An Introduction to Probabilistic Graphical Models

Michael I. Jordan
University of California, Berkeley

June 30, 2003

Chapter 9

Completely Observed Graphical Models

The models that we have discussed until now have, for the most part, involved a single node or a single node and its parents. We have seen how to find maximum likelihood estimates for a variety of models of this form.

In the current chapter, we learn that the techniques that we have developed extend with essentially no additional labor to the entire class of directed graphical models, under the assumption of *complete observations.* Thus, if we assume that our data set assigns values to all of the random variables in the model—i.e., that there are no *latent variables*—then we can find maximum likelihood estimates of parameters in a straightforward way, essentially by solving the problem separately at each node. The parameter estimation problem "decouples." The underlying reason for this appealing result is the product form of the joint probability distribution.

We also discuss the problem of maximum likelihood parameter estimation for undirected graphical models. In this case the parameter estimation problem decouples only for a special class of graphical models, known as *decomposable models.* For general undirected models the parameter estimation problem does not decouple, the essential reason being the presence of the global normalization factor Z. Nonetheless, we will show that there is a local characterization of maximum likelihood estimates for general undirected models, and we present algorithms for finding these estimates.

In Section 9.4, we discuss the problems that arise when we remove the restriction to completely observed models. The material in this section sets the stage for our treatment of latent variable models in Chapters 10 through 15.

Our focus in this chapter is on the likelihood function and maximum likelihood estimation. We do, however, provide a short discussion of Bayesian estimation in the completely observed setting in Section **??**.

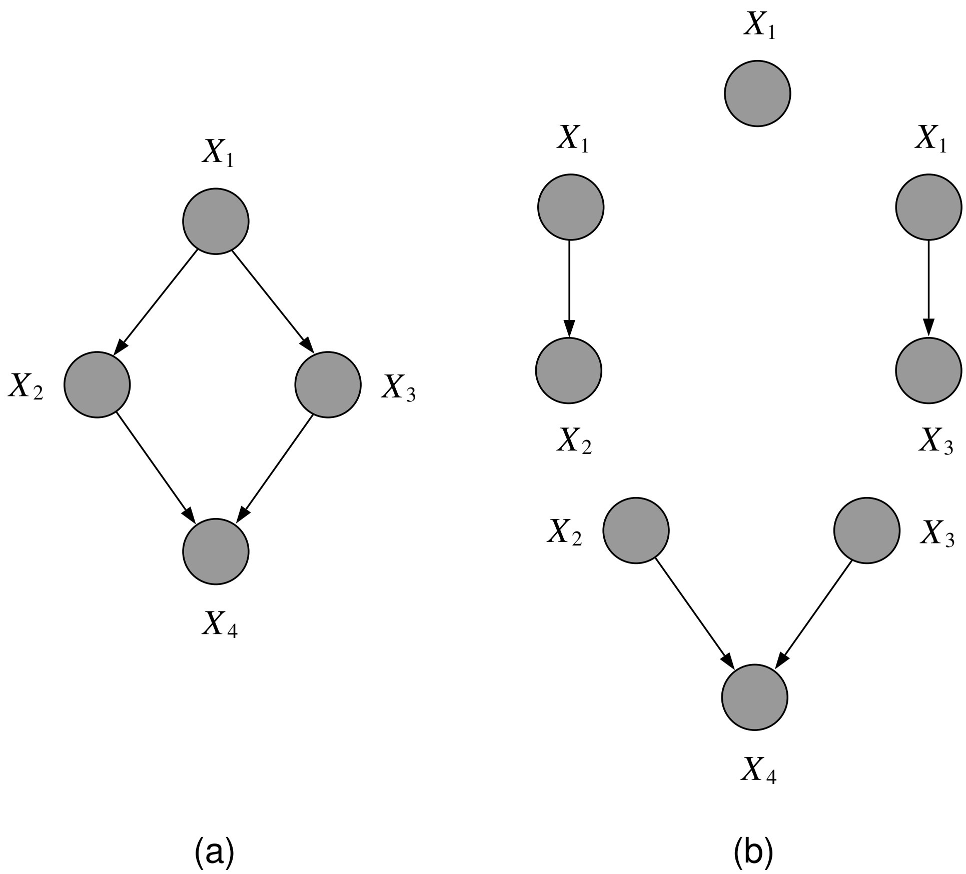

Figure 9.1: (a) A directed graphical model. (b) Solving the maximum likelihood problem in (a) is equivalent to solving separate maximum likelihood problems for each node conditioned on its parents.

9.1 The basic idea

Before jumping into the details, let us consider a simple example. The graphical model shown in Figure 9.1(a) has the following joint probability distribution:

$$p(x \mid \theta) = p(x_1 \mid \theta_1) p(x_2 \mid x_1, \theta_2) p(x_3 \mid x_1, \theta_3) p(x_4 \mid x_2, x_3, \theta_4). \tag{9.1}$$

Taking the logarithm of this distribution we have:

$$\log p(x \mid \theta) = \log p(x_1 \mid \theta_1) + \log p(x_2 \mid x_1, \theta_2) + \log p(x_3 \mid x_1, \theta_3) + \log p(x_4 \mid x_2, x3, \theta_4). \tag{9.2}$$

From this expression, we see that the parameters, θ_i, appear in different terms, and thus the maximization of the log probability with respect to a given θ_i can be carried out independently of the other maximizations.

Figure 9.1(b) provides a graphical depiction of this fact. The problem of finding the maximum likelihood parameters for the model in Figure 9.1(a) breaks into separate maximum likelihood problems, one for each node in the graph. As suggested in Figure 9.1(b), if we solve these separate maximum likelihood problems, we have solved the overall maximum likelihood problem.

9.2 Directed models

This chapter marks our first attempt to treat rather general families of graphical models. Although our results are rather straightforward, we will need to be a bit fussier with regards to notation than we have been in earlier chapters. This will allow us to state our results in a simple and general way, and will prepare the ground for later chapters.

Let $\mathcal{G}$ be a directed graph, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ the set of edges of the graph. We associate a random vector X with the graph, where the components of the vector are indexed by the nodes in the graph. Thus, X_u denotes the random variable associated with node $u \in \mathcal{V}$, and x_u denotes a realization of this random variable.

Recall that we also allow subsets of $\mathcal{V}$ to serve as indices; thus, X_C refers to the set of components indexed by a subset $C \subseteq \mathcal{V}$. The vector X will itself sometimes be written $X_{\mathcal{V}}$.

We define a probability model for a directed graph via a set of local conditional probability distributions. Thus, to each node $u \in \mathcal{V}$ we associate a local conditional probability distribution $p(x_u \,|\, x_{\pi_u}, \theta_u)$, where π_u denotes the set of indices of the parents of u and where θ_u is a parameter vector. The overall probability associated with the graph $\mathcal{G}$ is a product of these local conditional probabilities:

$$p(x_{\mathcal{V}} \,|\, \theta) = \prod_{u \in \mathcal{V}} p(x_u \,|\, x_{\pi_u}, \theta_u), \tag{9.3}$$

where $\theta = (\theta_1, \theta_2, \ldots, \theta_m)$.

We treat the case of complete observations in this chapter. A *complete observation* is an assignment of values to all of the random variables $X_{\mathcal{V}}$ in the model.

In many problems the data are assumed to consist of a set of N independent, identically distributed (IID) observations. Such an assumption requires no special treatment within the graphical model formalism—we simply replicate a basic graphical structure N times. The result is itself a graphical model. It is important, however, to be able to continue to refer to the probability model associated with a single observable vector $X_{\mathcal{V}}$, apart from any considerations of the sampling mechanism, and also to be able to refer to the overall model that assigns probability to the IID replicates of $X_{\mathcal{V}}$. Let us continue to use the notation $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and $p(x_{\mathcal{V}} \,|\, \theta)$ to refer to the probability model associated with a single observable vector $X_{\mathcal{V}}$. We also construct an augmented graphical model, $\mathcal{G}^{(N)} = (\mathcal{V}^{(N)}, \mathcal{E}^{(N)})$, that incorporates the IID sampling assumption—this graph consists of N disconnected replicates of $\mathcal{G}$. The nodes $\mathcal{V}^{(N)}$ in this augmented graph are indexed using a pair of labels, (u, n), where $u \in \mathcal{V}$ designates a node in the underlying graphical model $\mathcal{G}$, and where $n \in \{1, 2, \ldots, N\}$ designates the replication number (see Figure 9.2).

We again allow subsets of indices to be used wherever single indexes are used. Thus, (C, n) denotes the nth replicate of the set C, for $C \subseteq \mathcal{V}$. In particular, $(\mathcal{V}, n)$ denotes the nth replicate of

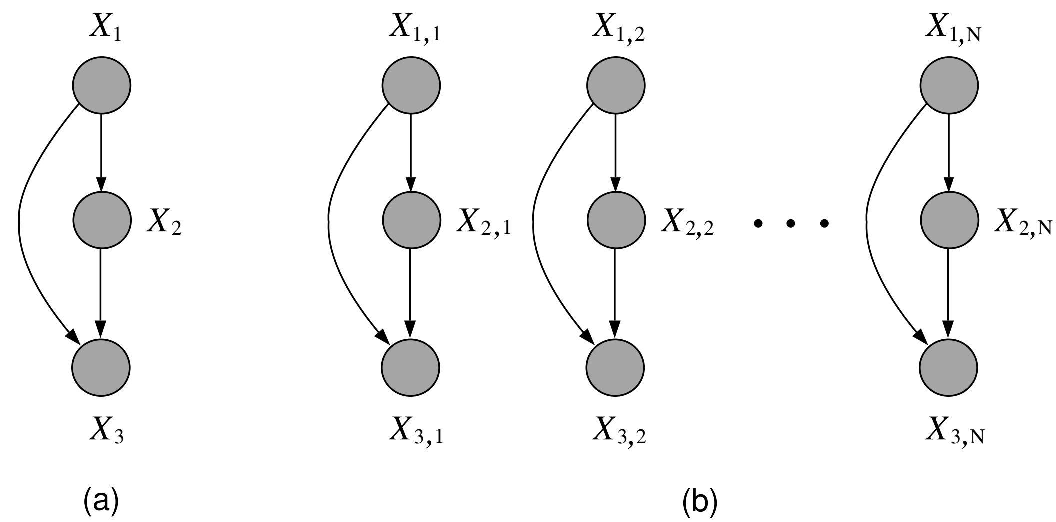

Figure 9.2: (a) An example of a graphical model $\mathcal{G}$, where $\mathcal{V} = \{1, 2, 3\}$. The set of random variables in the model is represented as $X_{\mathcal{V}} = (X_1, X_2, X_3)$, and $x_{\mathcal{V}} = (x_1, x_2, x_3)$ represents a realization of these variables. (b) An example of the graph $\mathcal{G}^{(N)}$, obtained by making N replicates of $\mathcal{G}$. Each random variable in this graph is indexed with two indices, (u, n), the first denoting the underlying node in $\mathcal{G}$, and the second denoting the replication number. The nth complete observation is an assignment of values to all of the nodes in the nth replicate, and is denoted $x_{\mathcal{V},n} = (x_{1,n}, x_{2,n}, x_{3,n})$.

the entire set of observable nodes $\mathcal{V}$. This notation allows us to write:

$$\mathcal{D} = (x_{\mathcal{V},1}, x_{\mathcal{V},2}, \ldots, x_{\mathcal{V},N}) \tag{9.4}$$

as our representation for the entire set of *observed data* in the completely observed setting.

The local conditional probability associated with a node (u, n) in $\mathcal{G}^{(N)}$ is defined to be the local conditional probability associated with the index u in the underlying graphical model $\mathcal{G}$. We thus obtain the following probability model for the graph $\mathcal{G}^{(N)}$:

$$\begin{aligned} p(\mathcal{D} \,|\, \theta) &= \prod_n p(x_{\mathcal{V},n} \,|\, \theta) && (9.5) \\ &= \prod_n \prod_u p(x_{u,n} \,|\, x_{\pi_u,n}, \theta_u), && (9.6) \end{aligned}$$

where Eq. (9.5) is just the independence assumption, but it is also worth noting that it follows from the usual construction of the joint probability of a graph as a product over local conditional probabilities; here the graph is formed from a disconnected set of subgraphs indexed by n.

Taking the logarithm of Eq. (9.6) we obtain the log likelihood:

$$l(\theta; \mathcal{D}) = \sum_n \sum_u \log p(x_{u,n} \,|\, x_{\pi_u,n}, \theta_u); \tag{9.7}$$

Note that the log likelihood is a sum of a number of terms, each of which refers to only one of the parameter vectors θ_u. In estimating θ_u we can ignore all terms that involve $\theta_{u'}$, for $u' \neq u$. Note, moreover, that those terms involving θ_u refer only to x_u and x_{π_u}, which are observations of node u and its parents π_u. Thus, for the purposes of estimating the parameter vector θ_u, we need only focus on the data associated with the node u and its parents—we can ignore the data associated with the other nodes in the graph. In other words, the local subset of observations $\{x_{u,n}, x_{\pi_u,n}\}_{n=1}^N$ associated with these nodes is *sufficient* for θ_u.

We can often say more. In particular, if the observations $\{x_{u,n}, x_{\pi_u,n}\}_{n=1}^N$ can themselves be summarized by finite-dimensional sufficient statistics, as they can if each local conditional probability model, $p(x_u \mid x_{\pi_u}, \theta_u)$, is an exponential family distribution, then the entire estimation problem can be reduced to a collection of finite-dimensional sufficient statistics. In essence, we can reduce our problem from observations associated with the graph $\mathcal{G}^{(N)}$ to statistics associated with the graph $\mathcal{G}$, and from there to statistics associated with single nodes and their parents. In the next section we illustrate this reduction in the context of discrete graphical models.

9.2.1 Discrete models

Let us work out the sufficient statistics for a general graphical model in which all nodes are discrete. In doing so, we exemplify the general results that we have just discussed and also introduce a few "tricks of the trade" that will be useful in the following section as well as in later chapters.

Counts and marginal counts

We begin by introducing a notation for counts, which are the sufficient statistics for multinomial random variables. For a given configuration $x_\mathcal{V}$, let $m(x_\mathcal{V})$ denote the number of times that $x_\mathcal{V}$ is observed among the observations in the dataset $\mathcal{D}$. (There are only a finite number of possible configurations $x_\mathcal{V}$, and each data point $x_{\mathcal{V},n}$ must be one of these configurations). We can represent this count as a sum:

$$m(x_\mathcal{V}) \triangleq \sum_n \delta(x_\mathcal{V}, x_{\mathcal{V},n}), \tag{9.8}$$

where $\delta(x_\mathcal{V}, x_{\mathcal{V},n})$ is one if its arguments are equal and zero otherwise.

We can also define "marginal counts"—the counts associated with subsets of nodes. For any given subset C, let $m(x_C)$ denote the number of times that configuration x_C is observed in the data set. This is obtained by computing:

$$m(x_C) \triangleq \sum_{x_{\mathcal{V}\setminus C}} m(x_\mathcal{V}), \tag{9.9}$$

which can be viewed as a "marginalization" operation.

As an example, suppose that $\mathcal{V} = \{1, 2, 3\}$. In this case, $m(x_\mathcal{V})$ can be represented as a three-dimensional table. Suppose that the parent of X_2 is X_1. To compute the marginal count $m(x_1, x_2)$, we sum over x_3:

$$m(x_1, x_2) = \sum_{x_3} m(x_1, x_2, x_3), \tag{9.10}$$

which is a two-dimensional table. To compute the marginal count $m(x_1)$, a one-dimensional table, we sum over x_2 and x_3:

$$m(x_1) = \sum_{x_2,x_3} m(x_1, x_2, x_3) = \sum_{x_2} m(x_1, x_2). \tag{9.11}$$

Note also that if we sum over all three variables we obtain the scalar N, the total number of observations.

A particular subset of interest is the subset consisting of a node u and its parents π_u—the *family* associated with node u. Letting $\phi_u \triangleq \{u\} \cup \pi_u$ denote this family, we have:

$$m(x_{\phi_u}) \triangleq \sum_{x_{\mathcal{V}\setminus\phi_u}} m(x_{\mathcal{V}}). \tag{9.12}$$

That is, $m(x_{\phi_u})$ is the count of the number of times a node u takes on a specific value and its parents π_u take on a specific configuration. Study the expression under the summation sign carefully. We are taking the sum over all nodes in $\mathcal{V}$ other than u and its parents π_u.

The joint probability

Let us turn to the representation of the joint probability distribution in the discrete case. We begin by discussing the tabular case, in which a separate parameter is associated with each possible joint configuration of a node and its parents. In particular, we define the parameter vector $\theta_v(x_{\phi_v})$ to be a nonnegative, multidimensional table indexed by the joint configuration of v and π_v. The normalization condition requires:

$$\sum_{x_v} \theta_v(x_{\phi_v}) = \sum_{x_v} \theta_v(x_v, x_{\pi_v}) = 1, \tag{9.13}$$

where we recall that $\phi_v \triangleq \{v\} \cup \pi_v$ is the family associated with node v.

Given such a normalized table we define:

$$p(x_v \mid x_{\pi(v)}, \theta_v) \triangleq \theta_v(x_{\phi_v}) \tag{9.14}$$

as the local conditional probability of node v. This is a generic tabular representation that places no constraints on the local conditional probabilities beyond the normalization constraint.

Taking the product over v, we obtain the joint probability distribution:

$$\begin{aligned} p(x_{\mathcal{V}} \mid \theta) &= \prod_v p(x_v \mid x_{\pi_v}, \theta_v) && (9.15) \\ &= \prod_v \theta_v(x_{\phi_v}), && (9.16) \end{aligned}$$

as a product of (normalized) potentials.

We take a further product over n to obtain the total probability of an IID data set $\mathcal{D} = (x_{\mathcal{V},1}, x_{\mathcal{V},2}, \ldots, x_{\mathcal{V},N})$. At this point, however, we can make our results look neater if we recognize that some of the observations have the same value. These observations necessarily have the same probability, and thus it is helpful to group them explicitly. To do this, we make use of a trick that should be familiar from our earlier work with the multinomial distribution:

$$p(x_{\mathcal{V},n} \mid \theta) = \prod_{x_{\mathcal{V}}} p(x_{\mathcal{V}} \mid \theta)^{\delta(x_{\mathcal{V}}, x_{\mathcal{V},n})}. \tag{9.17}$$

Here the dummy variable $x_{\mathcal{V}}$ ranges across configurations of the nodes rather than across data points. Using this representation we write the joint probability as follows, working in the log domain for convenience:

$$\begin{aligned}
\log p(\mathcal{D} \mid \theta) &= \log \left(\prod_n p(x_{\mathcal{V},n} \mid \theta) \right) && (9.18)\\
&= \sum_n \log \left(\prod_{x_{\mathcal{V}}} p(x_{\mathcal{V}} \mid \theta)^{\delta(x_{\mathcal{V}}, x_{\mathcal{V},n})} \right) && (9.19)\\
&= \sum_n \sum_{x_{\mathcal{V}}} \delta(x_{\mathcal{V}}, x_{\mathcal{V},n}) \log p(x_{\mathcal{V}} \mid \theta) && (9.20)\\
&= \sum_{x_{\mathcal{V}}} m(x_{\mathcal{V}}) \log p(x_{\mathcal{V}} \mid \theta). && (9.21)
\end{aligned}$$

Note that the sum over n has disappeared; we have in essence reduced our representation of joint probability from a function on the graph $\mathcal{G}^{(N)}$ to a function on the graph $\mathcal{G}$. Continuing the derivation, we have:

$$\begin{aligned}
\log p(\mathcal{D} \mid \theta) &= \sum_{x_{\mathcal{V}}} m(x_{\mathcal{V}}) \log p(x_{\mathcal{V}} \mid \theta) && (9.22)\\
&= \sum_{x_{\mathcal{V}}} m(x_{\mathcal{V}}) \log \left(\prod_v \theta_v(x_{\phi_v}) \right) && (9.23)\\
&= \sum_{x_{\mathcal{V}}} m(x_{\mathcal{V}}) \sum_v \log \theta_v(x_{\phi_v}) && (9.24)\\
&= \sum_v \sum_{x_{\phi_v}} \sum_{x_{\mathcal{V} \setminus \phi_v}} m(x_{\mathcal{V}}) \log \theta_v(x_{\phi_v}) && (9.25)\\
&= \sum_v \sum_{x_{\phi_v}} \left(\sum_{x_{\mathcal{V} \setminus \phi_v}} m(x_{\mathcal{V}}) \right) \log \theta_v(x_{\phi_v}) && (9.26)\\
&= \sum_v \sum_{x_{\phi_v}} m(x_{\phi_v}) \log \theta_v(x_{\phi_v}). && (9.27)
\end{aligned}$$

This result expresses the logarithm of the joint probability as a sum of terms defined on the families $\{\phi_v\}$.

Taking the exponential of both sides of Eq. (9.27), we see that we have an exponential family distribution for $\mathcal{D}$, with $m(x_{\phi_v})$ as the sufficient statistics and $\log \theta_v(x_{\phi_v})$ as the natural parameters.

Note also the decoupling discussed earlier. The parameters $\theta_v(x_{\phi_v})$ appear in separate terms which can be optimized independently of each other. In particular, to estimate $\theta_v(x_{\phi_v})$, we maximize $m(x_{\phi_v}) \log \theta_v(x_{\phi_v})$ with respect to $\theta_v(x_{\phi_v})$. Adding a Lagrangian term to handle the normalization constraint in Eq. (9.13), we obtain (see Exercise **??**):

$$\hat{\theta}_{v,ML}(x_{\phi_v}) = \frac{m(x_{\phi_v})}{m(x_{\pi_v})} = \frac{m(x_v, x_{\pi_v})}{m(x_{\pi_v})}. \tag{9.28}$$

The result has an intuitively-appealing form—the maximum likelihood estimate is the ratio of the count of the number of times a node and its parents are jointly in a specific configuration to the count of the number of times its parents are in that configuration. These estimates are formed independently at each of the nodes in the graph.

Tabular representations are useful for graphs with small families, but quickly become unwieldy when the number of parents grows (the size of a table is exponential in the number of parents). For large families we generally wish to constrain $p(x_v \mid x_{\pi_v}, \theta_v)$ in some way. The generalized linear models (GLIMs) studied in Chapter 8 provide one example of such a constrained representation. Moreover, for GLIMs we have an efficient algorithm—the IRLS algorithm—for obtaining parameter estimates. The decoupling of the log likelihood implies that if each local conditional model is a GLIM model, then we solve the maximum likelihood problem for the graph as a whole by running the IRLS algorithm separately at each node.

In general, in the case of completely observed data, any solution to the problem of estimating parameters at a single node (conditional on its parents) applies immediately to general directed graphs.

9.3 Undirected models

Undirected models are in certain respects more flexible than their directed counterparts. Whereas in directed models the potentials are restricted to conditional distributions that model the dependence of a variable on its parents, potentials in undirected models are unnormalized functions defined on arbitrary subsets of nodes. In domains such as vision and information retrieval, which involve large collections of variables that do not necessarily have a natural ordering, undirected models are often the preferred choice.

Such flexibility comes at a cost, however. In particular, undirected graphical models require an explicit global normalization factor—the $1/Z$ factor in the definition of the joint probability. This global normalization factor couples the parameters and complicates the parameter estimation problem. There is, however, a special class of undirected models—the decomposable models—for which the parameter estimation problem decouples, despite the presence of the global normalization factor. For general undirected models the problem does not decouple, but, nonetheless, effective parameter estimation algorithms that exploit the structure of the graph are available. We describe some of the basic alternatives in this section, and treat the topic of parameter estimation for undirected models in more detail in Chapter 19.

Much of the notation for undirected models transfers over without change from the directed setting. Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ the set of edges of the graph. We again associate a random vector $X_{\mathcal{V}}$ with the graph and refer to random vectors associated with subsets of the nodes using the notation X_C, for $C \subseteq \mathcal{V}$. When the observed data are IID replicates, we again use a second index for the replicates. Thus $X_{C,n}$ denotes the nth replicate of the subset C, and the observed data are denoted:

$$\mathcal{D} = (x_{\mathcal{V},1}, x_{\mathcal{V},2}, \ldots, x_{\mathcal{V},N}), \tag{9.29}$$

where the index $\mathcal{V}$ reflects our focus on complete data.

We parameterize an undirected graphical model via a set of clique potentials $\psi_C(x_C)$, for $C \in \mathcal{C}$, where $\mathcal{C}$ is a set of cliques. Note that we do not necessarily assume that $\mathcal{C}$ contains all of the cliques in the graph, nor that the cliques in $\mathcal{C}$ are maximal. Given such a set of clique potentials, we define the joint probability, $p(x_{\mathcal{V}} \,|\, \theta)$, as follows:

$$p(x_{\mathcal{V}} \,|\, \theta) = \frac{1}{Z} \prod_C \psi_C(x_C), \tag{9.30}$$

where $\theta = \{\psi_C(x_C), C \in \mathcal{C}\}$ is the collection of parameters, and where Z is the normalization factor:

$$Z = \sum_{x_{\mathcal{V}}} \prod_C \psi_C(x_C), \tag{9.31}$$

obtained by summing (or integrating in the continuous case) over all configurations $x_{\mathcal{V}}$.

9.3.1 Discrete models

In order to simplify our discussion we specialize to discrete random variables for the remainder of this section (see Chapter 19 for a discussion of continuous-valued undirected models).

Recall our notation for counts. The number of times that configuration $x_{\mathcal{V}}$ is observed in a dataset $\mathcal{D}$ is represented as follows:

$$m(x_{\mathcal{V}}) \triangleq \sum_n \delta(x_{\mathcal{V}}, x_{\mathcal{V},n}), \tag{9.32}$$

and

$$m(x_C) \triangleq \sum_{x_{\mathcal{V}\setminus C}} m(x_{\mathcal{V}}) \tag{9.33}$$

is the marginal count for clique C. Note in particular that

$$N = \sum_{x_{\mathcal{V}}} m(x_{\mathcal{V}}) \tag{9.34}$$

is the total number of observations.

To express the log likelihood in terms of counts (the sufficient statistics for discrete models) we proceed as in the directed case, introducing the dummy variable $x_{\mathcal{V}}$:

$$p(x_{\mathcal{V},n} \,|\, \theta) = \prod_{x_{\mathcal{V}}} p(x_{\mathcal{V}} \,|\, \theta)^{\delta(x_{\mathcal{V}}, x_{\mathcal{V},n})}, \tag{9.35}$$

and writing the probability of the observed data as follows:

$$\begin{aligned} p(\mathcal{D} \mid \theta) &= \prod_n p(x_{\mathcal{V},n} \mid \theta) &(9.36)\\ &= \prod_n \prod_{x_\mathcal{V}} p(x_\mathcal{V} \mid \theta)^{\delta(x_\mathcal{V}, x_{\mathcal{V},n})}. &(9.37)\end{aligned}$$

This trick allows us to write the log likelihood in terms of the marginal counts:

$$\begin{aligned} l(\theta; \mathcal{D}) &= \log p(\mathcal{D} \mid \theta) &(9.38)\\ &= \sum_n \sum_{x_\mathcal{V}} \delta(x_\mathcal{V}, x_{\mathcal{V},n}) \log p(x_\mathcal{V} \mid \theta) &(9.39)\\ &= \sum_{x_\mathcal{V}} m(x_\mathcal{V}) \log p(x_\mathcal{V} \mid \theta) &(9.40)\\ &= \sum_{x_\mathcal{V}} m(x_\mathcal{V}) \log \left(\frac{1}{Z} \prod_C \psi_C(x_C) \right) &(9.41)\\ &= \sum_{x_\mathcal{V}} m(x_\mathcal{V}) \sum_C \log \psi_C(x_C) - \sum_{x_\mathcal{V}} m(x_\mathcal{V}) \log Z &(9.42)\\ &= \sum_C \sum_{x_C} m(x_C) \log \psi_C(x_C) - N \log Z &(9.43)\end{aligned}$$

We see that the marginal counts $m(x_C)$, for $C \in \mathcal{C}$, are the sufficient statistics for our model. This is reminiscent of the directed case, where the cliques $\mathcal{C}$ were the families $\{\phi_v\}$. Note, however, an important difference between the undirected log likelihood and its directed counterpart—the appearance of the term $N \log Z$.

9.3.2 Maximum likelihood estimation

To find maximum likelihood estimates we take derivatives of the log likelihood and set to zero. Unfortunately, due to the $N \log Z$ term, such a calculation yields a coupled, nonlinear set of equations in which the parameters appear implicitly. It is not obvious how (in general) to solve these equations to obtain explicit estimates for the parameters. All is not lost, however; the implicit equations do reveal an interesting local property of the maximum likelihood estimates. Moreover, as we discuss in Section 9.3.4, this property provides the inspiration for an iterative algorithm for finding the parameter estimates.

Let us thus proceed to calculating the derivatives. In these calculations note that $\psi_C(x_C)$ is the independent variable, where both the clique C and the configuration x_C have been fixed (in the discrete case, this picks out the cell indexed by x_C in the table representing the potential function on clique C).

The derivative of the first term in the log likelihood, Eq. (9.43), with respect to $\psi_C(x_C)$ is obtained immediately; it is equal to $m(x_C)/\psi_C(x_C)$. We turn to $\log Z$:

$$\frac{\partial \log Z}{\partial \psi_C(x_C)} = \frac{1}{Z} \frac{\partial}{\partial \psi_C(x_C)} \left(\sum_{\tilde{x}} \prod_D \psi_D(\tilde{x}_D) \right) \qquad (9.44)$$

$$= \frac{1}{Z}\sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) \frac{\partial}{\partial \psi_C(x_C)} \left(\prod_D \psi_D(\tilde{x}_D) \right) \tag{9.45}$$

$$= \frac{1}{Z}\sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) \prod_{D \neq C} \psi_D(\tilde{x}_D) \tag{9.46}$$

$$= \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) \frac{1}{\psi_C(\tilde{x}_C)} \frac{1}{Z} \prod_D \psi_D(\tilde{x}_D) \tag{9.47}$$

$$= \frac{1}{\psi_C(x_C)} \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) p(\tilde{x}) \tag{9.48}$$

$$= \frac{p(x_C)}{\psi_C(x_C)}, \tag{9.49}$$

where we have dropped the explicit reference to θ in our notation. With this result in hand, we obtain the derivative of the log likelihood:

$$\frac{\partial l}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C(x_C)} - N \frac{p(x_C)}{\psi_C(x_C)}, \tag{9.50}$$

where we can assume without loss of generality that $\psi_C(x_C) > 0$ (see Exercise **??**).

Setting Eq. (9.50) equal to zero, we obtain:

$$\hat{p}_{ML}(x_C) = \frac{1}{N} m(x_C). \tag{9.51}$$

Defining the *empirical distribution* $\tilde{p}(x) \triangleq m(x)/N$, so that $\tilde{p}(x_C) \triangleq m(x_C)/N$ is a marginal under the empirical distribution, we can rewrite the result as:

$$\hat{p}_{ML}(x_C) = \tilde{p}(x_C). \tag{9.52}$$

Thus we have the following important characterization of maximum likelihood estimates—*for each clique $C \in \mathcal{C}$, the model marginals must be equal to the empirical marginals.*

While this result constrains maximum likelihood models, it leaves us somewhat short of our goal. How do we find maximum likelihood estimates of the *parameters*? Eq. (9.52) provides us with a system of equations (by ranging over all $C \in \mathcal{C}$) that constrains the maximum likelihood estimates, but the parameters $\psi_C(x_C)$ themselves appear implicitly in these equations.

It turns out that for some graphs, in particular for a special class of graphs known as *decomposable graphs*, the situation is rather benign. Indeed, for these graphs, the maximum likelihood estimation problem decouples, and we can write down maximum likelihood estimates by inspection.

9.3.3 Decomposable models

Consider the example shown in Figure 9.3, a Markov chain on $\{X_1, X_2, X_3\}$. The probability model can be written as:

$$p(x_\mathcal{V}) = \frac{1}{Z} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3) \tag{9.53}$$

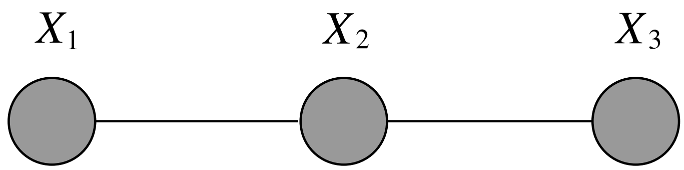

Figure 9.3: A Markov chain.

and the sufficient statistics under IID sampling are the empirical marginals $\tilde{p}(x_1, x_2)$ and $\tilde{p}(x_2, x_3)$.

For a model to be a maximum likelihood model, we require the marginals to equal the empirical marginals. How do we set the parameters so as to achieve this result?

Let us make the following guess:

$$\hat{p}_{ML}(x_1, x_2, x_3) = \frac{\tilde{p}(x_1, x_2)\tilde{p}(x_2, x_3)}{\tilde{p}(x_2)}, \tag{9.54}$$

where $\tilde{p}(x_2) = \sum_{x_1} \tilde{p}(x_1, x_2) = \sum_{x_3} \tilde{p}(x_2, x_3)$. That this is a good guess is readily verified:

$$\hat{p}_{ML}(x_1, x_2) = \sum_{x_3} \hat{p}_{ML}(x_1, x_2, x_3) = \tilde{p}(x_1, x_2) \tag{9.55}$$

$$\hat{p}_{ML}(x_2, x_3) = \sum_{x_1} \hat{p}_{ML}(x_1, x_2, x_3) = \tilde{p}(x_2, x_3), \tag{9.56}$$

and we see that the model marginals are indeed equal to the empirical marginals on the cliques $\mathcal{C}$.

Moreover, we can also easily match the terms in the guessed distribution to the parameters. For example, we can let:

$$\hat{\psi}_{12,ML}(x_1, x_2) = \tilde{p}(x_1, x_2) \tag{9.57}$$

and

$$\hat{\psi}_{23,ML}(x_2, x_3) = \frac{\tilde{p}(x_2, x_3)}{\tilde{p}(x_2)}, \tag{9.58}$$

which together imply that $Z = 1$. There are obviously many other sets of parameter estimates that yield the same joint distribution; these are all maximum likelihood estimates.

That this approach cannot work in general, however, is shown by the graph in Figure 9.4(a). As we invite the reader to verify in Exercise **??**, the analogous ratio of the empirical marginals does not yield a maximum likelihood distribution in this case.

The problem is not simply due to an inability to handle graphs with loops. This can be seen by considering Figure 9.4(b). Here the reader can verify that the guess:

$$\hat{p}_{ML}(x_1, x_2, x_3, x_4) = \frac{\tilde{p}(x_1, x_2, x_3)\tilde{p}(x_2, x_3, x_4)}{\tilde{p}(x_2, x_3)} \tag{9.59}$$

fits the empirical marginals correctly.

There is a subtlety in this latter case, however. To recover parameter estimates from the factored form in Eq. (9.59) we need to have potentials that have the appropriate arguments. That

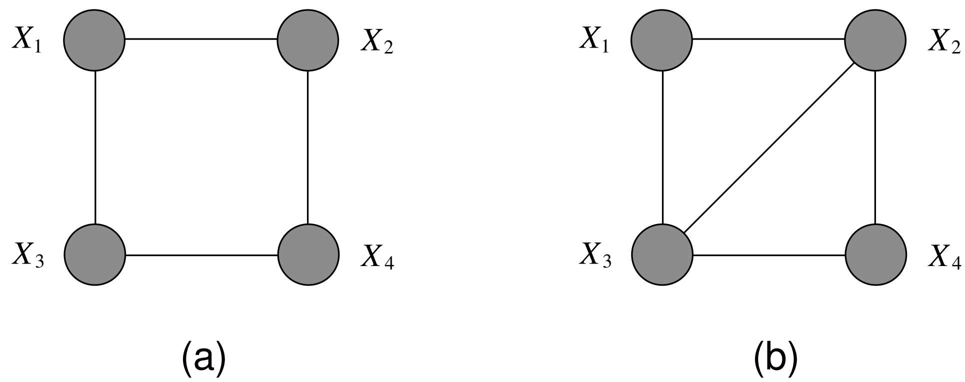

Figure 9.4: (a) A non-decomposable graph. (b) A decomposable graph.

is, in the case of Figure 9.4(b) we require that the model contain potentials $\psi_{123}(x_1, x_2, x_3)$ and $\psi_{234}(x_2, x_3, x_4)$. If the graph is parameterized with potentials that only refer to proper subsets of the maximal cliques, for example, pairwise potentials, then the guess in Eq. (9.59) is not in fact a maximum likelihood distribution because it is not in our model family.

In the remainder of this section, we briefly discuss the underlying concept that makes it possible to write maximum likelihood estimates by inspection in some cases but not in others. We will be brief because an understanding of this concept—that of a decomposable graph—requires additional machinery that we will not have in hand until Chapter 17. For completeness, however, let us define decomposability, and provide a recipe for constructing maximum likelihood estimates for decomposable graphs.

A graph is said to be *decomposable* if it can be recursively subdivided into disjoint sets A, B and S, where S separates A and B, and where S is complete.

The graph in Figure 9.3 is decomposable with $A = X_1$, $B = X_3$ and $S = X_2$. Similarly, the graph in Figure 9.4(b) is decomposable with $A = X_1$, $B = X_4$ and $S = \{X_2, X_3\}$. The reader can verify that the Figure 9.4(a) is not decomposable.

We can find maximum likelihood estimates for decomposable graphs by inspection, but only if the potentials are defined on maximal cliques. That is, our parameterization must be such that the set $\mathcal{C}$ ranges over the maximal cliques in the graph. Given this constraint, the recipe is the following:

- for every clique C, set the clique potential to the empirical marginal for that clique,
- for every non-empty intersection between cliques, associate an empirical marginal with that intersection, and divide that empirical marginal into the potential of one of the two cliques that form the intersection.

For example, for the graph in Figure 9.3, this recipe leads immediately to the estimates given in Eq. (9.57) and Eq. (9.58).

We have barely scratched the surface of the decomposability concept, but we hope to have piqued the reader's interest. To get a further hint of what is to come (in Chapter 17), note that there is another way to describe the differences between the graphs in Figures 9.3, 9.4(a), and 9.4(b): Only for Figure 9.4(a) is it impossible to eliminate the nodes in the graph without adding extra edges to the graph. This suggests a relationship between decomposability and graph elimination, and suggests (rightly) that decomposability lies with elimination at the core of the relationship between graphs and probabilities.

We now turn to an alternative algorithm for finding maximum likelihood estimates in all undirected graphs, whether decomposable and nondecomposable. In the decomposable case the algorithm turns out to converge in a finite number of iterations, updating each potential once. Indeed, in this case, the algorithm essentially implements the recipe for decomposable graphs that we described above. Thus the algorithm can be viewed as a general solution to the maximum likelihood estimation problem that takes advantage of the decomposable structure in the problem if it is present.

9.3.4 Iterative proportional fitting

A common strategy for solving systems of implicit equations is to "iterate," hoping that the iterations converge to a "fixed point"—a set of values that solve the original implicit equations. *Iterative proportional fitting (IPF)*, an algorithm for maximum likelihood estimation in undirected models, can be viewed as an example of such a strategy. However, it is also possible to say something stronger about IPF. In general, the iteration of fixed-point equations does not necessarily yield a convergent algorithm; moreover, even if the iteration converges, it is not necessarily the case that the algorithm behaves well in the sense of ascending an objective function at each step. IPF, however, *does* converge, and *does* behave well—the log likelihood is guaranteed to increase or remain the same after each IPF update. The facts about IPF follow from the fact that it is not only a fixed-point algorithm, but that it is also a *coordinate ascent algorithm*.

We begin with a simple, heuristic justification of IPF in terms of fixed-point iteration, and present the more satisfying justification in terms of coordinate ascent later in this section.

To develop an iterative approach to maximum likelihood estimation for undirected models, let us return to the gradient of the log likelihood, but retain the $\psi_C(x_C)$ factors. From Eq. (9.50) we have:

$$\frac{\tilde{p}(x_C)}{\psi_C(x_C)} = \frac{p(x_C)}{\psi_C(x_C)}, \tag{9.60}$$

where $\tilde{p}(x_C) \triangleq m(x_C)/N$ is the empirical marginal. Note that the parameter $\psi_C(x_C)$ appears explicitly in this equation in two places, but also appears implicitly in the marginal $p(x_C)$. We can obtain an iterative algorithm by holding the values of $\psi_C(x_C)$ fixed on the right-hand side of the equation, both in the numerator and the denominator, and solving for the free parameter $\psi_C(x_C)$ on the left-hand side.

In particular, let us denote the set of parameter estimates at iteration t by $\psi_C^{(t)}(x_C)$. Let $p^{(t)}(x)$ denote the joint probability distribution based on these parameter estimates. Rearranging Eq. (9.60), we define the following update for $\psi_C(x_C)$:

$$\psi_C^{(t+1)}(x_C) = \psi_C^{(t)}(x_C)\frac{\tilde{p}(x_C)}{p^{(t)}(x_C)}, \tag{9.61}$$

which is the IPF algorithm. The IPF algorithm cycles through all of the cliques $C \in \mathcal{C}$, applying Eq. (9.61) to each clique in turn; such a cycle constitutes a single iteration of the algorithm.

Before providing an interpretation of IPF in terms of coordinate ascent, let us explore some of the properties of the algorithm.

Properties of the IPF update equation

The IPF update equation in Eq. (9.61) has two interesting properties: (1) the marginal $p^{(t+1)}(x_C)$ is equal to the empirical marginal $\tilde{p}(x_C)$, and (2) the normalization factor Z remains constant across IPF updates.

To establish these properties, let us calculate the marginal probability of x_C for the updated distribution:

$$
\begin{aligned}
p^{(t+1)}(x_C) &= \sum_{x_{\mathcal{V}\setminus C}} p^{(t+1)}(x) && (9.62)\\
&= \sum_{x_{\mathcal{V}\setminus C}} \frac{1}{Z^{(t+1)}} \prod_D \psi_D^{(t+1)}(x_D) && (9.63)\\
&= \frac{1}{Z^{(t+1)}} \sum_{x_{\mathcal{V}\setminus C}} \psi_C^{(t+1)}(x_C) \prod_{D \neq C} \psi_D^{(t)}(x_D) && (9.64)\\
&= \frac{1}{Z^{(t+1)}} \sum_{x_{\mathcal{V}\setminus C}} \psi_C^{(t)}(x_C) \frac{\tilde{p}(x_C)}{p^{(t)}(x_C)} \prod_{D \neq C} \psi_D^{(t)}(x_D) && (9.65)\\
&= \frac{Z^{(t)}}{Z^{(t+1)}} \frac{\tilde{p}(x_C)}{p^{(t)}(x_C)} \sum_{x_{\mathcal{V}\setminus C}} \frac{1}{Z^{(t)}} \prod_D \psi_D^{(t)}(x_D) && (9.66)\\
&= \frac{Z^{(t)}}{Z^{(t+1)}} \frac{\tilde{p}(x_C)}{p^{(t)}(x_C)} \sum_{x_{\mathcal{V}\setminus C}} p^{(t)}(x) && (9.67)\\
&= \frac{Z^{(t)}}{Z^{(t+1)}} \frac{\tilde{p}(x_C)}{p^{(t)}(x_C)} p^{(t)}(x_C) && (9.68)\\
&= \frac{Z^{(t)}}{Z^{(t+1)}} \tilde{p}(x_C). && (9.69)
\end{aligned}
$$

Note that $\tilde{p}(x_C)$ is a proportion and is therefore normalized, and $p^{(t+1)}(x_C)$ is normalized because we have divided explicitly by $Z^{(t+1)}$. Thus, summing both sides of our result with respect to x_C, we have:

$$Z^{(t+1)} = Z^{(t)}. \tag{9.70}$$

We see that the normalization factor Z indeed remains constant across IPF updates. Moreover, now that we know that the factor $Z^{(t)}/Z^{(t+1)}$ is equal to one, we see that we have also derived property (1):

$$p^{(t+1)}(x_C) = \tilde{p}(x_C). \tag{9.71}$$

That is, IPF finds a distribution such that the marginal with respect to the variables x_C is equal to the empirical marginal $\tilde{p}(x_C)$.

This interpretation of IPF accords well with the characterization of the maximum likelihood estimates that we gave in Section 9.3.2. We saw that for the maximum likelihood model, the model marginals are equal to the empirical marginals, for all $C \in \mathcal{C}$. IPF works toward the goal of equal model and empirical marginals, by equating a single model marginal and empirical marginal at a time.

The fact that the normalization factor Z stays constant across IPF iterations has another interesting consequence. As we ask the reader to verify (Exercise **??**), the constancy of Z implies that we can write IPF in terms of joint probabilities:

$$p^{(t+1)}(x_{\mathcal{V}}) = p^{(t)}(x_{\mathcal{V}}) \frac{\tilde{p}(x_C)}{p^{(t)}(x_C)}, \tag{9.72}$$

and indeed this is how IPF is often defined.[1] From this equation, we immediately obtain:

$$p^{(t+1)}(x_{\mathcal{V}}) = p^{(t)}(x_{\mathcal{V} \setminus C} \,|\, x_C) \tilde{p}(x_C), \tag{9.73}$$

which provides an interpretation of an IPF iteration as retaining the "old" conditional probability $p^{(t)}(x_{\mathcal{V} \setminus C} \,|\, x_C)$, while replacing the "old" marginal probability $p^{(t)}(x_C)$ with the "new" marginal $\tilde{p}(x_C)$.

IPF as coordinate ascent

Let us now derive IPF as a coordinate ascent algorithm. A "coordinate" in this setting is a potential function.

To derive a coordinate ascent algorithm, we take the derivative of the log likelihood with respect to the "coordinate" $\psi_C(x_C)$, for fixed C and varying x_C, and solve for the maximizing values of these parameters while holding the remaining potentials fixed. To carry out this calculation, let us return to the calculation of the gradient in Section 9.3.2, pausing the derivation in midstream. From Eq. (9.46) we have:

$$\frac{\partial l}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C(x_C)} - \frac{N}{Z} \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) \prod_{D \neq C} \psi_D(\tilde{x}_D). \tag{9.74}$$

Let us take some care here—the parameter $\psi_C(x_C)$ on the right-hand side of this equation is a *variable* whose maximizing value we wish to solve for, where the remaining parameters $\psi_D(\tilde{x}_D)$ are being held fixed. Adding superscripts to the latter to reflect the fact that they are being held fixed at iteration t of our algorithm, the gradient of the log likelihood becomes:

$$\frac{\partial l}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C(x_C)} - \frac{N}{Z} \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) \prod_{D \neq C} \psi_D^{(t)}(\tilde{x}_D). \tag{9.75}$$

[1] Treating Eq. (9.72) as a definition of the IPF algorithm leaves open the actual implementation of an IPF step in terms of the structural components of the model, and we view Eq. (9.61) as the better definition.

If we were to multiply and divide the second term by the "old" parameter value $\psi_C^{(t)}(\tilde{x}_C)$, the sum would appear to reduce to the marginal $p^{(t)}(x_C)$ much as in our earlier derivation. Setting to zero would yield the IPF update. However, we must not forget that Z also depends on $\psi_C(x_C)$. As $\psi_C(x_C)$ varies, Z varies, and in general the varying value of Z is *not* the correct normalization for the *old* values of the parameters.

We have seen, however, that for a particular value of $\psi_C(x_C)$, namely $\psi_C^{(t+1)}(x_C)$ as defined by the IPF update, we have $Z^{(t+1)} = Z^{(t)}$, which *is* the correct normalization for the old values of the parameters. Taking advantage of this fact, we obtain the following value of the derivative of log likelihood, where we now evaluate the derivative at $\psi_C^{(t+1)}(x_C)$:

$$\begin{aligned}
\frac{\partial l}{\partial \psi_C(x_C)} &= \frac{m(x_C)}{\psi_C^{(t+1)}(x_C)} - \frac{N}{Z^{(t+1)}} \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) \prod_{D \neq C} \psi_D^{(t)}(\tilde{x}_D) && (9.76)\\
&= \frac{m(x_C)}{\psi_C^{(t+1)}(x_C)} - \frac{N}{Z^{(t)}} \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) \prod_{D \neq C} \psi_D^{(t)}(\tilde{x}_D) && (9.77)\\
&= \frac{m(x_C)}{\psi_C^{(t+1)}(x_C)} - \frac{N}{\psi^{(t)}(x_C)} \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) \frac{1}{Z^{(t)}} \prod_{D} \psi_D^{(t)}(\tilde{x}_D) && (9.78)\\
&= \frac{m(x_C)}{\psi_C^{(t+1)}(x_C)} - \frac{N}{\psi^{(t)}(x_C)} p^{(t)}(x_C), && (9.79)
\end{aligned}$$

and we see that the IPF update equation:

$$\psi_C^{(t+1)}(x_C) = \psi^{(t)}(x_C) \frac{\tilde{p}(x_C)}{p^{(t)}(x_C)} \tag{9.80}$$

does indeed set the gradient of the log likelihood to zero, and thus is a coordinate ascent step.

View from the KL divergence

We have shown that the IPF algorithm is coordinate ascent in the log likelihood. A somewhat more elegant derivation of this result can be obtained via the KL divergence.

The KL divergence has a useful decomposition that reflects the decomposition of a joint distribution into the product of a marginal and a conditional. In particular, writing $p(x_A, x_B) = p(x_B \,|\, x_A) p(x_A)$, and $q(x_A, x_B) = q(x_B \,|\, x_A) q(x_A)$, we have:

$$D(p(x_A, x_B) \parallel q(x_A, x_B)) = D(p(x_A) \parallel q(x_A)) + \sum_{x_A} p(x_A) D(p(x_B \,|\, x_A) \parallel q(x_B \,|\, x_A)). \tag{9.81}$$

(This result is derived in Appendix XXX).

Recall that the problem of maximizing the likelihood is equivalent to that of minimizing the following KL divergence:

$$D(\tilde{p}(x) \parallel p(x \,|\, \theta)) = \sum_x \tilde{p}(x) \log \frac{\tilde{p}(x)}{p(x \,|\, \theta)}, \tag{9.82}$$

where $\tilde{p}(x)$ is the empirical distribution. (The reason that these problems are equivalent is because the (negative) log likelihood and the KL divergence differ by the term $\sum_x \tilde{p}(x) \log \tilde{p}(x)$, which is independent of θ).

Thus we wish to perform coordinate descent in $D(\tilde{p}(x) \parallel p(x \mid \theta))$. That is, we pick a clique C and adjust the clique potential $\psi_C(x_C)$ so as to minimize the KL divergence. Applying Eq. (9.81), we have:

$$D(\tilde{p}(x) \parallel p(x \mid \theta)) = D(\tilde{p}(x_C) \parallel p(x_C \mid \theta)) + \sum_{x_C} \tilde{p}(x_C) D(\tilde{p}(x_{\mathcal{V} \setminus C} \mid x_C) \parallel p(x_{\mathcal{V} \setminus C} \mid x_C, \theta)). \quad (9.83)$$

Changes to the clique potential $\psi_C(x_C)$ have no effect on the conditional distribution $p(x_{\mathcal{V} \setminus C} \mid x_C, \theta)$.[2] Thus, the second term in Eq. (9.83) is unaltered by changes to $\psi_C(x_C)$, and minimizing the KL divergence reduces to minimizing the first term. This term is minimized by setting the marginal $p(x_C \mid \theta)$ equal to the empirical marginal $\tilde{p}(x_C)$. But this is exactly what an IPF update achieves. Thus IPF is coordinate descent in $D(\tilde{p}(x) \parallel p(x \mid \theta))$ or, equivalently, coordinate ascent in the log likelihood.

9.3.5 Gradient ascent

An alternative to IPF is to perform gradient ascent on the log likelihood. In this case, given that the gradient is evaluated only at the current value of the parameters, no subtleties arise.

Evaluating the gradient at $\psi_C^{(t)}(x_C)$, we have:

$$\frac{\partial l}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C^{(t)}(x_C)} - \frac{N}{\psi^{(t)}(x_C)} p^{(t)}(x_C), \quad (9.84)$$

which leads to the following gradient ascent algorithm:

$$\psi_C^{(t+1)}(x_C) = \psi_C^{(t)}(x_C) + \frac{\rho}{\psi_C^{(t)}(x_C)} \left(\tilde{p}(x_C) - p^{(t)}(x_C) \right), \quad (9.85)$$

where ρ is a step size. We see that the difference between the empirical marginals and the model marginals drives the algorithm.

Gradient ascent has the advantage compared to IPF that all of the parameters can be adjusted simultaneously, although we will present a variant of IPF in Chapter 19 that also allows parameters to be adjusted simultaneously, so this advantage is somewhat diminished. Disadvantages of the gradient ascent approach include the need to choose a step size, and, even more seriously, the fact that the normalization factor Z does not remain constant, but must be recalculated anew after each iteration.

[2]This is proved by simply writing out the conditional distribution and noting that $\psi_C(x_C)$ cancels in the numerator and denominator; see Exercise ?? for details.

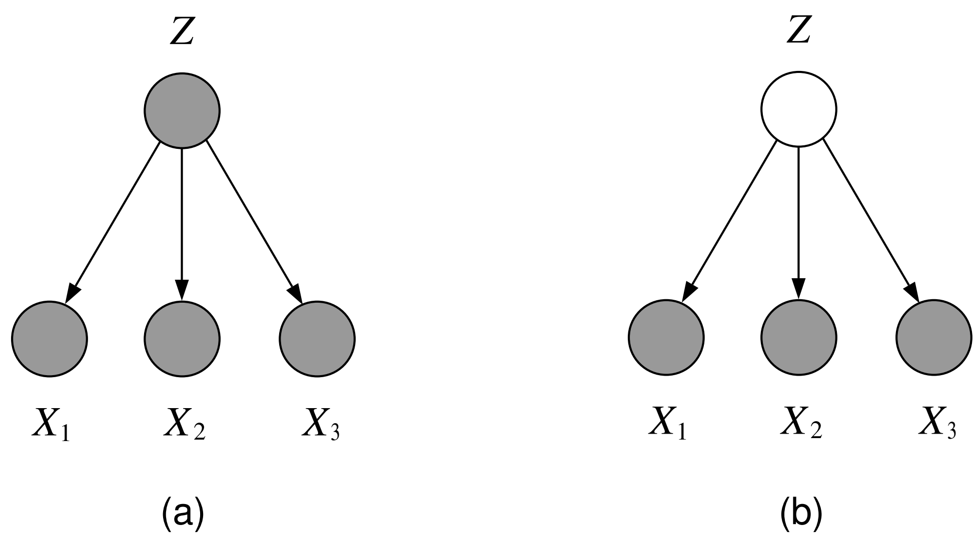

Figure 9.5: (a) The nodes X_i are conditionally independent given Z. (b) When Z is latent the variables X_i are no longer independent and this implies that the parameters are coupled in the log likelihood.

9.4 Latent variables

The algorithms that we have presented in this chapter all rely on the assumption of complete data. In the setting of complete data the likelihood is a product and the log likelihood is therefore a sum, where each of the parameters appear in different terms. This leads to a decoupling of the parameter estimation problem.

When some of the variables are unobserved, that is, when we have *latent variables*, this happy situation breaks down. When there are latent variables, the likelihood is a marginal probability, obtained by summing or integrating over the latent variables. The log likelihood is the logarithm of this sum, and the logarithm is prevented from moving past the sum to act on the product of potentials. The parameter estimation problem does not decouple.

Consider the graphical model shown in Figure 9.5(a). Here the nodes X_i are conditionally independent given Z; thus, when Z is observed the log likelihood decouples:

$$\begin{aligned} l(\theta; x, z) &= \log p(x, z \mid \theta) && (9.86)\\ &= \log\left[p(z \mid \theta_z) p(x_1 \mid z, \theta_1) p(x_2 \mid z, \theta_2) p(x_3 \mid z, \theta_3)\right] && (9.87)\\ &= \log p(z \mid \theta_z) + \log p(x_1 \mid z, \theta_1) + \log p(x_2 \mid z, \theta_2) + \log p(x_3 \mid z, \theta_3). && (9.88) \end{aligned}$$

If, on the other hand, Z is latent, as shown in Figure 9.5(b), the log likelihood does not decouple:

$$\begin{aligned} l(\theta; x) &= \log \sum_z p(x, z \mid \theta) && (9.89)\\ &= \log \sum_z \left[p(z \mid \theta_z) p(x_1 \mid z, \theta_1) p(x_2 \mid z, \theta_2) p(x_3 \mid z, \theta_3)\right]. && (9.90) \end{aligned}$$

Adjusting one parameter, say θ_1, has an effect on all of the other parameters. This makes good sense probabilistically—our uncertainty about Z should be reflected in a probabilistic dependence among the variables X_i and hence among the estimates of the parameters θ_i. This coupling is unfortunate from a computational point of view, however, in that it complicates the task of parameter estimation.

In Chapter 11 we present a general procedure for dealing with latent variable problems known as the *Expectation-Maximization (EM)* algorithm. The EM algorithm in essence allows us to treat latent variable problems with complete data tools. In particular, the EM algorithm allows us to solve the maximum likelihood problem for the graph in Figure 9.5(b) by solving a sequence of maximum likelihood problems based on the graph in Figure 9.5(a).

To give the briefest hint of what is involved in the EM algorithm, essentially EM makes use of convexity to move the logarithm past the sum in coupled log likelihoods such as Eq. (9.90). This decouples the problem and allows the complete data tools of the current chapter to be brought into play. Exactly how this is done in general is the story for Chapter 11. In the meantime, in the following chapter we work through a specific example of a latent variable problem, developing a special case of the EM algorithm that is particularly simple and intuitive.

9.5 Summary

In the current chapter we have discussed parameter estimation in completely observed graphical models, treating both directed graphs and undirected graphs.

For directed graphs, we saw that the log likelihood decouples into separate terms, one for each parameter, and that the problem of parameter estimation therefore also decouples. Essentially, one collects the sufficient statistics associated with each node and its parents and estimates the parameter vector at that node using those sufficient statistics. This is done independently at each node in the graph. In particular, if the probability model at each node is a generalized linear model, then we can run the IRLS algorithm independently at each node. This is a Newton algorithm on the graph as a whole.

For undirected graphs, we distinguish between decomposable models and nondecomposable models. Decomposable models are the easy case; for decomposable models we can solve for maximum likelihood estimates analytically. We have only briefly outlined the reasons for this in the current chapter; in Chapter 17, we provide a deeper discussion of decomposability, revealing the important relationship between decomposability and the general approach to inference known as the junction tree algorithm. We then return to the problem of parameter estimation in Chapter 20 and nail down the results that we have only sketched here.

For nondecomposable models, we discussed the iterative proportional fitting (IPF) approach to parameter estimation. IPF takes the form of a simple scaling algorithm in which the potentials are multiplied by a ratio of marginal probabilities. We showed that IPF can be viewed as a fixed point algorithm and also as a coordinate ascent algorithm. These themes—iterative algorithms, multiplicative updates of potentials, and coordinate ascent—will appear in several guises in the following chapters, and we will see several IPF-like algorithms as we proceed.

9.6 Historical remarks and bibliography