10-708: Probabilistic Graphical Models 10-708, Spring 2017

Lecture 8: The EM algorithm

Lecturer: Manuela M. Veloso, Eric P. Xing *Scribes: Huiting Liu, Yifan Yang*

1 Introduction

Previous lecture discusses how to use maximize log-likelihood (MLE) on joint probability $p(x;\theta)$, where random variables x are observed. There are lots of cases that we want to maximize $p(x,z;\theta)$, given latent variables z. Introducing latent variable will help us model the training data. For example, documents have latent variables like topics. Unfortunately, it is hard to solve MLE on $p(x,z;\theta)$ directly using methods like derivative. An approximate algorithm (e.g. EM algorithm) can be used to obtain the parameters.

The following sections organize as follow. In section 2, we give the Mixture Gaussian Problem. In section 3, we will introduce K-means algorithm, which is very popular in clustering. This algorithm is a simple case of EM algorithm. Section 3 discusses some mathematical fundamental in order to derive EM algorithm including convex, concave function and Jensen's inequality. Last Section explains the intuition and details of EM algorithm.

2 Mixture Gaussian Problem

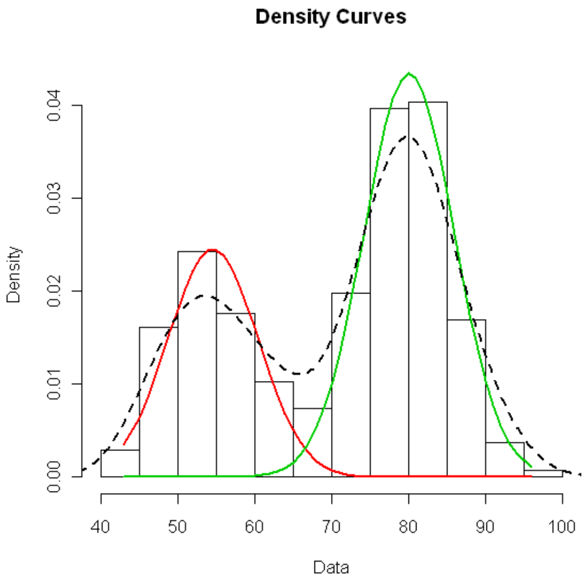

Figure 1: Mixture of Gaussian Problem [1]

Given data $(x_1, x_2, ..., x_n)$, let's suppose that the data didn't come from a single Gaussian but from a mixture of several Gaussian (two in the example figure). For every point, try to figure out the probability of which Gaussian it came from.

For this problem, MLE doesn't work. An alternative solution is to introduce latent variables Z, in which $z_{ij} = 1$ iff $x_i \in Gaussian_j$, otherwise, $z_{ij} = 0$. With these latent variables, we will be able to introduce EM algorithm to find the maximum likelihood. Next section introduces a simple version of EM, the K-means Algorithm.

3 K-means Algorithm

3.1 Description

Given a set of observations $(x_1, x_2, ..., x_n)$, where each observation is a d-dimensional real vector, K-means algorithm aims to partition the n observations into k $(k \leq n)$ sets $S = \{S_1, S_2, ..., S_k\}$ in order to minimize the sum of distances for every point toward their cluster centers. It is equivalent to minimize the following formula.

$$\sum_{i=1}^{k} \sum_{x \in S_i} \|x - u_i\|^2$$

where u_i is the mean of points in S_i.

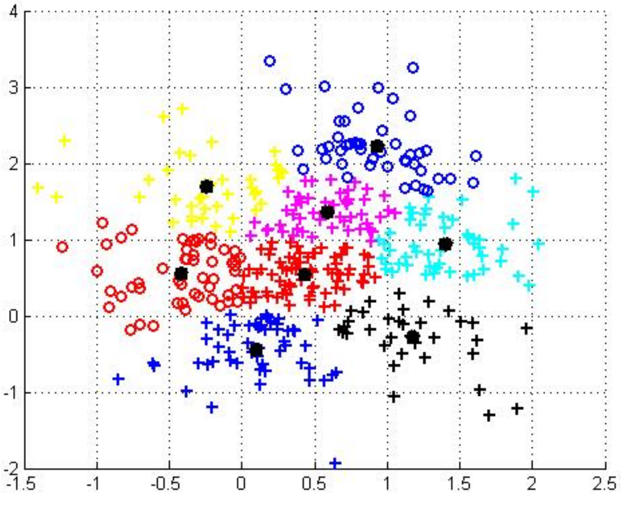

Figure 2: Clusters after using K-means. It represents different clusters with different color. [2]

[1] `https://i1.wp.com/3.bp.blogspot.com/-kbk_korLXMw/Tj2JMvEPiPI/AAAAAAAAADE/avAFubexWKk/s1600/mixtoolsFig01.png`

[2] urlhttps://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/52579/versions/9/screenshot.jpg

3.2 Algorithm Detail

1. Initialize k cluster centers randomly

$$\{u_1, u_2, ..., u_k\}$$

2. Repeat until convergence

   (a) For every point $x^{(i)}$ in the dataset, we search k cluster centers. The one, which is closest to $x^{(i)}$, will be assign as the point's new cluster center $c^{(i)}$.

$$c^{(i)} = argmin_j \left\| x^{(i)} - u_j \right\|^2$$

   After scanning all the points, each cluster will have a new set of points.

   (b) For every cluster j, calculate the new center based on average of all points belonging to this cluster.

$$u_j = \frac{\sum_{i=1}^{n} \sigma(c^{(i)=j})x^{(i)}}{\sum_{i=1}^{n} \sigma(c^{(i)=j})x^{(i)}} \tag{1}$$

$$\tag{2}$$

   Where $\sigma(x)$ is an indicator function.

It is worth noting that the two steps in K-means are actually using the idea from EM algorithm. The first step is to assign a cluster to every point, which is the E step of EM algorithm. And the second step is to update the center of each cluster, which is the M step of EM algorithm.

4 Convex (Concave) function and Jensen's inequality

The key component of EM algorithm is the use of Jensen's inequality. In the meantime, Jensen's inequality is highly connected to convex (concave) function.

4.1 Convex and Concave function

Here we give the definition of convex and concave function.

- $f(x)$ is convex, iff $f''(x) \geqslant 0$, $\forall$ x $\in R$.
- $f(x)$ is strictly convex, iff $f''(x) > 0$, $\forall$ x $\in R$.
- $f(x)$ is concave, iff $-f(X)$ is convex.

4.2 Jensen's inequality

For a random variable x, if $f(x)$ is convex, then

$$E[f(x)] >= f(E[x])$$

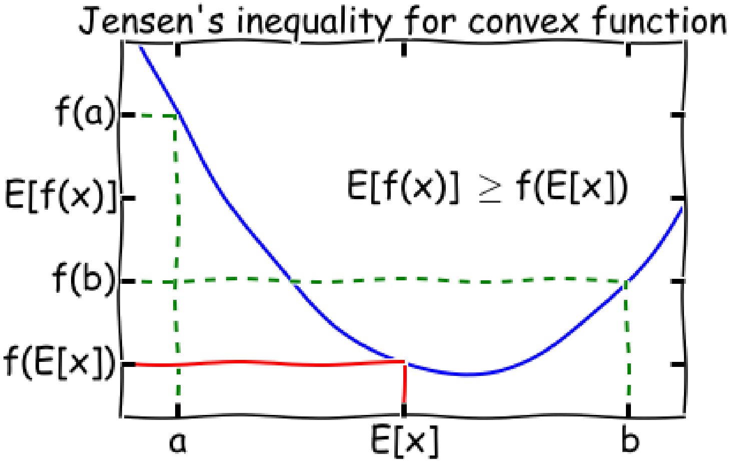

Figure 3: Jensen's inequality for convex function [3]

For a random variable x, if $f(x)$ is concave, then

$$f(E[x]) >= E[f(x)]$$

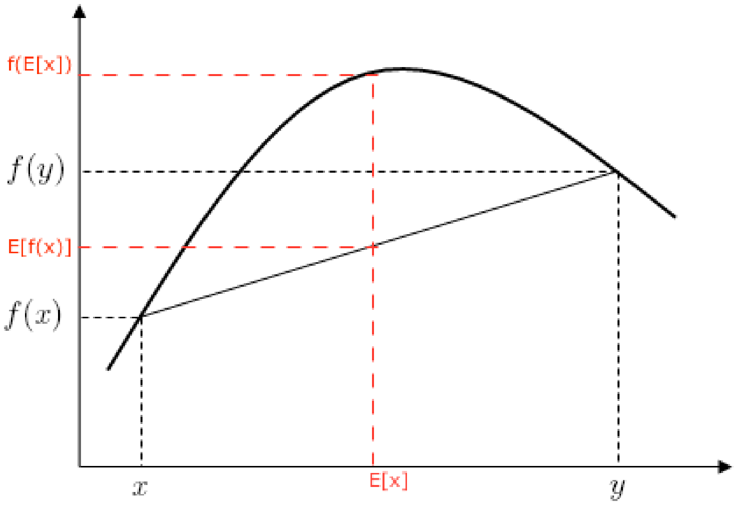

Figure 4: Jensen's inequality for concave function [4]

[3] http://people.duke.edu/~ccc14/sta-663-2016/_images/14_ExpectationMaximization_8_0.png

[4] https://en.wikipedia.org/wiki/Concave_function#/media/File:ConcaveDef.png

Using Jensen's inequality, we can derive a bound, which is extremely useful in the EM algorithm. It is also important to know when the equality holds in Jensen's inequality: $f(E[x]) = E[f(x)]$, iff x is a constant.

5 EM algorithm

Given training data set $\{x^{(1)}, ..., x^{(m)}\}$. We want to find parameters θ to fit a model $p(x, z; \theta)$, where z is latent variables. Mathematically, we want to maximize log-likelihood $l(\theta)$:

$$
\begin{aligned}
l(\theta) &= \sum_{i=1}^{m} \log p(x^{(i)}; \theta) && (3)\\
&= \sum_{i=1}^{m} \log \sum_{z^{(j)}} p(x^{(i)}, z^{(j)}; \theta) && (4)
\end{aligned}
$$

The intuition behind EM algorithm is to first create a lower bound of log-likelihood $l(\theta)$ and then push the lower bound to increase $l(\theta)$. EM algorithm is an iteration algorithm containing two steps for each iteration, called E step and M step. The following figure illustrates the process of EM algorithm. The black curve is log-likelihood $l(\theta)$ and the red curve is the corresponding lower bound. There are various of lower bound of $l(\theta)$. In E step, algorithm picks the lower bound which clings to $l(\theta)$. In M step, the lower bound is maximized. Since $l(\theta)$ is bigger than is lower bound, $l(\theta)$ will be increased.

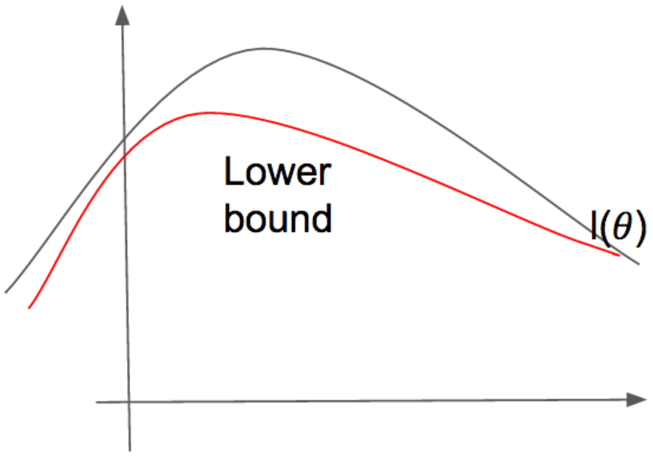

Figure 5: The E step makes the lower bound tight $l(\theta)$. The M step maximizes the lower bound to increase $l(\theta)$.

5.1 Lower Bound

The trick is that a new distribution $Q(z)$ is created. Since $Q(z)$ is a distribution, it must satisfy

$$\begin{aligned} Q(z^{(i)}) &\geq 0 && (5) \\ \sum_i Q(z^{(i)}) &= 1 && (6) \end{aligned}$$

Now we can create lower bound of $l(\theta)$ as $E_{z\sim Q}[\log \frac{p(x,z;\theta)}{Q(z)}]$

$$\begin{aligned} l(\theta) &= \sum_{i=1}^{m} \log \sum_{z^{(}j)} Q(z^{(j)})[\frac{p(x^{(i)}, z^{(j)}; \theta)}{Q(z^{(j)})}] && (7) \\ &= \sum_{i=1}^{m} \log E_{z^{(j)}\sim Q}[\frac{p(x^{(i)}, z^{(j)}; \theta)}{Q(z^{(j)})}] && (8) \end{aligned}$$

Since log is concave, using Jensen's inequality. We have

$$l(\theta) \geq \sum_{i=1}^{m} E_{z^{(j)}\sim Q}[\log \frac{p(x^{(i)}, z^{(j)}; \theta)}{Q(z^{(j)})}] \qquad (9)$$

5.2 E step

Suppose it is at iteration t+1. We want to find $Q^t(z)$, so that the lower bound equals to log-likelihood $l(\theta^t)$. According to Jensen's inequality, equality between lower bound and $l(\theta^t)$ holds if $\frac{p(x^{(i)},z^{(j)};\theta^t)}{Q^{t+1}(z^{(j)})}$ is constant. In order to make it constant, $p(x^{(i)}, z^{(j)}; \theta^t)$ have to be proportional with $Q^{t+1}(z^{(j)})$, that is

$$\begin{aligned} Q^{t+1}(z^{(j)}) &= \frac{p(x^{(i)}, z^{(j)}; \theta^t)}{\sum_z p(x^{(i)}, z; \theta^t)} && (10) \\ &= \frac{p(x^{(i)}, z^{(j)}; \theta^t)}{p(x^{(i)}; \theta^t)} && (11) \\ &= p(z^{(j)}|x^{(i)}; \theta^t) && (12) \end{aligned}$$

Therefore, we can use $p(z^{(j)}|x^{(i)}; \theta^t)$ to represent $Q^{t+1}(z^{(j)})$.

5.3 M step

In previous E step, lower bound is equal to log-likelihood by setting $Q^{t+1}(z^{(j)})$ to $p(z^{(j)}|x^{(i)}; \theta^t)$. Next, we are going to find parameters θ^{t+1} to maximize $l(\theta^{t+1})$.

$$\theta^{t+1} := arg \max_{\theta} \sum_{i=1}^{m} E_{z^{(j)}\sim Q^{t+1}}[\log \frac{p(x^{(i)}, z^{(j)}; \theta)}{Q^{t+1}(z^{(j)})}] \qquad (13)$$

5.4 Monotonicity

During the iteration between E step and M step, we want to know whether log-likelihood is increase. Mathematically, we want to prove $l(\theta^{t+1}) \geq l(\theta^t)$.

$$\begin{aligned} l(\theta^{t+1}) &= \max_\theta \sum_{i=1}^m E_{z^{(j)}\sim Q^{t+1}}[\log \frac{p(x^{(i)}, z^{(j)}; \theta)}{Q^{t+1}(z^{(j)})}] &(14)\\ &\geq \sum_{i=1}^m E_{z^{(j)}\sim Q^{t+1}}[\log \frac{p(x^{(i)}, z^{(j)}; \theta^t)}{Q^{t+1}(z^{(j)})}] &(15)\\ &\geq \sum_{i=1}^m E_{z^{(j)}\sim Q^{t}}[\log \frac{p(x^{(i)}, z^{(j)}; \theta^t)}{Q^{t}(z^{(j)})}] &(16)\\ &= l(\theta^t) &(17)\end{aligned}$$

As a conclusion, we can see that EM algorithm will make the log-likelihood get bigger and bigger as the iterations go on.

In the next lecture, we will talk about how to apply EM to solve the Mixture Gaussian Problem and introduce the HMMs and CRFs.