Expectation-Maximization (EM) Derivations

Mickey Chao

April 2016

1 Intuition

Expectation-Maximization (EM) is a form of unsupervised learning. We obtain an unlabeled dataset and have some idea of the type of distribution (e.g. multivariate normal, coin flips, etc.) from which the data was generated. It is straightforward to see how likely it is that a a distribution parameterized by $\bar{\theta}$ generated the datapoint $\bar{x}$. This is the quantity $P(\bar{x}|\bar{\theta})$. The idea is then to apply Bayes' rule to calculate how likely it is that given datapoints $\bar{x}$ came from a distribution parameterized by $\bar{\theta}$. This is the quantity $P(\bar{\theta}|\bar{x})$. Once we know how to calculate $P(\bar{\theta}|\bar{x})$, we will be able to find the θ^* that maximizes $P(\bar{\theta}|\bar{x})$. θ^* will then be our estimate of the true underlying distribution.

2 Problem Setup

The EM algorithm strongly resembles the K-Means algorithm. In fact, we might even say that EM is just a more general form of K-Means clustering.

Suppose there are k "clusters" or distributions from which we believe n observed datapoints came (each datapoint was generated by one of the k distributions). These distributions could be k different spherical Gaussians for the multivariate normal problem, or k different weighted coins with different probabilities of producing heads.

We have a hypothesis $\bar{\theta}$ of the parameters specifying the k distributions - that is, $\bar{\theta}$ consists of k different parameters sets, one set for each of the k distributions. For example, in the multivariate normal scenario, each set of parameters might represent the mean, μ_i, and variance, σ_i^2, of one spherical Gaussian. In the coin flip scenario, these sets of parameters might just contain p_i, the probability that the ith weighted coin gives heads. We wish to learn the most likely values of the $\bar{\theta}$ given the observed data.

As a concrete example, consider the coin flip scenario where we pick one of two coins and flip it three times. Suppose the results are $\{H, H, H\}, \{T, T, T\}, \{T, T, T\}, \{T, T, T\}$. By inspection, we would think that there is one coin that always gives heads and one coin that always gives tails. Hopefully, our EM algorithm will be able to make this conclusion as well and learn the parameters

$$\bar{\theta} = \left\{ \left\{ p(\text{select coin 1}) = \frac{1}{4}, p(H|\text{coin 1}) = 1 \right\}, \left\{ p(\text{select coin 2}) = \frac{3}{4}, p(H|\text{coin 2}) = 0 \right\} \right\}$$

3 Mathematical Derivations

Formally, let us make the following definitions:

- Let $S_n = \{\bar{x}^{(i)}\}_{i=1}^n$ be the observed data generated by the underlying distributions.
- Let $\bar{\theta}^{(i)}$ represent the unknown parameters of distribution i. Let $\Theta = \{\bar{\theta^{(i)}}\}_{i=1}^k$ be the set of parameters for all k underlying distributions that we believe exist in the model. The goal is to find the most likely Θ that would have generated the observed data S_n.
- Let $\gamma(i)$ represent the probability of selecting distribution i to generate a datapoint. That is, we expect about $\gamma(i)$ of the total points to have been generated by distribution i. Let $\bar{\gamma} = \{\gamma(1), \ldots, \gamma(k)\}$ be the set of parameters quantifying how frequently we choose each distribtion.

3.1 Attempted Solution for k Distributions - Hidden Variables Case

We can attempt to attack the problem head on by trying to find the optimal Θ that maximizes the likelihood of seeing all the observed data in general. The likelihood of seeing all the observed data given the parameters Θ is

$$L(S_n|\Theta, \bar{\gamma}) = \prod_{i=1}^{n} \sum_{j=1}^{k} \gamma(k) P(\bar{x}^{(i)}|\bar{\theta}^{(j)})$$

We can't easily take the gradient with respect to Θ because of the product operator, so instead, we can try taking the log-likelihood:

$$l(S_n|\Theta, \bar{\gamma}) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} \gamma(k) P(\bar{x}^{(i)}|\bar{\theta}^{(j)}) \right)$$

However, we still run into a problem because there is a summation inside the logarithm. If we were to try to take the gradient, it would be impossible to deal with the summation inside the logarithm. Therefore, we must try an alternative approach.

3.2 Solution for a Single Distribution

We will now consider the easier solution for when we believe there is only a single underlying distribution. This result will be used in a later section for solving the problem for k known distributions.

Our scenario corresponds to when $k = 1$ and $\gamma(1) = 1$. We can compute the likelihood of having seen the given data given some parameter setting $\bar{\theta}^{(1)}$:

$$L(S_n|\bar{\theta}^{(1)}) = \prod_{i=1}^{n} P(\bar{x}^{(i)}|\bar{\theta}^{(1)})$$

We can then compute the log-likelihood and maximize that:

$$l(S_n|\bar{\theta}^{(1)}) = \sum_{i=1}^{n} \log(P(\bar{x}^{(i)}|\bar{\theta}^{(1)}))$$

This is a relatively simple expression to maximize because we can just take the gradient with respect to $\bar{\theta}^{(1)}$ and set it equal to 0. That is, we solve

$$\nabla_{\bar{\theta}^{(1)}} l(S_n|\bar{\theta}^{(1)}) = 0$$

Of course, the actual expression for $\nabla_{\bar{\theta}^{(1)}} l(S_n|\bar{\theta}^{(1)})$ depends on our definition of what $P(\bar{x}^{(i)}|\bar{\theta}^{(1)})$ is. We will provide a concrete example of this later.

3.3 Solution for k Distributions - No Hidden Variables

Now we will consider solving a slightly more-difficult problem where there are k underlying distributions, but we know which distribution generated which datapoints. We can compute the likelihood of having seen the given data given the parameter settings:

$$L(S_n|\Theta) = \prod_{i=1}^{n} \sum_{j=1}^{k} I_{ij} P(\bar{x}^{(i)}|\bar{\theta}^{(j)})$$

where I_{ij} is an indicator variable that is 1 if distribution j generated datapoint i and 0 otherwise. We can include I_{ij} because we assume we know which distribution generated which datapoint so we can specify I_{ij} completely.

We can now compute the log-likelihood:

$$l(S_n|\Theta) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} I_{ij} P(\bar{x}^{(i)}|\bar{\theta}^{(j)}) \right)$$

It would seem at first that we are not better off than in the hidden variable case, but realize that for every setting of i, there is only one indicator I_{ij} that is nonzero. Basically, this comes from the idea that each observed $\bar{x}^{(i)}$ could only have been generated by exactly one distribution parameterized by $\bar{\theta}^{(j)}$. In fact, in each logarithm, there is only one nonzero term:

$$l(S_n|\Theta) = \sum_{i=1}^{n} \log\left(0 * P(\bar{x}^{(i)}|\bar{\theta}^{(j_1)}) + \cdots + 1 * P(\bar{x}^{(i)}|\bar{\theta}^{(j^*)}) + \cdots + 0 * P(\bar{x}^{(i)}|\bar{\theta}^{(j_k)})\right)$$

and so, in fact, this is equivalent to

$$l(S_n|\Theta) = \sum_{i=1}^{n}\sum_{j=1}^{k} I_{ij} \log(P(\bar{x}^{(i)}|\bar{\theta}^{(j)}))$$

because the I_{ij} will zero out any extraneous terms that shouldn't be included in the summation.

Now, the solution to γ is straightforward. The new estimate for $\gamma(j)$ should simply be the number of times the distribution produced a datapoint divided by the total number of datapoints observed:

$$\gamma(j) = \frac{\sum_{i=1}^{k} I_{ij}}{n}$$

Since there is no longer a summation within a logarithm, we can also compute the gradient with respect to Θ to solve for Θ:

$$\nabla_\Theta l(S_n|\Theta) = \nabla_\Theta \sum_{i=1}^{n}\sum_{j=1}^{k} I_{ij} \log(P(\bar{x}^{(i)})|\bar{\theta}^{(j)}) = \sum_{i=1}^{n} \log(P(\bar{x}^{(i)}|\bar{\theta}^{(j_i^*)})) = 0$$

where the j_i^*th distribution generated the ith observed datapoint. Notice that this corresponds to solving several solutions for a single distribution. If we take the gradient with respect to $\bar{\theta}^{(j)}$, then the solution is the same solution to

$$\sum_{i,j|I_{ij}=1} \log(P(\bar{x}^{(i)}|\bar{\theta}^{(j)})) = 0$$

3.4 Making use of the Solution to k Distributions - No Hidden Variables

Suppose we already had some estimates for Θ and γ. We can softly-assign datapoints to distributions based on the posterior probability that distribution was the true distribution that generated the observed datapoint. Let i be the observed datapoint and j be the distribution we are considering. This probability is the value

$$P(j|i) = \frac{P(j,i)}{P(i)} = \frac{\gamma(j)P(i|j)}{\sum_{t=1}^{k} \gamma(t)P(i|t)}$$

Given $P(j|i)$, rather than make a hard assignment with an indicator variable I_{ij}, we can make a soft assignment of this point to distribution j with probability $P(j|i)$.

We can slightly modify the solution to γ from the preceding section and get

$$\gamma(j) = \frac{\sum_{i=1}^{n} P(j|i)}{n}$$

When we take the gradient with respect to $\bar{\theta}^{(j)}$ we will get

$$\nabla_{\bar{\theta}^{(j)}} l((S_n|\Theta)) = \nabla_{\bar{\theta}^{(j)}} \sum_{i=1}^{n} P(j|i) \log(P(\bar{x}^{(i)}|\bar{\theta}^{(j)})) = 0$$

Now, we can update the estimates for Θ and γ.

The above calculations form the general outline of the Expectation-Maximization Algorithm. We randomly initialize our estimates for Θ and default our values of $\gamma(i) = \frac{1}{k}$. Then, given our current estimate of Θ and γ, we can softly assign points to clusters, calculate several probabilities, and then re-estimate the parameters Θ and γ to get a better prediction.

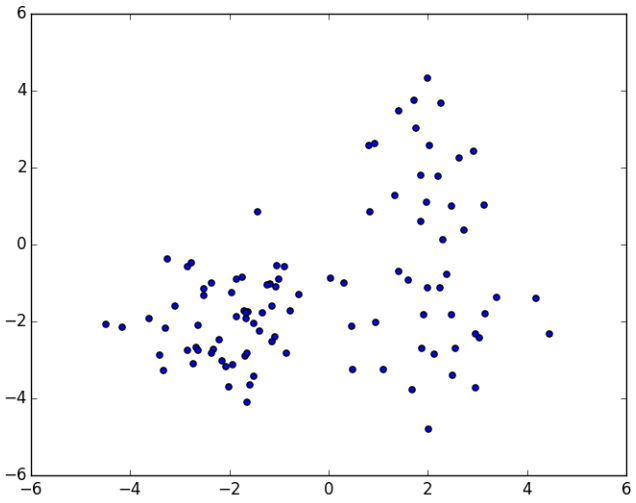
4 EM Algorithm for Gaussian Mixture Models

As a concrete example, we will now perform calculations to determine the computations to use when performing the EM Algorithm for Gaussian Mixture Models.

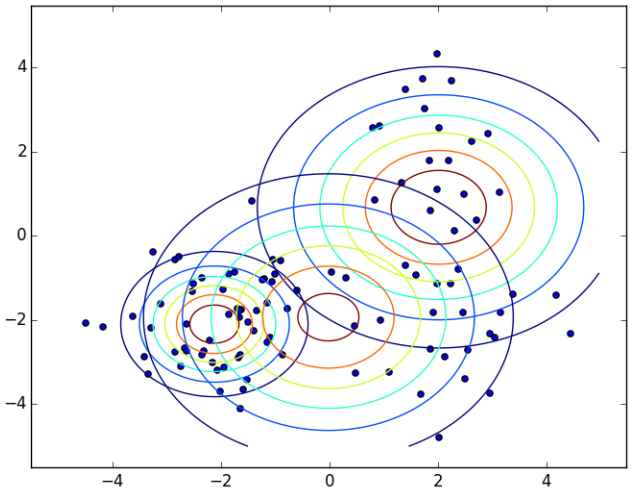
For this scenario, we will work with spherical Gaussians only. A spherical Gaussian distribution in d dimensions is parametrized by two values μ and σ where μ is the center, or mean, of the distribution and σ characterizes the variance. A spherical Gaussian has the probability density function

$$P(\bar{x}|\bar{\mu}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{(d/2)}} e^{-\frac{1}{2\sigma^2}\|\bar{x}-\bar{\mu}\|^2}$$

Here is an example of some datapoints that may be generated from three different Gaussian distributions: one centered at (2,2), one centered at (-2,-2), and one centered at (2,-2):



It should be somewhat apparent that there are three clusters of points centered around (2,2), (-2,-2) and (2,-2). The goal of EM is to predict the three Gaussians that generated these observed clusters of datapoints. Such a prediction may look like the following:



Of course, the algorithm will not perfectly find the encoded distributions, but it will come close.

4.1 Maximum Likelihood Estimates

We return to the expression

$$\nabla_{\bar{\theta}^{(j)}} \sum_{i=1}^{n} P(j|i) \log(P(\bar{x}^{(i)}|\bar{\theta}^{(j)})) = 0$$

which we never evaluated because we did not know the expression for $P(\bar{x}^{(i)}|\bar{\theta}^{(j)})$. For our specific Gaussian mixture problem, we can substitute in for $P(\bar{x}^{(i)}|\bar{\theta}^{(j)})$ and obtain

$$\nabla_{\bar{\theta}^{(j)}} \sum_{i=1}^{n} P(j|i) \log\left(\frac{1}{(2\pi\sigma^2)^{(d/2)}} e^{-\frac{1}{2\sigma^2}\|\bar{x}-\bar{u}\|^2}\right) =$$

$$\nabla_{\bar{\theta}^{(j)}} \sum_{i=1}^{n} P(j|i) \log\left(\frac{1}{(2\pi\sigma^2)^{(d/2)}}\right) + \nabla_{\bar{\theta}^{(j)}} \sum_{i=1}^{n} P(j|i) \left(-\frac{1}{2\sigma^2}\|\bar{x}-\bar{u}\|^2\right)$$

If we further consider the gradient with respect to $\bar{\mu}$, we get

$$\nabla_{\bar{\mu}} \sum_{i=1}^{n} P(j|i) \left(-\frac{1}{2\sigma^2}\|\bar{x}^{(i)}-\bar{u}\|^2\right) = -\frac{1}{\sigma^2} \sum_{i=1}^{n} P(j|i)(\bar{x}^{(i)}-\bar{\mu}) = 0 \qquad \Rightarrow \qquad \hat{\mu} = \frac{1}{\sum_{i=1}^{n} P(j|i)} \sum_{i=1}^{n} P(j|i)\bar{x}^{(i)}$$

If we consider the partial derivative with respect to σ^2, we get

$$\frac{\partial}{\partial\sigma^2} \sum_{i=1}^{n} P(j|i) \log\left(\frac{1}{(2\pi\sigma^2)^{(d/2)}} e^{-\frac{1}{2\sigma^2}\|\bar{x}^{(i)}-\bar{\mu}\|^2}\right) = \frac{\partial}{\partial\sigma^2} \sum_{i=1}^{n} P(j|i) \left(\log\left(\frac{1}{(2\pi\sigma^2)^{(d/2)}}\right) - \frac{1}{2\sigma^2}\|\bar{x}^{(i)}-\bar{\mu}\|^2\right)$$

$$= \frac{\partial}{\partial\sigma^2} \sum_{i=1}^{n} P(j|i) \left(-\frac{d}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|\bar{x}^{(i)}-\bar{\mu}\|^2\right)$$

$$= \left(-\frac{\sum_{i=1}^{n} P(j|i) d}{2}\right)\left(\frac{2\pi}{2\pi\sigma^2}\right) + \left(\frac{1}{2}\right)\left(\sum_{i=1}^{n} P(j|i)\|\bar{x}^{(i)}-\bar{\mu}\|^2\right)\frac{-1}{(\sigma^2)^2} = 0$$

We can multiply both sides by $2\sigma^4$ and get

$$-\left(\sum_{i=1}^{n} P(j|i)\right) d\frac{2\pi\sigma^2}{2\pi} + \sum_{i=1}^{n} P(j|i)\|\bar{x}^{(i)}-\bar{\mu}\|^2 = 0 \qquad \Rightarrow \qquad \sigma^2 = \frac{1}{d\sum_{i=1}^{n} P(j|i)} \sum_{i=1}^{n} P(j|i)\|\bar{x}^{(i)}-\bar{\mu}\|^2$$

Now that we have updates for $\bar{\mu}$ and σ, we are ready to outline the EM algorithm.

4.2 EM Step 0: Random Initialization

At the very start, we randomly initialize the means $\mu^{(1)}, \ldots, \mu^{(k)}$. This can be done effectively by choosing k different observed datapoints. We can also calculate the initial variances using the formula for σ that we derived previously:

$$\left(\sigma^{(1)}\right)^2, \ldots, \left(\sigma^{(k)}\right)^2 = \frac{1}{dn} \sum_{i=1}^{n} \|\bar{x}^{(i)} - \mu^{(j)}\|^2$$

Finally, we initialize $\gamma_j = \frac{1}{k}$ which is basically assuming that the k Gaussians are all equally likely to be chosen for sampling.

4.3 EM Step 1: E-Step

We can softly assign points to clusters according to the posterior probabilities:

$$P(j|i) = \frac{\gamma(i)P(\bar{x}^{(i)}|\bar{\mu}^{(j)} - \sigma^{(j)})}{\sum_{t=1}^{k} \gamma(k)P(\bar{x}^{(i)}|\bar{\mu}^{(t)}, \sigma^{(t)})}$$

and after assignment, we will have

$$\hat{n}_j = \sum_{i=1}^{n} P(j|i)$$

which is essentially the expected count, or the number of times we expect the jth Gaussian to generate a datapoint.

4.4 EM Step 2: M-Step

Now, we can maximize the likelihood of seeing the observed data given the parameters by recomputing γ, $\bar{\mu}$ and σ. We have

$$\gamma(j) = \frac{\hat{n}_j}{n} \quad , \quad \bar{\mu}^{(j)} = \frac{1}{\hat{n}_j} \sum_{i=1}^{n} P(j|i)\bar{x}^{(i)} \quad , \quad \left(\sigma^{(j)}\right)^2 = \frac{1}{d\hat{n}_j} \sum_{i=1}^{n} P(j|i)\|\bar{x}^{(i)} - \bar{\mu}^{(j)}\|^2$$

After we make this update, we can go back to step 1 of the EM algorithm and repeat until convergence

5 Regularization

One issue with our probability density function for Gaussian mixture models is that it can lead to an undesirable maximum if we set $\sigma^2 = 0$ in the expression

$$P(\bar{x}|\bar{\mu}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{(d/2)}} e^{-\frac{1}{2\sigma^2}\|\bar{x}-\bar{\mu}\|^2}$$

To guard against this kind of overfitting, we can use a slightly modified probability density function:

$$P(\bar{x}|\bar{\mu}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{(d/2)}} e^{-\frac{1}{2\sigma^2}\|\bar{x}-\bar{\mu}\|^2} + \left(\frac{1}{(2\pi\sigma)^{(d/2)}} e^{-\frac{1}{2\sigma^2}S^2}\right)^{\alpha}$$

where S^2 is some default variance that we might find from prior information and α is how much we believe the prior. This changes the log-likelihood function to become

$$l(S_n; \bar{\mu}, \bar{\theta}, \bar{\gamma}) = \sum_{i=1}^{n} \log\left(\frac{1}{(2\pi\sigma^2)^{(d/2)}} e^{-\frac{1}{2\sigma^2}\|\bar{x}-\bar{\mu}\|^2}\right) + \alpha \log\left(\frac{1}{(2\pi\sigma)^{(d/2)}} e^{-\frac{1}{2\sigma^2}S^2}\right)$$

As we can see, we will weight any poor solutions that set $\mu^{(j)} = \bar{x}^{(i)}$ to try and make $\sigma^2 = 0$ will be weighted less and the prior will help offset this poor solution. Finally, our update for σ^2 in the M-Step will then be changed to

$$\left(\sigma^2\right)^{(j)} = \frac{1}{d(\hat{n}_j + \alpha)} \left[\sum_{i=1}^{n} P(j|i)\|\bar{x}^{(i)} - \bar{\mu}^{(j)}\|^2 + \alpha S^2\right]$$

The values of α and S can be set via cross validation.