Expectation-Maximization
COS 424 Lecture-8

Muneeb Ali
muneeb@princeton.edu

March 12, 2010

Summary: The Expectation-Maximization (EM) algorithm is a general algorithm for maximum-likelihood estimation where the data is "incomplete" or the likelihood function involves latent variables. The notion of "incomplete data" and "latent variables" is related. When we have a latent variable, the data will be incomplete since we don't observe values of the latent variables. Similarly, when the data is incomplete, we can also associate some latent variable with the missing data. EM is an iterative method which alternates between performing an expectation (E) step (computing the expectation of the log-likelihood evaluated using the current estimate for the variables) and a maximization (M) step that computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E-step. EM is a convenient algorithm for certain maximum likelihood problems. It's a viable alternative to Newton or Conjugate Gradient algorithms (an algorithm for the numerical solution of particular systems of linear equations). EM is an offline algorithm that is more "fashionable" than Newton or Conjugate Gradient in the sense that there are more research papers using EM. It is also simpler. There are many extensions to EM e.g., variational methods or stochastic EM.

Gaussian Mixture Model: Consider a simple Gaussian mixture with K components (can visualize them as clusters with different shapes e.g., small or large). Each Gaussian is called a component (different circle for each Gaussian in Slide 4). To generate an observation:

- Pick a component k with probabilities $\lambda_1, \lambda_2, \ldots, \lambda_k$ and $\sum_k \lambda_k = 1$
- Generate x from component k with probability $\mathcal{N}(\mu_i, \sigma)$

For a simple Gaussian mixture model (GMM) the standard deviation σ is known and is a constant. Now consider a mixture of two Gaussians with trainable standard deviations. The likelihood becomes infinite when one of them specializes on a single observation (see figure on Slide 5). Maximum likelihood estimation works for all discrete probabilistic models and for some (and not all) continuous probabilistic models. The simple example of Slide 5 (of one Gaussian specializing on a single observation) is not one of them. So in theory MLE doesn't work on simple GMM, but people ignore the problem and get away with it anyway. There are two possible explanations for this:

- Unlike discrete distributions, densities are not bounded (ceiling fixes the problem). There is never just one example (on which the Gaussian can specialize on). Ceiling is reasonably low enough and we generally just want to avoid infinity. Because we will find local maxima, instead of global, it will work (because of non-convexivity).
- Probability, in general, is mathematically sound. You condition on the samples. From the point of view of strict probability, what we are doing is wrong. But in practice it works (an analogy can be the wild west, there were no formal rules but the system still seemed to somehow work).

Expectation-Maximization: Informally, the EM algorithm starts with randomly assigning values to all the parameters to be estimated. It then iteratively alternates between the expectation (E) step and the maximization (M) step. In the E-step, it computes the expected likelihood for the complete data (the Q-function) where the expectation is taken as the computed conditional distribution of the latent variables given the current settings of parameters and our observed (incomplete) data. In the M-step, it re-estimates all the parameters by maximizing the Q-function. Once we have a new generation of parameter values, we can repeat the E-step and another M-step. This process continues until the likelihood converges, i.e., reaching a local maxima. Intuitively, what EM does is to iteratively "guess" the values of the hidden variables and to re-estimate the parameters by assuming that the guessed values are the true values.

The EM algorithm is a "hill-climbing" approach, thus it can only be guanranteed to reach a local maxima. When there are multiple maximas, whether we will actually reach the global maxima clearly depends on where we start i.e., if we start at the "right hill", we will be able to find a global maxima. When there are multiple local maximas, it is often hard to identify the "right hill". There are two commonly used strategies to solving this problem. The first is that we try many different initial values and choose the solution that has the highest converged likelihood value. The second uses a much simpler model (ideally one with a unique global maxima) to determine an initial value for more complex models. The idea is that a simpler model can hopefully help locate a rough region where the global optima exists. We start from a value in that region to search for a more accurate optima using a more complex model.

We only observe the $x_1, x_2, \ldots$ values. Some models would be very easy to optimize if we knew which components $y_1, y_2, \ldots$ generated them. For a given X, we guess a distribution

$Q(Y|X)$. Regardless of our guess, $\log L(\theta) = \mathcal{L}(Q,\theta) + \mathcal{D}(Q,\theta)$ with

$$\mathcal{L}(Q,\theta) = \sum_{i=1}^{n}\sum_{y=1}^{K} Q(y|x_i)\log\frac{P_\theta(x_i|y)P_\theta(y)}{Q(y|x_i)} \tag{1}$$

and

$$\mathcal{D}(Q,\theta) = \sum_{i=1}^{n}\sum_{y=1}^{K} Q(y|x_i)\log\frac{Q(y|x_i)}{P_\theta(y|x_i)}\,. \tag{2}$$

Proof:

$$\begin{aligned}
\mathcal{L}(Q,\theta) + \mathcal{D}(Q,\theta) &= \sum_{i=1}^{n}\sum_{y=1}^{K} Q(y|x_i)\left[\log\frac{P_\theta(x_i|y)P_\theta(y)}{Q(y|x_i)} + \log\frac{Q(y|x_i)}{P_\theta(y|x_i)}\right] \\
&= \sum_{i=1}^{n}\sum_{y=1}^{K} Q(y|x_i)\log\frac{P_\theta(x_i|y)P_\theta(y)}{P_\theta(y|x_i)} \;=\; \sum_{i=1}^{n}\sum_{y=1}^{K} Q(y|x_i)\log P_\theta(x_i) \\
&= \sum_{i=1}^{n}\log P_\theta(x_i) \;=\; \log L(\theta)\,.
\end{aligned}$$

$\mathcal{L}(Q,\theta)$ is easy to maximize. Similar to maximum likelihood for a single gaussian using examples weighted by $Q(y|x_i)$.

$\mathcal{D}(Q,\theta)$ is the KL-divergence $D(Q_{Y|X}||P_{Y|X})$. The Kullback-Leibler (KL) divergence is a non-symmetric measure of the difference between two probability distributions P and Q. What is important is that it is positive.

Now for EM (think of it like "pushing", Slide 9), first change Q to minimize $\mathcal{D}$, leaving $\log L$ unchanged and change θ to maximize $\mathcal{L}$. Meanwhile $\mathcal{D}$ can only increase.

The E-step is:

$$q_{ik} \leftarrow \frac{\lambda_k}{\sqrt{|\sum_k|}} e^{-\frac{1}{2}(x_i-\mu_k)^T \sum_k^{-1}(x_i-\mu_k)} \tag{3}$$

which is quick as we don't even normalize and the M-step is:

$$\mu_k \leftarrow \frac{\sum_i q_{ik}x_i}{\sum_i q_{ik}} \qquad \Sigma_k \leftarrow \frac{\sum_i q_{ik}(x_i-\mu_k)(x_i-\mu_k)^T}{\sum_i q_{ik}} \qquad \lambda_k \leftarrow \frac{\sum_i q_{ik}}{\sum_{iy} q_{iy}} \tag{4}$$

Here the λ are normalized.

Some points to note for implementation are:

- The q_{ik} are often very small because e^{-x} gets quickly very small when x is moderately large. So compute $\log q_{ik}$ instead of q_{ik}.

- EM gets easily stuck in local maxima.
- Computing log likelihood is useful for monitoring progress. Right place to compute is after E-step and before M-step.

EM for GMM: In the illustration on Slide 11 - 18, the green, red, and blue big circles are approximations of Gaussians. Now, we can do the M-step. We see that after 20-iterations, it converges. The Gaussian Mixture Model (GMM) can be used for anomaly detection. We model PX with a GMM and declare anomaly when density falls below a threshold (see figure on Slide 19). The yellow area is where the density falls below a threshold - alarm condition. GMM can also be used for classification. Basically model $PX|Y = y$ for eery class with GMM and then calculate Bayes optimal decision boundary. In the example (Slide 20), there are 2 models. A red class A and a grey class B. In yellow we reject the examples and are ambiguous in the green area.

So GMM seems very useful, but what is the price for using it? For starters, estimating densities is nearly impossible. If k is high, can get any kinds of density. Secondly, can we trust the GMM distributions? Maybe when lots of data points or when in very low dimension. Decisions are actually more reliable than GMM distributions because they have a more clear decision boundary than distributions. However, you still need cross-validation to check results i.e., results can be good or suspicious (so need to do cross-validation).

Other Mixture Models: We also consider the Bernoulli distribution; which is named after Swiss scientist Jacob Bernoulli and is a discrete probability distribution that takes value 1 with success probability p and value 0 with failure probability $q = 1 - p$. Can think of dictionary of words and detect presence or absence of words; 1 if word appears and 0 if the word doesn't appear. As another example, suppose you are in "drug testing" and want to know how long a patient survives after treatment. Case-1: treatment doesn't work. Case-2: die after some time e.g., cancer patients. Case-3: die long after with something else (can't model that something else). Need to know what proportion was cured and what had life extended. Need lots of data points.

Consider problems where we need to find e.g., is something green? Is it square? Does it have tail? Hands? In Slide 24, we get μ_i if x_i is 1. The covariance is diagonal. x_i is the coefficient and μ_i are the parameters. With $\sum_k \lambda_k = 1$ (normalize), Covariance is no longer diagonal and components are different. For example, some will say yellow is 0, and some will say yellow is 1. By using Bernoulli, we weaken the independence assumptions.

For Bernoulli, the log likelihood is $\log L(\theta) = \sum_{i=1}^{n} \log \sum_{i=1}^{k} \lambda_k P_{\mu_k}(x_i)$. And we are given a dataset $x = x_1, x2, \ldots, x_n$. Decomposition is similar to Slide 8, but now:

$$q_{ik} \leftarrow \lambda_k P_{\mu_k(x_i)} \tag{5}$$

and

$$\mu_k \leftarrow \frac{\sum_i q_{ik} x_i}{\sum_i q_{ik}} \qquad \lambda_k \leftarrow \frac{\sum_i q_{ik}}{\sum_{iy} q_{iy}} \tag{6}$$

In Slide 26, we get basically the same graph (as Slide 8). We change θ and Q and can only go higher (didn't normalize). This is fairly simple.

Data with Missing Values: Data with missing values is a fact of life. If they were known, we can just fit a big Gaussian. Since we don't know something, we will guess it. $Q(y|x)$ is just random if patterns of missing are independent i.e., had correct example, but someone randomly removed points. But in real life missing values are there for a reason e.g., below some threshold (garbage) and not really random.

Conclusion: Probabilistic model is nice to have, if you can pay for it. We need to be pretty sure what we want and ask the question – can we afford it? For example, in a business know what you want i.e., want to maximize earnings (dollars). Or in a post office, reduce the cost of sending mail (can see what is the price of correcting misclassified envelopes manually vs. shipping them to the wrong address). However, if we want to improve quality of life and not just money then that's not so simple. With probabilistic model you are solving a harder problem i.e., needs to be good everywhere even near the boundaries. How do we know if we can afford a probability model? Most of the time we don't know and just try. Some people recommend that try to always use a probability model, unless you can't afford it.