COS 424: Interacting with Data

Lecturer: David Blei
Scribes: Yun-En, Ashwin
Lecture #11
03/11/2008

Let us start with a review of the EM algorithm, and try to gain some intuition on how it actually works.

$$\mathcal{L}(q,\theta) = \mathcal{E}_q p(x,z|\theta) - \mathcal{E}_q \log q(z) \tag{1}$$

When q is the posterior, $\mathcal{L}(q|\theta)$ is optimized with respect to q. The above expression can be expanded as:

$$\begin{aligned} \mathcal{L}(q|\theta) &= \sum_z q(z) \log \frac{p(z,x|\theta)}{q(z)} &(2)\\ &= \sum_z p(z|x,\theta)\, log \frac{p(z,x|\theta)}{p(z|x,\theta)} &(3)\end{aligned}$$

The denominator of the log term in the previous equation can be written as:

$$p(z|x,\theta) = \frac{p(z,x|\theta)}{p(x|\theta)}$$

Hence, the expression becomes:

$$\mathcal{L}(q,\theta) = \sum_z p(z|x,\theta) \log p(x|\theta) \tag{4}$$

The expression inside the log is independent of z. Hence, this can be taken out of the summation.

$$\begin{aligned} \mathcal{L}(q,\theta) &= \log p(x|\theta) \sum_z p(z|x,\theta) &(5)\\ &= \log p(x|\theta) &(6)\end{aligned}$$

since the summation is over the entire distribution of z.

Since $\mathcal{L}(q,\theta)$ is a bound on the objective function, and by substituting the posterior, we actually get the true objective, clearly, we cannot do any better than this with any other substitution.

As can be seen from Figure 1 , we get the lower bounding concave function $\mathcal{L}(q,\theta)$. Graphically, we push the $\mathcal{L}$ curve up until it meets the real log likelihood. In the M-step, we now try to maximize $\mathcal{L}(q,\theta)$ we just found, holding $\mathcal{L}$ fixed. Hence, we essentially move along the $\mathcal{L}$ curve to find the θ that maximizes $\mathcal{L}$. Since $\mathcal{L}$ is concave, we will have found a better log likelihood estimate than we had before. This pattern repeats by finding the best new concave function, and then moving along the curve to find the next best θ.

Note: Since we are performing a local maximization at each step, this method will be tricked by local maxima. Further, this method is likely to be slow if there is little change in θ in the M-step, which is likely to happen closer to the local extremum of the original objective function. To deal with this issue, we start off the algorithm at many different values of θ and pick the highest resulting log likelihood and its attendant θ value. To solve the slow convergence issue, we might resort to some other classification method once θ changes little in the M-step to avoid many iterations with tiny updates to θ.

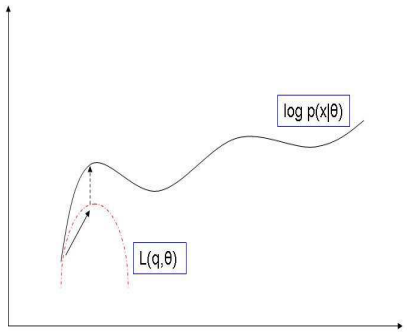

Figure 1: The E and M steps of the Expectation-Maximization Algorithm

Markov Models

We made some assumptions about our data in Naive Bayes and Mixture models: we assume that our data are IID. But words and many other data are not IID at all - their order matters, and holds information. We would like a new model that recognizes the fact that the interpretation of a word depends on the word(s) before it.

Examples of non-IID data include :

- Sequence of characters
- Spell Check
- Movements (like Walking: right usually comes after left)
- Time dependent data (eg. Weather)
- Commodity prices
- DNA Sequences

One very simple model that attempts to recognize these links between data points and the previous data point, is the Markov model, also known as the Markov chain. In this model, we assume that each point is dependent on the previous point only. This is, of course, not true a lot of the time - what word I say now probably depends on the previous several words, not just the one preceding word - but the Markov model can be generalized to include many preceding points, not just one, and can capture sequence information that Naive Bayes is blind to.

A First and Second order Markov model is represented by the Figure 2.

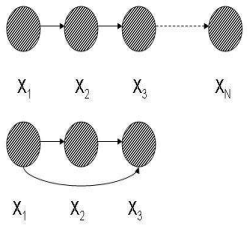

Figure 2: First and Second order Markov Models

For a first order Markov Model, the joint probability of the observed data is given by:

$$p(x_{1:N}) = p(x_1) \prod_{n=2}^{N} p(x_n|x_{n-1})$$

Parameterizing a MC

For a first order markov model, the parameterizing is done using a two-dimensional transition matrix **A**. If X_n can take up one of K values, **A** is a K x K matrix with elements

$$a_{ij} = p(x_n = j|x_{n-1} = i)$$

Similarly, for a 2^{nd} order MC, the transition matrix has dimensions K x K x K.

MLE of a 1^{st} order MC

Let us represent x_n as an *indicator vector*.

$$x_n = [0\,0\,0\,\ldots\,1\,\ldots\,0\,0]^T_{K*1}$$

where the value of the vector is 1 at position i if $x_n = i$.

Therefore, the likelihood function is:

$$p(x_{1:N}|A, \pi) = \prod_{i=1}^{K} \pi_i^{x_1^i} \prod_{n=2}^{N} \prod_{i=1}^{K} \prod_{j=1}^{K} a_{ij}^{x_{n-1}^i x_n^j} \tag{7}$$

The log likelihood function is given by:

$$\log p(x_{1:N}|A, \pi) = \sum_{i=1}^{K} x_1^i \log \pi_i \sum_{n=2}^{N} \sum_{i=1}^{K} \sum_{j=1}^{K} x_{n-1}^i x_n^j \log a_{ij} \tag{8}$$

Maximizing this expression with respect to the parameters $\mathbf{A}$ and π, we get:

$$\hat{a}_{ij} = \frac{\sum_{n=2}^{N} 1[x_{n-1} = i]1[x_n = j]}{\sum_{n=1}^{N} 1[x_n = i]} \tag{9}$$

$$\hat{\pi}_i = \frac{\sum_{n=1}^{N} 1[x_n = i]}{N} \tag{10}$$