Different Approaches to Analyze Dirichlet Process Mixture Model

Jifan Zhang

under the direction of
Professor Jun Zhu
Department of Computer Science and Technology
Tsinghua University

Research Science Initiative-Tsinghua
August 18, 2015
Abstract

Gaussian Mixture Model (GMM) that requires the pre-decided parameter about number of clusters has a significant limitation when dealing with high dimensional clustering tasks. Dirichlet Process (DP) is thus applied to the model to construct a Dirichlet Process Gaussian Mixture Model (DPGMM). This allows the clustering tasks to generate the clusters by a more intuitive hyper-parameter. Furthermore, the Gibbs Sampling method is also applied to the DPGMM, adding a stochastic feature for the model to generate different sets of clusters one out of a time. This further reduced the effectiveness of the hyper-parameter over the model. In this paper, both GMM and DPGMM are complemented. The efficiency of DPGMM is displayed and the convergence of Gibb’s Sampling Method is proved by experiments.

Summary

Clustering works has been wildly used in real world in many aspects of our lives. One of the very reasonable and well-developed clustering models is the Gaussian Mixture Model (GMM). GMM, however, has some limitations since people need to decide how many clusters there are. This requires an understanding of the data structure, i.e., how the data are distributed. Understanding the data structure will often be really hard especially when one data point has too many features. The Dirichlet Process (DP) is thus used to form the Dirichlet Process Gaussian Mixture Model (DPGMM) where the DP constructs some probability distribution for creating a new cluster. DPGMM then can generate out the number of clusters automatically. A stochastic method is further used called Gibb’s Sampling method. This method allows the program to generate different sets of clusters for the same data points and parameters. In our work, we showed the advantage of the DPGMM compare to the GMM. Also, we verified that the stochastic process would converge after a large number of iterations.
1. Introduction

Clustering algorithms are widely used in the real world today. For example, social networks analyze their users’ data, such as the books they would like to read, the songs they would like to listen, and the films they would like to watch. All these information is known as a person’s “data.” Then assuming there are $n$ features that were collected about a single person, the company put each data point (consists of a person’s features) into an $n$-dimensional space. Thus, those data points with similar features, meaning people who have similar interests, will be close to each other. By grouping these data points (people’s features) by closeness, each person in a group will share similar interests with others in the same group. Now the website will recommend these people to become friends with each other since they probably have similar personalities or daily life events. The personalities and daily life events are the “hidden” regularities among the data found on the social website.

In general, clustering algorithms, also known as unsupervised learning algorithms, aims to group the unlabeled data into sets of data points according to their distances with each other as well as their density. By applying such an algorithm to specific tasks, different clusters of data would result in some “hidden” regularities among the data, i.e., the personalities and daily life events in the social website example. In this paper, the clustering tasks that are achieved by building Mixture Models will be specifically addressed. Here, Mixture Models refers to a technique used for expressing a sophisticated probability distribution in a model by dissolving it into simpler forms of probability distributions. Such models have been widely used in different machine learning tasks such as density estimations [3] and clustering tasks. In the real-life tasks, Mixture Models with infinite components are relatively more common and useful, i.e., sequential data tasks (data continuous inputting over time) [8].

Lloyd proposed the classic K-mean algorithm in his paper in 1982 [7], which sometimes results in imprecise groupings according to different initial values. Later on, Gaussian Mixture Model (GMM) is proposed where each cluster corresponds with a Gaussian Probability Distribution [2,8]. The Gaussians are used to denote how the data points distribute in one specific cluster. Finally in 2000, Rasmussen and Neal wrote papers [9,10] that applied Dirichlet Process to GMM, which is called Dirichlet Process Gaussian Mixture Model (DPGMM). Both of the former two algorithms, K-mean algorithm and GMM, has to decide the number of clusters prior to train the model by data. Since the number of clusters is pre-decided, people have to gain an idea of how is the data generally distributed, which is
extremely hard as the dimension of data points growing higher. DPGMM, however, instead of setting the number of clusters, goes through a stochastic process and generates different trials. By taking a mean of the results, the number of clusters will be decided automatically. The model DPGMM does not need the parameter for number of clusters, and thus it is a type of Nonparametric Bayesian Model (NBM). Semantically, “nonparametric” means without pre-deciding the number of parameters, which therefore allow a more flexible prior assumption about the data; and “Bayesian” refers to that the Model is built under a Bayesian framework, which can prevent the problem of overfitting in Maximum Likelihood Estimation (MLE) methods [6].

Several criteria are used to test whether a model is sufficiently efficient: computational complexity, which will directly relate to the amount of time and storage capacity needed for performing such task; sensibility of hyper-parameters, which is how much a pre-determined parameter can influence the result of the whole model; choice of the base function, which is a input, also known as a hyper-distribution because of the need for the distribution to be pre-decided; and the accuracy of a model, which is presented on a set of data. In 2015, Ge wrote the paper that applied a distributed method on the computation of DPGMM [4], which dramatically increased the speed of computing while retaining a state-of-art performance on accuracy. In 2010, Görür tested both non-conjugate prior and conjugate prior that were used for the base function of DP [5], and the paper presented a series of comparison between them. It presented a non-conjugate prior that resulted in no or modest amount of loss of complexity, and equally good accuracy.

In this paper, we discuss about GMM in. DPGMM will be presented in 2.2 by going through the process of: 2.2.1 Dirichlet Process (DP), where an alternative view of DP will be presented; 2.2.2 Dirichlet Process Gaussian Mixture Model, where the model will be discussed mathematically in detail; and 2.2.3 Gibb’s Sample Process, where a pseudocode will be shown as well as the explanation for its functionality. In section 3, the algorithms are implemented and the experiments were conducted to test the advantages of DPGMM and convergence of Gibb’s Sampling algorithm. The results are described in detail in the section. In section 4, we discuss about the collected results with an analysis about the observations.

2. Dirichlet Process for Gaussian Mixture Model (DPGMM)

Here are the notations for the entire derivation. We denote all of the observed data by: $x = \{x_1, x_2, \ldots, x_N\}$. Notice that here we use $N$ to denote the total number of observed data
points. Since each data could have several different features or components, thus for any $i$ that satisfies $i \leq N$, $x_i$ could be a multi-dimensional vector. We also define some latent features here by $z = \{z_1, z_2, \ldots, z_N\}$. For each $z_i$, we have $z_i \in \{1, \ldots, K\}$, which is defined as the cluster index of the $i$-th data point. $K$ is used to denote the number of clusters. For all clusters, each corresponding to a multivariate Gaussian distribution, the mean is denoted by $\mu = \{\mu_1, \mu_2, \ldots, \mu_K\}$. In section 2.1 for Gaussian Mixture Model, $K$ is a constant since it is pre-decided by humans. Therefore, it is a parameter. But since in 2.2 Dirichlet Process Gaussian Mixture Model (DPGMM), a stochastic process will determine the number of clusters, $K$ will then be a variable and will have different values stochastically.

2.1 Gaussian Mixture Model (GMM)

The Central Limit Theorem states that for an infinitely large amount of data with independent variables, well-defined expected values and well-defined variances, the arithmetic mean of the data will be approximately normally distributed. Most distributions of infinite number of data can be assumed to be normally distributed. For a specific data point $x_n$, its probability of inside the $k$-th cluster will be $x_n \sim \mathcal{N}(\mu_k, \Sigma_k)$. $\mathcal{N}()$ is a normal distribution probability function with specific mean and covariance. In this work, the covariance matrices will all be set to be identity matrices, $I$, due to the complexity of calculating the inverse of a covariance matrix for a multivariate Gaussian distribution. The high complexity can cause a dramatic increase in the amount the programs take.

2.2 Dirichlet Process Gaussian Mixture Model (DPGMM)

2.2.1 Dirichlet Process

A formal definition of Dirichlet Process: For a DP with a base probability distribution function $G_0$ and a concentration parameter $\alpha$, if a set of data $x$ is generated by $x \sim DP(\alpha, G_0)$. Then for any partition of $x$, $\{x_i\}_{i=1}^m$ they all satisfy:

$$
(x(x_1), x(x_2), \ldots, x(x_m)) \sim \text{Dir}(\alpha G_0(x_1), \alpha G_0(x_2), \ldots, \alpha G_0(x_m))
$$

where $\text{Dir}()$ is a Dirichlet Distribution.

An simpler way of understanding the Dirichlet Process, however, is to alternatively think it as a Chinese Restaurant Process (CRP), which was first proposed by Aldous in 1983 [1]. In CRP, it is assumed that there is a Chines Restaurant with infinite many tables. The first customer arrives and sits at the first table. Later, for the $n+1$-th customer, it has the
probability of $p = \frac{n_k}{\alpha + n - 1}$ to sit at the $k$-th table. Notice, here $n$ can mean the number of people currently in the restaurant. $n_k$ denotes the number of people who sit at the $k$-th table. There’s also a probability of $p = \frac{\alpha}{\alpha + n - 1}$ for the new customer to sit at a new table, which is the $(K+1)$-th table. Here $K$ is the number of tables with at least one customer. Overall $\alpha$ is the parameter for controlling how likely it is for a person to sit at a new table. It is called a hyper-parameter, meaning the parameter that controls the overall process.

2.2.2 Dirichlet Process Gaussian Mixture Model

As mentioned in the introduction, a DPGMM is a type of Nonparametric Bayesian Model. The nonparametric feature can be implemented by using a CRP since the CRP does not need a pre-set parameter to represent the number of clusters, which is not intuitive for people most of the time. Instead, we will use the CRP, a stochastic process, to sample the data’s indices and automatically get the number of clusters indirectly. We use a hyper-parameter $\alpha$ that is more intuitive to people, in the way that describe how probable the model will cause the data to be assigned in a new cluster. Recall that for latent variables $z = \{z_1, z_2, \ldots, z_N\}$, $z_i \in \{1, \ldots, K\}$, which is the index for the $i$-th data point’s cluster. So for example, $z_p = q$ means the $p$-th data is in the $q$-th cluster. Thus the distribution over the $i$-th latent variable $z_i$ is:

$$z_i \sim CRP(\alpha)$$ (2)

As mentioned in section 2.1, each cluster corresponds with a multivariate Gaussian distribution. Given the mean and latent variable for a specific point, $x_i$ is distributed on a Gaussian as:

$$x_i | z_i, \mu_{z_i} \sim N(\mu_{z_i}, I)$$ (3)

Where the covariance matrix $I$ is set to be an identity matrix.

Now the derivation will present DPGMM’s “Bayesian” feature of the model. For a given set of data $x$, $p(x)$ is a constant term, which can be regard as a normalization term in a Bayesian framework:

$$P(\cdot | x) = \frac{P(x | \cdot) P(\cdot)}{P(x)} \propto P(x | \cdot) P(\cdot)$$ (4)
By applying such framework and independence between variables $\mu$ and $z$, the probability distribution can be obtained when a CRP assigns a data point to an existing cluster:

$$p(z_i = k \mid z_{-i}, \mu_k, x) \propto p(x_i \mid \mu_k, I, z_i, z_{-i}) \cdot p(z_i = k \mid z_{-i}) = \frac{n_k}{\alpha + n - 1} \mathcal{N}(x_i \mid \mu_k, I)$$

(5)

where $z_{-i}$ denotes all of the latent variable $z$ except for the $i$-th.

When a CRP assigns a data point into a new cluster, the derivation becomes more complex. Here we will only show some simple steps, and the full derivation is given in Appendix. The distribution for $z_i=k$ is:

$$p(z_i = K+1 \mid z_{-i}, x) \propto p(x_i \mid \mu_{K+1}, I, z_i, z_{-i}) \cdot p(\mu_{K+1}) d\mu_{K+1}$$

$$= \frac{\alpha}{\alpha + n - 1} \cdot \frac{1}{2^{\frac{D}{2}}} \cdot \frac{1}{\pi^{\frac{D}{2}}} e^{-\frac{1}{2} x_i^T x_i}$$

(6)

Here for computational simplicity, the probability $p(\mu_{K+1})$ is chosen to be the conjugate prior of a multivariate Gaussian distribution: $\mu_{K+1} \sim \mathcal{N}(0, I)$.

In order to get the distribution over $\mu_k \mid x, z$ for the $k$-th cluster, the Bayesian framework is applied again:

$$p(\mu_k \mid x, z) \propto p(\mu_k) \prod_{i=1}^{N} p(x_i \mid \mu_k, I)$$

(8)

Notice that the data points outside a specific cluster are irrelevant from the distribution of the specific cluster. Thus, the distribution is further proportionate to:

$$p(\mu_k \mid x, z) \propto p(\mu_k) \prod_{z_i=k} p(x_i \mid \mu_k, I) = \mathcal{N}(\frac{1}{n_k} \sum_{z_i=k} x_i, \frac{1}{n_k} \cdot I)$$

(9)

2.2.3 Gibb’s Sampling

Since the distribution of variables, $p(z_i = k \mid z_{-i})$ and $p(\mu_k \mid x, z)$, have already been derived, the next iteration of variables’ values, $z_i$ and $\mu_k$, will be sampled according to such distributions. Thus, those variables that have a higher probability density will be much easier to sample out compare to those variables with lower probability density. This process will converge after a large number of iterations. Pseudocode is presented in Algorithm 1.
Algorithm 1: Gibb’s Sampling for DPGMM

1. random Initialization for $z$, $x$ and $\mu$
2. for $l$ to Iteration
   3. for $k=1$ to $K$
      4. calculate $p(z_i = k | z_{-i}, \mu_k, x) \propto \frac{n_k}{\alpha + n - 1} \mathcal{N}(x_i | \mu_k, I)$
      5. calculate $p(\mu_k | x, z) \propto \mathcal{N}\left(\frac{1}{n_k} \sum_{i} x_i | \frac{1}{n_k} \cdot I\right)$
      6. calculate $p(z_i = K+1 | z_{-i}, x) \propto \frac{\alpha}{\alpha + n - 1} \cdot \frac{1}{2^{\frac{\text{Dim}}{2}}} \pi^{-\frac{1}{2}} e^{-\frac{1}{2} x_i^T x_i}$
      7. normalize $p(z_i | z_{-i}, x, \mu)$, $p(\mu_k | x, z)$
      8. sample new value of $z_i$ according to $p(z_i | z_{-i}, x, \mu)$
      9. sample new value of $\mu_k$ according to $p(\mu_k | x, z)$
3. end

3. Experiment and Results

300 hundred data points are generated from the following random functions:

<table>
<thead>
<tr>
<th>Groups</th>
<th>$x[1]$</th>
<th>$x[2]$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Group 1 (100 points)</strong></td>
<td>Random(0,2)</td>
<td>Random(0,2)</td>
</tr>
<tr>
<td><strong>Group 2 (100 points)</strong></td>
<td>Random(0,2) + 1.0</td>
<td>Random(0,2) + 2.3</td>
</tr>
<tr>
<td><strong>Group 3 (100 points)</strong></td>
<td>Random(0,2) + 2.5</td>
<td>Random(0,2) + 1.0</td>
</tr>
</tbody>
</table>

and here’s the graph of these points:

![Graph of generated points](image_url)

Figure 1

For all DPGMM program, $\alpha$ is set to be 0.5.
3.1 Clusters Generated for GMM

A defection for GMM is that the model will need to pre-decide the number of clusters instead of deciding by programs automatically. In this experiment, the maximum likelihood estimation algorithm was applied to Gaussian mixture models. A detail description can be found in Chapter 9 of PRML written by Bishop [2]. Here are the graphs for the clusters when setting the parameter $K$ (number of clusters) to be different values in Figure 2.

![Figure 2(a)](image1)
![Figure 2(b)](image2)
![Figure 2(c)](image3)

Figure 2(a) is for $K=1$, figure 2(b) is for $K=2$, and figure 2(c) is for $K=3$.

3.2 Data Generated from DPGMM

As mentioned before, the Gibb’s Sampling method is a stochastic method, where data are variables were generated randomly according to their probability density in their probability distributions. Thus, each time DPGMM will generate different amount of clusters and slightly different cluster distributions (cluster positions) from the stochastic Gibb’s Sampling methods and Dirichlet processes. Here are four intuitive graphs that are generated from the same program for four different times:

![Figure 3(a)](image4)
![Figure 3(b)](image5)
3.3 Number of Clusters versus Number of Iterations

Since the number of clusters is generated from a Dirichlet process using Gibb’s sampling method as mentioned before, the number of clusters will vary over time. As the program sample the latent variables and means more and more times, due to the feature of Dirichlet Process, it is more likely to generate new clusters. Here’s a graph shows the average number of clusters vary over different numbers of iterations. The graph below shows the data between the numbers of iterations and the average numbers of clusters:

For each different number of iterations, 100 times of the DPGMM were tested and the average of the number of clusters was taken. The experiment for each 100 times of DPGMM is conducted for 5, 10, 15… 500 iterations.
3.4 Distance from Each Data Point to Its Mean

A very direct way to evaluate how well a single data point is assigned to a specific cluster can be expressed by its Mahalanobis distance to the center of the cluster. Therefore, for the whole set of data points, the Mahalanobis distances were summed up in the way:

\[ D_M = \sum_{i=1}^{N} \left( x_i - \mu_i \right)^T \Sigma^{-1} \left( x_i - \mu_i \right) \]  

Here covariance \( \Sigma \) for this paper is identity matrix \( I \).

The program was run for 100 times for each different number of iterations (range from 1~100). Then the average of Mahalanobis distance for the 100 times was taken for each different number of iterations. The graph is shown below about the Mahalanobis distances versus the number of iterations:

![Iteration vs. Mahalanobis Distance to Cluster Center](image)

**Figure 5**

4. Discussion

Comparing the data results from section 3.1 and section 3.2, it is easy to observe that in 3.1, the number of clusters is prerequisite for the Gaussian Mixture Model. While the stochastic feature and Dirichlet process generated different clusters (Figure 3(b) and 3(c)) and different amounts of clusters (Figure 3(a), 3(b), and 3(c)) automatically without changing any parameter or prior knowledge about the structure of the data. The second model can achieve the same goal without changing any parameter. The clusters are simply generated by the program at different times.
In section 3.3, Figure 4, the data verified that for more and more iterations, according to Dirichlet Process, it is more and more likely to have more clusters. The number of clusters will finally converge to some point.

In section 3.4, Figure 5, the data shows the distances for all points to their cluster means are getting smaller and smaller, which means data points are fitted better and better into clusters. And the Mahalanobis distance will finally converge to 0.

5. Conclusion

In this paper, we presented a detailed derivation for Dirichlet Process Gaussian Mixture Model (DPGMM). Both the Gaussian Mixture Model (GMM) and DPGMM are implemented. The comparison between DPGMM and GMM provided an intuitive for understanding the defection of the pre-decided parameter $K$ (number of clusters). The efficiency of DPGMM was further analyzed from two approaches: number of clusters varies with time and Mahalanobis Distance for data points to their cluster mean. The first approach showed us the number of clusters will grow larger and larger due to the Dirichlet Process and will converge at some point. The second approach showed us how well the data points were fitted. It turned out that the data points are fitting better in clusters while the number of iterations was growing. The Mahalanobis distance would converge after a large number of iterations.

6. Acknowledgements

I would like to thank my mentors Professor Jun Zhu from Tsinghua University, as well as my tutor Professor Er Liu, for their advice and guidance during this entire summer. I would also like to thank Research Science Initiative, the Center for Excellence in Education, and Tsinghua University for providing me with such a wonderful opportunity and experience this summer.
References
[1] Aldous, D.J. Exchangeability and Related Topics, École d'Été de Probabilités de Saint-
2006).
the Base Distribution. Journal of Computer Science and Technology. 25(4), 615-626,
(2010).
Appendix. Derivation for Dirichlet Process Gaussian Mixture Model

For equation (4), a detail derivation is shown below:

Substitute equation (2), (3) and (7) into the equation (4) and knowing process (2) will generate out a new cluster:

\[
p(z_i = K + 1 | z_{-i}) \propto p(z_i = k | z_{-i}) \cdot \int p(x_i | \mu_{K+1}, I, z_i, z_{-i}) \cdot p(\mu_{K+1}) \, d\mu_{K+1}
\]

\[
= \frac{\alpha}{\alpha + n - 1} \cdot \frac{1}{(2\pi)^{2}} \cdot e^{-\frac{1}{2}(x_i - \mu_{K+1})^T (x_i - \mu_{K+1})} \cdot \frac{1}{(2\pi)^{\frac{D_{im}}{2}}} \cdot e^{-\frac{1}{2} \mu_{K+1}^T \mu_{K+1}} \, d\mu_{K+1} \quad (11)
\]

Further expand equation (10):

\[
p(z_i = K + 1 | z_{-i}, x_i) \propto \frac{\alpha}{\alpha + n - 1} \cdot \frac{1}{(2\pi)^{\frac{D_{im}}{2}}} \cdot e^{-\frac{1}{2} \mu_{K+1}^T \mu_{K+1}} \cdot \int \frac{1}{\pi} \cdot e^{-\frac{1}{2} x_i^T (x_i - \mu_{K+1})} \, d\mu_{K+1}
\]

Since the integral is over \( \mu_{K+1} \), some of the terms are extracted out:

\[
p(z_i = K + 1 | z_{-i}, x_i) \propto \frac{\alpha}{\alpha + n - 1} \cdot \frac{1}{2^{\frac{D_{im}}{2}}} \cdot e^{-\frac{1}{2} x_i^T x_i} \cdot \int \frac{1}{\pi} \cdot e^{-\frac{1}{2} x_i^T (x_i - \mu_{K+1})} \, d\mu_{K+1} \quad (13)
\]

By setting the covariance to be \( \frac{1}{2} I \) and mean to be \( \frac{x_i}{2} \) for a multivariate Gaussian distribution, the part inside the integral is determined to be equal to such distribution about \( \mu_{K+1} \):

\[
p(z_i = K + 1 | z_{-i}, x_i) \propto \frac{\alpha}{\alpha + n - 1} \cdot \frac{1}{2^{\frac{D_{im}}{2}}} \cdot e^{-\frac{1}{2} x_i^T x_i} \cdot \int N(\mu_{K+1} | \frac{x_i}{2}, \frac{1}{2} I) \, d\mu_{K+1} \quad (14)
\]

By recognizing the integral equals to one equation (13) is further changed into:

\[
p(z_i = K + 1 | z_{-i}, x_i) \propto \frac{\alpha}{\alpha + n - 1} \cdot \frac{1}{2^{\frac{D_{im}}{2}}} \cdot e^{-\frac{1}{2} x_i^T x_i} \quad (15)
\]