Label Switch in Mixture Model and Relabeling Algorithm

Project for Reading Course

Prepared by: Fanfu Xie, ZhengFei Chen

Label Switch in Mixture Model and Relabeling Algorithm

Abstract

When MCMC is used to perform Bayesian analysis for mixture models, the so-call label switch problem affects the clustering analysis. If the problem is not handled properly, the ergodic average of the MCMC samples is not appropriate for the estimation of the parameters. In this paper, we will review the Label Switch problem in mixture model, discuss and implement the re-labelling algorithm suggested by Stephens. To illustrate the problem, we apply data augmentation Gaussian Mixture Model to Galaxy data with different number of components.

Introduction

Cluster analysis in data mining is to classify the observed objects into disjoint groups. Mixture of Gaussian models and Bayesian methods are popular tools to perform these kinds of analysis. However, since the component from which a particular observation comes from is unknown and the posterior distribution of Bayesian $p(\Theta | x)$ is difficult to directly estimate. To solve this problem, a latent variable z is introduced for each observed data; z is an indicator variable indicating component for each observed data; therefore, the posterior distribution $p(\Theta | x)$ can be obtained from $p(\Theta | x, z)$ by integrating out z. The model with latent variable is the data augmentation model. The latent variable z for each observed data can help to perform the estimation, but it also increases the number of dimension of the parameter space in the model. As a result, it makes the quantity analysis very difficult. One possible solution for this issue is the Marko Chain Monte Carlo (MCMC) simulation. In MCMC, the Gibber Sampler can draw samples from the posterior distribution $p(\Theta | x, z)$ and the ergodic average can be used to estimate the model parameters. When MCMC is applied to mixture model, the so-called label-switch problem takes place during MCMC simulation. The label switch problem is caused by the invariance of the likelihood function with respect to the permutation of the component labels. The Gibbs Sampler mixes up its sample's component labels when label switch happens. As a result, the ergodic average is not appropriate for the parameter estimation. In this project, we will explore the label switch problem in details. We first introduce the label switch problem and discuss its related computational features; we also explore solutions to remove the label switch from the MCMC samples: a random permutation MCMC algorithm suggested by Fruhwirth-Schnatter (2001b) and re-labeling algorithm suggested by Stephens (2000b). In this paper, Finite Gaussian Mixture Model and Gibber Sampler are used for problem illustration and Galaxy data is used as an observed dataset. The Galaxy data obtained from: "http://www.stats.bris.ac.uk/~peter/mixdata" consists of the velocities of 82 distant galaxies, and analyzed under different mixture models by several authors including Escobar and West (1995) and Phillips and Smith (1996).

In our previous project (the final project for course Monte Carlo Method), we have already introduced the data augmentation Gaussian Mixture Model and its posterior distribution. The conditional distributions for Gibbs Sampler and prior distribution are provided there too; therefore, we do not repeat them here. They are included in Appendix A for reference.

The Label Switch Problem

When Bayesian approach is applied to parameter estimation and clustering analysis for mixture models, the so-called label switch problem might occur. The problem is mainly caused by the invariance of the likelihoods with respect to the permutations of the component labels in the mixture model. As a consequence, during MCMC simulation, when the prior distributions are exchangeable, the sampler encounters symmetries of the posterior distribution (MCMC label switching). To illustrate the problem, let $x = x_1, x_2, ..., x_n$ be independent observations from a mixture density with known finite k components, we have:

$$p(x \mid \pi, \Theta) = \pi_1 f(x; \Theta_1) + ... + \pi_k f(x; \Theta_k)$$

where $\Theta_i\,(i = 1, ..., k)$ is the parameter spaces for the ith component and $\pi_i\,(i = 1, ..., k)$ is the weight parameter with constrain $\pi_i > 0$ and $\sum \pi_i = 1$.

Let $(\rho(1), ..., \rho(k))$ denote a permutation of the integers 1,…,k. and define $\rho(\pi, \Theta) = (\pi_{\rho(1),...,}\pi_{\rho(k)}, \Theta_{\rho(1),...,}\Theta_{\rho(k)})$. Then

$$p(x \mid \pi, \Theta) = \pi_1 f(x; \Theta_1) + ... + \pi_k f(x; \Theta_k)$$

$$= \pi_{\rho(1)} f(x; \Theta_{\rho(1)}) + ... + \pi_{\rho(k)} f(x; \Theta_{\rho(k)})$$

$$= p(x \mid \rho(\pi, \Theta))$$

Then the likelihood function for the n observed data is:

$$L(\Theta; x) = \prod_{i=1}^{n} p(x_i \mid \pi, \Theta) = \prod_{i=1}^{n} p(x_i \mid \rho(\pi, \Theta)) = L(\rho(\pi, \Theta); x)$$

That is the likelihood function is invariant with respect to permutations of the components.
The posterior distribution for MCMC is

$$p(\pi, \Theta \mid x) = \frac{p(\pi, \Theta, x)}{p(x)} = \frac{p(x \mid \pi, \Theta) p(\pi, \Theta)}{p(x)}$$

If the prior distribution is exchangeable, then $p(\pi, \Theta)$ = $p(\rho(\pi, \Theta))$ and the posterior distribution $p(\pi, \Theta \mid x)$ = $p(\rho(\pi, \Theta) \mid x)$ therefore, during MCMC simulation, the sampler might encounter a symmetries of the posterior distribution.

Label switch problem is crucial for Bayesian inference and computational issues (Martin et al). During the MCMC simulation, the symmetries of the posterior distribution make the MCMC samples carry little component label information. Therefore, the component labels are mixed up and cannot be distinguished from each other. As a result, the marginal on the parameters for all components is identical and the posterior expectation for the parameters is identical too. Obviously, the identical posterior expectations are not appropriate for Bayesian inference any more.

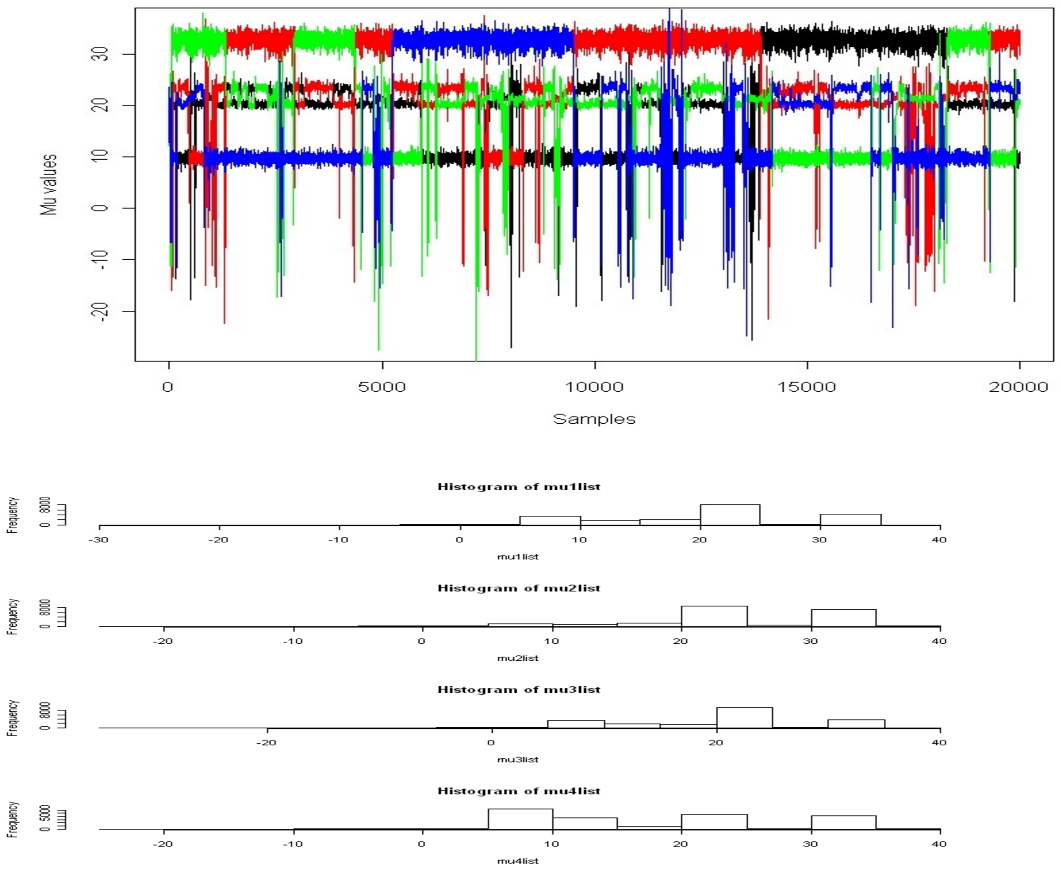

Figure One: Gibbs Sampler for Galaxy data. We fitted four-component normal mixture to the data. The black color is mu1; red is mu2; green is mu3 and blue is mu4. The parameters are mixed together for the components and the histograms for mu1, mu2 and mu3 are identical.

To demonstrate the label-switching problem, we implement the Gibbs Sampler on the galaxy data. We fit the data in four-component Gaussian Mixture Model with same variance in each component. In Figure one, label switches are clearly observed. They occur frequently among the components. The symmetric modes in the posterior distribution are clear and the Sampler visits most of the 4! (number of permutations) symmetric modes. The histograms of the first three components are identical; their mean value is 20.6, 19.4 and 19.2 respectively. The last component is different from the others; we explain it in the following paragraph.

Computational Features of Label Switch Problem

The label-switch problem in Bayesian analysis for mixture model is very clear from mathematical point of view. However, during MCMC computation, the problem is not always as expected. Given a mixture model with k components, there are $k!$ symmetric modes of the posterior distribution due to the invariance of the likelihood with respect to the permutation of the component labels. The MCMC sampler should switch from modes to modes between the iterations. If the Sampler can thoroughly and evenly travel all the $k!$ symmetric modes, the posterior expectation for each component parameters should be identical. This feature provides a useful convergence diagnostic at the simulation stage: the failure to visit the identical posterior expectations reveals that the MCMC sampler has not converged (Jasra, Holmes & Stephens).

However, the value of k! is very large as k increase and the posterior distribution surface is a very high multimodal space. It is hard for the regular Markov Chain Monte Carlo sampler to thoroughly and evenly explore the high multimodal distribution surface. The MCMC Sampler might be trapped into local modes and require an enormous number of iterations to escape from (Marin, Mengersen & Robert). When the Sampler fails to thoroughly and evenly explore the distribution surface, the unbalanced label switch takes place. According to Sylvia Fruhwirth-Schnatter, the unbalanced label switch will cause a very poor estimate in estimating the marginal density from the MCMC draws and the results might be very different from different runs of the MCMC sampler (Fruhwirth-Schnatter).

From our computational experiments, we observe two unbalanced label switch cases. In the first case, label switches only occur among some of component labels or label switches do not take place evenly among the components. For example, in Figure one, the fourth component is different from the other three because the Sampler fails to evenly visit all the modes. As another example to show the unbalanced label switch, we implement the Gibber Sampler to fit the Galaxy data using four-component Gaussian Mixture Model with different variance. From the figure two (a), we can observer that if there are only 2000 iterations in MCMC sampling, we can observe that the label switch happens between the red line and the blue line; but no label switch happens between the green line and the black line. In figure two (b), if there 20000 iterations in MCMC sampling, we can see the label switch happen a lot among the components parameters; but the MCMC sampler still fails to thoroughly and evenly travel all sample space in 4! modes, i.e.: most of the time, the green line stays near the value of 10 and occasionally travels to the value near 20; for a balanced label switch, the green line should travel almost the same number of times around the values near 10, 20, 21 and 30 and so do the other lines.

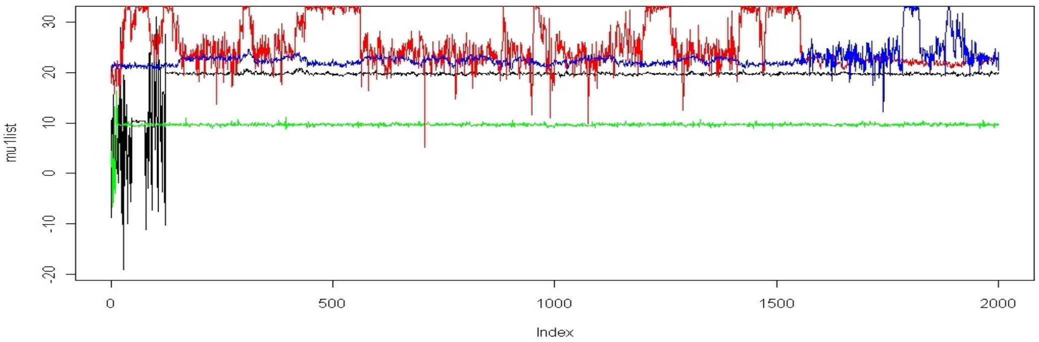

Figure Two (a): Gibbs Sampler for Galaxy data with 2000 iterations. We fitted four-component Gaussian Mixture Model with different variance. No label switches for green line and black line

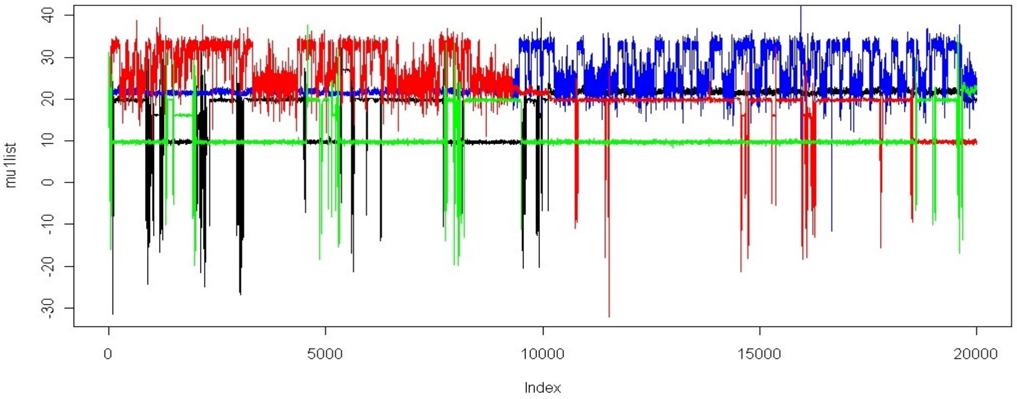

Figure Two (b): Gibbs Sampler for Galaxy data with 20000 iterations. We fitted four-component Gaussian Mixture Model with different variance. Label switches do not happen evenly for all the lines

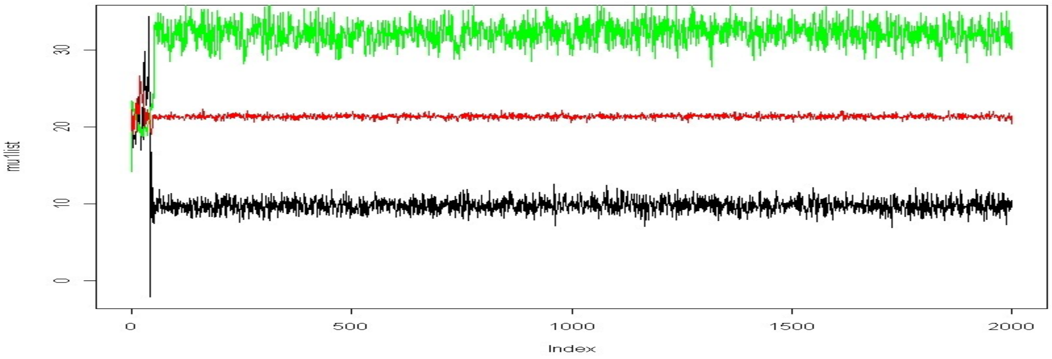

Figure Two (c): Gibbs Sampler for Galaxy data with 2000 iterations. We fitted three-component Gaussian Mixture Model with different variance. No label switches happen at all

Another case in unbalanced label switch is the label switches do not occur at all during the MCMC sampling. Under this case, the sampler might have a good performance in the sense that it has picked out the means from the data; but this good performance cannot be trusted because we can not guarantee no label-switching happened all the time. Especially, if the Sampler is run with enough iterations, we might see another symmetric mode (Jasra, Holmes & Stephens). This case can be clearly shown when we fit the Galaxy data in three components with different variance. In figure two (c), we do not see any label switch happen and the sampler performs very well in predicting the mean value of each components. However, the sampler only visits one of the 3! modes.

To provide more examples for unbalance label switch, we enlarge k to six and fit the Galaxy data with same variance for all components. From Figure three (a, b), we only observe label switching behaviours in components 1, 2, 3 and 5 but not in component 4 and 6. The histograms for components 1, 2, 3, and 5 differ from histogram for components 4 and 6 too. The Gibbs Sampler fails to thoroughly travel the posterior distribution surface. As a consequence, the MCMC can not provide the right estimations.

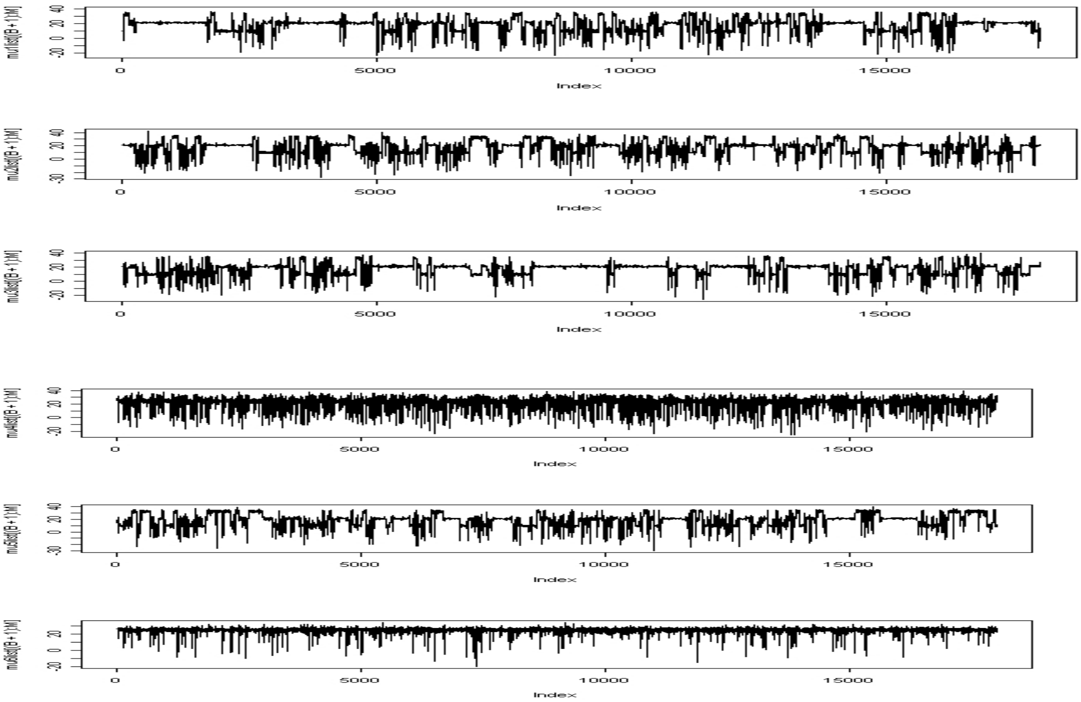

Figure Three (a): Gibbs Sampler for Galaxy data. We fitted six-component normal mixture to the data. The label switching behavior is found in 1,2,3 and 5 components parameters but not found in 4 and 6 components

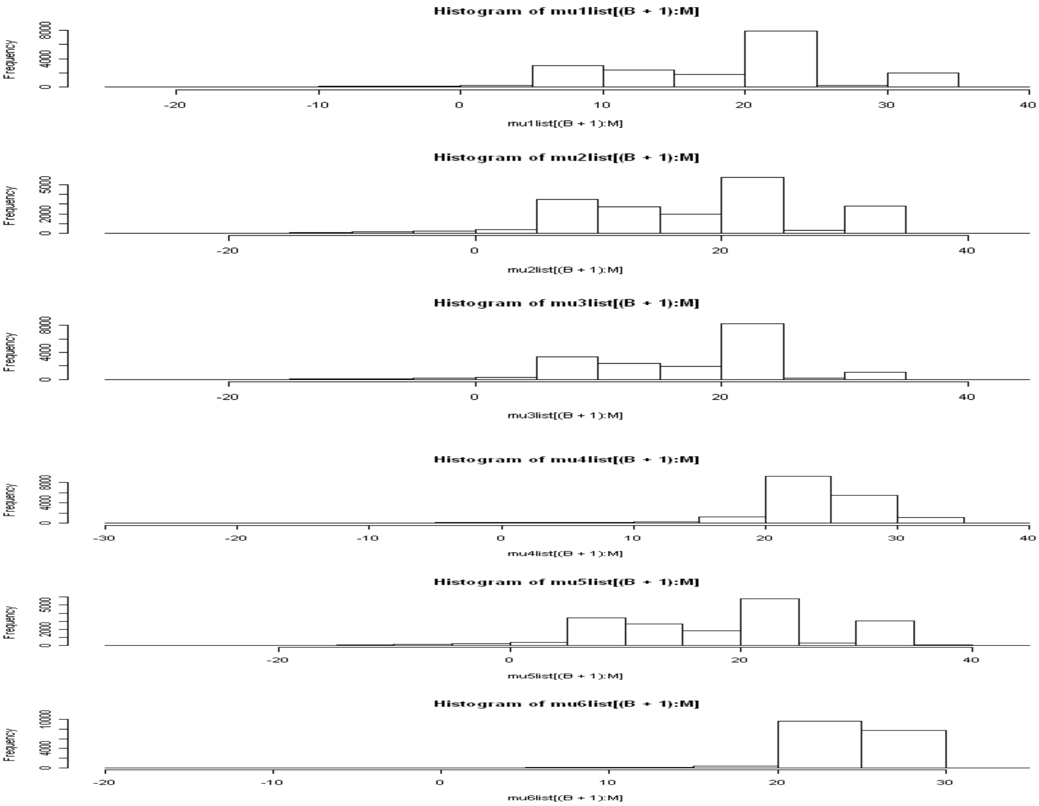

Figure Two (b): Gibbs Sampler for Galaxy data. We fitted six-component normal mixture to the data. The histograms for 1,2,3 and 5 components parameters differs from those of 4 and 6 components

Theoretically, the balanced label switch can be guaranteed if there are infinite iterations in the MCMC sampler; obviously, it is impossible for computation. In data augmentation model, the unbalance label switch is more common because the latent variable Z_i dramatically increase the dimensions of the posterior distribution. It is harder for a Sampler to visit all these dimensions.

Solutions for Label Switching Problem

Based on the above understanding, two issues should be addressed when applying MCMC simulation for Bayesian inference in mixture models. First, the MCMC sampler should be guaranteed to thoroughly and evenly travel all the $k!$ symmetric modes in the posterior distribution surface to avoid the unbalanced label switches. Secondly, remove label switches from the raw MCMC samples; that is: relabeling the components.

Random Permutation MCMC sampling (for the first issue: unbalanced label switching)

Fruhwirth-Schnatter suggests a Random Permutation MCMC sampler to solve the unbalanced label switching issue (Fruhwirth Schnatter, 2001b.) The algorithm is simple but efficient. It randomly draws one of $k!$ permutation label order and applies this label order to substitute the current MCMC sample's component order. For example, let $(\theta_1,\ldots,\theta_k)$ is the current parameter sample from Gibbs Sampler and $(\rho(1),\ldots,\rho(k))$ is the drawn permutation order of the labels, then substitute the $(\theta_1,\ldots,\theta_k)$ by $(\theta_{\rho(1)},\ldots,\theta_{\rho(k)})$ (Fruhwirth Schnatter, 2001b.). When k is large, the algorithm performs the exchanges of the labels with high probability ($1-\frac{1}{k!}$). This guarantees all the $k!$ modes being fully explored. Fruhwirth Schnatter has more discussions about theoretical properties of permutation sampling (Fruhwirth Schnatter 2001b). Figure four shows the Random Permutation samples thoroughly travel the $k!$ modes for the galaxy data with six components.

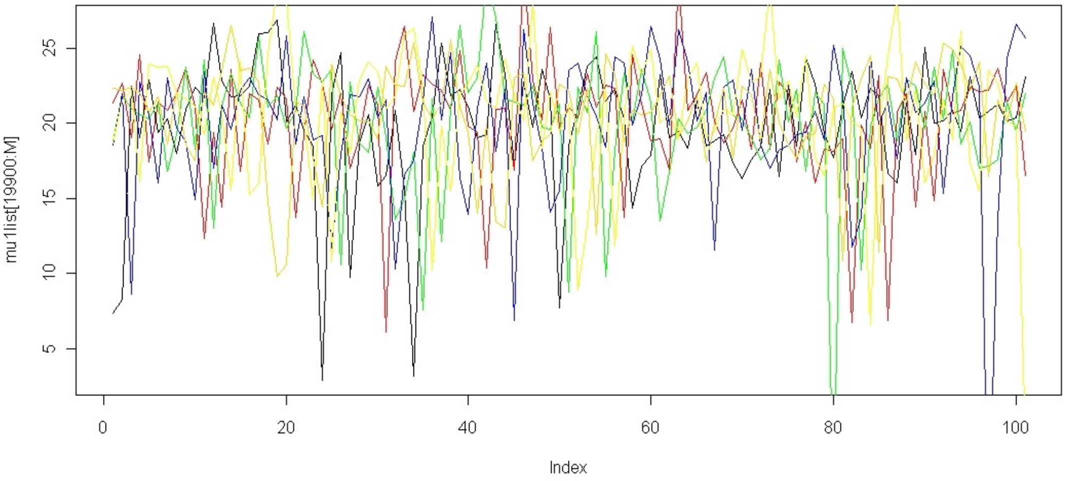

Figure Four: The Random Permutation Sampler is implemented for Galaxy data and the balanced label switching is guaranteed. The densities for each component parameter are identical (with mean value around 19.8)

A relabeling algorithm (for the second issue: label switching)

Sperrin, Jaki and Wit summary three kinds of strategies to re-label the MCMC samples so that the affect of label-switching can be removed (2009). The first and the easiest one is identifiability constraint. A constraint is imposed to the parameter space so that only one of specific permutation order for the samples satisfies it. For example, a possible constraint can be set as: $\mu_1 < \mu_2 < \mu_3 ... < \mu_k$. Although the identifiability constraint is simple, sometimes, it is not easy to find a proper constraint; for instance, in high dimension cases, it is hard to find a way to define $\mu_1 < \mu_2 < \mu_3 ... < \mu_k$. Moreover, many choices of identifiability constraint will be ineffective in removing the symmetry in the posterior distribution, and when the constraint is inappropriate, label-switching problem may remain after imposing an identifiability constraint (Stephens 1997). The second strategy is the deterministic relabeling algorithm which re-labels each sample in the MCMC output so that the expectation of the pre-defined loss function can be minimized. The third one is probabilistic approaches which recognize the uncertainty during the component relabeling for each MCMC output sample. In this paper, we will discuss the deterministic relabeling algorithm and implement a loss function suggested by Stephens.

When applying MCMC samplers in mixture models, the main problem is the likelihood is invariant under all permutations of the parameters. Stephens (2000) suggests a relabeling algorithm to solve the problem. According to Stephens, a loss function is defined on actions and real parameters; that is: $F : A \times \Theta \to R$. A single action α is chosen to handle the MCMC outcome samples so that the defined loss function can be minimized. The meaning of $F(\alpha, \theta)$ is the loss when the action α is chosen to manipulate the samples given the true parameter value θ. What's more, the chosen action and the defined loss function should be invariant under each permutation of the parameter samples. Therefore, the basic relabeling algorithm is:

Step 1: Given the observed data x, choosing the action $\hat{a}$ which minimize the posterior expected loss: $\Re(a) = E\{F(a;\theta) \mid x\}$ and

Step 2: For each MCMC outcome sample θ, $F(\alpha;\theta) = \min_{v}\{F_0(a; v(\theta))\}$, where $v(\theta)$ is the permutation of θ.

In step 1, given the MCMC outcome samples, the posterior expected loss can be estimated by the Monte Carlo Risk; that is

$\tilde{\Re}(a) = \frac{1}{N}\sum_{t=1}^{N} \min_{v_t}\left[F_0\left(a; v_t\left(\theta^{(t)}\right)\right)\right]$, where N is the number of samples in MCMC outcomes (after burn-in)

The above two steps are iterated until the permutation for each samples in MCMC converged into a fixed point. According to Stephens, the algorithm is guaranteed to convergence because there are only finite numbers of permutation for each MCMC sample and $\tilde{\Re}$ is decreased in each iteration (Stephens, 2000).

For the K components cluster analysis in mixture model, a natural way to choose the action is to define a matrix $Q = (q_{ij})$ where q_{ij} represents the probability that observation i is assigned to group j (so each row sums to 1) (Stephens, 2000). Therefore, Q can be interpreted as a distribution on k-component clustering of the data.

A classification probability matrix P(θ) is defined as:

$$p_{ij}(\theta) = \Pr(z_i = j \mid x, \pi, \mu, S) = \frac{\pi_i f(x_i; \mu_j, S)}{\sum_l \pi_l f(x_i; \mu_l, S)}$$

P(θ) can be considered as the true probability distribution on the clusters given x, π, μ, S; therefore, the loss function can be defined as the Kullback-Libler distance from the distribution P to the distribution Q (Stephens, 2000). That is:

$$L_0(Q;\theta)=\sum_{z_1=1}^{k}\dots\sum_{z_n=1}^{k}p_{1z_1}(\theta)\dots p_{nz_n}(\theta)\log\frac{p_{1z_1}(\theta)\dots p_{nz_n}(\theta)}{q_{1z_1}\dots q_{nz_n}}=\sum_{i=1}^{n}\sum_{j=1}^{k}p_{ij}(\theta)\log\frac{p_{ij}(\theta)}{q_{ij}}$$

Therefore, Stephens' relabeling algorithm for cluster analysis in K-component mixture models is as followings:

Set $v_1,...,v_N$ to the identity permutation; iterate the following two steps until a fixed point is reached.

[Step 1] Choose $\hat{Q}=(\hat{q}_{ij})$ to minimize

$$\sum_{t=1}^{N}\sum_{i=1}^{n}\sum_{j=1}^{k}p_{ij}(v_t(\theta^{(t)})\log\frac{p_{ij}(v_t(\theta^{(t)})}{\hat{q}_{ij}}$$

Actually, this is achieved by

$$\hat{q}_{ij}=\frac{1}{N}\sum_{t=1}^{N}p_{ij}(v_t(\theta^{(t)}))$$

[Step 2] For t=1,...,N choose v_t to minimize

$$\sum_{i=1}^{n}\sum_{j=1}^{k}p_{ij}(v_t(\theta^{(t)}))\log\frac{p_{ij}(v_t(\theta^{(t)}))}{\hat{q}_{ij}}$$

When k is small, this can be easily achieved by examining all k! possibilities for each v_t (Stephens, 2000).

Running Time of the Re-labelling Algorithm

The relabeling algorithm can effectively solve the label switch problem. However, the running time for the algorithm is $O(T*N*(k!)*M)$ which is exponential of the value K. T is the number of iterations before a fixed permutation point is reached for each sample. N is number of observations and M is the number of MCMC samples. Both N and M significantly affect the running time when they are large. K is the number of components; according to Sterling's approximation, $k!\approx\sqrt{2\pi k}(k/e)^k$ which is exponential in running time. Therefore, when implementing the relabeling algorithm, we need to find a way to reduce the running time. One way we tried is as following:

1) Randomly select 2000 samples from the M samples when M is very large and run the relabeling algorithm on the 2000 samples only.
2) Pre-compute the P matrix for all the 2000 samples on all K! permutations and save the results into the memory; therefore, in each iteration, it is not necessary to re-compute the P matrix. The access to the data structure (i.e.: array) in memory is only O(1).

Obviously, step two is only a trade-off between the memory space and the running time. When K is large, we will run out of the memory quickly. However, the method is quite effective when we implement the Galaxy data with four components. We get the result very quickly (in half an hour). When K is large, Stephens suggests a way to convert the problem into integer programming problem for which efficient algorithm exist. (Stephens, 2009)

Implementation of relabeling algorithm for Galaxy Data

We implemented the relabeling algorithm for the Galaxy data. We apply the Gaussian Mixture Model with four components to fit the data. Before the relabeling algorithm, Random Permutation MCMC algorithm is applied to the samples and 2000 samples are randomly selected for the implementation of relabeling algorithm. Figure Five is the samples after running the re-labelling algorithm. We can see that the re-labelling algorithm successfully re-labels the mixed parameter samples and the label-switch disappears from the plot. The estimated value for the mean value of each component is: 9.725554, 19.21909, 22.23947 and 32.56761 respectively. They are correctly estimated by the algorithm. In our example, it only takes four iterations for all the samples converging to their fixed permutation point. In our implementation, the total running time is about 30 minutes.

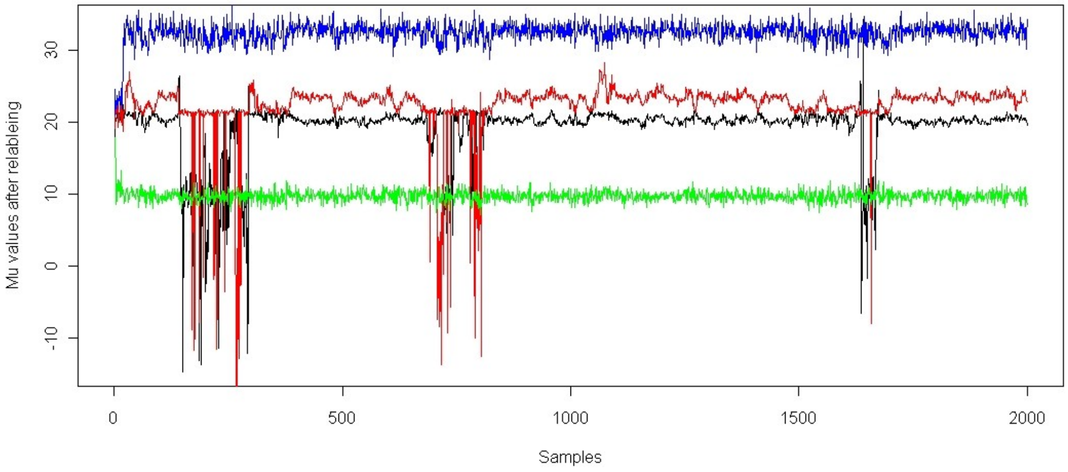

Figure Five: The results of re-labeling algorithm run on the samples from Random Permutation MCMC Sampler. All the label switches are removed and the posterior expectations provide good estimations for the component parameters. This one is run on Galaxy data with four components.

Implementation of relabeling algorithm for Synthetic Data

Besides the Galaxy data, we run the algorithm for a synthesis data as well. We simulated 50 data points from the four-component Gaussian Mixture model suggested by Jasra, Holmes and Stephens (2005) :

$$x_i = 0.25N(-3, 0.55^2) + 0.25N(0, 0.55^2) + 0.25N(3, 0.55^2) + 0.25N(0.55^2)\,.$$

Figure six (a) is the raw data from the Gibbs Sampler with 20000 iterations. We observe no label switch taken place and the sampler performs perfectly in the sense that the mean value of each component is well estimated. Such as Jasra, Holmes and Stephens point out, the good performance cannot be trusted and the sampler is not stable for the runs (Jasra, Holmes and Stephens).

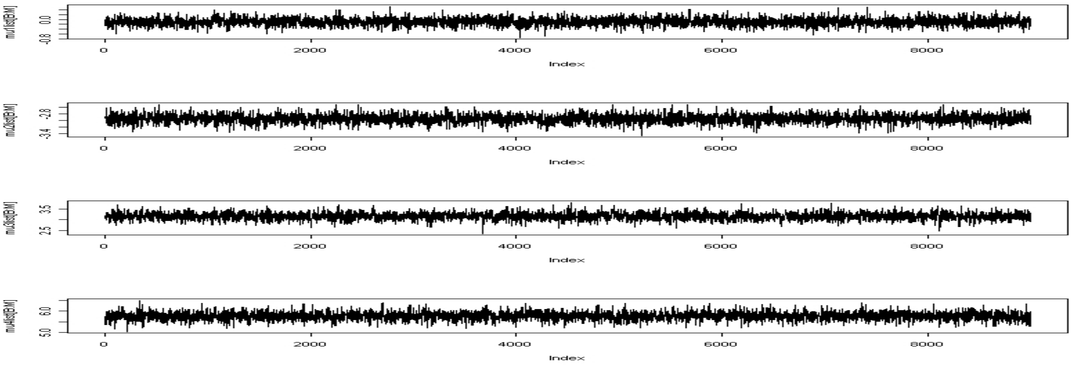

Figure Six (a): Raw MCMC samples for synthesis data with 4 components in Gaussian Mixture model. No label switches occur.

We apply the Random Permutation MCMC algorithm for the data to force the balance label switch. From the figure six (b), we see the balance label switch. The Sampler smoothly goes through all the 24 (4!) modes in the posterior distribution. We get the estimates of 1.717891, 1.586022, 1.728616 and 1.628728 for each component respectively; they are identical each other.

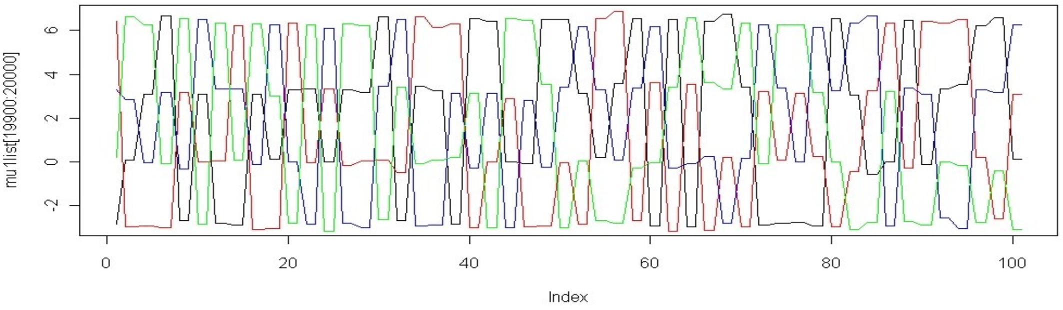

Figure Six (b): Random Permutation MCMC samples for synthesis data with 4 components in Gaussian Mixture model. The Sampler thoroughly and evenly visits all the posterior distribution space. The samples mix up the component labels.

Based on the samples from Random Permutation MCMC, we randomly draw 2000 samples to run the re-labelling algorithm. From the figure, we see the algorithm correctly re-label the samples. The estimate for each component is: -3.173219, -0.2285065, 3.254033 and 6.238465. It takes three iterations for all the samples convergent to its fixed permutation point. The running time for this example is about 20 minutes.

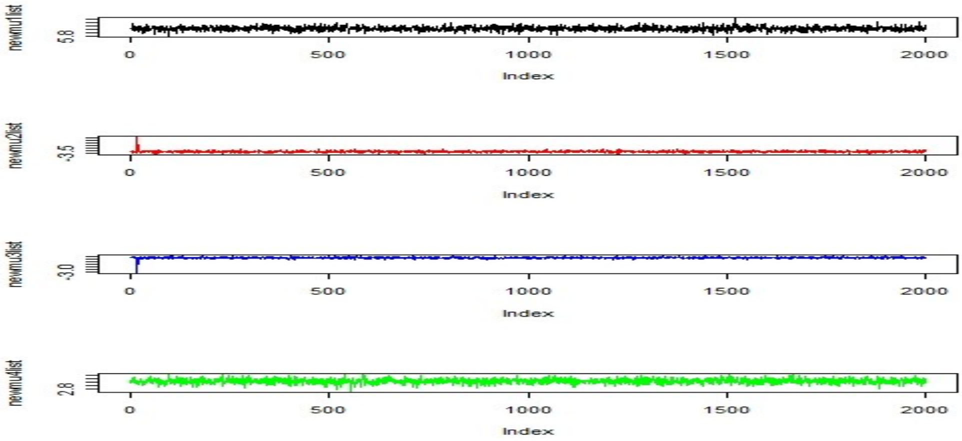

Figure Six (c): The relabeling algorithm successfully removes the label switches from the Random Permutation MCMC samples in (b) and the ergodic averages provide good estimations for component parameters.

Relabeling algorithm for component with high dimensions

This part is included in "Relabelling Algorithm in High Dimension".

Summary and Future Study

In this paper, we explore the so-called label switch problem in details. The problem is caused by the invariance of the likelihood function to the permutation of the labels for the components in mixture models and thus the Bayesian and MCMC approach is affected. We also discuss the unbalanced label switch -- the computational features of the label switch problem. To remove the unbalance and balance label switch, the Random Permutation MCMC algorithm and re-labelling algorithm are discussed. The re-labelling algorithm is to define a loss function and find a way to re-label the samples so that the loss function is minimized. For illustration, we implement the algorithms for Galaxy data and the simulated data using data augmentation Gaussian Mixture Models. The results for both data sets look good and re-labelling algorithm successfully remove the label switches from the samples.

Currently, some authors (Jasra) introduce the probabilistic approaches to the re-labelling algorithm. The basic idea of the probabilistic approaches is to introduce the uncertainty in selecting the labels on each iteration of the MCMC output (Sperrin, Jaki and Wit, 2009). The EM (Expectation-Maximization) algorithm is used to implement the probabilistic relabeling and the missing data in EM algorithm is the order of the components at each iteration of the MCMC (Sperrin, Jaki and Wit, 2009). The probabilistic relabeling is attractive because the uncertainty in relabeling is recognized and under the control while the deterministic relabeling ignores those uncertainty. For the future study, we will explore these topics in details.

Acknowledgements

We feel short of words to express our gratefulness to Prof. Rosenthal for his guidance in this project. Without his help, this project is impossible. His enthusiasm toward research and education will certainly encourage us in our future work and study.

References

Marin, Jean-Michel, Mengersen, Kerrie, Robert, Christian P. (2009): Bayesian Modelling and Inference on Mixtures of Distributions. Available from http://www.ceremade.dauphine.fr/~xian/mixo.pdf

Jasra, A., Holmes, C.C., Stephens, D.A.: Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modelling. Stat. Sci. 20, 50-67 (2005)

Sperrin, M., Jaki, T., Wit, E.: Probabilistic relabeling strategies for the label switching problem in Bayesian mixture models. Stat Comput, DOI 10.1007/s11222-009-9129-8

Fruhwirth-Schnatter, Sylvia (2001b). MCMC Estimation of Classical and Dynamic Switching and Mixture Models. Journal of the American Statistical Association 96, 194-209.

Fruhwirth-Schnatter, Sylvai: Finite Mixture and Markov Switching Models. Springer 2006.

Stephens, M. (2000) Bayesian analysis of mixtures with an unknown number of components - an alternative to reversible jump methods. *Annals of Statistics.* Available from http://www.stats.ox.ac.uk/~stephens

Lopes, H. (2008) Multivariate Mixture of Normals. Available from http://faculty.chicagobooth.edu/hedibert.lopes/teaching/41913-Winter2008/

Celeux, G.(1997) Contribution to the discussion of paper by Richardson and Green (1997).

Journal of the Royal Statistical Society, series B, 59, 775-776

Celeux, G. (1998) Beyesian inference for mixtures: The label-switching problem. In COMPSTAT 98 (Eds R. Payne and P.Green), pp. 227-232. Physica-Verlag.

Celeux, G., Hurn, M. And Robert, C.P.(2000) Computational and inferential difficulties with mixture posterior distributions. Available from the MCMC Preprint service.

Stephens, M. (1997) Contribution to the discussion of paper by Richardson and Green (1997). *Journal of the Royal Statistics Society, series B*, 59, 768-769

Stephens, M. (1997) Bayesian Methods for Mixtures of Normal Distributions. Ph.D. Thesis, University of Oxford. Available from http://www.stats.ox.ac.uk/~stepens.

Rosenthal, J. (2009) Notes on MCMC. Available from http://www.probability.ca/jeff/teaching/.

Appendix A: the Gaussian mixture model and its Gibbs Sampler

Introduction

In this paper, we first will give a Gibbs sampler for a mixture normal model with known components by introducing a latent indicator variable and posing Dirichlet and Wishart priors to our estimate parameters accordingly. After that, we will discuss "label-switch" problem, and describe a relabelling algorithm. Also a realization of the modified algorithm will be given. In the discussion, we will give a new proposal to this kind of questions.

Gibbs Sampler

Let $x_1,...,x_n$ (each x_i is a p-vector) be a random sample from a mixture of normals, i.e.

$$p(x \mid \mu,\pi,S)=\sum_{i=1}^{K}\pi_k dN(x \mid \mu_k,S) \quad (1)$$

where $dN(x \mid \mu,S)$ denote the density function of multivariate normal with mean μ and variance-covariance matrix S. K is the number of components(fixed and known). $\mu=(\mu_1,...,\mu_K)$ and $\pi=(\pi_1,...,\pi_K)$. $\pi_i \ge 0, i=1,...,K$ and $\sum_{i=1}^{K}\pi_i=1$.

We want to write down the likelihood function for this sample. However, since we do not know the component from which a particular observation x_i comes from, we need to introduce, for each observation x_i, an indicator variable z_i with $P(z_i = j) = \pi_j$ for all i=1,…n and j=1,…,K. Then the likelihood function can be written as:

$$p(x \mid z, \mu, S) = \prod_{j=1}^{K} \prod_{i \in I_j} dN(x_i \mid \mu_j, S)$$

where $x = (x_1, ..., x_n)$, $z = (z_1, ..., z_n)$, $I_j = \{i : z_i = j, i = 1, ..., n\}$ (i.e. I_j counts the number of observations that come from component j). η_j is the number of elements in

$$I_j \text{ (i.e.} \eta_j = \sum_{i=1}^{n} I_{\{z_i = j\}} \text{).}$$

We put priors on π, $\Phi = S^{-1}$, and μ respectively:

The prior distribution of $\pi = (\pi_1, ..., \pi_K)$ is a Dirichlet distribution with parameter

$$\eta_0 = (\eta_{01}, ..., \eta_{0K}) : f_\pi(\pi \mid \eta_0) = \frac{1}{B(\eta_0)} \prod_{i=1}^{K} \pi_i^{\eta_{0i} - 1},$$

here we do not need to bother with $B(\eta_0)$ since it is just a normalizing constant.

The prior distribution of $\Phi = S^{-1}$ is a Wishart distribution with parameter v_0 (df) and $v_0^{-1}\Phi_0$:

$$f_\Phi(\Phi \mid v_0, v_0^{-1}\Phi_0) = \frac{|\Phi|^{\frac{v_0 - p - 1}{2}} e^{-\frac{1}{2} trace(v_0 \Phi_0^{-1} \Phi)}}{2^{\frac{v_0 p}{2}} |v_0^{-1}\Phi_0|^{\frac{v_0}{2}} \Gamma_p(\frac{v_0}{2})}$$

The prior distribution for each μ_k is a multivariate normal with known mean μ_{k0} and variance-covariance matrix V_{k0}.

Given π, z has a multinomial distribution $z \mid \pi \sim M(K, \pi)$.

Now, we are ready to write down the joint density of (π, z, μ, S) "

$$f(\pi, z, \mu, \Phi) = f_\pi(\pi \mid \eta_0) f_z(z \mid \pi) \prod_{j=1}^{K} dN(\mu_j \mid \mu_{j0}, V_{j0}) f_\Phi(\Phi \mid v_0, v_0^{-1}\Phi_0) \prod_{j=1}^{K} \prod_{i \in I_j} dN(x_i \mid \mu_j, \Phi)$$

$$= C(\prod_{i=1}^{K} \pi_i^{\eta_{0i} - 1})(\prod_{i=1}^{K} \pi_i^{n_i})(|\Phi|^{\frac{v_0 - p - 1}{2}} e^{-\frac{1}{2} trace(v_0 \Phi_0^{-1} \Phi)})(e^{-\frac{1}{2}\sum_{j=1}^{K} (\mu_j - \mu_{j0})^T V_{j0}^{-1} (\mu_j - \mu_{j0})}) |\Phi|^{\frac{n}{2}} (e^{\sum_{j=1}^{K} \sum_{i \in I_j} (x_i - \mu_j)^T \Phi (x_i - \mu_j)})$$

We shall use Gibbs sampler to sample from the above density. So we next compute the conditional distribution of each parameter given the current value of all others.

A. condition distribution of π

$$f(\pi \mid z,\mu,\Phi)=\frac{f(\pi,z,\mu,\Phi)}{f(z,\mu,\Phi)}=\frac{(\prod_{i=1}^{K}\pi_i^{\eta_{0i}-1})(\prod_{i=1}^{K}\pi_i^{\eta_j})}{\int(\prod_{i=1}^{K}\pi_i^{\eta_{0i}-1})(\prod_{i=1}^{K}\pi_i^{\eta_j})d\pi}=C\prod_{i=1}^{K}\pi_i^{\eta_{0i}+\eta_j-1}$$

So the conditional distribution of π is again Dirichlet with parameter $\eta_0+\eta$, where

$\eta_0=(\eta_{01},\ldots,\eta_{0K})$ and $\eta=(\eta_1,\ldots,\eta_K)$.

B. Conditional distribution of Φ

$$f(\Phi \mid z,\mu,\pi)=\frac{f(\pi,z,\mu,\Phi)}{f(z,\mu,\pi)}=\frac{(|\Phi|^{\frac{v_0-p-1}{2}}e^{-\frac{1}{2}trace(v_0\Phi_0{}^{-1}\Phi)})|\Phi|^{\frac{n}{2}}(e^{\sum_{j=1}^{K}\sum_{i\in I_j}(x_i-\mu_j)^T\Phi(x_i-\mu_j)})}{\int(|\Phi|^{\frac{v_0-p-1}{2}}e^{-\frac{1}{2}trace(v_0\Phi_0{}^{-1}\Phi)})|\Phi|^{\frac{n}{2}}(e^{\sum_{j=1}^{K}\sum_{i\in I_j}(x_i-\mu_j)^T\Phi(x_i-\mu_j)})d\Phi}$$

$$=C|\Phi|^{\frac{v_0+n-p-1}{2}}e^{-\frac{1}{2}trace\{[v_0\Phi_0{}^{-1}+\sum_{j=1}^{K}\sum_{i\in I_j}(x_i-\mu_j)(x_i-\mu_j)^T]\Phi\}}$$

(here, we use the fact:

$(x_i-\mu_j)^T\Phi(x_i-\mu_j)=trace\{(x_i-\mu_j)^T\Phi(x_i-\mu_j)\}=trace\{(x_i-\mu_j)(x_i-\mu_j)^T\Phi\}$)

So the conditional distribution of Φ is again Wishart with parameter $v_1=v_0+n$ and

$$v_1^{-1}\Phi_1=v_0\Phi_0{}^{-1}+\sum_{j=1}^{K}\sum_{i\in I_j}(x_i-\mu_j)(x_i-\mu_j)^T$$

C. Condition distribution of μ

For each j in {1,…,K}

$$f(\mu_j \mid z,\Phi,\pi)=\frac{f(\pi,z,\mu,\Phi)}{f(\pi,z,\mu_{(j)},\Phi)}=\frac{e^{\frac{1}{2}\{(\mu_j-\mu_{j0})^TV_{j0}{}^{-1}(\mu_j-\mu_{j0})+\sum_{i\in I_j}(x_i-\mu_j)^T\Phi(x_i-\mu_j)\}}}{\int e^{\frac{1}{2}\{(\mu_j-\mu_{j0})^TV_{j0}{}^{-1}(\mu_j-\mu_{j0})+\sum_{i\in I_j}(x_i-\mu_j)^T\Phi(x_i-\mu_j)\}}d\mu_j}$$

We can see that the conditional distribution of μ_j is again Gaussian. To identify its mean and variance, we set $\mu_j\sim N(\mu_{j1},V_{j1})$, then we have:

$V_{j1}{}^{-1}=V_{j0}{}^{-1}+\eta_j\Phi$, so $V_{j1}=(V_{j0}{}^{-1}+\eta_j\Phi)^{-1}$

$$\mu_{j1}=V_{j1}[\Phi(\sum_{i\in I_j}x_i)+V_{j0}{}^{-1}\mu_{j0}]$$

D. Conditional distribution of z

The conditional distribution of z is a little bit tricky since z does not explicitly appear in the joint density.

$$P(z_i = j \mid x, z_{(i)}, \mu, \pi, \Phi) = P(z_i = j \mid x, z_{(i)}) = \frac{P(z_i = j, z_{(i)}, x)}{P(z_{(i)}, x)}$$

$$= \frac{P(z_i = j, z_{(i)}, x)}{P(z_{(i)}, x_{(i)})} \frac{P(z_{(i)}, x_{(i)})}{P(z_{(i)}, x)} = C \frac{P(z_i = j, z_{(i)}, x)}{P(z_{(i)}, x_{(i)})}$$

$$= CP(z_i = j \mid z_{(i)}, x_{(i)}) P(x_i \mid z_i = j, z_{(i)}, x_{(i)}) = CP(z_i = j) P(x_i \mid z_i = j, z_{(i)}, x_{(i)})$$

$$= C\pi_j P(x_i \mid z_i = j, z_{(i)}, x_{(i)})$$

where $P(x_i \mid z_i = j, z_{(i)}, x_{(i)}) = \int P(x_i \mid z_i = j, z_{(i)}, x_{(i)}, \mu) P(\mu \mid z_{(i)}, x_{(i)}) d\mu$

$$= \int P(x_i \mid z_i = j, \mu_j) P(\mu_j \mid z_{(i)}, x_{(i)}) d\mu_j \times \prod_{k \neq j} \int P(\mu_k \mid z_{(i)}, x_{(i)}) d\mu_k$$

$$= \int P(x_i \mid z_i = j, \mu_j) P(\mu_j \mid z_{(i)}, x_{(i)}) d\mu_j$$

$$= \int dN(x_i \mid \mu_j, S) P(\mu_j \mid z_{(i)}, x_{(i)}) d\mu_j$$

By similar derivation as in part C, we know that:

$$P(\mu_j \mid z_{(i)}, x_{(i)}) = dN(\mu_j \mid \mu_{j1,i}, V_{j1,i})$$

where $V_{j1,i} = (\eta_{j,i}\Phi + V_{j0}{}^{-1})^{-1}$, $\mu_{j1,i} = V_{j1,i}[\Phi(\sum_{l \in I_{j,i}} x_l) + V_{j0}{}^{-1}\mu_{j0}]$ and

$I_{j,i} = \{l : z_l = j, l \in \{1,...,n\} \setminus \{i\}\}$, $\eta_{j,i}$ is size of $I_{j,i}$.

So

$$P(x_i \mid z_i = j, z_{(i)}, x_{(i)}) = \int dN(x_i \mid \mu_j, S) dN(\mu_j \mid \mu_{j1,i}, V_{j1,i}) d\mu_j = CdN(x_i \mid \mu_{j1,i}, V_{j1,i} + \Phi^{-1})$$

So

$$P(z_i = j \mid x, z_{(i)}, \mu, \pi, \Phi) = C\pi_j dN(x_i \mid \mu_{j1,i}, V_{j1,i} + \Phi^{-1})$$

Given the above derivation, we are ready to write a Gibbs sampler algorithm to sample from the posterior distribution of (π, S, μ, z).