EM ALGORITHM FOR A GAUSSIAN MIXTURE MODEL

REPRESENTATION OF THE MIXTURE MODEL

OBSERVED DATA

Our observed data contains n examples, each of which has l continuous-valued features. This can be stored in a matrix of size $n \times l$. $l_{i,min}$ represents the minimum value of feature i over all data points; $l_{i,max}$ has a similar definition. We assume that all features are independent.

CLUSTERS

At the beginning of the algorithm, we must specify the number of clusters we wish to generate, k. Each cluster c_i contains the following parameters:

- A cluster prior, $P(c_i)$, such that $0 \leq P(c_i) \leq 1$, which estimates the probability that a given data point was generated by that cluster.
- A list of means of length l, such that $l_{j,min} \leq \mu_{i,j} \leq l_{j,max}$.
- A list of variances of length l, each of which is $\sigma_{i,j}^2$.

You can see that for each cluster, there are actually l Gaussian distributions, each with its own mean and variance, one for each feature. We can make the assumption that these distributions are independent because we know each feature is independent.

OTHER PERSISTENT VARIABLES

To aid in the calculations we will be forming during the EM algorithm, we should keep track of the following:

- The log likelihoods of each cluster given each example. This is a matrix of size $n \times k$. Each value (i, j) in this matrix represents $\log P(c_j|x_i)$, the log of the probability that data point x_i was generated by cluster c_j.

INITIALIZATION OF CLUSTERS

To begin the EM algorithm, we must first initialize our clusters. We must keep in mind that clusters cannot start out as equivalent because if they do, they will end up fitting to the data the same way. Initialization of parameters is done in the following way for each cluster c_i:

- The cluster prior, $P(c_i) = \frac{1}{k}$. This ensures that $\sum_{i=0}^{k} P(c_i) = 1$ and each cluster's prior is equal (at least at the beginning of the algorithm).
- For each mean, $\mu_{i,j} = Uniform(l_{j,min}, l_{j,max})$. In other words, for each feature's mean, pick a value at random within that feature's range.
- For each variance, $\sigma_{i,j}^2 = g(l_{j,max} - l_{j,min})$ and $0 < g < 1$. In other words, each feature's variance should be a fixed fraction of the feature's range. g should be constant over all features and clusters.

EXPECTATION STEP

This step calculates and updates the log likelihoods of each cluster given each example, $\log P(c_j|x_i)$. Again, each of these represents the probability that example x_i was generated by cluster c_j. We know that, from conditional probability,

$$P(c_j|x_i) = \frac{P(c_j)P(x_i|c_j)}{P(x_i)}$$

The following are defined as:

- $P(c_j)$ is cluster c_j's prior. We already know this.
- $P(x_i|c_j)$ is the Gaussian probability of x_i being in cluster c_j.
- $P(x_i)$ is the probability of x_i occurring at all in the model.

First, since we already know $P(c_j)$, let us look at $P(x_i|c_j)$. We cannot simply calculate this because each example x_i consists of l features, each described by c_j's Gaussian distribution for that feature. We know that

$$P(x_i|c_j) = \prod_{a=0}^{l} \frac{e^{-\frac{(x_{i,a}-\mu_{j,a})^2}{2\sigma_{j,a}^2}}}{\sqrt{2\pi\sigma_{j,a}^2}} = \prod_{a=0}^{l} Gauss(x_{i,a}|c_{j,a})$$

Where $\mu_{j,a}$ is the mean for c_j, feature a; $\sigma_{j,a}^2$ is the variance for c_j, feature a.

We can also look at $P(x_i)$.

$$P(x_i) = \sum_{b=0}^{k} P(c_b)P(x_i|c_b)$$

Again, consider that $P(x_i|c_b) = \prod_{a=0}^{l} Gauss(x_{i,a}|c_{b,a})$.

So we can now write $P(c_j|x_i)$ as:

$$P(c_j|x_i) = \frac{P(c_j)\prod_{a=0}^{l} Gauss(x_{i,a}|c_{j,a})}{\sum_{b=0}^{k}(P(c_b)\prod_{a=0}^{l} Gauss(x_{i,a}|c_{b,a}))}$$

However, we are not computing $P(c_j|x_i)$, we are actually computing $\log P(c_j|x_i)$. So we can simplify this as:

$$\log P(c_j|x_i) = \log\left(\frac{P(c_j)\prod_{a=0}^{l} Gauss(x_{i,a}|c_{j,a})}{\sum_{b=0}^{k}(P(c_b)\prod_{a=0}^{l} Gauss(x_{i,a}|c_{b,a}))}\right)$$

$$= \log\left(P(c_j)\prod_{a=0}^{l} Gauss(x_{i,a}|c_{j,a})\right) - \log\left(\sum_{b=0}^{k}\left(P(c_b)\prod_{a=0}^{l} Gauss(x_{i,a}|c_{b,a})\right)\right)$$

$$= \log\left(P(c_j)\right) + \log\left(\prod_{a=0}^{l} Gauss(x_{i,a}|c_{j,a})\right) - \log\left(\sum_{b=0}^{k}\left(P(c_b)\prod_{a=0}^{l} Gauss(x_{i,a}|c_{b,a})\right)\right)$$

$$= \log\left(P(c_j)\right) + \sum_{a=0}^{l} \log\left(Gauss(x_{i,a}|c_{j,a})\right) - \log\left(\sum_{b=0}^{k}\left(P(c_b)\prod_{a=0}^{l} Gauss(x_{i,a}|c_{b,a})\right)\right)$$

We can calculate the first two parts of this equation, $\log\left(P(c_j)\right) + \sum_{a=0}^{l} \log\left(Gauss(x_{i,a}|c_{j,a})\right)$, quite easily. The problem lies in what was formerly the denominator of this equation. If any of the values of the Gaussians are close to zero, which is very likely, the whole term will become zero and will cause an underflow. To solve this problem, we can use the log-sum-exp trick.

Generally, this trick is the following:

$$\log\left(\sum_i e^{x_i}\right) = x_{max} + \log\left(\sum_i e^{x_i - x_{max}}\right)$$

Let's rewrite our problem term in log-sum-exp form:

$$\log\left(\sum_i e^{x_i}\right) = \log\left(\sum_{b=0}^{k}\left(P(c_b)\prod_{a=0}^{l} Gauss(x_{i,a}|c_{b,a})\right)\right)$$

$$e^{x_i} = P(c_b)\prod_{a=0}^{l} Gauss(x_{i,a}|c_{b,a})$$

$$\log e^{x_i} = \log\left(P(c_b)\prod_{a=0}^{l} Gauss(x_{i,a}|c_{b,a})\right)$$

$$x_i = \log\left(P(c_b)\prod_{a=0}^{l} Gauss(x_{i,a}|c_{b,a})\right)$$

$$= \log P(c_b) + \log\left(\prod_{a=0}^{l} Gauss(x_{i,a}|c_{b,a})\right)$$

$$= \log P(c_b) + \sum_{a=0}^{l} \log\left(Gauss(x_{i,a}|c_{b,a})\right)$$

To understand the general scope of this, we want to calculate this term for each cluster. Once we calculate this for each cluster, we have a set of p such that each $x_i = \log P(c_b) + \sum_{a=0}^{l} \log\left(Gauss(x_{i,a}|c_{b,a})\right)$ corresponds to one cluster. To find $\log\left(\sum_{b=0}^{k}\left(P(c_b)\prod_{a=0}^{l} Gauss(x_{i,a}|c_{b,a})\right)\right)$, then, we just need to apply the log-sum-exp trick as it is described above, with each x_i corresponding to a category.

Another point where underflow may occur is in the Gaussian function. Sometimes the Gaussian may return a value very close to 0, and taking the log of that will result in negative infinity. Notice that every time we use the Gaussian function, we only really care about the log of it. So we can simplify the term to be the following:

$$\log\left(Gauss(x_{i,a}|c_{j,a})\right) = \log\frac{e^{-\frac{(x_{i,a}-\mu_{j,a})^2}{2\sigma_{j,a}^2}}}{\sqrt{2\pi\sigma_{j,a}^2}}$$

$$= \log\left(e^{-\frac{(x_{i,a}-\mu_{j,a})^2}{2\sigma_{j,a}^2}}\right) - \log\left(\sqrt{2\pi\sigma_{j,a}^2}\right)$$

$$= -\frac{\left(x_{i,a} - \mu_{j,a}\right)^2}{2\sigma_{j,a}^2} - \frac{\log\left(2\pi\sigma_{j,a}^2\right)}{2}$$

$$= -\frac{1}{2}\left(\frac{\left(x_{i,a} - \mu_{j,a}\right)^2}{\sigma_{j,a}^2} + \log\left(2\pi\sigma_{j,a}^2\right)\right)$$

So, in general – we can write each $\log P\left(c_j \middle| x_i\right)$ as:

$$\log P\left(c_j \middle| x_i\right) = \log\left(P\left(c_j\right)\right) - \frac{1}{2}\sum_{a=0}^{l}\left(\frac{\left(x_{i,a} - \mu_{j,a}\right)^2}{\sigma_{j,a}^2} + \log\left(2\pi\sigma_{j,a}^2\right)\right) - \left(x_{max} + \log\sum_{b=0}^{k} e^{x_b - x_{max}}\right)$$

Where each $x_b = \log P(c_b) - \frac{1}{2}\sum_{a=0}^{l}\left(\frac{\left(x_{i,a} - \mu_{j,a}\right)^2}{\sigma_{j,a}^2} + \log\left(2\pi\sigma_{j,a}^2\right)\right)$.

MAXIMIZATION STEP

The second step in EM is to update each cluster's parameters (prior, means, and variances) given the new log likelihoods we have calculated for each cluster-example pair so that we can maximize the likelihood of the data having been generated by the model. Remember, $\log P(c_j|x_i)$ is the value in pair (i, j) of the log-likelihood pair and represents the probability that data point x_i was generated by cluster c_j. In the following calculations, we want $P(c_j|x_i)$. Since we are simply summing these values (as opposed to multiply) and not within a log, we can find this value by computing $e^{\log P(c_j|x_i)}$ for each value-cluster pair and using that where we need to use $P(c_j|x_i)$.

First update the cluster priors. For each cluster, the new cluster prior for cluster c_j is given as:

$$P\left(c_j\right) = \frac{1}{n}\sum_{i=0}^{n} P(c_j|x_i)$$

Next update the cluster means. This has to be done for each feature, obviously. This is given as (for cluster c_j and feature m):

$$\mu_{j,m} = \frac{\sum_{i=0}^{n} x_{i,m} P(c_j|x_i)}{\sum_{i=0}^{n} P(c_j|x_i)}$$

Where each part iterates over each example and $x_{i,m}$ is the value of feature m for example x_i.

Lastly, update the cluster variances. This also has to be done for each feature. For cluster c_j and feature m,

$$\sigma_{i,m}^2 = \frac{\sum_{i=0}^{n}\left(x_{i,m} - \mu_{j,m}\right)^2 P(c_j|x_i)}{\sum_{i=0}^{n} P(c_j|x_i)}$$

Where $\mu_{j,m}$ is the mean we just calculated in the last step.

LOG LIKELIHOOD AND STOPPING CONDITIONS

We want to repeat the expectation and maximization steps iteratively until we have reached convergence. That is, we want to continue the process until the log likelihood (which is what we're attempting to maximize) does not improve much more (e.g. improves less than 0.1% between iterations).

First of all, to find the improvement between trials, we should probably compare a previous log likelihood to the current one. We can use a formula similar to percent error to tell the improvement between trials:

$$\frac{|LL_{prev} - LL_{actual}|}{LL_{actual}}$$

Now we should be concerned about how to calculate the log likelihood. This involves transforming the matrix of log-likelihoods for cluster-example pairs, $\log P(c_j|x_i)$, into one scalar value. The equation for this is as follows:

$$LL = \sum_{i=0}^{n} \log\left(\sum_{j=0}^{k} P(c_j|x_i)\right)$$

For this, we also need to use the log-sum-exp trick as we are taking the log of a sum. In this case, it is slightly simpler than the previous case.

$$\log\left(\sum_{i} e^{x_i}\right) = \log\left(\sum_{j=0}^{k} P(c_j|x_i)\right)$$

$$e^{x_i} = P(c_j|x_i)$$

$$\log(e^{x_i}) = \log\left(P(c_j|x_i)\right) = x_i$$

Since our table actually keeps track of each $\log\left(P(c_j|x_i)\right)$, this is actually very handy. Basically, for each example, we need to find the max $\log P(c_j|x_i)$ for the clusters, and use that to compute the log-sum-exp. We can rewrite the log likelihood as:

$$LL = \sum_{i=0}^{n}\left(logP_{max} + \sum_{j=0}^{k} e^{logP(c_j|x_i) - logP_{max}}\right)$$