Bayesian Solutions to the Label Switching Problem

Kai Puolamäki and Samuel Kaski

Helsinki Institute for Information Technology HIIT
Department of Information and Computer Science
Helsinki University of Technology
P.O. Box 5400, FI-02015 TKK, Finland

Abstract. The label switching problem, the unidentifiability of the permutation of clusters or more generally latent variables, makes interpretation of results computed with MCMC sampling difficult. We introduce a fully Bayesian treatment of the permutations which performs better than alternatives. The method can even be used to compute summaries of the posterior samples for nonparametric Bayesian methods, for which no good solutions exist so far. Although being approximative in that case, the results are very promising. The summaries are intuitively appealing: A summarized cluster is defined as a set of points for which the likelihood of being in the same cluster is maximized.

1 Introduction

In the recent years there has been a dramatic increase in the use of sampling methods in computing with probabilistic models. The main reason naturally is that Markov Chain Monte Carlo (MCMC) methods make it possible to use complex-structured models, for which variational and other techniques are not feasible. MCMC methods are not without their problems, however.

One of these problems is *label switching* of discrete latent or hidden variables of the probabilistic model, which makes interpretation of the results hard. The problem arises if the prior and the likelihood function, and hence the posterior probability distribution, are invariant under a permutation of the values, "labels," of the discrete latent variable. This leads to non-identifiability of the labels of the latent discrete variable.

As a simple example, consider a Gaussian mixture model with two mixture components, and a data set with two well separated groups A and B. In one MCMC sample the groups A and B may be represented by mixture components 1 and 2, and in another sample by 2 and 1, respectively. It follows that if we try to compute some mixture component-specific quantities as averages over the posterior samples, as we normally would in Bayesian data analysis, we get meaningless results. For instance, the mean position of data points in a given mixture component becomes the mean of the whole data set.

The label switching is inherent to sampling—there is no problem if we are content with a point solution, such as a maximum likelihood or maximum a

N. Adams et al. (Eds.): IDA 2009, LNCS 5772, pp. 381–392, 2009.

posteriori solution, or a variational approximation where we can choose an arbitrary labeling.

The non-identifiability poses no problems if the quantities of interest are invariant under permutations of the labels. Problems occur when the individual parameter values need to be compared across samples, as in common measures of convergence of the MCMC simulation, for instance. Another problematic area are label-specific summaries and interpretations. The label switching problem has been discussed extensively in the framework of mixture models. When interpreting a cluster in terms of its typical parameter values, or with the list of data samples mapped to it, switching changes the cluster radically.

There have been many suggestions as to how to deal with the label switching problem. The most straightforward solution is to use a sampler that is inefficient in the sense that the labels are very unlikely to switch. Therefore, a reasonable assumption is that the mixture labels are not permuted across the samples. Many of the samplers, such as the Gibbs sampler for mixture models [1], fall into this category. It turns out that the sampler may perform otherwise adequately even if it is unable to switch labels [2]. This solves the label switching problem in practice, but in an arguably inelegant manner that cannot be proven to always work.

Another solution is to use artificial identifiability constraints to break the symmetry in the likelihood [3]. For example, if the component parameters are denoted by μ_j, a possible constraint is $\mu_j < \mu_{j+1}$, where $j < k$ and k is the number of components. Unfortunately, in the Bayesian context these constraints do not however always perform adequately [4].

[5, 6, 7] re-labeled the mixture components in each sample by using a k-means-type approach. [8, 9] relabeled the points in each sample using label-invariant loss functions, such as functions that compare whether the cluster assignments of the data points are equal in pairs of samples; see also [10]. [11] introduced a probabilistic relabeling method, but not in the context of Bernoulli mixture models as in our work. See [4, 11] for a recent review of attempts to solve the label switching problem in mixture models.

A drawback of the earlier relabeling approaches is that they associate a certain more or less heuristic labeling, or permutation of labels, to each sample to make the samples comparable. As can be seen from the variety of approaches, however, the labeling is not unique. Furthermore, assigning a fixed labeling is slightly inelegant considering that the actual modeling follows the Bayesian approach.

2 Summary of Our Contribution

We propose a probabilistic relabeling that can be implemented in a straightforward way for the MCMC samples from any probability distribution that includes a discrete latent variable. We show that our approach gives a consistent probabilistic labeling. In our examples the probabilistic models are mixture models, but we want to emphasize that the results generalize to all probabilistic models which suffer from the label switching problem.

In a nutshell the idea is to include an additional Bernoulli mixture model that can model the distribution of the discrete latent variable in the original probabilistic model. We will explain below why this is a good solution.

Our contributions in this paper are:

- Fully Bayesian treatment of the label switching problem.
- A straightforward way to obtain mixing matrices that are not affected by label switching (Algorithm 1).
- A principled and probabilistic relabeling of samples in order to compute expectations (Section 4.2).
- An approximation scheme having a polynomial time complexity (instead of the naive $O(k!)$), where k is the number of discrete states in the latent variable, to compute the expectations.
- Experimental proofs of concept, including application to Dirichlet Process Mixture model with varying number of mixture components.

3 Definitions

We introduce the problem first in the general notation and then discuss the case of mixture models in more detail.

3.1 General Derivation

We denote the data by $\mathbf{x} = (x_1, \ldots, x_n)$. We have a probabilistic model that has n instances of a discrete-valued latent variable $\mathbf{z} = (z_1, \ldots, z_n)$ having labels $z_i \in [k]$, where $[k] = \{1, \ldots, k\}$. We can alternatively use binary indicator variables z_{ij} such that $z_{ij} = 1$ if $z_i = j$, $z_{ij} = 0$ otherwise.

We denote by $\phi = (\mathbf{z}, \theta)$ all parameters of the model. Here θ are all other parameters besides the latent variable $\mathbf{z}$. Denote by $\sigma \in S_k$ a *permutation* function of the labels $[k]$. We use $\sigma(\mathbf{z})$ as a shorthand of the application of σ to all the z_i, and by $\sigma(\phi)$ the permutation by σ of labels of the parameters of the model of σ. Invariance under the permutation means that for all permutations $\sigma \in S_k$, the prior and the likelihood satisfy $p(\phi) = p(\sigma(\phi))$ and $p(\mathbf{x} \mid \phi) = p(\mathbf{x} \mid \sigma(\phi))$, respectively; hence the posterior probability distribution $p(\phi \mid \mathbf{x})$ is also invariant under the permutation of the labels as $p(\phi \mid \mathbf{x}) \propto p(\mathbf{x} \mid \phi)p(\phi)$. In the remainder we assume that the model is invariant under the permutation of labels of z_i.

Given the above definitions, we can make th trivial observation that

Observation 1. *For a probabilistic model containing a latent variable z_i and for which the prior probability density and likelihood are invariant under the permutations of the labels of z_i, hence the symmetry needs to be broken before meaningful summaries of the latent variables can be computed.*

$$p(z_{ij} = 1 \mid \mathbf{x}) = \frac{1}{k}.$$

Algorithm 1. Bernoulli Labeling

BernoulliLabeling($\{z_{ij}^t\}$) {Input: $\{z_{ij}^t\}$, the indicator variables z_{ij}^t for all $t \in [T]$, $i \in [n]$ and $j \in [k]$. Output: $\tilde{\beta}$, a $k \times n$ parameter matrix of the Bernoulli mixture model.}
Let $Z(r,i) \leftarrow z_{ij}^t$, where $r = k(t-1)+j$ and $Z \in \{0,1\}^{Tk \times n}$ for all $i \in [n]$, $j \in [k]$, and $t \in [T]$.
Let $\tilde{\beta} \leftarrow$ BernoulliMixture(Z,k). {Algorithm 3}
return $\tilde{\beta}$.

Algorithm 2. Generalized Bernoulli Labeling

Generalized BernoulliLabeling($\{z_i^t\}$, k) {Input: $\{z_i^t\}$, the cluster indices z_i^t for all $t \in [T]$ and $i \in [n]$; k, cluster of components in Bernoulli Labeling. Output: $\tilde{\beta}$, a $k \times n$ parameter matrix of the Bernoulli mixture model.}
Let Z be an empty matrix with n columns.
for $t = 1$ to T **do**
  Let k^t be the number of non-empty cluster in sample t, and let Y_{ji} be 1 if $z_i^t = j$, 0 otherwise.
  Append the rows of matrix Y to the rows of matrix Z.
end for
Let $\tilde{\beta} \leftarrow$ BernoulliMixture(Z,k).
return $\tilde{\beta}$.

The central contribution of this work is to apply a Bernoulli mixture model to the indicator variables z_{ij}^t as given by Algorithm 1. We discuss the motivation of the algorithm in more detail in Section 4. Briefly put, the idea is to apply a Bernoulli mixture model to the rows of a matrix of indicator variables Z where the rows correspond to mixture components in different samples.

Algorithm 1 uses the Bernoulli mixture model having the likelihood

$$p(Z \mid \tilde{\beta}) = \prod_{r=1}^{R} \sum_{j=1}^{k} \frac{1}{k} \prod_{i=1}^{n} \tilde{\beta}(j,i)^{Z(r,i)} \left(1 - \tilde{\beta}(j,i)\right)^{1-Z(r,i)}, \tag{1}$$

where the parameters are given as a mixture matrix $\tilde{\beta} \in [0,1]^{k \times n}$. The data matrix is $Z \in \{0,1\}^{R \times n}$, where $R = Tk$. We use Algorithm 3 to maximize the likelihood of Equation (1); the algorithm is a standard EM algorithm. Because EM is guaranteed to find a local but not necessary a global optimum, in our experiments we run Algorithm 3 ten times with different random initializations and pick the solution with the largest likelihood.

Another approach is to take explicitly into account the fact that there is a unique permutation of labels for each sample. The mixing matrix $\tilde{\beta}$ can then be found by optimizing the cost function given by

$$\prod_{t=1}^{T} \sum_{\sigma \in S_k} \frac{1}{k!} \prod_{i=1}^{n} \tilde{\beta}(\sigma(z_i^t), i), \tag{2}$$

Algorithm 3. Bernoulli Mixture

BernoulliMixture(Z, k) {Input: Z, a $R \times n$ binary matrix; k, the number of mixture components. Output: $\tilde{\beta}$, a $k \times n$ maximum likelihood parameter matrix of the Bernoulli Mixture model.}
Initialize $\tilde{\beta} \in [0,1]^{k \times n}$ at random.
repeat
{E step:}
Let $\gamma(r,j) \leftarrow \prod_{i=1}^{n} \tilde{\beta}(j,i)^{Z(r,i)} \left(1 - \tilde{\beta}(j,i)\right)^{1-Z(r,i)}$ for all $r \in [R]$ and $j \in [k]$.
Let $Z(r) \leftarrow \sum_{j=1}^{k} \gamma(r,j)$ for all $r \in [R]$.
Let $\gamma(r,j) \leftarrow \gamma(r,j)/Z(r)$ for all $r \in [R]$ and $j \in [k]$.
{M step:}
Let $\tilde{\beta}(j,i) \leftarrow \sum_{r=1}^{R} \gamma(r,j) Z(r,i) / \sum_{r=1}^{R} \gamma(r,j)$ for all $j \in [k]$ and $i \in [n]$.
until convergence
return $\tilde{\beta}$.

Algorithm 4. Bernoulli Mixture Permutation

BernoulliMixturePerm($\{z_i^t\}$) {Input: $\{z_i^t\}$, the cluster indices z_i^t for all $t \in [T]$ and $i \in [n]$. Output: $\tilde{\beta}$, $k \times n$ maximum likelihood parameter matrix of the Bernoulli Mixture model.}
Initialize $\tilde{\beta} \in [0,1]^{k \times n}$ in random.
repeat
{E step:}
Let $\gamma(t,\sigma) \leftarrow \prod_{i=1}^{n} \tilde{\beta}(\sigma^{-1}(z_i^t), i)$ for all $t \in [T]$ and $\sigma \in S_k$.
Let $Z(t) \leftarrow \sum_{\sigma \in S_k} \gamma(t,\sigma)$ for all $t \in [T]$.
Let $\gamma(t,\sigma) \leftarrow \gamma(t,\sigma)/Z(t)$ for all $t \in [T]$ and $\sigma \in S_k$.
{M step:}
Let $\tilde{\beta}(j,i) \leftarrow \sum_{t=1}^{T} \sum_{\sigma \in S_k} \gamma(t,\sigma) z^t_{\sigma(i)} / T$ for all $j \in [k]$ and $i \in [n]$.
until convergence
return $\tilde{\beta}$.

by using EM algorithm, presented by Algorithm 4. We call this model Bernoulli Mixture Permutation model.

3.2 Mixture Models

Mixture models are a common class of probabilistic models where the label switching is a problem. We use the mixture model as a prototype probabilistic model which suffers from the label switching problem.

In a mixture model with k mixture components, there is a discrete-valued latent variable $\mathbf{z} = (z_1, \ldots, z_n)$. Here $z_i \in [k]$ tells which mixture component the data point x_i comes from. The other parameters θ consist of the mixture probabilities $\pi = (\pi(1), \ldots, \pi(k))$ that satisfy $\sum_{j=1}^{k} \pi(j) = 1$, and the component-specific parameters $\beta = (\beta(1), \ldots, \beta(k))$ that define the likelihood of a data point x_i given a mixture component z_i, according to any parametric likelihood function $p(x_i \mid \beta(z_i))$ such as the multivariate Gaussian. In summary, here $\theta = (\pi, \beta)$.

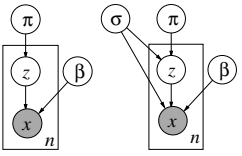

Fig. 1. Graphical representation of mixture model presented in Section 3.2 without (left) and with (right) a permutation sampled from S_k; see the likelihoods of Equations (3) and (4), respectively. The likelihoods and therefore the generative processes of the two models are equivalent.

The likelihood of the mixture model, shown graphically in Figure 1 (left), is given by

$$p(\mathbf{x} \mid \pi, \beta) = \prod_{i=1}^{n} \sum_{z_i=1}^{k} \pi(z_i) p(x_i \mid \beta(z_i)). \tag{3}$$

The likelihood does not change if the labels are permuted by any permutation σ. A special case is where the permutation σ is sampled from S_k uniformly at random, see Figure 1 (right); the likelihood is then

$$p(\mathbf{x} \mid \pi, \beta) = \frac{1}{k!} \sum_{\sigma \in S_k} \prod_{i=1}^{n} \sum_{z_i=1}^{k} \pi(\sigma(z_i)) p(x_i \mid \beta(\sigma(z_i))). \tag{4}$$

See [4] for further discussion.

4 Theoretical Properties

Our idea is, intuitively, to find an assignment of the data items into k mixture components such that if a set of data items co-occurs in the same mixture components in several samples then they should be assigned into the same mixture component.

In this section, we show that the Bernoulli mixture cost function optimized by Algorithm 1 is invariant under the permutation of the labels of the original probabilistic model. We further show that Algorithm 1 exactly reproduces the mixture components of the mixture model described in Section 3.2.

We also provide a principled probabilistic relabeling algorithm of the samples in Section 4.2.

4.1 Properties of the Bernoulli Labeling

Observation 2. *The cost function optimized by Algorithm 1 is invariant under the permutation of labels of the probabilistic model.*

Proof. The Algorithm 1 finds the mixture matrix by maximizing the likelihood given by Equation (1). Any permutation of the labels in the original probabilistic model (in which the discrete variables $\mathbf{z}$ are parameters) corresponds to a

permutation of rows of the matrix Z. The likelihood, Equation (1), remains unchanged in any such permutation. □

The following theorem shows that Algorithm 1 gives consistent results for a mixture model with given fixed parameters π and β.

Theorem 3. *Given a mixture model, parametrized by $\phi = (\mathbf{z}, \theta)$, and data as defined in Section 3.1, Algorithm 1 finds the probability distribution $p(z_i = j \mid \mathbf{x}; \theta)$ in the limit of infinitely many samples, $T \to \infty$.*

Proof. A randomly picked row of matrix Z represents a given component $j \in [k]$ with probability $1/k$. The probability of ones in the ith dimension of component j is given by the distribution $p(z_i = j \mid \mathbf{x}; \theta)$; this distribution can be computed from the mixture model. If we set $\tilde{\beta}(j, i) = p(z_i = j \mid \mathbf{x}; \theta)$, then the Z can be thought of as having been sampled from the Bernoulli mixture model (Equation 1), the probability of each component being $1/k$. Hence, at the limit of infinitely many samples ($T \to \infty$), the maximum of Equation (1) is given by $\tilde{\beta}(j, i) = p(z_i = j \mid \mathbf{x}; \theta)$. Furthermore, as the number of rows of Z approaches infinity, the posterior probability density of the Bernoulli mixture model is essentially a multimodal point estimate with $k!$ modes corresponding to different permutations of the labels, one of the modes being at $\tilde{\beta}$. □

Theorem 3 is illustrated graphically for a mixture model of Section 3.2 in Figure 2.

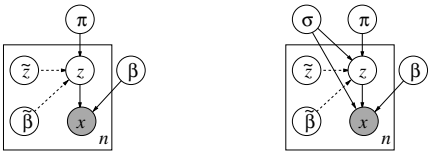

Fig. 2. *Left:* Graphical representation of the mixture model presented in Section 3.2, with Bernoulli labeling of Section 4.1. The distribution of the latent variables $\mathbf{z}$ can be equivalently derived either from the mixture model (solid lines), ignoring the dashed lines, or from the Bernoulli mixture model (dashed lines), ignoring the solid lines. *Right:* Graphical representation of mixture model presented in Section 3.2 using Probabilistic Bernoulli Relabeling of Section 4.2: the distribution of the latent variables $\mathbf{z}$ can be equivalently derived either from the mixture model (solid lines), ignoring the dashed lines; or the Bernoulli Mixture model with permutation σ (dashed lines), ignoring parameters π and β.

Observation 2 and Theorem 3 together imply that our approach is completely insensitive to label switching. That is, we can do an arbitrary permutation of labels within each MCMC sample without affecting the results.

It follows that $\tilde{\beta}$ obtained by Algorithm 1 can be used as a principled "point estimate" to summarize the mixture components.

The previous work on relabeling of MCMC samples has focused on finding a single permutation for each sample such that the resulting samples, having

permuted labels, can be aggregated that has suffered from the fact that although there usually exists one "most likely" permutation of labels for each sample, the probabilities of *all* possible permutations should be non-vanishing due to the probabilistic nature of the model.

4.2 Probabilistic Bernoulli Relabeling

In this section we consider the problem of computing an expectation of some function $f(\mathbf{z}, \theta)$ of the parameters of the probabilistic model, using the T independently drawn samples at the limit of infinite (or very large) T.

One of the motivations for the model is that it is consistent in the sense that if we had a probabilistic model with a fixed set of parameters θ, but such that the labels in the parameters $\mathbf{z}$ and θ have been relabeled by a permutation function $\sigma \in S_k$, drawn uniformly at random, then the algorithm would asymptotically find correct values.

Now, our task is to compute the posterior distribution for the probability of the permutation for each sample. We can use Algorithm 1 to find the Bernoulli mixing matrix $\tilde{\beta}$ because by Observation 2 the algorithm is unaffected by any permutation σ. Because the number of samples T is very large the posterior distribution is a multimodal point distribution with one of the $k!$ peaks at $\tilde{\beta}$. Figure 2 shows the structure of the model in the case of the mixture model defined in Section 3.2.

Given a fixed $\tilde{\beta}$ we can derive the probability of a permutation for each sample $p(\sigma \mid \mathbf{z}^t, \tilde{\beta})$, and then propose to compute an expectation using *Probabilistic Bernoulli Relabeling* as follows:

$$E_B\left[f(\mathbf{z}, \theta)\right] = \frac{1}{T} \sum_{t=1}^{T} \sum_{\sigma \in S_k} p(\sigma \mid \mathbf{z}^t, \tilde{\beta}) f(\sigma(\mathbf{z}^t), \sigma(\theta^t)), \tag{5}$$

where the posterior probability of a permutation σ for a sample t is given by

$$p(\sigma \mid \mathbf{z}^t, \tilde{\beta}) \propto \sum_{j=1}^{k} \frac{1}{k} \prod_{i=1}^{n} \tilde{\beta}(j,i)^{z^t_{i\sigma(j)}} \left(1 - \tilde{\beta}(j,i)\right)^{1 - z^t_{i\sigma(j)}}, \tag{6}$$

with a normalization defined by $\sum_{\sigma \in S_k} p(\sigma \mid \mathbf{z}^t, \tilde{\beta}) = 1$.

We first note that the expectation defined in Equation (5) reduces to normal expectation in the absence of any label switching.

Observation 4. *If the expectation defined by $f(\mathbf{z}, \theta)$ is invariant under permutation of labels, that is, $f(\mathbf{z}, \theta) = f(\sigma(\mathbf{z}), \sigma(\theta))$ for all $\sigma \in S_k$, then the expectation of Equation (5) reduces to normal expectation of $E[f] = \frac{1}{T} \sum_{t=1}^{T} f(\mathbf{z}^t, \theta^t)$.*

The Probabilistic Bernoulli Relabeling of Equation (5) requires the summation over all $k!$ permutations in S_k. The sum is computable for small enough values of k. For larger values of k, however, the summation can be approximated in polynomial time in k by first finding the most likely permutation by using the

Hungarian algorithm [12]; the time complexity of the Hungarian algorithm is $O(k^3)$. One can then apply Equation (6) to all permutation functions σ that can be reached by at most l swaps from the most likely permutation found by the Hungarian algorithm; the number of these permutation functions is $O(k^{2l})$. All permutations σ which are reachable with more than l permutations can to a good accuracy be approximated with $p(\sigma \mid \mathbf{z}^t, \tilde{\beta}) \approx 0$. As a result, the sum of Equation (5) has only $O(k^{2l})$ non-vanishing terms and the approximate expectation can be therefore be computed in $O(k^3 + k^{2l})$ time.

Finally, we note that although the Figures of Sections 4.1 and 4.2 were given for label switching in the context of the mixture model defined in Section 3.2, the derivations are otherwise general. The method can be applied for any probabilistic model having a non-identifiable discrete latent variable.

5 Experiments

5.1 Mixture Model

We generate an artificial data set by drawing samples from three Gaussian distributions, n samples from each. Each Gaussian has unit variance, and their means are $-x$, 0 and x. We then run a Gibbs sampler for a normal mixture model having $k = 3$ components and conjugate priors (with variance of each component fixed to unity) with parallel tempering as described by [13]. As a consequence of parallel tempering, the sampler switches labels. After 1000 burn-in samples we use the next 1000 samples in our analysis.

Our data analysis task is to use the samples to (i) estimate the means of the mixture components (MEANS), and (ii) to estimate the cluster assignments of the data points (ASSIGN). The error measure in the first task is the difference between the estimated cluster centroids and the "true" cluster centroids at $-x$, 0 and x. The objective measure in the second task is the classification accuracy (sum of probabilities of correct classes) when the true classes (the index of the generating distribution) are known.

The problem is easy for large values x or n; then all methods give equivalent results. For small or moderate values of x and n the methods differ.

Our methods are the Bernoulli mixture model (BM) and Bernoulli mixture model with permutations (BMP). The baseline methods are the identity constraint model (IC), where the samples are permuted such that the means of the mixture components are ordered in an increasing order. The second baseline method [7], denoted by STE, finds permutations using an EM-type approach. We include as a baseline a dummy model DUMB, in which all cluster probabilities are $\frac{1}{3}$.

We chose $n = 5$ and $x = 2$ (tasks MEANS and ASSIGN-2 or $x = \frac{2}{3}$ (task ASSIGN-2/3), and created 100 data sets.

The performance of IC is generally worse than that of the others (Table 1). In task MEANS all algorithms performed comparably. In task ASSIGN-2, BMP was the best, although the differences are very small. In task ASSIGN-2/3 the Bernoulli mixture model (BM) was superior; the reason is that all clusters are

Table 1. Squared prediction errors for the task MEANS (smaller is better); classification accuracy for tasks ASSIGN-2 having $x = 2$ and ASSIGN-2/3 having $x = \frac{2}{3}$ (larger is better), for a data set with $n = 5$. In task ASSIGN-2, BMP outperforms all the other models ($p < 0.05$). The differences are small, however. In task ASSIGN-2/3, BM outperforms all other models ($p < 10^{-9}$). All tests were one-tailed Wilcoxon Signed Rank Tests.

	BM	BMP	DUMB	IC	STE
MEANS	**0.676**	**0.680**	1.632	1.109	**0.676**
ASSIGN-2	0.598	**0.5995**	0.333	0.575	0.5990
ASSIGN-2/3	**0.442**	0.382	0.333	0.386	0.380

very similar and in many samples one of the clusters remained empty. The BM model will then assign one mixture component to such an empty cluster. The other models suffer from the strong assumption that there must be three clusters (although effectively the number of clusters is smaller).

5.2 Dirichlet Process Mixture

We studied the capability of the Bernoulli mixture to handle varying number of clusters in nonparametric Bayesian settings, by implementing a Dirichlet Process Mixture Model according to [14], with parallel tempering and the hyperparameter α fixed to one. The Gaussian mixture components had a conjugate prior with unit variance. We applied the model to the GALAXY data set [15] consisting of zero-mean relative velocities in 1000 km/sec of 82 galaxies from 6 well-separated conic sections of an unfilled survey of the Corona Borealis region. Multimodality in such surveys is evidence for voids and superclusters in the far universe. The means of the non-empty mixture components are shown in Figure 3. Due to the parallel tempering, the sampling mixes well and there is label switching.

Because the number of clusters varies, out of the introduced algorithms only the generalized Bernoulli labeling (Algorithm 2) is applicable. We ran the algorithm with three numbers of Mixture components, $k = 5$, $k = 6$ and $k = 7$; the

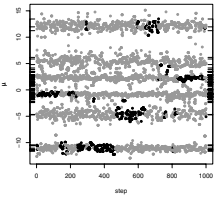

Fig. 3. The cluster means of the Dirichlet Process Mixture on the GALAXY data. Mixture component 2 is highlighted with a darker shade; label switching is evident from the plot. The rug plot on the vertical axis show the data, that is, the relative velocities of the 82 galaxies. There are on average 7.21 non-empty clusters in a sample, the average occupancy of each cluster being 11.57.

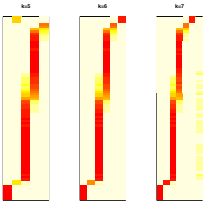

Fig. 4. The mixing matrices $\tilde{\beta}$ for the GALAXY data simulated using the Dirichlet Process Mixture. The y axis corresponds to the 82 galaxies, ordered according to their velocity (lowest velocity at the bottom). The x axis shows the cluster index of the Bernoulli mixture model. Dark shades correspond to a matrix entry of 1, while light shades correspond to zero. Here k is the number of clusters in the Bernoulli mixture model. The mixture components have been ordered for visual clarity.

results are shown in Figure 4. For $k = 7$ mixture components, one of the components turned out to be essentially empty, indicating that the data effectively exhibits six clusters. For $k = 6$ mixture components the mixing matrix looks otherwise similar, except that there is no empty component. For $k = 5$ mixture components, two of the smallest components have been merged to one.

In summary, the Bernoulli Labeling algorithm was capable of extracting the structure of six clusters from the complicated set of samples of the Dirichlet Process Mixture model.

6 Conclusions

We introduced a Bernoulli mixture model for relabeling cluster assignments in mixture models. The model is better motivated than existing solutions to the label switching problem, and outperformed them. The fully Bayesian version requires computation of posteriors for the permutation function which is manageable for models with a fixed number of clusters. For nonparametric Bayesian methods where the number of clusters varies in the MCMC samples, a fully Bayesian method should take into account splits and merges as well, which would be computationally prohibitive.

It turned out that using a Bernoulli mixture without averaging over the posterior of permutations worked very well in solving the label switching problem for nonparametric Bayesian methods, and was rather insensitive to the chosen number of clusters.

In this paper we focused on mixture models, where there is one latent variable per data point, telling which mixture component the point comes from. Furthermore, the simulations were done on one-dimensional data. Both restrictions can naturally be easily removed.

Acknowledgments. SK belongs to Finnish Centre of Excellence in Adaptive Informatics Research and KP to Finnish Centre of Excellence in Algorithmic Data Analysis Research. The work was also supported in part by the PASCAL EU Network of Excellence.

References

[1] Diebolt, J., Robert, C.P.: Estimation of finite mixture distributions through Bayesian sampling. Journal of the Royal Statistical Society. Series B (Methodological) 56(2), 275–363 (1994)
[2] Geweke, J.: Interpretation and inference in mixture models: Simple MCMC works. Computational Statistics & Data Analysis 51, 3529–3550 (2007)
[3] McLachlan, G., Peel, D.: Finite Mixture Models. Wiley Interscience, Hoboken (2000)
[4] Jasra, A., Holmes, C.C., Stephens, D.A.: Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. Statistical Science 20(1), 50–67 (2005)
[5] Stephens, M.: Bayesian methods for mixtures of normal distributions. PhD thesis, University of Oxford (1997)
[6] Celeux, G.: Bayesian inference for mixtures: The labels-switching problem. In: Payne, R., Green, P. (eds.) Proceedings of XIII Symposium on Computational Statistics (COMPSTAT 1998), Bristol, August 1998, pp. 227–232. Physica-Verlag (1998)
[7] Stephens, M.: Dealing with label switching in mixture models. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 26(4), 795–809 (2000)
[8] Celeux, G., Hurn, M., Robert, C.P.: Computational and inferential difficulties with mixture posterior distributions. Journal of the American Statistical Association 95, 957–970 (2000)
[9] Hurn, M., Justel, A., Robert, C.P.: Estimating mixtures of regressions. Journal of Computational & Graphical Statistics 12(1), 55–79 (2003)
[10] Gerber, G.K., Dowell, R.D., Jaakkola, T.S., Gifford, D.K.: Automated discovery of functional generality of human gene expression programs. PLoS Computational Biology 3(8), 148 (2007)
[11] Jasra, A.: Bayesian Inference for Mixture Models via Monte Carlo Computation. PhD thesis, Imperial College London (2005)
[12] Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics 5(1), 32–38 (1957)
[13] Liu, J.S.: Monte Carlo Strategies in Scientific Computing. Springer, Heidelberg (2001)
[14] Neal, R.M.: Markov Chain sampling methods for Dirichlet process mixture models. Journal of Computational and Graphical Statistics 9(2), 249–265 (2000)
[15] Postman, M., Huchra, J.P., Geller, M.J.: Probes of large-scale structures in the Corona Borealis region. Astrophysical Journal 92, 1238–1247 (1986)