Re: MH-within-Gibbs for asymmetric Gaussian mixture

Fu Amos

Mon 2017-07-10 9:20 PM

To: Nizar Bouguila <nizar.bouguila@concordia.ca>;

Dear Doctor,

The implementation of MCMC for asymmetric Gaussian mixture model is finished. Compared to symmetric MCMC, the main differences including:

- **Defined PDF of Asymmetric Model**

Based on the PDF formula proposed by Tarek in his paper <Background subtraction using finite mixtures of asymmetric Gaussian distributions and shadow detection>, I realized I can modify it in order to use it in MCMC algorithm in a dimension-by-dimension way:

$$P(z_i^{(t)} = j) \propto p_j^{(t-1)} \prod_{k=1}^{d} \frac{1}{(\sigma_{l_{jk}}^{(t-1)} + \sigma_{r_{jk}}^{(t-1)})} \times$$

$$\begin{cases} \exp\left[-\frac{(X_k - \mu_{jk}^{(t-1)})^2}{2(\sigma_{l_{jk}}^{(t-1)})^2}\right] & if\ X_k < \mu_{jk}^{(t-1)} \\ \exp\left[-\frac{(X_k - \mu_{jk}^{(t-1)})^2}{2(\sigma_{r_{jk}}^{(t-1)})^2}\right] & if\ X_k > \mu_{jk}^{(t-1)} \end{cases}$$

Using this formula and dirichlet distribution, we can generate the membership vector for every jump and the acceptance ration R for decision making step.

- **Sampling Test Data**

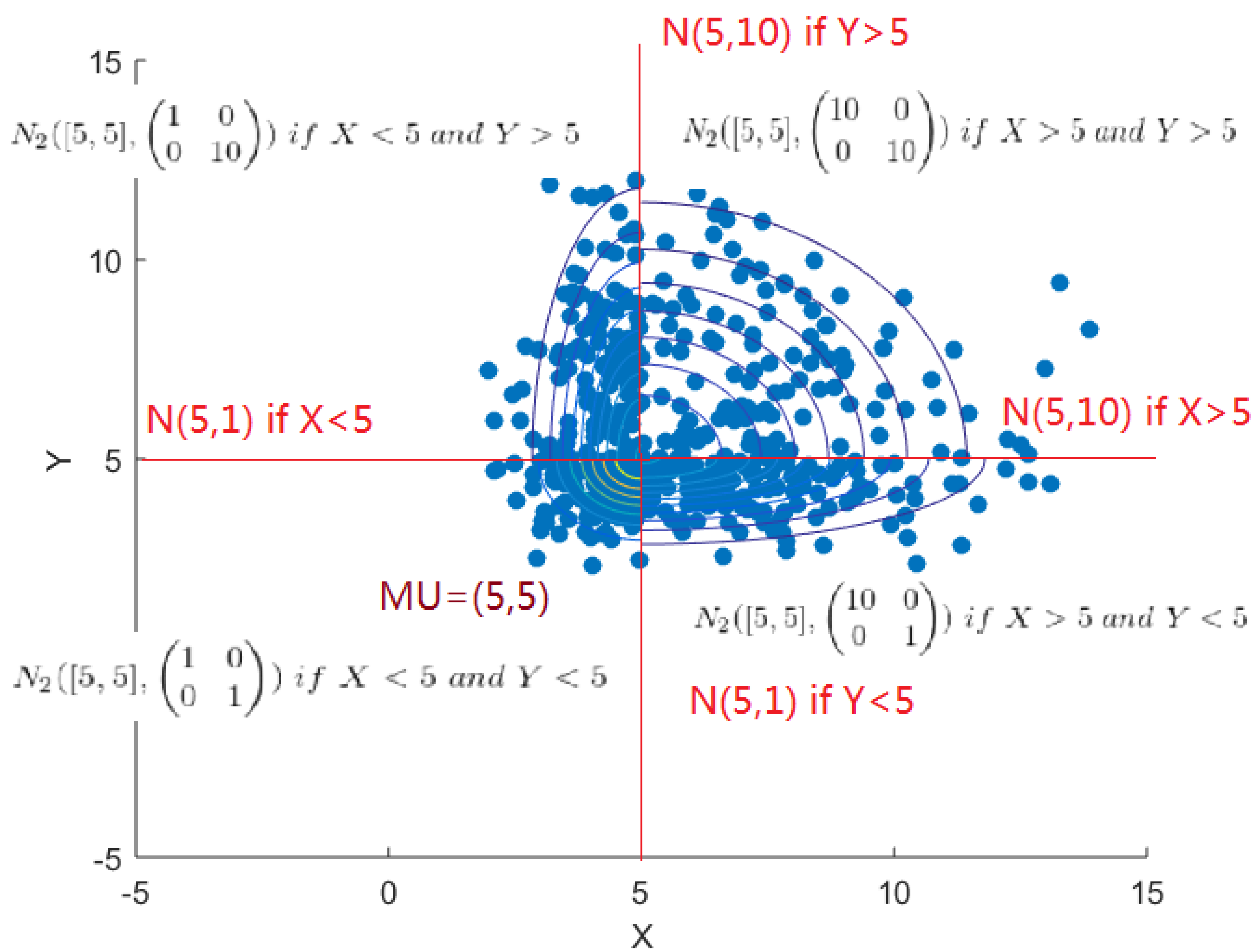
For clustering purpose, the first step is to generate enough data points from asymmetric model. I realized that high-dimensional Gaussian distributions can be break down to 1D Gaussian. For example 2D Gaussian can be written like this:

5

Let $x \sim N_2(\mu, \Sigma)$ where $\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$. Then det($\Sigma$) = $\sigma_1^2\sigma_2^2$

$$
\begin{aligned}
&\frac{1}{\sqrt{(2\pi)^2|\Sigma|}}\exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \\
=&\frac{1}{2\pi\sigma_1\sigma_2}\exp\left\{-\frac{1}{2}[x_1-\mu_1 \quad x_2-\mu_2]\begin{bmatrix} 1/\sigma_1^2 & 0 \\ 0 & 1/\sigma^2 \end{bmatrix}\begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}\right\} \\
=&\frac{1}{2\pi\sigma_1\sigma_2}\exp\left\{-\frac{1}{2}\begin{bmatrix} \frac{x_1-\mu_1}{\sigma_1^2} & \frac{x_2-\mu_2}{\sigma_2^2} \end{bmatrix}\begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}\right\} \\
=&\frac{1}{2\pi\sigma_1\sigma_2}\exp\left\{-\frac{(x_1-\mu_1)^2}{2\sigma_1^2}-\frac{(x_2-\mu_2)^2}{2\sigma_2^2}\right\}.
\end{aligned}
$$

This feature gave me an opportunity to simply use two 1D Gaussian with same MU but different SIGAM to simulate a 2D asymmetric Gaussian. Let's look at a sampling example for instance:



So we can easily generate high-dimensional asymmetric Gaussian by conditionally combine multiple 1D Gaussians. Since the accumulative probability of both lift and right parts is equal to 50% so I assume the data point amount in the 4 regions are the same and conditionally generate all the data points region-by-region.

- **Proposal Distributions for MU and SIGMA**

With the membership vector in hand, the next step will be generating new parameters from proposal distribution. In my implementation I selected 2D normal distribution as the proposal for MU:

$$q(\mu(t)|\mu(t-1)) =)N_2(\mu(t-1), \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix})$$

and 1D Gaussian for both lift and right part of every dimension of SIGMA:

$$q(\sigma_{ld}^{(t)}|\sigma_{ld}^{(t-1)}) =)N(\sigma_{ld}^{(t-1)}, 1)$$

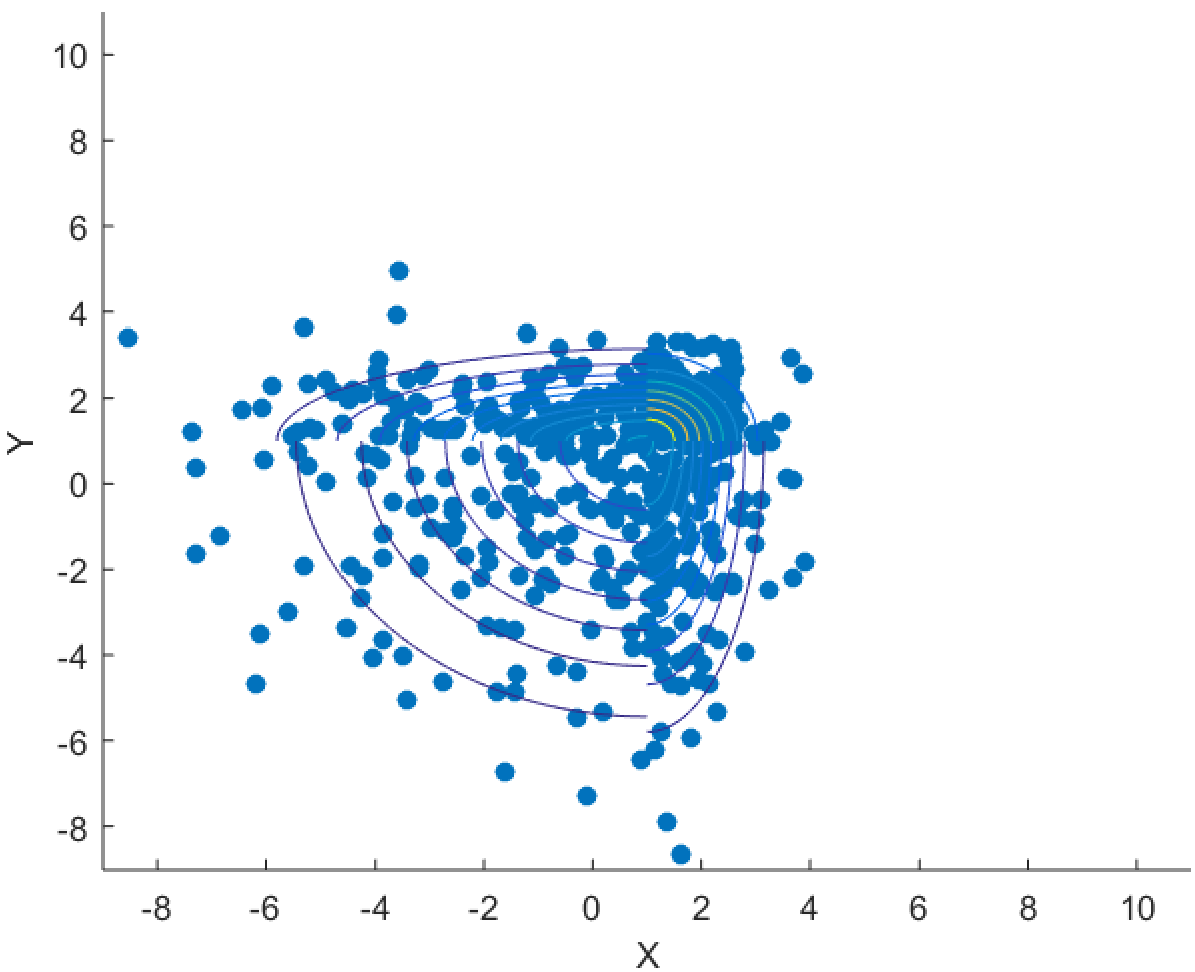
and

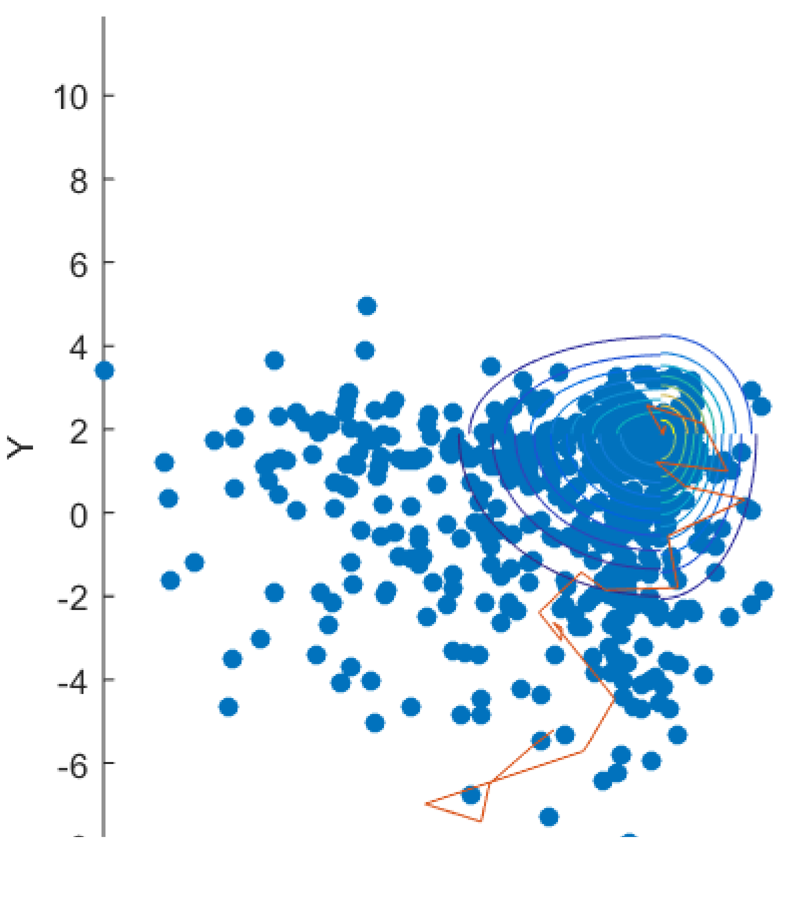
$$q(\sigma_{rd}^{(t)}|\sigma_{rd}^{(t-1)}) =)N(\sigma_{rd}^{(t-1)}, 1)$$

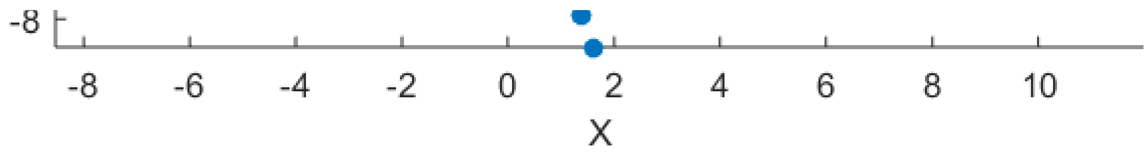
These proposal distributions will form a Markov chain which can be used for generating and verifying new parameters.

Based on the ideas mentioned above, I implemented them and tested it with 1 and 2 components and the test result is as following:

- one component

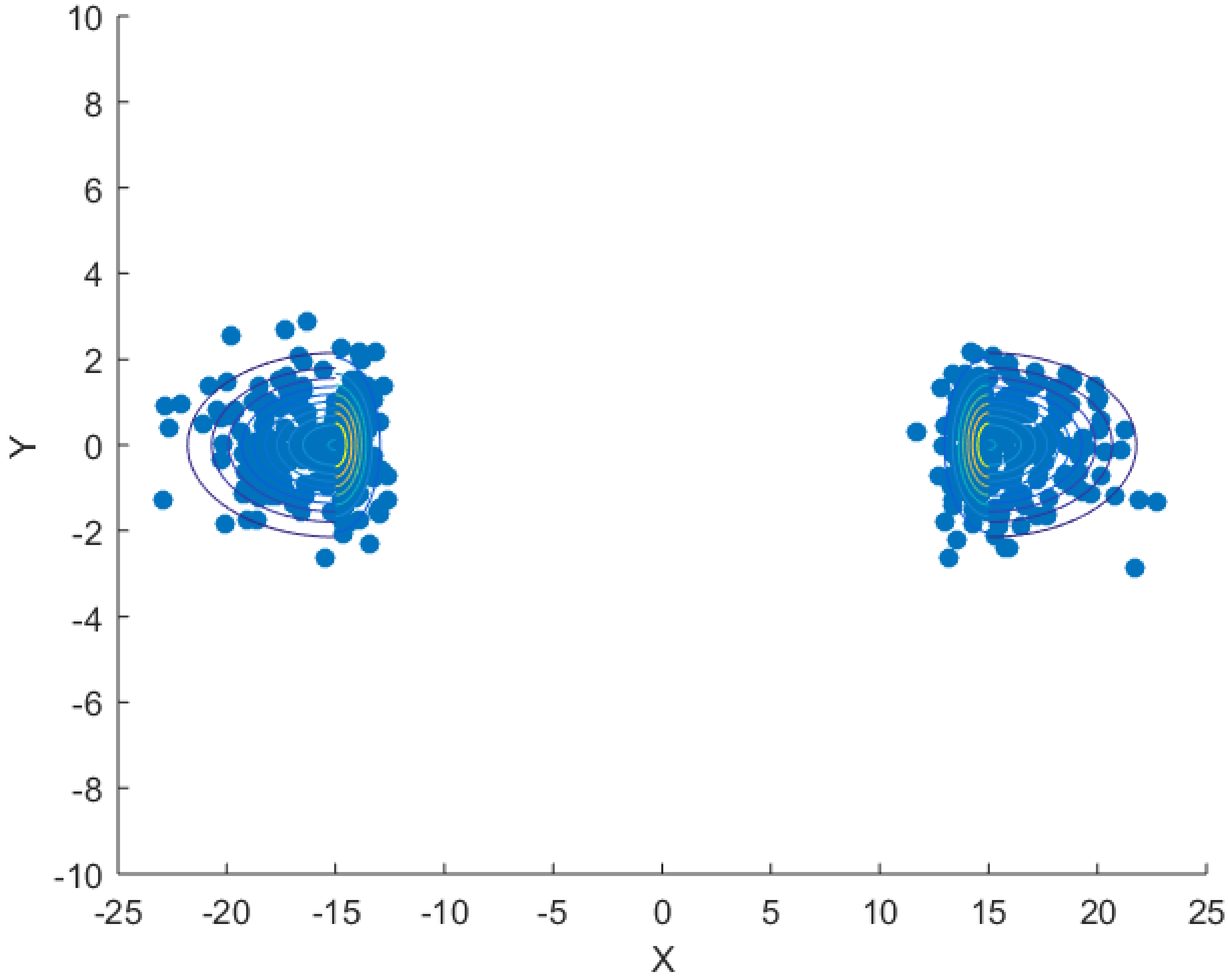

10
8
6
4
2
0
-2
-4
-6
-8
Y
-8
-6
-4
-2
0
2
4
6
8
10
X

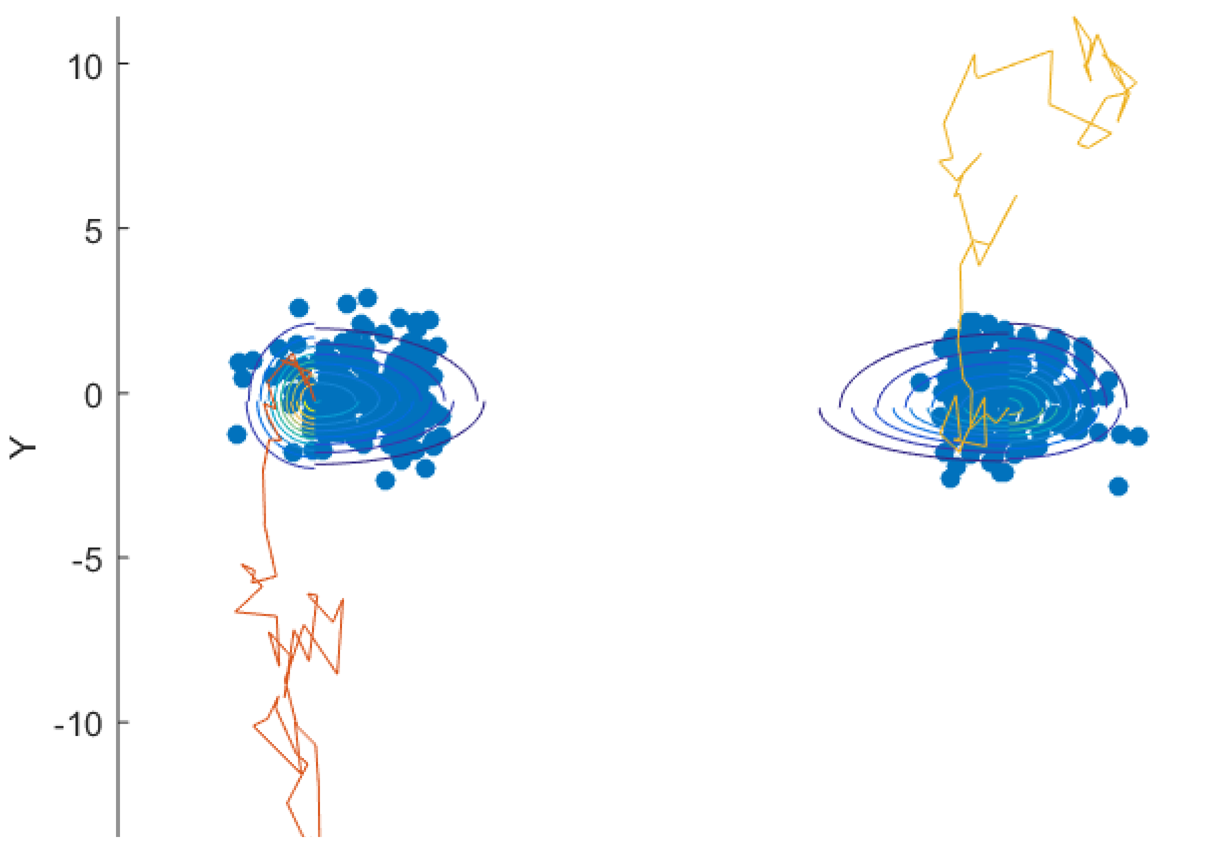

10
8
6
4
2
0
-2
-4
-6
Y

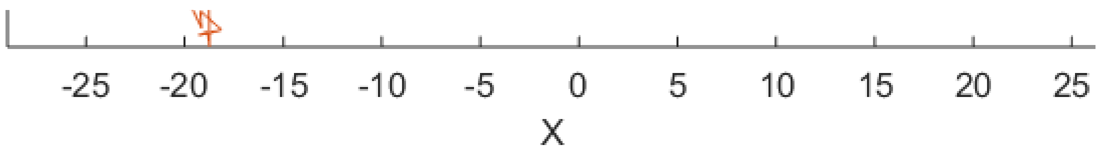


For one component this algorithm works fine, compared to the model which generated the data points, the generated sigmas are much less because the points far away from the center only have a very small probability which will be ignored during clustering.

- 2 components


10
8
6
4
2
0
-2
-4
-6
-8
-10
Y
-25
-20
-15
-10
-5
0
5
10
15
20
25
X


10
5
0
-5
-10
Y



With 2 components this model has a problem that both groups will affect each other because this asymmetric model is not a mixture model which means that all components are relatively independent (every component is trying to use a big sigma to include the points of another group), I would like to have your suggestion of this issue.

I' not sure if my idea is correct and I'm looking forward to discussing this with you on Wednesday.

Thank you!
Shuai Fu

From: Fu Amos <fs1984@msn.com>
Sent: July 6, 2017 2:51 PM
To: Nizar Bouguila
Subject: Re: MH-within-Gibbs for asymmetric Gaussian mixture

Hello Professor,
I'm working on the implementation based on the idea I sent you before. Using MCMC method for the clustering of asymmetric Gaussian mixture is possible but the implementation is harder compared to the symmetric ones because there is no existing Matlab libs that I can use directly for sampling,calculating the membership vector and draw graphics etc,. I will finish it during this weekend and will send you the result before next Monday.
I wonder if we could meet next week if you have time.

Thank you!
Shuai

From: Fu Amos <fs1984@msn.com>
Sent: June 7, 2017 5:00 PM
To: Nizar Bouguila
Subject: MH-within-Gibbs for asymmetric Gaussian mixture

Dear Doctor,
I'm trying to extend the MH-within-Gibbs method from symmetric Gaussian mixture to asymmetric ones. Based on my understanding, I'm trying to apply the existing PDF function of asymmetric Gaussian to Gibbs sampling process and thinking the way of sampling new Gaussian parameters.

From the paper <Background subtraction using finite mixtures of asymmetric Gaussian distributions and shadow detection>, formula 2 & 4:

$$p(\mathbf{X}|\xi_j) = \prod_{k=1}^{d} \sqrt{\frac{2}{\pi}} \frac{1}{(\sigma_{l_j} + \sigma_{r_j})} \times \begin{cases} \exp\left[-\frac{(X_k-\mu_{jk})^2}{2\sigma_{l_{jk}}^2}\right] \text{if } X_k < \mu_{jk} \\ \exp\left[-\frac{(X_k-\mu_{jk})^2}{2\sigma_{r_{jk}}^2}\right] \text{if } X_k \geq \mu_{jk} \end{cases} \quad (2)$$

$$p(\mathcal{X}, Z|\Theta) = \prod_{i=1}^{N}\prod_{j=1}^{M} (p(\mathbf{X}_i|\xi_j)p_j)^{Z_{ij}} \quad (4)$$

The likelihood function with membership vector can be used directly into the Gibbs process without modification. So for each iteration:

$$P(z_i^{(t)} = j) \propto p_j^{(t-1)} \prod_{k=1}^{d} \frac{1}{(\sigma_{l_{jk}}^{(t-1)} + \sigma_{r_{jk}}^{(t-1)})} \times \begin{cases} \exp\left[-\frac{(X_k-\mu_{jk}^{(t-1)})^2}{2(\sigma_{l_{jk}}^{(t-1)})^2}\right] & if\ X_k < \mu_{jk}^{(t-1)} \\ \exp\left[-\frac{(X_k-\mu_{jk}^{(t-1)})^2}{2(\sigma_{r_{jk}}^{(t-1)})^2}\right] & if\ X_k > \mu_{jk}^{(t-1)} \end{cases}$$

It looks fine to re-generate the membership vector $z_i^{(t)}$ dimension-by-dimension based on this formula and therefor, generate the weight $p^{(t)}$ for all the mixture components .

Having the weight $p^{(t)}$ in hand, next step should be sampling new Gaussian mixture parameters from the proposal distributions. I will assume σ and μ are independent and they all follow normal distribution to simplify this process. Since the likelihood function is dimension-by-dimension based so we can sample σ and μ dimension-by-dimension as well. For example: for 2D mixture with 2 components, we will sample $\mu_{11}, \mu_{12}, \mu_{21}, \mu_{22}$ from the same 1d normal distribution and then

$\mu_1 = \{\mu_{11}, \mu_{12}\}\ and\ \mu_2 = \{\mu_{21}, \mu_{22}\}$. The same rule applies to $\sigma_l\ and\ \sigma_r$.

Once we got new parameters from the proposal distribution, we can follow the basic MH algorithm and calculate the acceptance ration r and make a decision whether accept or refuse these new parameters.

Before I start the implementation I want to have your opinion about this plan and please feel free to correct me if there is any mistake.

Thank you very much!

Best regards,
Shuai Fu