Non-parametric Bayesian Methods for Clustering

Blessen George 17111053 **Dushyant Kumar** 150242 **Pawan Kr. Patel** 14807453

Abstract

Most of machine learning algorithms is concerned with the problem of determining appropriate model to train data. This model selection problem constitutes the challenge of selecting models that discovers the causes and structure of underlying data without over-fitting and under-fitting it. This problem arises in many settings e.g. determining the number of clusters in clustering problem, how many dictionary atoms to use in a dictionary learning or sparse coding problem, the number of hidden states in a hidden Markov model. Non-parametric Bayesian methods(1) provides a principled way to address these model size selection questions, where the size of the model is allowed to grow with the data.

1. Introduction (1)

How do we design machine learning algorithms where the size of the model can be learned with data? For example, how many classes should I use in mixture models, how many latent variables should I use in latent variable model, how many factors should I use in factor analysis? Answers to these questions require us to carefully observe and analyze the data. A natural way to address to these questions is to evaluate model performance at various values of hyper-parameters(number of clusters in clustering problems, number of factors in factor analysis) involved in models and select one with best performance. However model selection also depends on the complexity of the model(simpler models with good performance are generally more favourable).

Bayesian Non-parametric(BNP, 1) models provide a different approach to answer above questions. Rather than comparing models with different complexity, BNP approach is to learn a non-parametric model that can adjust its complexity with the data. For example, consider the problem of clustering the data. Traditionally mixture modeling approach need to require to specify the number of cluster beforehand to cluster the data. The BNP models itself estimates the number of cluster needed to cluster the observed data and also allow future data to exhibit previously unseen clusters. The hidden structure grows with the data in BNP models which distinguishes itself from other Bayesian models. The model complexity is a part of the posterior distribution.

2. Problem Statement

The objective is to explore the non-parametric Bayesian model Dirichlet Process Mixture Model and it inference using Gibbs sampler for clustering the data. Furthermore to explore the sampling of concentration parameter α to get better performance.

3. Background and Literature Review (1, 3)

3.1 Dirichlet Process (3)

The Dirichlet Process (3) $\mathrm{DP}(\alpha, G_0)$ is a distribution over distributions with α as the concentration parameter and G_0 as the base distribution. Each sample of the Dirichlet process is an infinite discrete distribution with probability 1 over the sample space of the base distribution.

For a random distribution G to be distributed according to a DP, its marginal distributions have to be Dirichlet distributed. Specifically, for any finite measurable partition $A_1, \ldots, A_r$ of parameter space $\mathbf{\Theta}$. G is Dirichlet process distributed with base distribution G_0 and concentration parameter α, written as $G \sim DP(\alpha, G_0)$, if

$$(G(A_1), G(A_2), \ldots, G(A_r)) \sim \mathrm{Dir}(\alpha G_0(A_1), \alpha G_0(A_2), \ldots, \alpha G_0(A_r)) \tag{1}$$

$$\text{Where } \mathbb{E}[G(A)] = G_0.(A) \tag{2}$$

$$\mathrm{Var}[G(A)] = \frac{G_0(A)(1 - G_0(A))}{(\alpha + 1)} \tag{3}$$

The larger α is, the smaller the variance, and the DP will concentrate more of its mass around the mean, as $\alpha \to \infty$, we have $G(A) \to G_0(A)$.

3.1.1 Posterior of DP (3)

Let $G \sim DP(\alpha, G_0)$. Since G is a (random) distribution, we can draw samples from G itself. Let $\theta_1, \ldots, \theta_N$ be independent draws from G. We are interested in the posterior distribution of G given observed values of $\theta_1, \ldots, \theta_N$. Let $A_1, \ldots, A_r$ be a finite measurable partition of θ, and let $n_k = \#\{i : \theta_i \in A_k\}$ be the number of observed values in A_k . Since Dirichlet and the multinational distributions are conjugate and using the property, marginal of Dirichlet is also Dirichlet we have:

$$[G(A_1), \ldots, G(A_K)]|\theta_1, \ldots, \theta_N \sim \mathrm{Dir}(\alpha G_0(A_1) + n_1, \ldots, \alpha G_0(A_K) + n_K) \tag{4}$$

$$G|\theta_1, \ldots, \theta_N \sim \mathrm{DP}\left(\alpha + N, \frac{\alpha}{\alpha + N} G_0 + \frac{N}{\alpha + N} \frac{\sum_{i=1}^{N} \delta_{\theta_i}}{N}\right) \tag{5}$$

So, the posterior of DP is another DP with base distribution as a convex combination of the prior distribution and empirical distribution. The weight associated with the prior base distribution is proportional to α while weight for empirical distribution is proportional to the number of observations N.

3.1.2 Predictive Distribution (3)

Now, we are interested in finding **predictive distribution** $p(\theta_{N+1}|\theta_1, \ldots, \theta_N)$. For any measurable $A \subset \mathbf{\Theta}$, we have

$$\begin{aligned}
p(\theta_{N+1} \in A|\theta_1, \ldots, \theta_N) &= \int p(\theta_{N+1} \in A|G, \theta_1, \ldots, \theta_N) p(G|\theta_1, \ldots, \theta_N) dG \\
&= \int p(\theta_{N+1} \in A|G) p(G|\theta_1, \ldots, \theta_N) dG \\
&= \mathbb{E}[G(A)|\theta_1, \ldots, \theta_N] \\
&= \frac{\alpha}{\alpha + N} G_0(A) + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta_{\theta_i}(A)
\end{aligned}$$

Hence after marginalizing G we have

$$\theta_{N+1}|\theta_1,\ldots,\theta_N \sim \frac{\alpha}{\alpha+N}G_0 + \frac{1}{\alpha+N}\sum_{i=1}^{N}\delta_{\theta_i} \tag{6}$$

Therefore θ_{N+1} will be equal to a previous θ_i with probability proportional to $\sum_{j=1}^{N}\delta_{\theta_j=\theta_i}$, and will be a new value with probability proportional to α.

3.1.3 Interpretation of DP: Blackwell-MacQueen urn scheme (3)

The sequence of predictive distributions for $\theta_1, \theta_2, \ldots$ is called Blackwell-MacQueen urn scheme. We can use metaphor to interpret the equation 6. We can consider draws $\theta \sim G$ are balls having value as colour of the ball, and we have an urn to store the previously seen balls, so the process begin as follows:

In the very beginning we do not have any balls in the urn so we pick a colour $\theta_1 \sim G_0$. paint a ball with the observed colour and put it into the urn. let now consider the process at $(n+1)^{th}$ step, we pick a new colour with probability $\frac{\alpha}{\alpha+N}$, paint it and put into the urn. On the other hand with probability $\frac{N}{\alpha+N}$ we draw a ball from urn and paint a new ball with the same colour and drop both the ball into the urn. Well the drawing order of balls does not really affect the underlying probability so the order can be scuffled. With this scheme, existence of DP can be shown, we can create conditional distributions and using the Finett's theorem existence can be shown.

3.1.4 Clustering Property: Chinese Restaurant Process (1, 3)

The clustering property of DP can be seen the unique values of $\theta_1,\ldots,\theta_n$, which in turn induces the random partitions of n observations into m (let say) clusters. Since the distribution is random and this distribution over partitions is called Chinese Restaurant Process (CRP) .

CRP can be understood by simple metaphor of a restaurant having infinite number of tables, each of which can accommodate infinite number of customers. Now first customer comes in and sit on first table, and then new customers comes and decide to sit on already occupied table with the probability proportional to the number of customers (n_k) already seated on the table or on the new table with probability proportional to the α. Now we can calculate the expected number of clusters (m) and its variance, given number of observations (n) and concentration parameter (α)

$$\begin{aligned}
E[m|n] &= \sum_{i=1}^{n}\frac{\alpha}{\alpha+i-1} \\
&= \alpha(\psi(\alpha+n)-\psi(\alpha)), \psi \text{ is digamma function} \\
&\simeq \alpha\log\left(1+\frac{n}{\alpha}\right) \quad \text{for } n,\alpha \gg 0 \\
V[m|n] &= \alpha(\psi(\alpha+n)-\psi(\alpha)) + \alpha^2\left(\psi'(\alpha+n)-\psi'(\alpha)\right) \\
&\simeq \alpha\log\left(1+\frac{n}{\alpha}\right) \quad \text{for } n > \alpha \gg 0
\end{aligned}$$

The above equations shows how α controls the number of clusters to be generated by DP. As we can see that larger the α more the number of clusters, so we can actually control the actual number of clusters by setting appropriate value of α

3.1.5 Stick-breaking Construction (3, 4)

The random draws $\theta_1, \ldots, \theta_n$ can be considered as point masses and the long sequence of draws can be repeated by another draw as well. This implies that G itself is composed only of weighted sum of point masses i.e. G is a discrete distribution. We will show that discreteness by Stick-breaking construction.

The construction for $G \sim \mathrm{DP}(\alpha, G_0)$ is as follows:

$$
\begin{aligned}
&\textit{Step:1} \quad \beta_k \sim \mathrm{Beta}(1, \alpha) \\
&\textit{Step:2} \quad \pi_k = \beta_k \prod_{l=1}^{k-1} (1 - \beta_l) \\
&\textit{Step:3} \quad \theta_k^* \sim G_0 \\
&\textit{Step:4} \quad G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k^*}
\end{aligned}
$$

Informally the above process can be seen as recursively breaking the stick of unit length. start breaking the stick at length β_1 and assign it to π_1 and so on. Due to simplicity of the construction it has been used widely for various inferences about DP.

3.2 Hierarchical Dirichlet process (4)

Suppose we have different groups of data and we want to cluster each group using a DP mixture model. We would like to see the relationship between the clusters of different DP mixtures. Each DP mixture is represented by the cluster parameters θ_k, sharing the clusters among different DP mixture models means sharing the cluster parameter θ_k. In a DP mixture model, θ_k is drawn from a distribution which itself is drawn from Dirichlet process, which means each of these distribution must be drawn from Dirichlet process with common base distribution G_0 and common concentration parameter α_0. If we allow G_0 to be continuous then each draw from $DP(\alpha_0, G_0)$ will be unique and therefore won't be able to share the parameters. So, G_0 should be a discrete distribution and the most obvious solution is to draw G_0 from another Dirichlet process. Note that drawing G_0 from a Dirichlet process doesn't put a limit on number of clusters. The generative model for HDP (4) is as follows:

$$
\begin{aligned}
&G_0|H \sim DP(\gamma, H) && \\
&G_i|G_0 \sim DP(\alpha_0, G_0) && \forall i \\
&\theta_{ij}|G_i \sim G_i && \forall i, j \\
&x_{ij}|\theta_{ij} \sim p(x_{ij}|\theta_{ij}) && \forall i, j
\end{aligned}
$$

3.3 Dirichlet Process Mixture Model (DPMM) (3, 4)

The clustering problem is generally modelled as a mixture model and Dirichlet Processes naturally models infinite mixture models. In the clustering problem, we have a dataset $\mathbf{X} = \{x_n\}, n \in [N]$ with each point associated with θ_n denoting the parameter of cluster that the point belongs to. The generative story is as follows:

$$
\begin{aligned}
G &\sim \mathrm{DP}(\alpha, G_0) \\
\theta_n &\sim G \\
x_n &\sim p(x_n|\theta_n)
\end{aligned}
\tag{7}
$$

3.4 Inference in DPMM (2)

We use gibbs sampling for the inference of parameters $\theta_1, \ldots, \theta_N$. We repeatedly draw values for each θ_i's from it's conditional distribution given both data and the θ_j for $j \neq i$. Deriving CP i.e. $p(\theta_i|\theta_{-i}, \mathbf{X})$

$$\begin{aligned} p(\theta_i|\theta_{-i}, \mathbf{X}) &\propto p(\theta_i|\theta_{-i})p(x_i|\theta_i) \\ &= \frac{1}{\alpha + N - 1}\sum_{j \neq i} p(x_i|\theta_j)\delta_{\theta_j} + \frac{\alpha}{\alpha + N - 1}(\int p(x_i|\theta)p(\theta|G)d\theta)\ p(\theta|x_i) \end{aligned}$$

Thus, the **algorithm 1** is as follows:

Algorithm 1 Slow convergence Gibbs sampling algorithm

Input: $\mathbf{X}$, α
Output: Markov Chain of $\theta = \{\theta^{(t)}\}_{t=0}^{T}$

1. Initialize $\theta_i = \{\theta_i^{(0)}\}_{i=0}^{N}$
2. for t=1, . . . ,T
   - Sample $\theta_1^{(l)} = p(\theta_1|\theta_2^{(l-1)}, \ldots, \theta_N^{(l-1)}, \mathbf{X})$
   - Sample $\theta_2^{(l)} = p(\theta_2|\theta_1^{(l)}, \theta_3^{(l-1)} \ldots, \theta_N^{(l-1)}, \mathbf{X})$

     $\vdots$
   - Sample $\theta_i^{(l)} = p(\theta_i|\theta_1^{(l)}, \ldots, \theta_{i-1}^{(l)}, \theta_{i+1}^{(l-1)}, \ldots \theta_N^{(l-1)}, \mathbf{X})$

     $\vdots$
   - Sample $\theta_N^{(l)} = p(\theta_N|\theta_1^{(l)}, \ldots, \theta_{N-1}^{(l)}, \mathbf{X})$

Algorithm 1 leads to slow convergence (2)
Reason: In this way, convergence to the posterior distribution may be rather slow and sampling therefore may be inefficient. The problem is that there are often groups of observations that with high probability are associated with same θ. Since algorithm can't change the θ for more that one observation simultaneously, a change to the theta values for observations in such a group can only change rarely, as such a change requires passage through a low-probability intermediate state in which observations in the group do not all have the same θ value.

4. Towards Faster Convergence (2, 3)

4.1 Infinite Mixture Model (2, 3)

Model equivalent to (7) can also be obtained by taking the limit as K goes to infinity of finite mixture models with K components. Recall that finite mixture model has the following generative story:

$$\begin{aligned} \pi &\sim \text{Dir}(\alpha/K, \ldots, \alpha/K) \\ \phi_k &\sim G_0 \\ z_i &\sim \text{Multinoulli}(\pi) \\ x_i &\sim p(x_i|\phi_{z_i}) \end{aligned}$$

where π is mixing proportion, ϕ_k are the parameters of cluster k, z_i is latent variable associated with x_i which is basically cluster id of x_i.

4.2 Inference (2, 3)

We have derive collapsed version of Gibbs sampler for this version of Infinite Mixture Model in the class. **Algorithm 2** is as follows

---

Algorithm 2 Faster convergence Gibbs sampling

Input: $\mathbf{X}$, α

Output: Markov Chain of $\phi, \mathbf{Z} = \{\phi^{(t)}, Z^{(t)}\}_{t=0}^{T}$

1. Initialize $\phi = \{\phi_k^{(0)}\}_{k=1}^{K}$ & $Z = \{z_i^{(0)}\}_{i=1}^{N}$ randomly.
2. for t=1,...,T

- For each observation $i = 1, \ldots, N$, Sample the cluster id $z_i^{(t)}$ as

$$p(z_i = k | Z_{-i}^{(t-1)}, \phi^{(t-1)}, \mathbf{X}) \propto n_k^{(t-1)} p(x_i | \phi_k^{(t-1)}) = \hat{\pi}_{ik} \quad (k = 1, \ldots, K)$$

$$p(z_i = k_{new} | Z_{-i}^{(t-1)}, \phi^{(t-1)}, \mathbf{X}) \propto \alpha \int p(x_i|\phi) p(\phi|G_0) d\phi = \hat{\pi}_{ik_{new}}$$

$$z_i^{(t)} \sim \text{Multinoulli}(\hat{\pi}_{i1}, \ldots, \hat{\pi}_{ik_{new}})$$

$$\text{if } z_i = k_{new}$$

$$\text{set} \quad K = K + 1$$

$$\text{Sample} \quad \phi_K^{(t-1)} \sim p(\phi_k | x_i)$$

- Sample mixture components mixture parameters $\phi_k^{(t)}$ as :

$$p(\phi_k | \mathbf{X}, Z^{(t)}, \phi_{-k}^{(t-1)}) \propto p(\phi_k) \prod_{i=1}^{N} \mathbb{I}[z_i^{(t)} = k] p(x_i | \phi_k^{(t-1)})$$

$$\phi_k^{(t)} \sim p(\phi_k | \mathbf{X}, Z^{(t)}, \phi_{-k}^{(t-1)})$$

In the experimentation, ϕ_k denotes the means of clusters i.e $\phi_k = \mu_k$. We have taken $G_0 = \mathcal{N}(\mu_0, \rho^2 \mathbf{I})$ & $p(x|\mu_k) = \mathcal{N}(\mu_k, \sigma^2 \mathbf{I})$. So our final equations become

$$p(z_i = k | Z_{-i}^{(t-1)}, \phi^{(t-1)}, \mathbf{X}) \propto n_k^{(t-1)} \mathcal{N}(x_i | \mu_k^{(t-1)}, \sigma^2 \mathbf{I}) = \hat{\pi}_{\mathbf{ik}} \quad (\mathbf{k = 1}, \ldots, \mathbf{K})$$

$$p(z_i = k_{new} | Z_{-i}^{(t-1)}, \phi^{(t-1)}, \mathbf{X}) \propto \alpha \int \mathcal{N}(x_i | \mu, \sigma^2 \mathbf{I}) \mathcal{N}(\mu | \mu_{\mathbf{0}}, \rho^{\mathbf{2}} \mathbf{I}) \mathbf{d}\mu$$

$$\propto \alpha \mathcal{N}(x_i | \mu_0, (\rho^2 + \sigma^2)\mathbf{I}) = \hat{\pi}_{\mathbf{ik_{new}}}$$

$$p(\mu_k | \mathbf{X}, Z^{(t)}, \mu_{-k}^{(t-1)}) \propto p(\mu_k) \prod_{i=1}^{N} \mathbb{I}[z_i^{(t)} = k] p(x_i | \mu_k^{(t-1)})$$

$$= \mathcal{N}(\mu_k | \mu, \sigma^2 \mathbf{I})$$

$$\text{where} \quad \frac{1}{\sigma^2} = \frac{1}{\rho^2} + \frac{\sum_{i=1}^{N} \mathbb{I}[z_i^{(t)} = k]}{\sigma^2}$$

$$\& \quad \mu = \frac{\sigma^2}{\sum_{i=1}^{N} \mathbb{I}[z_i^{(t)} = k] \rho^2 + \sigma^2} \mu_0 + \frac{\rho^2}{\sum_{i=1}^{N} \mathbb{I}[z_i^{(t)} = k] \rho^2 + \sigma^2} \sum_{i=1}^{N} \mathbb{I}[z_i^{(t)} = k] x_i$$

5. Estimation of hyperparameter α (5)

α is a critical hyperparameter in Dirichlet process mixture model that strongly influences resulting inferences about numbers of mixture components. In this section we will review the paper(5) by Mike West where he derived the posterior of α in a simple conditional form from which α can be easily simulated.As a result, we can integrate the inference of α in our existing routine Gibbs sampling algorithm.

Suppose we have sampled values of the parameter K(no. of clusters) and all the other model parameters. From our model, the data X are initially conditionally independent of α given all the other parameters. The conditional posterior of α is written as:

$$\begin{aligned} p(\alpha|X,\pi,K) &\propto p(\alpha|K) \\ &\propto p(\alpha)p(K|\alpha) \end{aligned} \tag{8}$$

And $p(K|\alpha) = c_n(k)n!\alpha^K \frac{\Gamma(\alpha)}{\Gamma(\alpha+N)}$ which follows from the results of Antoniak (1974). N is total no. of data points. Since α is a positive quantity so lets take gamma prior i.e. $\alpha \sim G(a,b)$. Using a sequence of mathematical equations, author have shown that α is generated from the mixture of gamma distributions. Following are the steps for the sampling of α.

- first sampling an x value from the simple beta distribution (9) conditional on α and K fixed at their most recent values
$$x|\alpha,K \sim B(\alpha+1,n) \tag{9}$$
- Compute the mixing proportional probabilities as follows
$$\frac{\pi_x}{(1-\pi_x)} = \frac{(a+K-1)}{N(b-\log(x))} \tag{10}$$
- Sampling the new α value from the mixture of gammas in (11) based on the same K and the x value just generated in step 1.
$$\alpha|x,K \sim \pi_x G(a+K, b-\log(x)) + (1-\pi_x)G(a+K-1, b-\log(x)) \tag{11}$$

Now adding the sampling of α in Algorithm 2 to produce algorithm 3 which is as follows:

Algorithm 3 Integrated faster convergence Gibbs sampling (2, 5)

Input: $\mathbf{X}$

Output: Markov Chain of $\alpha, \phi, \mathbf{Z} = \{\alpha^{(t)}, \phi^{(t)}, Z^{(t)}\}_{t=0}^{T}$

1. Initialize $\alpha = \alpha^{(0)}$, $\phi = \{\phi_k^{(0)}\}_{k=1}^{K}$ & $Z = \{z_i^{(0)}\}_{i=1}^{N}$ randomly.
2. for t=1,...,T
   - For each observation $i = 1, \ldots, N$, Sample the cluster id $z_i^{(t)}$ as

$$p(z_i = k | Z_{-i}^{(t-1)}, \phi^{(t-1)}, \mathbf{X}) \propto n_k^{(t-1)} p(x_i | \phi_k^{(t-1)}) = \hat{\pi}_{ik} \quad (k = 1, \ldots, K)$$

$$p(z_i = k_{new} | Z_{-i}^{(t-1)}, \phi^{(t-1)}, \mathbf{X}) \propto \alpha^{(t-1)} \int p(x_i|\phi) p(\phi|G_0) d\phi = \hat{\pi}_{ik_{new}}$$

$$z_i^{(t)} \sim \text{Multinoulli}(\hat{\pi}_{i1}, \ldots, \hat{\pi}_{ik_{new}})$$

$$\text{if } z_i = k_{new}$$

$$\text{set} \quad K = K + 1$$

$$\text{Sample} \quad \phi_K^{(t-1)} \sim p(\phi_k | x_i)$$

   - Sample mixture components mixture parameters $\phi_k^{(t)}$ as :

$$p(\phi_k | \mathbf{X}, Z^{(t)}, \phi_{-k}^{(t-1)}) \propto p(\phi_k) \prod_{i=1}^{N} \mathbb{I}[z_i^{(t)} = k] p(x_i | \phi_k^{(t-1)})$$

$$\phi_k^{(t)} \sim p(\phi_k | \mathbf{X}, Z^{(t)}, \phi_{-k}^{(t-1)})$$

   - Sample $\alpha^{(t)}$ as follows:

$$x | \alpha^{(t-1)}, K \sim B(\alpha^{(t-1)} + 1, N)$$

$$\frac{\pi_x}{(1 - \pi_x)} = \frac{(a + K - 1)}{N(b - \log(x))}$$

$$\alpha^{(t)} | x, K \sim \pi_x G(a + K, b - \log(x)) + (1 - \pi_x) G(a + K - 1, b - \log(x))$$

6. Experiments

In order to study how Dirichlet Process Mixture Models work in practice, we created a toy dataset of 2-D datapoints drawn from a mixture of gaussians. The DPMM was expected to produce the approximately correct number of clusters and the correct cluster ids for each of the data points if the initialization was done correctly. We run the algorithm 2.

6.1 Dataset specification

The dataset comprises of 2d data points drawn from 20 different gaussians with 800 data points in each given by the following.

$$\mu_k \sim N(\mu_0, \sigma^2 I) \quad \forall k \in [1, 20] \quad \text{where } \sigma^2 = 150 \ \& \ \mu_0 = (0, 0)$$
$$x_{ki} \sim N(\mu_k, I) \quad \forall i \in [1, 800]$$
$$X = \{x_{ki}\} \quad \forall k \in [1, 20], \ \forall i \in [1, 800]$$

Such large value of σ^2 is chosen so that cluster means μ_k are well separated. But still if you see fig 1, some cluster means are very close to each other which could really confuse the algorithm for assigning cluster ids to some of points belonging to these close clusters.

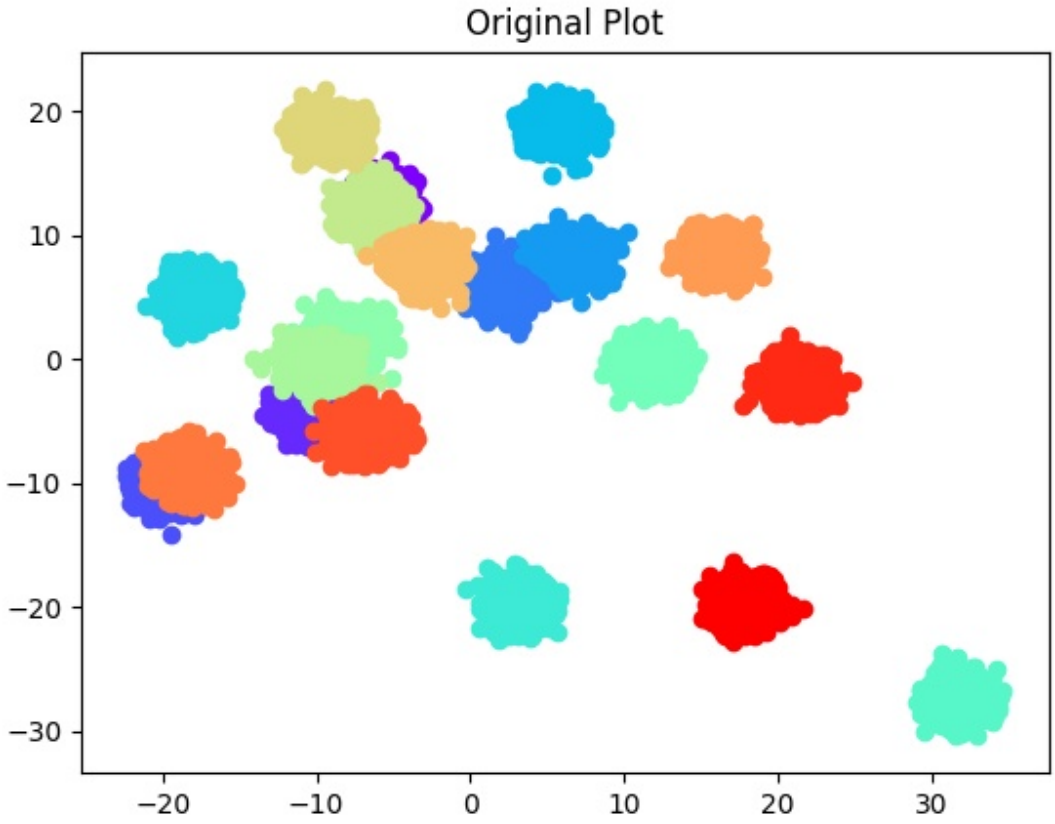


Figure 1: True 20 clusters toy datasets. X-axis represent x-coordinate and y-axis represent y-coordinate of dataset. Some of the clusters on the left side are overlapping therefore algorithm might under-estimate the no. of clusters. The clusters on the right side are well separated so we can expect our algorithm to identify them correctly.

6.2 Model Specification

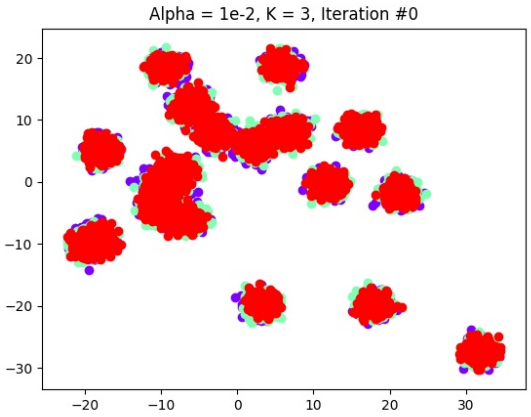
The good thing when working with toy dataset is that one can provide our model a nice initialization because we already know how the data is generated. The MCMC algorithms (in our case Gibbs Sampling) has a very high dependency on the initialization and sometime you can question your implementation as the results are not expected due to improper initialization. This has also happen to us when we try to run our implementation on MNIST dataset. More about this failure case, we will discuss in section 7.

We use a range of initial values of K and α values to conduct the experiment. Rightnow we are not considering the sampling of α. We will just fix some value of α.
K = [1, 3, 5, 10]
α = [1e-20, 1e-10, 1e-2, 1, 2]

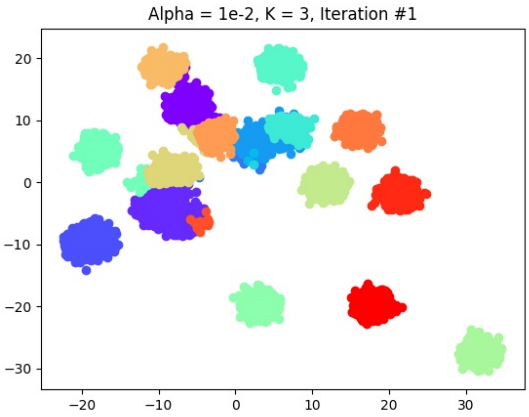
We apply the Gibbs sampling procedure (Algorithm 2) using the equations for conditional posteriors as mentioned above. We run the Gibbs sampling for each K and α pair for fixed 30 no. of iterations.

6.3 Results

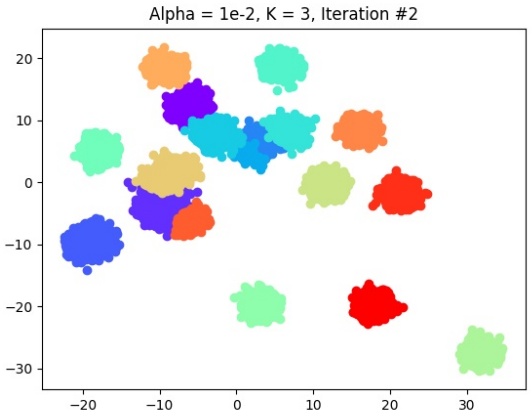
Best result is obtained for α = 1e-2 and are shown below.



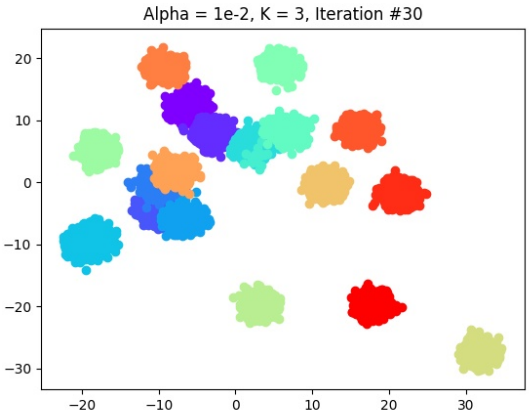
(a) Random 3 clusters initialization.



(b) Total 20 custers are formed in iteration 1.



(c) Total 18 clusters remains in iteration 2.



(d) Finally we 19 clusters after 30th iteration.

Figure 2: Evolution of clusters in Gibbs sampling algorithm. Initially we have 3 random clusters and finally the no. of clusters become 19 pretty close to actual number which was 20. Algorithm really did perform good on overlapping clusters as well. It correctly find out the exact no. of clusters in those overlapping regions probably due to our accurate initialization of clusters variances.

Following is the table 1 shows the no. of clusters obtained for different pairs of α & initial K value.

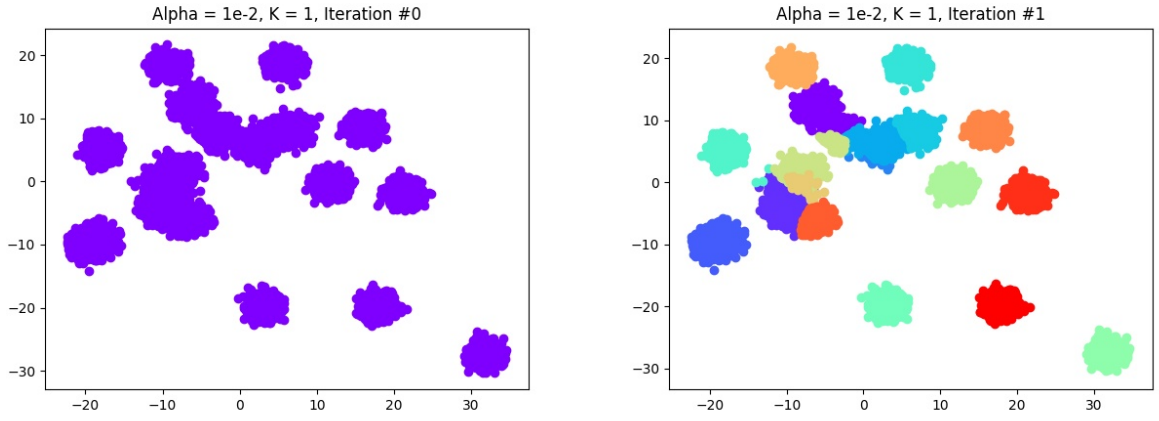
Initial value of K	α = 1e-20	α = 1e-10	α = 1e-2	α = 1	α = 2
1	8	11	17	32	52
3	10	12	19	38	55
5	12	15	18	36	45

Table 1: Table showing the clusters formed from parameters values. As we can see that initial values of K doesn't have any affect on the final no. of clusters formed. Optimal value of α is 1e-2. As evident from this table that as we increase the value of concentration parameter α, more no. of clusters are being formed.

6.4 Insights from experiment

Following are beautiful insights that we have gained from this experiment:

- Algorithm 2 converges very fast with proper initialization. If the clusters are well separated, even 2-3 iteration of gibbs samping would suffice. This is evident from fig 2a & 2b also from 3a & 3b where well separated clusters are captured in only on iteration..



(a) All points assigned to same clusters initially.

(b) Clusters formed after 1 iteration.

Figure 3: As we can see from these figures that well separated clusters are captured in 1st iteration only. So if dataset has well separated clusters then with proper initialization, one just need to run 2-3 iterations only.

- Gibbs sampling algo exhibits a nice property where new clusters can be formed and old clusters can disappear during an iteration. Disappearance of old clusters are due to the fact the some point have created it and no other points have supported it then this point itself leaves this clusters and joins some other existing clusters.
- Almost all the time, as iteration progress data points gets associated to new clusters and initialized clusters are completely ignored i.e. no point is associated with those initialized clusters.

7. Failure Cases

7.1 MNIST dataset

In order to evaluate our DPMM model, we initially started with the MNIST dataset. The goal was to begin with an arbitrary number of clusters and allow the DPMM to find the right number of clusters along with the cluster means. But running our DPMM model on the dataset, we noticed that the results were quite noisy. We feel it was because the MNIST digit images form a complex structure in the high dimensional image space that wasn't accurately captured by our infinite gaussians mixture model. After this, we decided to scale down to a simpler dataset, the 2 dimensional dataset.

7.2 Alpha hyperparameter estimation

In order to have the α hyperparameter automatically tuned, we attempted to use Algorithm 3. However, we found that the algorithm exhibits unbounded growth for the α hyper parameter. We found that there is a reinforcement loop between the number of clusters, K chosen and the α. As seen in equation (11) as α increases the number of clusters chosen K increases, which in turn causes a further increase in α.

8. Learnings and Future work

We learnt the concepts of Dirichlet Process and Hierarchical Dirichlet Process and got an handful experience of how DP works in practice. Proper initialization is very necessary in MCMC methods. We have also learnt the sampling of concentration parameter α.

As future work, we can look into more on our MNIST failure case and try to use more general co-variance matrices and learn cluster co-variance matrix also. We can also try out other alpha estimation methods.

References

[1] Samuel J. Gershman and David M. Blei. A tutorial on bayesian nonparametric models, 2012. URL `https://www.sciencedirect.com/science/article/pii/S002224961100071X`.

[2] Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

[3] Yee Whye Teh. Dirichlet process, 2010. URL `https://www.stats.ox.ac.uk/~teh/research/npbayes/Teh2010a.pdf`.

[4] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. 2005.

[5] Mike West. Hyperparameter estimation in dirichlet process mixture models. URL `http://www2.stat.duke.edu/~mw/.downloads/DP.learnalpha.pdf`.