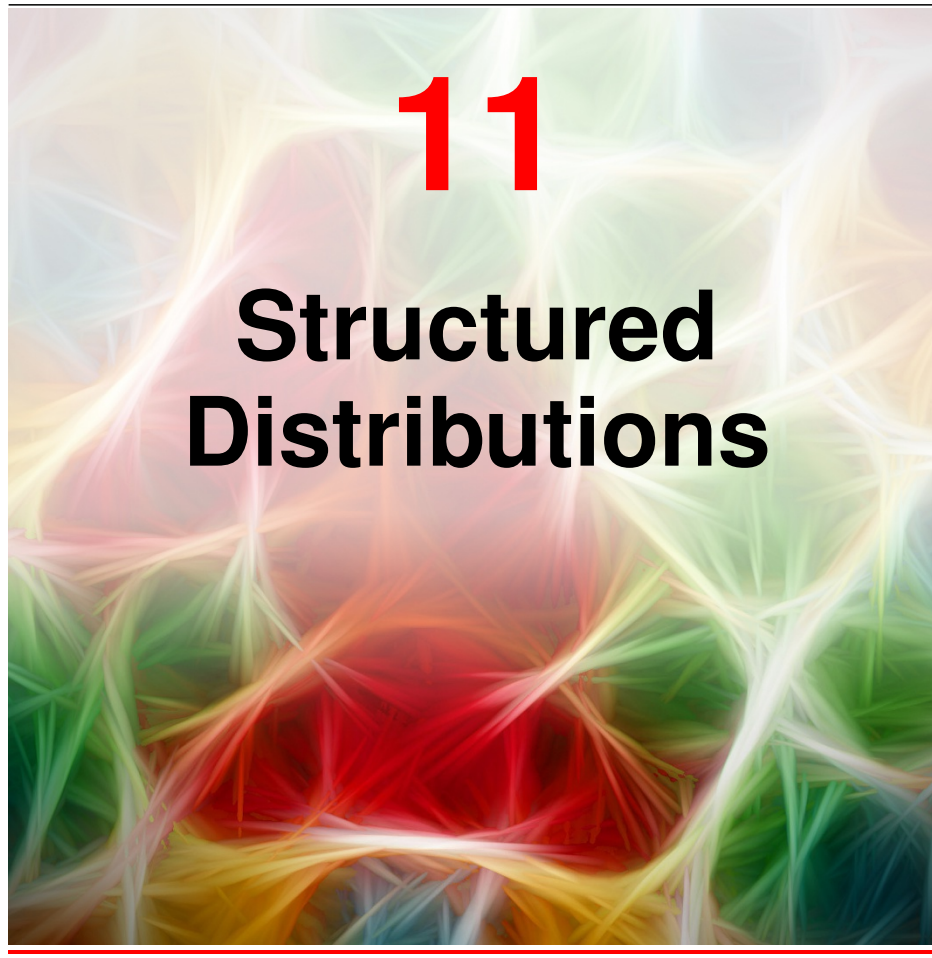


11 Structured Distributions

We have seen that probability forms one of the most important foundational concepts for deep learning. For example, a neural network used for binary classification is described by a conditional probability distribution of the form

$$p(t|\mathbf{x}, \mathbf{w}) = y(\mathbf{x}, \mathbf{w})^t \left\{1 - y(\mathbf{x}, \mathbf{w})\right\}^{(1-t)} \tag{11.1}$$

where $y(\mathbf{x}, \mathbf{w})$ represents a neural network function that takes a vector $\mathbf{x}$ as input and is governed by a vector $\mathbf{w}$ of learnable parameters. The corresponding cross-entropy likelihood forms the basis for defining an error function used to train the neural network. Although the network function might be extremely complex, the conditional distribution in (11.1) has a simple form. However, there are many important deep learning models that have a much richer probabilistic structure, such as large language models, normalizing flows, variational autoencoders, diffusion models, and many others. To describe and exploit this structure, we introduce a powerful

C. M. Bishop, H. Bishop, *Deep Learning*, https://doi.org/10.1007/978-3-031-45468-4_11

framework called *probabilistic graphical models*, or simply *graphical models*, which allows structured probability distributions to be expressed in graphical form. When combined with neural networks to define associated probability distributions, graphical models offer huge flexibility when creating sophisticated models that can be trained end to end using stochastic gradient descent in which gradients are evaluated efficiently using auto-differentiation. In this chapter, we will focus on the core concepts of graphical models needed for applications in deep learning, whereas a more comprehensive treatment of graphical models for machine learning can be found in Bishop (2006).

11.1. Graphical Models

Section 2.1

Probability theory can be expressed in terms of two simple equations known as the *sum rule* and the *product rule*. All of the probabilistic manipulations discussed in this book, no matter how complex, amount to repeated application of these two equations. In principle, we could therefore formulate and use complex probabilistic models purely by using algebraic manipulation. However, we will find it advantageous to augment the analysis using diagrammatic representations of probability distributions, as these offer several useful properties:

1. They provide a simple way to visualize the structure of a probabilistic model and can be used to design and motivate new models.
2. Insights into the properties of the model, including conditional independence properties, can be obtained by inspecting the graph.
3. The complex computations required to perform inference and learning in sophisticated models can be expressed in terms of graphical operations, such as message-passing, in which the underlying mathematical operations are carried out implicitly.

Although such graphical models have nodes and edges much like neural network diagrams, their interpretation is specifically probabilistic and carries a richer semantics. To help avoid confusion, in this book we denote neural network diagrams in blue and probabilistic graphical models in red.

11.1.1 Directed graphs

A graph comprises *nodes*, also called *vertices*, connected by *links*, also known as *edges*. In a probabilistic graphical model, each node represents a random variable, and the links express probabilistic relationships between these variables. The graph then captures the way in which the joint distribution over all the random variables can be decomposed into a product of factors each depending only on a subset of the variables. In this chapter we will focus on graphical models in which the links of the graphs have a particular direction indicated by arrows. These are known as *directed graphical models* and are also called *Bayesian networks* or *Bayes nets*.

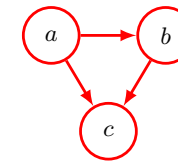


Figure 11.1 A directed graphical model representing the joint probability distribution over three variables a, b, and c, corresponding to the decomposition on the right-hand side of (11.3).

The other major class of graphical models are *Markov random fields*, also known as *undirected graphical models*, in which the links do not carry arrows and have no directional significance. Directed graphs are useful for expressing causal relationships between random variables, whereas undirected graphs are better suited to expressing soft constraints between random variables. Both directed and undirected graphs can be viewed as special cases of a representation called *factor graphs*. From now on we focus our attention on directed graphical models. Note, however, that undirected graphs, without the probabilistic interpretation, will also arise in our discussion of *graph neural networks* in which the nodes represent deterministic variables as in standard neural networks.

Chapter 13

11.1.2 Factorization

To motivate the use of directed graphs to describe probability distributions, consider first an arbitrary joint distribution $p(a, b, c)$ over three variables a, b, and c. Note that at this stage, we do not need to specify anything further about these variables, such as whether they are discrete or continuous. Indeed, one of the powerful aspects of graphical models is that a specific graph can make probabilistic statements for a broad class of distributions. By application of the product rule of probability (2.9), we can write the joint distribution in the form

$$p(a, b, c) = p(c|a, b)p(a, b). \tag{11.2}$$

A second application of the product rule, this time to the second term on the right-hand side of (11.2), gives

$$p(a, b, c) = p(c|a, b)p(b|a)p(a). \tag{11.3}$$

Note that this decomposition holds for any choice of the joint distribution. We now represent the right-hand side of (11.3) in terms of a simple graphical model as follows. First we introduce a node for each of the random variables a, b, and c and associate each node with the corresponding conditional distribution on the right-hand side of (11.3). Then, for each conditional distribution we add directed links (depicted as arrows) from the nodes corresponding to the variables on which the distribution is conditioned. Thus, for the factor $p(c|a, b)$, there will be links from nodes a and b to node c, whereas for the factor $p(a)$, there will be no incoming links. The result is the graph shown in Figure 11.1. If there is a link going from node a to node b, then we say that node a is the *parent* of node b, and we say that node b is the *child* of node a. Note that we will not make any formal distinction between a node and the variable to which it corresponds but will simply use the same symbol to refer to both.

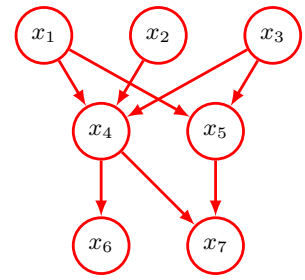


Figure 11.2 Example of a directed graph describing the joint distribution over variables $x_1, \ldots, x_7$. The corresponding decomposition of the joint distribution is given by (11.5).

An important point to note about (11.3) is that the left-hand side is symmetrical with respect to the three variables a, b, and c, whereas the right-hand side is not. In making the decomposition in (11.3), we have implicitly chosen a particular ordering, namely a, b, c, and had we chosen a different ordering we would have obtained a different decomposition and hence a different graphical representation.

For the moment let us extend the example of Figure 11.1 by considering the joint distribution over K variables given by $p(x_1, \ldots, x_K)$. By repeated application of the product rule of probability, this joint distribution can be written as a product of conditional distributions, one for each of the variables:

$$p(x_1, \ldots, x_K) = p(x_K|x_1, \ldots, x_{K-1}) \ldots p(x_2|x_1)p(x_1). \tag{11.4}$$

For a given choice of K, we can again represent this as a directed graph having K nodes, one for each conditional distribution on the right-hand side of (11.4), with each node having incoming links from all lower numbered nodes. We say that this graph is *fully connected* because there is a link between every pair of nodes.

So far, we have worked with completely general joint distributions, and so their factorization, and associated representation as fully connected graphs, will be applicable to any choice of distribution. As we will see shortly, it is the *absence* of links in the graph that conveys interesting information about the properties of the class of distributions that the graph represents. Consider the graph shown in Figure 11.2. Note that it is not a fully connected graph because, for instance, there is no link from x_1 to x_2 or from x_3 to x_7. We take this graph and extract the corresponding representation of the joint probability distribution written in terms of the product of a set of conditional distributions, one for each node in the graph. Each such conditional distribution will be conditioned only on the parents of the corresponding node in the graph. For instance, x_5 will be conditioned on x_1 and x_3. The joint distribution of all seven variables is therefore given by

$$p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5). \tag{11.5}$$

The reader should take a moment to study carefully the correspondence between (11.5) and Figure 11.2.

We can now state in general terms the relationship between a given directed graph and the corresponding distribution over the variables. The joint distribution defined by a graph is given by the product, over all of the nodes of the graph, of

a conditional distribution for each node conditioned on the variables corresponding to the parents of that node in the graph. Thus, for a graph with K nodes, the joint distribution is given by

$$p(x_1, \ldots, x_K) = \prod_{k=1}^{K} p(x_k|\mathrm{pa}(k)) \tag{11.6}$$

where $\mathrm{pa}(k)$ denotes the set of parents of x_k. This key equation expresses the *factorization* properties of the joint distribution for a directed graphical model. Although we have considered each node to correspond to a single variable, we can equally well associate sets of variables and vector-valued or tensor-valued variables with the nodes of a graph. It is easy to show that the representation on the right-hand side of (11.6) is always correctly normalized provided the individual conditional distributions are normalized.

Exercise 11.1

The directed graphs that we are considering are subject to an important restriction, namely that there must be no *directed cycles*. In other words, there are no closed paths within the graph such that we can move from node to node along links following the direction of the arrows and end up back at the starting node. Such graphs are also called *directed acyclic graphs*, or *DAGs*. This is equivalent to the statement that there exists an ordering of the nodes such that there are no links that go from any node to any lower-numbered node.

Exercise 11.2

11.1.3 Discrete variables

We have discussed the importance of probability distributions that are members of the exponential family, and we have seen that this family includes many well-known distributions as special cases. Although such distributions are relatively simple, they form useful building blocks for constructing more complex probability distributions, and the framework of graphical models is very useful in expressing the way in which these building blocks are linked together. There are two particular choices for the component distributions that are widely used, corresponding to discrete variables and to Gaussian variables. We begin by examining the discrete case.

Section 3.4

The probability distribution $p(\mathbf{x}|\boldsymbol{\mu})$ for a single discrete variable $\mathbf{x}$ having K possible states (using the 1-of-K representation) is given by

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k} \tag{11.7}$$

and is governed by the parameters $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)^{\mathrm{T}}$. Due to the constraint $\sum_k \mu_k = 1$, only $K - 1$ values for μ_k need to be specified to define the distribution.

Now suppose that we have two discrete variables, $\mathbf{x}_1$ and $\mathbf{x}_2$, each of which has K states, and we wish to model their joint distribution. We denote the probability of observing both $x_{1k} = 1$ and $x_{2l} = 1$ by the parameter μ_{kl}, where x_{1k} denotes the

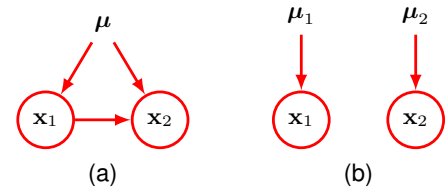


Figure 11.3 (a) This fully connected graph describes a general distribution over two K-state discrete variables having a total of $K^2 - 1$ parameters. (b) By dropping the link between the nodes, the number of parameters is reduced to $2(K - 1)$.

kth component of $\mathbf{x}_1$, and similarly for x_{2l}. The joint distribution can be written

$$p(\mathbf{x}_1, \mathbf{x}_2|\boldsymbol{\mu}) = \prod_{k=1}^{K}\prod_{l=1}^{K} \mu_{kl}^{x_{1k}x_{2l}}.$$

Because the parameters μ_{kl} are subject to the constraint $\sum_k \sum_l \mu_{kl} = 1$, this distribution is governed by $K^2 - 1$ parameters. It is easily seen that the total number of parameters that must be specified for an arbitrary joint distribution over M variables is $K^M - 1$ and therefore grows exponentially with the number M of variables.

Using the product rule, we can factor the joint distribution $p(\mathbf{x}_1, \mathbf{x}_2)$ in the form $p(\mathbf{x}_2|\mathbf{x}_1)p(\mathbf{x}_1)$, which corresponds to a two-node graph with a link going from the $\mathbf{x}_1$ node to the $\mathbf{x}_2$ node as shown in Figure 11.3(a). The marginal distribution $p(\mathbf{x}_1)$ is governed by $K - 1$ parameters, as before. Similarly, the conditional distribution $p(\mathbf{x}_2|\mathbf{x}_1)$ requires the specification of $K - 1$ parameters for each of the K possible values of $\mathbf{x}_1$. The total number of parameters that must be specified in the joint distribution is therefore $(K - 1) + K(K - 1) = K^2 - 1$ as before.

Now suppose that the variables $\mathbf{x}_1$ and $\mathbf{x}_2$ are independent, corresponding to the graphical model shown in Figure 11.3(b). Each variable is then described by a separate discrete distribution, and the total number of parameters would be $2(K-1)$. For a distribution over M independent discrete variables, each having K states, the total number of parameters would be $M(K - 1)$, which therefore grows linearly with the number of variables. From a graphical perspective, we have reduced the number of parameters by dropping links in the graph, at the expense of having a more restricted class of distributions.

More generally, if we have M discrete variables $\mathbf{x}_1, \ldots, \mathbf{x}_M$, we can model the joint distribution using a directed graph with one variable for each node. The conditional distribution at each node is given by a set of non-negative parameters subject to the usual normalization constraint. If the graph is fully connected, then we have a completely general distribution having $K^M - 1$ parameters, whereas if there are no links in the graph, the joint distribution factorizes into the product of the marginal distributions, and the total number of parameters is $M(K - 1)$. Graphs having intermediate levels of connectivity allow for more general distributions than the fully factorized one while requiring fewer parameters than the general joint distribution. As an illustration, consider the chain of nodes shown in Figure 11.4. The marginal distribution $p(\mathbf{x}_1)$ requires $K - 1$ parameters, whereas each of the $M - 1$ conditional distributions $p(\mathbf{x}_i|\mathbf{x}_{i-1})$, for $i = 2, \ldots, M$, requires $K(K - 1)$ parameters.

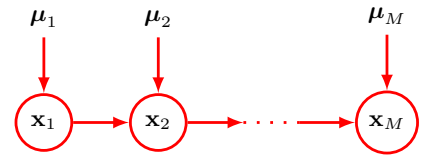


Figure 11.4 This chain of M discrete nodes, each having K states, requires the specification of $K-1+(M-1)K(K-1)$ parameters, which grows linearly with the length M of the chain. In contrast, a fully connected graph of M nodes would have K^M-1 parameters, which grows exponentially with M.

This gives a total parameter count of $K-1+(M-1)K(K-1)$, which is quadratic in K and which grows linearly (rather than exponentially) with the length M of the chain.

An alternative way to reduce the number of independent parameters in a model is by *sharing* parameters (also known as *tying* of parameters). For instance, in the chain example of Figure 11.4, we can arrange that all the conditional distributions $p(\mathbf{x}_i|\mathbf{x}_{i-1})$, for $i=2,\ldots,M$, are governed by the same set of $K(K-1)$ parameters, giving the model shown in Figure 11.5. Together with the $K-1$ parameters governing the distribution of $\mathbf{x}_1$, this gives a total of K^2-1 parameters that must be specified to define the joint distribution.

Another way of controlling the exponential growth of the number of parameters in models of discrete variables is to use parameterized representations for the conditional distributions instead of complete tables of conditional probability values. To illustrate this idea, consider the graph in Figure 11.6 in which all the nodes represent binary variables. Each of the parent variables x_i is governed by a single parameter μ_i representing the probability $p(x_i=1)$, giving M parameters in total for the parent nodes. The conditional distribution $p(y|x_1,\ldots,x_M)$, however, would require 2^M parameters representing the probability $p(y=1)$ for each of the 2^M possible settings of the parent variables. Thus, in general the number of parameters required to specify this conditional distribution will grow exponentially with M. We can obtain a more parsimonious form for the conditional distribution by using a logistic sigmoid function acting on a linear combination of the parent variables, giving

Section 3.4

$$p(y=1|x_1,\ldots,x_M)=\sigma\left(w_0+\sum_{i=1}^{M}w_ix_i\right)=\sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x}) \tag{11.8}$$

where $\sigma(a)=(1+\exp(-a))^{-1}$ is the logistic sigmoid, $\mathbf{x}=(x_0,x_1,\ldots,x_M)^{\mathrm{T}}$ is an $(M+1)$-dimensional vector of parent states augmented with an additional variable x_0 whose value is clamped to 1, and $\mathbf{w}=(w_0,w_1,\ldots,w_M)^{\mathrm{T}}$ is a vector of $M+1$ parameters. This is a more restricted form of conditional distribution than the general case but is now governed by a number of parameters that grows linearly with M. In

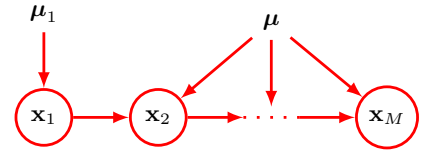


Figure 11.5 As in Figure 11.4 but with a single set of parameters $\boldsymbol{\mu}$ shared amongst all the conditional distributions $p(\mathbf{x}_i|\mathbf{x}_{i-1})$.

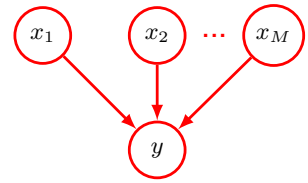


Figure 11.6 A graph comprising M parents $x_1, \ldots, x_M$ and a single child y, used to illustrate the idea of parameterized conditional distributions for discrete variables.

this sense, it is analogous to the choice of a restrictive form of covariance matrix (for example, a diagonal matrix) in a multivariate Gaussian distribution.

11.1.4 Gaussian variables

We now turn to graphical models in which the nodes represent continuous variables having Gaussian distributions. Each distribution is conditioned on the state of its parents in the graph. That dependence could take many forms, and here we focus on a specific choice in which the mean of each Gaussian is some linear function of the states of the Gaussian parent variables. This leads to a class of models called *linear-Gaussian models*, which include many cases of practical interest such as probabilistic principal component analysis, factor analysis, and linear dynamical systems (Roweis and Ghahramani, 1999).

Section 16.2

Consider an arbitrary directed acyclic graph over D variables in which node i represents a single continuous random variable x_i having a Gaussian distribution. The mean of this distribution is taken to be a linear combination of the states of its parent nodes $\mathrm{pa}(i)$ of node i:

$$p(x_i|\mathrm{pa}(i)) = \mathcal{N}\left(x_i \,\middle|\, \sum_{j\in\mathrm{pa}(i)} w_{ij}x_j + b_i, v_i\right) \tag{11.9}$$

where w_{ij} and b_i are parameters governing the mean and v_i is the variance of the conditional distribution for x_i. The log of the joint distribution is then the log of the product of these conditionals over all nodes in the graph and hence takes the form

$$\ln p(\mathbf{x}) = \sum_{i=1}^{D} \ln p(x_i|\mathrm{pa}(i)) \tag{11.10}$$

$$= -\sum_{i=1}^{D} \frac{1}{2v_i}\left(x_i - \sum_{j\in\mathrm{pa}(i)} w_{ij}x_j - b_i\right)^2 + \text{const} \tag{11.11}$$

where $\mathbf{x} = (x_1, \ldots, x_D)^{\mathrm{T}}$ and 'const' denotes terms independent of $\mathbf{x}$. We see that this is a quadratic function of the components of $\mathbf{x}$, and hence the joint distribution $p(\mathbf{x})$ is a multivariate Gaussian.

We can find the mean and covariance of this joint distribution as follows. The mean of each variable is given by the recursion relation

Exercise 11.6

$$\mathbb{E}[x_i] = \sum_{j\in\mathrm{pa}(i)} w_{ij}\mathbb{E}[x_j] + b_i. \tag{11.12}$$

Figure 11.7 A directed graph over three Gaussian variables with one missing link.

and so we can find the components of $\mathbb{E}[\mathbf{x}] = (\mathbb{E}[x_1], \ldots, \mathbb{E}[x_D])^{\mathrm{T}}$ by starting at the lowest numbered node and working recursively through the graph, where we assume that the nodes are numbered such that each node has a higher number than its parents. Similarly, the elements of the covariance matrix of the joint distribution satisfy a recursion relation of the form

Exercise 11.7

$$\mathrm{cov}[x_i, x_j] = \sum_{k \in \mathrm{pa}(j)} w_{jk} \mathrm{cov}[x_i, x_k] + I_{ij} v_j \tag{11.13}$$

and so the covariance can similarly be evaluated recursively starting from the lowest numbered node.

We now consider two extreme cases of possible graph structures. First, suppose that there are no links in the graph, which therefore comprises D isolated nodes. In this case, there are no parameters w_{ij} and so there are just D parameters b_i and D parameters v_i. From the recursion relations (11.12) and (11.13), we see that the mean of $p(\mathbf{x})$ is given by $(b_1, \ldots, b_D)^{\mathrm{T}}$ and the covariance matrix is diagonal of the form $\mathrm{diag}(v_1, \ldots, v_D)$. The joint distribution has a total of $2D$ parameters and represents a set of D independent univariate Gaussian distributions.

Now consider a fully connected graph in which each node has all lower numbered nodes as parents. In this case the total number of independent parameters $\{w_{ij}\}$ and $\{v_i\}$ in the covariance matrix is $D(D+1)/2$ corresponding to a general symmetric covariance.

Exercise 11.8

Graphs having some intermediate level of complexity correspond to joint Gaussian distributions with partially constrained covariance matrices. Consider for example the graph shown in Figure 11.7, which has a link missing between variables x_1 and x_3. Using the recursion relations (11.12) and (11.13), we see that the mean and covariance of the joint distribution are given by

Exercise 11.9

$$\boldsymbol{\mu} = (b_1, b_2 + w_{21}b_1, b_3 + w_{32}b_2 + w_{32}w_{21}b_1)^{\mathrm{T}} \tag{11.14}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} v_1 & w_{21}v_1 & w_{32}w_{21}v_1 \\ w_{21}v_1 & v_2 + w_{21}^2 v_1 & w_{32}(v_2 + w_{21}^2 v_1) \\ w_{32}w_{21}v_1 & w_{32}(v_2 + w_{21}^2 v_1) & v_3 + w_{32}^2(v_2 + w_{21}^2 v_1) \end{pmatrix}. \tag{11.15}$$

We can readily extend the linear-Gaussian graphical model to a situation in which the nodes of the graph represent multivariate Gaussian variables. In this case, we can write the conditional distribution for node i in the form

$$p(\mathbf{x}_i | \mathrm{pa}(i)) = \mathcal{N}\left(\mathbf{x}_i \,\middle|\, \sum_{j \in \mathrm{pa}(i)} \mathbf{W}_{ij}\mathbf{x}_j + \mathbf{b}_i, \boldsymbol{\Sigma}_i\right) \tag{11.16}$$

where now $\mathbf{W}_{ij}$ is a matrix (which is non-square if $\mathbf{x}_i$ and $\mathbf{x}_j$ have different dimensionality). Again it is easy to verify that the joint distribution over all variables is Gaussian.

Exercise 11.10

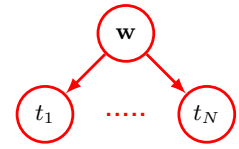


Figure 11.8 Directed graphical model representing the binary classifier model described by the joint distribution (11.17) showing only the stochastic variables $\{t_1, \ldots, t_N\}$ and $\mathbf{w}$.

11.1.5 Binary classifier

Section 2.6.2

We can illustrate the use of directed graphs to describe probability distributions using a two-class classifier model with Gaussian prior over the learnable parameters. We can write this in the form

$$p(\mathbf{t}, \mathbf{w}|\mathbf{X}, \lambda) = p(\mathbf{w}|\lambda) \prod_{n=1}^{N} p(t_n|\mathbf{w}, \mathbf{x}_n) \tag{11.17}$$

where $\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$ is the vector of target values, $\mathbf{X}$ is the data matrix with rows $\mathbf{x}_1^{\mathrm{T}}, \ldots \mathbf{x}_N^{\mathrm{T}}$, and the distribution $p(t|\mathbf{x}, \mathbf{w})$ is given by (11.1). We also assume a Gaussian prior over the parameter vector $\mathbf{w}$ given by

$$p(\mathbf{w}|\lambda) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda\mathbf{I}). \tag{11.18}$$

The stochastic variables in this model are $\{t_1, \ldots, t_N\}$ and $\mathbf{w}$. In addition, this model contains the noise variance σ^2 and the hyperparameter λ, both of which are parameters of the model rather than stochastic variables. If we consider for a moment only the stochastic variables, then the distribution given by (11.17) can be represented by the graphical model shown in Figure 11.8.

When we start to deal with more complex models, it becomes inconvenient to have to write out multiple nodes of the form $t_1, \ldots, t_N$ explicitly as in Figure 11.8. We therefore introduce a graphical notation that allows such multiple nodes to be expressed more compactly. We draw a single representative node t_n and then surround this with a box, called a *plate*, labelled with N to indicate that there are N nodes of this kind. Rewriting the graph of Figure 11.8 in this way, we obtain the graph shown in Figure 11.9.

11.1.6 Parameters and observations

We will sometimes find it helpful to make the parameters of a model, as well as its stochastic variables, explicit in the graphical representation. To do this, we will adopt the convention that random variables are denoted by open circles and deterministic parameters are denoted by floating variables. If we take the graph of

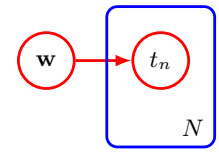


Figure 11.9 An alternative, more compact, representation of the graph shown in Figure 11.8 in which we have introduced a *plate* (the box labelled N) that represents N nodes of which only a single example t_n is shown explicitly.

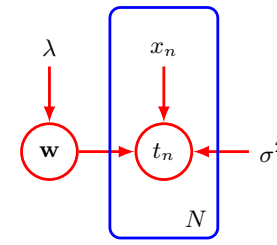


Figure 11.10 The same model as in Figure 11.9 but with the deterministic parameters shown explicitly by the floating variables.

Figure 11.9 and include the deterministic parameters, we obtain the graph shown in Figure 11.10.

When we apply a graphical model to a problem in machine learning, we will typically set some of the random variables to specific observed values. For example, the stochastic variables $\{t_n\}$ in the linear regression model will be set equal to the specific values given in the training set. In a graphical model, we denote such *observed variables* by shading the corresponding nodes. Thus, the graph corresponding to Figure 11.10 in which the variables $\{t_n\}$ are observed is shown in Figure 11.11.

Note that the value of $\mathbf{w}$ is not observed, and so $\mathbf{w}$ is an example of a *latent* variable, also known as a *hidden* variable. Such variables play a crucial role in many of the models discussed in this book. We therefore have three kinds of variables in a directed graphical model. First, there are unobserved (also called latent, or hidden) stochastic variables, which are denoted by open red circles. Second, when stochastic variables are observed, so that that they are set to specific values, they are denoted by red circles shaded with blue. Finally, non-stochastic parameters are denoted by floating variables, as seen in Figure 11.11.

Note that model parameters such as $\mathbf{w}$ are generally of little direct interest in themselves, because our ultimate goal is to make predictions for new input values. Suppose we are given a new input value $\widehat{x}$ and we wish to find the corresponding probability distribution for $\widehat{t}$ conditioned on the observed data. The joint distribution of all the random variables in this model, conditioned on the deterministic parameters, is given by

$$p(\widehat{t}, \mathbf{t}, \mathbf{w}|\widehat{x}, \mathbf{X}, \lambda) = p(\mathbf{w}|\lambda)p(\widehat{t}|\mathbf{w}, \widehat{x}) \prod_{n=1}^{N} p(t_n|\mathbf{w}, \mathbf{x}_n) \tag{11.19}$$

and the corresponding graphical model is shown in Figure 11.12.

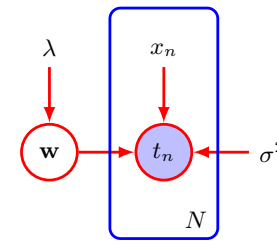


Figure 11.11 As in Figure 11.10 but with the nodes $\{t_n\}$ shaded to indicate that the corresponding random variables have been set to their observed values given by the training set.

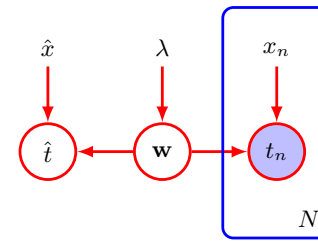


Figure 11.12 The classification model, corresponding to Figure 11.11, showing a new input value $\widehat{x}$ together with the corresponding model prediction $\widehat{t}$.

The required predictive distribution for $\widehat{t}$ is then obtained from the sum rule of probability by integrating out the model parameters $\mathbf{w}$. This integration over parameters represents a fully Bayesian treatment, which is rarely used in practice, especially with deep neural networks. Instead, we approximate this integral by first finding the most probable value $\mathbf{w}_{\text{MAP}}$ that maximizes the posterior distribution and then using just this single value to make predictions using $p(\widehat{t}|\mathbf{w}_{\text{MAP}}, \widehat{x})$.

11.1.7 Bayes' theorem

When stochastic variables in a probabilistic model are set equal to observed values, the distributions over other unobserved stochastic variables change accordingly. The process of calculating these updated distributions is known as *inference*. We can illustrate this by considering the graphical interpretation of Bayes' theorem. Suppose we decompose the joint distribution $p(x, y)$ over two variables x and y into a product of factors in the form $p(x, y) = p(x)p(y|x)$. This can be represented by the directed graph shown in Figure 11.13(a). Now suppose we observe the value of y, as indicated by the shaded node in Figure 11.13(b). We can view the marginal distribution $p(x)$ as a prior over the latent variable x, and our goal is to infer the corresponding posterior. Using the sum and product rules of probability we can evaluate

$$p(y) = \sum_{x'} p(y|x')p(x'), \tag{11.20}$$

which can then be used in Bayes' theorem to calculate

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}. \tag{11.21}$$

Thus, the joint distribution is now expressed in terms of $p(x|y)$ and $p(y)$. From a graphical perspective, the joint distribution $p(x, y)$ is represented by the graph shown in Figure 11.13(c), in which the direction of the arrow is reversed. This is the simplest example of an inference problem for a graphical model.

For complex graphical models that capture rich probabilistic structure, the process of calculating posterior distributions once some of the stochastic variables are observed can be complex and subtle. Conceptually, it simply involves the systematic application of the sum and product rules of probability, or equivalently Bayes' theorem. In practice, however, managing these calculations efficiently can benefit greatly from an exploitation of the graphical structure. These calculations can be expressed

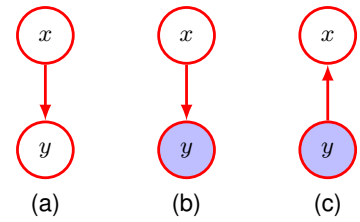


Figure 11.13 A graphical representation of Bayes' theorem showing (a) a joint distribution over two variables x and y expressed in factorized form, (b) the case with y set to an observed value, and (c) the resulting posterior distribution over x, given by Bayes' theorem.

in terms of elegant calculations on the graph that involve sending local messages between nodes. Such methods give exact answers for tree-structured graphs and give approximate iterative algorithms for graphs with loops. Since we will not discuss these further here, see Bishop (2006) for a more comprehensive discussion in the context of machine learning.

11.2. Conditional Independence

An important concept for probability distributions over multiple variables is that of *conditional independence* (Dawid, 1980). Consider three variables a, b, and c, and suppose that the conditional distribution of a given b and c is such that it does not depend on the value of b, so that

$$p(a|b,c) = p(a|c). \tag{11.22}$$

We say that a is conditionally independent of b given c. This can be expressed in a slightly different way if we consider the joint distribution of a and b conditioned on c, which we can write in the form

$$\begin{aligned} p(a,b|c) &= p(a|b,c)p(b|c) \\ &= p(a|c)p(b|c) \end{aligned} \tag{11.23}$$

where we have used the product rule of probability together with (11.22). We see that, conditioned on c, the joint distribution of a and b factorizes into the product of the marginal distribution of a and the marginal distribution of b (again both conditioned on c). This says that the variables a and b are statistically independent, given c. Note that our definition of conditional independence will require that (11.22), or equivalently (11.23), must hold for every possible value of c, and not just for some values. We will sometimes use a shorthand notation for conditional independence (Dawid, 1979) in which

$$a \perp\!\!\!\perp b \mid c \tag{11.24}$$

denotes that a is conditionally independent of b given c. Conditional independence properties play an important role in probabilistic models for machine learning because they simplify both the structure of a model and the computations needed to perform inference and learning under that model.

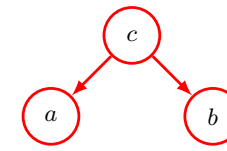


Figure 11.14 The first of three examples of graphs over three variables a, b, and c used to discuss conditional independence properties of directed graphical models.

If we are given an expression for the joint distribution over a set of variables in terms of a product of conditional distributions (i.e., the mathematical representation underlying a directed graph), then we could in principle test whether any potential conditional independence property holds by repeated application of the sum and product rules of probability. In practice, such an approach would be very time-consuming. An important and elegant feature of graphical models is that conditional independence properties of the joint distribution can be read directly from the graph without having to perform any analytical manipulations. The general framework for achieving this is called *d-separation*, where the 'd' stands for 'directed' (Pearl, 1988). Here we will motivate the concept of d-separation and give a general statement of the d-separation criterion. A formal proof can be found in Lauritzen (1996).

11.2.1 Three example graphs

We begin our discussion of the conditional independence properties of directed graphs by considering three simple examples each involving graphs having just three nodes. Together, these will motivate and illustrate the key concepts of d-separation. The first of the three examples is shown in Figure 11.14, and the joint distribution corresponding to this graph is easily written down using the general result (11.6) to give

$$p(a, b, c) = p(a|c)p(b|c)p(c). \tag{11.25}$$

If none of the variables are observed, then we can investigate whether a and b are independent by marginalizing both sides of (11.25) with respect to c to give

$$p(a, b) = \sum_c p(a|c)p(b|c)p(c). \tag{11.26}$$

In general, this does not factorize into the product $p(a)p(b)$, and so

$$a \not\perp\!\!\!\perp b \mid \emptyset \tag{11.27}$$

where $\emptyset$ denotes the empty set, and the symbol $\not\perp\!\!\!\perp$ means that the conditional independence property does not hold in general. Of course, it may hold for a particular distribution by virtue of the specific numerical values associated with the various conditional probabilities, but it does not follow in general from the structure of the graph.

Now suppose we condition on the variable c, as represented by the graph of Figure 11.15. From (11.25), we can easily write down the conditional distribution of

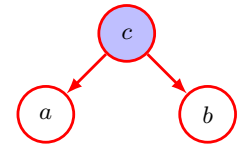


Figure 11.15 As in Figure 11.14 but where we have conditioned on the value of variable c.

a and b, given c, in the form

$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned}$$

and so we obtain the conditional independence property

$$a \perp\!\!\!\perp b \mid c.$$

We can provide a simple graphical interpretation of this result by considering the path from node a to node b via c. The node c is said to be *tail-to-tail* with respect to this path because the node is connected to the tails of the two arrows, and the presence of such a path connecting nodes a and b causes these nodes to be dependent. However, when we condition on node c, as in Figure 11.15, the conditioned node 'blocks' the path from a to b and causes a and b to become (conditionally) independent.

We can similarly consider the graph shown in Figure 11.16. The joint distribution corresponding to this graph is again obtained from our general formula (11.6) to give

$$p(a,b,c) = p(a)p(c|a)p(b|c). \tag{11.28}$$

First, suppose that none of the variables are observed. Again, we can test to see if a and b are independent by marginalizing over c to give

$$p(a,b) = p(a)\sum_c p(c|a)p(b|c) = p(a)p(b|a)$$

which in general does not factorize into $p(a)p(b)$, and so

$$a \not\perp\!\!\!\perp b \mid \emptyset \tag{11.29}$$

as before.

Now suppose we condition on node c, as shown in Figure 11.17. Using Bayes'

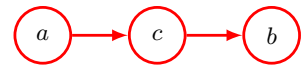
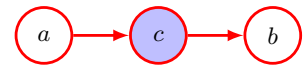


Figure 11.16 The second of our three examples of three-node graphs used to motivate the conditional independence framework for directed graphical models.

Figure 11.17 As in Figure 11.16 but now conditioning on node c.



theorem together with (11.28), we obtain

$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= \frac{p(a)p(c|a)p(b|c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned}$$

and so again we obtain the conditional independence property

$$a \perp\!\!\!\perp b \mid c.$$

As before, we can interpret these results graphically. The node c is said to be *head-to-tail* with respect to the path from node a to node b. Such a path connects nodes a and b and renders them dependent. If we now observe c, as in Figure 11.17, then this observation 'blocks' the path from a to b and so we obtain the conditional independence property $a \perp\!\!\!\perp b \mid c$.

Finally, we consider the third of our three-node examples, shown by the graph in Figure 11.18. As we will see, this has a more subtle behaviour than the two previous graphs. The joint distribution can again be written down using our general result (11.6) to give

$$p(a,b,c) = p(a)p(b)p(c|a,b). \tag{11.30}$$

Consider first the case where none of the variables are observed. Marginalizing both sides of (11.30) over c we obtain

$$p(a,b) = p(a)p(b)$$

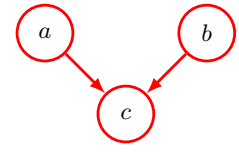
and so a and b are independent with no variables observed, in contrast to the two previous examples. We can write this result as

$$a \perp\!\!\!\perp b \mid \emptyset. \tag{11.31}$$

Now suppose we condition on c, as indicated in Figure 11.19. The conditional distribution of a and b is then given by

$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= \frac{p(a)p(b)p(c|a,b)}{p(c)}, \end{aligned}$$

Figure 11.18 The last of our three examples of three-node graphs used to explore conditional independence properties in graphical models. This graph has rather different properties from the two previous examples.



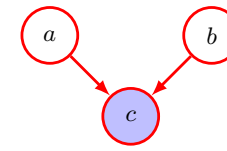


Figure 11.19 As in Figure 11.18 but conditioning on the value of node c. In this graph, the act of conditioning induces a dependence between a and b.

which in general does not factorize into the product $p(a|c)p(b|c)$, and so

$$a \not\perp\!\!\!\perp b \mid c.$$

Thus, our third example has the opposite behaviour from the first two. Graphically, we say that node c is *head-to-head* with respect to the path from a to b because it connects to the heads of the two arrows. The node c is sometimes called a *collider node*. When node c is unobserved, it 'blocks' the path, and the variables a and b are independent. However, conditioning on c 'unblocks' the path and renders a and b dependent.

There is one more subtlety associated with this third example that we need to consider. First we introduce some more terminology. We say that node y is a *descendant* of node x if there is a path from x to y in which each step of the path follows the directions of the arrows. Then it can be shown that a head-to-head path will become unblocked if either the node, *or any of its descendants*, is observed.

Exercise 11.13

In summary, a tail-to-tail node or a head-to-tail node leaves a path unblocked unless it is observed, in which case it blocks the path. By contrast, a head-to-head node blocks a path if it is unobserved, but once the node and/or at least one of its descendants is observed the path becomes unblocked.

11.2.2 Explaining away

It is worth spending a moment to understand further the unusual behaviour of the graph in Figure 11.19. Consider a particular instance of such a graph corresponding to a problem with three binary random variables relating to the fuel system on a car, as shown in Figure 11.20. The variables are B, which represents the state of a battery that is either charged ($B = 1$) or flat ($B = 0$), F which represents the state of the fuel tank that is either full of fuel ($F = 1$) or empty ($F = 0$), and G, which is the state of an electric fuel gauge and which indicates that the fuel tank is either full ($G = 1$) or empty ($G = 0$). The battery is either charged or flat, and independently,

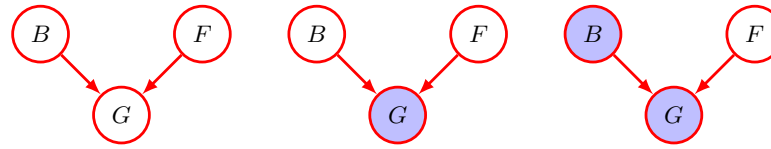


Figure 11.20 An example of a three-node graph used to illustrate 'explaining away'. The three nodes represent the state of the battery (B), the state of the fuel tank (F), and the reading on the electric fuel gauge (G). See the text for details.

the fuel tank is either full or empty, with prior probabilities

$$\begin{aligned} p(B=1) &= 0.9 \\ p(F=1) &= 0.9. \end{aligned}$$

Given the state of the fuel tank and the battery, the fuel gauge reads full with probabilities given by

$$\begin{aligned} p(G=1|B=1,F=1) &= 0.8 \\ p(G=1|B=1,F=0) &= 0.2 \\ p(G=1|B=0,F=1) &= 0.2 \\ p(G=1|B=0,F=0) &= 0.1 \end{aligned}$$

so this is a rather unreliable fuel gauge! All remaining probabilities are determined by the requirement that probabilities sum to one, and so we have a complete specification of the probabilistic model.

Before we observe any data, the prior probability of the fuel tank being empty is $p(F=0)=0.1$. Now suppose that we observe the fuel gauge and discover that it reads empty, i.e., $G=0$, corresponding to the middle graph in Figure 11.20. We can use Bayes' theorem to evaluate the posterior probability of the fuel tank being empty. First we evaluate the denominator for Bayes' theorem:

$$p(G=0) = \sum_{B\in\{0,1\}} \sum_{F\in\{0,1\}} p(G=0|B,F)p(B)p(F) = 0.315 \tag{11.32}$$

and similarly we evaluate

$$p(G=0|F=0) = \sum_{B\in\{0,1\}} p(G=0|B,F=0)p(B) = 0.81 \tag{11.33}$$

and using these results, we have

$$p(F=0|G=0) = \frac{p(G=0|F=0)p(F=0)}{p(G=0)} \simeq 0.257 \tag{11.34}$$

and so $p(F=0|G=0) > p(F=0)$. Thus, observing that the gauge reads empty makes it more likely that the tank is indeed empty, as we would intuitively expect. Next suppose that we also check the state of the battery and find that it is flat, i.e., $B=0$. We have now observed the states of both the fuel gauge and the battery, as shown by the right-hand graph in Figure 11.20. The posterior probability that the fuel tank is empty given the observations of both the fuel gauge and the battery state is then given by

$$p(F=0|G=0,B=0) = \frac{p(G=0|B=0,F=0)p(F=0)}{\sum_{F\in\{0,1\}} p(G=0|B=0,F)p(F)} \simeq 0.111 \tag{11.35}$$

where the prior probability $p(B=0)$ has cancelled between the numerator and denominator. Thus, the probability that the tank is empty has *decreased* (from 0.257

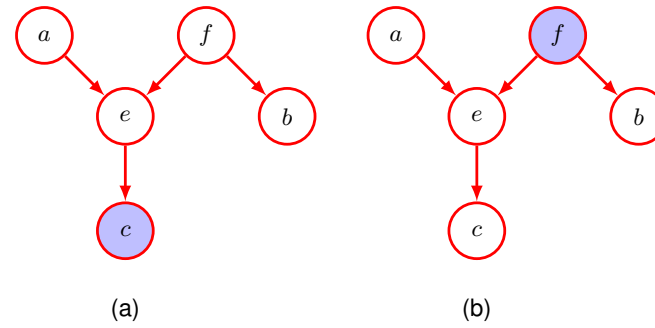


Figure 11.21 Illustration of d-separation. See the text for details.

to 0.111) as a result of the observation of the state of the battery. This accords with our intuition that finding that the battery is flat *explains away* the observation that the fuel gauge reads empty. We see that the state of the fuel tank and that of the battery have indeed become dependent on each other as a result of observing the reading on the fuel gauge. In fact, this would also be the case if, instead of observing the fuel gauge directly, we observed the state of some descendant of G, for example a rather unreliable witness who reports seeing that the gauge was reading empty. Note that the probability $p(F = 0|G = 0, B = 0) \simeq 0.111$ is greater than the prior probability $p(F = 0) = 0.1$ because the observation that the fuel gauge reads zero still provides some evidence in favour of an empty fuel tank.

Exercise 11.14

11.2.3 D-separation

We now give a general statement of the d-separation property (Pearl, 1988) for directed graphs. Consider a general directed graph in which A, B, and C are arbitrary non-intersecting sets of nodes (whose union may be smaller than the complete set of nodes in the graph). We wish to ascertain whether a particular conditional independence statement $A \perp\!\!\!\perp B \mid C$ is implied by a given directed acyclic graph. To do so, we consider all possible paths from any node in A to any node in B. Any such path is said to be *blocked* if it includes a node such that either

(a) the arrows on the path meet either head-to-tail or tail-to-tail at the node, and the node is in the set C, or

(b) the arrows meet head-to-head at the node and neither the node, nor any of its descendants is in the set C.

If all paths are blocked, then A is said to be d-separated from B by C, and the joint distribution over all the variables in the graph will satisfy $A \perp\!\!\!\perp B \mid C$.

D-separation is illustrated in Figure 11.21. In graph (a), the path from a to b is not blocked by node f because it is a tail-to-tail node for this path and is not observed, nor is it blocked by node e because, although the latter is a head-to-head node, it has a descendant c in the conditioning set. Thus, the conditional independence statement $a \perp\!\!\!\perp b \mid c$ does *not* follow from this graph. In graph (b), the path from a to b is blocked by node f because this is a tail-to-tail node that is observed, and so the conditional independence property $a \perp\!\!\!\perp b \mid f$ will be satisfied by any distribution that factorizes

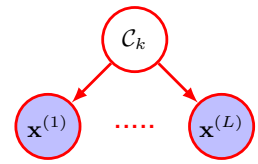


Figure 11.22 A graphical representation of the naive Bayes model for classification. Conditioned on the class label $\mathcal{C}_k$, the elements of the observed vector $\mathbf{x} = (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(L)})$ are assumed to be independent.

according to this graph. Note that this path is also blocked by node e because e is a head-to-head node and neither it nor its descendant are in the conditioning set. In d-separation, parameters such as λ in Figure 11.12, which are indicated by floating variables, behave in the same way as observed nodes. However, there are no marginal distributions associated with such nodes, and consequently parameter nodes never themselves have parents and so all paths through these nodes will always be tail-to-tail and hence blocked. Consequently they play no role in d-separation.

Section 2.3.2

Another example of conditional independence and d-separation is provided by i.i.d. (independent and identically distributed) data. Consider the binary classification model shown in Figure 11.12. Here the stochastic nodes correspond to $\{t_n\}$, $\mathbf{w}$, and $\widehat{t}$. We see that the node for $\mathbf{w}$ is tail-to-tail with respect to the path from $\widehat{t}$ to any one of the nodes t_n, and so we have the following conditional independence property:

$$\widehat{t} \perp\!\!\!\perp t_n \mid \mathbf{w}. \tag{11.36}$$

Thus, conditioned on the network parameters $\mathbf{w}$, the predictive distribution for $\widehat{t}$ is independent of the training data $\{t_1, \ldots, t_N\}$. We can therefore first use the training data to determine the posterior distribution (or some approximation to the posterior distribution) over the coefficients $\mathbf{w}$ and then we can discard the training data and use the posterior distribution for $\mathbf{w}$ to make predictions of $\widehat{t}$ for new input observations $\widehat{x}$.

11.2.4 Naive Bayes

A related graphical structure arises in an approach to classification called the *naive Bayes* model, in which we use conditional independence assumptions to simplify the model structure. Suppose our data consists of observations of a vector $\mathbf{x}$, and we wish to assign values of $\mathbf{x}$ to one of K classes. We can define a class-conditional density $p(\mathbf{x}|\mathcal{C}_k)$ for each of the classes, along with prior class probabilities $p(\mathcal{C}_k)$. The key assumption of the naive Bayes model is that, conditioned on the class $\mathcal{C}_k$, the distribution of the input variable factorizes into the product of two or more densities. Suppose we partition $\mathbf{x}$ into L elements $\mathbf{x} = (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(L)})$. Naive Bayes then takes the form

$$p(\mathbf{x}|\mathcal{C}_k) = \prod_{l=1}^{L} p(\mathbf{x}^{(l)}|\mathcal{C}_k) \tag{11.37}$$

where it is assumed that (11.37) holds for each of the classes $\mathcal{C}_k$ separately. The graphical representation of this model is shown in Figure 11.22. We see that an observation of $\mathcal{C}_k$ would block the path between $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ for $j \neq i$ because such

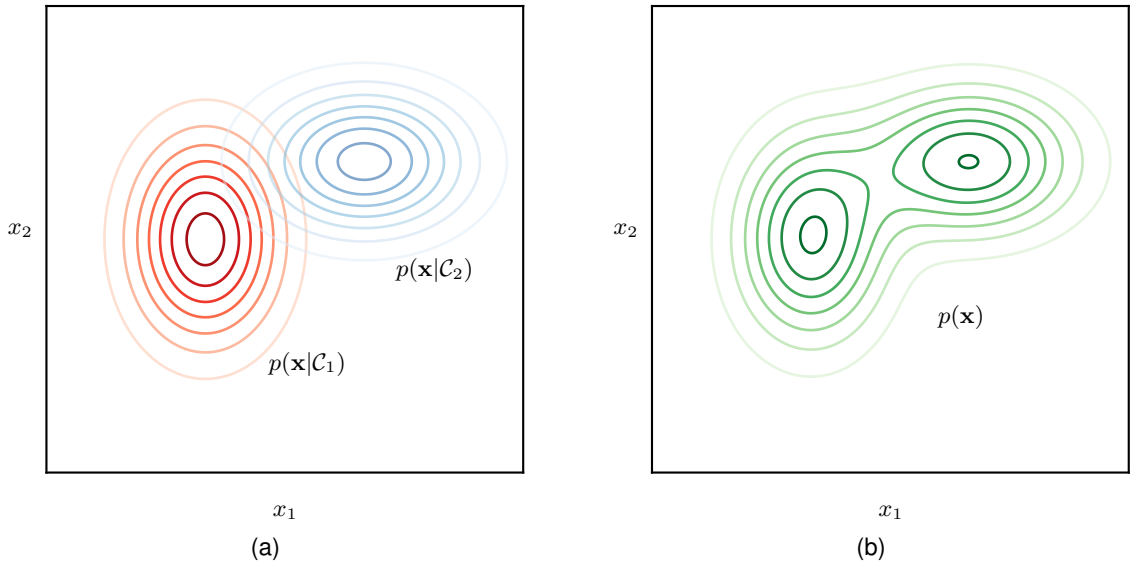


Figure 11.23 Illustration of a naive Bayes classifier for a two-dimensional data space, showing (a) the conditional distributions $p(\mathbf{x}|\mathcal{C}_k)$ for each of the two classes and (b) the marginal distribution $p(\mathbf{x})$ in which we have assumed equal class priors $p(\mathcal{C}_1) = p(\mathcal{C}_2) = 0.5$. Note that the conditional distributions factorize with respect to x_1 and x_2, whereas the marginal distribution does not.

paths are tail-to-tail at the node $\mathcal{C}_k$, and so $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are conditionally independent given $\mathcal{C}_k$. If, however, we marginalize out $\mathcal{C}_k$, the tail-to-tail path from $\mathbf{x}^{(i)}$ to $\mathbf{x}^{(j)}$ is no longer blocked, which tells us that in general the marginal density $p(\mathbf{x})$ will not factorize with respect to the elements $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(L)}$.

If we are given a labelled training set, comprising observations $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ together with their class labels, then we can fit the naive Bayes model to the training data using maximum likelihood by assuming that the data are drawn independently from the model. The solution is obtained by fitting the model for each class separately using the corresponding labelled data and then setting the class priors $p(\mathcal{C}_k)$ equal to the fraction of training data points in each class. The probability that a vector $\mathbf{x}$ belongs to class $\mathcal{C}_k$ is then given by Bayes' theorem in the form

Exercise 11.15

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})} \tag{11.38}$$

where $p(\mathbf{x}|\mathcal{C}_k)$ is given by (11.37), and $p(\mathbf{x})$ can be evaluated using

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k). \tag{11.39}$$

The naive Bayes model is illustrated for a two-dimensional data space in Figure 11.23 in which $\mathbf{x} = (x_1, x_2)$. Here we assume that the conditional densities

$p(\mathbf{x}|\mathcal{C}_k)$ for each of the two classes are axis-aligned Gaussians, and hence that they each factorize with respect to x_1 and x_2 so that

$$p(\mathbf{x}|\mathcal{C}_k) = p(x_1|\mathcal{C}_k)p(x_2|\mathcal{C}_k). \tag{11.40}$$

However, the marginal density $p(\mathbf{x})$ given by

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k) \tag{11.41}$$

is now a mixture of Gaussians and does not factorize with respect to x_1 and x_2. We have already encountered a simple application of the naive Bayes model in the context of fusing data from different sources, such as blood tests and skin images for medical diagnosis.

Section 5.2.4

The naive Bayes assumption is helpful when the dimensionality D of the input space is high, making density estimation in the full D-dimensional space more challenging. It is also useful if the input vector contains both discrete and continuous variables, since each can be represented separately using appropriate models (e.g., Bernoulli distributions for binary observations or Gaussians for real-valued variables). The conditional independence assumption of this model is clearly a strong one that may lead to rather poor representations of the class-conditional densities. Nevertheless, even if this assumption is not precisely satisfied, the model may still give good classification performance in practice because the decision boundaries can be insensitive to some of the details in the class-conditional densities, as illustrated in Figure 5.8.

11.2.5 Generative models

Many applications of machine learning can be viewed as examples of *inverse problems* in which there is an underlying, often physical, process that generates data, and the goal is to learn now to invert this process. For example, an image of an object can be viewed as the output of a generative process in which the type of object is selected from some distribution of possible object classes. The position and orientation of the object are also chosen from some prior distributions, and then the resulting image is created. Given a large data set of images labelled with the type, position, and scale of the objects they contain, the goal is to train a machine learning model that can take new, unlabelled images and detect the presence of an object including its location within the image and its size. The machine learning solution therefore represents the inverse of the process that generated the data.

One approach would be to train a deep neural network, such as a convolutional network, to take an image as input and to generate outputs that describe the object's type, position, and scale. This approach therefore tries to solve the inverse problem directly and is an example of a *discriminative model*. It can achieve high accuracy provided ample examples of labelled images are available. In practice, unlabelled images are often plentiful, and much of the effort in obtaining a training set goes into proving the labels, which may be done by hand. Our simple discriminative model cannot directly make use of unlabelled images during training.

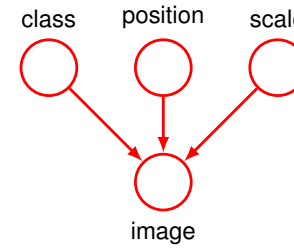


Figure 11.24 A graphical model representing the process by which images of objects are created. The identity of an object (a discrete variable) and the position and orientation of that object (continuous variables) have independent prior probabilities. The image (an array of pixel intensities) has a probability distribution that is dependent on the identity of the object as well as on its position and orientation.

An alternative approach is to model the generative process and then subsequently to invert it computationally. In our image example, if we assume that the object's class, position, and scale are all chosen independently, then we can represent the generative process using a directed graphical model as shown in Figure 11.24. Note that the directions of the arrows correspond to the sequence of generative steps, and so the model represents the *causal* process (Pearl, 1988) by which the observed data is generated. This is an example of a *generative model* because once it is trained, it can be used to generate synthetic images by first selecting values for object's class, position, and scale from the learned prior distributions and then subsequently sampling an image from the learned conditional distribution. We will later see how diffusion models and other generative models can synthesize impressive high-resolution images based on a textual description of the desired content and style of the image.

Chapter 20

The graph in Figure 11.24 assumes that, when no image is observed, the class, position, and scale variables are independent. This follows because every path between any two of these variables is head-to-head with respect to the image variable, which is unobserved. However, when we observe an image, those paths become unblocked, and the class, position, and scale variables are no longer independent. Intuitively this is reasonable because being told the identity of the object within the image provides us with very relevant information to assist us with determining its location.

The hidden variables in a probabilistic model need not, however, have any explicit physical interpretation but may be introduced simply to allow a more complex joint distribution to be constructed from simpler components. For example, models such as normalizing flows, variational autoencoders, and diffusion models all use deep neural networks to create complex distributions in the data space by transforming hidden variables having a simple Gaussian distribution.

11.2.6 Markov blanket

A conditional independence property that is helpful when discussing more complex directed graphs is called the *Markov blanket* or *Markov boundary*. Consider a joint distribution $p(\mathbf{x}_1, \ldots, \mathbf{x}_D)$ represented by a directed graph having D nodes, and consider the conditional distribution of a particular node with variables $\mathbf{x}_i$ conditioned on all the remaining variables $\mathbf{x}_{j \neq i}$. Using the factorization property (11.6),

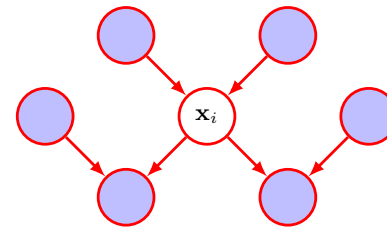


Figure 11.25 The Markov blanket of a node $\mathbf{x}_i$ comprises the set of parents, children, and co-parents of the node. It has the property that the conditional distribution of $\mathbf{x}_i$, conditioned on all the remaining variables in the graph, is dependent only on the variables in the Markov blanket.

we can express this conditional distribution in the form

$$p(\mathbf{x}_i|\mathbf{x}_{\{j\neq i\}}) = \frac{p(\mathbf{x}_1,\ldots,\mathbf{x}_D)}{\displaystyle\int p(\mathbf{x}_1,\ldots,\mathbf{x}_D)\,\mathrm{d}\mathbf{x}_i} = \frac{\displaystyle\prod_k p(\mathbf{x}_k|\mathrm{pa}(k))}{\displaystyle\int \prod_k p(\mathbf{x}_k|\mathrm{pa}(k))\,\mathrm{d}\mathbf{x}_i}$$

in which the integral is replaced by a summation for discrete variables. We now observe that any factor $p(\mathbf{x}_k|\mathrm{pa}(k))$ that does not have any functional dependence on $\mathbf{x}_i$ can be taken outside the integral over $\mathbf{x}_i$ and will therefore cancel between numerator and denominator. The only factors that remain will be the conditional distribution $p(\mathbf{x}_i|\mathrm{pa}(i))$ for node $\mathbf{x}_i$ itself, together with the conditional distributions for any nodes $\mathbf{x}_k$ such that node $\mathbf{x}_i$ is in the conditioning set of $p(\mathbf{x}_k|\mathrm{pa}(k))$, in other words for which $\mathbf{x}_i$ is a parent of $\mathbf{x}_k$. The conditional $p(\mathbf{x}_i|\mathrm{pa}(i))$ will depend on the parents of node $\mathbf{x}_i$, whereas the conditionals $p(\mathbf{x}_k|\mathrm{pa}(k))$ will depend on the children of $\mathbf{x}_i$ as well as on the *co-parents*, in other words variables corresponding to parents of node $\mathbf{x}_k$ other than node $\mathbf{x}_i$. The set of nodes comprising the parents, the children, and the co-parents is called the *Markov blanket* and is illustrated in Figure 11.25.

We can think of the Markov blanket of a node $\mathbf{x}_i$ as being the minimal set of nodes that isolates $\mathbf{x}_i$ from the rest of the graph. Note that it is not sufficient to include only the parents and children of node $\mathbf{x}_i$ because explaining away means that observations of the child nodes will not block paths to the co-parents. We must therefore observe the co-parent nodes as well.

11.2.7 Graphs as filters

We have seen that a particular directed graph represents a specific decomposition of a joint probability distribution into a product of conditional probabilities, and it also expresses a set of conditional independence statements obtained through the d-separation criterion. The d-separation theorem is really an expression of the equivalence of these two properties. To make this clear, it is helpful to think of a directed graph as a filter. Suppose we consider a particular joint probability distribution $p(\mathbf{x})$ over the variables $\mathbf{x}$ corresponding to the (unobserved) nodes of the graph. The filter will allow this distribution to pass through if, and only if, it can be expressed in

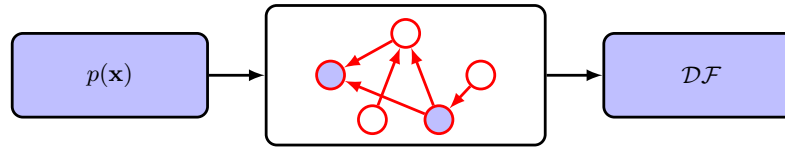


Figure 11.26 We can view a graphical model (in this case a directed graph) as a filter in which a probability distribution $p(\mathbf{x})$ is allowed through the filter if, and only if, it satisfies the directed factorization property (11.6). The set of all possible probability distributions $p(\mathbf{x})$ that pass through the filter is denoted $\mathcal{DF}$. We can alternatively use the graph to filter distributions according to whether they respect all the conditional independence properties implied by the d-separation properties of the graph. The d-separation theorem says the same set of distributions $\mathcal{DF}$ will be allowed through this second kind of filter.

terms of the factorization (11.6) implied by the graph. If we present to the filter the set of all possible distributions $p(\mathbf{x})$ over the set of variables $\mathbf{x}$, then the subset of distributions that are passed by the filter is denoted $\mathcal{DF}$, for *directed factorization*. This is illustrated in Figure 11.26.

Alternatively, we can use the graph as a different kind of filter by first listing all the conditional independence properties obtained by applying the d-separation criterion to the graph and then allowing a distribution to pass only if it satisfies all of these properties. If we present all possible distributions $p(\mathbf{x})$ to this second kind of filter, then the d-separation theorem tells us that the set of distributions that will be allowed through is precisely the set $\mathcal{DF}$.

It should be emphasized that the conditional independence properties obtained from d-separation apply to any probabilistic model described by that particular directed graph. This will be true, for instance, whether the variables are discrete or continuous or a combination of these. Again, we see that a particular graph describes a whole family of probability distributions.

At one extreme, we have a fully connected graph that exhibits no conditional independence properties at all and which can represent any possible joint probability distribution over the given variables. The set $\mathcal{DF}$ will contain all possible distributions $p(\mathbf{x})$. At the other extreme, we have a fully disconnected graph, i.e., one having no links at all. This corresponds to joint distributions that factorize into the product of the marginal distributions over the variables comprising the nodes of the graph. Note that for any given graph, the set of distributions $\mathcal{DF}$ will include any distributions that have additional independence properties beyond those described by the graph. For instance, a fully factorized distribution will always be passed through the filter implied by any graph over the corresponding set of variables.

11.3. Sequence Models

There are many important applications of machine learning in which the data consists of a *sequence* of values. For example, text comprises a sequence of words, whereas a protein comprises a sequence of amino acids. Many sequences are ordered by

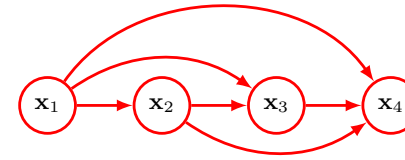


Figure 11.27 An illustration of a general autoregressive model of the form (11.42) with four nodes.

time, such as the audio signals from a microphone or daily rainfall measurements at a particular location. Sometimes the terminology of 'time' as well as 'past' and 'future' are used when referring to other types of sequential data, not just temporal sequences. Applications involving sequences include speech recognition, automatic translation between languages, detecting genes in DNA, synthesizing music, writing computer code, holding a conversation with a modern search engine, and many others.

We will denote a data sequence by $\mathbf{x}_1, \ldots, \mathbf{x}_N$ where each element $\mathbf{x}_n$ of the sequence comprises a vector of values. Note that we might have several such sequences drawn independently from the same distribution, in which case the joint distribution over all the sequences factorizes into the product of the distributions over each sequence individually. From now on, we focus on modelling just one of those sequences.

We have already seen in (11.4) that by repeated application of the product rule of probability, a general distribution over N variables can be written as the product of conditional distributions, and that the form of this decomposition depends on a specific of ordering for the variables. For vector-valued variables, and if we chose an ordering that corresponds to the order of the variables in the sequence, then we can write

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \prod_{n=1}^{N} p(\mathbf{x}_n | \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}). \tag{11.42}$$

This corresponds to a directed graph in which each node receives a link from every previous node in the sequence, as illustrated using four variables in Figure 11.27. This is known as an *autoregressive* model.

This representation has complete generality and therefore from a modelling perspective adds no value since it encodes no assumptions. We can constrain the space of models by introducing conditional independence properties by removing links from the graph, or equivalently by removing variables from the conditioning set of the factors on the right-hand-side of (11.42).

The strongest assumption would be to remove all conditioning variables, giving a joint distribution of the form

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \prod_{n=1}^{N} p(\mathbf{x}_n), \tag{11.43}$$

which treats the variables as independent and therefore completely ignores the ordering information. This corresponds to a probabilistic graphical model without links, as shown in Figure 11.28.

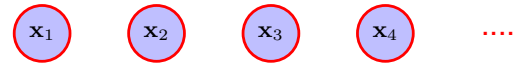


Figure 11.28 The simplest approach to modelling a sequence of observations is to treat them as independent, corresponding to a probabilistic graphical model without links.

Interesting models that capture sequential properties while introducing modelling assumptions lie between these two extremes. One strong assumption would be to assume that each conditional distribution depends only on the immediately preceding variable in the sequence, giving a joint distribution of the form

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_1) \prod_{n=2}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1}). \tag{11.44}$$

Note that the first variable in the sequence is treated slightly differently since it has no conditioning variable. The functional form (11.44) is known as a *Markov model*, or *Markov chain*, and is represented by a graph consisting of a simple chain of nodes, as seen in Figure 11.29. Using d-separation, we see that the conditional distribution for observation $\mathbf{x}_n$, given all of the observations up to time n, is given by

Section 11.2.3

$$p(\mathbf{x}_n|\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}), \tag{11.45}$$

which is easily verified by direct evaluation starting from (11.44) and using the product rule of probability. Thus, if we use such a model to predict the next observation in a sequence, the distribution of predictions will depend only on the value of the immediately preceding observation and will be independent of all earlier observations.

Exercise 11.16

More specifically, (11.44) is known as a first-order Markov model because only one conditioning variable appears in each conditional distribution. We can extend the model by allowing each conditional distribution to depend on the two preceding variables, giving a second-order Markov model of the form

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_1)p(\mathbf{x}_2|\mathbf{x}_1) \prod_{n=3}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{x}_{n-2}). \tag{11.46}$$

Note that the first two variables are treated differently as they have fewer than two conditioning variables. This model is shown as a directed graph in Figure 11.30.

Exercise 11.17

By using d-separation (or by direct evaluation using the rules of probability), we see that in the second-order Markov model, the conditional distribution of $\mathbf{x}_n$ given all previous observations $\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}$ is independent of the observations $\mathbf{x}_1, \ldots \mathbf{x}_{n-3}$. We can similarly consider extensions to an M^{th} order Markov chain in

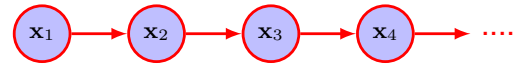


Figure 11.29 A first-order Markov chain of observations in which the distribution of a particular observation $\mathbf{x}_n$ is conditioned on the value of the previous observation $\mathbf{x}_{n-1}$.

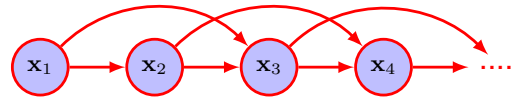


Figure 11.30 A second-order Markov chain in which the conditional distribution of a particular observation $\mathbf{x}_n$ depends on the values of the two previous observations $\mathbf{x}_{n-1}$ and $\mathbf{x}_{n-2}$.

which the conditional distribution for a particular variable depends on the previous M variables. However, we have paid a price for this increased flexibility because the number of parameters in the model is now much larger. Suppose the observations are discrete variables having K states. Then the conditional distribution $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ in a first-order Markov chain will be specified by a set of $K-1$ parameters for each of the K states of $\mathbf{x}_{n-1}$ giving a total of $K(K-1)$ parameters. Now suppose we extend the model to an M^{th} order Markov chain, so that the joint distribution is built up from conditionals $p(\mathbf{x}_n|\mathbf{x}_{n-M},\ldots,\mathbf{x}_{n-1})$. If the variables are discrete and if the conditional distributions are represented by general conditional probability tables, then such a model will have $K^{M-1}(K-1)$ parameters. Thus, the number of parameters grows exponentially with M, which will generally render this approach impractical for larger values of M.

11.3.1 Hidden variables

Suppose we wish to build a model for sequences that is not limited by the Markov assumption to any order and yet can be specified using a limited number of free parameters. We can achieve this by introducing additional latent variables, thus permitting a rich class of models to be constructed out of simple components. For each observation $\mathbf{x}_n$, we introduce a corresponding latent variable $\mathbf{z}_n$ (which may be of different type or dimensionality to the observed variable). We now assume that it is the latent variables that form a Markov chain, giving rise to the graphical structure known as a *state-space model*, which is shown in Figure 11.31. It satisfies the key conditional independence property that $\mathbf{z}_{n-1}$ and $\mathbf{z}_{n+1}$ are independent given $\mathbf{z}_n$, so that

$$\mathbf{z}_{n+1} \perp\!\!\!\perp \mathbf{z}_{n-1} \mid \mathbf{z}_n. \tag{11.47}$$

The joint distribution for this model is given by

$$p(\mathbf{x}_1,\ldots,\mathbf{x}_N,\mathbf{z}_1,\ldots,\mathbf{z}_N) = p(\mathbf{z}_1)\left[\prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1})\right]\prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n). \tag{11.48}$$

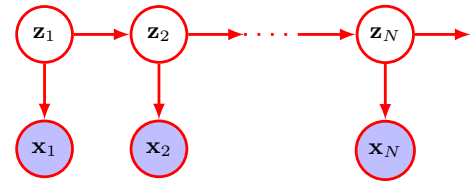


Figure 11.31 A state-space model expresses the joint probability distribution over a sequence of observed states $\mathbf{x}_1,\ldots,\mathbf{x}_N$ in terms of a Markov chain of hidden states $\mathbf{z}_1,\ldots,\mathbf{z}_N$ in the form (11.48).

Using the d-separation criterion, we see that in the state-space model there is always a path connecting any two observed variables $\mathbf{x}_n$ and $\mathbf{x}_m$ via the latent variables and that this path is never blocked. Thus, the predictive distribution $p(\mathbf{x}_{n+1}|\mathbf{x}_1, \ldots, \mathbf{x}_n)$ for observation $\mathbf{x}_{n+1}$ given all previous observations does not exhibit any conditional independence properties, and so our predictions for $\mathbf{x}_{n+1}$ depend on all previous observations. The observed variables, therefore, do not satisfy the Markov property at any order.

There are two important models for sequential data that are described by this graph. If the latent variables are discrete, then we obtain a *hidden Markov model* (Elliott, Aggoun, and Moore, 1995). Note that the observed variables in a hidden Markov model may be discrete or continuous, and a variety of different conditional distributions can be used to model them. If both the latent and the observed variables are Gaussian (with a linear-Gaussian dependence of the conditional distributions on their parents), then we obtain a *linear dynamical system*, also known as a *Kalman filter* (Zarchan and Musoff, 2005). Both hidden Markov models and Kalman filters are discussed at length, along with algorithms for training them, in Bishop (2006). Such models can be made considerably more flexible by replacing the simple discrete probability tables, or linear-Gaussian distributions, used to define $p(\mathbf{x}_n|\mathbf{z}_n)$ with deep neural networks.

Exercises

11.1 ($\star$) By marginalizing out the variables in order, show that the representation (11.6) for the joint distribution of a directed graph is correctly normalized, provided each of the conditional distributions is normalized.

11.2 ($\star$) Show that the property of there being no directed cycles in a directed graph follows from the statement that there exists an ordered numbering of the nodes such that for each node there are no links going to a lower-numbered node.

11.3 ($\star\star$) Consider three binary variables $a, b, c \in \{0, 1\}$ having the joint distribution given in Table 11.1. Show by direct evaluation that this distribution has the property that a and b are marginally dependent, so that $p(a, b) \neq p(a)p(b)$, but that they become independent when conditioned on c, so that $p(a, b|c) = p(a|c)p(b|c)$ for both $c = 0$ and $c = 1$.

Table 11.1 The joint distribution over three binary variables.

a	b	c	$p(a, b, c)$
0	0	0	0.192
0	0	1	0.144
0	1	0	0.048
0	1	1	0.216
1	0	0	0.192
1	0	1	0.064
1	1	0	0.048
1	1	1	0.096