Chapter 10

Parameter Estimation in Graphical Models

10.1 Introduction

A graphical model has two components: the graph structure (the nodes and their connections), and the conditional probability distributions/potential functions, which are usually expressed in parametric form. In this chapter:

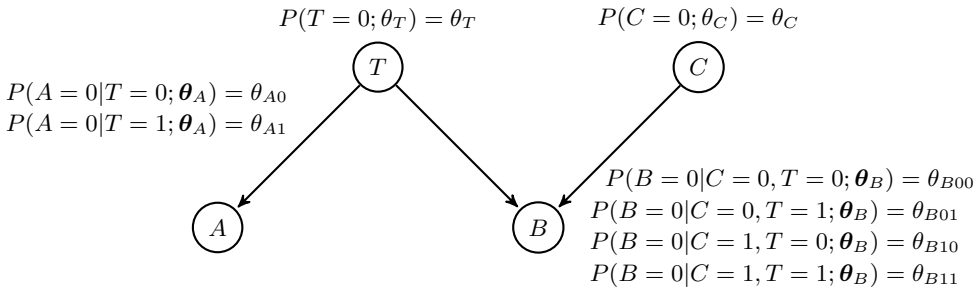
- We will consider the problem of estimating the parameters in graphical models. The problem is simpler in the case of Bayesian networks and for simplicity, that is were our attention will be focused.
- However, we will not consider the more challenging problem of learning the structure of a network. The best case scenario is that you have good reason to design a graph in a certain way, e.g., based on causality.

10.2 Maximum Likelihood Estimation in Bayesian Networks

Consider a BN with known graph with m nodes $x_1, \ldots, x_m$ in which the parameters of the conditional distribution are unknown. There are m conditional probability distributions (**CPD**s)[1], one for each node, and each of these has an unknown parameter vector. We denote the concatenated vector of all parameters as $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m)$. To determine the parameters, we collect a dataset $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ of n iid samples, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{im})$.

Example 94. As an example, we may consider the network from previous chapters with the vector of parameters $\boldsymbol{\theta} = (\theta_T, \theta_C, \boldsymbol{\theta}_A, \boldsymbol{\theta}_B)$.

[1] Some of the nodes do not have any parents so their distribution is not conditioned on any other nodes. We view these as conditioned on the empty set and thus refer to all probability distributions in a Bayesian Network as *conditional* probability distributions.



Our goal is to determine $\boldsymbol{\theta}$ by collecting data and determine the conditional probability distributions, thereby determining the network. To collect data, on n days, we record whether there is heavy traffic and whether Alice, Bob, and/or Charlie are late. △

We can find the parameters through maximum likelihood. Given that our network can have many nodes, the size of the parameter vector may be very large. This would create computational difficulties since it would require maximizing a function of many variables. Fortunately, in the case of Bayesian networks, the problem decomposes to estimating the parameters for each nodes separately as we will show. To see why this is helpful, suppose that we optimize by grid search, i.e., trying a set of values at regular intervals. If we try K points for one dimension, for m dimensions we need to try K^m points to get the same precision. However, if we optimize m parameters separately, then we only need to try mK points, typically a significantly smaller number.

Decomposability of likelihood. For the ith data sample, the likelihood function is

$$p(\boldsymbol{x}_i;\boldsymbol{\theta}) = \prod_{j=1}^{m} p(x_{ij}|\operatorname{pa}(x_{ij});\boldsymbol{\theta}_j)$$

and thus the log-likelihood of the whole dataset is

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ln p(\boldsymbol{x}_i;\boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{j=1}^{m} \ln p(x_{ij}|\operatorname{pa}(x_{ij});\boldsymbol{\theta}_j) = \sum_{j=1}^{m}\sum_{i=1}^{n} \ln p(x_{ij}|\operatorname{pa}(x_{ij});\boldsymbol{\theta}_j).$$

Thus for a given j, $\boldsymbol{\theta}_j$ only appears in the term $\sum_{i=1}^{n} \ln p(x_{ij}|\operatorname{pa}(x_{ij});\boldsymbol{\theta}_j)$ and no other θ_k appears in this term. So each $\boldsymbol{\theta}_j$, and thus each conditional probability distribution, can be learned independently of the others.

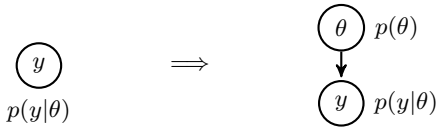
Exercise 95. For the TABC network above, what would our data look like? What is the ML estimate for each parameter based on this data?

10.3 Bayesian Parameter Estimation in Bayesian Networks

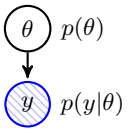
Suppose that we want to estimate the parameters of the conditional probability distributions of a Bayesian network using Bayesian inference. Since in the Bayesian view, parameters are considered random, we can augment the Bayesian network by adding the parameters as nodes. In particular, we can recast Bayesian estimation problems that we have seen before as Bayesian networks.

10.3.1 Bayesian Estimation formulated as Bayesian Networks

Example 96. As a simple example, consider the Bayesian network consisting of a singe node y whose distribution has an unknown parameter θ. We can transform this to a network with node θ and y, in which the conditional distributions are the prior $p(\theta)$ and the likelihood $p(y|\theta)$.

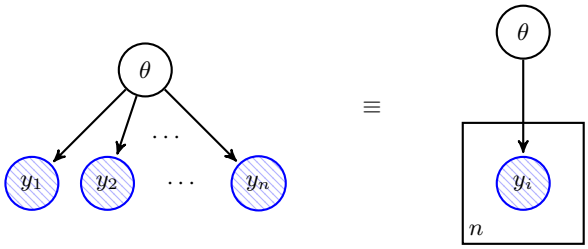


The joint distribution resulting from the network is $p(\theta, y) = p(\theta)p(y|\theta)$, which indeed factorizes with respect to the network on the right. Now, if y is given (which we show by a hatched pattern), we can find $p(\theta|y)$ using Bayes rule,



$$p(\theta|y) = \frac{p(y,\theta)}{p(y)} = \frac{p(\theta)p(y|\theta)}{p(y)}$$

Example 97. The problem in Example 96 becomes more interesting when we have n independent samples, $\mathcal{D} = \{y_1, y_2, \ldots, y_n\}$, from the distribution. We can simplify the network with the plate notation, by representing nodes that have the same conditional probability distribution (and are independent) using *plates*, as shown below.



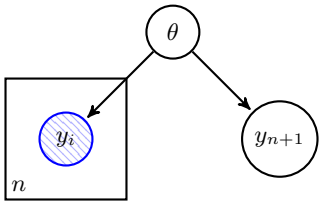
The joint distribution of θ and y_1^n can be written as

$$p(y_1^n, \theta) = p(\theta) \prod_{i=1}^{n} p(y_i|\theta),$$

and the posterior distribution for θ as

$$p(\theta|y_1^n) \propto p(y_1^n, \theta) = p(\theta) \prod_{i=1}^{n} p(y_i|\theta).$$

Example 98. Following Example 97, suppose we have n independent samples $\mathcal{D} = \{y_1, y_2, \ldots, y_n\}$ from the distribution. We want to predict the distribution of the next sample $p(y_{n+1}|y_1^n)$. The graph is shown below.



We have

$$p(y_{n+1}|y_1^n) = \int p(y_{n+1}, \theta|y_1^n)d\theta = \int p(\theta|y_1^n)p(y_{n+1}|\theta, y_1^n)d\theta = \int p(\theta|y_1^n)p(y_{n+1}|\theta)d\theta$$

where in the last step we have used $y_{n+1} \perp\!\!\!\perp y_1^n \mid \theta$, which follows from d-separation. Furthermore,

$$\mathbb{E}[y_{n+1}|y_1^n] = \mathbb{E}[\mathbb{E}[y_{n+1}|\theta, y_1^n]|y_1^n] = \mathbb{E}[\mathbb{E}[y_{n+1}|\theta]|y_1^n]. \tag{10.1}$$

Roughly speaking, to learn about y_{n+1} given y_1^n, we must first learn about θ since this is the node that connects y_1^n and y_{n+1}.

For example, assume $p(\theta) \propto 1$, $y_i|\theta \sim \text{Ber}(\theta)$, and that out of the n samples y_i, there s 1s and f 0s. Then

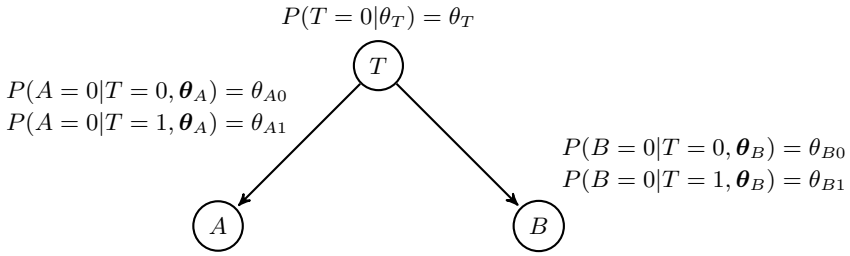
$$p(y_{n+1} = 1|y_1^n) = \mathbb{E}[y_{n+1}|y_1^n] = \mathbb{E}[\mathbb{E}[y_{n+1}|\theta]|y_1^n] = \mathbb{E}[\theta|y_1^n] = \frac{s+1}{s+f+2}.$$

10.3.2 Estimating Parameters of CPDs in Bayesian Networks

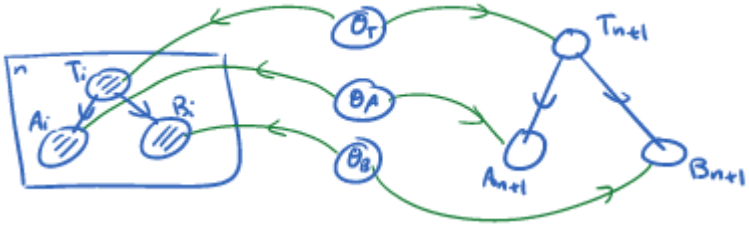
So far for the most part, we have cast Bayesian inference problems that we had seen before as Bayesian networks. In the next example, we consider the problem of estimating the parameters of the conditional probability distributions (**CPD**s) of Bayesian network.

Similar to Section 10.2, consider a BN with m nodes $x_1, \ldots, x_m$ in which the parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m)$ of the CPDs are unknown. Our dataset is $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ consisting of n iid samples, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{im})$. We are interested in determining $p(\boldsymbol{\theta}|\mathcal{D})$ and $p(\boldsymbol{x}_{n+1}|\mathcal{D})$.

Example 99. Let us consider a simpler version of the network given in Example 94, with unknown parameter vector $\boldsymbol{\theta} = (\theta_T, \boldsymbol{\theta}_A, \boldsymbol{\theta}_B)$,

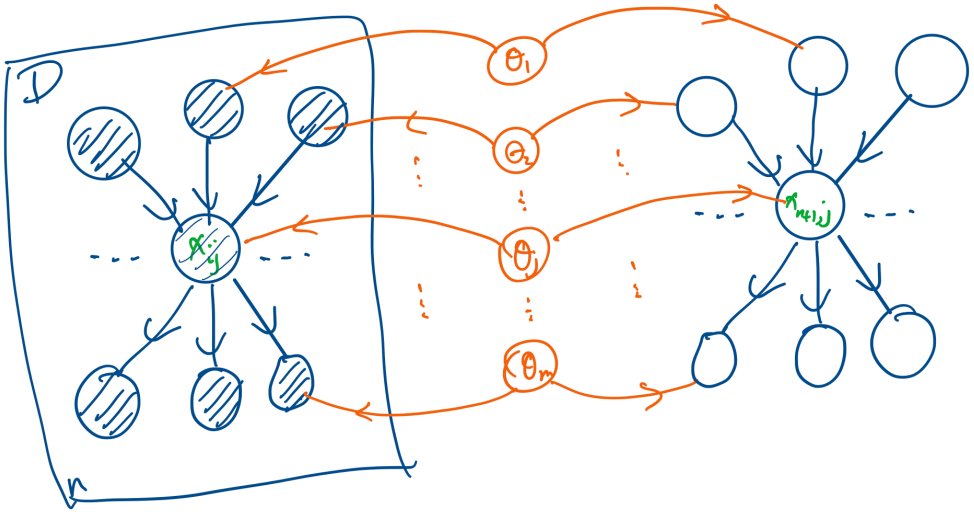


Given n samples $\mathcal{D} = \{(T_1, A_1, B_1), \ldots, (T_n, A_n, B_n)\}$, and our goal is to estimate the posterior we augment the network as



so that we can learn about $p(\boldsymbol{\theta}_A, \boldsymbol{\theta}_B, \theta_T|\mathcal{D})$ and $p(T_{n+1}, A_{n+1}, B_{n+1}|\mathcal{D})$. △

Decomposability of posterior and predictive posterior. Consider a Bayesian network with $n \times m$ nodes for the data $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{im})$; m nodes for $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m$; and m nodes for the future observation $x_{n+1,1}, \ldots, x_{n+1,m}$ as shown below (see also the second graph in Example 99 for a concrete example)



Let us start by trying to decompose $p(\boldsymbol{\theta}|\mathcal{D})$. First, note that by d-separation

$$p(\boldsymbol{\theta}|\mathcal{D}) = \prod_{j=1}^{m} p(\boldsymbol{\theta}_j|\mathcal{D}).$$

Next, define

$$N_j = \{x_{1j}, \ldots, x_{nj}, \text{pa}(x_{1j}), \ldots, \text{pa}(x_{nj})\}, \tag{10.2}$$

i.e., the set of children and parents of children of $\boldsymbol{\theta}_j$ among the nodes of $\mathcal{D}$. Similar to Markov blankets, we see that $\boldsymbol{\theta}_j \perp\!\!\!\perp \mathcal{D} \backslash N_j \mid N_j$. That is, given N_j, $\boldsymbol{\theta}_j$ is independent of all other nodes in $\mathcal{D}$, and so $p(\boldsymbol{\theta}_j|\mathcal{D}) = p(\boldsymbol{\theta}_j|N_j)$. Hence,

$$p(\boldsymbol{\theta}|\mathcal{D}) = \prod_{j=1}^{m} p(\boldsymbol{\theta}_j|\mathcal{D}) = \prod_{j=1}^{m} p(\boldsymbol{\theta}_j|N_j). \tag{10.3}$$

This is good news, because it means we can find the posterior for the parameters of each CPD can be computed separately.

Exercise 100. Using the Bayesian network above, prove that the last two equalities in the expression below hold:

$$p(\boldsymbol{\theta}, \boldsymbol{x}_{n+1} \mid \mathcal{D}) = p(\boldsymbol{\theta}|\mathcal{D})p(\boldsymbol{x}_{n+1} \mid \mathcal{D}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathcal{D})p(\boldsymbol{x}_{n+1}|\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathcal{D}) \prod_{j=1}^{m} p(x_{n+1,j}|\boldsymbol{\theta}_j, \text{pa}(x_{n+1,j})). \tag{10.4}$$

We can find the posterior predictive $p(\boldsymbol{x}_{n+1}|\mathcal{D})$ by integrating the above expression with respect to $\boldsymbol{\theta}$. △

Example 101. Getting back to Example 99, let us find $p(\boldsymbol{\theta}_A|\mathcal{D})$ and $p(A_{n+1}, B_{n+1}|\mathcal{D})$. As in (10.2), the set of children and parents of children of $\boldsymbol{\theta}_A$ among data nodes are $N_A = \{A_1, \ldots, A_n, T_1, \ldots, T_n\}$ and

$$p(\boldsymbol{\theta}_A|\mathcal{D}) = p(\boldsymbol{\theta}_A|A_1^n, T_1^n).$$

This makes intuitive sense: to estimate the probability of Alice being late as a function of traffic, only the part of data that deals with Alice's arrival time and traffic is relevant.

Assuming that the prior satisfies $p(\boldsymbol{\theta}_A) = p(\theta_{A0})p(\theta_{A1})$,

$$\begin{aligned}
p(\boldsymbol{\theta}_A|A_1^n, T_1^n) &\propto p(\boldsymbol{\theta}_A)p(T_1^n|\boldsymbol{\theta}_A)p(A_1^n|T_1^n, \boldsymbol{\theta}_A) \\
&= p(\boldsymbol{\theta}_A)p(T_1^n)p(A_1^n|T_1^n, \boldsymbol{\theta}_A) \\
&\propto p(\boldsymbol{\theta}_A)p(A_1^n|T_1^n, \boldsymbol{\theta}_A) \\
&= p(\boldsymbol{\theta}_A)\prod_{i=1}^{n} p(A_i|T_1^n, \boldsymbol{\theta}_A) \\
&= p(\boldsymbol{\theta}_A)\prod_{i=1}^{n} p(A_i|T_i, \boldsymbol{\theta}_A) \\
&= \left(p(\theta_{A0})\prod_{i:T_i=0} p(A_i|T_i=0, \theta_{A0})\right)\left(p(\theta_{A1})\prod_{i:T_i=1} p(A_i|T_i=1, \theta_{A1})\right).
\end{aligned}$$

Since the terms depending on θ_{A0} and θ_{A1} separate, they are conditionally independent and we can estimate them separately: Hence, the estimators of θ_A^0 and θ_A^1 are

$$\begin{aligned}
p(\theta_{A0}|\mathcal{D}) &\propto p(\theta_{A0})\prod_{i:T_i=0} p(A_i|T_i=0, \theta_{A0}), \\
p(\theta_{A1}|\mathcal{D}) &\propto p(\theta_{A1})\prod_{i:T_i=1} p(A_i|T_i=1, \theta_{A1}).
\end{aligned}$$

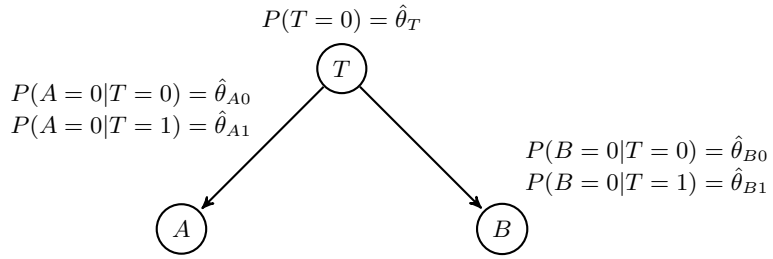
Suppose $p(\theta_A^0) \sim \text{Beta}(1,1)$ and out of 100 days with no traffic, in 40 days Alice was on time. Hence,

$$\theta_{A0}|\mathcal{D} \sim \text{Beta}(41, 61).$$

Furthermore, the posterior probability of the next sample (A_{n+1}, B_{n+1}) is

$$\begin{aligned}
p(A_{n+1}, B_{n+1}|\mathcal{D}) &= \int_{\boldsymbol{\theta}} p(A_{n+1}, B_{n+1}, \boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \\
&= \int_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathcal{D})p(A_{n+1}, B_{n+1}|\boldsymbol{\theta})d\boldsymbol{\theta}.
\end{aligned}$$

In general, such integrals may be difficult to find analytically. In practice, we rely on computational methods such as Markov Chain Monte Carlo (MCMC). Alternatively, to predict future values, we can use a Bayesian point estimate for $\boldsymbol{\theta}$, and then assume that they are known as shown below.



10.4 Parameter Estimation in MRFs

Recall that for an MRF G, the probability distribution is given as

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \prod_{c \text{ is a clique in } G} \psi_{\boldsymbol{\theta}}(\boldsymbol{x}_c)/Z(\boldsymbol{\theta}),$$

where $Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \prod_c \psi_{\boldsymbol{\theta}}(\boldsymbol{x}_c)$ is the partition function. Let us consider the frequentist estimation of $\boldsymbol{\theta}$, e.g., maximum likelihood. Unfortunately, the log-likelihood function does not decompose into terms each depending on one component of $\boldsymbol{\theta}$. This is due to the presence of the partition function, which generally depends on all the components of $\boldsymbol{\theta}$, leading to a high-dimensional problem. Furthermore, computing the partition function is a computationally difficult task since it involves computing a sum with possibly exponentially many terms.

Bibliography

[1] B. Hajek, *Random Processes for Engineers*. 2014.

[2] T. T. Nguyen and S. Sanner, "Algorithms for direct 0-1 loss optimization in binary classification," in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, (Atlanta, GA, USA), pp. III–1085–III–1093, JMLR.org, June 2013.