Parameter Learning in Graphical Models

Amir Globerson

So far we assumed that the parameters of our graphical model were given. For example, the functions $\phi_c(x_c)$ in undirected models or $p(x_i|x_{Pa(i)})$ in directed models. In many problems this is not the case, and even if we could potentially set these manually by using an expert this would be too costly. The other problem is of course that we do not necessarily know even the graph G. In the following classes we will focus on the problem of learning the parameters of a graphical model for a fixed graph G. This problem is known as parameter learning or parameter estimation. Afterwards, we will also discuss how to learn the graph structure. This problem is known as structure learning.

Lets get more specific about what we mean by parameters. Say we have an undirected model:

$$p(\boldsymbol{x}) = \frac{1}{Z}\prod_c \phi_c(x_c) \tag{1}$$

1 Parametric Models

When learning, we might want to make some further assumptions on the structure of the functions ϕ. Here are some examples:

- Say we have a pairwise MRF.

$$p(\boldsymbol{x}) = \frac{1}{Z}\prod_{ij\in E}\phi_{ij}(x_i,x_j)\prod_i \phi_i(x_i) \tag{2}$$

  Then we may want to assume that ϕ_{ij} is the same for all pairs. Thus, there is some $\theta(x_i,x_j)$ such that:

$$\phi_{ij}(x_i,x_j) = e^{\theta(x_i,x_j)} \tag{3}$$

- We could make a stronger assumption that ϕ_{ij} is not only the same for all edges but also has a specific structure. For example, we may assume that:

$$\phi_{ij}(x_i,x_j) = e^{|x_i-x_j|\theta} \tag{4}$$

  For some parameter θ. Alternatively, we can assume that ϕ_{ij} have the above structure but θ does depend in ij so that

$$\phi_{ij}(x_i,x_j) = e^{|x_i-x_j|\theta_{ij}} \tag{5}$$

- In a Bayesian network with binary variables say we have model $p(x_i|x_{Pa(i)})$ as:

$$p(x_i|x_{Pa(i)}) \propto e^{x_i\sum_{k\in Pa(i)}x_k\theta_{ki}} \tag{6}$$

  This is known as a logistic function. It says that the probability that $x_i = 1$ is a function of a linear combination of its parents (or alternatively that $\log\frac{p(x_i=1|x_{Pa(i)})}{p(x_i=0|x_{Pa(i)})}$ is a linear combination of the parents).

In all these cases, we assumed that there exists a set of parameters $\boldsymbol{\theta}$ and that the parameters of the graphical model depends on these. To make this explicit, we shall write $p(\boldsymbol{x};\theta)$ to denote a graphical model whose parameters depend on the parameters θ. The simplest case here is when θ are exactly the parameters of the graphical model. For example $\phi_c(x_c) = e^{\theta_c(x_c)}$ or $p(x_i|x_{Pa(i)}) = \theta(x_i|x_{Pa(i)})$. We shall call this a full or saturated parameterization.

Our goal will be to learn the parameters θ from data. We shall define the likelihood of a point $\boldsymbol{x}$ according to the model with parameters $\boldsymbol{\theta}$ by $p(\boldsymbol{x};\boldsymbol{\theta})$.

2 Learning from Data

When we talk about learning we usually mean learning from data. Assume we are interested in modeling the joint distribution of a set variables $\boldsymbol{x} = x_1,\ldots,x_n$ and we would like to do this with a graphical model $p(\boldsymbol{x})$. The true distribution that generates $\boldsymbol{x}$ is very likely not a graphical model of the type we use to model, so we denote it by $p^*(\boldsymbol{x})$. If there exists a θ^* such that $p^*(\boldsymbol{x}) = p(\boldsymbol{x};\theta^*)$ then p^* is a graphical model of the type we use for learning. In this case, there is hope of finding the exact true distribution (since it is in the set of distributions we are searching over).

Typically, we assume we have an IID sample from p^*. Denote this sample by $\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(M)}$. We want to use this sample to learn parameters $\boldsymbol{\theta}$. How should we do that? You should remember that there is no single answer to this question or one method that is always better than others. The main problem is that we want to be close to p^* but we only see samples from it.

2.1 Maximum Likelihood

The method of maximum likelihood is a popular approach to estimating parameters and indeed it has some nice desirable theoretical properties. We shall focus mostly on it in the next classes. It is defined as follows. Define the likelihood function as the following function of the parameter $\boldsymbol{\theta}$ and the data $\mathcal{D}$:

$$\ell(\boldsymbol{\theta};\mathcal{D}) = \frac{1}{M}\log\{\prod_m p(\boldsymbol{x}^{(m)};\boldsymbol{\theta})\} = \frac{1}{M}\sum_m \log p(\boldsymbol{x}^{(m)};\boldsymbol{\theta}) \tag{7}$$

The maximum likelihood estimator is then:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta};\mathcal{D}) \tag{8}$$

Why is this a sensible estimate? There are many reasons. We will show that in a sense it finds the closest distribution of the type $p(\boldsymbol{x};\boldsymbol{\theta})$ to our empirical data. First define the following divergence measure between distributions (called the Kullback Leibler divergence):

$$D_{KL}[p|q] = \sum_x p(x)\log\frac{p(x)}{q(x)} \tag{9}$$

It is easy to show that the above is always non-negative and is zero if and only if $p(x) = q(x)$ for all x.

Define the following empirical distribution:

$$p_D(\boldsymbol{x}) = \frac{n_{\boldsymbol{x}}}{M} \tag{10}$$

where $n_{\boldsymbol{x}}$ is the number of times that the assignment $\boldsymbol{x}$ was observed in the data.

Lets write the likelihood using this distribution:

$$\begin{aligned}\ell(\boldsymbol{\theta};\mathcal{D}) &= \frac{1}{M}\sum_m \log p(\boldsymbol{x}^{(m)};\boldsymbol{\theta}) = \frac{1}{M}\sum_{\boldsymbol{x}} n_{\boldsymbol{x}} \log p(\boldsymbol{x};\boldsymbol{\theta}) \\ &= \sum_{\boldsymbol{x}} p_D(\boldsymbol{x}) \log p(\boldsymbol{x};\boldsymbol{\theta}) = -D_{KL}[p_D(\boldsymbol{x})|p(\boldsymbol{x};\boldsymbol{\theta})] + \sum_{\boldsymbol{x}} p_D(\boldsymbol{x}) \log p_D(\boldsymbol{x}) \\ &= -D_{KL}[p_D(\boldsymbol{x})|p(\boldsymbol{x};\boldsymbol{\theta})] - H[p_D]\end{aligned}$$

We can thus interpret maximum likelihood as:

$$\hat{\boldsymbol{\theta}} = \arg\min D_{KL}[p_D(\boldsymbol{x})|p(\boldsymbol{x};\boldsymbol{\theta})] \tag{11}$$

It can thus be understood as finding $\boldsymbol{\theta}$ such that $p(\boldsymbol{x};\boldsymbol{\theta})$ is closest to the empirical distribution.

What can be said about the "correctness" of ML? Assume that the data was generated by a distribution $p(\boldsymbol{x};\boldsymbol{\theta}^*)$. As $n \to \infty$ we have that $p_D(\boldsymbol{x}) \to p(\boldsymbol{x};\boldsymbol{\theta}^*)$. So we will be able to find $\hat{\boldsymbol{\theta}}$ such that $D_{KL}[p(\boldsymbol{x};\hat{\boldsymbol{\theta}}^*)|p(\boldsymbol{x};\hat{\boldsymbol{\theta}})] = 0$. That is, we will learn the correct distribution. The parameter $\hat{\boldsymbol{\theta}}$ itself may not equal $\boldsymbol{\theta}^*$, since $\boldsymbol{\theta}^*$ may not be unique.

So the general scheme of ML estimation is to:

- Construct a parametric model of your distribution $p(\boldsymbol{x};\boldsymbol{\theta})$. This means that if you have parameters $\boldsymbol{\theta}$ this lets you construct the full distribution $p(\boldsymbol{x};\boldsymbol{\theta})$.
- Given data $\mathcal{D}$ find $\boldsymbol{\theta}$ that maximize the likelihood $\ell(\boldsymbol{\theta};\mathcal{D})$.

3 Learning Parameters of Bayesian Networks

Say we have a Bayesian network whose CPDs are parameterized by $\boldsymbol{\theta}$:

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \prod_i p(x_i|x_{Pa(i)};\boldsymbol{\theta}^{(i)}) \tag{12}$$

In other words each conditional distribution is defined using a different set of parameters. Let us assume that this parameterization is full, i.e. we have parameters $\theta^{(i)}(x_i|x_{Pa(i)})$ for every assignment $x_i, x_{Pa(i)}$ and:

$$p(x_i|x_{Pa(i)};\boldsymbol{\theta}^{(i)}) = \theta^{(i)}(x_i|x_{Pa(i)}) \tag{13}$$

Where we have the following constraints on the parameters for the distribution to be valid:

$$\begin{aligned}\theta^{(i)}(x_i|x_{Pa(i)}) &\geq 0 \\ \sum_{x_i}\theta^{(i)}(x_i|x_{Pa(i)}) &= 1.\end{aligned}$$

Let us now find the maximum likelihood estimate for these parameters.

$$\ell(\boldsymbol{\theta};\mathcal{D}) = \sum_{\boldsymbol{x}} p_D(\boldsymbol{x}) \log p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} p_D(\boldsymbol{x}) \sum_i \log p(x_i|x_{Pa(i)};\boldsymbol{\theta}^{(i)}) \tag{14}$$

now plug in our specific parametrization of the CPDs:

$$\ell(\boldsymbol{\theta};\mathcal{D}) = \sum_{\boldsymbol{x}} p_D(\boldsymbol{x}) \sum_i \log \theta^{(i)}(x_i|x_{Pa(i)}) \tag{15}$$

Switch the summation order to get:

$$\ell(\boldsymbol{\theta};\mathcal{D}) = \sum_i \sum_{\boldsymbol{x}} p_D(\boldsymbol{x}) \log \theta^{(i)}(x_i|x_{Pa(i)}) = \sum_i \sum_{x_i, x_{Pa(i)}} p_D(x_i, x_{Pa(i)}) \log \theta^{(i)}(x_i|x_{Pa(i)}) \tag{16}$$

Note that the likelihood only depends on empirical marginals over subsets of $x_i, x_{Pa(i)}$. These are known as sufficient statistics (since they are functions of the data $\mathcal{D}$ that are sufficient for estimation of the parameter $\boldsymbol{\theta}$).

Let us now try to find the maximum likelihood parameters. We want to maximize over possible parameters $\boldsymbol{\theta}$. Remember that we have the extra constraints on $\boldsymbol{\theta}$ as in Eq. 14.

For now, we will ignore the non-negativity constraints. Specifically, we will maximize over θ without this constraint, and see that the optimum will satisfy them automatically. To maximize over $\ell(\boldsymbol{\theta};\mathcal{D})$ with an equality constraint (Eq. 14) we define the Lagrangian:

$$\mathcal{L}(\boldsymbol{\theta},\lambda) = \ell(\boldsymbol{\theta};\mathcal{D}) - \sum_i \sum_{x_{Pa(i)}} \lambda_{i,x_{Pa(i)}} \left[\sum_{x_i} \theta^{(i)}(x_i|x_{Pa(i)}) - 1 \right] \tag{17}$$

The optimal pair $\lambda, \boldsymbol{\theta}$ are the ones where the above Lagrangian has zero gradient. Derive w.r.t. a particular parameter $\theta^{(i)}(x_i|x_{Pa(i)})$ to get:

$$\begin{aligned} \frac{p_D(x_i, x_{Pa(i)})}{\theta^{(i)}(x_i|x_{Pa(i)})} - \lambda_{i,x_{Pa(i)}} &= 0 \\ \theta^{(i)}(x_i|x_{Pa(i)}) &= \frac{1}{\lambda_{i,x_{Pa(i)}}} p_D(x_i, x_{Pa(i)}) \end{aligned}$$

We now need to add the constraint that $\theta^{(i)}(x_i|x_{Pa(i)})$ normalizes to 1, to find $\lambda_{i,x_{Pa(i)}}$.

$$1 = \sum_{x_i} \theta^{(i)}(x_i|x_{Pa(i)}) = \frac{1}{\lambda_{i,x_{Pa(i)}}} \sum_{x_i} p_D(x_i, x_{Pa(i)}) = \frac{1}{\lambda_{i,x_{Pa(i)}}} p_D(x_{Pa(i)})$$

From which we conclude:

$$\lambda_{i,x_{Pa(i)}} = p_D(x_{Pa(i)}) \tag{18}$$

So altogether we have:

$$\theta^{(i)}(x_i|x_{Pa(i)}) = \frac{p_D(x_i, x_{Pa(i)})}{p_D(x_{Pa(i)})} = p_D(x_i|x_{Pa(i)}) \tag{19}$$

This makes perfect sense! The ML parameters are obtained by calculating the corresponding conditional distribution from the empirical data. Note that we obtain non-negative parameters, so it's ok that we did not optimize over that constraint.

We note that this is only true for the full parameterization. When the parameterization is different the updates are not as simple. You will see an example in the recitation.

4 Learning in Markov Networks

Say we have a Markov network that is parameterized in the following way: $\phi_c(x_c;\boldsymbol{\theta}) = e^{\theta_c(x_c)}$. In other words, for every value of c, x_c we have a different parameter $\theta_c(x_c)$. The distribution is thus:

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_c \phi_c(x_c;\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_c e^{\theta_c(x_c)} = \frac{1}{Z(\boldsymbol{\theta})} e^{\sum_c \theta_c(x_c)} \tag{20}$$

where $Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} e^{\sum_c \theta_c(x_c)}$. The above is a full parameterization of the Markov net, since it fully specifies all the parameters of the network.

What is the maximum likelihood estimator in this case? Write the likelihood as:

$$\begin{aligned}\ell(\boldsymbol{\theta}; \mathcal{D}) &= a \sum_{\boldsymbol{x}} p_D(\boldsymbol{x}) \log p(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{\boldsymbol{x}} p_D(\boldsymbol{x}) \left[-\log Z(\boldsymbol{\theta}) + \sum_c \theta_c(x_c) \right] \\ &= -\log Z(\boldsymbol{\theta}) + \sum_c \sum_{\boldsymbol{x}} p_D(\boldsymbol{x}) \theta_c(x_c) = -\log Z(\boldsymbol{\theta}) + \sum_c \sum_{x_c} p_D(x_c) \theta_c(x_c)\end{aligned}$$

Again we see that the likelihood only depends on marginals of the empirical distribution, namely on the marginals $p_D(x_c)$. It turns out that the above is a concave function of $\boldsymbol{\theta}$. This is a result of $Z(\boldsymbol{\theta})$ being a convex function of $\boldsymbol{\theta}$ (calculate its Hessian to see why).

To maximize the likelihood we can use gradient descent, assuming we can efficiently calculate the gradient. In some cases we cannot because the gradient involves the marginals (see below), but we can still use approximate inference algorithms for these, resulting in an approximate gradient. A "cleaner" way of approaching this is to replace the partition function $Z(\boldsymbol{\theta})$ with some approximation for which we can calculate gradients.

In what follows we characterize the optimal parameters, since this will give us some useful intuitions. The maximum likelihood $\hat{\boldsymbol{\theta}}$ will be the parameter such that $\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathcal{D}) = 0$. This is a set of equations, one for each parameter $\theta_c(x_c)$. To get the equation for $\theta_c(x_c)$ we take the derivative of the likelihood w.r.t. this parameter. Lets first get the derivative of the log partition function:

$$\frac{\partial \log Z(\boldsymbol{\theta})}{\partial \theta_c(x_c)} = \frac{1}{Z(\boldsymbol{\theta})} \sum_{\boldsymbol{x}' : x'_c = x_c} e^{\sum_c \theta_c(x'_c)} = \sum_{\boldsymbol{x}' : x'_c = x_c} p(\boldsymbol{x}'; \boldsymbol{\theta}) = p(x_c; \boldsymbol{\theta}) \tag{21}$$

Now taking the derivative of the likelihood in Eq. 21 we get:

$$\frac{\partial \ell(\boldsymbol{\theta}; \mathcal{D})}{\theta_c(x_c)} = p_D(x_c) - p(x_c; \boldsymbol{\theta}) \tag{22}$$

Equating to zero we get:

$$\boxed{p_D(x_c) = p(x_c; \boldsymbol{\theta})} \tag{23}$$

The maximum likelihood estimator needs to satisfy the above. Unfortunately, there is no general closed form solution for $\boldsymbol{\theta}$. But, it gives good intuition about what the ML estimate does. Namely, $\hat{\boldsymbol{\theta}}$ yields a model $p(\boldsymbol{x}; \hat{\boldsymbol{\theta}})$ whose marginals over x_c are identical to those of the empirical distribution.

In some cases there is a closed form estimate. For example, when the sets c are pairwise and correspond to edges and nodes of a tree graph G. Lets consider an even simple example. Say your model is:

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} e^{\sum_i \theta_i(x_i)} = \frac{1}{Z(\boldsymbol{\theta})} \prod_i e^{\theta_i(x_i)} \tag{24}$$

The ML equations say we want the marginals of the above to equal $p_D(x_i)$. It's easy to see that this is accomplished if we choose:

$$\hat{\theta}_i(x_i) = \log p_D(x_i) \tag{25}$$

since then:

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \prod_i p_D(x_i) \tag{26}$$

This is a valid (normalized) distribution with the desired marginals.

A more complicated case is where we have:

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} e^{\sum_i \theta_i(x_i) + \sum_{ij \in E} \theta_{ij}(x_i, x_j)} \tag{27}$$

where E is a tree graph. It can be shown in this case that the optimal parameters are:

$$\begin{aligned}\hat{\theta}_i(x_i) &= \log p_D(x_i) \\ \hat{\theta}_i(x_i, x_j) &= \log \frac{p_D(x_i, x_j)}{p_D(x_i)p_D(x_j)}\end{aligned}$$

An alternative parameterization that results in the same model is:

$$\begin{aligned}\hat{\theta}_i(x_i) &= (1 - d_i) \log p_D(x_i) \\ \hat{\theta}_i(x_i, x_j) &= \log p_D(x_i, x_j)\end{aligned}$$