COS 513: MIXTURE MODELS AND THE EM ALGORITHM
LECTURE 14

JOSÉ FERNANDES AND JESÚS PUENTE

MIXTURE MODELS

Mixture models are a type of latent variable models. They can be used for expressing complicated densities that cannot be described by an exponential family distribution, or to cluster data points.

Beware that fitting a latent variable model will always find structure, whether there is it or not. The decision on the number of clusters is of the realm of model selection, is problem dependent and requires external validation criteria.

There is a connection between mixture models and kernel density estimation. The latter can be seen as a mixture model, with as many distributions as data points. In fact kernel density estimation can be a good starting method to have an idea of the shape of the distribution of the data.

As an example, contemplate the data points $x_n,\ n = 1, \ldots, N$ in figure 1

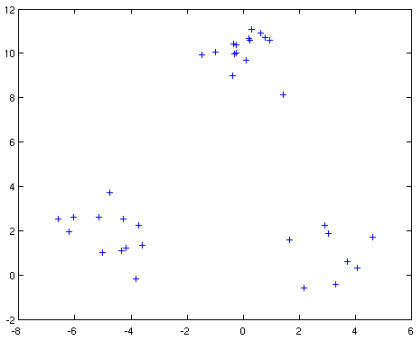

FIGURE 1. Mixture of three Gaussians

We can assume them to belong to one of the K different Gaussian distributions with means $\mu_{1:K}$ and the same covariance matrix. Then we can model our sample with a probabilistic generative process using the graphical model depicted in figure 2:

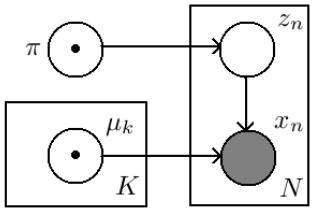

FIGURE 2. Mixture GM

The variable z_n is a multinomial latent variable which determines the mean x_n is centered around. z_n has parameter $\pi = (\pi_1, \ldots, \pi_K)$ (the mixture proportions) where $\pi_i, i = 1, \ldots, K$ is the probability of drawing x_n from the i-th cluster. In our notation z_n is an indicator vector, of length K, where $z_n^i = 1$ if x_n belongs to the i-th cluster and 0 otherwise.

Given this graphical model the joint probability distribution of the sample factorizes as

$$p(x_{1:N}, z_{1:N}|\pi, \mu_{1:K}) = \prod_{n=1}^{N} p(z_n|\pi)p(x_n|z_n, \mu_{1:K}).$$

In our interest to do ML estimation on the parameters of the model, we need to obtain the marginal probability of the data given the parameters. To do so it is necessary to marginalize out unobserved variables, which in our case are the cluster assignments:

$$p(x_{1:N}|\pi, \mu_{1:K}) = \prod_{n=1}^{N} \sum_{z} p(z_n|\pi)p(x_n|z_n, \mu_{1:K}).$$

The log likelihood is given by:

$$\begin{aligned} l(\pi, \mu_{1:K}|x_{1:N}) &= \sum_n \log \sum_z p(z_n|\pi)p(x_n|z_n, \mu_{1:K}) \\ &= \sum_n \log \left[\sum_z \left(\prod_i \pi_i^{z_n^i} \right) \left(\prod_i p(x_n|\mu_i)^{z_n^i} \right) \right] \end{aligned}$$

Notice that because we had to marginalize out latent variables the sum is inside the logarithm, which makes the maximization problem cumbersome. We could simply use a black-box algorithm for optimization. However, we can exploit characteristics of the log likelihood, and we do so by deriving the expectation-maximization algorithm.

The Expectation-Maximization (EM) Algorithm. The EM algorithm is a general purpose strategy for finding MLE's in latent variable models, many

algorithms are just instations of it. It is usually credited to Dempster et al. (1977).

Assume we can observe the latent variables. Then we can work with the complete log likelihood and we could write

$$\begin{aligned}\log p(x_{1:N}, z_{1:N}|\mu_{1:K}, \pi) &= \sum_{n=1}^{N}\left(\log\prod_i \pi_i^{z_n^i} + \log\prod_i p(x|\mu_i)^{z_n^i}\right)\\ &= \sum_{n=1}^{N}\left(\sum_{i=1}^{k} z_n^i \log\pi_i + \sum_{i=1}^{k} z_n^i \log p(x_n|\mu_i)\right)\end{aligned}$$

and we could find

$$\hat{\pi} = \sum_{n=1}^{N}\frac{z_n}{N}$$

as the ML estimator for the mixture proportions (note that this is a vector and so we drop the superscript i).

Then we could also derive

$$\hat{\mu}_i = \frac{\sum_n z_n^i x_n}{\sum_n z_n^i}$$

In reality though, we cannot observe z_n. To circumvent this, in EM we start with some idea of where the $\mu_{1:K}$ are and replace the grouping z_n^i with the expected posterior grouping $\mathbb{E}\left[z_n^i|x_n, \mu_{1:K}, \pi\right]$. We then reestimate where the groups are based on the expected posterior groupings, and we iterate between one step and the other.

Therefore, we have

- **E-step**: Replace z_n^i with

$$\begin{aligned}\mathbb{E}\left[z_n^i|x_n, \mu_{1:K}, \pi\right] &= p(z_n^i = 1|x_n, \mu_{1:K}, \pi)\\ &\propto p(z_n^i = 1|\pi)\,p(x_n|\mu_i)\\ &\propto \pi_i\, p(x_n|\mu_i)\end{aligned}$$

- **M-step**: Calculate the new MLEs from the new z_n^i's.

This reminds us of the K-means algorithm, where we start picking K initial means at random and assign our sample points to the closest mean. We then re-estimate the means of the groups, and iterate between these two steps. The EM algorithm differs from this in that it makes "soft assignments" of data points. In other words, it does not calculate which mean each point corresponds to, but the probability that the point corresponds to each of the different means.

There is a problem with EM: since the problem is not convex, it will find a local maximum of log likelihood but we are not guaranteed to find a global maximum. Thus, it is advisable to run the algorithm starting in different places and then compare the solutions to see which one is the best.

THE GENERAL EM ALGORITHM

The EM algorithm can be used in more diverse settings than the one described in the previous section. It is a versatile algorithm that we use when our graphical models have latent variables.

As before, it has two parts: the E-step and the M-step. The E-step fills in the values of latent variables via the posteriors and the M-step fits the parameters to match the filled in values through ML estimation. We iterate both steps until our likelihood stabilizes.

Call the sample data $x_{1:N}$, the hidden variables $z_{1:N}$, and the parameters θ. Then if z_n were observed, we could split the log likelihood maximization problem as

$$\hat{\theta} = \arg\max_{\theta} \log p(z|\theta) + \log p(x|z,\theta)$$

but since z_n are not observed in reality, we have

$$\hat{\theta} = \arg\max_{\theta} \log \sum_{z} p(x,z|\theta)$$

which is usually hard to handle.

We now recall that if $\lambda \in (0,1)$ and ϕ is a concave function, then Jensen's inequality tells us

$$\lambda\phi(x) + (1-\lambda)\phi(y) \leq \phi(\lambda x + (1-\lambda)y)$$

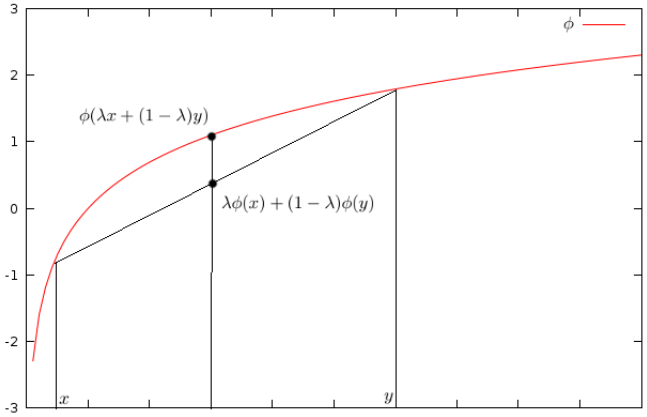

FIGURE 3. Jensen's inequality

or, in general, that $\mathbb{E}[\phi(X)] \leq \phi(\mathbb{E}[X])$. With the help from this inequality, we will bound the marginal likelihood (which is hard to evaluate) with the expected complete likelihood (which we can compute). Let $l(\theta|x) = \log p(x|\theta)$, then

$$\begin{aligned} l(\theta|x) &= \log \sum_z p(x, z|\theta) \\ &= \log \sum_z p(x, z|\theta) \frac{q(z)}{q(z)} \\ &= \log \mathbb{E}_q \left[\frac{p(x, z|\theta)}{q(z)} \right] \\ &\geq \mathbb{E}_q \left[\log p(x, z|\theta) \right] - \mathbb{E}_q \left[\log q(z) \right] \\ &=: \mathcal{L}(q, \theta) \end{aligned}$$

where we call $\mathcal{L}$ the EM objective function.

The M-step will then consist of finding

$$\theta^{(t+1)} = \arg\max_\theta \mathcal{L}(q^{(t+1)}, \theta)$$

and the E-step will consist of optimizing

$$q^{(t+1)} = \arg\max_q \mathcal{L}(q, \theta^{(t)}) = p(z|x, \theta^{(t)}).$$

Notice the last equality in the previous equation, it is true because the maximum is achieved when $q = p(z|x, \theta^{(t)})$ which means that the bound is tight. The end result of this is

$$\begin{aligned} l(\theta^{(t+1)}|x) &= \mathcal{L}(q^{(t+1)}, \theta^{(t+1)}) \\ &\geq \mathcal{L}(q^{(t)}, \theta^{(t+1)}) \\ &\geq \mathcal{L}(q^{(t)}, \theta^{(t)}) = l(\theta^{(t)}|x) \end{aligned}$$

Therefore at every step of the EM algorithm we are climbing uphill the log likelihood.