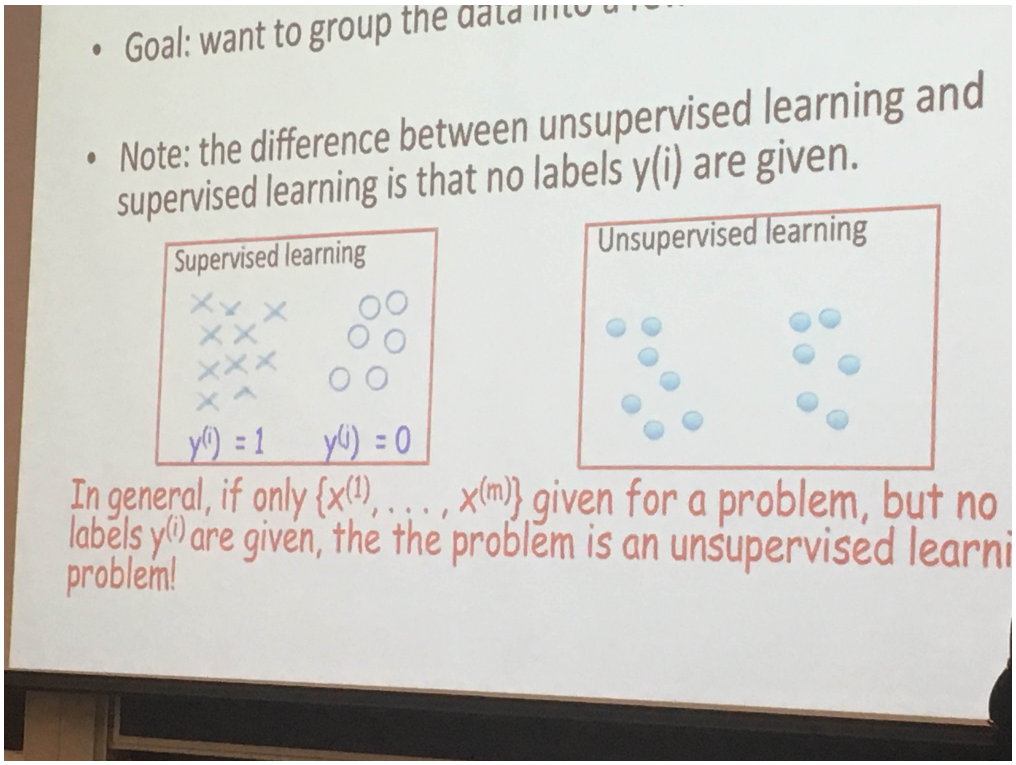
Mathematics of Big Data I

Lecture 7: Unsupervised Learning, K-Means Clustering, Gaussian Mixture, Jensen's Inequality

October 24th, 2016

1 K-Means Clustering Algorithm

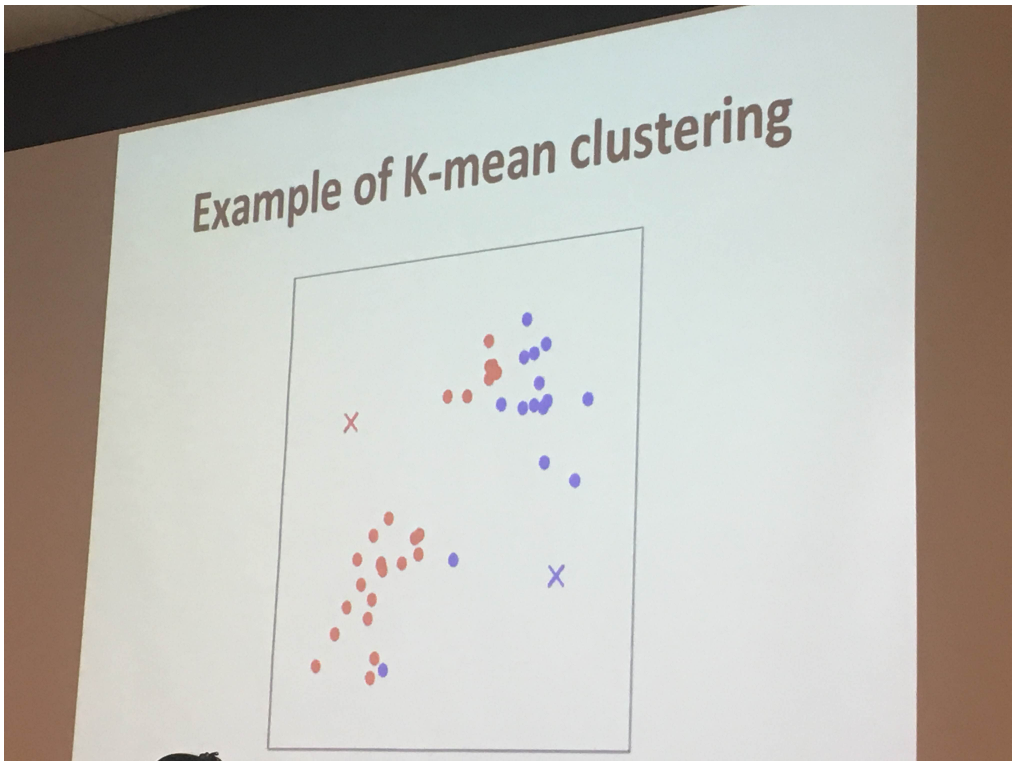
What is a clustering problem? Given a training set $\left\{x^{(1)}, \ldots, x^{(m)}\right\}$ where each $x^{(i)} \in \mathbb{R}^n$ we want to group the data into cohesive "clusters." The difference between supervised learning and unsupervised learning is that we are not given any labels $y^{(i)}$ that we are trying to fit the training data to.



The clustering algorithm is as follows:

1. Initialize the cluster centroids $\mu_1, \ldots, \mu_k \in \mathbb{R}^n$.
2. Repeat these steps until convergence: for every i, set $c^{(i)} = \arg\min_j \left\|x^{(i)} - \mu_j\right\|^2$. Then for each j set

$$\mu_j = \frac{\sum_{i=1}^m \mathbb{1}_{\{c^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m \mathbb{1}_{\{c^{(i)}=j\}}}$$



A reasonable question to ask is whether this algorithm always converges? The answer is a little nuanced: we are always guaranteed convergence of the algorithm, but the point of convergence may not in fact be the global minimum. We can define the **distortion function**

$$J(c, \mu) = \sum_{i=1}^{m} \left\| x^{(i)} - \mu_{c^{(i)}} \right\|^2.$$

J measures the sum of squared distances between each training sample and each centroid. K-means clustering is exactly the same as gradient descent with respect to the function J. We have that we can minimize the value of J in a monotonic fashion, but we are not guaranteed that we find the global minimum of J. Regardless of this fact however, k-means clustering works very well. If in practice you get stuck in a bad local minimum, you can try a heuristic method (try a different initialization of the means, for example) to arrive at a different solution. You can try many different initializations for k-means clustering and take the minimum of all the trials.

2 Gaussian Mixture Models

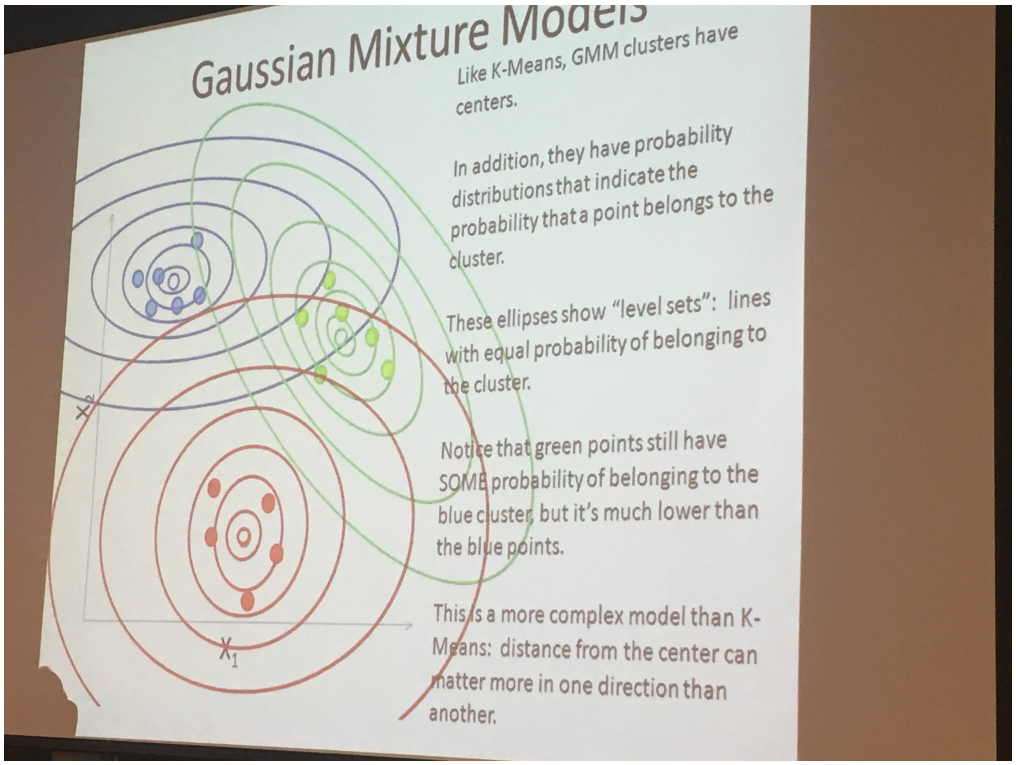
A **Gaussian mixture model** (GMM) is simply a probability distribution that can be written as a linear combination of normal random variables. For instance

$$p(\theta) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\mu_i, \Sigma_i)$$

where each $\phi_j \geq 0$ and

$$\sum_{i=1}^{K} \phi_i = 1.$$

GMMs have different centers, and so they are similar to k-means clusters, however they differ in some key ways.



The ellipses in the photo above indicate probabilities that certain points belong to which cluster. This is more complicated than k-means clustering since we need to consider the direction from a training point to a center, since the direction will change the probability that a point belongs to that Gaussian. What makes this problem hard? The posterior distribution $p(\theta|x)$ is also a Gaussian mixture model. Note that mixture models don't have to be limited to Gaussians as a template. We can take mixtures of any model that we please (Poisson, uniforms, etc.).

3 The Expectation-Maximization (EM) Algorithm

We will carefully work through the details of this problem. We ask, expectation of what? Maximization of what? It's important to note that the EM-algorithm works almost exclusively for exponential families. This is fine however since most of the distributions we will be looking at in this class will belong to the exponential family.

Recall that we learned about maximum likelihood estimation (MLE) for an optimization problem, or maximize a posterior (MAP) for parameter estimation. Now, let's consider the problem where we have some data missing (unobserved data). One example of this is a hidden Markov chain. What "hidden" means in this context is that we only know the marginal distribution for some random variable. For an exponential family, we have that the EM-algorithm works particularly well.

Given we have data $x = \left\{x^{(1)}, \ldots, x^{(m)}\right\}$ where each $x^{(i)} \in \mathbb{R}^d$. In our model, let z be the hidden variable, and we can write $(x, z) \sim P_\theta$ where typically we will take P_θ to be an exponential family, for some unknown $\theta \in \Theta$, where Θ is some space of parameters. Our goal is to find

$$\theta_{MLE} = \arg\max_\theta P_\theta(x).$$

The issue here is that we have

$$P_\theta(x) = \sum_z P_\theta(x, z)$$

hence this is difficult to maximize due to this sum above. We will have maximums associated to this problem, but the issue is that we cannot solve this problem analytically. The key idea here is that we are going to solve this problem iteratively. In general we will do the following steps:

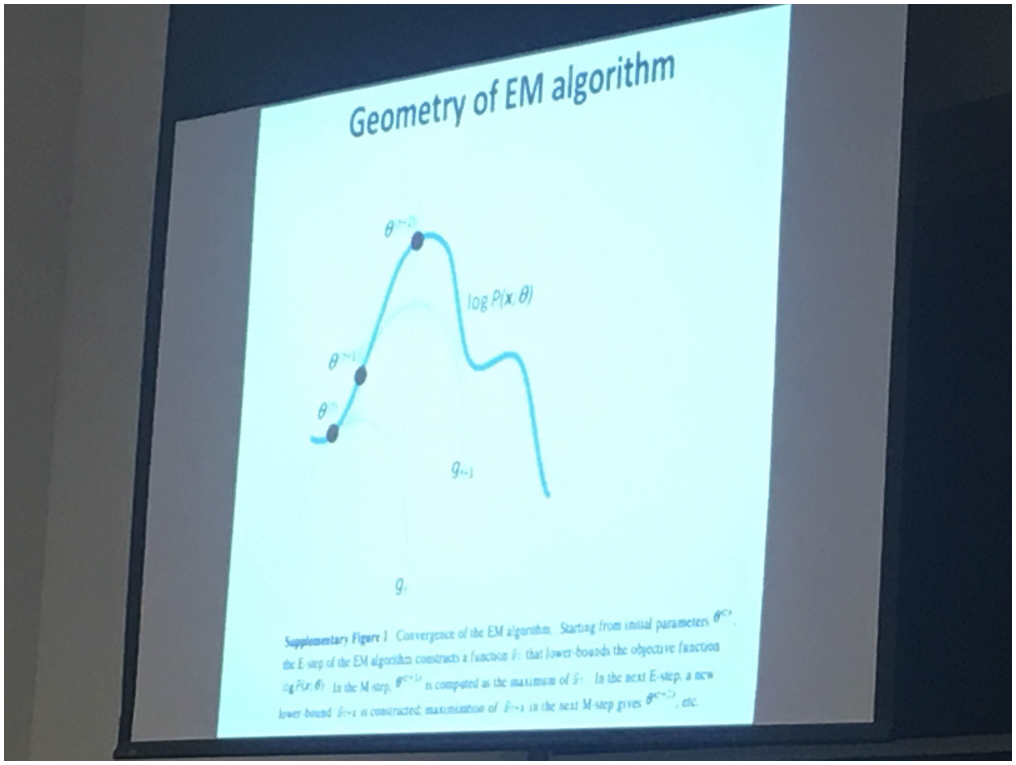
1. Initialize $\theta_0 \in \Theta$.
2. For $t = 0, 1, 2, \ldots$ then at step t, we will perform the expectation (call this the E-step). We find

$$Q(\theta, \theta_t) \triangleq E_{\theta_t}\left[\log P_\theta(X, Z) \mid X = x\right].$$

3. Perform the M-step: choose

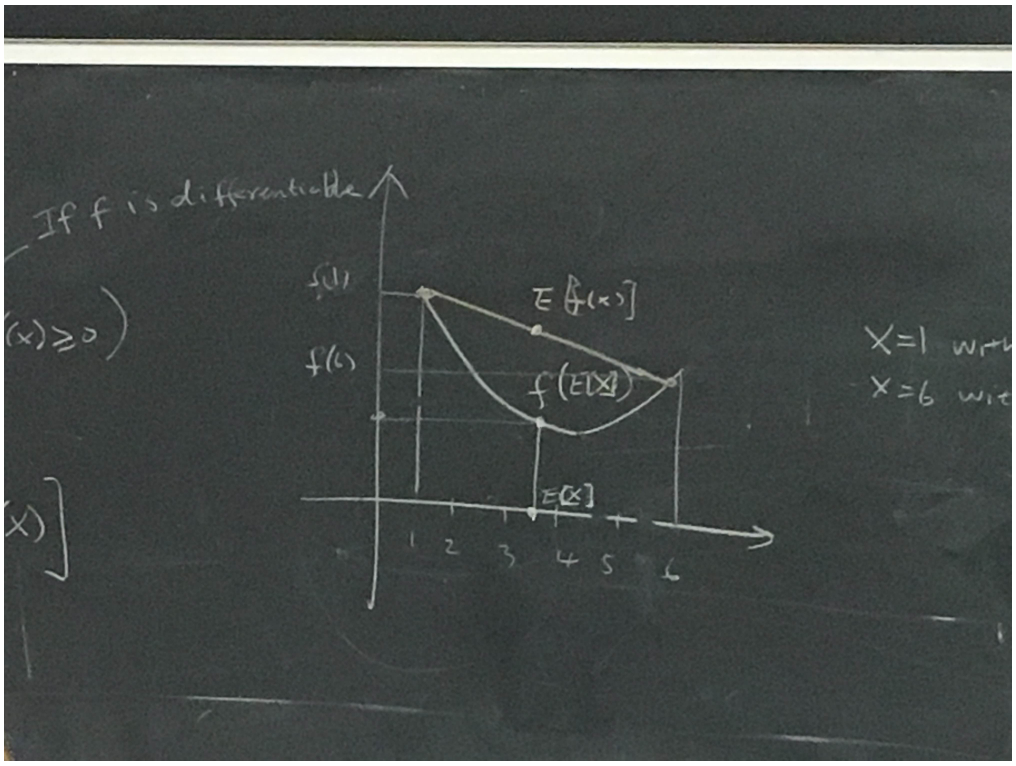
$$\theta_{t+1} = \arg\max_\theta Q(\theta, \theta_t).$$

What's the geometric intuition behind this problem? We will first do this for the Gaussian distribution, and later we will adapt it for an exponential family for the general case.



3.1 Jensen's Inequality

Theorem 1 *Let f be a convex function and X a random variable. Then $E[f(X)] \geq f(E[X])$. Moreover if f is strictly convex then equality holds true if and only if $X = E[X]$ with probability 1 (i.e. X is constant).*



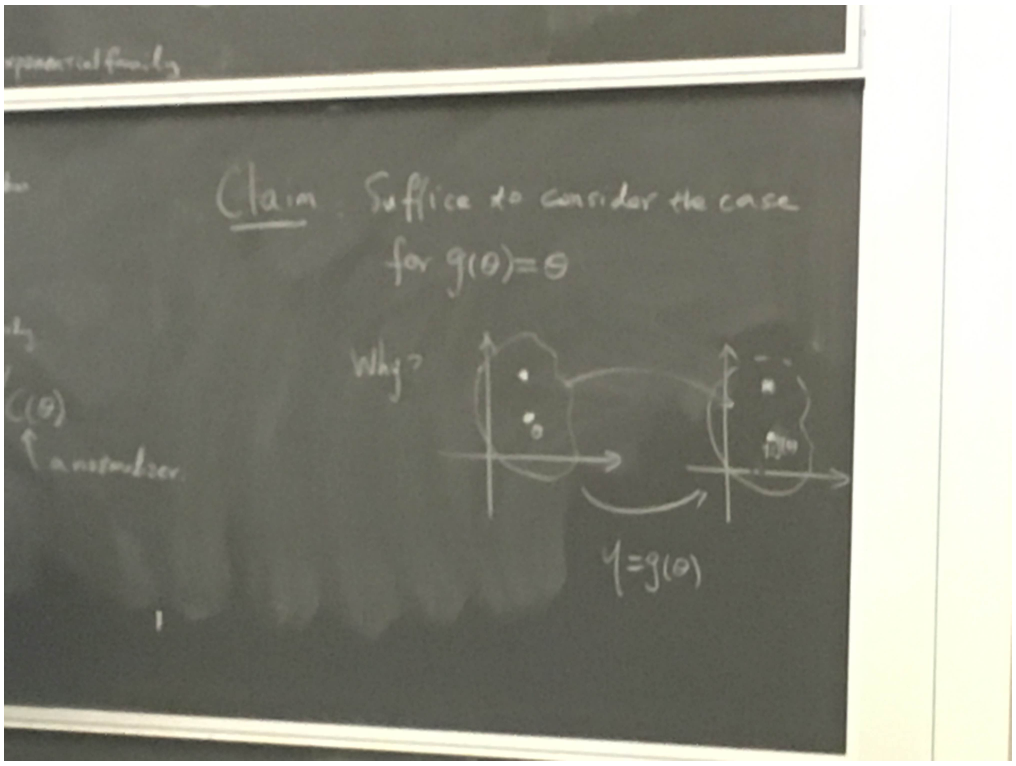
If for instance we have $f(x) = \log x$ we have $f'(x) = \frac{1}{x}$ and $f''(x) = -\frac{1}{x^2} < 0$, where since the second derivative is always strictly negative then $f(x)$ is a concave function. Jensen's inequality yields an analagous statement for the concave, namely that $f(E[X]) \leq E[f(X)]$ with all of the remaining statements holding just the same as in the original theorem.

3.2 Motivation for the EM-Algorithm

We will specialize this to the case where we are dealing with an exponential family, hence we can write the density functions as

$$P_\theta(x, z) \quad = \quad \frac{h(x, z)}{c(\theta)} e^{g(\theta)^T s(x,z)}.$$

We claim that it suffices to consider the case where $g(\theta) = \theta$



To maximize this function is equivalent to maximizing the logarithm of this function, hence we can take

$$\log p_\theta(x,z) \;=\; \theta^T s(x,z) + \log h(x,z) - \log c(\theta).$$

Taking the derivative and setting it equal to zero we take

$$0 \;=\; \partial_i \log p_\theta(x,z) \;=\; \frac{1}{p_\theta(x,z)} \sum_i \frac{\partial}{\partial \theta_i} p_\theta(x,z).$$

we have that

$$\frac{\partial}{\partial \theta_i} p_\theta(x,z) \;=\; \underbrace{\left(s_i(x,z) - \frac{\partial}{\partial \theta_i} \log c(\theta)\right)}_{=E_\theta[s(x,z)]} h(x,z).$$

Therefore we obtain

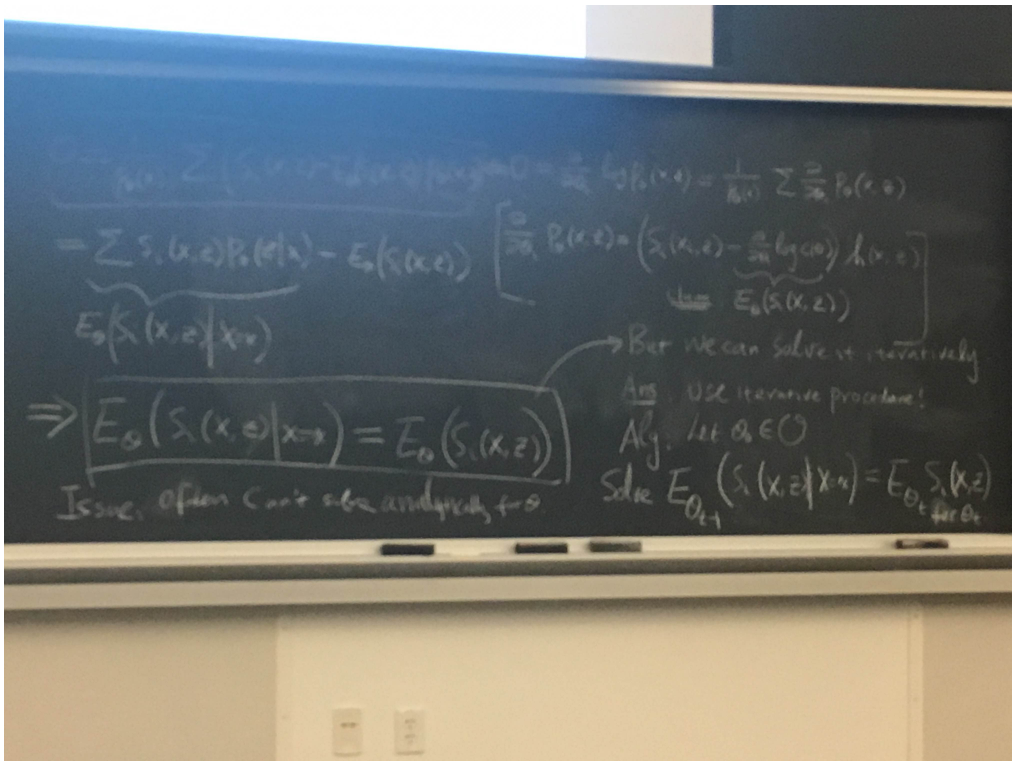
$$\begin{aligned} 0 \;&=\; \frac{1}{p_\theta(x,z)} \sum_i \left(s_i(x,z) - E_\theta[s_i(x,z)]\right) p_\theta(x,z) \\ &=\; \underbrace{\sum_i s_i(x,z) p_\theta(z|x)}_{E_\theta[s_i(X,Z)|X=x]} - E_\theta[s_i(x,z)] \end{aligned}$$

where we have $E_\theta[s_i(X,Z)|X=x] = E_\theta[s_i(X,Z)]$. While we can't solve this analytically for θ, we can solve it iteratively. We then follow this algorithm:

1. Let $\theta_0 \in \Theta$ be a chosen parameter.
2. Now solve for

$$E_{\theta_{t-1}}\left[s_i(X,Z) \mid X = x\right] \;=\; E_{\theta_t}[s_i(X,Z)]$$

for θ_t.



In the standard EM-algorithm, we have the quantity

$$Q(\theta,\theta_t) \;=\; E_{\theta_t}\left[\log p_\theta(X,Z) \mid X = x\right].$$

For an exponential family we have $p_\theta(x,z) = e^{\theta^T s(x,z) - \log c(\theta)} h(x,z)$. We take the conditional expectation

$$E_{\theta_0}\left[\log p_\theta(X,Z) \mid X = x\right] \;=\; \theta^T E_{\theta_0}\left[s(X,Z) \mid X = x\right] - \log c(\theta) + const.$$

and in taking the partial derivative we have

$$0 \;=\; \frac{\partial}{\partial\theta} Q \;=\; E\left[s_i(X,Z) \mid X = x\right] - \frac{\partial}{\partial\theta}\log c(\theta)$$

hence we conclude that $E_{\theta_0}\left[s_I(X,Z) \mid X = x\right] = E_\theta[s_i(X,Z)]$.

Example: If we have a Gaussian mixture model, we have

$$p\left(x^{(i)}, z^{(i)}\right) \;=\; p\left(x^{(i)}|z^{(i)}\right) p(z^{(i)}).$$

where the $z^{(i)}$ are multinomial random variables with respect to the parameter ϕ. Then

$$x^{(i)}|z^{(i)} = j \;\sim\; \mathcal{N}\left(\mu_j, \Sigma_j\right).$$

If we knew the $z^{(i)}$ then we can use MLE with our likelihood function given by

$$
\begin{aligned}
l(\phi, \mu, \Sigma) &= \sum_{i=1}^{m} \log p\left(x^{(i)}, z^{(i)}; \phi, \mu, \Sigma\right) \\
\phi_j &= \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{\{z^{(i)}=j\}} \\
\mu_j &= \frac{\sum_{i=1}^{m} \mathbb{1}_{\{z^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^{m} \mathbb{1}_{\{z^{(i)}=j\}}}.
\end{aligned}
$$

We repeat these steps where in the E-step we guess the value of the $z^{(i)}$'s. We let

$$
\begin{aligned}
W_j^{(i)} &= \frac{p\left(x^{(i)} \mid z^{(i)} = j\right) p\left(z^{(i)} = j\right)}{\sum_l p\left(x^{(i)} \mid z^{(i)} = l\right) p\left(z^{(i)} = l\right)} \\
&= \frac{\phi_j}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} \exp\left[\left(x^{(i)} - \mu_j\right)^T \Sigma_i^{-1} \left(x^{(i)} - \mu_j\right)\right].
\end{aligned}
$$

The M-step is then to take the following:

$$
\begin{aligned}
\phi_j &\coloneqq \frac{1}{m} \sum_{i=1}^{m} W_j^{(i)} \\
\mu_j &\coloneqq \frac{\sum_{i=1}^{m} W_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} W_j^{(i)}} \\
\Sigma_j &\coloneqq \frac{\sum_{i=1}^{m} W^{(i)} \left(x^{(i)} - \mu_j\right) \left(x^{(i)} - \mu_j\right)^T}{\sum_{i=1}^{m} W_j^{(i)}}.
\end{aligned}
$$

There are different pros and cons of using the EM algorithm.

- **Pros**: First we get that $p_{\theta_{t+1}} \geq p_{\theta_t}$ and that EM works well in practice.
- **Cons**: We are not guaranteed to achieve MLE when we use this algorithm. EM can get stuck on a local maximum as opposed to achieving the global maximum. Convergence can be quite slow, and this type of algorithm is really specialized for exponential families, but not many others.

Some common applications of the EM algorithm are for density estimation and anomaly detection.