CS 181 Section: Expectation Maximization for a Bernoulli Mixture Model

Vincent Nguyen

Week of April 4, 2016

In some sections, we didn't have time to get to working out the EM algorithm in its entirety. To fully understand expectation maximization (EM), let us run through the EM algorithm for the Bernoulli mixture model. EM has two steps as outlined below

1. E-Step: Take the expectation conditioned on z for the log-likelihood. We denote this as $\mathbb{E}[z_{nk}]$ or $\gamma(z_{nk})$.
2. M-Step: Find the maximum model parameters, $\hat{\mu}_k$ and $\hat{\pi}_k$. This is also known as an update. We plug them into the step 1 when we repeat.

As a reminder, a Bernoulli model for one input example with D different features is

$$p(x|\mu) = \prod_{d=1}^{D} \mu_d^{x_d}(1-\mu_d)^{(1-x_d)}$$

Since we are dealing with mixture models, we have K different models. On top of that, if we use all our data, then we have N different examples. In addition, we introduce a latent variable z which acts as an indicator function giving a value of 1 when the training example class matches some class k and a 0 otherwise. We need to introduce this variable because otherwise, the EM algorithm would be intractable. In short, the latent variable structure makes it easy to compute the parameters.

Taking all of this into consideration, let us define the likelihood of the data for a Bernoulli mixture model as

$$\begin{aligned}
\prod_{n=1}^{N}\prod_{k=1}^{K} p(x_n, z_{nk}|\mu_k, \pi_k) &= \prod_{n=1}^{N}\prod_{k=1}^{K} \Big(\pi_k p(x_n|\mu_k)\Big)^{z_{nk}} \\
&= \prod_{n=1}^{N}\prod_{k=1}^{K} \Big(\pi_k \prod_{d=1}^{D} p(x_{nd}|\mu_{kd})\Big)^{z_{nk}} \\
&= \prod_{n=1}^{N}\prod_{k=1}^{K} \Big(\pi_k \prod_{d=1}^{D} \mu_{kd}^{x_{nd}}(1-\mu_{kd})^{(1-x_{nd})}\Big)^{z_{nk}}
\end{aligned}$$

The log-likelihood is then

$$\log p(x, z|\mu, \pi) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left(\log \pi_k + \sum_{d=1}^{D} \Big(x_{nd} \log \mu_{kd} + (1 - x_{nd}) \log(1 - \mu_{kd}) \Big) \right)$$

1 E-Step

In this step, we take the expectation of the log-likelihood with respect to the latent variable z. To think of this another way, $\mathbb{E}[z_{nk}] = p(z_{nk} = 1|x_n, \mu_k)$:

$$\begin{aligned} p(z_{nk} = 1|x_n) &= \frac{p(z_{nk} = 1)p(x_n|z_{nk} = 1)}{\sum_{k'=1}^{K} p(z_{nk'} = 1)p(x_n|z_{nk'} = 1)} && (1) \\ &= \frac{\pi_k p(x_n|\mu_k)}{\sum_{k'=1}^{K} \pi_k p(x_n|\mu_{k'})} && (2) \end{aligned}$$

Going forward we will use the function γ to say that $\mathbb{E}[z_{nk}] = p(z_{nk} = 1|x_n) = \gamma(z_{nk})$. If we are running this EM algorithm for the very first time, then we take some random guess of μ_k and π_k and use it to compute $\gamma(z_{nk})$. Normally, we'd take the optimal μ_k and π_k we get from the M-Step (of the previous iteration) and plug it in.

2 M-Step

This step is identical to maximum likelihood estimation (MLE). For this model, we want to find the optimal μ_{kd} and π_k. These are the parameters whose values will maximize our likelihood. As the $\log$ is a monotonically increasing function, it's easier to deal in $\log$ space to maximize our likelilhood. Remember, MLE involves taking the log-likelihood (we've already done that), taking the derivative with respect to one of the variables, setting to 0, and isolating that variable. Also, we incorporate the γ function since we are dealing with the expected log-likelihood. The likelihood function above does not yet consider γ.

Let's start with the prior, $\hat{\pi}_k$. We'll have to use a Lagrange multiplier. This involves using some constraint. We use $\sum_{k=1}^{K} \pi_k = 1$ There are two additive halves to this log-likelihood. We can effectively ignore the second half since there are no π_ks there.

$$\begin{aligned} \frac{\partial \mathbb{E}_z[\log p(x, z, \lambda|\mu_k, \pi_k)]}{\partial \pi_k} &= \frac{\partial}{\partial \pi_k} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \log \pi_k + \lambda \left(1 - \sum_{k=1}^{K} \pi_k \right) && (3) \\ &= \sum_{n=1}^{N} \frac{\gamma(z_{nk})}{\pi_k} - \lambda = 0 && (4) \\ \pi_k &= \sum_{n=1}^{N} \frac{\gamma(z_{nk})}{\lambda} && (5) \end{aligned}$$

$$\sum_{k=1}^{K} \pi_k = \sum_{k=1}^{K} \sum_{n=1}^{N} \frac{\gamma(z_{nk})}{\lambda} = 1 \tag{6}$$

$$\lambda = \sum_{k=1}^{K} \sum_{n=1}^{N} \gamma(z_{nk}) = N \tag{7}$$

$$\frac{\partial \mathbb{E}_z[\ldots]}{\partial \pi_k} = \sum_{n=1}^{N} \gamma(z_{nk}) \left(\frac{1}{\pi_k} - N \right) = 0 \tag{8}$$

$$\hat{\pi}_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})}{N} \tag{9}$$

The optimal μ_{kd} is a bit more straightforward since Lagrange constraints aren't needed. Like before, we'll just ignore the first additive half because there are no μ_{kd}s.

$$\frac{\partial \mathbb{E}_z[\ldots]}{\partial \mu_{kd}} = \frac{\partial}{\partial \mu_{kd}} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \sum_{d=1}^{D} \Big(x_{nd} \log \mu_{kd} + (1 - x_{nd}) \log(1 - \mu_{kd}) \Big) \tag{10}$$

$$= \sum_{n=1}^{N} \gamma(z_{nk}) \left(\frac{x_{nd}}{\mu_{kd}} + \frac{x_{nd} - 1}{1 - \mu_{kd}} \right) \tag{11}$$

$$= \sum_{n=1}^{N} \gamma(z_{nk}) \left(\frac{x_{nd} - x_{nd}\mu_{kd} + \mu_{kd}x_{nd} - \mu_{kd}}{\mu_{kd}(1 - \mu_{kd})} \right) \tag{12}$$

$$= \sum_{n=1}^{N} \gamma(z_{nk}) \left(x_{nd} - \mu_{kd} \right) = 0 \tag{13}$$

$$\hat{\mu}_{kd} = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_{nd}}{\sum_{n=1}^{N} \gamma(z_{nk})} \tag{14}$$

The denominator can be interpreted as our expectation of how many counts of class k examples there are. Remember, this is an unsupervised learning algorithm so we don't actually know which data point belongs to which class.