EM (Expectation Maximization) Algorithm: Part I

Motivating Example:

- Have two coins: Coin 1 and Coin 2
- Each has it's own probability of seeing "H" on any one flip. Let

$$p_1 = P(\text{ H on Coin 1 })$$

$$p_2 = P(\text{ H on Coin 2 })$$

- Select a coin at random and flip that one coin m times.
- Repeat this process n times.
- Now have data

$$\begin{array}{cccc} X_{11} & X_{12} & \cdots & X_{1m} \\ X_{21} & X_{22} & \cdots & X_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nm} \end{array} \qquad \begin{array}{c} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{array}$$

  Here, the X_{ij} are Bernoulli random variables taking values in $\{0, 1\}$ where

$$X_{ij} = \begin{cases} 1 \quad , & \text{if the jth flip for the ith coin chosen is H} \\ 0 \quad , & \text{if the jth flip for the ith coin chosen is T} \end{cases}$$

  and the Y_i live in $\{1, 2\}$ and indicate which coin was used on the nth trial.

  Note that all the X's are independent and, in particular

$$X_{i1}, X_{i2}, \ldots, X_{im} | Y_i = j \overset{iid}{\sim} Bernoulli(p_j)$$

  We can write out the joint pdf of all $nm + n$ random variables and formally come up with MLEs for p_1 and p_2. Call these MLEs $\widehat{p}_1$ and $\widehat{p}_2$. They will turn out as expected:

$$\widehat{p}_1 = \frac{\text{total \# of times Coin 1 came up H}}{\text{total \# times Coin 1 was flipped}}$$

$$\widehat{p}_2 = \frac{\text{total \# of times Coin 2 came up H}}{\text{total \# times Coin 2 was flipped}}$$

- Now suppose that the Y_i are not observed but we still want MLEs for p_1 and p_2. The data set now consists of only the X's and is "incomplete".
- The goal of the EM Algorithm is to find MLEs for p_1 and p_2 in this case.

Notation for the EM Algorithm:

- Let X be observed data, generated by some distribution depending on some parameters. Here, X represents something high-dimensional. (In the coin example it is an $n \times m$ matrix.) These data may or may not be iid. (In the coin example it is a matrix with iid observations in each row.) X will be called an "incomplete data set".

- Let Y be some "hidden" or "unobserved data" depending on some parameters. Here, Y can have some general dimension. (In the coin example, Y is a vector.)

- Let $Z = (X, Y)$ represent the "complete" data set. We say that it is a "completion" of the data given by X.

- Assume that the distribution of Z (likely a big fat joint distribution) depends on some (likely high-dimensional) parameter θ and that we can write the pdf for Z as

$$f(z;\theta) = f(x,y;\theta) = f(y|x;\theta)f(x;\theta).$$

  It will be convenient to think of the parameter θ as "given" and to write this instead as

$$f(z|\theta) = f(x,y|\theta) = f(y|x,\theta)f(x|\theta).$$

  (Note: Here, the f's are <u>different</u> pdfs identified by their arguments. For example $f(x) = f_X(x)$ and $f(y) = f_Y(y)$. We will use subscripts only if it becomes necessary.)

- We usually use $L(\theta)$ to denote a likelihood function and it always depends on some random variables which are not shown by this notation. Because there are many groups of random variables here, we will be more explicit and write $L(\theta|Z)$ or $L(\theta|X)$ to denote the **complete likelihood** and **incomplete likelihood** functions, respectively.

- The complete likelihood function is

$$L(\theta|Z) = L(\theta|X,Y) = f(X,Y|\theta).$$

- The incomplete likelihood function is

$$L(\theta|X) = f(X|\theta).$$

The Algorithm

The EM Algorithm is a numerical iterative for finding an MLE of θ. The rough idea is to start with an initial guess for θ and to use this and the observed data X to "complete" the data set by using X and the guessed θ to postulate a value for Y, at which point we can then find an MLE for θ in the usual way. The actual idea though is slightly more sophisticated. We will use an initial guess for θ and postulate an entire distribution for Y, ultimately averaging out the unknown Y. Specifically, we will look at the expected complete likelihood (or log-likelihood when it is more convenient) $\mathsf{E}[L(\theta|X,Y)]$ where the expectation is taken over the conditional distribution for the random vector Y given X and our guess for θ.

We proceed as follows.

1 Let $k = 0$. Give an initial estimate for θ. Call it $\widehat{\theta}^{(k)}$.

2 Given observed data X and assuming that $\widehat{\theta}^{(k)}$ is correct for the parameter θ, find the conditional density $f(y|X,\widehat{\theta}^{(k)})$ for the completion variables.

3 Calculate the conditional expected log-likelihood or "Q-function":

$$Q(\theta|\widehat{\theta}^{(k)}) = \mathsf{E}[\ln f(X,Y|\theta)|X,\widehat{\theta}^{(k)}].$$

Here, the expectation is with respect to the conditional distribution of Y given X and $\widehat{\theta}^{(k)}$ and thus can be written as

$$Q(\theta|\widehat{\theta}^{(k)}) = \int \ln(f(X,y|\theta)) \cdot f(y|X,\widehat{\theta}^{(k)})\, dy.$$

(The integral is high-dimensional and is taken over the space where Y lives.)

4 Find the θ that maximizes $Q(\theta|\widehat{\theta}^{(k)})$. Call this $\widehat{\theta}^{(k+1)}$.

Let $k = k + 1$ and return to Step 2.

The EM Algorithm is iterated until the estimate for θ stops changing. Usually, a tolerance ε is set and the algorithm is iterated until

$$||\widehat{\theta}^{(k+1)} - \widehat{\theta}^{(k)}|| < \varepsilon.$$

We will show that this stopping rule makes sense in the sense that once that distance is less than ε it will remain less than ε.

At this point, burning questions remain. How do we choose starting values? When will this algorithm converge to something?