CSED524 Final Project

Inhyuk Na

2018/12/21

1 Introduction

Dirichlet Process Mixture Model(DPMM) is bayesian non-parametric, and infinite mixture model which can find clusters of given dataset. In this paper we explain detailed description about DPMM and derivation of variational inference of it. Then, we experiment with two dataset with this model. This paper is basically based on [1] and the lectures of POSTECH CSED524, and we follow the notations and details from the lecture.

2 Dirichlet Process Mixture Model

In this section, we explain about DPMM.

2.1 Bayesian Parametric vs Non-parametric Models

DPMM is a Bayesian non-parametric model. There are two kinds of Bayesian approach for estimating the density of data $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]$. First is parametric approach. This have finite parameters θ for describing model. This model assumes $\boldsymbol{x}_n|\theta \sim p(\cdot|\theta)$. This model has prior over parameters $p(\theta)$, and posterior over parameters $p(\theta|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\theta)p(\theta)}{p(\boldsymbol{X})}$. Therefore, prediction is done by $p(\boldsymbol{x}_*|\boldsymbol{X}) = \int p(\boldsymbol{x}_*|\theta)p(\theta|\boldsymbol{X})d\theta$. Non-parametric approach is different. This model assumes that $\boldsymbol{x}_n \sim F$, for a distribution F, and this distribution also follows specific distribution, $F \sim DP(\alpha, G_0)$. The posterior and prediction are similar with parametric approach.

Parametric models are totally determined by its finite parameters, but non-parametric models have infinite dimensional spaces, so it is not feasible. Therefore in practice we use some techniques, such as weak distributions, explicit representations, implicit representations, and finite representations.

2.2 Finite vs Infinite Mixture Models

DPMM is also a infinite mixture model. Mixture models assume that data are based on some clusters. There are finite and infinite mixture models. The finite mixture model assumes $p(\boldsymbol{x}_i) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}_i|\theta_k^*)$, where $\boldsymbol{\pi} = [\pi_1, \cdots, \pi_K]^T$ and $\{\theta_k^*\}$ are parameters. The popular example of this finite mixture model is Gaussian mixture model. In this case, $\theta_k^* = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, and $p(\boldsymbol{x}_i|\theta_k^*)$ is a Guassian with mean and covariance are $\boldsymbol{\mu}_k and \boldsymbol{\Sigma}_k$, respectively. The finite mixture model introduces the latent variables $\boldsymbol{z}_i$, which represent which cluster data are in. We then write the distribution as

$$p(\boldsymbol{x}_i|\pi,\theta^*) = \sum_{k=1}^{K} p(z_{k,i}=1|\pi)p(\boldsymbol{x}_i|z_{k,i}=1,\theta^*)$$
$$= \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}_i|\theta_k^*).$$

With introducing $G = \sum_{k=1}^{K} \pi_k \delta_{\theta_k^*}$, we can re-write the density as

$$p(\boldsymbol{x}_i) = \int p(\boldsymbol{x}_i|\theta)G(\theta)d\theta.$$

The infinite mixture model extends G as

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k^*},$$

with using Bayesian non-parametric approaches. When we place a DP prior on G, this model become DPMM.

2.3 Dirichlet Distribution

Dirichlet distribution is a multivariate distribution over K random variables, and we notate them as π_k in this section, for $k = 1, \cdots, K$. They are all in $[0, 1]$ and their sum is 1. The detailed formula of this distribution is

$$Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{1}{Z(\alpha_1, ..., \alpha_K)} \prod_{j=1}^{K} \pi_j^{\alpha_j - 1},$$

where $Z(\alpha_1, ..., \alpha_K) = \frac{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)}{\Gamma(\alpha_1+\cdots+\alpha_K)}$.

We introduce two characteristics of Dirichlet distribution. First is an aggregation property, which is that aggregations of any subsets of categories produce Dirichlet distributions of

$$\Big(\sum_{i=1}^{r_1}\pi_i, \sum_{i=r_1+1}^{r_2}\pi_i \cdots, \sum_{i=r_L-1}^{r_L}\pi_i\Big) \sim Dir(\sum_{i=1}^{r_1}\alpha_i, \sum_{i=r_1+1}^{r_2}\alpha_i, \cdots, \sum_{i=r_L-1}^{r_L}\alpha_i),$$

for $0 < r_1 < \cdots < r_L = K$.

The second property is that the Dirichlet distribution is that the posterior distribution over multinomial variables is the Dirichlet distribution. Consider a multinomial distribution of

$$p(x_1, \cdots, x_N|\pi_1, \cdots, \pi_K) = \frac{N!}{N_1! \cdots N_K!} \prod_{j=1}^{K} \pi_j^{N_j},$$

where $N_j = \sum_{n=1}^{N} \delta(x_n, j)$ is the count of x in category j. Then, the posterior distribution is

$$p(\pi_1, \cdots, \pi_K | x_1, \cdots, x_N) \propto p(x_1, \cdots, x_N | \pi_1, \cdots, \pi_K) p(\pi_1, \cdots, \pi_K)$$
$$\propto \Big[\prod_{j=1}^{K} \pi_j^{N_j} \Big] \Big[\prod_{j=1}^{K} \pi_j^{\alpha_j - 1} \Big],$$

leading to

$$p(\pi_1, \cdots, \pi_K | x_1, \cdots, x_N) = Dir(\alpha_1 + N_1, \cdots, \alpha_K + N_K).$$

There are several ways to generate a realization of the Dirichlet distribution, such as Polya's Urn model, transformation of Gamma random variables, finite stick-breaking approach, and Sethuraman's stick-breaking construction.

2.4 Dirichlet Process

We first introduce the formal definition of the Dirichlet process. Let G_0 be a probability distribution on a measurable space Θ and α a positive scalar. Consider a finite partition $(A_1, \cdots, A_K)$ of Θ. Then, a random probability distribution G on Θ is drawn from a DP if its measure on every finite measurable partition, $(A_1, \cdots, A_K)$, follows a Dirichlet distribution as

$$\Big(G(A_1), \cdots, G(A_K) \Big) \sim Dir\Big(\alpha G_0(A_1), \cdots, \alpha G_0(A_K) \Big).$$

There is a theorem which is that for any base measure G_0 and concentration parameter $\alpha > 0$, there exists a unique process satisfying these conditions, denoted by $DP(\alpha, G_0)$,

$$G \sim DP(\alpha, G_0),$$

and we call this a Dirichlet process(DP).

G_0, one of the parameters of DP, specifies the mean of DP. For any measurable set $A_j \subset \Theta$, we can easilyt see taht the following is satisfied:

$$\mathbb{E}[G(A_j)] = \frac{\alpha G_0(A_j)}{\alpha G_0(A_1) + \cdots + \alpha G_0(A_K)}$$
$$= G_0(A_j).$$

α, the concentration parameter of DP, can be understood as inverse variance. This is similar to the precision of a finite Dirichlet distribution and this determines the average deviation of samples from the base measure, since

$$var(G(A_j)) = \frac{\alpha G_0(A_j)(\alpha - \alpha G_0(A_j))}{\alpha^2(\alpha + 1)}$$
$$= \frac{G_0(A_j)(1 - G_0(A_j))}{(\alpha + 1)}.$$

This is also referred to a strength parameter, since this refers the strength of the prior when we use DP as a non-parametric prior over distributions in a Bayesian non-parametric model.

DP has a property of conjugacy. For $G \sim DP(\alpha, G_0)$ and assume N independent observations $\theta_i \sim G$. Then the posterior measure also follows a DP such as

$$p(G|\theta_1, \cdots, \theta_N, \alpha, G_0) = DP(\alpha + N, \frac{\alpha}{\alpha + N} G_0 + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta_{\theta_i}).$$

By using above, we can also compute the predictive distribution for θ_{N+1} with G marginalized out. The above equation implies that

$$\begin{aligned} p(G(A_1), \cdots, G(A_K)|\theta_1, \cdots, \theta_N, \alpha, G_0) &= Dir(\alpha G_0(A_1) + \sum_{i=1}^{N} \delta_{\theta_i}(A_1), \cdots, \alpha G_0(A_K) + \sum_{i=1}^{N} \delta_{\theta_i}(A_K)) \\ &= Dir(\alpha G_0(A_1) + N_1, \cdots, \alpha G_0(A_K) + N_K). \end{aligned}$$

Then, for any $A_j \subset \Theta$,

$$\begin{aligned} p(\theta_{N+1} \in A_j|\theta_1, \cdots, \theta_N) &= \mathbb{E}[G(A_j)|\theta_1, \cdots, \theta_N] \\ &= \frac{\alpha G_0(A_j) + \sum_{i=1}^{N} \delta_{\theta_i}(A_j)}{\sum_{j=1}^{K} \left[\alpha G_0(A_j) + \sum_{i=1}^{N} \delta_{\theta_i}(A_j)\right]} \\ &= \frac{\alpha G_0(A_j) + \sum_{i=1}^{N} \delta_{\theta_i}(A_j)}{\alpha \sum_{j=1}^{K} G_0(A_j) + \sum_{j=1}^{K} \sum_{i=1}^{N} \delta_{\theta_i}(A_j)} \\ &= \frac{\alpha G_0(A_j) + \sum_{i=1}^{N} \delta_{\theta_i}(A_j)}{\alpha + N}. \end{aligned}$$

Thus with G marginalized out, we have

$$\theta_{N+1}|\theta_1, \cdots, \theta_N \sim \frac{1}{\alpha + N} \Big(\alpha G_0 + \sum_{i=1}^{N} \delta_{\theta_i}\Big),$$

and the sequence of these predictive distributions is called Polya Urn scheme.

2.5 Various representations of Dirichlet Processes

There are various representations of DP. In this section, we introduce three of them: Polya Urn process, Stick-breaking process, and Chinese restaurant process.

2.5.1 Polya Urn Process

DP by Polya urn process is naturally comes out with the above equation. Recall that we have the predictive distribution

$$\theta_{N+1}|\theta_1, \cdots, \theta_N \sim \frac{1}{\alpha + N} \Big(\alpha G_0 + \sum_{i=1}^{N} \delta_{\theta_i}\Big).$$

In ths interpretation, each value in Θ is thought to be a unique color and $\theta \sim G$ are thought to be balls with the drawn value being the color of the value. In the beginning, the urn is empty and we pick a ball

with a color $\theta_1 \sim G_0$ and put it into the urn. Inductively, in $(N+1)$ steps, we pick a new color ball $\theta_{N+1} \sim G_0$ with probability $\frac{\alpha}{\alpha+N}$. With another probability $\frac{N}{\alpha+N}$, we pick a color ball from the urn. Then we put a new ball with the same color to the urn.

Note that we can modify the above equation as

$$\mathbb{E}\{G(A)|\theta_1, \cdots, \theta_N\} = \frac{1}{\alpha+N}\Big(\alpha G_0(A) + \sum_{k=1}^{K} N_k \delta_{\theta_k^*}(A)\Big),$$

where $N_k = \sum_{i=1}^{N} \delta(\theta_i, \theta_k^*)$. Then we can derive

$$\begin{aligned}\lim_{N\to\infty} \mathbb{E}\{G(A)|\theta_1, \cdots, \theta_N\} &= \lim_{N\to\infty} \frac{\alpha G_0(A) + \sum_{i=1}^{N} \delta_{\theta_i}(A)}{\alpha+N} \\ &= \sum_{k=1}^{\infty} \pi_k \delta_{\theta_i}(A),\end{aligned}$$

and these implies that G are discrete distributions.

2.5.2 Stick-Breaking Process

Stick breaking process uses beta distributions to draw π_k. We can draw π_k as

$$\begin{aligned}&\beta_k \sim Beta(1, \alpha), \\ &\pi_k = \beta_k \prod_{j=1}^{k-1}(1-\beta_j)\beta_k(1 - \sum_{j=1}^{k-1}\pi_j).\end{aligned}$$

If we construct

$$G(\theta) = \sum_{k=1}^{\infty} \pi_k \delta(\theta, \theta_k^*), \text{ where } \theta_k^* \sim G_0,$$

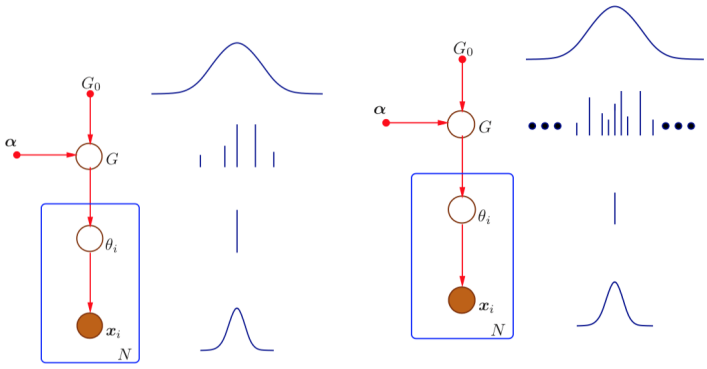
this is guaranteed to $G \sim DP(\alpha, G_0)$. The inverse of this statement also is guaranteed. We use this Stick-breaking process to handle DP in DPMM in the following sections.

2.6 Chinese Restaurant Process

The Chinese restaurant process describes the infinite number of customers and tables in Chinese restaurant. Each customers sit at a table, and these refer θ_k. Similar with Polya urn representation, the first customer sits at the first table. Inductively, the $(N+1)$th subsequent customer sits at a table drawn from the next occupied table with $\frac{\alpha}{N+\alpha}$, and sits in previously occupied table with $\frac{N_k}{N+\alpha}$, where N_k represents the number of customers currently at table k.

2.7 Dirichlet Process Mixture Model

Now we can introduce our main topic, DPMM. Before we start, recall the Bayesian finite mixture models first. We borrow some great figures from the lecture. As shown in Figure 1-(a), The Bayesian finite mixture models use parametric approach which is that



(a) Bayesian finite MM (b) DPMM

Figure 1: Comparing graphical models between Bayesian finite MM and DPMM

$$
\begin{aligned}
\pi_k &\sim Dir(\frac{\alpha}{K}, \cdots, \frac{\alpha}{K}), \\
\theta_k^* &\sim G_0, \\
G &= \sum_{k=1}^{K} \pi_k \delta_{\theta_k^*} \\
\theta_i &\sim G, \\
\boldsymbol{x}_i &\sim p(\cdot|\theta_i).
\end{aligned} \tag{1}
$$

We have to give a fixed hyperparameter K which indicates the total number of clusters. However, when it is hard to figure out the meta property of data, specifying K can be difficult job. DPMM solves this problem, like Figure 1-(b). We can introduce DP to the mixture model, and this becomes to

$$
\begin{aligned}
G &\sim DP(\alpha, G_0), \\
\theta_i^* &\sim G, \\
\boldsymbol{x}_i &\sim p(\cdot|\theta_i).
\end{aligned}
$$

To handle this, we use stick-breaking representation for DP such as

$$
\begin{aligned}
\beta_k &\sim Beta(1, \alpha), \\
\pi_k &= \beta_k \prod_{j=1}^{k-1} (1 - \beta_j), \\
G &= \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k^*}.
\end{aligned}
$$

2.8 Inference techniques of DPMM

To inference DPMM, there are several ways to inference DPMM. There was Monte-Carlo Markov chain sampling methods to inference this, however, [1] points out the slowness of the method. We introduce their variational inference techniques in the following sections. There are also Gibbs sampling and the other well-knonwn techniques to inference DPMM.

2.9 Variational Inference of DPMM

As the other mixture models, we introduce hidden variables $\boldsymbol{z}_i$ to out graphical model. $\boldsymbol{z}_i$ refer to the cluster which the data are in. $\boldsymbol{z}_i$ follow the multinomial distribution of $\boldsymbol{\pi}$, such that

$$\begin{aligned} z_i|\pi_1, \pi_2, ... &\sim Mult(\boldsymbol{\pi}) \\ &= \prod_k \pi_k^{\delta(z_i,k)}. \end{aligned}$$

Thus, we assume that conditional distribution of $\boldsymbol{X}$ and $\boldsymbol{\theta}^*$ is members of the exponential family. We write them as

$$p(\boldsymbol{x}_i|z_i, \boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \cdots) = \prod_k \left[h(\boldsymbol{x}_i) \exp\{\boldsymbol{\theta}_k^{*T} \boldsymbol{x}_i - a(\boldsymbol{\theta}_k^*)\}\right]^{\delta(z_i,k)},$$

$$p(\boldsymbol{\theta}_k^*|\boldsymbol{\lambda}) = h(\boldsymbol{\theta}_k^*) \exp\{\boldsymbol{\lambda}_1^T \boldsymbol{\theta}_k^* - \lambda_2 a(\boldsymbol{\theta}_k^*) - a(\boldsymbol{\lambda})\},$$

and one can know that $G_0 = p(\boldsymbol{\theta}_k^*|\boldsymbol{\lambda})$ and $\boldsymbol{\lambda}$ are parameters of G_0.

We now find the approximate probability distribution for latent variables $\{\boldsymbol{\beta}, \boldsymbol{\theta}^*, \boldsymbol{Z}\}$, which minimizes the marginal log likelihood of probability distribution of observed data $\boldsymbol{X}$ with hyperparameters $\alpha, \boldsymbol{\lambda}$.

We take Jensen's inequation to marginal log-likelihood,

$$\begin{aligned} \log p(\boldsymbol{X}|\alpha, \boldsymbol{\lambda}) &= \log \int p(\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\theta^*}, \boldsymbol{Z}|\alpha, \boldsymbol{\lambda}) d\boldsymbol{\beta} d\boldsymbol{\theta}^* d\boldsymbol{Z} \\ &\geq \int q(\boldsymbol{\beta}, \boldsymbol{\theta}^*, \boldsymbol{Z}) \log \frac{p(\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\theta^*}, \boldsymbol{Z}|\alpha, \boldsymbol{\lambda})}{q(\boldsymbol{\beta}, \boldsymbol{\theta^*}, \boldsymbol{Z})} d\boldsymbol{\beta} d\boldsymbol{\theta}^* d\boldsymbol{Z} \\ &= \mathcal{F}(q), \end{aligned}$$

and we get $\mathcal{F}(q)$, the variational lower-bound(ELBO).

Note that our graphical model follows the factorization,

$$p(\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\theta^*}, \boldsymbol{Z}|\alpha, \boldsymbol{\lambda}) = p(\boldsymbol{\beta}|\alpha) p(\boldsymbol{Z}|\boldsymbol{\beta}) p(\boldsymbol{\theta}^*|\boldsymbol{\lambda}) p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\theta}^*),$$

where

$$p(\boldsymbol{\beta}|\alpha) = \prod_k Beta(\beta_k|1, \alpha),$$

$$p(\boldsymbol{Z}|\boldsymbol{\beta}) = \prod_{i=1}^N p(z_i|\boldsymbol{\beta}) = \prod_{i=1}^N \prod_k \pi_k^{\delta(z_i,k)} = \prod_{i=1}^N \prod_k \left[\beta_k \prod_{l=1}^{k-1} (1 - \beta_l)\right]^{\delta(z_i,k)},$$

$$p(\boldsymbol{\theta}^*|\boldsymbol{\lambda}) = \prod_k p(\boldsymbol{\theta}_k^*|\boldsymbol{\lambda}) = \prod_k \left[h(\boldsymbol{\theta}_k^*) \exp\{\boldsymbol{\lambda}_1^T \boldsymbol{\theta}_k^* - \lambda_2 a(\boldsymbol{\theta}_k^*) - a(\boldsymbol{\lambda})\}\right],$$

$$p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\theta^*}) = \prod_{i=1}^N p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{\theta}^*) = \prod_{i=1}^N \prod_k \left[h(\boldsymbol{x}_i) \exp\{\boldsymbol{\theta}_k^{*T} \boldsymbol{x}_i - a(\boldsymbol{\theta}_k^*)\}\right]^{\delta(z_i,k)}$$

Now we introduce some approximation techniques to handle DPMM practically. First is the truncated stick-breaking representation. Since the term infinite is not feasible, We fix the truncation level K and let

$$q(\beta_k = 1) = 1,$$

which implies that the mixture proportions π_k are equal to 0 for $k > K$. Second is mean-field approximation. We assume that variational distribution $q(\boldsymbol{\beta}, \boldsymbol{\theta}^*, \boldsymbol{Z})$ factorizes as

$$\begin{aligned} q(\boldsymbol{\beta}, \boldsymbol{\theta^*}, \boldsymbol{Z}) &= q(\boldsymbol{\beta}) q(\boldsymbol{\theta}^*) q(\boldsymbol{Z}) \\ &= \Big[\prod_{k=1}^{K-1} q(\beta_k | \gamma_k) \Big] \Big[\prod_{k=1}^{K} q(\boldsymbol{\theta}_k^* | \boldsymbol{\tau_k}) \Big] \Big[\prod_{i=1}^{N} q(z_i | \phi_i) \Big], \end{aligned}$$

where

$$\begin{aligned} q(\beta_k | \gamma_k) &= Beta(\beta_k | \gamma_{k,1} \gamma_{k,2}), \\ q(\boldsymbol{\theta}_k^* | \boldsymbol{\tau}_k) &= h(\boldsymbol{\theta}_k^*) \exp\{\boldsymbol{\tau}_{k,1}^T \boldsymbol{\theta}_k^* - \tau_{k,2} a(\boldsymbol{\theta}_k^*) - a(\boldsymbol{\tau}_k)\}, \\ q(z_i | \phi_i) &= \prod_{k=1}^{K} \phi_{i,k}^{\delta(z_i, k)}, \end{aligned}$$

and we determine variational parameters $\{\gamma_k, \boldsymbol{\tau}_k, \phi_i\}$ to minimize ELBO.

We now apply mean-filed coordinate ascent algorithm to our variational parameters to minimize ELBO. In the mean-filed coordinate ascent algorithm, we update the variational parameters iteratively, to satisfy the following conditions:

$$\begin{aligned} \log q(\boldsymbol{\beta}) &\propto \mathbb{E}_{q_{\boldsymbol{\theta^*}, \boldsymbol{Z}}} \big[\log p(\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\theta^*}, \boldsymbol{Z} | \alpha, \boldsymbol{\lambda}) \big], \\ \log q(\boldsymbol{\theta}^*) &\propto \mathbb{E}_{q_{\boldsymbol{\beta}, \boldsymbol{Z}}} \big[\log p(\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\theta^*}, \boldsymbol{Z} | \alpha, \boldsymbol{\lambda}) \big], \\ \log q(\boldsymbol{Z}) &\propto \mathbb{E}_{q_{\boldsymbol{\beta}, \boldsymbol{\theta^*}}} \big[\log p(\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\theta^*}, \boldsymbol{Z} | \alpha, \boldsymbol{\lambda}) \big]. \end{aligned}$$

We first solve $\log q(\boldsymbol{\beta}) \propto \mathbb{E}_{q_{\boldsymbol{\theta^*},\boldsymbol{Z}}}\big[\log p(\boldsymbol{X},\boldsymbol{\beta},\boldsymbol{\theta^*},\boldsymbol{Z}|\alpha,\boldsymbol{\lambda})\big]$. With using above factorizations,

$$
\begin{aligned}
\log q(\beta_k) &= \mathbb{E}_{q_{\boldsymbol{\theta^*},\boldsymbol{Z}}}\big[\log p(\boldsymbol{X},\boldsymbol{\beta},\boldsymbol{\theta^*},\boldsymbol{Z}|\alpha,\boldsymbol{\lambda})\big] + const \\
&= \mathbb{E}_{q_{\boldsymbol{\theta^*},\boldsymbol{Z}}}\big[\log p(\boldsymbol{\beta}|\alpha) + \log p(\boldsymbol{Z}|\boldsymbol{\beta}) + \log p(\boldsymbol{\theta}^*|\boldsymbol{\lambda}) + \log p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{\theta}^*)\big] + const \\
&= \mathbb{E}_{q_{\boldsymbol{\theta^*},\boldsymbol{Z}}}\big[\log p(\boldsymbol{\beta}|\alpha)\big] + \mathbb{E}_{q_{\boldsymbol{\theta^*},\boldsymbol{Z}}}\big[\log p(\boldsymbol{Z}|\boldsymbol{\beta})\big] + const \\
&= \mathbb{E}_{q_{\boldsymbol{\theta^*},\boldsymbol{Z}}}\big[\sum_k \log Beta(\beta_k|1,\alpha)\big] + \mathbb{E}_{q_{\boldsymbol{\theta^*},\boldsymbol{Z}}}\big[\log p(\boldsymbol{Z}|\boldsymbol{\beta})\big] + const \\
&= \mathbb{E}_{q_{\boldsymbol{\theta^*},\boldsymbol{Z}}}\big[\sum_k \log\Gamma(1+\alpha) - \log\Gamma(1) - \log\Gamma(\alpha) + (0)\log(\beta_k) + (\alpha-1)\log(1-\beta_k)\big] + \mathbb{E}_{q_{\boldsymbol{\theta^*},\boldsymbol{Z}}}\big[\log p(\boldsymbol{Z}|\boldsymbol{\beta})\big] + cons \\
&= (\alpha-1)\log(1-\beta_k) + \mathbb{E}_{q_{\boldsymbol{\theta^*},\boldsymbol{Z}}}\big[\log p(\boldsymbol{Z}|\boldsymbol{\beta})\big] + const \\
&= (\alpha-1)\log(1-\beta_k) + \mathbb{E}_{q_{\boldsymbol{\theta^*},\boldsymbol{Z}}}\big[\sum_{i=1}^{N}\sum_k \delta(z_i,k)(\log(\beta_k) + \sum_{l=1}^{k-1}\log(1-\beta_l))\big] + const \\
&= (\alpha-1)\log(1-\beta_k) + \sum_{i=1}^{N}\sum_k q(z_i=k)\mathbb{E}_{q_{\boldsymbol{\theta^*},\boldsymbol{Z}}}[(\log(\beta_k) + \sum_{l=1}^{k-1}\log(1-\beta_l))] + const \\
&= (\alpha-1)\log(1-\beta_k) + \sum_{i=1}^{N}[q(z_i=k)(\log(\beta_k)) + q(z_i>k)(\log(1-\beta_k))] + const \\
&= (\alpha-1)\log(1-\beta_k) + \sum_{i=1}^{N}[\phi_{i,k}(\log(\beta_k)) + \sum_{i=k+1}^{K}\phi_{i,k}(\log(1-\beta_k))] + const \\
&= (1+\sum_{i=1}^{N}\phi_{i,k}-1)\log(\beta_k) + (\alpha+\sum_{i=1}^{N}\sum_{l=k+1}^{K}\phi_{i,l}-1)\log(1-\beta_k) + const.
\end{aligned}
$$

Since

$$
\begin{aligned}
\log q(\beta_k|\boldsymbol{\gamma}_k) &= \log(\frac{\Gamma(\gamma_{k,1}+\gamma_{k,2})}{\Gamma\gamma_{k,1}\Gamma_{k,2}}\beta_k^{\gamma_{k,1}-1}(1-\beta_k)^{\gamma_{k,2}-1}) \\
&= (\gamma_{k,1}-1)\log(\beta_k) + (\gamma_{k,2}-1)\log(1-\beta_k) + const,
\end{aligned}
\tag{2}
$$

we can see

$$
\begin{aligned}
\gamma_{k,1} &= 1 + \sum_{i=1}^{N}\phi_{i,k}, \\
\gamma_{k,2} &= \alpha + \sum_{i=1}^{N}\sum_{l=k+1}^{K}\phi_{i,l}.
\end{aligned}
$$

We then solve for $\log q(\boldsymbol{\theta}^*) \propto \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{Z}}}\big[\log p(\boldsymbol{X},\boldsymbol{\beta},\boldsymbol{\theta^*},\boldsymbol{Z}|\alpha,\boldsymbol{\lambda})\big]$. With similar factorizations above,

$$
\begin{aligned}
\log q(\boldsymbol{\theta}_k^*) &= \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{Z}}}\big[\log p(\boldsymbol{X},\boldsymbol{\beta},\boldsymbol{\theta^*},\boldsymbol{Z}|\alpha,\boldsymbol{\lambda})\big] + const \\
&= \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{Z}}}\big[\log p(\boldsymbol{\beta}|\alpha) + \log p(\boldsymbol{Z}|\boldsymbol{\beta}) + \log p(\boldsymbol{\theta}^*|\boldsymbol{\lambda}) + \log p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{\theta}^*)\big] + const \\
&= \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{Z}}}\big[\log p(\boldsymbol{\theta}^*|\boldsymbol{\lambda}) + \log p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{\theta}^*)\big] + const \\
&= \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{Z}}}\big[\log p(\boldsymbol{\theta}^*|\boldsymbol{\lambda})\big] + \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{Z}}}\big[\log p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{\theta}^*)\big] + const \\
&= \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{Z}}}\big[\log p(\boldsymbol{\theta}^*|\boldsymbol{\lambda})\big] + \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{Z}}}\big[\log p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{\theta}^*)\big] + const \\
&= \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{Z}}}\big[\sum_k \big[\log h(\boldsymbol{\theta}_k^*) + \boldsymbol{\lambda}_1^T\boldsymbol{\theta}_k^* - \lambda_2 a(\boldsymbol{\theta}_k^*) - a(\boldsymbol{\lambda})\big]\big] + \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{Z}}}\big[\log p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{\theta}^*)\big] + const \\
&= \sum_k \big[\log h(\boldsymbol{\theta}_k^*) + \boldsymbol{\lambda}_1^T\boldsymbol{\theta}_k^* - \lambda_2 a(\boldsymbol{\theta}_k^*) - a(\boldsymbol{\lambda})\big] + \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{Z}}}\big[\log p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{\theta}^*)\big] + const \\
&= \log h(\boldsymbol{\theta}_k^*) + \boldsymbol{\lambda}_1^T\boldsymbol{\theta}_k^* - \lambda_2 a(\boldsymbol{\theta}_k^*) + \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{Z}}}\big[\log p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{\theta}^*)\big] + const \\
&= \log h(\boldsymbol{\theta}_k^*) + \boldsymbol{\lambda}_1^T\boldsymbol{\theta}_k^* - \lambda_2 a(\boldsymbol{\theta}_k^*) + \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{Z}}}\big[\sum_{i=1}^N\sum_k \delta(z_i,k)\big[\log h(\boldsymbol{x_i}) + \boldsymbol{\theta_k^*}^T\boldsymbol{x_i} - a(\boldsymbol{\theta_k^*})\big]\big] + const \\
&= \log h(\boldsymbol{\theta}_k^*) + \boldsymbol{\lambda}_1^T\boldsymbol{\theta}_k^* - \lambda_2 a(\boldsymbol{\theta}_k^*) + \sum_{i=1}^N \phi_{i,k}(\boldsymbol{\theta}_k^{*T}\boldsymbol{x_i} - a(\boldsymbol{\theta}_k^*)) + const \\
&= \log h(\boldsymbol{\theta}_k^*) + (\boldsymbol{\lambda_1} + \sum_{i=1}^N \phi_{i,k}\boldsymbol{x_i})^T\boldsymbol{\theta}_k^* - (\lambda_2 + \sum_{i=1}^N \phi_{i,k})a(\boldsymbol{\theta}_k^*) + const.
\end{aligned}
$$

Since

$$
\begin{aligned}
\log q(\boldsymbol{\theta}_k^*|\boldsymbol{\tau}_k) &= \log(h(\boldsymbol{\theta}_k^*)\exp\{\boldsymbol{\tau}_{k,1}^T\boldsymbol{\theta}_k^* - \tau_{k,2}a(\boldsymbol{\theta}_k^*) - a(\boldsymbol{\tau}_k)\}) \\
&= \log h(\boldsymbol{\theta}_k^*) + \boldsymbol{\tau}_{k,1}^T\boldsymbol{\theta}_k^* - \tau_{k,2}a(\boldsymbol{\theta}_k^*) + const,
\end{aligned} \tag{3}
$$

we can see

$$
\boldsymbol{\tau_{k,1}} = \boldsymbol{\lambda}_1 + \sum_{i=1}^N \phi_{i,k}\boldsymbol{x}_i
$$

$$
\tau_{k,2} = \lambda_2 + \sum_{i=1}^N \phi_{i,k}.
$$

Finally, for $\log q(\boldsymbol{Z}) \propto \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}\big[\log p(\boldsymbol{X},\boldsymbol{\beta},\boldsymbol{\theta^*},\boldsymbol{Z}|\alpha,\boldsymbol{\lambda})\big]$, we can derive

$$
\begin{aligned}
\log q(z_i) &= \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}\big[\log p(\boldsymbol{X},\boldsymbol{\beta},\boldsymbol{\theta^*},\boldsymbol{Z}|\alpha,\boldsymbol{\lambda})\big] + const \\
&= \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}\big[\log p(\boldsymbol{\beta}|\alpha) + \log p(\boldsymbol{Z}|\boldsymbol{\beta}) + \log p(\boldsymbol{\theta}^*|\boldsymbol{\lambda}) + \log p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{\theta}^*)\big] + const \\
&= \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}\big[\log p(\boldsymbol{Z}|\boldsymbol{\beta}) + \log p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{\theta}^*)\big] + const \\
&= \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}\big[\sum_{i=1}^{N}\sum_{k}\delta(z_i,k)(\log(\beta_k) + \sum_{l=1}^{k-1}\log(1-\beta_l))\big] + \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}\big[\log p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{\theta}^*)\big] + const \\
&= \sum_{k}\delta(z_i,k)(\mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}\big[\log(\beta_k)\big] + \sum_{l=1}^{k-1}\mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}\big[\log(1-\beta_l))\big]) + \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}\big[\log p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{\theta}^*)\big] + const \\
&= \sum_{k}\delta(z_i,k)(\mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}\big[\log(\beta_k)\big] + \sum_{l=1}^{k-1}\mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}\big[\log(1-\beta_l))\big]) + \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}\big[\sum_{i=1}^{N}\sum_{k}\delta(z_i,k)\big[\log h(\boldsymbol{x}_i) + \boldsymbol{\theta}_k^{*T}\boldsymbol{x_i} - a(\boldsymbol{\theta}_k^*)\big]\big] \\
&\quad + const \\
&= \sum_{k}\delta(z_i,k)(\mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}\big[\log(\beta_k)\big] + \sum_{l=1}^{k-1}\mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}\big[\log(1-\beta_l))\big] + \sum_{k}\log h(\boldsymbol{x}_i) + \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}[\boldsymbol{\theta}_k^*]^T\boldsymbol{x}_i - \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}a(\boldsymbol{\theta}_k^*)]) + const.
\end{aligned}
\tag{4}
$$

Since

$$
\log q(z_i|\boldsymbol{\phi}_i) = \sum_{k=1}^{K}\delta(z_i,k)\phi_{i,k}, \tag{5}
$$

we can see

$$
\phi_{i,k} \propto \exp(\mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}[\log(\beta_k)] + \sum_{l=1}^{k-1}\mathbb{E}[\log(1-\beta_l)] + \mathbb{E}[\boldsymbol{\theta}_k^*]^T\boldsymbol{x}_i - \mathbb{E}_{q_{\boldsymbol{\beta},\boldsymbol{\theta^*}}}[a(\boldsymbol{\theta}_k^*)])).
$$

Like above, iteratively updating the variational parameters, we can achieve variational inference.

3 Experiments

In this section, we describe the results of experimenting DPMM. We used Gaussians for member of exponential family. The full code can be seen at `https://github.com/inyukwo1/POSTECH_CSED524`.

3.1 Old faithful data

We first show experiments of old faithful data. This data contains 272 entries and each entry has 2 features. The 2-D plot of this dataset is shown in Figure 2. We can realize that there is two clusters. We applied DPMM to this dataset. The result is shown in Figure 3. THe elipsoid is drawn by the mean and variances of Gaussians. The iteration converged in 70 iterations. By giving another random seeds, sometimes DPMM find more than two clusters. That kind of result is shown in Figure 4

3.2 MNIST and EMNIST data

We experimented DPMM for MNIST(Modified National Institute of Standards and Technology database) and EMNIST(Extended MNIST) dataset. MNIST is a famous handwritten dataset, and each datum represents one digits. It is consisted of 60000 entries and each entry has 28 by 28 pixels. EMNIST is extended version of MNIST, which has similar structure with MNIST, but has more data. EMNIST also has letters data which are not just a digit.

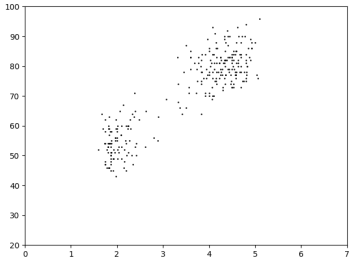
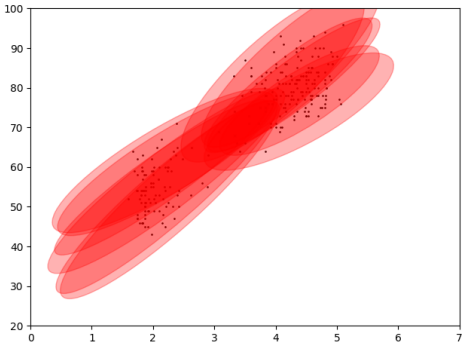
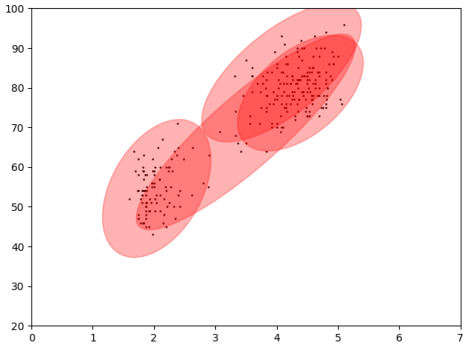


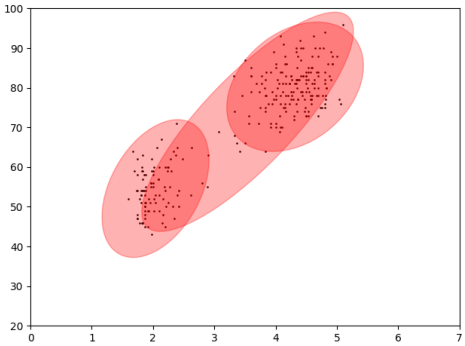
Figure 2: Old faithful dataset



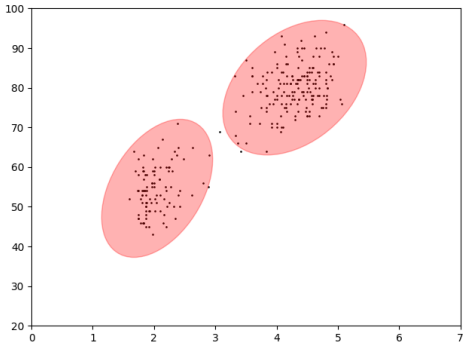
(a) 1 iteration



(b) 20 iteration



(c) 50 iteration



(d) 70 iteration

Figure 3: Result of DPMM in old faithful dataset

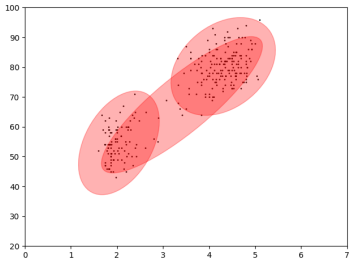


Figure 4: DPMM finds more than two clusters

```
>>> print(components[0][0:50])
[6, 6, 6, 6, 6, 6, 0, 6, 6, 6, 6, 6, 6, 6]
>>> print(components[1][0:50])
[7, 4, 4, 1, 1, 1, 7, 1, 2, 7, 1, 1, 1, 1, 7, 1, 1, 4, 7, 1, 1, 1, 1, 7, 7, 1, 7, 1, 1, 1, 7, 1, 1, 2, 1, 1, 1, 1, 7, 2, 2, 1, 4, 4, 2, 1, 1, 1, 7, 1]
>>> print(components[2][0:50])
[3, 3, 3, 3, 3, 3, 3, 3, 9, 3, 3, 3, 3, 3, 3, 3, 5, 3, 0, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]
>>> print(components[3][0:50])
[9, 4, 4, 7, 9, 9, 7, 4, 9, 9, 9, 9, 4, 2, 4, 4, 4, 9, 5, 9, 4, 9, 9, 4, 4, 9, 0, 9, 4, 4, 7, 4, 9, 4, 4, 9, 4, 4, 4, 4, 4, 9, 4, 9, 9, 4, 4, 4, 4, 9]
>>> print(components[4][0:50])
[6, 6, 6, 8, 6, 6, 6, 6, 6, 6, 6, 8, 6, 6, 6, 6, 6, 6, 6, 6, 2, 6, 6, 6, 6, 6, 6, 6, 2]
>>> print(components[5][0:50])
[9, 9, 4, 9, 9, 9, 4, 9, 4, 9, 4, 9, 8, 9, 9, 7, 9, 4, 4, 9, 9, 9, 4, 9, 4, 9, 9, 9, 9, 9, 4, 4, 9, 9, 4, 9, 9, 9, 4, 4, 4, 9, 4, 4, 4, 4, 4, 9, 9]
>>> print(components[6][0:50])
[6, 4, 6, 6, 6, 6, 6, 9, 6, 0, 6, 6, 6, 6, 2, 6, 6, 6, 6, 6, 4, 6, 6, 6, 4, 6, 6, 6, 6, 6, 6, 4, 6, 6, 4]
```

Figure 5: The first seven clusters in DPMM with MINST

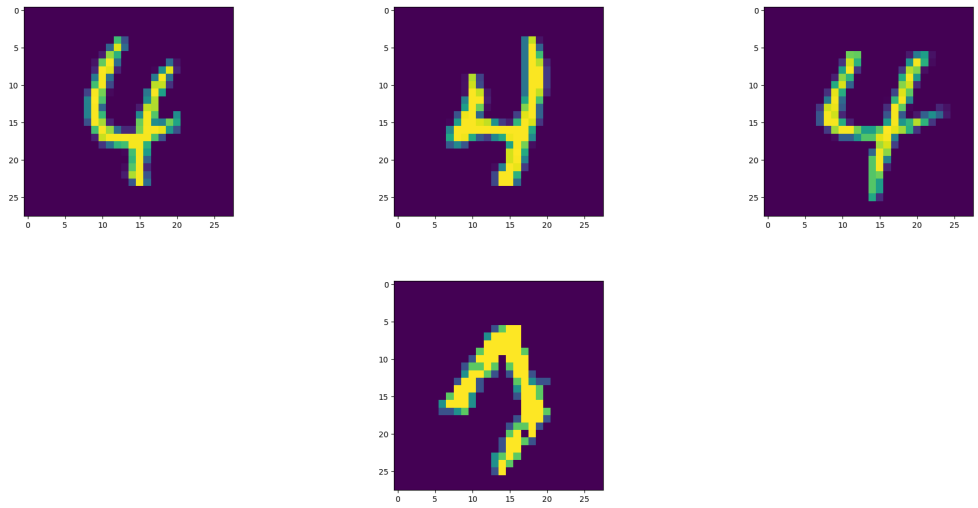


Figure 6: Some entries in cluster 4

3.2.1 Reason of choosing dataset

DPMM has a strength when we don't know the number of clusters of dataset. MNIST is obviously consisted of 10 classes(0to 9), but we don't have any idea about EMNIST. Therefore it is hard to apply the other mixture models to classify EMNIST dataset. We first wanted to classify MNIST, to be convinced that DPMM has ability to classify handwritten dataset. Then we applied DPMM to EMNIST, to see how DPMM finds clusters of EMNIST.

3.2.2 MNIST

We used 5000 entries of MNIST, and applied DPMM. We give features to be pixel values of data. We set $K = 100$ for truncated stick-breaking process. We printed the labels of clustered data. The first 7 clusters are shown in Figure 5. We can see DPMM classified data based on its digits. The interesting fact is that DPMM tends to classify 1 and 7 to the same dataset and 4 and 9 to the same dataset. It is acceptable result since they look similar. We also show the reults of some entries of fourth clusters in Figure 6.

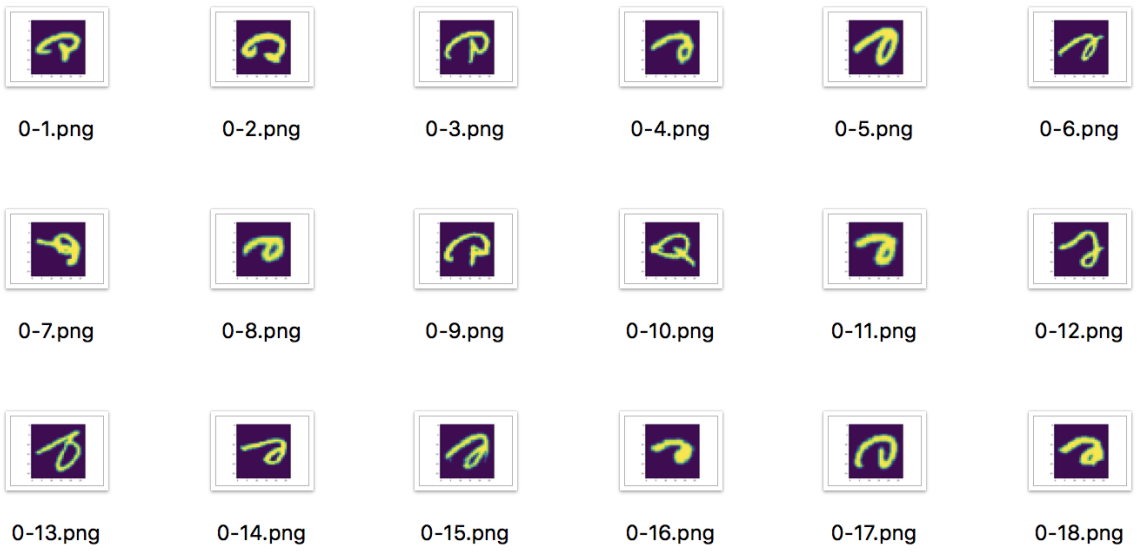
3.2.3 EMNIST

Similar with MNIST, we used 5000 entries of EMNIST, and applied DPMM. Samples of the results are shown in Figure 7 and 8. We can see some patterns between clusters. The first cluster consists of d-shaped figures, and the second cluster consists of 9-shaped figures, ... and so on. These seems well

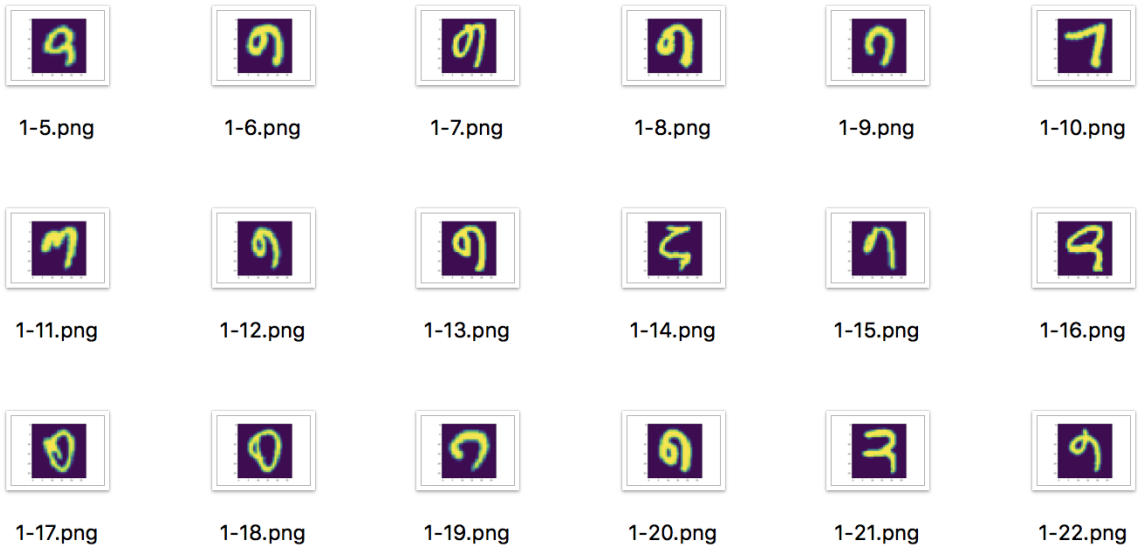
classified, as expected. However, it looks like that DPMM is not rotate-invariant. This can be cons when we want to use this to image classification. Also, both in MNIST and EMNIST, the similar sized clusters are too many made. This also can be cons in some cases.

References

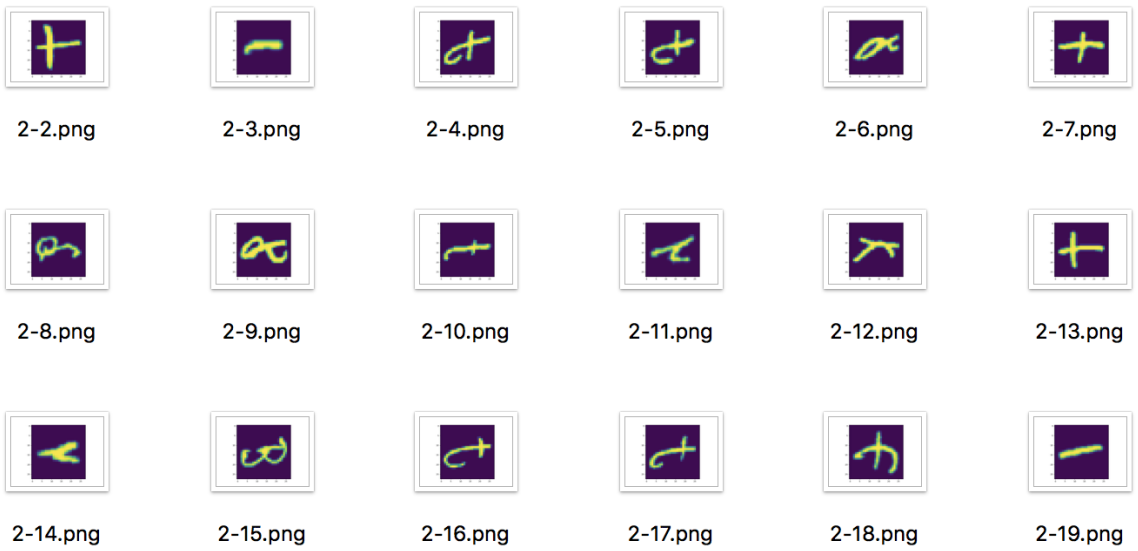
[1] David M Blei, Michael I Jordan, et al. Variational inference for dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143, 2006.



(a) 1st cluster

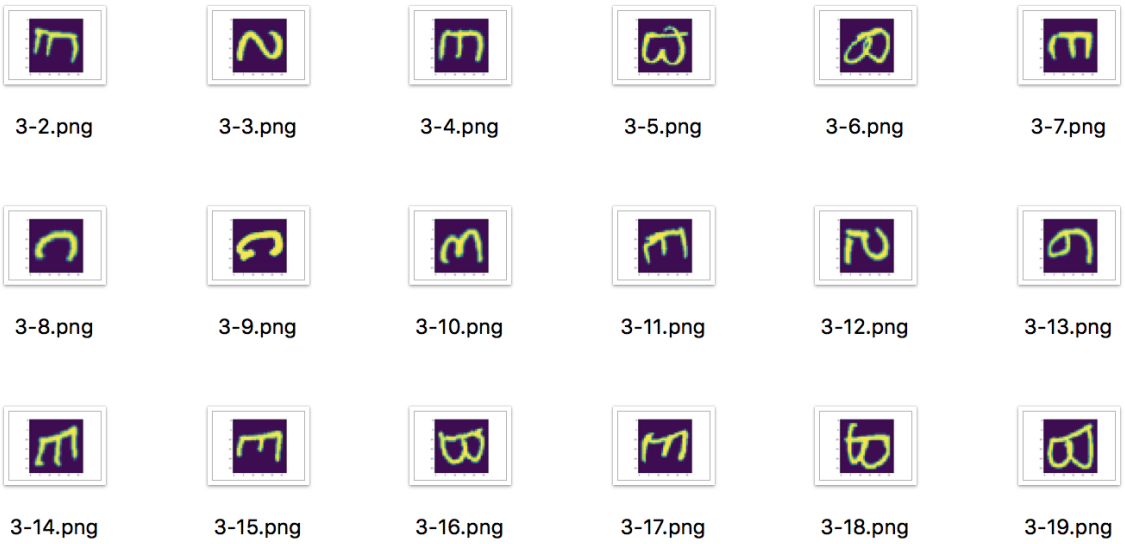


(b) 2nd cluster

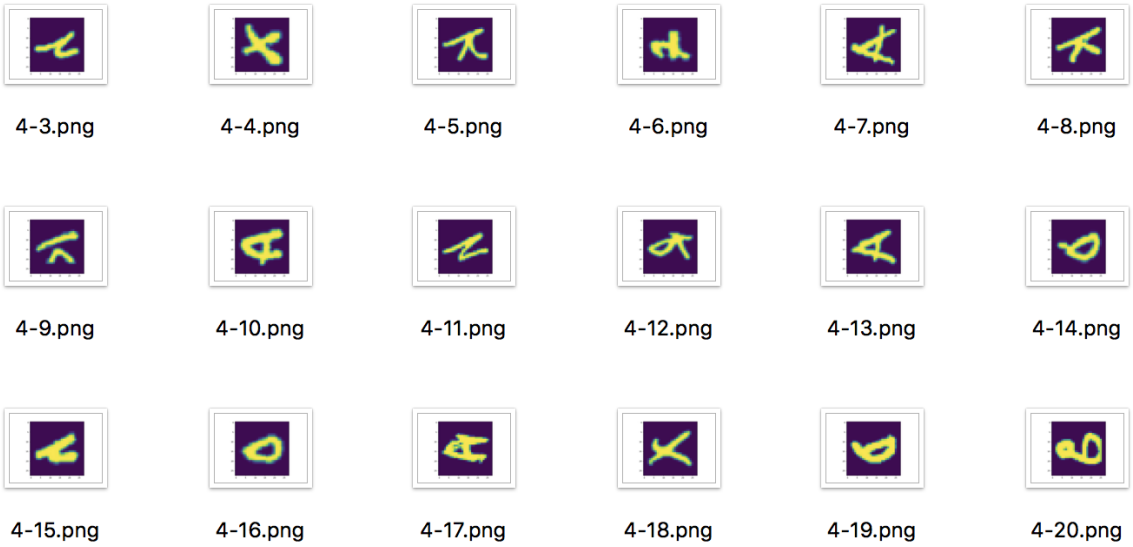


(c) 3rd cluster

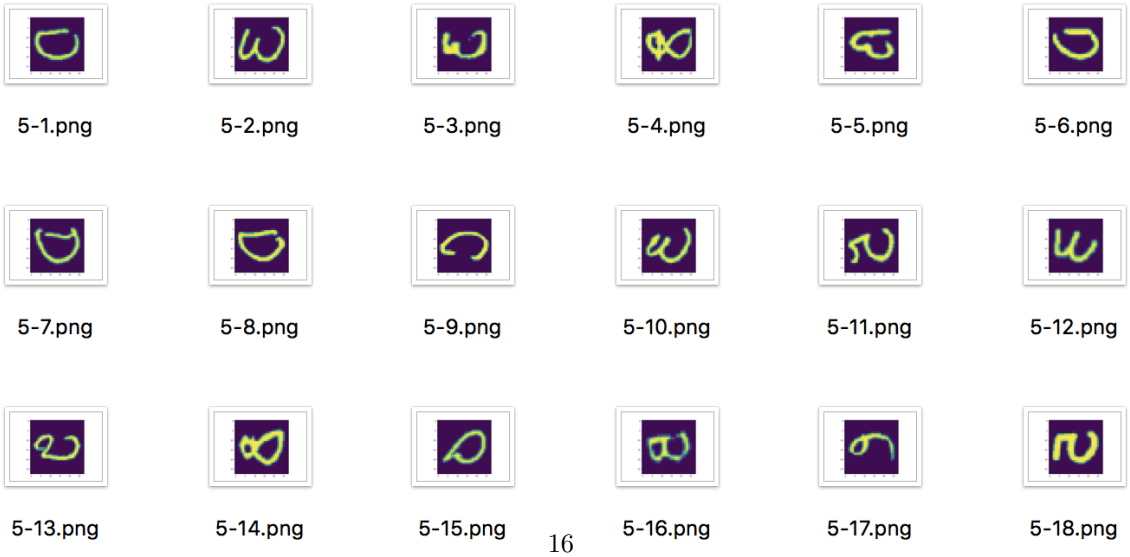
Figure 7: The result of six clusters of EMNIST



(a) 4th cluster



(b) 5th cluster



(c) 6th cluster

Figure 8: The result of six clusters of EMNIST