20 Learning Graphical Models

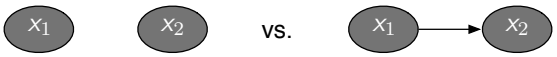
In this course, we first saw how distributions could be represented using graphs. Next, we saw how to perform statistical inference efficiently in these distributions by exploiting the graph structure. We now begin the third and final phase of the course, on learning graphical models from data.

In many situations, we can avoid learning entirely because the right model follows from physical constraints. For instance, the structure and parameters may be dictated by laws of nature, such as Newtonian mechanics. Alternatively, we may be analyzing a system, such as an error correcting colde, which was specifically engineered to have certain probabilistic dependencies. However, when neither of these applies, we can't set the parameters a priori, and instead we must learn them from data. The data we use for learning the model is known as "training data."

The learning task of graphical model would involve learning the graphical structure and associated parameters (or potentials). The training data could be about all variables or only partial observations. Given this, the following is a natural sets of learning scenarios:

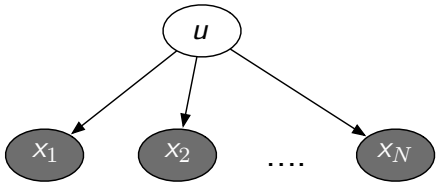
1. The graphical structure is known, but parameters are unknown. The observations available to learn the parameters involve all variables.

   Example. If we have to variables x_1 and x_2, we may be interested in determining whether they are independent. This is equivalent to determining whether or not there is an edge between x_1 and x_2. This is equivalent to determini al or not.

2. The graphical structure is known, but parameters are unknown. The observations available to learn the parameters involve only a subset (partial) of variables, while other are unobserved (hidden).

   Example. The naive Bayes model, as shown in the following figure, is often used for clustering. In it, we assume every observation $\mathbf{x} = (x_1, \ldots, x_N)$ is generated by first choosing a cluster assignment u, and then independently sampling each x_i from some distribution $p_{x_i|u}$. We observe $\mathbf{x}$, but not u. And we do *know* the graphical structure.



3. Both graphical structure and parameters are unknown. The observations available to learn the parameters involve all variables.

   Example. We are given a collection of image data for hand-written characters/alphabets. The goal is to use it to product graphical model for each character so as to utilize it for automated character recognition: given an unknown image of a character, find the probability of it being a particular character, say c, using the graphical model; declare it to be character $c*$ for which the probability is maximum. Now at gray-scale, image for a given alphabet can be viewed as a graphical model with each pixel representing binary variables. The data is available about all pixels/variables. But graphical structure and associated parameters are unknown.

4. Both graphical structure and parameters are unknown. The observations available to learn the parameters involve only a subset (partial) of variables, while other are unobserved (hidden).

   Example. Let $\mathbf{x} = (x_1, \ldots, x_N)$ denote prices of stocks at a given time and variables $\mathbf{y} = (y_1, \ldots, y_M)$ capture the related un-observable ambient state of the world, e.g. internal information about business operation, etc. We only observe $\mathbf{x}$. The structure of graphical model between variables $\mathbf{x}, \mathbf{y}$ is entirely unknown as well as $\mathbf{y}$ are unobserved. The goal is to learn the graphical model and associated parameters between these variables to potentially better predict future price variation or develop better portfolio of investment.

We shall spend remainder of the lectures dealing with each of these scenarios, one at a time. For each of these scenarios, we shall distinguish cases of directed and undirected graphical models. As we discuss learning algorithms, we'll find ourselves using inference algorithms, that we have learnt thus far, utilizing in an unusual manner.

20.1 Learning Simplest Graphical Model: One Node

We shall start with the question of learning simplest possible graphical model: graph with one node. As we shall see, it will form back-bone of our first learning task in general - known graph structure, unknown parameters and all variables are observed.

Let x be one node graphical model, or simply a random variable. Let $p(x;\theta)$ be a distribution over a random variable x parameterized by θ. For instance, $p(x;\theta)$ may be a Bernoulli distribution where $\mathsf{x}=1$ with probability θ and $\mathsf{x}=0$ with probability $1-\theta$. We treat the parameter θ as nonrandom, but unknown. We can view $p(x;\theta)$ as a function of x or θ:

- $p(\cdot;\theta)$ is the pdf or pmf for a distribution parameterized by θ,
- $L(\cdot;x) \triangleq p(x;\cdot)$ is the *likelihood function* for a given observation x.

We are given observations in terms of S samples, $\mathcal{D} = \{x(1),\ldots,x(S)\}$. The goal is to learn parameter θ.

A reasonable philosophy for learning unknown parameter given observations is to use *maximum likelihood estimation*: it is the unknown parameter with respect to which the observations are most likely or have maximum likelihood.

That is, if we have one observation $x = x(1)$, then

$$\hat{\theta}_{ML}(x) = \arg\max_{\theta} p(x;\theta).$$

We shall assume that when multiple observations are present, they are generated independently (and from identical unknown distribution). Therefore, the likelihood function for $\mathcal{D} = \{x(1),\ldots,x(S)\}$ is given by

$$\begin{aligned} L(\theta;\mathcal{D}) &\equiv L(\theta;x(1),\ldots,x(S)) \\ &= \prod_{s=1}^{S} p(x(s);\theta). \end{aligned} \tag{1}$$

Often we find it easier to work with the log of the likelihood function, or log-likelihood:

$$\begin{aligned} \ell(\theta;\mathcal{D}) &\equiv \ell(\theta;x(1),\ldots,x(S)) \\ &\triangleq \log L(\theta;x(1),\ldots,x(S)) \qquad (2) \\ &= \sum_{s=1}^{S} \log p(x(s);\theta). \qquad (3) \end{aligned}$$

Observe that the likelihood function is invariant to permutation of the data, because we assumed the observations were all drawn i.i.d. from $p(\cdot;\theta)$. This suggests we should pick a representation of the data which throws away the order. One such representation is the *empirical distribution* of the data, which gives the relative frequency of different symbols in the alphabet:

$$\hat{p}(a) = \frac{1}{S}\sum_{s=1}^{S} 1_{x(s)=a}.$$

Using short-hand $\mathcal{D} = \{x(1),\ldots,x(S)\}$, $\hat{p}_{\mathcal{D}}(a)$ is another way of writing the empirical distribution. We can also write the log-likelihood function as $\ell(\cdot;\mathcal{D})$. Note that $\hat{p}$ is a vector of length $|\mathcal{X}|$. We can think of it as a *sufficient statistic* of the data, since it is sufficient to infer the maximum likelihood parameters.

20.1.1 Bernoulli variable

Going back to Bernoulli distribution (or coin flips), $\hat{p}_{\mathcal{D}}(0)$ is the empirical fraction of 0's in a sample, and $\hat{p}_{\mathcal{D}}(1)$ is the empirical fraction of 1's. The likelihood function is given by

$$\ell(\theta; \mathcal{D}) = S\hat{p}(1) \log \theta + S\hat{p}(0) \log(1 - \theta).$$

We find the maximum of this function by differentiating and setting the derivative to zero:

$$\begin{aligned} 0 &= \frac{\partial \ell}{\partial \theta} \\ &= S\frac{\hat{p}_{\mathcal{D}}(1)}{\theta} - S\frac{\hat{p}_{\mathcal{D}}(0)}{(1-\theta)}. \end{aligned}$$

Since $\hat{p}_{\mathcal{D}}(0) = 1 - \hat{p}_{\mathcal{D}}(1)$, we have

$$\frac{\hat{p}_{\mathcal{D}}(1)}{1 - \hat{p}_{\mathcal{D}}(1)} = \frac{\theta}{1-\theta}.$$

Therefore, we obtain

$$\hat{\theta}_{ML} = \hat{p}_{\mathcal{D}}(1).$$

That is, the empirical estimation is the maximum likelihood estimation. By the strong law of large numbers, the empirical distribution will eventually converge to the correct probabilities, so we see that maximum likelihood is consistent in this case.

20.1.2 General discrete variable

When the variables take values in a general discrete alphabet $\mathcal{X} = \{a_1, \ldots, a_M\}$ rather than $\{0, 1\}$, we use the multinomial distribution rather than the Bernoulli distribution. In other words, $p(x = a_m) = \theta_m$, where the parameters θ_m satisfy:

$$\begin{aligned} \sum_{m=1}^{M} \theta_m &= 1, \\ \theta_m &\in [0, 1]. \end{aligned}$$

The likelihood function is given by

$$L(\boldsymbol{\theta}; \mathcal{D}) = \prod_m \theta_m^{S\hat{p}_{\mathcal{D}}(a_m)}.$$

Following exactly the same reasoning as in the Bernoulli case, we find that the maximum likelihood estimation of parameters is given by empirical distribution, i.e. $\hat{\theta}_{ML} = (\hat{p}_{\mathcal{D}}(a_m))_{1 \leq m \leq M}$.

20.1.3 Information theoretic interpretation

Earlier in the course, we saw that we could perform approximate inference in graphical models by solving a variational problem minimizing information divergence between the true distribution p and an approximating distribution q. It turns out that maximum likelihood has an analogous interpretation. We can rewrite the log-likelihood function in terms of information theoretic quantities:

$$\begin{aligned}
\ell(\boldsymbol{\theta}; \mathcal{D}) &= \sum_{s=1}^{S} \log p(x(s); \boldsymbol{\theta}) \\
&= S \sum_{a \in \mathcal{X}} \hat{p}_{\mathcal{D}}(a) \log p(a; \boldsymbol{\theta}) \\
&= S E_{\hat{p}_{\mathcal{D}}}[\log p(x; \boldsymbol{\theta})] \\
&= S \left(H(\hat{p}_{\mathcal{D}}) - D(\hat{p}_{\mathcal{D}} \| p(\cdot; \boldsymbol{\theta})) \right)
\end{aligned}$$

We can ignore the entropy term because it is a function of the empirical distribution, and therefore fixed. Therefore, maximizing the likelihood is equivalent to minimizing the information divergence $D(\hat{p}_{\mathcal{D}} \| p(\cdot; \boldsymbol{\theta}))$. Note the difference between this variational problem and the one we considered in our discussion of approximate inference: there we were minimizing $D(q\|p)$ with respect to q, whereas here we are minimizing with respect to the second argument.

Recall that KL divergence is zero when the two arguments are the same distribution. In the multinomial case, since we are optimizing over the set of all distributions over a finite alphabet $\mathcal{X}$, we can match the distribution exactly, i.e. set $p(\cdot; \boldsymbol{\theta}) = \hat{p}_{\mathcal{D}}$. However, in most interesting problems, we cannot match the data distribution exactly. (If we could, we would simply be overfitting.) Instead, we generally optimize over a restricted class of distributions parameterized by $\boldsymbol{\theta}$.

20.2 Learning Parameters for Directed Graphical Model

Now we discuss how the parameter estimation for one-node graphical model extends neatly for learning parameters of a generic directed graphical model.

Let us start with an example of a directed graphical model as shown in Figure 1. This graphical model obeys factorization

$$p(x_1, \ldots, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2, x_3).$$

The joint distribution can be represented with four sets of parameters $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_4$. Each $\boldsymbol{\theta}_i$ defines a multinomial distribution corresponding to each joint assignment for the parents of x_i. Suppose we have S samples of the 4-tuple $(\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3, \mathsf{x}_4)$. The log-likelihood function can be written as

$$\ell(\mathbf{x}; \boldsymbol{\theta}) = \log p(x_1; \boldsymbol{\theta}_1) + \log p(x_2|x_1; \boldsymbol{\theta}_2) + \log p(x_3|x_1; \boldsymbol{\theta}_3) + \log p(x_4|x_2, x_3; \boldsymbol{\theta}_4).$$

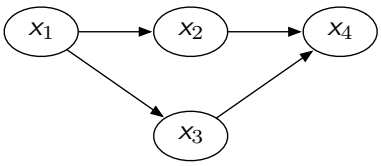


Figure 1

Observe that each $\boldsymbol{\theta}_i$ appears in exactly one of these terms, so all of the terms can be optimized separately. Since each individual term is simply the likelihood function of a multinomial distribution, we simply plug in the empirical distributions as before. Precisely, in the above example, we compute empirical conditional distribution for each variable x_i with respect to it's parents, e.g. $\hat{p}_{\mathsf{x}_2|\mathsf{x}_1}(\cdot|x_1)$ for all possible values of x_1. And this will be the maximum likelihood estimation of the directed graphical model!

Remarks. It is worth highlighting two features of this problem which were necessary for the maximum likelihood problem to decompose into independent subproblems. First, we had complete observations, i.e. all of the observation tuples gave the values of all of the variables. Second, the parameters were treated as independent for each conditional probability distribution (CPD). If either of these assumptions is violated, the maximum likelihood problem does not decompose into an independent subproblem for each variable.

20.3 Bayesian parameter estimation

Here we discuss a method that utilizes *prior* information about unknown parameters to learn them from data. It can also be viewed as a way to avoid the so called *overfitting* by means of *penalty*. Again, as discussed above, this method can be naturally extended to learning directed graphical model as long as the priors over parameters corresponding to different variables are independent.

Recall that in maximum likelihood parameter estimation, we modeled the parameter $\boldsymbol{\theta}$ as nonradom, but unknown. In *Bayesian parameter estimation*, we model $\boldsymbol{\theta}$ as a random variable. We assume we are given a prior distribution $p(\boldsymbol{\theta})$ as well as a distribution $p(x|\boldsymbol{\theta})$ corresponding to the likelihood function. Notice that we use the notation $p(x|\boldsymbol{\theta})$ rather than $p(x;\boldsymbol{\theta})$ because $\boldsymbol{\theta}$ is a random variable. The probability of the data x is therefore given by

$$p(\mathbf{x}) = \int p(\boldsymbol{\theta})p(\mathbf{x}|\theta)d\boldsymbol{\theta}.$$

In ML estimation, we tried to find a particular value $\boldsymbol{\theta}$ to use. In Bayesian estimation, we instead use a weighted mixture of all possible values, where the weights

correspond to the *posterior distribution*

$$p(\boldsymbol{\theta}|\mathbf{x}(1),\ldots,\mathbf{x}(S)).$$

In other words, in order to make predictions about future samples, we use the *predictive distribution*

$$p(\mathbf{x}'|\mathbf{x}(1),\ldots,\mathbf{x}(S)) = \int p(\boldsymbol{\theta}|\mathbf{x}(1),\ldots,\mathbf{x}(S))p(\mathbf{x}'|\boldsymbol{\theta})d\boldsymbol{\theta}.$$

For an arbitrary choice of prior, it is quite cumbersome to represent the posterior distribution and to compute this integral. However, for certain priors, called *conjugate priors*, the posterior takes the same form as the prior, making these computations convenient. For the case of the multinomial distribution, the conjugate prior is the Dirichlet prior. The parameter $\boldsymbol{\theta}$ is drawn from the Dirichlet distribution Dirichlet$(\alpha_1,\ldots,\alpha_M)$ if

$$p(\boldsymbol{\theta}) \propto \prod_{m=1}^{M} \boldsymbol{\theta}_m^{\alpha_m - 1}.$$

It turns out that if $\{x(1),\ldots,x(S)\}$ are i.i.d. samples from a multinomial distribution over alphabet size M, then if $\boldsymbol{\theta} \sim$ Dirichlet$(\alpha_1,\ldots,\alpha_M)$, the posterior is given by

$$\boldsymbol{\theta}|\mathcal{D} \sim \text{Dirichlet}(\alpha_1 + S\hat{p}_{\mathcal{D}}(a_1),\ldots,\alpha_M + S\hat{p}_{\mathcal{D}}(a_M)).$$

The predictive distribution also turns out to have a convenient form:

$$p(\mathbf{x}' = a_m|\mathbf{x}(1),\ldots,\mathbf{x}(S)) = \frac{\alpha_m + S\hat{p}_{\mathcal{D}}(a_m)}{\sum_{m=1}^{M}\alpha_m + S}.$$

In the special case of Bernoulli random variables,

$$\begin{aligned} \theta &\sim \frac{1}{Z}\theta^{\alpha_1 - 1}(1-\theta)^{\alpha_0 - 1} \\ \theta|\mathcal{D} &\sim \frac{1}{Z'}\theta^{\alpha_1 + S\hat{p}_{\mathcal{D}}(1) - 1}(1-\theta)^{\alpha_0 + S\hat{p}_{\mathcal{D}}(0) - 1}. \end{aligned}$$

The special case of the Dirichlet distribution for binary alphabets is also called the *beta distribution*.

In a fully Bayesian setting, we would make predictions about new data by integrating out θ. However, if we want to decide a single value of θ, we can use the MAP criterion:

$$\hat{\theta}_{MAP} = \arg\max_{\theta} p(\theta|\mathcal{D}).$$

When we differentiate and set the derivative equal to zero, we find that

$$\hat{\theta}_{MAP} = \frac{\alpha_1 + S\hat{p}(1) - 1}{\alpha_0 + \alpha_1 + S - 2}.$$

Observe that, unlike the maximum likelihood criterion, the MAP criterion takes the number of training examples S into account. In particular, when S is small, the MAP value is close to the prior; when S is large, the MAP value is close to the empirical distribution. In this sense, Bayesian parameter estimation can be seen as controlling overfitting by penalizing more complex models, i.e. those farther from the prior.

MIT OpenCourseWare
http://ocw.mit.edu

6.438 Algorithms for Inference

Fall 2014

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.