PGM Course Notes: Exact Learning Methods

Karl Pichotta
pichotta@cs.utexas.edu

November 19, 2012

1 ML Estimate

We have

$$p(x;\theta) = \exp\{\theta^T \phi(x) - A(\theta)\}$$

We assume φ is given and we want to learn θ given data. We assume the data

$$x^{(1)}, \ldots, x^{(n)}$$

are drawn iid from some distribution.

We maximize the log likelihood:

$$\max_\theta \ell(\theta) = \max_\theta \log \prod_{i=1}^n p(x^{(i)};\theta) \tag{1}$$

$$= \max_\theta \sum_{i=1}^n \log p(x^{(i)};\theta) \tag{2}$$

$$= \max_\theta \frac{1}{n} \sum_{i=1}^n \log p(x^{(i)};\theta) \tag{3}$$

$$= \max_\theta \frac{1}{n} \sum_{i=1}^n \left(\theta^T \phi(x^{(i)}) - A(\theta)\right) \tag{4}$$

$$= \max_\theta \left\{ \theta^T \left(\sum_{i=1}^n \frac{1}{n} \phi(x^{(i)}) \right) - A(\theta) \right\} \tag{5}$$

we define the sum to be $\hat{\mu}$, the empirical expectation. Writing it out:

$$\max_\theta \left\{\theta^T \hat{\mu} - A(\theta)\right\}$$

we see the similarities to the problems we solved when doing inference.

Differentiating the above, it becomes clear we want $\hat{\theta}$ that satisfies

$$\nabla A(\hat{\theta}) = \hat{\mu}$$

As an example, suppose we're using the canonical overcomplete parametrization:

$$p(x;\theta) = \exp\left\{\sum_s \theta_s(x_s) + \sum_{(s,t)} \theta_{st}(x_s, x_t) - A(\theta)\right\}$$

with e.g.

$$\theta_{st}(x_s, x_t) = \sum_{jk} \theta_{stjk} \mathbf{1}_{stjk}(x_s, x_t)$$

So we have

$$\hat{\mu}_{sj} = \hat{\mathbb{P}}[x_s = j] = \frac{1}{n}\sum_i \mathbf{1}[x_s^{(i)} = j]$$

and

$$\hat{\mu}_{stjk} = \hat{\mathbb{P}}[x_s = j, x_t = k].$$

So the question we ask now is: given $\hat{\mu}$, what is a $\hat{\theta}$ such that $\nabla A(\hat{\theta}) = \hat{\mu}$?

2 MLE on trees

We consider a tree. We assume $\hat{\mu}_s(x_s)$ and $\hat{\mu}_{st}(x_s, x_t)$ are positive, so we can take the logs. We define

$$\hat{\theta}_s(x_s) = \log \hat{\mu}_s(x_s)$$

and

$$\hat{\theta}_s(x_s, x_t) = \log \frac{\hat{\mu}_s(x_s)}{\hat{\mu}_s(x_s)\hat{\mu}_t(x_t)}$$

We can verify that, assuming we have a tree, with these definitons we get:

$$\nabla A(\hat{\theta}) = \hat{\mu}$$

that is

$$P_{\hat{\theta}}[X_s] = \hat{\mu}_s(x_s) \tag{6}$$
$$P_{\hat{\theta}}[X_s, X_t] = \hat{\mu}_{st}(x_s, x_t). \tag{7}$$

How do we get this? Recall we have

$$p_{\hat{\theta}}(x) = \exp\left\{\sum_s \hat{\theta}_s(x_s) + \sum_{st} \hat{\theta}_{st}(x_s, x_t) - A(\hat{\theta}))\right\} \tag{8}$$
$$= \exp\left\{\sum_s \log \hat{\mu}_s(x_s) + \sum_{st} \log \frac{\hat{\mu}_{st}}{\hat{\mu}_s\hat{\mu}_t} - A(\hat{\theta}))\right\} \tag{9}$$
$$= \prod_s \hat{\mu}_s(x_s) \prod_{st} \frac{\hat{\mu}_{st}(x_s, x_t)}{\hat{\mu}_s\hat{\mu}_t} \exp(-A(\hat{\theta})) \tag{10}$$

First, note that

$$A(\hat{\theta}) = 0$$

So therefore we have

$$P_{\hat{\theta}}(x) = \prod_s \hat{\mu}_s(x_s) \prod_{st} \frac{\hat{\mu}_{st}(x_s, x_t)}{\hat{\mu}_s \hat{\mu}_t}$$

Now, we can verify through induction that the marginals you get under a tree are

$$\mathbb{P}_{\hat{\theta}}(x_s) = \hat{\mu}_s$$
$$\mathbb{P}_{\hat{\theta}}[x_s, x_t] = \hat{\mu}_{st}(x_s, x_t)$$

So in a tree the marginals are totally specified by the $\hat{\mu}$'s. Convince yourself of this by drawing out a simple three-node tree and marginalizing out one of the leafs. A leaf participates in only one pairwise marginal, and if you marginalize that out, it will create a nodewise marginal. Using this insight, we unroll the marginals, and what you end up with are the marginals specified as above.

So if we have a tree and we know what the graph looks like, then learning the ML parameters is easy. We have a closed-form solution that we can calculate that is linear in the number of edges.

3 From Trees to Graphs

So what if we have a more general distribution

$$p(x; \theta) = \exp\left\{ \sum_{C \in \mathcal{C}} \theta_C(x_C) - A(\theta) \right\}$$

Note that

$$\theta_C(x_C) = \sum_{J \in \mathcal{X}^{|C|}} \theta_{C;j} \mathbf{1}_{c;j}(x_C)$$

The key insight is in (10), which should look a lot like junction tree. Recall that, in JT, if the graph is triangulated, we can write

$$\mathbb{P}(X) = \frac{\prod_{C \in \mathcal{C}} P(X_C)}{\prod_{S \in \mathcal{S}} P(X_S)}$$

With $\mathcal{C}$ the set of cliques and $\mathcal{S}$ the set of separators. So JT did some local separations so that what we ended up with were local marginals:

$$\frac{\prod_C \psi_C(x_C)}{\prod_S \psi_S(x_S)}$$

Recall that any distribution that's a graphical model for a triangulated graph can be written down this way.

So let us consider

$$\hat{\mu}_C(x_C) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}[X_C^{(i)} = x_C]$$

the empirical marginal of the clique C. Note that these really are marginals of the variables in X_C. These are the moments w.r.t. the empirical distribution, assigning an equal mass to all $1/n$ training items.

What is $\hat{\theta}$? Well, consider:

$$\hat{\theta}_C(x_C) = \log \frac{\hat{\mu}_C(x_C)}{\prod_{S \in \mathcal{S}(C)} \hat{\mu}_s(x_s)}$$

and

$$\hat{\theta}(x_S) = \log \hat{\mu}_S(x_S)$$

We can verify that

$$p_{\hat{\theta}}(x) = \exp\left\{ \sum_{S \in \mathcal{S}} \hat{\theta}_S(x_S) + \sum_{C \in \mathcal{C}} \hat{\theta}_C(X_C) \right\}$$

satisfies

$$\mathbb{E}_{\hat{\theta}}[\mathbf{1}_{C,J}(x)] = \hat{\mu}_{C,J}$$

Note that we don't have $A(\hat{\theta})$, but we don't need it if our graph is triangulated (because the graph will normalize).

We can see that

$$\mathbb{E}_{P_{\hat{\theta}}}[\phi(x)] = \hat{\mu}$$

So we can go from one set of moments to another set of moments. These are called the "moment matching conditions". That is, you want a distribution such that the moments above match.

If you have a junction tree (which is, recall, a clique tree satisfying a certain set of properties we talked about before), all you need to learn the MLE of the graphical model is to give the corresponding clique and separator functions.

Suppose we have strong reasons to believe a nontriangulated graph is our proper structure. Then our distribution lies in a set of distributions given by a graph. There's nothing stopping us from adding edges to make it triangulated, because our distributions will exist in the new graph's family of distributions. So requiring a triangulated graph is not a particularly restrictive condition.

So it seems like we solved learning. What's the catch? Supposing we have an $n \times n$ grid graph—pairwise cliques over the n^2 nodes. Each node has a constant number of neighbors, regardless of n. To construct the clique tree, though, note the size of the largest clique depends on the treewidth. So the clique sizes required to construct a junction tree could be really large. In the grid example, the size of cliques could be as big as $\sqrt{p}$ (with p the number of nodes), so storage could be exponential in n. (Seeing that this is the case is somewhat nontrivial. We can think of constructing a junction tree as the result of doing variable elimination. If you imagine doing VE on the grid, you'll end up with pretty big cliques).

The upshot is, if the clique sizes aren't too large, then we're in business. If the cliques get big, though, our closed form formulas aren't too useful. This is similar to the drawbacks we find in the regular Junction Tree and Variable Elimination algorithms. Otherwise we have to use optimization-based methods.

4 Iterative Proportional Fitting / Block Coordinate Ascent

Let us consider the general distribution

$$p(x;\theta) = \exp\left\{\sum_{C\in\mathcal{C}} \theta_C(x_C) - A(\theta)\right\}$$

And suppose we want to find the ML estimate. We want

$$\max_\theta\{\theta^T\hat{\mu} - A(\theta)\} \equiv \max_\theta \ell(\theta)$$

in general, this doesn't hjave a closed form (for example, a nontriangulated graph, like a grid). One standard way to do the optimization is with **block coordinate ascent**. So what are our parameters?

$$\theta := (\theta_{C,J} : C \in \mathcal{C}, J \in \mathcal{X}^{|C|})$$

Let's focus on the parameters for a particular clique C:

$$(\theta_{CJ})_{J\in\mathcal{X}^{|C|}}$$

and optimize over that, fixing every other parameter. This ends up giving us simple closed form expressions and is called **iterative proportional fitting**, which has been used for a long time. So what do we get when we solve for θ_{CJ} when we hold everything else fixed? We do

$$\frac{\partial \ell(\theta)}{\partial \theta_{CJ}} = 0 \tag{11}$$

$$\hat{\mu}_{CJ} - \frac{\partial A(\theta)}{\partial \theta_{CJ}} = 0 \tag{12}$$

$$\hat{\mu}_{CJ} - \mu_{CJ} = 0 \tag{13}$$

where we use the fact that, for an exponential family,

$$\frac{\partial A(\theta)}{\partial \theta_{CJ}} = \mu_{CJ}$$

So were are matching the moments for a single clique.

What does the algorithm look like?

1. Iterate for $t = 1, \ldots$.
   (a) Pick a clique $C = C(t)$. Calculate $\theta^{(t)}$, your current estimate of the params. Compute the moments of the current estimates:

$$\mu_{CJ}^{(t)} = \mathbb{P}_{\theta^{(t)}}[X_c = J]$$

(b) Set

$$\theta_{CJ}^{(t+1)} := \theta_{CJ}^{(t)} + \log \frac{\hat{\mu}_{CJ}}{\mu_{CJ}^{(t)}}$$

for all $J \in \mathcal{X}^{|C|}$, and keep the other θ params the same.

So we are just matching one moment at a time, instead of matching all the moments simultaneously. Since $A(\theta)$ is convex, this is guaranteed to converge to the global optimum. We can't really do this for complicated graphs, because at our step of computing the marginals $(\mu_{CJ}^{(t)})$, we're performing inference, so it's not particularly tractable in general. However, we can use it for relatively small, nontriangulated graphs.

What is

$$\mathbb{E}_{\theta^{(t+1)}}[\mathbb{I}_{CJ}(X_C)]?$$

If we satisfy the stationary condition

$$\hat{\mu}_{CJ} - \mu_{CJ} = 0$$

then we'll get an optimal answer. So this means

$$\mathbb{E}_{\theta^{(t+1)}}[\mathbb{I}_{CJ}(X_C)] = \hat{\mu}_{CJ}$$

and that will give us that Iterative proportional fitting is block coordinate descent. That is, we're setting θ_{CJ} so that the corresponding moments match, that is,

$$\mu_{CJ} = \hat{\mu}_{CJ}.$$