# Chapter 10

# Parameter Estimation in Graphical Models

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:

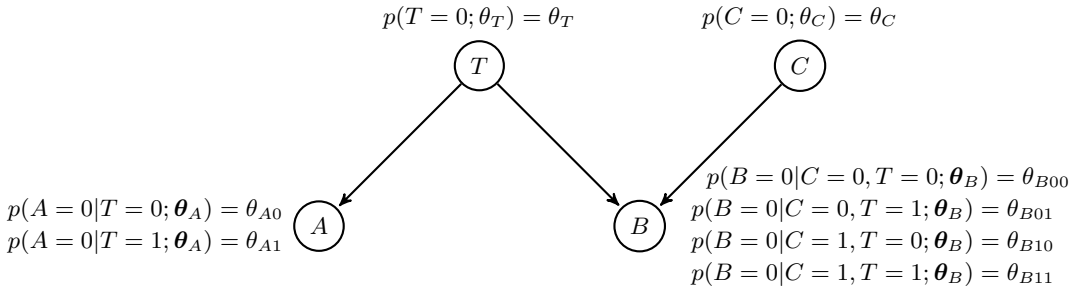
- We will consider the problem of estimating the parameters in graphical models. The problem is simpler in the case of Bayesian networks and for simplicity, that is were our attention will be focused.
- We will not consider the more challenging problem of learning the structure of a network. The best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

Consider a BN with a 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 (**CPDs**)[1], one for each node, and each of these has an unknown parameter vector. We denote the concatenated vector of all parameters as $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m)$. We collect a dataset $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ of $n$ iid samples, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{im})$. Our goal is to estimate $\boldsymbol{\theta}$ and possibly also to predict the next outcome $\boldsymbol{x}_{n+1} = (\boldsymbol{x}_{n+1,1}, \ldots, \boldsymbol{x}_{n+1,m})$.

## 10.1 MLE for Parameters of Bayesian Networks

We will start with maximum likelihood estimation via an example.

**Example 10.1.** 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.

To collect data, on $n$ days, we record whether there is heavy traffic and whether Alice, Bob, and/or Charlie are late, resulting in $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, where $\boldsymbol{x}_i = (T_i, A_i, B_i, C_i)$. Then we maximize the likelihood

$$\arg\max_{\boldsymbol{\theta}} p(\mathcal{D}; \boldsymbol{\theta}) = \arg \max_{\theta_T, \boldsymbol{\theta}_A, \boldsymbol{\theta}_B, \theta_C} p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n; (\theta_T, \boldsymbol{\theta}_A, \boldsymbol{\theta}_B, \theta_C)) \tag{10.1}$$

△

Note that in the above example, the maximization evidently involve 6 dimensions. In real-world problems the networks have many more parameters. 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 see.

**Decomposability of likelihood.** For a network with $m$ nodes, parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m)$ and data $\mathcal{D} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, the likelihood function is

$$p(\mathcal{D}; \boldsymbol{\theta}) = \prod_{i=1}^{n} p(\boldsymbol{x}_i; \boldsymbol{\theta}),$$

where for the $i$th data sample, we have

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

and thus the log-likelihood of the whole dataset is

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

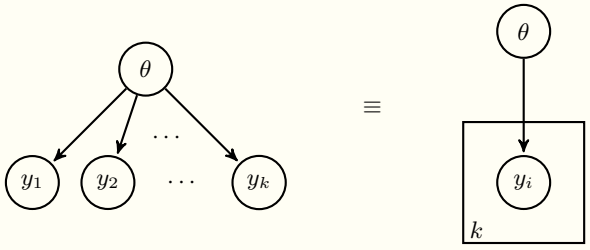
Thus for a given $j$, $\boldsymbol{\theta}_j$ only appears in the term $\sum_{i=1}^{n} \ln p(x_{ij} | \operatorname{pa}(x_{ij}); \boldsymbol{\theta}_j)$ and no other $\boldsymbol{\theta}_k$ appears in this term. So each $\boldsymbol{\theta}_j$, and thus each conditional probability distribution, can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

**Exercise 10.2.** For the TABC network above, what would our data look like? What is the ML estimate for each parameter based on this data? △

## 10.2 Bayesian Parameter Estimation for Bayesian Networks

An alternative approach is using Bayesian inference. Since in the Bayesian view, parameters are considered random, we can augment the Bayesian network by adding the parameters as nodes.

**Side note 1: the plate notation.** Before proceeding, we introduce the plate notation which is helpful for simplifying repeated elements in graphical models, especially a set of iid nodes. Specifically, instead of repeating a node $k$ times, we enclose one instance and indicate how many times that segment of the graph is repeated. Both of the following graphs represent the factorization $p(y_1, \ldots, y_k, \theta) = \prod_{i=1}^{k} p(y_i|\theta)$.



**Side note 2: Conditioning for sets of nodes.** Consider a BN with nodes $y_1, \ldots, y_m$. Assume that the set of nodes can be partitioned into two sets $S_1 = \{y_1, \ldots, y_r\}$ and $S_2 = \{y_{r+1}, \ldots, y_m\}$ such that there are no edges from $S_2$ to $S_1$. Then the following hold

$$p(S_1) = p(y_1, \ldots, y_r) = \prod_{i=1}^{r} p(y_i | \text{ pa}(y_i)), \tag{10.2}$$

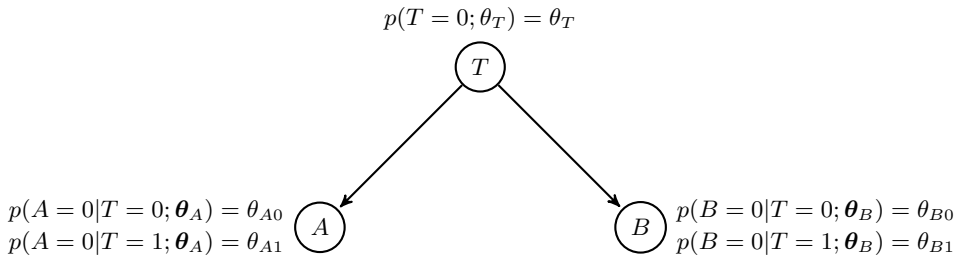
$$p(S_2|S_1) = p(y_{r+1}, \ldots, y_m | y_1, \ldots, y_r) = \prod_{i=r+1}^{m} p(y_i | \text{ pa}(y_i)). \tag{10.3}$$

However, in general,

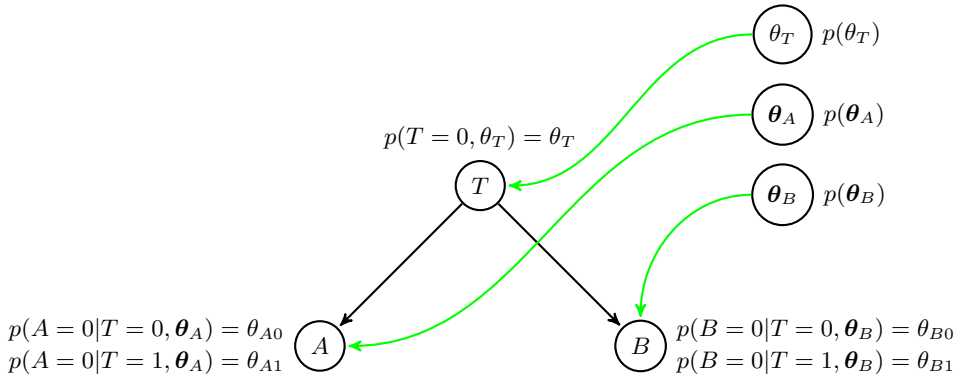
$$p(S_2) = p(y_{r+1}, \ldots, y_m) \neq \prod_{i=r+1}^{m} p(y_i | \text{ pa}(y_i)), \tag{10.4}$$

$$p(S_1|S_2) = p(y_1, \ldots, y_r | y_{r+1}, \ldots, y_m) \neq \prod_{i=1}^{r} p(y_i | \text{ pa}(y_i)) \tag{10.5}$$

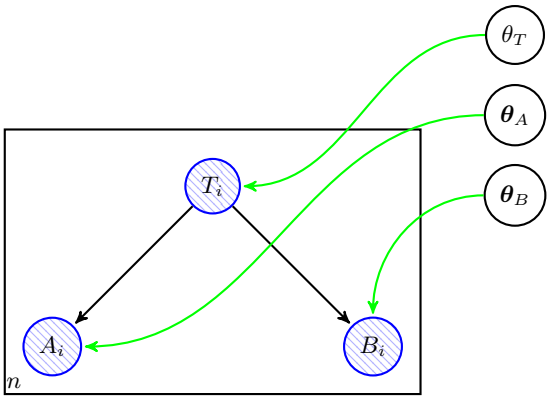
**Bayesian networks with explicit representation of the parameters and data.** Let us consider a simpler version of the network given in Example 10.1, with unknown parameter vector $\boldsymbol{\theta} = (\theta_T, \boldsymbol{\theta}_A, \boldsymbol{\theta}_B)$:




At this point, we are still viewing the parameters as unknown constants. Now to formulate the Bayesian estimation of these parameters, we need to view them as random and add nodes for them to the graph, estimating the parameters as the network:




Incorporating the $n$ samples $\mathcal{D} = \{(T_1, A_1, B_1), \ldots, (T_n, A_n, B_n)\}$, we represent the problem of estimating the parameters as the network, where the CPDs are omitted for clarity:




**Posterior distributions of the parameters.** We are interested in finding $p(\boldsymbol{\theta}|\mathcal{D}) = p(\theta_T, \boldsymbol{\theta}_A, \boldsymbol{\theta}_B|\mathcal{D})$. Note that

$$\begin{aligned} p(\boldsymbol{\theta}|\mathcal{D}) &= p(\theta_T|\mathcal{D})p(\boldsymbol{\theta}_A, \boldsymbol{\theta}_B|\mathcal{D}) && (10.6)\\ &= p(\theta_T|\mathcal{D})p(\boldsymbol{\theta}_A|\mathcal{D})p(\boldsymbol{\theta}_B|\mathcal{D}) && (10.7)\\ &= p(\theta_T|T_1^n)p(\boldsymbol{\theta}_A|T_1^n, A_1^n)p(\boldsymbol{\theta}_B|T_1^n, B_1^n), && (10.8) \end{aligned}$$

where the first equality follows from the fact that given $T_1^i \subset \mathcal{D}$, $\theta_T$ is independent of all other nodes, including $\boldsymbol{\theta}_A, \boldsymbol{\theta}_B$. In other words, $T_1^i$ is the Markov blanket of $\theta_T$. The second equality also holds because $\mathcal{D}$ contains the Markov blanket for $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$. Similarly, the last equality follows from a Markov blanket-type argument.

In other words, to estimate each parameter, we need to only consider the part of the data that is in the parameter's Markov blanket. This also makes intuitive sense: For example, to estimate the probability of Alice being late given the state of traffic, only the part of data that deals with Alice's arrival time and traffic is relevant. The fact that the posterior for each parameter can be determined separately significantly reduces the computational complexity.

**Example 10.3.** Let us find $p(\boldsymbol{\theta}_A|\mathcal{D})$, assuming that the prior satisfies $p(\boldsymbol{\theta}_A) = p(\theta_{A0})p(\theta_{A1})$,

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

(Why does the relation shown as $\overset{(*)}{\propto}$ hold?) Since the terms depending on $\theta_{A0}$ and $\theta_{A1}$ separate, they are conditionally independent and we can estimate them separately: Hence, the estimators of $\theta_A^0$ and $\theta_A^1$ are

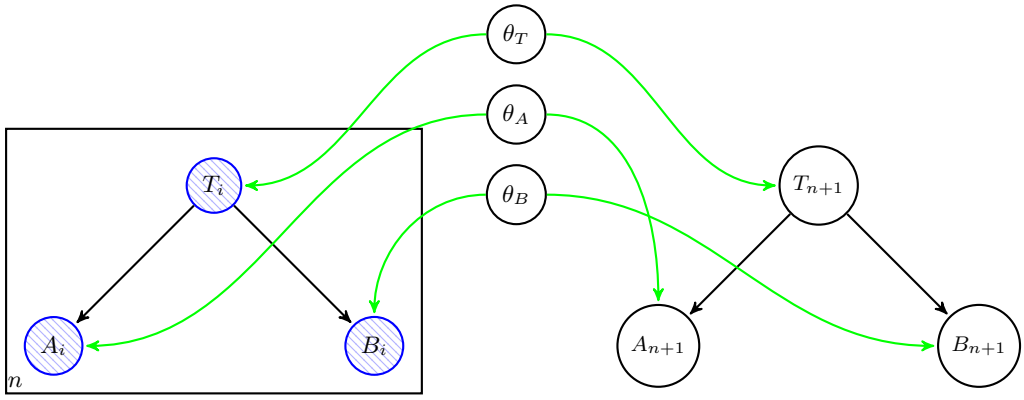
$$\begin{aligned} p(\theta_{A0}|\mathcal{D}) &\propto p(\theta_{A0})\prod_{i:T_i=0} p(A_i|T_i=0, \theta_{A0}),\\ p(\theta_{A1}|\mathcal{D}) &\propto p(\theta_{A1})\prod_{i:T_i=1} p(A_i|T_i=1, \theta_{A1}). \end{aligned}$$

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,

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

△

**Predicting future outcomes.** We can also add future outcomes to predict their value to the network:




Let $\boldsymbol{x}_{n+1} = (T_{n+1}, A_{n+1}, B_{n+1})$. We have

$$p(\boldsymbol{x}_{n+1}, \boldsymbol{\theta}|\mathcal{D}) = p(\boldsymbol{\theta}|\mathcal{D})p(\boldsymbol{x}_{n+1}|\mathcal{D}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathcal{D})p(\boldsymbol{x}_{n+1}|\boldsymbol{\theta}). \tag{10.9}$$

We have already seen how to find $p(\boldsymbol{\theta}|\mathcal{D})$. We can decompose $p(\boldsymbol{x}_{n+1}|\boldsymbol{\theta})$ as given by the Bayesian network:

$$p(\boldsymbol{x}_{n+1}|\boldsymbol{\theta}) = p(T_{n+1}|\theta_T)p(A_{n+1}|\boldsymbol{\theta}_A, T_{n+1})p(B_{n+1}|\boldsymbol{\theta}_B, T_{n+1}). \tag{10.10}$$

Note that the terms on the right are known probability distributions.

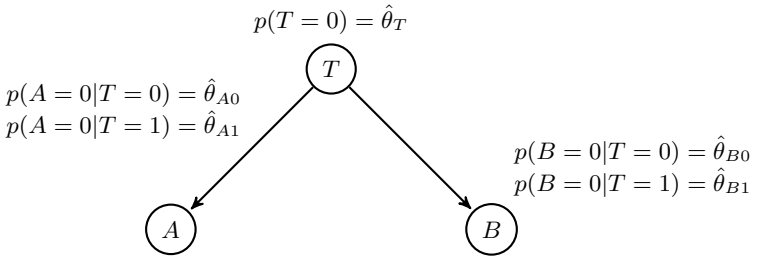
Finally, if we are interested in a specific future outcome, e.g., $p(A_{n+1}|\mathcal{D})$, we can find it through an appropriate integration/summation of $p(\boldsymbol{x}_{n+1}, \boldsymbol{\theta}|\mathcal{D})$.

**Example 10.4.** The posterior probability of the next sample $(A_{n+1}, B_{n+1})$ is

$$p(A_{n+1}, B_{n+1}|\mathcal{D}) = \int_{\boldsymbol{\theta}} \sum_{T_{n+1}} p(A_{n+1}, B_{n+1}, T_{n+1}, \boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta},$$

where we can find the integrand/summand as described in (10.9). In general, such integrals may be difficult to find analytically. In practice, we rely on computational methods such as Markov Chain Monte Carlo (MCMC).

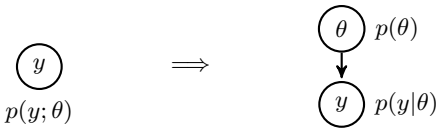
Alternatively, to predict future values, we can use a Bayesian point estimate for $\boldsymbol{\theta}$, and then assume that they are known as shown below.



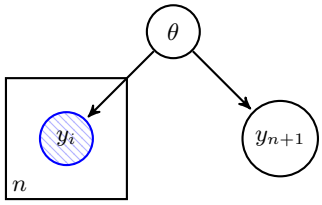

△

We can use graphical models to represent some of the estimation/learning problems we have already discussed in previous chapters.

**Example 10.5.** Let $y$ have a distribution $p$ with an unknown parameter $\theta$. Below, on the left a graphical model for $y$ is shown without explicit representation of $\theta$ and on the right, $\theta$ is added as a node:




We have $n$ independent samples, $\mathcal{D} = \{y_1, y_2, \ldots, y_n\}$, from the distribution and our goal is to predict the next outcome $y_{n+1}$. We can augment the graph to represent the problem as follows:




with a joind distribution that can be written as $p(\theta, y_1^n, y_{n+1}) = p(\theta)p(y_1^n|\theta)p(y_{n+1}|\theta)$.

We can perform similar analysis as we have done in the Bayesian Estimation chapter, using d-separation to verify independence relations. 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,

$$\mathbb{E}[y_{n+1}|y_1^n] = \mathbb{E}[\mathbb{E}[y_{n+1}|\theta, y_1^n]|y_1^n] = \mathbb{E}[\mathbb{E}[y_{n+1}|\theta]|y_1^n]. \tag{10.11}$$

Roughly speaking, to learn about $y_{n+1}$ given $y_1^n$, we must first learn about $\theta$ 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

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

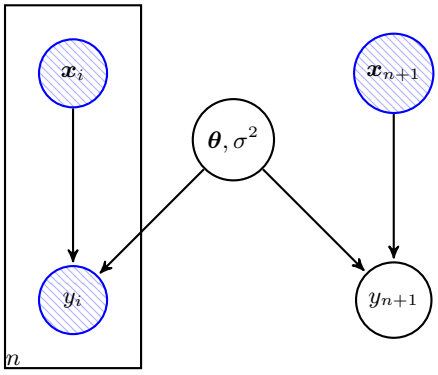
△

In more general cases, some of the "future outcomes" may also be known. But the same principles discussed above, still apply.

**Example 10.6** (Bayesian Linear Regression)**.** Consider the regression problem

$$p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta}, \sigma^2) \sim \mathcal{N}(\boldsymbol{\theta}^T\boldsymbol{x}_i, \sigma^2),$$

with data $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$. We are interested in determining $p(y_{n+1}|\boldsymbol{x}_{n+1}, \mathcal{D})$. The problem can be represented as the graph



We note that

$$p(y_{n+1}, \boldsymbol{\theta}, \sigma^2 | \boldsymbol{x}_{n+1}, \mathcal{D}) = p(\boldsymbol{\theta}, \sigma^2 | \boldsymbol{x}_{n+1}, \mathcal{D}) p(y_{n+1} | \boldsymbol{\theta}, \sigma^2, \boldsymbol{x}_{n+1}, \mathcal{D}) \tag{10.12}$$

$$= p(\boldsymbol{\theta}, \sigma^2 | \mathcal{D}) p(y_{n+1} | \boldsymbol{x}_{n+1}, \boldsymbol{\theta}, \sigma^2), \tag{10.13}$$

where we have used the following facts: $\boldsymbol{\theta}, \sigma^2 \perp\!\!\!\perp \boldsymbol{x}_{n+1} | \mathcal{D}$ and $y_{n+1} \perp\!\!\!\perp \mathcal{D} | \boldsymbol{x}_{n+1}, \boldsymbol{\theta}, \sigma^2$. We know how to find $p(\boldsymbol{\theta}, \sigma^2 | \mathcal{D})$ and $p(y_{n+1} | \boldsymbol{x}_{n+1}, \boldsymbol{\theta}, \sigma^2)$ is given by assumption. While we can find $p(y_{n+1} | \boldsymbol{x}_{n+1})$ through integration analytically, as discussed in the linear regression chapter, we normally produce samples for $\boldsymbol{\theta}, \sigma^2$ and then proceed to produce samples for $p(y_{n+1} | \boldsymbol{x}_{n+1}, \boldsymbol{\theta}, \sigma^2)$.

△

## 10.3 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. We will discuss computational approaches to this problem later in the course.

Helpful references: [2, 3, 1]

## References

[1] Christopher M. Bishop. *Pattern Recognition And Machine Learning*. New York: Springer, 2006. URL: http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%5C%20-%5C%20Pattern%5C%20Recognition%5C%20And%5C%20Machine%5C%20Learning%5C%20-%5C%20Springer%5C%20%5C%202006.pdf (visited on 02/14/2017).

[2] Michael I Jordan. *An Introduction to Probabilistic Graphical Models (Preprints and Course Notes)*. University of California, Berkeley, 2003.

[3] David JC MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge university press, 2003.