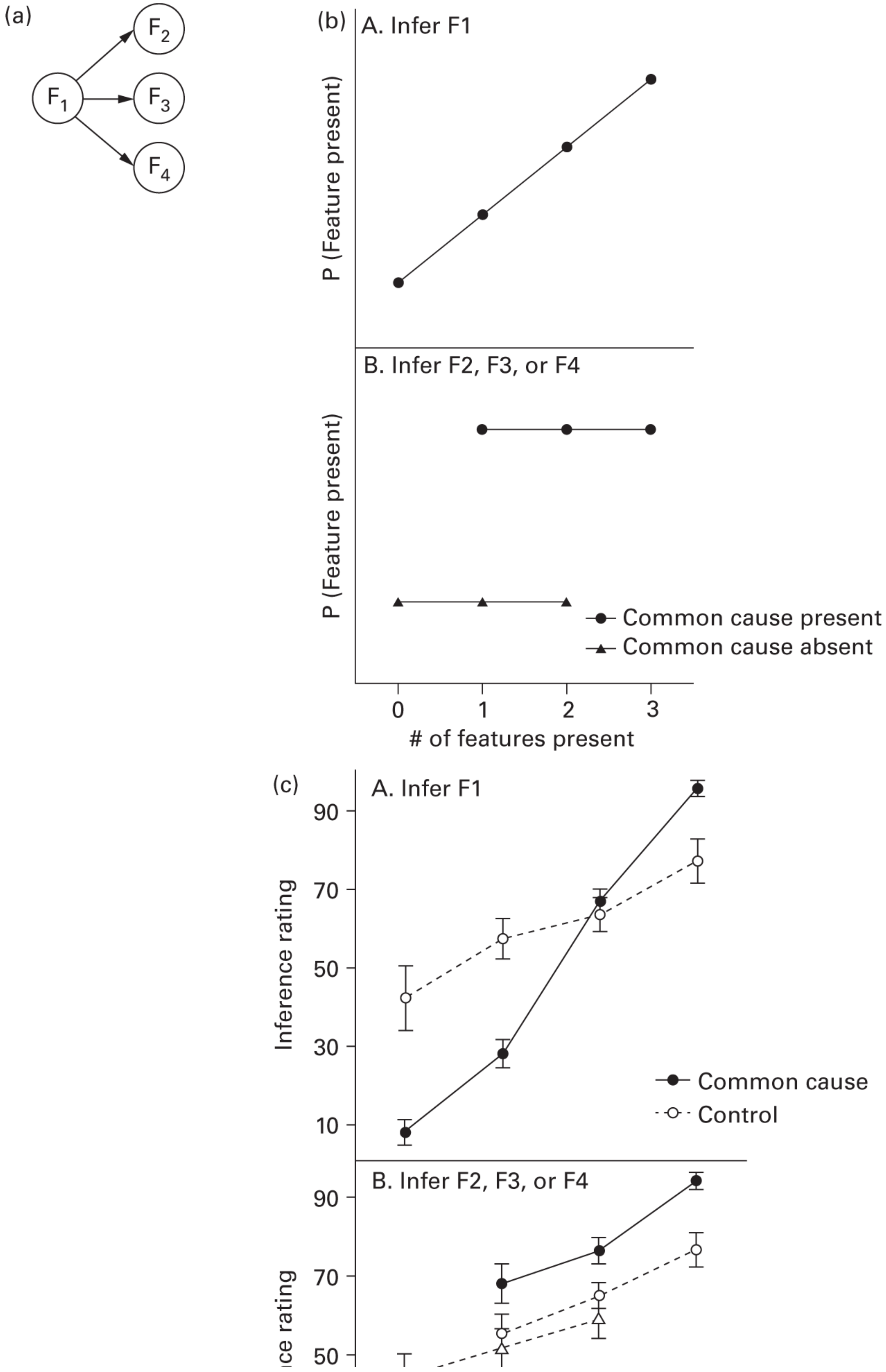
(a)
F1
F2
F3
F4
(b)
A. Infer F1
P (Feature present)
B. Infer F2, F3, or F4
P (Feature present)
Common cause present
Common cause absent
0
1
2
3
# of features present
(c)
A. Infer F1
90
70
50
30
10
Inference rating
Common cause
Control
B. Infer F2, F3, or F4
90
70
50
ce rating

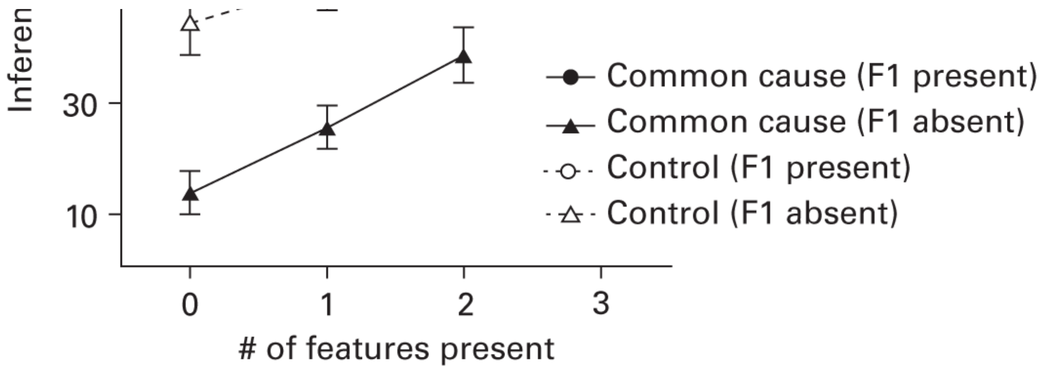

Figure 4.3

Feature inference (after Rehder & Burnett, 2005). (a) A common cause relationship between a set of features. (b) If the common cause (F_1) is unobserved, each additional feature that is observed provides more evidence about its value. However, once the common cause is observed, the other features become independent and do not provide evidence for one another's value. (c) People rated the strength of the inference that each feature was present based on the presence or absence of other features. People's judgments do not seem to obey the Markov condition in this case: they continue to treat features as mutually informative even when the common cause is observed. Figure adapted from Rehder and Burnett (2005).

(F_1 F_2). Blood high in iron sulfate causes a hyperactive immune system. The iron sulfate molecules are detected as foreign by the immune system, and the immune system is highly active as a result.

(F_1 F_3). Blood high in iron sulfate causes thick blood. Iron sulfate provides the extra iron that the ant uses to produce extra red blood cells. The extra red blood cells thicken the blood.

(F_1 F_4). Blood high in iron sulfate causes faster nest building. The iron sulfate stimulates the enzymes responsible for manufacturing the nest-building secretions, and an ant can build its nest faster with more secretions.

Participants were then tested to make sure that they understood the right structure of these relationships. The other group of participants were a control group that didn't learn about the causal relationships.

After being educated about this new category of objects, participants were asked to assess the probability of one feature given the observation of other features. Critically, some features were effects of the common cause (F_2, F_3, and F_4 in our example), while others were the cause itself. The Markov condition makes very different predictions about how these features should affect people's inferences.

As shown in figure 4.3b, if the unobserved feature is the common cause then observing each effect provides more evidence that the feature is present. By contrast, if the unobserved feature is one of the effects, then once the common cause is observed observing other effects provides no information. This is a consequence of the Markov condition: effects are independent of one another once the common cause is observed.

Rehder and Burnett found that people violated the Markov condition in this case. As shown in figure 4.3c, people continued to treat effects as providing evidence about other effects even when the common cause was observed. This suggests that people are allowing dependencies between these variables that go beyond those represented in the graphical model.

What could account for this result? One way to understand it is that people *do* follow the Markov condition, but they assume a more complex graphical model than the one they learned in the experiment. Rehder and Burnett performed a series of follow-up experiments to explore this possibility, finding a pattern of results consistent with a domain-general bias toward postulating a hidden mechanism behind category membership—an additional variable that is unobserved. This extra variable makes the features of objects that belong to a category dependent on one another, producing violations of the Markov condition in the original graphical model. This idea provides a connection to the literature on developmental psychology on essentialism (e.g., Gelman, 2003), which shows that children believe there is an unobservable "essence" that makes something a member of a category and is not altered by modifying its external features. Rehder has subsequently explored violations of the Markov condition that appear in people’s judgments in a variety of other settings (Rehder, 2014; Rehder & Waldmann, 2017; Rehder, 2018).

4.1.5 Defining Generative Models

Beyond being useful for formalizing probabilistic reasoning, Bayesian networks provide an intuitive representation for the structure of many probabilistic models. By breaking the process of producing data into a sequence of simple steps—a generative model—it becomes much easier to define the corresponding probability distribution.

Examples of Simple Generative Models We previously discussed the problem of estimating the weight of a coin, , using a Beta(,) prior. One detail that we left implicit in that discussion was the assumption that successive coin flips are independent, given a value for . This conditional independence assumption is expressed in the graphical model shown in figure 4.4a, where x_1, x_2, …, x_n are the outcomes (heads or tails) of n successive tosses. Applying the Markov condition, this structure represents the probability distribution

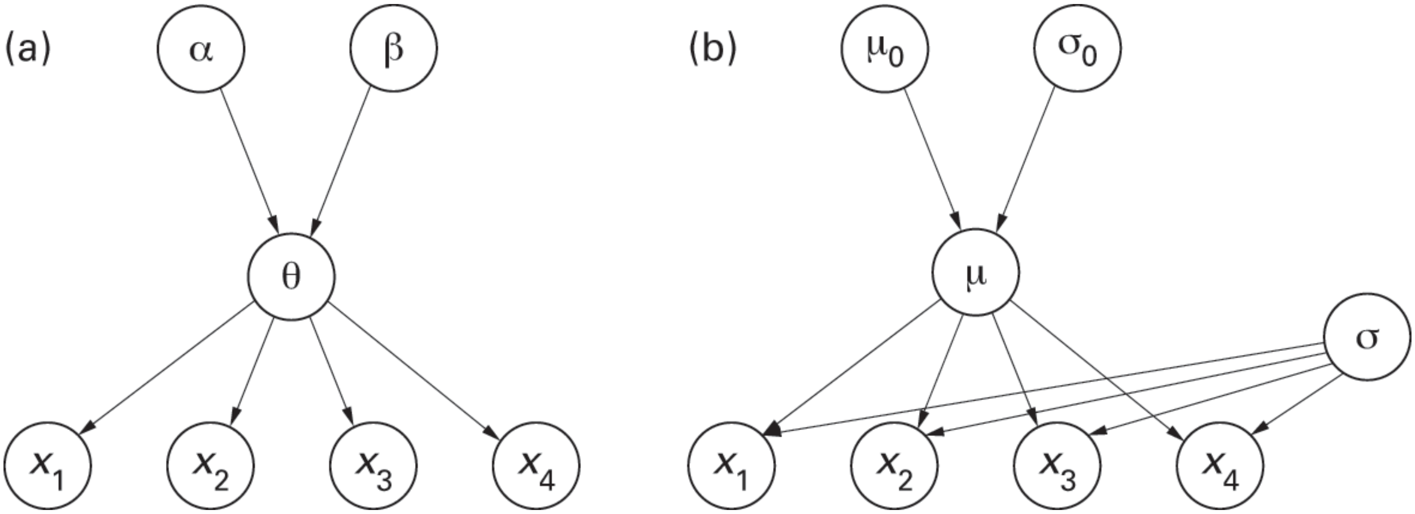

Figure 4.4

Graphical models for two simple examples of Bayesian inference. (a) Inferring a proportion, where is the weight of a coin, and are the hyperparameters of the prior, and x_1, x_2, …, x_n are the outcomes of n flips. (b) Inferring the mean of a Gaussian, where is the mean, $_0$ and $_0$ define the prior on the mean, is the standard deviation, and x_1, x_2, …, x_n are n observations.

$$P(x_1, x_2, \ldots, x_n, \theta, \alpha, \beta) = p(\alpha)p(\beta)p(\theta|\alpha,\beta)\prod_{i=1}^{n} P(x_i|\theta), \tag{4.3}$$

in which x_i is independent given the value of . Often, we would assume that the hyperparameters and are known, giving the conditional distribution

$$P(x_1, x_2, \ldots, x_n, \theta|\alpha,\beta) = p(\theta|\alpha,\beta)\prod_{i=1}^{n} P(x_i|\theta), \tag{4.4}$$

which can be used to infer when we condition on the values of $x_1, x_2, \ldots, x_n$.

Figure 4.4b shows a Bayesian network for the problem of estimating the mean of a Gaussian, , when that Gaussian has standard deviation and follows a Gaussian prior with mean $_0$ and standard deviation $_0$. The joint distribution is given by

$$p(x_1, x_2, \ldots, x_n, \mu, \sigma, \mu_0, \sigma_0) = p(\mu_0)p(\sigma_0)p(\mu|\mu_0, \sigma_0)p(\sigma)\prod_{i=1}^{n} p(x_i|\mu, \sigma), \tag{4.5}$$

and the conditional distribution when , $_0$, and $_0$ are known is

$$p(x_1, x_2, \ldots, x_n, \mu|\sigma, \mu_0, \sigma_0) = p(\mu|\mu_0, \sigma_0)\prod_{i=1}^{n} p(x_i|\mu, \sigma), \tag{4.6}$$

where we condition on $x_1, x_2, \ldots, x_n$ to infer .

These examples also illustrate that the "graphical" part of a graphical model is not itself sufficient to specify the associated probability distribution. We can derive equations (4.3)–(4.6) from the form of the associated graphs, but the graphs do not tell us what the actual distributions are that are used to generate the random variables. This information needs to be provided separately, defining a conditional probability distribution for each variable given its parents. The complete specification of a Bayesian network includes the graph and these conditional distributions, and together they define the associated joint distribution. For example, in figure 4.4a, we would specify the conditional distributions

$$\theta \mid \alpha, \beta \sim \text{Beta}(\alpha, \beta) \tag{4.7}$$

$$x_i \mid \theta \sim \text{Bernoulli}(\theta), \tag{4.8}$$

while in figure 4.4b, we would have the conditional distributions

$$\mu \mid \mu_0, \sigma_0 \sim \text{Gaussian}(\mu_0, \sigma_0) \tag{4.9}$$

$$x_i \mid \mu, \sigma \sim \text{Gaussian}(\mu, \sigma), \tag{4.10}$$

where should be read as "is distributed as." This is what makes these graphical models correspond to the examples presented earlier in this book.

Generative Models as a Sequence of Steps When introducing the basic ideas behind Bayesian inference, we emphasized the fact that hypotheses correspond to different assumptions about the process that could have generated the observed data. Bayesian networks help to make this idea transparent. Every Bayesian network indicates a sequence of steps that one could follow to generate samples from the joint distribution over the random variables in the network. First, one samples the values of all variables with no parents in the graph. Then, one samples the variables with parents taking known values, one after the other. For example, the graphical model shown in figure 4.4a corresponds to generating $x_1, x_2, \ldots, x_n$ by choosing and , sampling conditioned on those values, and then sampling each x_i conditioned on . Likewise, the graphical model shown in figure 4.4b corresponds to generating x_1, x_2, …, x_n by choosing $_0$, $_0$, and , sampling conditioned on $_0$ and $_0$, and then sampling each x_i conditioned on and . The directed graph associated with a probability distribution provides an intuitive representation for the steps involved in such a generative model.

The generative models shown in figure 4.4 both assume that observations are independent of one another, conditioned on the parameters or and . Other dependency structures are possible. For example, the flips could be generated in a *Markov chain*, a sequence of random variables in which each variable is independent of all its predecessors given the variable that immediately precedes it (e.g., Norris, 1997). We could use a Markov chain to represent a hypothesis space of coins that are particularly biased toward alternating or maintaining their last outcomes, letting the parameter be the probability that the outcome x_i takes the same value as x_{i1} (and assuming that x_1 is heads with probability 0.5). This distribution would correspond to the graphical model shown in figure 4.5b. (Figure 4.5a reproduces the independent case for the sake of comparison, suppressing the dependence of on and .) Applying the Markov condition, this structure represents the probability distribution

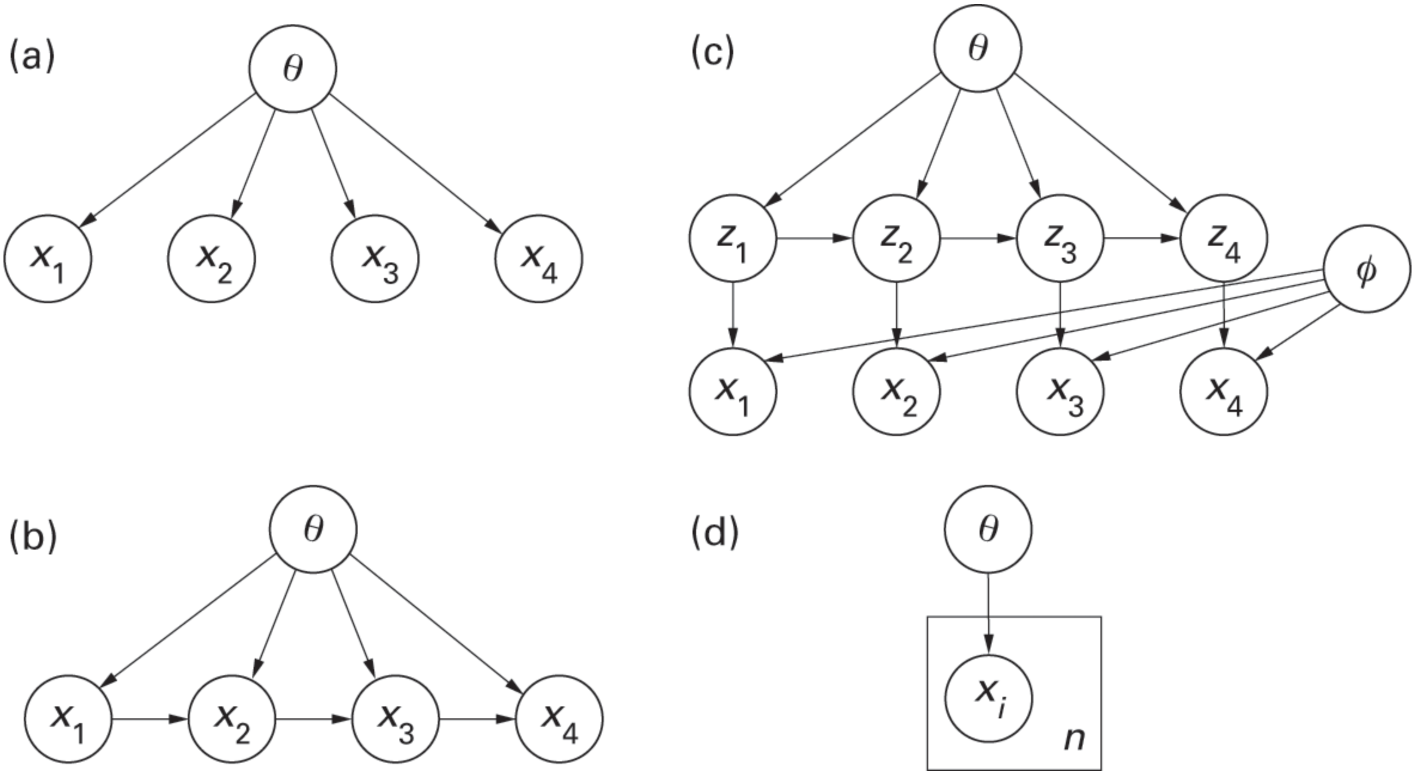

Figure 4.5
Graphical models showing different kinds of processes that could generate a sequence of coin flips. (a) Independent flips, with parameters determining the probability of heads. (b) A Markov chain, where the probability of heads depends on the result of the previous flip and the parameters defines the probability of heads after heads and after tails. (c) A hidden Markov model, in which the probability of heads depends on a latent state variable z_i. Transitions between values of the latent state are set by parameter , while an other parameter determines the probability of heads for each value of the latent state z_i. This kind of model is commonly used in computational linguistics, where the x_i might be the sequence of words in a document, and the z_i the syntactic classes from which they are generated. (d) A plate (the rectangle) can be used to indicate the replication of an element of a graphical model. Here, x_i is replicated n times. Taking n = 4 produces the graphical model shown in (a).

$$P(x_1, x_2, \ldots, x_n, \theta) = p(\theta) P(x_1) \prod_{i=2}^{n} P(x_i | x_{i-1}, \theta), \tag{4.11}$$

in which each x_i depends only on x_{i1}, given . More elaborate structures are also possible: any directed acyclic graph on x_1, x_2, …, x_n and corresponds to a valid set of assumptions about the dependencies among these variables.

Plate Notation and Latent Variables When dealing with graphical models involving large numbers of variables, it can be convenient to use a summary notation indicating repeated structure. The standard notation for doing this is using *plates*, which are rectangles enclosing a set of variables. A plate includes a number indicating how many times it should be replicated. All replicated variables have the same incoming and outgoing edges, and any edges between variables on the plate are replicated. Statistical independence is a common context in which structure is replicated, with the independent variables sharing common incoming edges from some parameters that they depend on, but not having edges between them. Figure 4.5d shows how the graphical model from figure 4.5a can be expressed more efficiently using plate notation.

For the generative models represented by figure 4.5a or 4.5b, we have assumed that all variables except are observed in each sample from the model or each data point. More generally, generative models can include a number of steps that refer to unobserved or *latent variables*. Introducing latent variables can lead to apparently complicated dependency structures among the observable variables. For example, in the graphical model shown in figure 4.5c, a sequence of latent variables z_1, z_2, …, z_n influences the probability that each respective coin flip in a sequence x_1, x_2, …, x_n comes up heads (in conjunction with a set of parameters). The latent variables form a Markov chain, with the value of z_i depending only on the value of z_{i1} (in conjunction with the parameters). This model, called a *hidden Markov model* (HMM), has historically been used in computational linguistics, where z_i might be the syntactic class (such as noun or verb) of a word, encodes the probability that a word of one class will appear after another (capturing simple syntactic constraints on the structure of sentences), and encodes the probability that each word will be generated from a particular syntactic class (e.g., Charniak, 1993; Jurafsky & Martin, 2000; Manning & Schütze, 1999). The dependencies among the latent variables induce dependencies among the observed variables—in the case of language, the constraints on transitions between syntactic classes impose constraints on which words can follow one another. We will return to HMMs in chapters.

4.2 Probabilistic Inference in Graphical Models

Recognizing the structure in a probability distribution can also greatly simplify the computations that we want to perform with that distribution. When variables are independent or conditionally independent of others, it reduces the number of terms that appear in sums over subsets of variables necessary to compute marginal beliefs about a variable or conditional beliefs about a variable, given the values of one or more other variables. A variety of algorithms have been developed to perform these probabilistic inferences efficiently on complex models by recognizing and exploiting conditional independence structures in Bayesian networks (Pearl, 1988; Mackay, 2003). These algorithms are used in AI systems, making it possible to reason efficiently under uncertainty (Korb & Nicholson, 2003; Russell & Norvig, 2021). In this section, we will illustrate the way in which knowing the dependencies between variables simplifies probabilistic computations and describe the particularly important kind of probabilistic inference, known as explaining away in more detail.

4.2.1 Simplifying Probabilistic Computations

While we will not go into detail about different inference algorithms here, we can provide an intuition for how knowing about patterns of statistical dependency can simplify inference by returning to the “psychic friend” example introduced earlier in the chapter. Let's say that we observed that the coin toss produced heads ($x_1 = 1$) and we wanted to infer the values of the other variables (x_2, x_3, and x_4). We then want to compute $P(x_2, x_3, x_4|x_1 = 1)$, which we obtain via Bayes' rule, with

$$P(x_2, x_3, x_4|x_1 = 1) = \frac{P(x_1 = 1, x_2, x_3, x_4)}{P(x_1 = 1)}. \tag{4.12}$$

Computing the denominator requires a sum of $P(x_1 = 1, x_2, x_3, x_4)$ over all values of x_2, x_3, and x_4, of which there are eight possible combinations. However, we can exploit the independence between variables to simplify this sum as

$$\begin{aligned}
P(x_1 = 1) &= \sum_{x_2,x_3,x_4} P(x_1 = 1, x_2, x_3, x_4) && (4.13)\\
&= \sum_{x_2,x_3,x_4} P(x_1 = 1|x_2, x_3, x_4)P(x_2|x_3, x_4)P(x_3|x_4)P(x_4) && (4.14)\\
&= \sum_{x_2,x_3,x_4} P(x_1 = 1|x_3, x_4)P(x_2|x_3)P(x_3)P(x_4) && (4.15)\\
&= \sum_{x_3,x_4} P(x_1 = 1|x_3, x_4)P(x_3)P(x_4) \sum_{x_2} P(x_2|x_3) && (4.16)\\
&= \sum_{x_3,x_4} P(x_1 = 1|x_3, x_4)P(x_3)P(x_4), && (4.17)
\end{aligned}$$

where equation (4.14) applies the chain rule to factorize the probability distribution, equation (4.15) uses the Markov condition for this graphical model, and equation (4.17) exploits the fact that the sum of x_2 over $P(x_2|x_3)$ is just 1. Now our sum involves only four combinations rather than eight. Consequently, we are able to reduce the amount of computation we need to do by a factor of 2. This reduction in computation is purely a consequence of independence—if we know the value of X_3, then X_2 is independent of X_1. With many variables, knowing when different sets of variables are independent of one another can significantly speed up probabilistic computations.

4.2.2 Explaining Away

Part of the original motivation behind the development of Bayesian networks is that they naturally handle cases where there are multiple competing explanations for observed data. These cases posed a challenge for other systems for automated reasoning that were used in AI research before Bayesian networks, such as expert systems based on production rules (Pearl, 1988). The characteristic pattern of inference that allowed Bayesian networks to deal with these cases appropriately is known as "explaining away," which we briefly introduced earlier in the chapter.

We will illustrate explaining away with the "psychic friend" example. If you observe that the coin has come up heads, you know there are three possible explanations: your friend has psychic powers, a two-headed coin, or the coin came up heads by chance. If you had to assign a probability to whether your friend is psychic, the fact that the coin had come up heads would factor into that. However, if you inspect your friend’s coin and discover that it is two-headed, you immediately think that it is less likely that your friend is psychic. The two-headed coin “explains away” the evidence that informed this inference.

If we work through the math, we come to the same conclusions. Assume that the probability that the coin toss produces heads (i.e., $x_1 = 1$) is 1 if either your friend has psychic powers or a two-headed coin, and otherwise is 0.5. Before observing the coin toss, the probability of your friend having psychic powers is $P(x_3 = 1) =$, and the probability of a two-headed coin is $P(x_4 = 1) =$. After observing the coin being tossed and coming up heads, the probability of both of these variables being true is

$$P(x_3 = 1|x_1 = 1) = \frac{P(x_1 = 1|x_3 = 1)P(x_3 = 1)}{P(x_1 = 1)} \tag{4.18}$$

$$= \frac{P(x_1 = 1|x_3 = 1)P(x_3 = 1)}{\sum_{x_3,x_4} P(x_1 = 1|x_3, x_4)P(x_3)P(x_4)} \tag{4.19}$$

$$= \frac{\psi}{0.5(1-\psi)(1-\gamma) + \psi(1-\gamma) + (1-\psi)\gamma + \psi\gamma} \tag{4.20}$$

$$= \frac{\psi}{1 - 0.5(1-\psi)(1-\gamma)}, \tag{4.21}$$

where we use the result from equation (4.17) for $P(x_1 = 1)$ and then simplify. Alternatively, we can calculate $P(x_1 = 1)$ by observing that $x_1 = 0$ only when neither x_3 nor x_4 is true, and occurs only half the time even in this case. The analogous result for $P(x_4 = 1|x_1 = 1)$ can be obtained by substituting for and vice versa. Since the denominator is less than 1, the evidence in favor of both x_3 and x_4 being true is increased by observing $x_1 = 1$.

Now, consider what would happen when you inspected your friend’s coin and discovered that it actually did have two heads ($x_4 = 1$). The resulting distribution for x_3 is given by

$$P(x_3 = 1|x_1 = 1, x_4 = 1) = \frac{P(x_1 = 1|x_3 = 1, x_4 = 1)P(x_4 = 1)P(x_3 = 1)}{P(x_1 = 1, x_4 = 1)} \tag{4.22}$$

$$= \frac{\gamma\psi}{\gamma}, \tag{4.23}$$

where we obtain the denominator by observing that $x_1 = 1$ whenever $x_4 = 1$. Simplifying, $P(x_3 = 1|x_1 = 1, x_4 = 1)$ is just , exactly the same probability as $P(x_3)$. The effect that observing the coin come up heads had on your beliefs about your friend’s psychic powers has completely disappeared, with the outcome of the coin toss being fully explained away by the two-headed coin.

Explaining away may seem obvious when presented in this way, but it has several deep implications. First, it is a characteristic pattern of dependency that is observed only between variables that form this common effect structure. As a consequence, seeing this pattern of dependency can be a strong clue about the relationships between variables. Second, it instantiates a principle that is very relevant when thinking about psychological and neural mechanisms: causes of the same effect should inhibit one another.

4.2.3 An Example: Visual Inference

Common cause and common effect structures also arise in vision. We will illustrate this with examples from Kersten and Yuille (2003). As we have briefly mentioned previously, vision is naturally viewed as a problem of Bayesian inference. Figure 4.6a shows a simple graphical model depicting a visual inference problem (in this case, the nodes correspond to high-dimensional random variables, rather than the binary variables we have considered so far). A variety of three-dimensional (3D) objects project to the same two-dimensional (2D) image on the retina. Interpreting these data requires making an inference about the generative process, informed by prior distributions over 3D structures in the world. But it also requires making some assumptions about the dependencies between the various random variables involved.

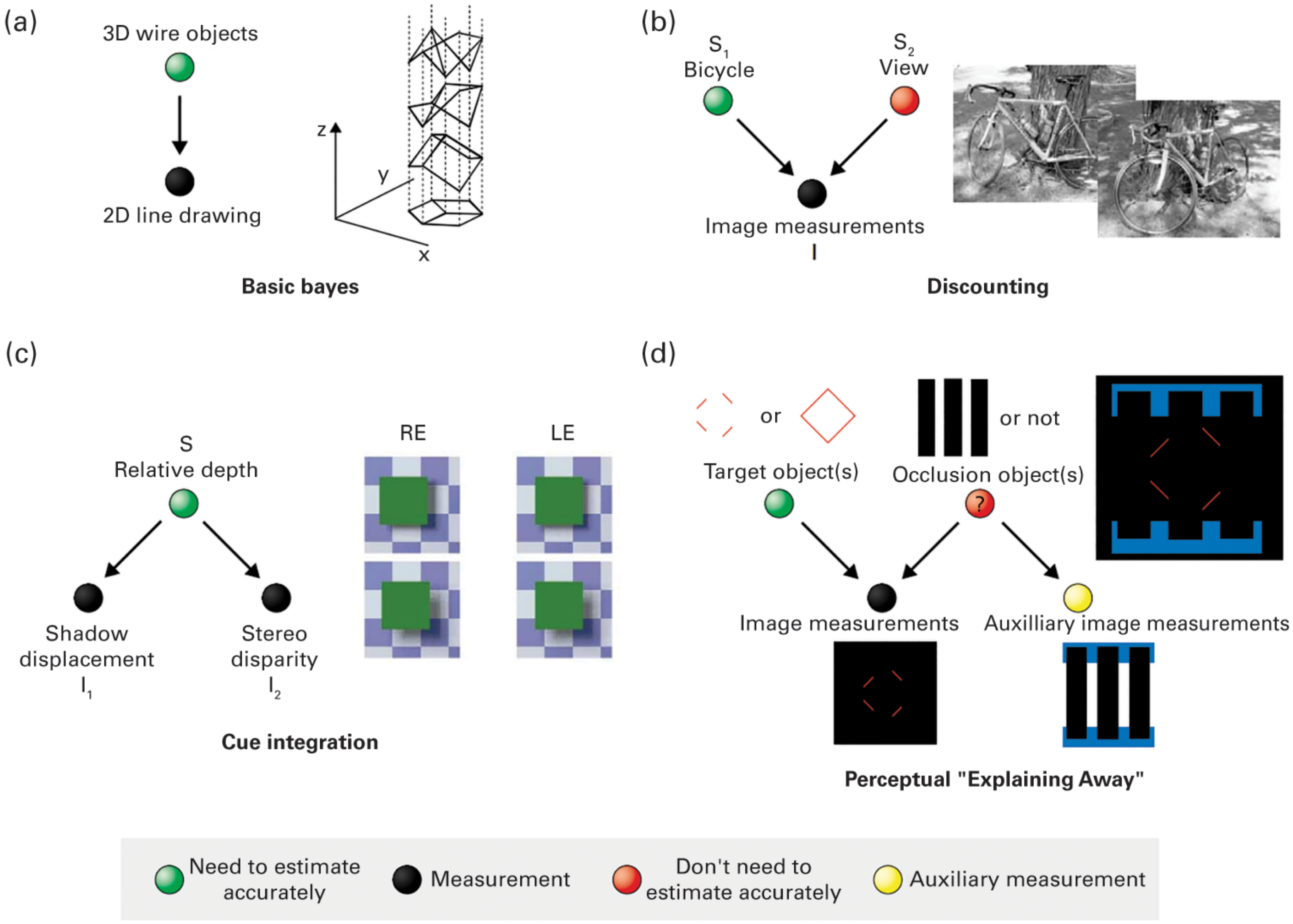

Figure 4.6

Graphical models for visual inference. (a) Formulation of the basic problem of visual inference. (b) The image of the bicycle is caused by the pose of the bicycle, the viewpoint of the camera, and the lighting conditions. (c) An example of common cause. The shading and binocular stereo cues are caused by the same event—two surfaces with one partially occluding the other. (d) Explaining away in vision. 3D = three-dimensional; 2D = two-dimensional. Figure reproduced with permission from Kersten and Yuille (2003).

It is often natural to think of multiple factors combining to generate an image, similar to a common effect structure. For example, in figure 4.6b the image of a bicycle is the common effect of the shape and reflectance properties of the bicycle and the viewpoint and illumination conditions. Often one of the causes is considered a nuisance variable and will be marginalized over. This will depend on the task. If we only want to identify the bicycle, we will not care about the viewpoint and illumination and will integrate them out. But if we want to grab the bicycle, then we need to estimate its shape and viewpoint, but the lighting can be discounted. In a few unusual situations, we may want to estimate the lighting and ignore the viewpoint or the shape and other properties of the bicycle.

Common cause structures also arise in vision, typically where there are multiple cues that could be generated from a single source. An example of a common cause structure occurs when there are two surfaces with one slightly above the other, as in figure 4.6c. There are two types of cues that indicate the relative depth of the surfaces. One cue is binocular stereo (viewing the surface with two eyes and estimating depth by trigonometry). Another cue arises from the shadow patterns thrown by the upper surface on the lower. The relationship between these cues can be captured by the graphical model shown in figure 4.6c. In this case, the random variables represent the positions in 3D space of all points on the two surfaces.

Visual inference can also demonstrate explaining away. Figure 4.6d shows a situation where two possible explanations exist for a percept: it could be a red diamond behind an occluding set of vertical lines, or four separate line segments. In this case, perceptual evidence for the occluder will change the interpretation of the red lines: if the occluder seems to be present, then the percept is a diamond, while if it is absent, it is just four line segments. This structure is directly analogous to the “psychic friend” example that we have been using throughout the chapter: it’s exactly the same graphical model, but with a different set of variables.

4.3 Causal Graphical Models

So far, we have been talking informally about causality in considering how graphical models can capture the structure of generative processes. However, recent work has resulted in a more precise specification of how graphical models can be used to represent causal processes, based on a calculus for understanding the consequences of actions, which are characterized as *interventions* on the values of variables (Pearl, 2000; Spirtes et al., 1993). In a standard Bayesian network, an edge between variables indicates only a statistical dependency between them. To reason about causality, we need to augment directed graphical models with a stronger assumption about the relationships indicated by edges: that they indicate direct causal relationships (Pearl, 2000; Spirtes et al., 1993).

4.3.1 From Graphical Models to Causal Graphical Models

The assumption that edges indicate causal relationships, not just dependency, allows causal graphical models to represent not just the probabilities of events that one might observe, but also the probabilities of events that one can produce through intervening on a system—reaching into the system and setting variables to particular values. The inferential implications of an event can differ strongly, depending on whether it was observed passively or under conditions of intervention. For example, observing that nothing happens when your friend attempts to levitate a pencil would provide evidence against her claim of having psychic powers; but secretly intervening to hold the pencil down while your friend attempts to levitate it would make the pencil’s nonlevitation unsurprising and uninformative about her powers.

In causal graphical models, the consequences of intervening on a particular variable can be assessed by removing all incoming edges to that variable and performing probabilistic inference in the resulting “mutilated” model (Pearl, 2000). This procedure produces results that align with our intuitions in the psychic powers example: intervening on the pencil levitation (X_2) breaks its connection with your friend’s psychic powers (X_3), rendering the two variables independent. As a consequence, X_2 cannot provide evidence about the value of X_3. Pearl defined a version of the Markov condition—the *causal Markov condition*—to describe how causal graphical models are used to compute probabilities under both observation and intervention.

Pearl (2000) formalized his notion of causality by introducing a “do” operator to indicate an intervention that fixes the value of a variable. Using this operator, we could write our query about our friend’s psychic powers when we have intervened to prevent the pencil from levitating in terms of the conditional probability $P(X_3|\text{do}(X_2 = 0))$, where $\text{do}(X_2 = 0)$ indicates that X_2 has been fixed to take the value 0 rather than observed to have that value. This conditional probability is different from what we would have obtained if we had just observed $X_2 = 0$, which is $P(X_3|X_2 = 0)$—the two would be evaluated in different graphical models. In Pearl’s formulation, a conditional probability that involves a “do” operator is computed by taking the causal graphical model that would normally be used to compute the relevant conditional probability and severing the incoming edges for all variables whose values are set by “do.”

Extending graphical models to capture probability distributions under intervention makes it possible to reason about and potentially learn from a wider range of data. Relationships that might be impossible to identify from observational data alone are potentially identifiable when interventions are allowed. Several papers have investigated whether people are sensitive to the consequences of intervention, generally finding that people differentiate between observational and interventional evidence appropriately (Hagmayer, Sloman, Lagnado, & Waldmann, 2007; Lagnado & Sloman, 2004; Steyvers, Tenenbaum, Wagenmakers, & Blum, 2003). Introductions to causal graphical models that consider applications to human cognition are provided by Glymour (2001) and Sloman (2005).

4.3.2 An Example: Do We “Do”?

Pearl (2000) provided a way to use causal graphical models ot reason about the consequences of intervening on particular variables, in the form of his “do” operator. A natural question to ask is whether people make the same kind of inferences. Sloman and Lagnado (2005) explored this question in a series of experiments. The first of these presented participants with a simple scenario: “There are three billiard balls on a table that act in the following way: Ball 1’s movement causes Ball 2 to move. Ball 2’s movement causes Ball 3 to move.” Participants were then asked to answer two questions:

1. Imagine that Ball 2 could not move; would Ball 1 still move? Circle one of the three options:
   It could. It could not. I don’t know.
2. Imagine that Ball 2 could not move; would Ball 3 still move? Circle one of the three options:
   It could. It could not. I don’t know.

The causal graphical model underlying this scenario is one in which variables denoting the notion of the three balls (call them B_1, B_2, and B_3) are linked in a causal chain: B_1 B_2 B_3. Intervening on B_2 breaks the link from B_1 to B_2, rendering B_1 and B_2 independent. Consequently, the use of the “do” operator predicts that participants should answer the first question with “It could” and the second question with “It could not.”

Sloman and Lagnado (2005) found exactly this pattern of results—90 percent of participants answered each question in the way that Pearl’s framework predicts. A second experiment showed that similar results hold when the underlying relationship is probabilistic (as opposed to the deterministic relationship between the balls described here). Interestingly, the causal formulation seems critical to producing these results—using a “logical” scenario (“Someone is showing off her logical abilities. She is moving balls without breaking the following rules: If Ball 1 moves, then Ball 2 moves. If Ball 2 moves, then Ball 3 moves.”) did not produce the same pattern of results.

Pearl (2000) also outlined a way in which the “do” operator is also used in counterfactual reasoning—basically, in forming a counterfactual, we are imagining the consequence of intervening on a variable. Sloman and Lagnado (2005) also found that people’s counterfactual reasoning was broadly consistent with Pearl’s account, and subsequent work has built on this in investigating the formal structure of human counterfactual reasoning in more detail (for a review, see Sloman & Lagnado, 2015).

4.4 Learning Graphical Models

We have seen how graphical models can be used to specify complex generative models and capture causal relationships—key components of the intuitive models of the world around us that we build. But people need to be able to *learn* those world models from experience. We now turn to the question of how graphical models can be learned.

4.4.1 Different Approaches to Causal Learning

The prospect of using graphical models to express the probabilistic consequences of causal relationships has led researchers in several fields to ask whether these models could serve as the basis for learning causal relationships from data. Every introductory class in statistics teaches that “correlation does not imply causation.” This is certainly the case, but it doesn’t mean that correlation carries *no* information about causation. Patterns of causation imply patterns of correlation, meaning that a learner should be able to work backwards from observed correlations (or statistical dependencies) to make probabilistic inferences about the underlying causal structures likely to have generated those observed data.

Constructing a graphical model from a set of observed data involves two kinds of learning: structure learning and parameter estimation. *Structure learning* refers to identification of the topology of the underlying graph, while *parameter estimation* involves determining the parameters of the conditional probability distributions of the different variables. Structure learning is arguably more fundamental than parameter estimation, since the parameters can only be estimated once the structure is known. Learning the structure of a graph defined on many variables is a difficult computational problem, as the number of possible structures is a super exponential function of the number of variables (Koller & Friedman, 2009).

Solutions to the problem of parameter estimation take the general approach for inferring parameter values presented in chapter 3 (e.g., Heckerman, 1998). In particular, maximum-likelihood estimation or Bayesian methods can be used, with the likelihood function being based on the probability of the observed values of the variables in the graph. In typical machine-learning applications, Bayesian networks are learned from large databases that provide multiple observations of the values of these variables. For example, the parameters for a Bayesian network connecting diseases with symptoms could be estimated from a database of patients, each of whom has some symptoms and a diagnosis, providing multiple samples from the joint distribution implied by the network.

Structure learning attempts to identify the dependency structure underlying a set of observed data. There are two major approaches to structure learning: constraint-based learning and Bayesian inference. Constraint-based algorithms for structure learning (e.g., Pearl, 2000; Spirtes et al., 1993) proceed in two steps. First, standard statistical hypothesis tests such as Pearson’s ^{2} test are used to identify which variables are dependent and independent. Since the Markov condition implies that different causal structures should result in different patterns of dependency among variables, the observed dependencies provide constraints on the set of possible structures. The second step of the algorithms identifies this set, reasoning deductively from the pattern of dependencies. The result is one or more structures that are consistent with the statistically significant dependencies exhibited by the data.

In contrast, the Bayesian approach to structure learning evaluates each graph structure in terms of the probability that it assigns to a data set. By integrating over the values that parameters could assume, it is possible to compute the probability of a data set given a graphical structure without committing to a particular choice of parameter values (e.g., Cooper & Herskovits, 1992). This computation is a form of model selection, as discussed in detail in chapter 3. Often, priors either are uniform (giving equal probability to all graphs) or give lower probability to more complex structures. Bayesian structure learning proceeds by either searching the space of structures to find the one with the highest posterior probability (Friedman, 1997; Heckerman, 1998), or evaluating particular causal relationships by integrating over the posterior distribution over graphs using sophisticated Monte Carlo methods (Friedman & Koller, 2000).

Constraint-based and Bayesian approaches to causal learning represent two philosophies, with constraint-based methods relying on the discrete results from frequentist statistical tests to turn an inductive problem into a deductive problem, and Bayesian methods casting structure learning as a special case of Bayesian inference. These two approaches have different advantages and disadvantages. Constraint-based methods are potentially more scalable, as the Bayesian approach is highly computationally intensive. However, the Bayesian approach makes it possible to integrate multiple pieces of weak evidence and incorporate prior knowledge—something that is important when modeling human cognition.

4.4.2 An Example: Structure and Strength in Causal Induction

Much psychological research on causal induction has focused on this simple causal learning problem: given a candidate cause, C[2], and a candidate effect, E, people are asked to give a numerical rating assessing the degree to which C causes E. The exact wording of the judgment question varies, and until recently it was not the focus of much attention, although as we will see, it is potentially quite important. Most studies present information corresponding to the entries in a 2 × 2 contingency table, as in table 4.1. People are given information about the frequency with which the effect occurs in the presence and absence of the cause, represented by the numbers $N(e^+, c^+)$, $N(e, c)$ and so forth. In a standard example, C might be injecting a chemical into a mouse and E the expression of a particular gene. $N(e^+, c^+)$ would be the number of injected mice expressing the gene, while $N(e, c)$ would be the number of uninjected mice not expressing the gene. We refer to tasks of this sort as *elemental causal induction* tasks.

Table 4.1
Contingency table representation used in elemental causal induction

	Effect Present (e^+)	Effect Absent (e)
Cause Present (c^+)	$N(e^+, c^+)$	$N(e, c^+)$
Cause Absent (c)	$N(e^+, c)$	$N(e, c)$

Psychological Models of Elemental Causal Induction The leading psychological models of elemental causal induction are measures of association that can be computed from simple combinations of the frequencies in table 4.1. A classic model first suggested by Jenkins and Ward (1965) asserts that the degree of causation is best measured by the quantity *P*, defined as

$$\Delta P = \frac{N(e^+,c^+)}{N(e^+,c^+)+N(e^-,c^+)} - \frac{N(e^+,c^-)}{N(e^+,c^-)+N(e^-,c^-)} = P(e^+|c^+) - P(e^+|c^-), \tag{4.24}$$

where $P(e^+|c^+)$ is the empirical conditional probability of the effect given the presence of the cause, estimated from the contingency table counts $N(\cdot)$. *P* thus reflects the change in the probability of the effect occuring as a consequence of the occurence of the cause. An alternative model proposed by Cheng (1997) suggested that people's judgments are better captured by a measure called *causal power*:

$$\text{power} = \frac{\Delta P}{1 - P(e^+|c^-)}, \tag{4.25}$$

which takes *P* as a component but predicts that *P* will have a greater effect when $P(e^+|c)$ is large. Intuitively, dividing by 1 $P(e^+|c)$ normalizes *P* by the maximum value that it could possibly take. It thus estimates the proportion of the times when *E* wouldn't occur on its own that it *does* occur when *C* is present—that is, the power of *C* to influence *E*.

Several experiments have been conducted with the aim of evaluating *P* and causal power as models of human jugments. In one such study, Buehner and Cheng (1997, experiment 1B; this experiment also appears in Buehner, Cheng, & Clifford, 2003) asked people to evaluate causal relationships for 15 sets of contingencies expressing all possible combinations of $P(e^+|c)$ and *P* in increments of 0.25. Specifically, they were asked to rate how strongly the cause produced the effect on a scale from 0 ("not at all") to 100 ("every time"). The results of this experiment are shown in figure 4.7, together with the predictions of *P* and causal power. As can be seen from the figure, both *P* and causal power capture some of the trends in the data, producing correlations of *r* = 0.89 and *r* = 0.88, respectively. However, since the trends predicted by the two models are essentially orthogonal, neither model provides a complete account of the data.[3]

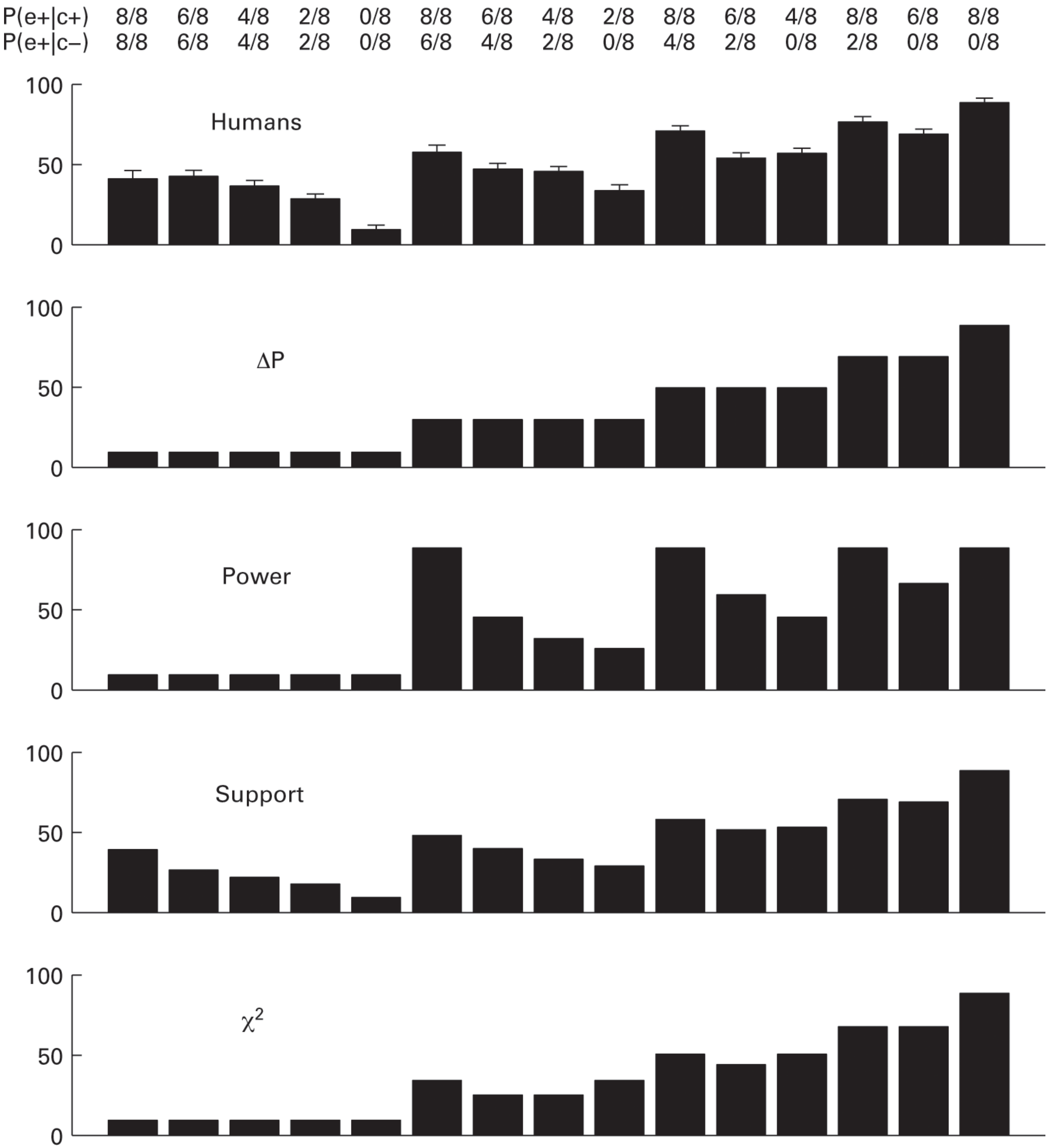

Figure 4.7
Predictions of models compared with the performance of human participants from Buehner and Cheng (1997, experiment 1B). The numbers across the top of the figure show stimulus contingencies. The bars show the magnitude of mean human ratings for how strongly the cause produced the effect, on a scale from 0–100, with error bars showing one standard error, and the corresponding model predictions. “Power” refers to causal power (Cheng, 1997), “Support” to causal support (Griffiths & Tenenbaum, 2005), and ^{2} to the Pearson ^{2} test statistic for the corresponding contingency table. Adapted from Griffiths and Tenenbaum (2005).

Formulating the Problem Using Graphical Models*P* and causal power seem to capture some important elements of human causal induction but miss others. We can gain some insight into the assumptions behind these models, and identify some possible alternative models, by considering the computational problem behind causal induction using the tools of causal graphical models and Bayesian inference. The task of elemental causal induction can be seen as trying to infer which causal graphical model best characterizes the relationship between variables *C* and *E*. Figure 4.8 shows two possible causal structures relating *C*, *E*, and another variable, *B*, which summarizes the influence of all of the other "background" causes of *E* (which are assumed to be constantly present). The problem of learning which causal graphical model is correct has two aspects: inferring the right causal structure, a problem of model selection; and determining the right parameters assuming a particular structure, a problem of parameter estimation.

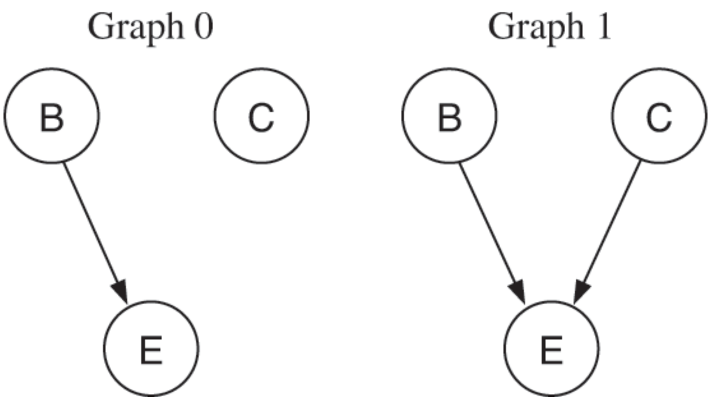

Figure 4.8
Directed graphs involving three variables, *B*, *C*, and *E*, relevant to elemental causal induction. *B* represents background variables, *C* a potential causal variable, and *E* the effect of interest. *Graph* 1 is assumed for computing *P* and causal power. Computing causal support involves comparing the structure of graph 1 to that of graph 0, in which *C* and *E* are independent.

To formulate the problems of model selection and parameter estimation more precisely, we need to make some further assumptions about the nature of the causal graphical models shown in figure 4.8. In particular, we need to define the form of the conditional probability distribution $P(E|B, C)$ for the different structures, often called the *parameterization* of the graphs.

Sometimes the parameterization is trivial—for example, *C* and *E* are independent in *Graph* 0, so we just need to specify $P_0(E|B)$, where the subscript indicates that this probability is associated with graph 0. This can be done using a single numerical parameter w_0, which provides the probability that the effect will exist in the presence of the background cause, $P_0(e^+|b^+; w_0) = w_0$. However, when a node has multiple parents, there are many different ways in which the functional relationship between causes and effects could be defined. For example, in graph 1, we need to account for how causes *B* and *C* interact to produce the effect *E*.

Different Parameterizations of Causal RelationshipsA simple and widely used parameterization for Bayesian networks of binary variables is the *noisy-OR distribution* (Pearl, 1988). The noisy-OR can be given a natural interpretation in terms of causal relations between multiple causes and a single joint effect. For graph 1, these assumptions are that *B* and *C* are both generative causes, each having an independent opportunity to produce the effect. The probability of *E* in the presence of just *B* is w_0, and in the presence of just *C*, it is w_1. When both are present, the probability of *E* occurring is $w_0 + w_1 \quad w_0 w_1$, where the last term prevents double-counting cases where both *B* and *C* produce the effect. We can write out all the cases in a single equation as follows:

$$P_1(e^+|b, c; w_0, w_1) = 1 - (1 - w_0)^b (1 - w_1)^c, \tag{4.26}$$

where w_0 and w_1 are the parameters associated with the strength of *B* and *C*, respectively, and $b^+ = c^+ = 1$ and $b = c = 0$ for the purpose of arithmetic operations.

This parameterization is called a noisy-OR because if w_0 and w_1 are both 1, equation (4.26) reduces to the logical OR function: the effect occurs if and only if *B* or *C* is present, or both are. With w_0 and w_1 in the range [0, 1], the noisy-OR softens this function but preserves its essentially disjunctive interaction: the effect occurs if and only if *B* causes it (which happens with probability w_0), *C* causes it (which happens with probability w_1), or both.

An alternative to the noisy-OR might be a linear parameterization of graph 1, asserting that the probability of *E* occuring is a linear function of *B* and *C*. This corresponds to assuming that the presence of a cause simply increases the probability of an effect by a constant amount, regardless of any other causes that might be present. The result is

$$P_1(e^+|b,c;w_0,w_1) = w_0 \cdot b + w_1 \cdot c. \tag{4.27}$$

This parameterization requires that we constrain $w_0 + w_1$ to lie between 0 and 1 to ensure that equation (4.27) results in a legal probability distribution. Because of this dependence between parameters that seem intuitively like they should be independent, such a linear parameterization is not normally used in Bayesian networks. However, it is relevant for understanding models of human causal induction.

Parameter Estimation versus Structure Learning Given a particular causal graph structure and a particular parameterization—for example, graph 1 parameterized with a noisy-OR function—inferring the strength parameters that best characterize the causal relationships in that model is straightforward. We can use any of the parameter-estimation methods discussed in chapter 3, such as maximum-likelihood or maximum a posteriori estimation, to find the values of the parameters (w_0 and w_1 in graph 1) that best fit a set of observed contingencies. Tenenbaum and Griffiths (2001b; Griffiths & Tenenbaum, 2005) showed that the two psychological models of causal induction introduced in this chapter—*P* and causal power—both correspond to maximum-likelihood estimates of the causal strength parameter w_1, but under different assumptions about the parameterization of graph 1. *P* results from assuming the linear parameterization, while causal power results from assuming the noisy-OR.

This view of *P* and causal power helps to reveal their underlying similarities and differences: they are similar in being maximum-likelihood estimates of the strength parameter describing a causal relationship, but differ in the assumptions that they make about the form of that relationship. This analysis also suggests another class of models of causal induction: models of learning causal graph structure or causal model selection rather than parameter estimation. Recalling our discussion of model selection, we can express the evidence that a set of contingencies *d* provide in favor of the existence of a causal relationship (i.e., graph 1 over graph 0) as the log-likelihood ratio in favor of graph 1. Terming this quantity *causal support*, we have

$$\text{support} = \log \frac{P(d|\text{Graph 1})}{P(d|\text{Graph 0})}, \tag{4.28}$$

where *P*(*d*|*Graph* 1) and *P*(*d*|*Graph* 0) are computed by integrating over the parameters associated with the different structures:

$$P(d|\text{Graph } 1) = \int_0^1 \int_0^1 P_1(d|w_0, w_1, \text{Graph } 1)\ P(w_0, w_1|\text{Graph } 1)\ dw_0\ dw_1 \quad (4.29)$$

$$P(d|\text{Graph } 0) = \int_0^1 P_0(d|w_0, \text{Graph } 0)\ P(w_0|\text{Graph } 0)\ dw_0. \quad (4.30)$$

Tenenbaum and Griffiths (2001b; and also Griffiths & Tenenbaum, 2005) proposed this model, and specifically assumed a noisy-OR parameterization for graph 1 and uniform priors on w_0 and w_1. Equation (4.30) is related to the normalizing constant for the beta distribution and has an analytic solution (see chapter 3). Evaluating equation (4.29) is more of a challenge, but one that we will return to later in this chapter when we discuss Monte Carlo methods for approximate probabilistic inference.

The results of computing causal support for the stimuli used by Buehner and Cheng (1997) are shown in figure 4.7. Causal support provides an excellent fit to these data, with r = 0.97. The model captures the trends predicted by both P and causal power, as well as trends that are predicted by neither model. These results suggest that when people evaluate contingency, they may be taking into account the evidence that those data provide for a causal relationship, as well as the strength of the relationship they suggest. The figure also shows the predictions obtained by applying Pearson's ^{2} test to these data, a standard hypothesis-testing method of assessing the evidence for a relationship (and a common ingredient in nonBayesian approaches to structure learning; e.g., Spirtes et al., 1993). These predictions miss several important trends in the human data, suggesting that the ability to assert expectations about the nature of a causal relationship that go beyond mere dependency (such as the assumption of a noisy-OR parameterization) is contributing to the success of this model. Causal support predicts human judgments on several other data sets that are problematic for P and causal power, and also accommodates causal learning based upon the rate at which events occur (see Griffiths & Tenenbaum, 2005, for more details).

Learning More Complex Causal Relationships The Bayesian approach to causal induction can be extended to cover a variety of more complex cases, including learning and intervening in larger causal networks (e.g., Steyvers et al., 2003; Bramley, Dayan, Griffiths, & Lagnado, 2017), continuous causes and effects (e.g., Griffiths & Pacer, 2011; Davis, Bramley, & Rehder, 2020; Lu, Rojas, Beckers, & Yuille, 2016), and continuous time (e.g., Pacer & Griffiths, 2012; Pacer & Griffiths, 2015).

Modeling learning in these more complex cases often requires us to work with stronger and more structured prior distributions than were needed before to explain elemental causal induction. This prior knowledge can be usefully described in terms of intuitive domain theories (Carey, 1985; Wellman & Gelman, 1992; Gopnik & Meltzoff, 1997), systems of abstract concepts and principles that specify the kinds of entities that can exist in a domain, their properties and possible states, and the kinds of causal relations that can exist between them. These abstract *causal theories* can be formalized as probabilistic generators for hypothesis spaces of causal graphical models, using probabilistic forms of generative grammars, predicate logic, or other structured representations that we discuss in more detail later in the book (e.g., Griffiths & Tenenbaum, 2009; Kemp et al., 2010b). Given observations of causal events relating a set of objects, these probabilistic theories generate the relevant variables for representing those events, a constrained space of possible causal graphs over those variables, and the allowable parameterizations for those graphs. They also generate a prior distribution over this hypothesis space of candidate causal models, which provides the basis for Bayesian causal learning in the spirit of the methods described earlier in this chapter.

4.4.3 An Example: Multisensory Integration with Structural Uncertainty

Human observers are sensitive to both visual and auditory cues. Sometimes these cues have a common cause—for instance, you see a dog moving and hear it barking. In other situations the auditory and visual cues are due to different causes—for instance, a cat moves and a nearby dog barks. Ventriloquists are able to fake these interactions by making the audience think that a puppet is speaking by associating the sound (produced by the ventriloquist) with the movement of the puppet. The *ventriloquism* effect occurs when visual and auditory cues have different causes—and thus are in conflict—but the audience perceive them as having the same cause.

Körding et al. (2007) developed a Bayesian model intended to capture this ventriloquism effect. The model formulates this problem as one of determining whether two cues have a common cause. They formulated this using variable C to denote the causal structure, as shown in figure 4.9. When C = 1, there is a common cause behind the sensory signals, so the positions of the auditory cue X_A and the visual cue X_V are generated from the same underlying location S. The resulting joint distribution is $p(x_A$,

$x_V, s) = p(x_A|s)p(x_V|s)p(s)$. Here, $p(x_A|s)$ and $p(x_V|s)$ are Gaussian distributions with the same mean s and variances σ_A^2 and σ_V^2. It is assumed that the visual cues are more precise than the auditory cues so $\sigma_V^2 < \sigma_A^2$. The true position S is drawn from a probability distribution $p(s)$, which is assumed to be a Gaussian with mean 0 and variance σ_P^2.

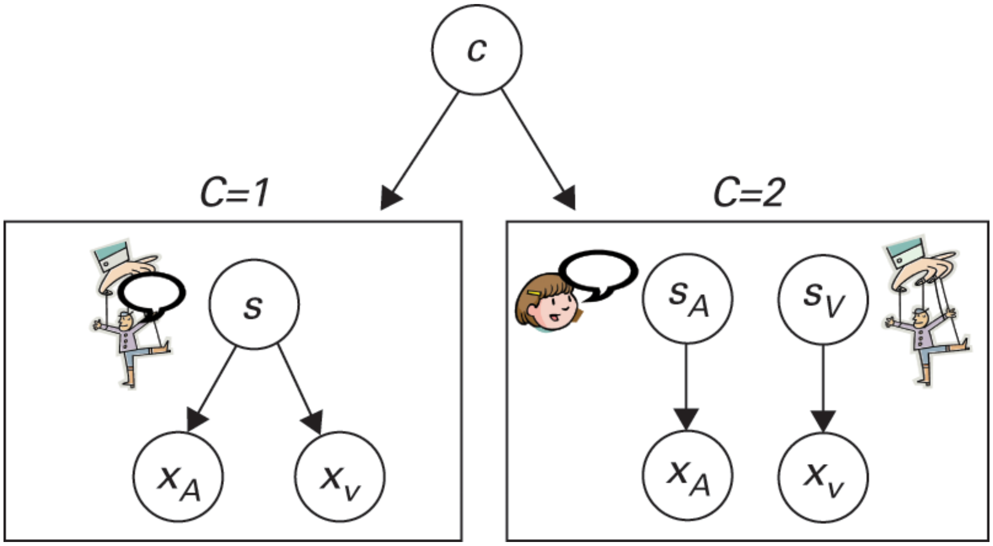

Figure 4.9
The ventriloquism effect. The participant is asked to estimate the position of the cues and to judge whether the cues are from a common cause—that is, at the same location—or not. In Bayesian terms, the task of judging whether there is a common cause can be formulated as model selection—that is, are the auditory and visual cues more likely to generated from a single cause (C = 1) or by two independent causes (C = 2)? Figure reproduced with permission from Körding et al. (2007).

By contrast, C = 2 means that the cues are generated from two locations S_A and S_B, in which case we have $p(x_A|s_A)$ and $p(x_V|s_V)$, both Gaussian with mean and variance (s_A, σ_A^2) and (s_V, σ_V^2), respectively. We assume that S_A and S_V are independent samples from a Gaussian distribution with mean 0 and variance σ_P^2. The resulting joint distribution is $p(x_A, x_V, s_A, s_V) = p(x_A|s)p(x_V|s)p(s_A)p(s_V)$.

Deciding between C = 1 and C = 2 requires performing model selection. The posterior distribution $P(C|x_A, x_V)$ is calculated by summing out the estimated locations of the cues to obtain the likelihoods $p(x_A, x_V|C)$. For C = 1, we have

$$p(x_A, x_V|C=1) = \int_{-\infty}^{\infty} p(x_A|s)p(x_V|s)p(s)\, ds, \tag{4.31}$$

while for C = 2, we have

$$p(x_A, x_V|C=2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x_A|s_A)p(x_V|s_V)p(s_A)p(s_V)\, ds_A\, ds_V. \tag{4.32}$$

There are two ways to combine the cues. The first is model selection. This estimates the most probable model $C^* = \arg\max P(C|x_V, x_A)$ from the input x_A, x_V and then uses this model to estimate the most likely positions s_A, s_V of the cues from the posterior distribution. The second way to combine the cues is by model averaging. This does not commit itself to choosing C^* but instead averages over both models:

$$P(s_V, s_A|x_V, x_A) = \sum_C P(s_V, s_A|x_V, x_A, C)P(C|x_V, x_A), \tag{4.33}$$

where $s_V = s_A = s$ when $C = 1$.

This model was compared to experiments where brief auditory and visual stimuli were presented simultaneously with varying amounts of spatial disparity. Participants were asked to identify the spatial location of the cue and whether they perceive a common cause (Wallace et al., 2004). The closer the visual stimulus was to the audio stimulus, the more likely people perceived a common cause. In this case, people's estimate of its position is strongly biased by the visual stimulus (because it is considered more precise with $\sigma_V^2 < \sigma_A^2$). But if people perceive *distinct* causes, then their estimate is pushed away from the visual stimulus and exhibits *negative bias*. Körding et al. (2007) argue that this bias is a selection bias stemming from restricting to trials in which causes are perceived as being distinct. For example, if the auditory stimulus is at the center and the visual stimulus at 5 degrees to the right of center, then sometimes the very noisy auditory cue will be close to the visual cue and hence judged to have a common cause, while on other cases, the auditory cue will be farther away (more than 5 degrees). Hence, the auditory cue will have a truncated Gaussian (if judged to be distinct) and will yield negative bias.

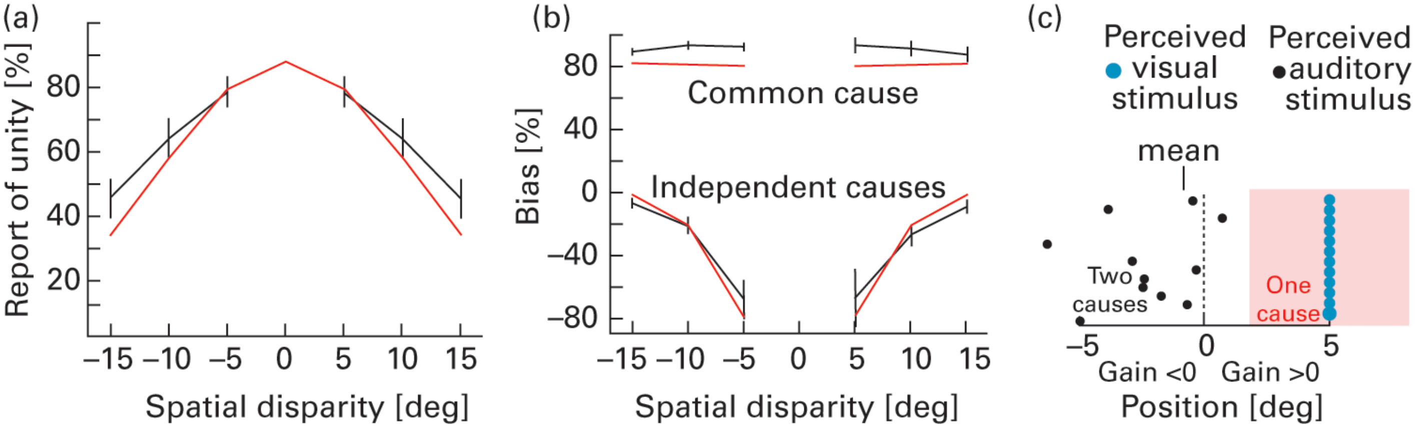

Figure 4.10

Reports of causal inference in the ventriloquism effect. (a) The relative frequency of subjects reporting one cause (black) is shown with the prediction of the causal inference model (red). (b) The bias, i.e., the influence of vision on the perceived auditory position is shown (gray and black). The predictions of the model are shown in red. (c) A schematic illustration explaining the finding of negative biases. Blue and black dots represent the perceived visual and auditory stimuli, respectively. In the pink area people perceive a common cause. Figure reproduced with permission from Körding et al. (2007).

Natarajan, Murray, Shams, and Zemel (2009) investigated these issues further. In particular, they showed that human performance on these types of experiments could be better modeled by replacing the Gaussian distributions by a more robust alternative. It is well known that Gaussian distributions are not robust because the tails of their distributions fall off rapidly, which gives very low probability to rare events. Hence, in many real-world applications, distributions with heavier tails are preferred. Following this reasoning, Natarajan et al. (2009) assumed that the observations X_A and X_V were generated by distributions with heavier tails. More precisely, they assumed that the data is distributed by a mixture of a Gaussian distribution (as in the models described here) and a uniform distribution that yields heavier tails. The resulting model was able to better account for human behavior.

4.5 Summary

Graphical models are a valuable tool both for working with probability distributions and for characterizing the inferences that people make about causal relationships. The ideas covered in this chapter are a step toward letting us define more expressive world models, and give us new tools for understanding how people might learn those models through experience. In particular, they show how a level of abstraction can be valuable in defining generative models—a theme that will become increasingly important in later chapters as we begin to consider how intuitive theories might be formalized.

In the following chapters, we will make extensive use of graphical models as we begin to work with increasingly complex probability distributions. The formalism provides us with a language that we can use to think about and define these distributions. The first step in building a Bayesian model of some aspect of human cognition is often trying to figure out a way to express the hypotheses that a learner might use—a generative model for the data that the learner gets to observe. Graphical models make that process significantly easier, allowing us to explore richer generative models.

[1] Genuine cases of cyclic causality can typically be "unrolled" over time. For example, if we say that wage growth causes inflation and inflation causes wage growth, we really mean that wage growth at time t causes inflation at time t + 1 and inflation at time t causes wage growth at time t + 1. If two variables are really coupled in a simultaneous relationship, then the solution is to represent those two variables with a single node in the Bayesian network, as if they were a single variable that takes on values corresponding to the joint distribution of the original variables.

[2] As elsewhere in this book, we will represent variables such as C and E with capital letters, and their instantiations with lowercase letters, with c^+, e^+ indicating that the cause or effect is present, and c, e indicating that the cause or effect is absent.

[3] See Griffiths and Tenenbaum (2005) for the details of how these correlations were evaluated, using a power-law transformation to allow for nonlinearities in participants' judgment scales.

5

Building Complex Generative Models

Thomas L. Griffiths and Alan Yuille

As we start to build more expressive models of the world, we run into a new problem: we may not have some of the information that we need to make an inference. If we want to determine whether a chemical causes gene expression, we need to rule out other variables that could be involved. If we want to infer whether one object caused another object to move, it might help to know the masses of those objects. If we are going to take the fact that a friend ordered pasta as an indicator of his preferences, we should check whether he saw that there was a separate pizza menu.

In the models that we have discussed so far, all the variables we have needed to reason about have either been observable data or the hypotheses or parameters that we want to infer from those data. However, as our generative models become more complex, the steps that produce the observable data are likely to also reflect some kind of underlying structure that we do not have the opportunity to observe. This structure is captured using *latent variables*.

Clustering is a classic example of a latent variable problem. Imagine that you visited an animal rescue facility and were told that they currently housed three breeds of dogs. As you walked around and looked at the dogs, you might be trying to figure out which dog was of each breed. Even if you didn't have any information about what differentiates one breed from another, you could probably come up with some good guesses based on the observed similarities between the dogs, clustering some of the dogs together in your mind.

In this setting, the cluster assignments of the dogs are latent variables—variables that influence our observations but are not themselves observable. Likewise, we might imagine documents being organized by the topics that they discuss, the properties of plants resulting from their positions in a taxonomic hierarchy, or faces being characterized in terms of a small number of meaningful psychological dimensions. Each of these representations posits a different set of latent variables, corresponding to topic assignments, nodes in trees, or locations in space that we might seek to infer from our observations.

Understanding how people make inferences of this kind is a step toward understanding how we engage in *unsupervised learning*—learning about the structure of our world without explicit labels. Some of the most impressive scientific breakthroughs have involved postulating new representations in this way—think of Mendeleev's organization of the elements into the periodic table, or Darwin's insight that species should be organized into trees. However, latent variables are also key to building more complex generative models that are able to capture some of the structure that underlies everyday experience, and hence to providing potential explanations for how people make inferences that exploit that structure. In this chapter, we will explore how such models can be defined, how their parameters can be estimated, and how they can be used to explain aspects of human cognition.

5.1 Mixture Models and Density Estimation

In general, a latent variable model involves two kinds of variables: those that we can observe and those that we cannot. The latter are the latent variables. We will use X_i to denote the observable variables and Z_i to denote the latent variables. Typically, these variables would be related via a generative model that specifies how the X_i are generated via the latent structure Z_i. Such a model can ultimately be used to make inferences about the latent structure associated with new data points, as well as providing a more accurate model of the distribution of X_i.

Perhaps the most widely used latent variable model is the *mixture model*, in which the distribution of X_i is assumed to be a mixture of several other distributions. We have briefly discussed mixtures of distributions in previous chapters as a convenient way of specifying prior distributions without making the latent variables involved explicit. In a mixture model, the latent variable Z_i indicates which component of the mixture is used to generate each observed data point X_i.

More formally, a mixture model specifies the probability of X_i as a mixture of a set of k component distributions:

$$p(x_i) = \sum_{j=1}^{k} p(x_i|z_i = j)P(z_i = j), \tag{5.1}$$

where $p(x_i|z_i = j)$ is the distribution associated with component j and $P(z_i = j)$ is the probability that a data point would be generated from that component. The underlying generative model is one in which we first sample the value of the latent variable Z_i, determining the component that will be used to generate X_i, and then sample X_i from the distribution associated with that component. Figure 5.1 shows a simple graphical model expressing this structure, suppressing the parameters used to define the underlying distributions.

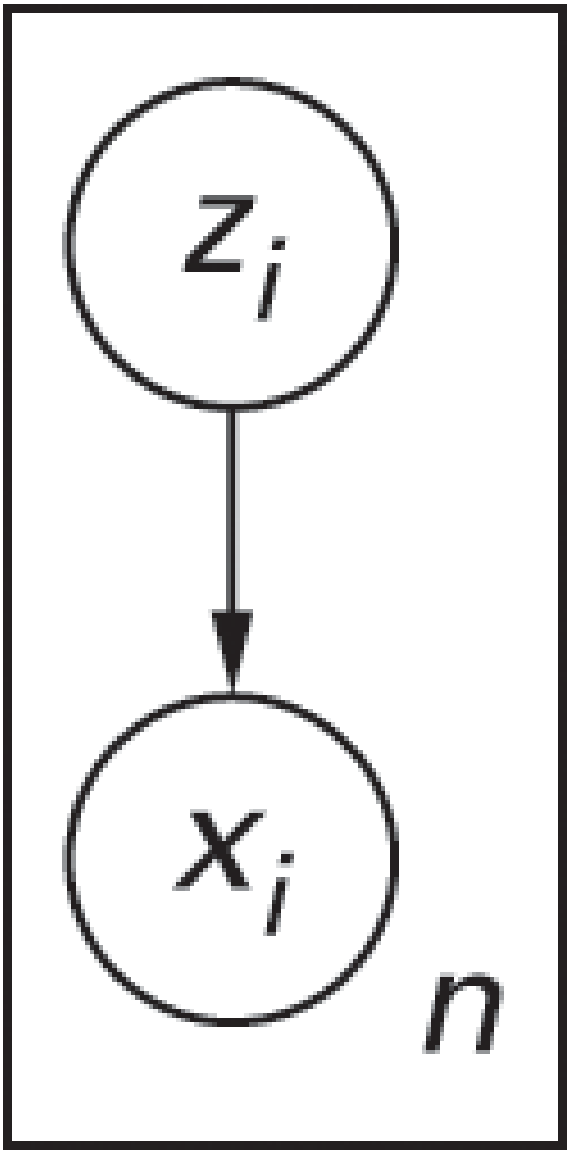

Figure 5.1

Basic graphical model for clustering with a mixture of Gaussians. Here, x_i is the ith data point and z_i its cluster assignment. The box around the variables is a plate indicating that this structure is replicated across n observations.

One common application of mixture models is to the problem of clustering. In this setting, the latent variable Z_i indicates which cluster an observation X_i comes from, and the distribution $p(x_i|z_i)$ characterizes the form of the clusters. For example, we might believe that our data were generated from two clusters, each associated with a different Gaussian distribution. $P(Z_i)$ would specify the prevalence of each Gaussian distribution in the data, and the parameters of each Gaussian would define $p(x_i|z_i)$. If we can estimate the parameters that characterize these distributions, we can infer the probable cluster membership (z_i) for any data point (x_i).

More generally, mixture models allow us to capture probability distributions that have a shape that is not a good match for any of the simple distributions discussed so far. For this reason, mixture models are often used for *density estimation*—estimating the form of a probability density function. This use of mixture models turns out to have an interesting connection to models of how humans represent categories.

5.1.1 An Example: Mixture Models and Categorization

In chapter 3, we briefly discussed a model of memory in which categories were represented as Gaussian distributions (Huttenlocher et al., 2000). This way of representing categories makes it easy not just to reconstruct objects from memory, but also to decide upon the category membership of those objects. The presentation here draws on that in Griffiths, Sanborn, Canini, Navarro, and Tenenbaum (2011a).

Categorization as Bayesian InferenceA standard way to formalize the problem of categorization in psychology is to assume that people are given a set of n 1 stimuli with features $\mathbf{x}_{n1} = (x_1, x_2, \ldots, x_{n1})$ and category labels $\mathbf{c}_{n1} = (c_1, c_2, \ldots, c_{n1})$, and need to compute the probability that a new stimulus with features x_n is assigned to category c. We can calculate this probability by applying Bayes' rule, with

$$P(c_n = c|x_n, \mathbf{x}_{n-1}, \mathbf{c}_{n-1}) = \frac{p(x_n|c_n = c, \mathbf{x}_{n-1}, \mathbf{c}_{n-1})P(c_n = c|\mathbf{c}_{n-1})}{\sum_c p(x_n|c_n = c, \mathbf{x}_{n-1}, \mathbf{c}_{n-1})P(c_n = c|\mathbf{c}_{n-1})}, \tag{5.2}$$

where the posterior probability of category c is proportional to the product of the probability of an object with features x_n being produced from that category and the prior probability of choosing that category, taking into account the features and labels of the previous n 1 objects (assuming that only category labels influence the prior).

This formulation of the problem of categorization makes it clear that learning a category is a problem of determining the form of these probability distributions—a problem of density estimation. From the previous observations $\mathbf{x}_{n1}$ and $\mathbf{c}_{n1}$ we need to infer the probability density for the distribution of x_n given each value of c_n. One strategy for solving this problem is to assume that each category is a distribution of a standard form—such as a Gaussian—and estimate the parameters of these distributions from the previous observations. This can be done using the methods for maximum-likelihood estimation or maximum a posteriori (MAP) estimation introduced in chapter 3. Such an approach is called *parametric density estimation*, as it reduces the problem to parameter estimation (e.g., Rice, 1995).

Categories and Mixtures Mixture models provide a more flexible way to define probability distributions. Intuitively, it seems natural to think about certain kinds of categories as mixtures. To return to the example from the introduction to this chapter, the category of dogs contains many breeds that vary in their attributes and might each be captured by a distinct mixture component. We might thus try to solve our density estimation problem using mixture models.

Rosseel (2002) proposed an account of human category learning based on this idea. The Mixture Model of Categorization assumes that $P(x_n|c_n = c, \mathbf{x}_{n1}, \mathbf{c}_{n1})$ is a mixture distribution. Specifically, the model assumes that each object x_i comes from a cluster z_i, and each cluster is associated with a probability distribution over the features of the objects generated from that cluster.

When evaluating the probability of a new object x_n, it is necessary to sum over all the clusters from which that object might have been drawn, with

$$p(x_n|c_n = c, \mathbf{x}_{n-1}, \mathbf{c}_{n-1}) = \sum_{j=1}^{k_c} p(x_n|z_n = j, \mathbf{x}_{n-1}, \mathbf{z}_{n-1}) P(z_n = j|\mathbf{z}_{n-1}, c_n = c, \mathbf{c}_{n-1}), \quad (5.3)$$

where k_c is the total number of clusters for category c, $p(x_n|z_n = j, \mathbf{x}_{n1}, \mathbf{z}_{n1})$ is the probability of x_n under cluster j, and $P(z_n = j|\mathbf{z}_{n1}, c_n = c, \mathbf{c}_{n1})$ is the probability of generating a new object from cluster j in category c. The clusters can either be shared between categories or specific to a single category (in which case $P(z_n = j|\mathbf{z}_{n1}, c_n = c, \mathbf{c}_{n1})$ is 0 for all clusters not belonging to category c).

Psychological Models of Categorization While we have explicitly formulated the problem of categorization in terms of Bayesian inference and density estimation, psychological models of human categorization have traditionally been defined in different terms. The account presented here focuses on Marr’s (1982) computational level, considering the abstract problem that human minds need to solve and its ideal solution. Previous psychological models have been expressed at the algorithmic level, focusing on the representations and processes that support categorization. Our presentation of these models in this section is based on Griffiths et al. (2008c), which provides further details and information about ways in which these models can be extended.

Psychological models typically identify the problem of categorization as one of assigning stimuli to categories based on a subjective sense of similarity (e.g., Reed, 1972; Medin & Schaffer, 1978; Nosofsky, 1986). In this formulation, the probability that x_n is assigned to category c is given by

$$P(c_n = c|x_n, \mathbf{x}_{n-1}, \mathbf{c}_{n-1}) = \frac{\eta_{nc}\beta_c}{\sum_c \eta_{nc}\beta_c}, \quad (5.4)$$

where η_{nc} is the similarity of stimulus x_n to category c and β_c is the response bias for category c, capturing how much people are predisposed to producing that category label. Different models of categorization can be defined by making different assumptions about how η_{nc}, the similarity of a stimulus to a category, is computed. Three such strategies are illustrated in figure 5.2.

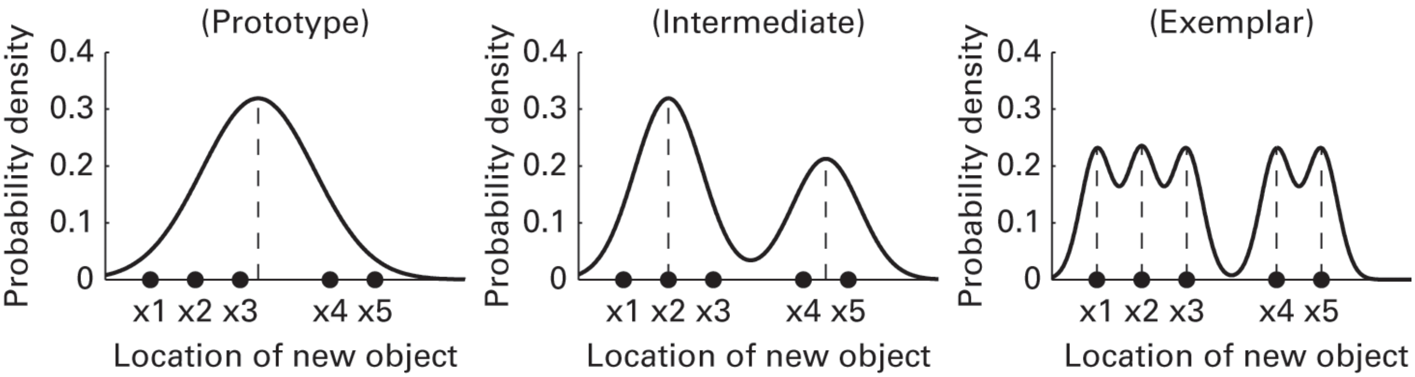

Figure 5.2
Different models of categorization can be expressed as different kinds of mixture models. In the prototype model (left), a category is represented as a single parametric probability distribution, such as a Gaussian. In an exemplar model (right), the category is represented as the sum of a set of kernels centered on the exemplars. This can be thought of as a mixture model with as many components as there are data points. Mixture models with fewer clusters (center) provide a way to interpolate between these extremes.

In a *prototype model* (e.g., Reed, 1972), category *c* is represented by a single prototypical instance. Under this account, your category of dogs would be represented by a single prototypical dog that captures the general properties of dogs. To formalize this, the similarity of stimulus *n* to category *c* is defined to be

$$\eta_{nc} = \eta_{np_c}, \tag{5.5}$$

c *c* *np* *c*

where *p* is the prototypical instance of the category and is a measure of the similarity between stimulus *n* and prototype *p* . One common way of defining the prototype is as the centroid of all instances of the category in some psychological space; that is,

$$p_c = \frac{1}{n_c} \sum_{i|c_i=c} x_i, \tag{5.6}$$

c *i*

where *n* is the number of instances of the category (i.e., the number of stimuli for which *c* = *c*).

In an *exemplar model* (e.g., Medin & Schaffer, 1978; Nosofsky, 1986), a category is represented by all the stored instances of that category. Under this account, your representation of the category of dogs is simply all the dogs you have ever seen. The similarity of stimulus *n* to category *c* is calculated by summing the similarity of the stimulus to all stored instances of the category. That is,

$$\eta_{nc} = \sum_{i|c_i=c} \eta_{ni}, \tag{5.7}$$

ni *n* *i*

where is a symmetric measure of the similarity between the two stimuli *x* and *x* . The similarity measure is typically either an exponential or a Gaussian function of the distance between the two stimuli.

Exemplar and prototype models represent two extreme solutions to the problem of defining the similarity between stimuli. Vanpaemel, Storms, and Ons (2005) observed that we can formalize a set of interpolating models by allowing the instances of

c *c*

each category to be partitioned into clusters, where the number of clusters *k* ranges from 1 to *n* . Then each cluster is represented by a prototype, and the similarity of stimulus *n* to category *c* is defined to be

$$\eta_{nc} = \sum_{j=1}^{k_c} \eta_{np_{jc}}, \tag{5.8}$$

where p_{jc} is the prototype of cluster *j* in category *c*. When $k_c = 1$ for all *c*, this is equivalent to the prototype model, and when $k_c = n_c$ for all *c*, this is equivalent to the exemplar model.

Connecting Levels of Analysis While they were originally proposed purely as an account of the psychological processes behind categorization, we can actually give a reasonable computational-level interpretation of prototype and exemplar models. Ashby and Alfonso-Reese (1995) observed a connection between the Bayesian solution to the problem of categorization presented in equation (5.2) and the way that the probabilities of category membership is computed in exemplar and prototype models (i.e., equation (5.4)). Specifically, $_{nc}$ can be identified with $p(x_n|c_n = c, \mathbf{x}_{n1}, \mathbf{c}_{n1})$, while $_{c}$ captures the prior probability of category *c*, $p(c_n = c|\mathbf{c}_{n1})$. The difference between exemplar and prototype models thus comes down to different ways of estimating $p(x_n|c_n = c, \mathbf{x}_{n1}, \mathbf{c}_{n1})$.

The definition of $_{nc}$ used in an exemplar model (equation (5.7)) corresponds to estimating $P(x_n|c_n = c, \mathbf{x}_{n1}, \mathbf{c}_{n1})$ as the sum of a set of functions (known as *kernels*) centered on the x_i that are already labeled as belonging to category *c*, with

$$p(x_n|c_n = c, \mathbf{x}_{n-1}, \mathbf{c}_{n-1}) \propto \sum_{i|c_i=c} f(x_n, x_i), \tag{5.9}$$

where $f(x, x_i)$ is a probability distribution centered on x_i.[1] This is a method that is widely used for approximating distributions in statistics, being a form of *nonparametric density estimation* (meaning that it can be used to identify distributions without assuming that they come from an underlying parametric family) called *kernel density estimation* (e.g., Silverman, 1986).

The definition of $_{nc}$ used in a prototype model (equation (5.5)) corresponds to estimating $p(x_n|c_n = c, \mathbf{x}_{n1}, \mathbf{c}_{n1})$ by assuming that the distribution associated with each category comes from an underlying parametric family, and then finding the parameters that best characterize the instances labeled as belonging to that category. The prototype corresponds to these parameters—for a Gaussian distribution, it would be the mean. Again, this is a common method for estimating a probability distribution—it is the parametric density estimation strategy we introduced earlier.

The interpretation of exemplar and prototype models as different schemes for density estimation suggests that a similar interpretation might be found for interpolating models. Indeed, the corresponding solution is given by Rosseel's (2002) Mixture Model of Categorization. By a similar argument to that used for the exemplar model, we can connect equation (5.3) with the definition of $_{nc}$ in equation (5.8), providing a justification for adopting representations that interpolate between exemplars and prototypes. In fact, this model can produce all the kinds of representations that we have discussed: it reduces to kernel density estimation when each stimulus has its own cluster and the clusters are equally weighted, and parametric density estimation when each category is represented by a single cluster.

5.2 Mixture Models as Priors

Mixture models provide a good way to define complex distributions using simple parts. In chapter 3, we introduced the idea of using conjugate priors to simplify Bayesian inference for continuous variables, but many plausible prior distributions are not in the family of conjugate priors for a given likelihood function. Taking a prior that is a mixture of conjugate priors retains much of the attractive tractability that conjugate priors provide, but also allows us to define more expressive prior distributions. As a simple example, we can return to the problem of estimating the mean of a Gaussian and see what happens when we use a mixture of Gaussians rather than a single Gaussian as a prior.

Assume that we observe data *x* generated from a Gaussian distribution with mean and standard deviation . Rather than simply assuming a Gaussian prior on as we did in chapter 3, assume that the prior on is a mixture distribution with $p(\mu) = \sum_z p(\mu|z)P(z)$, where *z* ranges over the components of the mixture and *p*(|*z*) is Gaussian. We can then calculate the joint posterior distribution on *z* and given an observation *x*, *p*(, *z*|*x*) by applying Bayes' rule. If we choose to factorize this joint distribution as *p*(|*x*, *z*)*P*(*z*|*x*), we obtain the following expression for the posterior mean:

$$\bar{\mu} = \int \mu \sum_z p(\mu|x,z)P(z|x)\, d\mu \tag{5.10}$$

$$= \sum_z \int \mu p(\mu|x,z)P(z|x)\, d\mu \tag{5.11}$$

$$= \sum_z P(z|x) \int \mu p(\mu|x,z)\, d\mu, \tag{5.12}$$

where we seize the opportunity to change the order of summation for probability distributions (in this case, switching around the order in which we evaluate the sum and the integral).

Since *p*(|*x*, *z*) is the posterior distribution on using the Gaussian associated with mixture component *z* as a prior, we can substitute our previous results from chapter 3 for the posterior mean of under a Gaussian prior for *p*(|*x*, *z*) *d*. We can write this posterior mean as

$$\bar{\mu} = \frac{\frac{x}{\sigma^2} + \frac{\mu_0}{\sigma_0^2}}{\frac{1}{\sigma^2} + \frac{1}{\sigma_0^2}} = \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2}x + \frac{\sigma^2}{\sigma_0^2 + \sigma^2}\mu_0, \tag{5.13}$$

2 0

where is the variance of *x* given and and σ_0^2 are the mean and variance of the prior. The posterior mean using the mixture is then the average of the posterior means obtained using each of the components as a prior, weighted by the probability of that component. In the case where all the components have the same variance σ_0^2, we obtain

$$\bar{\mu} = \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2}x + \frac{\sigma^2}{\sigma_0^2 + \sigma^2} \sum_j p(z = j|x)\mu_0^{(z)}, \tag{5.14}$$

where $\mu_0^{(j)}$ denotes the mean of the *j*th component. The posterior mean thus linearly interpolates between the observed value of *x* and the weighted average of the means of the components of the prior.

5.2.1 An Example: The Perceptual Magnet Effect

The *perceptual magnet effect* is a phenomenon that has been documented in the perception of speech sounds, in which sounds that are close to the center of a phonetic category are perceived as being more similar to one another than sounds that are close to the boundary between two categories (Iverson & Kuhl, 1995). We can give a rational explanation for why such an effect might be observed by using a model based on a mixture of Gaussians (Feldman & Griffiths, 2007; Feldman et al., 2009).

When we perceive a speech sound, we receive a continuous signal *x* and try to reconstruct the sound produced by the speaker, . Assume that the noise in the transmission of the signal is Gaussian, with the distribution of *x* given having mean and standard deviation . If there is only one category of speech sounds, and we assume that the category corresponds to a Gaussian distribution over possible values of with mean $_0$ and standard devaition $_0$, then we are back in the familiar territory of estimating the mean of a Gaussian with a Gaussian prior. That is, the posterior distribution of given *x* will be Gaussian, with a mean that interpolates between *x* and $_0$. Taking the perceived speech sound to be the expectation of given *x*, the posterior mean, implies that perceptual space will be a linear transformation of the stimulus space, compressing stimuli into the region near the mean of the category. This is illustrated in the left panel of figure 5.3.

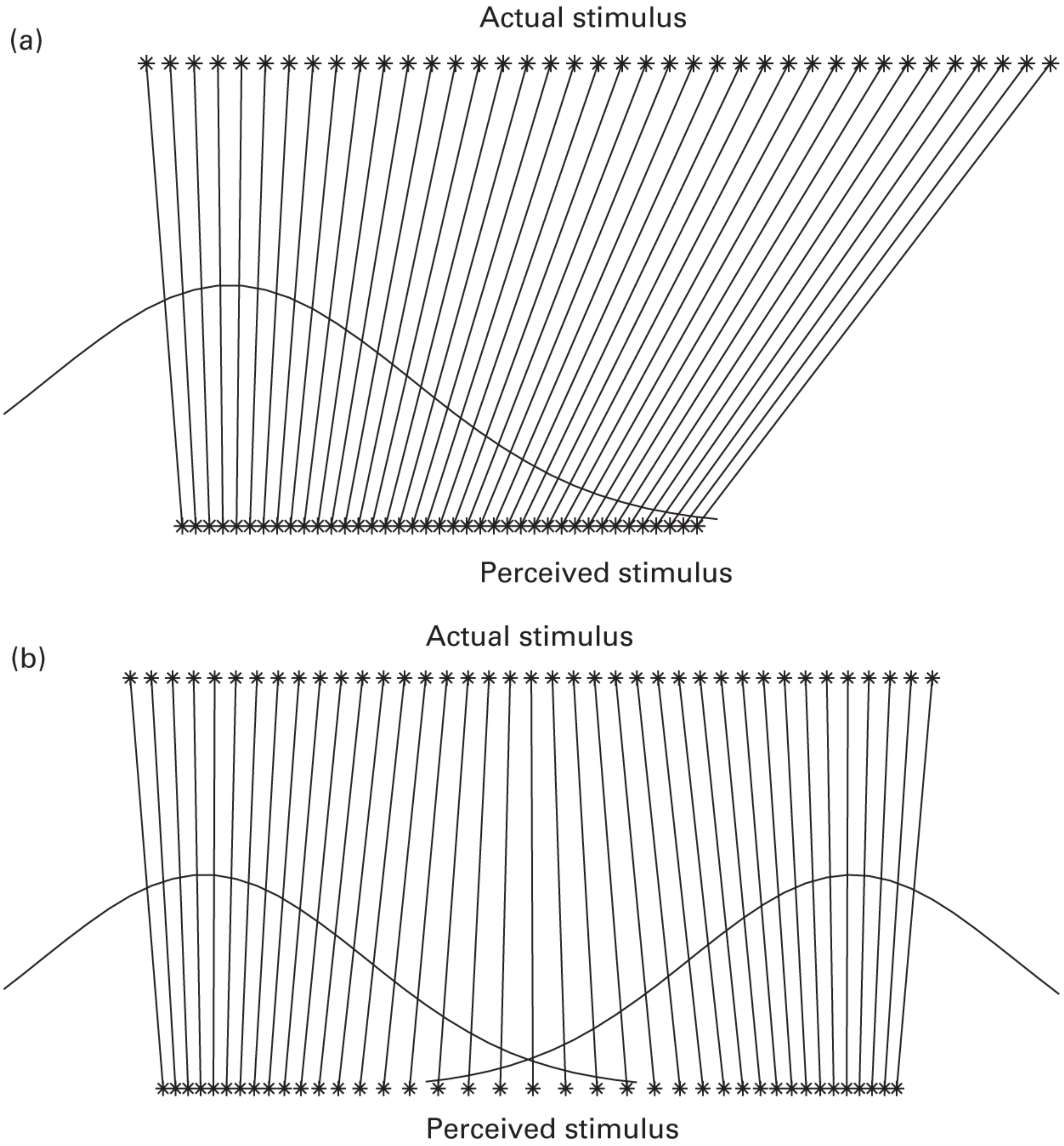

Figure 5.3
Different patterns of warping of perceptual space. With a single category, the perceived stimulus will be drawn to the mean of the category. With two categories, the perceived stimulus is drawn to the means of both categories, with the influence of each category determined by the posterior probability of having been generated from that category. As a consequence, perceptual space is compressed near the means of the categories and expanded at the boundary between categories. Figure reproduced with permission from Feldman and Griffiths (2007).

In reality, languages have multiple categories of speech sounds. The appropriate prior for is thus a mixture of Gaussians, with each component corresponding to a different speech sound. The reconstruction of a perceived speech sound will thus be affected by all the categories that the sound could belong to, with the mean of the posterior of given *x* being given by equation (5.14). The right panel of figure 5.3 illustrates the predictions that result from this equation for the case of two categories of speech sounds. When the stimulus is close to the mean of one category, reconstructions are drawn to the mean of that category. However, as they approach the region where their category membership is ambiguous, both categories exert an influence. As a consequence, the mean of the posterior is much closer to the original stimulus value. This produces a compression of perceptual space near the means of the categories and an expansion between the boundaries, exactly what occurs in the perceptual magnet effect.

5.3 Estimating Parameters in the Presence of Latent Variables

The presence of latent variables in a model poses two problems: inferring the values of the latent variables conditioned on observed data, and learning the probability distribution characterizing both the observable and latent variables. In the probabilistic framework, both these forms of inference reduce to inferring the values of unknown variables, conditioned on known variables. This is conceptually straightforward, but the computations involved are difficult and can require complex algorithms.

To understand the challenge, imagine that we have a model for data **x** that has parameters and latent variables **z**. A mixture model is one example of such a model. The likelihood for this model is $p(\mathbf{x}|) = {}_{\mathbf{z}}\ p(\mathbf{x}, \mathbf{z}|)$, where the latent variable **z** are unknown. To apply maximum-likelihood (or MAP) estimation, we would need to compute the derivative of the likelihood (or log-likelihood) with respect to . This can be a challenge since the likelihood involves a sum over **z**. In particular, this approach is intractable when there are many possible values for **z** (e.g., with k clusters and n data points, we would have to sum over k^n possible values of **z** since each of the n cluster assignments z_i can take on k values).

5.3.1 The Expectation-Maximization Algorithm

A standard approach to solving the problem of estimating probability distributions involving latent variables without needing to deal with this intractable sum is the *Expectation-Maximization (EM) algorithm* (Dempster, Laird, & Rubin, 1977). The EM algorithm is a procedure for obtaining a maximum-likelihood (or MAP) estimate for without resorting to differentiating log $p(\mathbf{x}|)$. The key idea is that we have two problems, each of which could be solved if we were able to solve the other problem: if we knew the value of the latent variable **z**, then we could find by using the standard methods for estimation discussed in the previous chapters; on the other hand, if we knew , we could compute $P(\mathbf{z}|\mathbf{x},)$ and infer the values of the latent variable **z**.

Surprisingly, we can make progress in inferring both and **z** by using of the partial knowledge we have about each. The EM algorithm for maximum-likelihood estimation proceeds by repeatedly alternating between assigning probabilities to **z** based on **x** and our current guess of , and using these probabilities to guess the values of **z** and hence estimate . More formally, this results in an iterative procedure with two steps: evaluating the expectation of the *complete log-likelihood* log $p(\mathbf{x}, \mathbf{z}|)$ with respect to $P(\mathbf{z}|\mathbf{x},)$ (the *E-step*, short for Expectation), and maximizing the resulting quantity with respect to (the *M-step*, short for Maximization). This algorithm is guaranteed to converge to a *local maximum* of $p(\mathbf{x}|)$, finding different solutions depending on the value of used to initialize it (Dempster et al., 1977).

While the EM algorithm can be used for a variety of latent variable models, we will illustrate it for the case of clustering. Assume that our observations $\mathbf{x} = (x_1, \ldots x_n)$ are generated from a Gaussian mixture model where the two components have a known common standard deviation, , and unknown means $^{(1)}$ and $^{(2)}$. The cluster assignments $\mathbf{z} = (z_1, \ldots, z_n)$ are our latent variables, each sampled from a *discrete distribution* (i.e., a multinomial distribution with just two outcomes). The probability that an observation is drawn from the first cluster, , is also unknown. The parameters that we want to estimate are thus = ($^{(1)}$, $^{(2)}$,), corresponding to the parameters in the graphical model shown in figure 5.4. We start by choosing some arbitrary values for .

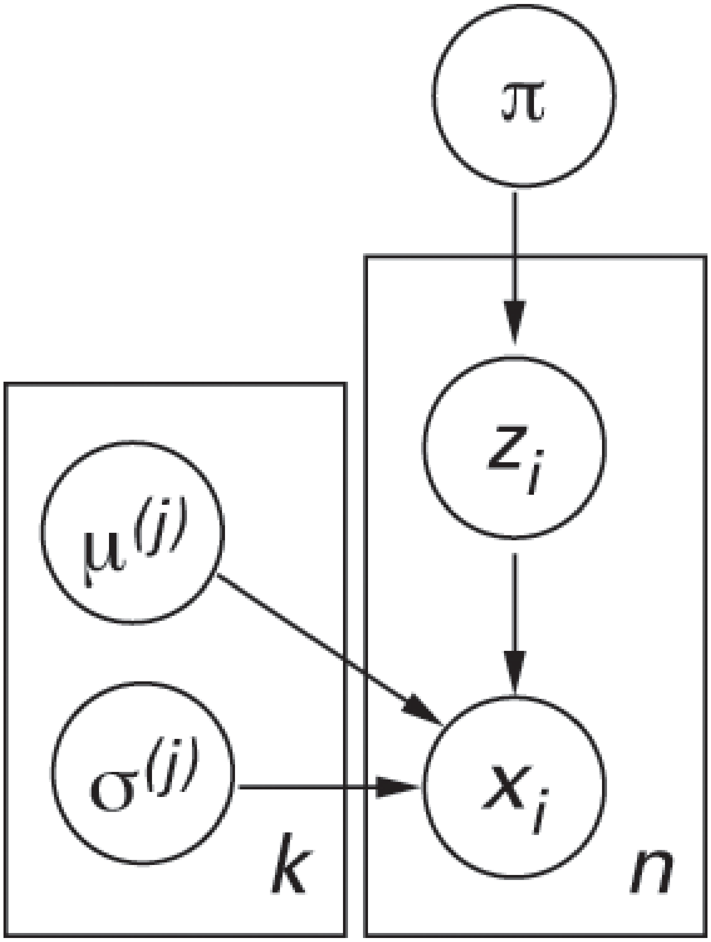

Figure 5.4

A more detailed graphical model for clustering with a mixture of Gaussians. Here, x_i is the ith data point and z_i its cluster assignment; are the parameters of the discrete distibution on z_i with $P(z_i = j) = {}_j$ and ${}^{(j)}$ and ${}^{(j)}$ for the parameters of the Gaussian corresponding to the jth cluster. The distribution $p(x_i|z_i = j, ,)$ is Gaussian(${}^{(j)}$, ${}^{(j)}$). The boxes around the variables are plates that indicate that this structure is replicated across n observations and k clusters.

The E-step In the E-step of the algorithm, we evaluate the expectation of the complete log-likelihood with respect to $P(\mathbf{z}|\mathbf{x},)$:

$$E_{P(\mathbf{z}|\mathbf{x},\theta)}[\log p(\mathbf{x},\mathbf{z}|\theta)] = \sum_{\mathbf{z}} P(\mathbf{z}|\mathbf{x},\theta)\log p(\mathbf{x},\mathbf{z}|\theta). \qquad (5.15)$$

Since each x_i is independent of all other variables given z_i and , and the z_i are independent given , this simplifies to

$$E_{P(\mathbf{z}|\mathbf{x},\theta)}[\log p(\mathbf{x},\mathbf{z}|\theta)] = \sum_{\mathbf{z}} P(\mathbf{z}|\mathbf{x},\theta)\sum_{i=1}^{n}\left[\log p(x_i|z_i,\theta) + \log P(z_i|\theta)\right]. \qquad (5.16)$$

We can then reverse the order in which we carry out the sums, obtaining

$$E_{P(\mathbf{z}|\mathbf{x},\theta)}[\log p(\mathbf{x},\mathbf{z}|\theta)] = \sum_{i=1}^{n}\sum_{\mathbf{z}} P(\mathbf{z}|\mathbf{x},\theta)\,[\log p(x_i|z_i,\theta) + \log p(z_i|\theta)] \tag{5.17}$$

$$= \sum_{i=1}^{n}\sum_{z_i} P(z_i|x_i,\theta)\,[\log p(x_i|z_i,\theta) + \log P(z_i|\theta)]\,, \tag{5.18}$$

i i i i

where the second line uses the fact that $p(x\,|z\,,\ )$ and $P(z\,|)$ are constant when we sum over **z** . Exploiting this conditional
i i

independence of the x and z is what allows us to overcome the intractable sum over all values for **z**.
i i i

Substituting our Gaussian and discrete distributions for $p(x\,|z\,,\ )$ and $P(z\,|)$, respectively, we obtain

$$E_{P(\mathbf{z}|\mathbf{x},\theta)}[\log p(\mathbf{x},\mathbf{z}|\theta)] = \sum_{i=1}^{n}\sum_{z_i} P(z_i|x_i,\theta)\left[-\frac{1}{2}\log 2\pi\sigma^2 - \frac{(x_i-\mu^{(z_i)})^2}{2\sigma^2}\right.$$

$$\left. + I(z_i=1)\log\pi + I(z_i=2)\log(1-\pi)\right],$$

$$= -\frac{n}{2}\log 2\pi\sigma^2 - \sum_{i=1}^{n}\sum_{z_i} P(z_i|x_i,\theta)\frac{(x_i-\mu^{(z_i)})^2}{2\sigma^2}$$

$$+ \sum_{i=1}^{n} P(z_i=1|x,\theta)\log\pi + \sum_{i=1}^{n} P(z_i=2|x,\theta)\log(1-\pi) \tag{5.19}$$

where $I(\cdot)$ is the indicator function, taking the value 1 when its argument is true and 0 otherwise.

The M-step In the M-step, we seek to maximize the expected complete log-likelihood with respect to . Since appears in this
old

expression in two places, it is useful to denote the “old” values of (those used in computing $P(\mathbf{z}|\mathbf{x},\ )$) as , and the “new” values
new new

that we aim to find that maximize the expected complete log-likelihood as . Our goal is thus to find the value of that
old
$P(\mathbf{z}|\mathbf{x},\ )$ new

maximizes E [log $p(\mathbf{x},\mathbf{z}|\ )$]. In the case of the mixture of Gaussians, this means maximizing the expression in equation
(1) (2)

(5.19) with respect to , , and . Differentiating this expression with respect to these parameters, setting the derivative to zero, and solving the resulting equations give

$$\hat{\mu}^{(j)} = \frac{\sum_{i=1}^{n} P(z_i = j|x_i, \theta^{\text{old}})x_i}{\sum_{i=1}^{n} P(z_i = j|x_i, \theta^{\text{old}})} \tag{5.20}$$

$$\hat{\pi} = \frac{\sum_{i=1}^{n} P(z_i = 1|x_i, \theta^{\text{old}})}{n}, \tag{5.21}$$

both of which have a very simple interpretation: the estimate of the mean of component z is the weighted average of the x_i, where the weights correspond to the probability that x_i belongs to component z, and the estimate of is the expected number of the x_i belonging to the first component.

Iterating to Convergence In the M-step, the parameters used in computing the probability that observation x_i is assigned to component z_i are the old parameters, $^{\text{old}}$, that we chose when we initialized the algorithm. However, the algorithm now iterates back and forth between the E- and M-steps, using the new estimates of produced in the M-step to compute the expectation in the E-step. The algorithm thus proceeds to develop better parameter estimates and a better idea of which component each observation came from over time, and ultimately converges to a steady state that is a local maximum of p(**x**|).

An illustration of EM for a two-dimensional (2D) mixture of Gaussians appears in figure 5.5. While the analysis given here focuses on maximum-likelihood estimation, MAP estimation can be done by including a prior p() and computing the expected complete joint log probability, log p(**x**, **z**,), in the E-step.

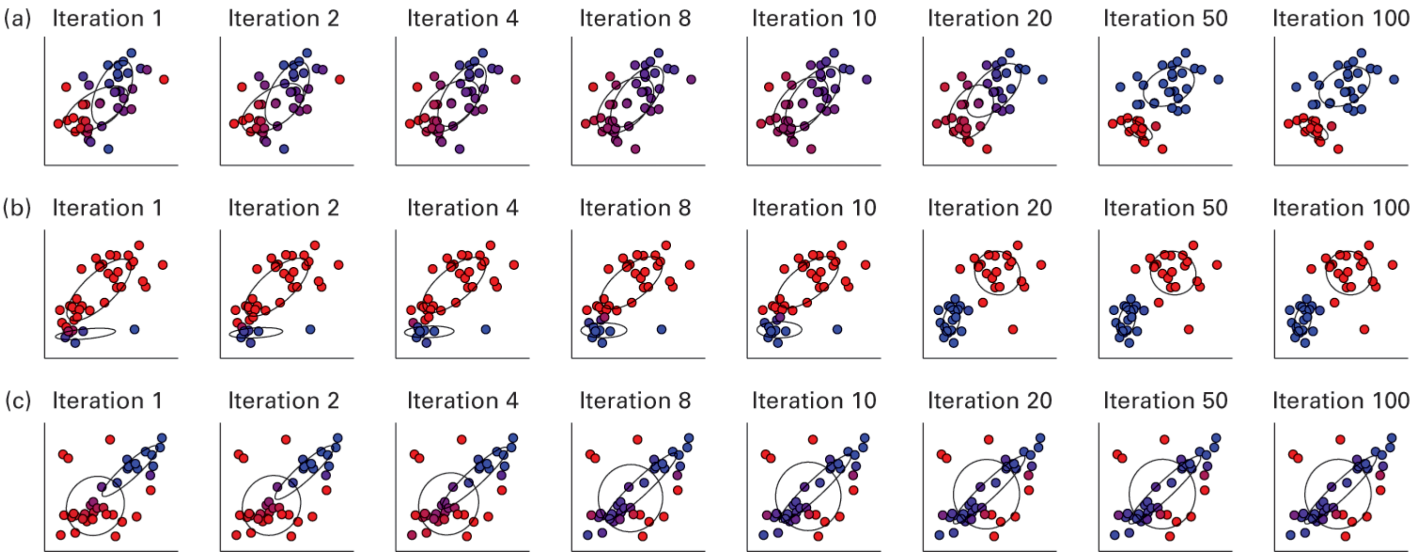

Figure 5.5

Three runs of the EM algorithm for a mixture of two Gaussians in two dimensions. Each row shows a single run of the algorithm, with different initial parameter values and data being used each time. The columns show the resulting parameter estimates after different numbers of iterations. The color of each point indicates its posterior probability of being assigned to each component, and the two Gaussians are indicated with equiprobability ellipses picking out a set of points that are equally probable under the inferred parameters. While EM often picks out natural clusters, as in (a) and (b), it only finds a local maximum of the likelihood, meaning that some initializations of the parameters result in other solutions, such as that shown in (c).

Alternating between updating the cluster assignments and updating the parameters that define the clusters is an intuitive way of solving the problem posed by clustering. It is also the strategy used in the classic *k-means clustering algorithm*, which alternates between assigning each data point to the cluster with the nearest mean and updating the cluster means based on those assignments. Indeed, the *k*-means algorithm is equivalent to the EM algorithm for a Gaussian mixture model where the standard deviation $^{(j)}$ are equal for all *j* and approach 0. In this case, $p(z_i = j|x_i,$) approaches 1 for the cluster with $^{(j)}$ closest to x_i.

5.3.2 Analyzing the EM Algorithm

The EM algorithm works by reducing a problem that is hard to solve—finding the that maximizes $p(\mathbf{x}|)$ when we have to marginalize over latent variables **z**—to two problems that are easy to solve—computing the posterior distribution on **z** when is known, $P(\mathbf{z}|\mathbf{x},$), and estimating when **z** is known by maximizing $p(\mathbf{x}, \mathbf{z}|)$. Starting with an initial guess of and alternating between these two steps result in more accurate estimates of , which then provide more accurate guesses about the values of the latent variables, which again result in more accurate estimates of . Intuitively, maximizing the expected complete log-likelihood makes sense because if we knew the exact value of **z**, we would just want to maximize the complete log-likelihood log $p(\mathbf{x}, \mathbf{z}|)$, so averaging over the possible values of **z** provides a way to take into account our uncertainty. However, it is also possible to give a more formal analysis of why the EM algorithm works.

One way of understanding the EM algorithm is to recognize that the expected complete log-likelihood is a lower-bound on the log-likehood. To see this, note that we can write

$$P(\mathbf{z}|\mathbf{x},\theta) = \frac{p(\mathbf{x},\mathbf{z}|\theta)}{p(\mathbf{x}|\theta)}. \tag{5.22}$$

Since $P(\mathbf{z}|\mathbf{x},$) is nonzero everywhere $P(\mathbf{x}, \mathbf{z}|)$ is nonzero, we can write

$$p(\mathbf{x}|\theta) = \frac{p(\mathbf{x},\mathbf{z}|\theta)}{P(\mathbf{z}|\mathbf{x},\theta)}. \tag{5.23}$$

Taking the logarithm of both sides, we have

$$\log p(\mathbf{x}|\theta) = \log p(\mathbf{x},\mathbf{z}|\theta) - \log P(\mathbf{z}|\mathbf{x},\theta). \tag{5.24}$$

If we now take the expectation of both sides with respect to $P(\mathbf{z}|\mathbf{x},$), we obtain

$$\log p(\mathbf{x}|\theta) = \sum_{\mathbf{z}} P(\mathbf{z}|\mathbf{x},\theta)\log p(\mathbf{x},\mathbf{z}|\theta) - \sum_{\mathbf{z}} P(\mathbf{z}|\mathbf{x},\theta)\log P(\mathbf{z}|\mathbf{x},\theta) \tag{5.25}$$

$$= E_{P(\mathbf{z}|\mathbf{x},\theta)}[\log p(\mathbf{x},\mathbf{z}|\theta)] + H[P(\mathbf{z}|\mathbf{x},\theta)] \tag{5.26}$$

where we use the fact that log $p(\mathbf{x}|)$ is constant in **z**, and $H[P(\mathbf{z}|\mathbf{x},$)] is the *entropy* of $P(\mathbf{z}|\mathbf{x},$), which we introduce in more detail in chapter 7. Since the entropy is nonnegative (see Cover & Thomas, 1991), $E_{P(\mathbf{z}|\mathbf{x},)}$ [log $p(\mathbf{x}, \mathbf{z}|)$]—the expected complete log-likelihood—provides a lower bound on log $p(\mathbf{x}|)$.

The EM algorithm thus alternates between computing a function that provides a lower bound on log $p(\mathbf{x}|)$ in the E-step and maximizing this function with respect to in the M-step. Both steps can also be interpreted as performing hillclimbing on a single *free energy* function that has local maxima corresponding to the local maxima of $p(\mathbf{x}|)$ (Neal & Hinton, 1998).

Applying EM is easiest in models where the latent variables are independent when conditioned on data and parameters (as in a mixture model), and the distributions for which we want to estimate parameters belong to exponential families (introduced in chapter 3). In this case, estimators typically take a form similar to that seen in equations (5.20) and (5.21), being based on observations weighted by the posterior probability of the values of the relevant latent variables. In models using more complex distributions, it is sufficient to improve the expected complete log-likelihood rather than maximizing it, so gradient descent or other optimization methods can be employed (Neal & Hinton, 1998).

5.3.3 An Example: Unsupervised Category Learning

Applying the EM algorithm to mixture models illustrates how it is possible to learn the parameters of the distributions that characterize clusters without needing any of the observations to be labeled. This kind of “unsupervised” learning characterizes much of human experience as well: while as a child, you probably had some observations labeled as “dogs” and others as “cats,” your understanding of what dogs and cats are is just as dependent on all the unlabeled examples that you have seen throughout your life. In an even more extreme case, naturalists visiting a new continent are able to recognize that they are seeing animals from different species even without those animals being given verbal labels. So how are people able to learn categories without labels?

Fried and Holyoak (1984) set out to answer this question, comparing supervised and unsupervised learning for simple categories. They defined categories by choosing some simple 2D binary arrays as “standards” and then generating other arrays by randomly modifying the arrays, as shown in figure 5.6a. They told participants that they were going to see abstract designs by two artists—Smith and Wilson—and then compared how well people learned to categorize the images by artist. In one condition, people received feedback on their decisions. In the other, they did not.