CSE250A Fall '12: Discussion Week 6

Aditya Menon (akmenon@ucsd.edu)

November 8, 2012

1 Schedule for today

- Recap of EM algorithm.
- Example: flipping two coins.
- Mixture models.

2 Recap of EM

We have applied the principle of maximum likelihood to learn the CPT parameters θ of a Bayesian network over nodes $X = (X_1, \ldots, X_N)$. The principle was to choose

$$\hat{\theta} = \text{argmax}_\theta \, \mathcal{L}(\theta)$$

where $\mathcal{L}(\theta) = \log \Pr[X; \theta]$. In particular, this procedure may be employed given some set of random samples $\{x^{(t)}\}_{t=1}^T$ from the network.

The principle of maximum likelihood applies regardless of the nature of $\mathcal{L}(\theta)$. But sometimes, it may be the case that $\mathcal{L}(\theta)$ is a "complicated" function in the sense of being not convex, or not differentiable. In such cases, solving the maximization to find $\hat{\theta}$ may be computationally difficult.

Fortunately, we can sometimes estimate $\hat{\theta}$ if $\Pr[X; \theta]$ breaks into several simpler components. In particular, we can often introduce a *hidden* variable Z, and use the fact that

$$\Pr[X; \theta] = \sum_Z \Pr[X, Z; \theta] = \sum_Z \Pr[X|Z; \theta] \Pr[Z; \theta].$$

Now suppose that we could get samples from both X and Z. Then, if $\Pr[X|Z]$ and $\Pr[Z]$ are "simple" functions, estimating θ by $\text{argmax}_\theta \Pr[X, Z; \theta]$ is in principle feasible.

However, we generally only have samples available from X. So, the above seems like it may be a fruitless exercise. Fortunately, there is a solution: the EM algorithm. The basic idea can be illustrated as follows:

1. Pick a random initial value for θ, call it θ_0. Remember that this means we generate random values for the CPT entries. This likely gives a terrible value of $\mathcal{L}$, so we'd like to refine it.
2. If we knew Z, we claimed that estimating θ would be simple. But we only know X. So we just *generate* an instance of Z from $\Pr[Z|X; \theta_0]$. That is, based on the current guess for the CPT parameters, and the data that we observed, we take a guess as to what the corresponding hidden variable might look like. Call the resulting sample z.

3. Now apply the principle of maximum likelihood to

$$\tilde{\mathcal{L}}(\theta) := \log \Pr[X, z; \theta]$$

We use $\tilde{\mathcal{L}}$ to denote that this is not really the true likelihood of the observed data X, but rather a crude approximation to it based on our sample z. We hope that the resulting solution θ_1 improves the true likelihood $\mathcal{L}$.

4. Repeat steps (2)-(3) until convergence.

The EM algorithm is a refinement of the above. Instead of taking a sample z from $\Pr[Z|X;\theta_0]$ and optimizing $\log \Pr[X, z; \theta]$, we consider optimizing the *expected* value of $\log \Pr[X, Z; \theta]$ under the distribution $\Pr[Z|X;\theta_0]$. It is a remarkable fact that the resulting solution necessarily improves the true likelihood, $\log \Pr[X;\theta]$.

3 Example: flipping two coins

We previously considered a setting where we have a biased coin that lands heads with probability p. We represented this as a trivial Bayesian network with a single node X. We showed that the maximum likelihood estimate of p given samples (or coin flips) $\{x^{(t)}\}_{t=1}^T$ is $\hat{p} = (1/T)\sum_{t=1}^T x_t$.

Now suppose that we have *two* biased coins, with biases p_0 and p_1. If we separately flipped the coins a number of times, we could apply the above estimate to compute both biases. Now suppose that these are the only two coins available for a toss to decide which team gets first play in a sports match. So, the coin toss for a game occurs as follows: the umpire picks one of the two coins, flips it, and the result is used to determine who plays first. There are two questions we are interested in: how often does each coin get flipped, and what are the biases of each coin?

Formally, as a Bayesian network, we have a binary node X representing the outcome of the umpire's flip. This node has as parent a binary node Z which reflects the coin that the umpire chooses to flip. That is,

$$\Pr[X = 1|Z = z] = p_z$$
$$\Pr[Z = z] := \pi_z.$$

Clearly $\pi_0 + \pi_1 = 1$. Our goal is thus to estimate $\theta = (p_0, p_1, \pi_0)$. Note that

$$\Pr[X = x;\theta] = \sum_z \Pr[X = x|z;\theta] \cdot \Pr[z;\theta] = \sum_z p_z^x (1 - p_z)^{(1-x)} \cdot \pi_z.$$

We will consider different settings in terms of the data available, and explain how to estimate θ in each.

Setting (a). Suppose the umpire tells us which coin he is going to flip. That is, we have samples $\{(x^{(t)}, z^{(t)})\}$ at our disposal. We can then just apply standard maximum likelihood on $\mathcal{L}(\theta)$,

$$\mathcal{L}(\theta) = \sum_{t=1}^T \log \Pr[x^{(t)}, z^{(t)};\theta] = \sum_{t=1}^T (x^{(t)} \log p_{z^{(t)}} + (1 - x^{(t)}) \log(1 - p_{z^{(t)}})) + \log \pi_{z^{(t)}}.$$

Intuitively, we would expect that

$$\hat{p}_z = \frac{\sum_{t=1}^T \mathbf{1}[z = z_t] \cdot x_t}{\sum_{t=1}^T \mathbf{1}[z = z_t]}$$
$$\hat{\pi}_z = \frac{1}{T}\sum_{t=1}^T \mathbf{1}[z = z_t].$$

That is, to find the bias of each coin, restrict attention to cases where that coin was flipped. To find the probability of choosing a coin, find the fraction of times it was picked.

Setting (b). Suppose that we just know the outcome of the umpire's flip. That is, we have samples $\{x^{(t)}\}$. The likelihood is

$$\begin{aligned}\mathcal{L}(\theta) &= \sum_{t=1}^{T} \log \Pr[x^{(t)}; \theta] \\ &= \sum_{t=1}^{T} \log \sum_{z} \Pr[x^{(t)}, z; \theta] \\ &= \sum_{t=1}^{T} \log \left(\sum_{z} p_z^{x^{(t)}} (1-p_z)^{(1-x^{(t)})} \cdot \pi_z \right).\end{aligned}$$

We must find the optimal values of (p_0, p_1, π_0). This looks hard! Fortunately, we can look to apply EM here. Let's apply the templates for the E-step and M-step:

E-step. We need to compute $\Pr[Z|X;\theta]$. By Bayes' rule, this is

$$\begin{aligned}\Pr[Z=z|X=x] &= \frac{\Pr[X=x|Z=z]\Pr[Z=z]}{\sum_{z'}\Pr[X|z']\Pr[z']} \\ &= \frac{p_z^x(1-p_z)^{(1-x)}\cdot\pi_z}{\sum_{z'} p_{z'}^x(1-p_{z'})^{(1-x)}\cdot\pi_{z'}}.\end{aligned}$$

Let's call this quantity $\gamma(z,x)$ for clarity. Given the outcome of a coin flip, this tells us how likely our parameters think it is that the umpire chose each of the two coins.

M-step. We need to update all CPT parameters, that is, $\pi_z = \Pr[Z=z]$ and $p_z = \Pr[X=1|Z=z]$. The updates are

$$\begin{aligned}\pi_z &= \frac{\sum_{t=1}^{T}\Pr[Z=z|x^{(t)}]}{\sum_{t=1}^{T}\Pr[\emptyset|x^{(t)}]} \\ &= \frac{\sum_{t=1}^{T}\gamma(z,x^{(t)})}{T}\end{aligned}$$

$$\begin{aligned}p_z &= \frac{\sum_{t=1}^{T}\Pr[X=1,Z=z|x^{(t)}]}{\sum_{t=1}^{T}\Pr[Z=z|x^{(t)}]} \\ &= \frac{\sum_{t=1}^{T}\Pr[X=1|Z=z,x^{(t)}]\Pr[Z=z|x^{(t)}]}{\sum_{t=1}^{T}\Pr[Z=z|x^{(t)}]} \\ &= \frac{\sum_{t=1}^{T}\gamma(z,x^{(t)})\cdot x^{(t)}}{\sum_{t=1}^{T}\gamma(z,x^{(t)})}\end{aligned}$$

The updates are intuitive. For π_z, we average all the posteriors $\Pr[Z|X]$. This means that we look at how likely each coin was for all the samples, and take the average score as being the probability of choosing that particular coin. For p_z, we count the number of heads as before, except that we now weight each such heads with our uncertainty as to which coin it was that was responsible for that flip.

Contrast the updates to those in setting (a): instead of using a hard indicator function, we use soft weights γ to denote our uncertainty in the true z values.

Setting (c). Suppose we know the outcome of the umpire's flip, but not which coin was flipped. That is, we have samples $\{x^{(t)}\}$. However, the generous umpire tells us beforehand what the biases of each coin are.

That is, p_z is known, but π_z is not, and we must estimate it. We need to optimize the same likelihood as setting (b) with respect to π_z; remember that the p_z's are fixed constants. This can be done by setting the gradient equal to 0 and with Lagrange multipliers. But we can also apply the EM updates from the previous setting. Of course, we do not look to update p_z in the M-step, because that is a fixed constant. Similarly, when computing $\gamma(z, x)$, we use the given value of p_z, but for π_z use our current estimate.

4 Mixture models

The above can be thought of as a special case of a *mixture model*. We say that X has a mixture distribution if

$$\Pr[X = x; \theta] = \sum_{z=0}^{K-1} \pi_z f_z(x; \theta_z),$$

where $0 \leq \pi_z \leq 1$, $1^T \pi = 1$, and for each z, f_z denotes some distinct probability distribution. In the above, we had

$$\Pr[X = x; \theta] = \sum_{z=0}^{1} \pi_z \text{Bernoulli}(x; \theta_z),$$

where $\theta_z = (p_z)$. In general the π_z's are known as *mixing weights* and the f_z's as *mixture components*.

We can also think of a mixture distribution as being the result of the influence of a hidden variable Z on X. Specifically, if we have

$$\Pr[X = x | Z = z] = f_z(x; \theta_z)$$
$$\Pr[Z = z] = \pi_z,$$

then it should be clear that we recover the same distribution for $\Pr[X = x]$ above.

In general, when we wish to fit a mixture model, we need to estimate two things: the mixing weights, and the parameters for each mixture component. EM is one approach to doing this, and the analysis of the previous section may be largely borrowed to handle the general mixture model case. In particular, we compute the $\gamma(z, x)$'s as before (called the *responsibilities*), and the M-step update for the mixing weights stays the same. What changes is the M-step update for the parameters θ_z, because this depends on the nature of the distribution f_z.