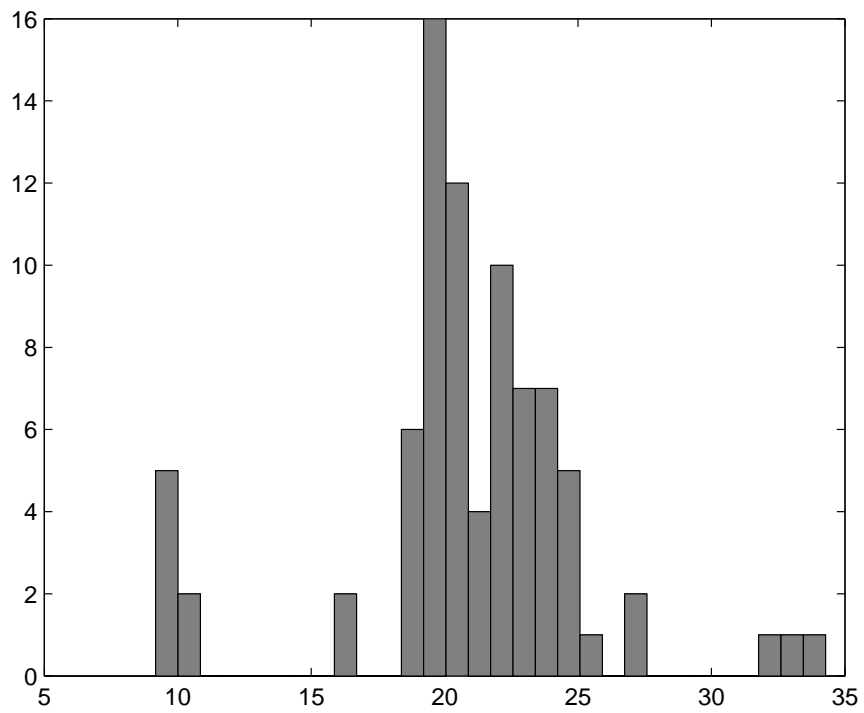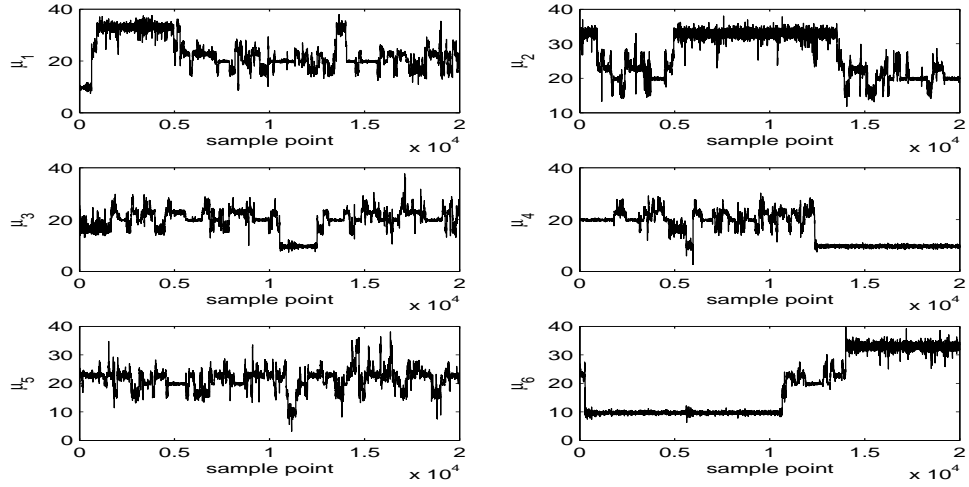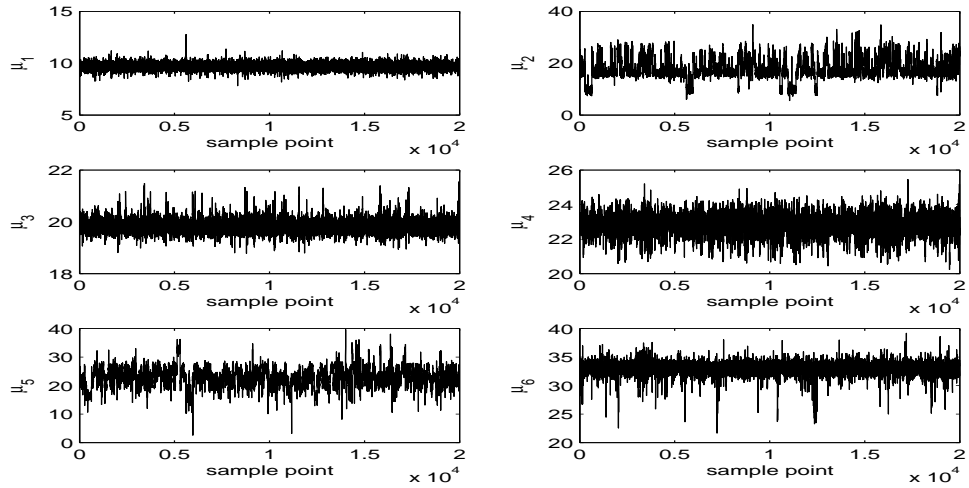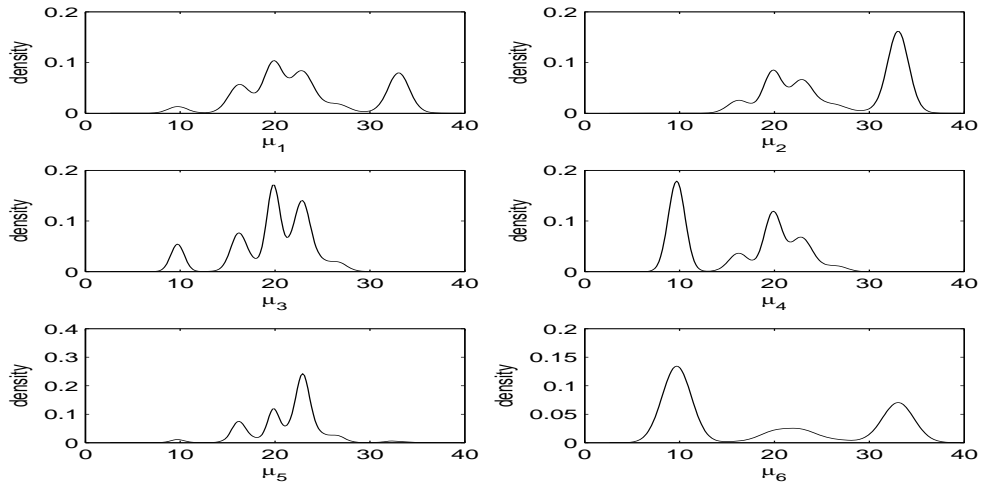
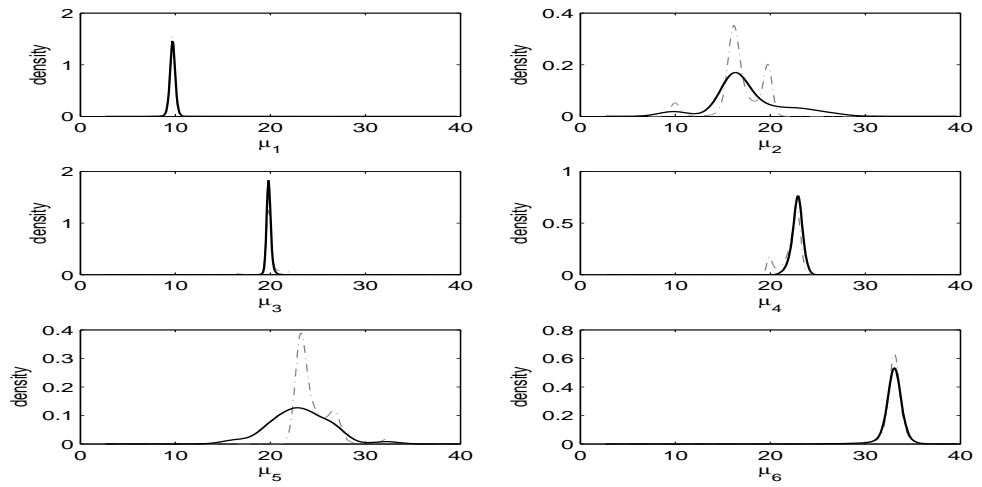Figure 6: Histogram plot of galaxy data. The number of bins used is 30.

Figure 7: Trace plots of the Gibbs samples of component means for galaxy data: (a) original Gibbs samples; (b) labeled samples by DETCOV.

Figure 8: Plots of estimated marginal posterior densities of component means for galaxy data based on: (a) original Gibbs samples; (b) labeled samples by DETCOV (line) and labeled samples by OC (dash-dot).

References

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

Carlin, B. P. and Chib, S. (1995). Bayesian model choice via Markov chain Monte Carlo methods. Journal of Royal Statistical Society, *B57, 473-484.*

*Celeux, G. (1998), Bayesian inference for mixtures: The label switching problem. In Comp-stat 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.*

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, *B56, 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.*

−−(2006), Finite Mixture and Markov Switching Models, *Springer, 2006.*

Gallegos, M.T. and Ritter, G. (2005). A robust method for cluster analysis. Annals of Statistics, *33, 347-380.*

Geweke, J. (2007). Interpretation and inference in mixture models: Simple MCMC works. Computational Statistics and Data Analysis, *51, 3529-3550.*

Grun, 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.*

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

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.*

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

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.*

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, *B59, 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.*

Rousseeuw, P. J. (1984). Least meadian of squares regression. Journal of the American Statistical Association, *79, 871-880.*

Stephens, M. (2000). Dealing with label switching in mixture models. Journal of Royal Statistical Society, *B62, 795-809.*

Walker, A. M. (1969). On the asymptotic behaviour of posterior distributions. Journal of the Royal Statistical Society, *B31, 80-88.*

Yao, W. and Lindsay, B. G. (2009). Bayesian Mixture Labeling by Highest Posterior Density. Journal of American Statistical Association, *104, 758-767.*