

# CS229 Lecture notes

Andrew Ng

## Mixtures of Gaussians and the EM algorithm

In this set of notes, we discuss the EM (Expectation-Maximization) for density estimation.

Suppose that we are given a training set  $\{x^{(1)}, \dots, x^{(m)}\}$  as usual. Since we are in the unsupervised learning setting, these points do not come with any labels.

We wish to model the data by specifying a joint distribution  $p(x^{(i)}, z^{(i)}) = p(x^{(i)}|z^{(i)})p(z^{(i)})$ . Here,  $z^{(i)} \sim \text{Multinomial}(\phi)$  (where  $\phi_j \geq 0$ ,  $\sum_{j=1}^k \phi_j = 1$ , and the parameter  $\phi_j$  gives  $p(z^{(i)} = j)$ ), and  $x^{(i)}|z^{(i)} = j \sim \mathcal{N}(\mu_j, \Sigma_j)$ . We let  $k$  denote the number of values that the  $z^{(i)}$ 's can take on. Thus, our model posits that each  $x^{(i)}$  was generated by randomly choosing  $z^{(i)}$  from  $\{1, \dots, k\}$ , and then  $x^{(i)}$  was drawn from one of  $k$  Gaussians depending on  $z^{(i)}$ . This is called the **mixture of Gaussians** model. Also, note that the  $z^{(i)}$ 's are **latent** random variables, meaning that they're hidden/unobserved. This is what will make our estimation problem difficult.

The parameters of our model are thus  $\phi$ ,  $\mu$  and  $\Sigma$ . To estimate them, we can write down the likelihood of our data:

$$\begin{aligned}\ell(\phi, \mu, \Sigma) &= \sum_{i=1}^m \log p(x^{(i)}; \phi, \mu, \Sigma) \\ &= \sum_{i=1}^m \log \sum_{z^{(i)}=1}^k p(x^{(i)}|z^{(i)}; \mu, \Sigma) p(z^{(i)}; \phi).\end{aligned}$$

However, if we set to zero the derivatives of this formula with respect to the parameters and try to solve, we'll find that **it is not possible to find the maximum likelihood estimates of the parameters in closed form**. (Try this yourself at home.)

The random variables  $z^{(i)}$  indicate which of the  $k$  Gaussians each  $x^{(i)}$  had come from. Note that if we knew what the  $z^{(i)}$ 's were, the maximum

likelihood problem would have been easy. Specifically, we could then write down the likelihood as

$$\ell(\phi, \mu, \Sigma) = \sum_{i=1}^m \log p(x^{(i)} | z^{(i)}; \mu, \Sigma) + \log p(z^{(i)}; \phi).$$

Maximizing this with respect to  $\phi$ ,  $\mu$  and  $\Sigma$  gives the parameters:

$$\begin{aligned}\phi_j &= \frac{1}{m} \sum_{i=1}^m 1\{z^{(i)} = j\}, \\ \mu_j &= \frac{\sum_{i=1}^m 1\{z^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{z^{(i)} = j\}}, \\ \Sigma_j &= \frac{\sum_{i=1}^m 1\{z^{(i)} = j\} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^m 1\{z^{(i)} = j\}}.\end{aligned}$$

Indeed, we see that if the  $z^{(i)}$ 's were known, then maximum likelihood estimation becomes nearly identical to what we had when estimating the parameters of the Gaussian discriminant analysis model, except that here the  $z^{(i)}$ 's playing the role of the class labels.<sup>1</sup>

However, in our density estimation problem, the  $z^{(i)}$ 's are *not* known. What can we do?

The EM algorithm is an iterative algorithm that has two main steps. Applied to our problem, in the E-step, it tries to “guess” the values of the  $z^{(i)}$ 's. In the M-step, it updates the parameters of our model based on our guesses. Since in the M-step we are pretending that the guesses in the first part were correct, the maximization becomes easy. Here's the algorithm:

Repeat until convergence: {

(E-step) For each  $i, j$ , set

$$w_j^{(i)} := p(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma)$$

---

<sup>1</sup>There are other minor differences in the formulas here from what we'd obtained in PS1 with Gaussian discriminant analysis, first because we've generalized the  $z^{(i)}$ 's to be multinomial rather than Bernoulli, and second because here we are using a different  $\Sigma_j$  for each Gaussian.

(M-step) Update the parameters:

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

In the E-step, we calculate the posterior probability of our parameters the  $z^{(i)}$ 's, given the  $x^{(i)}$  and using the current setting of our parameters. I.e., using Bayes rule, we obtain:

$$p(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma) = \frac{p(x^{(i)} | z^{(i)} = j; \mu, \Sigma) p(z^{(i)} = j; \phi)}{\sum_{l=1}^k p(x^{(i)} | z^{(i)} = l; \mu, \Sigma) p(z^{(i)} = l; \phi)}$$

Here,  $p(x^{(i)} | z^{(i)} = j; \mu, \Sigma)$  is given by evaluating the density of a Gaussian with mean  $\mu_j$  and covariance  $\Sigma_j$  at  $x^{(i)}$ ;  $p(z^{(i)} = j; \phi)$  is given by  $\phi_j$ , and so on. The values  $w_j^{(i)}$  calculated in the E-step represent our “soft” guesses<sup>2</sup> for the values of  $z^{(i)}$ .

Also, you should contrast the updates in the M-step with the formulas we had when the  $z^{(i)}$ 's were known exactly. They are identical, except that instead of the indicator functions “ $1\{z^{(i)} = j\}$ ” indicating from which Gaussian each datapoint had come, we now instead have the  $w_j^{(i)}$ 's.

The EM-algorithm is also reminiscent of the K-means clustering algorithm, except that instead of the “hard” cluster assignments  $c(i)$ , we instead have the “soft” assignments  $w_j^{(i)}$ . Similar to K-means, it is also susceptible to local optima, so reinitializing at several different initial parameters may be a good idea.

It's clear that the EM algorithm has a very natural interpretation of repeatedly trying to guess the unknown  $z^{(i)}$ 's; but how did it come about, and can we make any guarantees about it, such as regarding its convergence? In the next set of notes, we will describe a more general view of EM, one

<sup>2</sup>The term “soft” refers to our guesses being probabilities and taking values in  $[0, 1]$ ; in contrast, a “hard” guess is one that represents a single best guess (such as taking values in  $\{0, 1\}$  or  $\{1, \dots, k\}$ ).

*reminiscent*

that will allow us to easily apply it to other estimation problems in which there are also latent variables, and which will allow us to give a convergence guarantee.