Chapter 3

Probability and Information Theory

In this chapter, we describe probability theory. Probability theory is a mathematical framework for representing uncertain statements. It provides a means of quantifying uncertainty and axioms for deriving new uncertain statements. In artificial intelligence applications, we use probability theory in two major ways. First, the laws of probability tell us how AI systems should reason, so we design our algorithms to compute or approximate various expressions derived using probability theory. Second, we can use probability and statistics to theoretically analyze the behavior of proposed AI systems.

Probability theory is a fundamental tool of many disciplines of science and engineering. We provide this chapter to ensure that readers whose background is primarily in software engineering with limited exposure to probability theory can understand the material in this book. If you are already familiar with probability theory, feel free to skip this chapter. If you have absolutely no prior experience with probability, this chapter should be sufficient to successfully carry out deep learning research projects, but we do suggest that you consult an additional resource, such as (Jaynes, 2003).

3.1 Why Probability?

Many branches of computer science deal mostly with entities that are entirely deterministic and certain. A programmer can usually safely assume that a CPU will execute each machine instruction flawlessly. Errors in hardware do occur, but are rare enough that most software applications do not need to be designed to account for them. Given that many computer scientists and software engineers work in a relatively clean and certain environment, it can be surprising that

machine learning makes heavy use of probability theory.

This is because machine learning must always deal with uncertain quantities, and sometimes may also need to deal with stochastic (non-deterministic) quantities. Uncertainty and stochasticity can arise from many sources. Researchers have made compelling arguments for quantifying uncertainty using probability since at least the 1980s. Many of the arguments presented here are summarized from or inspired by Pearl (1988).

Nearly all activities require some ability to reason in the presence of uncertainty. In fact, beyond mathematical statements that are true by definition, it is difficult to think of any proposition that is absolutely true or any event that is absolutely guaranteed to occur.

There are three possible sources of uncertainty:

1. Inherent stochasticity in the system being modeled. For example, most interpretations of quantum mechanics describe the dynamics of subatomic particles as being probabilistic. We can also create theoretical scenarios that we postulate to have random dynamics, such as a hypothetical card game where we assume that the cards are truly shuffled into a random order.

2. Incomplete observability. Even deterministic systems can appear stochastic when we cannot observe all of the variables that drive the behavior of the system. For example, in the Monty Hall problem, a game show contestant is asked to choose between three doors and wins a prize held behind the chosen door. Two doors lead to a goat while a third leads to a car. The outcome given the contestant's choice is deterministic, but from the contestant's point of view, the outcome is uncertain.

3. Incomplete modeling. When we use a model that must discard some of the information we have observed, the discarded information results in uncertainty in the model's predictions. For example, suppose we build a robot that can exactly observe the location of every object around it. If the robot discretizes space when predicting the future location of these objects, then the discretization makes the robot immediately become uncertain about the precise position of objects: each object could be anywhere within the discrete cell that it was observed to occupy.

In many cases, it is more practical to use a simple but uncertain rule rather than a complex but certain one, even if the true rule is deterministic and our modeling system has the fidelity to accommodate a complex rule. For example, the simple rule "Most birds fly" is cheap to develop and is broadly useful, while a rule of the form, "Birds fly, except for very young birds that have not yet learned to fly, sick or injured birds that have lost the ability to fly, flightless species of birds

including the cassowary, ostrich and kiwi. . . " is expensive to develop, maintain and communicate, and after all of this effort is still very brittle and prone to failure.

Given that we need a means of representing and reasoning about uncertainty, it is not immediately obvious that probability theory can provide all of the tools we want for artificial intelligence applications. Probability theory was originally developed to analyze the frequencies of events. It is easy to see how probability theory can be used to study events like drawing a certain hand of cards in a game of poker. These kinds of events are often repeatable. When we say that an outcome has a probability p of occurring, it means that if we repeated the experiment (e.g., draw a hand of cards) infinitely many times, then proportion p of the repetitions would result in that outcome. This kind of reasoning does not seem immediately applicable to propositions that are not repeatable. If a doctor analyzes a patient and says that the patient has a 40% chance of having the flu, this means something very different—we can not make infinitely many replicas of the patient, nor is there any reason to believe that different replicas of the patient would present with the same symptoms yet have varying underlying conditions. In the case of the doctor diagnosing the patient, we use probability to represent a *degree of belief*, with 1 indicating absolute certainty and 0 indicating absolute uncertainty. The former kind of probability, related directly to the rates at which events occur, is known as *frequentist probability* , while the latter, related to qualitative levels of certainty, is known as*Bayesian probability.* Bayesian probability

It turns out that if we list several properties that we expect common sense reasoning about uncertainty to have, then the only way to satisfy those properties is to treat Bayesian probabilities as behaving exactly the same as frequentist probabilities. For example, if we want to compute the probability that a player will win a poker game given that she has a certain set of cards, we use exactly the same formulas as when we compute the probability that a patient has a disease given that she has certain symptoms. For more details about why a small set of common sense assumptions implies that the same axioms must control both kinds of probability, see (Ramsey, 1926).

Probability can be seen as the extension of logic to deal with uncertainty. Logic provides a set of formal rules for determining what propositions are implied to be true or false given the assumption that some other set of propositions is true or false. Probability theory provides a set of formal rules for determining the likelihood of a proposition being true given the likelihood of other propositions.

3.2 Random Variables

A *random variable* is a variable that can take on different values randomly. We typically denote the random variable itself with a lower case letter in plain typeface, and the values it can take on with lower case script letters. For example, x_1 and x_2 are both possible values that the random variable x can take on. For vector-valued variables, we would write the random variable as $\mathbf{x}$ and one of its values as $\boldsymbol{x}$. On its own, a random variable is just a description of the states that are possible; it must be coupled with a probability distribution that specifies how likely each of these states are.

Random variables may be discrete or continuous. A discrete random variable is one that has a finite or countably infinite number of states. Note that these states are not necessarily the integers; they can also just be named states that are not considered to have any numerical value. A continuous random variable is associated with a real value.

3.3 Probability Distributions

A *probability distribution* is a description of how likely a random variable or set of random variables is to take on each of its possible states. The way we describe probability distributions depends on whether the variables are discrete or continuous.

3.3.1 Discrete Variables and Probability Mass Functions

A probability distribution over discrete variables may be described using a *probability mass function* (PMF). We typically denote probability mass functions with a capital P. Often we associate each random variable with a different probability mass function and the reader must infer which probability mass function to use based on the identity of the random variable, rather than the name of the function; $P(\mathrm{x})$ is usually not the same as $P(\mathrm{y})$.

The probability mass function maps from a state of a random variable to the probability of that random variable taking on that state. $P(x)$ denotes the probability that $\mathrm{x} = x$, with a probability of 1 indicating that $\mathrm{x} = x$ is certain and a probability of 0 indicating that $\mathrm{x} = x$ is impossible. Sometimes to disambiguate which PMF to use, we write the name of the random variable explicitly: $P(\mathrm{x} = x)$. Sometimes we define a variable first, then use $\sim$ notation to specify which distribution it follows later: $x \sim P(\mathrm{x})$.

Probability mass functions can act on many variables at the same time. Such a probability distribution over many variables is known as a *joint probability*

distribution. $P(\mathrm{x} = x, \mathrm{y} = y)$ denotes the probability that $\mathrm{x} = x$ and $\mathrm{y} = y$ simultaneously. We may also write $P(x, y)$ for brevity.

To be a probability mass function on a set of random variables x, a function P must meet the following properties:

- The domain of P must be the set of all possible states of x.
- $\forall x \in \mathrm{x}, 0 \leq P(x) \leq 1$. An impossible event has probability 0 and no state can be less probable than that. Likewise, an event that is guaranteed to happen has probability 1, and no state can have a greater chance of occurring.
- $\sum_{x \in \mathrm{x}} P(x) = 1$. ($P$ must guarantee that *some* state occurs.)

For example, consider a single discrete random variable x with k different states. We can place a *uniform distribution* on x—that is, make each of its states equally likely—by setting its probability mass function to

$$P(\mathrm{x} = x_i) = \frac{1}{k}$$

for all i. We can see that this fits the requirements for a probability mass function. The value $\frac{1}{k}$ is positive because k is a positive integer. We also see that $\sum_i P(\mathrm{x} = x_i) = \sum_i \frac{1}{k} = \frac{k}{k} = 1$, so the distribution is properly normalized.

3.3.2 Continuous Variables and Probability Density Functions

When working with continuous random variables, we describe probability distributions using a *probability density function (PDF)* rather than a probability mass function. To be a probability density function, a function p must satisfy the following properties:

- The domain of p must be the set of all possible states of x.
- $\forall x \in \mathrm{x}, p(x) \geq 0$. Note that we do not require $p(x) \leq 1$.
- $\int p(x)dx = 1$.

A probability density function $p(x)$ does not give the probability of a specific state directly, instead the probability of landing inside an infinitesimal region with volume δx is given by $p(x)\delta x$.

We can integrate the density function to find the actual probability mass of a set of points. Specifically, the probability that x lies in some set S is given by the integral of $p(x)$ over that set. In the univariate example, the probability that x lies in the interval $[a, b]$ is given by $\int_{[a,b]} p(x)dx$.

For an example of a probability density function corresponding to a specific probability density over a continuous random variable, consider a uniform distribution on an interval of the real numbers. We can do this with a function $u(x; a, b)$, where a and b are the endpoints of the interval, with $b > a$. (The ";" notation means "parametrized by"; we consider x to be the argument of the function, while a and b are parameters that define the function) To ensure that there is no probability mass outside the interval, we say $u(x; a, b) = 0$ for all $x \notin [a, b]$. Within $[a, b]$, $u(x; a, b) = \frac{1}{b-a}$. We can see that this is nonnegative everywhere. Additionally, it integrates to 1. We often denote that x follows the uniform distribution on $[a, b]$ by writing $x \sim U(a, b)$.

3.4 Marginal Probability

Sometimes we know the probability distribution over a set of variables and we want to know the probability distribution over just a subset of them. The probability distribution over the subset is known as the *marginal probability* distribution.

For example, suppose we have discrete random variables x and y, and we know $P(\mathrm{x}, \mathrm{y})$. We can find $P(\mathrm{x})$ with the *sum rule*:

$$\forall x \in \mathrm{x}, P(\mathrm{x} = x) = \sum_y P(\mathrm{x} = x, \mathrm{y} = y).$$

The name "marginal probability" comes from the process of computing marginal probabilities on paper. When the values of $P(\mathrm{x}, \mathrm{y})$ are written in a grid with different values of x in rows and different values of y in columns, it is natural to sum across a row of the grid, then write $P(x)$ in the margin of the paper just to the right of the row.

For continuous variables, we need to use integration instead of summation:

$$p(x) = \int p(x, y)dy.$$

3.5 Conditional Probability

In many cases, we are interested in the probability of some event, given that some other event has happened. This is called a *conditional probability*. We denote the conditional probability that $\mathrm{y} = y$ given $\mathrm{x} = x$ as $P(\mathrm{y} = y \mid \mathrm{x} = x)$. This conditional probability can be computed with the formula

$$P(\mathrm{y} = y \mid \mathrm{x} = x) = P(\mathrm{y} = y, \mathrm{x} = x)/P(\mathrm{x} = x).$$

Note that this is only defined when $P(\mathrm{x} = x) > 0$. We cannot compute the conditional probability conditioned on an event that never happens.

It is important not to confuse conditional probability with computing what would happen if some action were undertaken. The conditional probability that a person is from Germany given that they speak German is quite high, but if a randomly selected person is taught to speak German, their country of origin does not change. Computing the consequences of an action is called making an *intervention query*. Intervention queries are the domain of *causal modeling*, which we do not explore in this book.

3.6 The Chain Rule of Conditional Probabilities

Any joint probability distribution over many random variables may be decomposed into conditional distributions over only one variable:

$$P(\mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(n)}) = P(\mathrm{x}^{(1)})\Pi_{i=2}^{n} P(\mathrm{x}^{(i)} \mid \mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(i-1)}).$$

This observation is known as the *chain rule* or *product rule* of probability. It follows immediately from the definition of conditional probability. For example, applying the definition twice, we get

$$\begin{aligned} P(\mathrm{a}, \mathrm{b}, \mathrm{c}) &= P(\mathrm{a} \mid \mathrm{b}, \mathrm{c})P(\mathrm{b}, \mathrm{c}) \\ P(\mathrm{b}, \mathrm{c}) &= P(\mathrm{b} \mid \mathrm{c})P(\mathrm{c}) \\ P(\mathrm{a}, \mathrm{b}, \mathrm{c}) &= P(\mathrm{a} \mid \mathrm{b}, \mathrm{c})P(\mathrm{b} \mid \mathrm{c})P(\mathrm{c}). \end{aligned}$$

Note how every statement about probabilities remains true if we add conditions (stuff on the right-hand side of the vertical bar) consistently on all the "P"'s in the statement. We can use this to derive the same thing differently:

$$\begin{aligned} P(\mathrm{a}, \mathrm{b} \mid \mathrm{c}) &= P(\mathrm{a} \mid \mathrm{b}, \mathrm{c})P(\mathrm{b} \mid \mathrm{c}) \\ P(\mathrm{a}, \mathrm{b}, \mathrm{c}) &= P(\mathrm{a}, \mathrm{b} \mid \mathrm{c})P(\mathrm{c}) = P(\mathrm{a} \mid \mathrm{b}, \mathrm{c})P(\mathrm{b} \mid \mathrm{c})P(\mathrm{c}). \end{aligned}$$

3.7 Independence and Conditional Independence

Two random variables x and y are *independent* if their probability distribution can be expressed as a product of two factors, one involving only x and one involving only y:

$$\forall x \in \mathrm{x}, y \in \mathrm{y},\ p(\mathrm{x} = x, \mathrm{y} = y) = p(\mathrm{x} = x)p(\mathrm{y} = y).$$

Two random variables x and y are *conditionally independent* given a random variable z if the conditional probability distribution over x and y factorizes in this way for every value of z:

$$\forall x \in \mathrm{x}, y \in \mathrm{y}, z \in \mathrm{z},\ p(\mathrm{x} = x, \mathrm{y} = y \mid \mathrm{z} = z) = p(\mathrm{x} = x \mid \mathrm{z} = z)p(\mathrm{y} = y \mid \mathrm{z} = z).$$

We can denote independence and conditional independence with compact notation: $\mathrm{x} \perp \mathrm{y}$ means that x and y are independent, while $\mathrm{x} \perp \mathrm{y} \mid \mathrm{z}$ means that x and y are conditionally independent given z.

3.8 Expectation, Variance and Covariance

The *expectation* or *expected value* of some function $f(x)$ with respect to a probability distribution $P(\mathrm{x})$ is the average or mean value that f takes on when x is drawn from P. For discrete variables this can be computed with a summation:

$$\mathbb{E}_{x \sim P}[f(x)] = \sum_x P(x)f(x),$$

while for continuous variables, it is computed with an integral:

$$\mathbb{E}_{x \sim P}[f(x)] = \int p(x)f(x)dx.$$

When the identity of the distribution is clear from the context, we may simply write the name of the random variable that the expectation is over, e.g. $\mathbb{E}_{\mathrm{x}}[f(x)]$. If it is clear which random variable the expectation is over, we may omit the subscript entirely, e.g. $\mathbb{E}[f(x)]$. By default, we can assume that $\mathbb{E}[\cdot]$ averages over the values of all the random variables inside the brackets. Likewise, when there is no ambiguity, we may omit the square brackets.

Expectations are linear, for example, $\mathbb{E}[\alpha f(x) + \beta g(x)] = \alpha\mathbb{E}[f(x)] + \beta\mathbb{E}[g(x)]$, when α and β are fixed (not random and not depending on x).

The *variance* gives a measure of how much the different values of a function are spread apart:

$$\mathrm{Var}(f(x)) = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right].$$

When the variance is low, the values of $f(x)$ cluster near their expected value. The square root of the variance is known as the *standard deviation*.

The *covariance* gives some sense of how much two values are linearly related to each other, as well as the scale of these variables:

$$\mathrm{Cov}(f(x), g(y)) = \mathbb{E}\left[(f(x) - \mathbb{E}\left[f(x)\right])\left(g(y) - \mathbb{E}\left[g(y)\right]\right)\right].$$

High absolute values of the covariance mean that the values change a lot and are both far from their respective means at the same time. If the sign of the covariance is positive, then the values tend to change in the same direction, while if it is negative, they tend to change in opposite directions. Other measures such as *correlation* normalize the contribution of each variable in order to measure only how much the variables are related, rather than also being affected by the scale of the separate variables.

The notions of covariance and dependence are conceptually related, but are in fact distinct concepts. Two random variables that have non-zero covariance are dependent. However, they may have zero covariance without being independent. For example, suppose we first generate x, then generate $s \in \{-1, 1\}$ with each state having probability 0.5, then generate y as $s(x - \mathbb{E}[x])$. Clearly, x and y are not independent, because y only has two possible values given x. However, $\text{Cov}(x, y) = 0$.

The *covariance matrix* of a random vector $\boldsymbol{x} \in \mathbb{R}^n$ is an $n \times n$ matrix, such that

$$\text{Cov}(\mathbf{x})_{i,j} = \text{Cov}(\mathrm{x}_i, \mathrm{x}_j).$$

Note that the diagonal elements give $\text{Cov}(\mathrm{x}_i, \mathrm{x}_i) = \text{Var}(\mathrm{x}_i)$.

3.9 Information Theory

Information theory is a branch of applied mathematics that revolves around quantifying how much information is present in a signal. It was originally invented to study sending messages from discrete alphabets over a noisy channel, such as communication via radio transmission. In this context, information theory tells how to design optimal codes and calculate the expected length of messages sampled from specific probability distributions using various encoding schemes. In the context of machine learning, we can also apply information theory to continuous variables where some of these message length interpretations do not apply. This field is fundamental to many areas of electrical engineering and computer science. In this textbook, we mostly use a few key ideas from information theory to characterize probability distributions or quantify similarity between probability distributions. For more detail on information theory, see (Cover and Thomas, 2006; MacKay, 2003).

The basic intuition behind information theory is that learning that an unlikely event has occurred is more informative than learning that a likely event has occurred. A message saying "the sun rose this morning" is so uninformative as to be unnecessary to send, but a message saying "there was a solar eclipse this morning" is very informative.

We would like to quantify information in a way that formalizes this intuition. Specifically,

- Likely events should have low information content, and in the extreme case, events that are guaranteed to happen should have no information content whatsoever.
- Less likely events should have higher information content.
- Independent events should have additive information. For example, finding out that a tossed coin has come up as heads twice should convey twice as much information as finding out that a tossed coin has come up as heads once.

In order to satisfy all three of these properties, we define the *self-information* of an event $\mathrm{x} = x$ to be

$$I(x) = -\log P(x). \tag{3.1}$$

In this book, we always use log to mean the natural logarithm, with base e. Our definition of $I(x)$ is therefore written in units of *nats*. One nat is the amount of information gained by observing an event of probability $\frac{1}{e}$. Other texts use base-2 logarithms and units called *bits* or *shannons*; information measured in bits is just a rescaling of information measured in nats.

When x is continuous, we use the same definition of information by analogy, but some of the properties from the discrete case are lost. For example, an event with unit density still has zero information, despite not being an event that is guaranteed to occur.

Self-information deals only with a single outcome. We can quantify the amount of uncertainty in an entire probability distribution using the *Shannon entropy*[1]:

$$H(\mathrm{x}) = \mathbb{E}_{\mathrm{x}\sim P}[I(x)] = -\mathbb{E}_{\mathrm{x}\sim P}[\log P(x)]. \tag{3.2}$$

also denoted $H(P)$. In other words, the Shannon entropy of a distribution is the expected amount of information in an event drawn from that distribution. It actually gives a lower bound on the number of bits (if the logarithm is base 2, otherwise the units are different) needed in average to encode symbols drawn from a distribution P. Distributions that are nearly deterministic (where the outcome is nearly certain) have low entropy; distributions that are closer to uniform have high entropy. See Fig. 3.1 for a demonstration. When x is continous, the Shannon entropy is known as the *differential entropy*.

[1]Shannon entropy is named for Claude Shannon, the father of information theory (Shannon, 1948, 1949). For an interesting biographical account of Shannon and some of his contemporaries, see *Fortune's Formula* by William Poundstone (Poundstone, 2005).

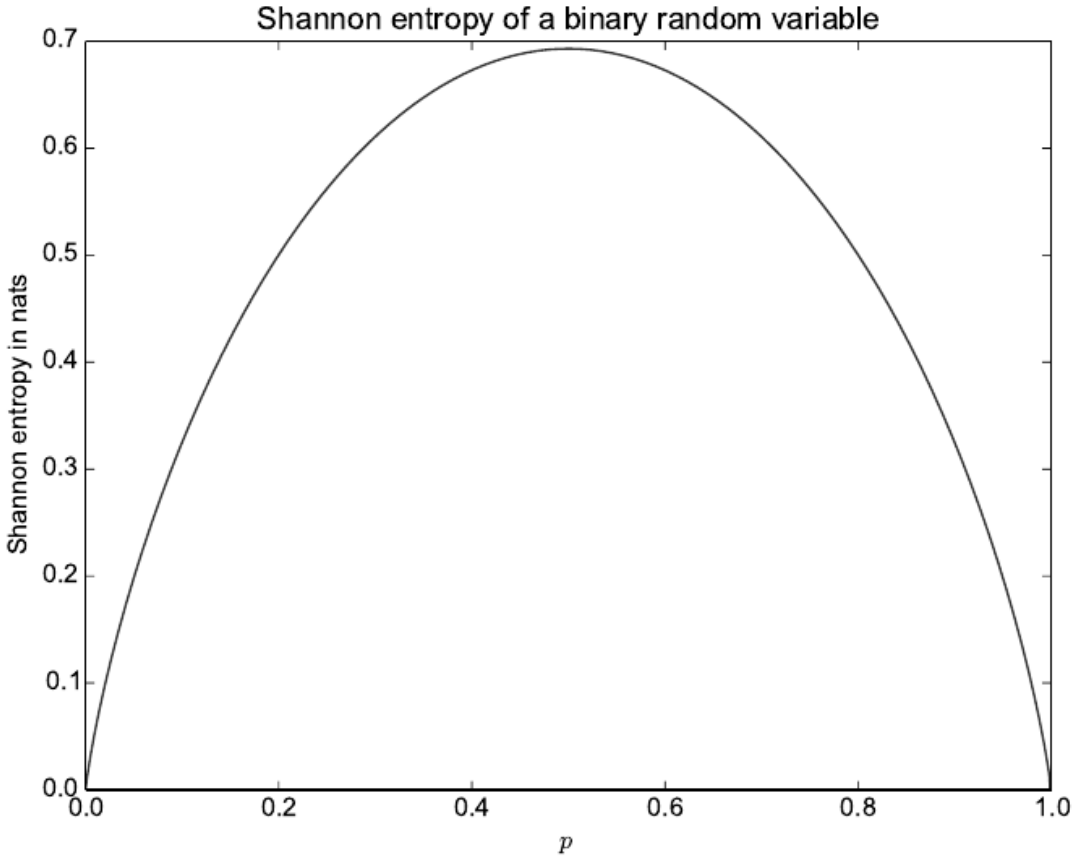

Figure 3.1: This plot shows how distributions that are closer to deterministic have low Shannon entropy while distributions that are close to uniform have high Shannon entropy. On the horizontal axis, we plot p, the probability of a binary random variable being equal to 1. When p is near 0, the distribution is nearly deterministic, because the random variable is nearly always 0. When p is near 1, the distribution is nearly deterministic, because the random variable is nearly always 1. When $p = 0.5$, the entropy is maximal, because the distribution is uniform over the two outcomes.

If we have two separate probability distributions $P(\mathrm{x})$ and $Q(\mathrm{x})$ over the same random variable x, we can measure how different these two distributions are using the *Kullback-Leibler (KL) divergence*:

$$D_{\mathrm{KL}}(P\|Q) = \mathbb{E}_{x\sim P}\left[\log \frac{P(x)}{Q(x)}\right] = \mathbb{E}_{x\sim P}\left[\log P(x) - \log Q(x)\right]. \tag{3.3}$$

In the case of discrete variables, it is the extra amount of information (measured in bits if we use the base 2 logarithm, but in machine learning we usually use nats and the natural logarithm) needed to send a message containing symbols drawn from probability distribution P, when we use a code that was designed to minimize the length of messages drawn from probability distribution Q.

The KL divergence has many useful properties, most notably that it is non-negative. The KL divergence is 0 if and only if P and Q are the same distribution in the case of discrete variables, or equal "almost everywhere" in the case of continuous variables (see section 3.13 for details). Because the KL divergence is non-negative and measures the difference between two distributions, it is often conceptualized as measuring some sort of distance between these distributions. However, it is not a true distance measure because it is not symmetric, i.e. $D_{\mathrm{KL}}(P\|Q) \neq D_{\mathrm{KL}}(Q\|P)$ for some P and Q.

A quantity that is closely related to the KL divergence is the cross entropy $H(P,Q) = H(P) + D_{\mathrm{KL}}(P\|Q)$, which is similar to the KL divergence but lacking the term on the left:

$$H(P,Q) = \mathbb{E}_{x\sim P}\log Q(x).$$

When computing many of these quantities, it is common to encounter expressions of the form $0\log 0$. By convention, in the context of information theory, we treat these expressions as $\lim_{x\to 0} x\log x = 0$.

3.10 Common Probability Distributions

Several simple probability distributions are useful in many contexts in machine learning.

3.10.1 Bernoulli Distribution

The *Bernoulli* distribution is a distribution over a single binary random variable. It is controlled by a single parameter $\phi \in [0,1]$, which gives the probability of the random variable being equal to 1. It has the following properties:

$$P(\mathrm{x} = 1) = \phi$$

$$P(\mathrm{x} = 0) = 1 - \phi$$

$$P(\mathrm{x} = x) = \phi^{x}(1 - \phi)^{1-x}$$

$$\mathbb{E}_{\mathrm{x}}[\mathrm{x}] = \phi$$

$$\mathrm{Var}_{\mathrm{x}}(\mathrm{x}) = \phi(1 - \phi)$$

$$H(\mathrm{x}) = (\phi - 1)\log(1 - \phi) - \phi \log \phi.$$

3.10.2 Multinoulli Distribution

The *multinoulli* or *categorical* distribution is a distribution over a single discrete variable with k different states, where k is finite[2] . The multinoulli distribution is parametrized by a vector $\boldsymbol{p} \in [0, 1]^{k-1}$, where p_i gives the probability of the i-th state. The final, k-th state's probability is given by $1 - \mathbf{1}^{\top}\boldsymbol{p}$. Note that we must constrain $\mathbf{1}^{\top}\boldsymbol{p} \leq 1$. Multinoulli distributions are often used to refer to distributions over categories of objects, so we do not usually assume that state 1 has numerical value 1, etc. For this reason, we do not usually need to compute the expectation or variance of multinoulli-distributed random variables.

The Bernoulli and multinoulli distributions are sufficient to describe any distribution over their domain. This is because they model discrete variables for which it is feasible to simply enumerate all of the states. When dealing with continuous variables, there are uncountably many states, so any distribution described by a small number of parameters must impose strict limits on the distribution.

3.10.3 Gaussian Distribution

The most commonly used distribution over real numbers is the *normal distribution*, also known as the *Gaussian distribution*:

$$\mathcal{N}(x \mid \mu, \sigma^2) = \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^{2}}(x - \mu)^2\right).$$

See Fig. 3.2 for a schematic.

The two parameters $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}^{+}$ control the normal distribution. μ gives the coordinate of the central peak. This is also the mean of the distribution, i.e. $\mathbb{E}[\mathrm{x}] = \mu$. The standard deviation of the distribution is given by σ, i.e. $\mathrm{Var}(\mathrm{x}) = \sigma^2$.

[2] "Multinoulli" is a recently coined term. The multinoulli distribution is a special case of the *multinomial* distribution. A multinomial distribution is the distribution over vectors in $\{0, \ldots, n\}^{k}$ representing how many times each of the k categories is visited when n samples are drawn from a multinoulli distribution. Many texts use the term "multinomial" to refer to multinoulli distributions without clarifying that they refer only to the $n = 1$ case.

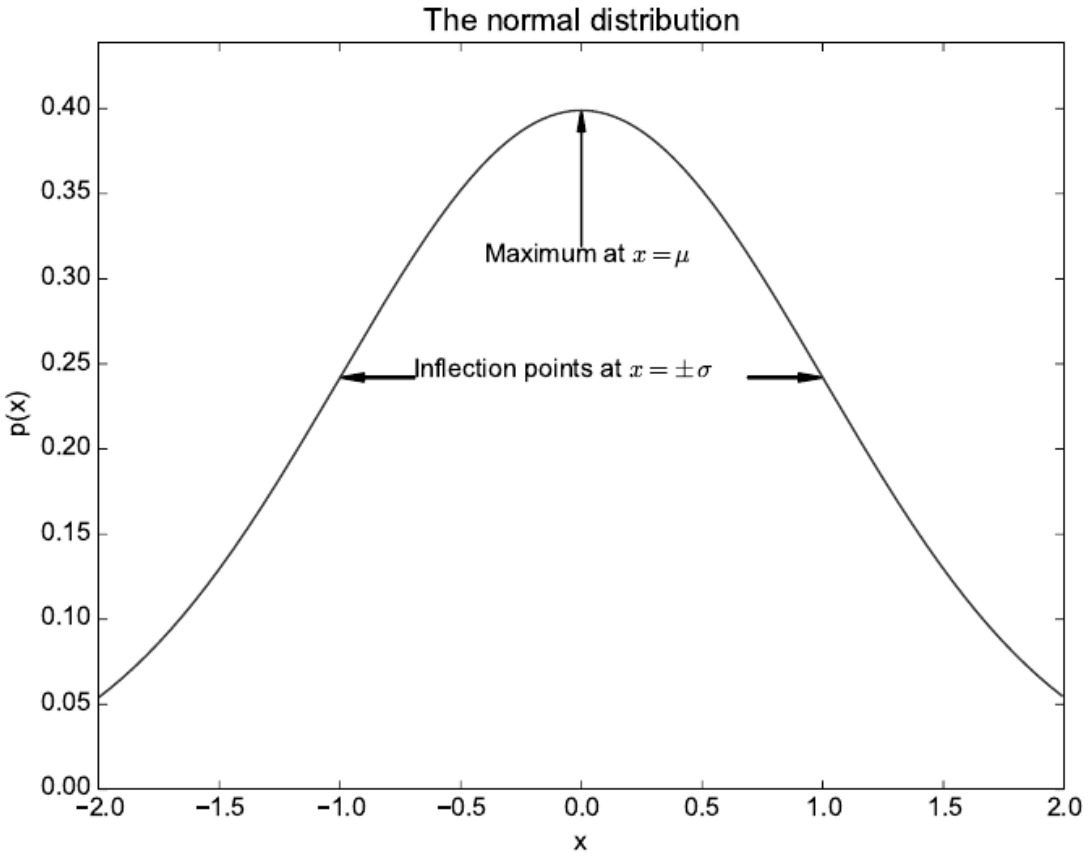

Figure 3.2: *The normal distribution*: The normal distribution $\mathcal{N}(x \mid \mu, \sigma^2)$ exhibits a classic "bell curve" shape, with the x coordinate of its central peak given by μ, and the width of its peak controlled by σ. In this example, we depict the *standard normal distribution*, with $\mu = 0$ and $\sigma = 1$.

Note that when we evaluate the PDF, we need to square and invert σ. When we need to frequently evaluate the PDF with different parameter values, a more efficient way of parametrizing the distribution is to use a parameter $\beta \in \mathbb{R}^+$ to control the *precision* or inverse variance of the distribution:

$$\mathcal{N}(x \mid \mu, \beta^{-1}) = \sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{1}{2}\beta(x-\mu)^2\right).$$

Normal distributions are a sensible choice for many applications. In the absence of prior knowledge about what form a distribution over the real numbers should take, the normal distribution is a good default choice for two major reasons.

First, many distributions we wish to model are truly close to being normal distributions. The *central limit theorem* shows that the sum of many independent random variables is approximately normally distributed. This means that in practice, many complicated systems can be modeled successfully as normally distributed noise, even if the system can be decomposed into parts with more structured behavior.

Second, the normal distribution in some sense makes the fewest assumptions of any distribution over the reals, so choosing to use it inserts the least amount of prior knowledge into a model. Out of all distributions with the same variance, the normal distribution has the highest entropy. It is not possible to place a uniform distribution on all of $\mathbb{R}$. The closest we can come to doing so is to use a normal distribution with high variance.

The normal distribution generalizes to $\mathbb{R}^n$, in which case it is known as the *multivariate normal distribution*. It may be parametrized with a positive definite symmetric matrix $\boldsymbol{\Sigma}$:

$$\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sqrt{\frac{1}{(2\pi)^n \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right).$$

The parameter $\boldsymbol{\mu}$ still gives the mean of the distribution, though now it is vector-valued. The parameter $\boldsymbol{\Sigma}$ gives the covariance matrix of the distribution. As in the univariate case, the covariance is not necessarily the most computationally efficient way to parametrize the distribution, since we need to invert $\boldsymbol{\Sigma}$ to evaluate the PDF. We can instead use a *precision matrix* $\boldsymbol{\beta}$:

$$\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\beta}^{-1}) = \sqrt{\frac{\det(\boldsymbol{\beta})}{(2\pi)^n}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\beta}(\boldsymbol{x}-\boldsymbol{\mu})\right).$$

3.10.4 Exponential and Laplace Distributions

In the context of deep learning, we often want to have a probability distribution with a sharp point at $x = 0$. To accomplish this, we can use the *exponential distribution*:

$$p(x; \lambda) = \lambda \mathbf{1}_{x \geq 0} \exp(-\lambda x).$$

The exponential distribution uses the indicator function $\mathbf{1}_{x \geq 0}$ to assign probability zero to all negative values of x.

A closely related probability distribution that allows us to place a sharp peak of probability mass at an arbitrary point μ is the *Laplace distribution*

$$p(x; \mu, \lambda) = \frac{1}{2\lambda} \exp\left(-\frac{|x - \mu|}{\lambda} \right).$$

The Laplace distribution is sometimes also called the *double exponential distribution* because it can be written as the sum of two exponential distributions, with one flipped and both shifted to reposition the mode to $x = \mu$. The term "Laplace distribution" is preferred because "double exponential distribution" also has other meanings.

3.10.5 Dirac Distribution

In some cases, we wish to specify that all of the mass in a probability distribution clusters around a single point. This can be accomplished by defining a PDF using the Dirac delta function, $\delta(x)$:

$$p(x) = \delta(x - \mu).$$

The Dirac delta function is defined such that it is zero-valued everywhere but 0, yet integrates to 1. By defining $p(x)$ to be δ shifted by $-\mu$ we obtain an infinitely narrow and infinitely high peak of probability mass where $x = \mu$.

A common use of the Dirac delta distribution is as a component of the so-called *empirical distribution*,

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} \delta(x - x_i) \tag{3.4}$$

which puts probability mass $\frac{1}{n}$ on each of the n points $x_1, \ldots x_n$ forming a given data set or collection of samples. The Dirac delta distribution is only necessary to define the empirical distribution over continuous variables. For discrete variables, the situation is simpler: an empirical distribution can be conceptualized as a multinoulli distribution, with a probability associated to each possible input value that is simply equal to the *empirical frequency* of that value in the training set.

We can view the empirical distribution formed from a dataset of training examples as specifying the distribution that we sample from when we train a model on this dataset. Another important perspective on the empirical distribution is that it is the probability density that maximizes the likelihood of the training data (see Section 5.6). Many machine learning algorithms can be configured to have arbitrarily high capacity. If given enough capacity, these algorithms will simply learn the empirical distribution. This is a bad outcome because the model does not generalize at all and assigns infinitesimal probability to any point in space that did not occur in the training set. A central problem in machine learning is studying how to limit the capacity of a model in a way that prevents it from simply learning the empirical distribution while also allowing it to learn complicated functions.

The empirical distribution is a particular form of *mixture*, discussed next.

3.10.6 Mixtures of Distributions

It is also common to define probability distributions by composing other simpler probability distributions. One common way of combining distributions is to construct a *mixture distribution*. A mixture distribution is made up of several component distributions. On each trial, the choice of which component distribution generates the sample is determined by sampling a component identity from a multinoulli distribution:

$$P(\mathrm{x}) = \sum_i P(\mathrm{c} = i) P(\mathrm{x} \mid \mathrm{c} = i)$$

where $P(\mathrm{c})$ is the multinoulli distribution over component identities. In chapter 13, we explore the art of building complex probability distributions from simple ones in more detail. Note that we can think of the variable c as a non-observed (or latent) random variable that is related to x through their joint distribution $P(\mathrm{x}, \mathrm{c}) = P(\mathrm{x} \mid \mathrm{c}) P(\mathrm{c})$. Latent variables are discussed further in Section 13.4.2.

A very powerful and common type of mixture model is the *Gaussian mixture* model, in which the components $P(\mathrm{x} \mid \mathrm{c} = i)$ are Gaussians, each with its mean μ_i and covariance Σ_i. Some mixtures can have more constraints, for example, the covariances could be shared across components, i.e., $\Sigma_i = \Sigma_j = \Sigma$, or the covariance matrices could be constrained to be diagonal or simply equal to a scalar times the identity. A Gaussian mixture model is a *universal approximator* of densities, in the sense that any smooth density can be approximated to a particular precision by a Gaussian mixture model with enough components. Gaussian mixture models have been used in many settings, and are particularly well known for their use as acoustic models in speech recognition (Bahl *et al.*, 1987).

3.11 Useful Properties of Common Functions

Certain functions arise often while working with probability distributions, especially the probability distributions used in deep learning models.

One of these functions is the *logistic sigmoid*:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}.$$

The logistic sigmoid is commonly used to produce the ϕ parameter of a Bernoulli distribution because its range is $(0, 1)$, which lies within the valid range of values for the ϕ parameter. See Fig. 3.3 for a graph of the sigmoid function.

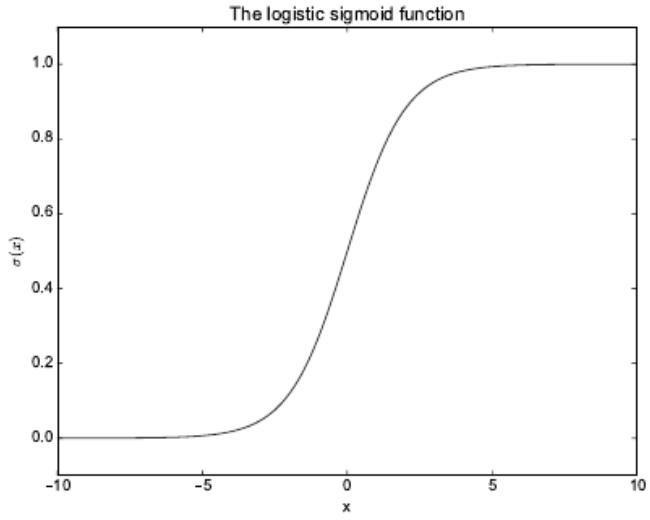

Figure 3.3: The logistic sigmoid function.

Another commonly encountered function is the *softplus* function (Dugas *et al.*, 2001):

$$\zeta(x) = \log\left(1 + \exp(x)\right).$$

The softplus function can be useful for producing the β or σ parameter of a normal distribution because its range is $\mathbb{R}^+$. It also arises commonly when manipulating expressions involving sigmoids, as it is the primitive of the sigmoid, i.e., the integral from $-\infty$ to x of the sigmoid. The name of the softplus function comes from the fact that it is a smoothed or "softened" version of

$$x^+ = \max(0, x).$$

See Fig. 3.4 for a graph of the softplus function.

The following properties are all useful enough that you may wish to memorize them:

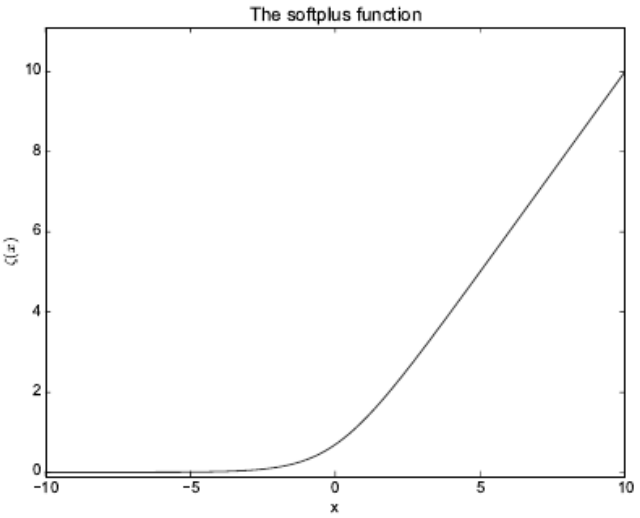

Figure 3.4: The softplus function.

$$\sigma(x) = \frac{\exp(x)}{\exp(x) + \exp(0)}$$

$$\frac{d}{dx}\sigma(x) = \sigma(x)(1 - \sigma(x))$$

$$1 - \sigma(x) = \sigma(-x)$$

$$\log \sigma(x) = -\zeta(-x)$$

$$\frac{d}{dx}\zeta(x) = \sigma(x)$$

$$\forall x \in (0, 1),\ \sigma^{-1}(x) = \log\left(\frac{x}{1 - x}\right)$$

$$\forall x > 0,\ \zeta^{-1}(x) = \log\left(\exp(x) - 1\right)$$

$$\zeta(x) - \zeta(-x) = x$$

The function $\sigma^{-1}(x)$ is called the *logit* in statistics, but this term is more rarely used in machine learning. The final property provides extra justification for the name "softplus", since $x^+ - x^- = x$.

3.12 Bayes' Rule

We often find ourselves in a situation where we know $P(\mathrm{y} \mid \mathrm{x})$ and need to know $P(\mathrm{x} \mid \mathrm{y})$. Fortunately, if we also know $P(\mathrm{x})$, we can compute the desired quantity

using *Bayes' rule*:

$$P(\mathrm{x} \mid \mathrm{y}) = \frac{P(\mathrm{x})P(\mathrm{y} \mid \mathrm{x})}{P(\mathrm{y})}.$$

Note that while $P(\mathrm{y})$ appears in the formula, it is usually feasible to compute $P(\mathrm{y}) = \sum_x P(\mathrm{y} \mid x)P(x)$, so we do not need to begin with knowledge of $P(\mathrm{y})$.

Bayes' rule is straightforward to derive from the definition of conditional probability, but it is useful to know the name of this formula since many texts refer to it by name. It is named after the Reverend Thomas Bayes, who first discovered a special case of the formula. The general version presented here was independently discovered by Pierre-Simon Laplace.

3.13 Technical Details of Continuous Variables

A proper formal understanding of continuous random variables and probability density functions requires developing probability theory in terms of a branch of mathematics known as *measure theory*. Measure theory is beyond the scope of this textbook, but we can briefly sketch some of the issues that measure theory is employed to resolve.

In section 3.3.2, we saw that the probability of a continuous vector-valued $\mathbf{x}$ lying in some set S is given by the integral of $p(\boldsymbol{x})$ over the set S. Some choices of set S can produce paradoxes. For example, it is possible to construct two sets S_1 and S_2 such that $P(S_1) + P(S_2) > 1$ but $S_1 \cap S_2 = \emptyset$. These sets are generally constructed making very heavy use of the infinite precision of real numbers, for example by making fractal-shaped sets or sets that are defined by transforming the set of rational numbers[3]. One of the key contributions of measure theory is to provide a characterization of the set of sets that we can compute the probability of without encountering paradoxes. In this book, we only integrate over sets with relatively simple descriptions, so this aspect of measure theory never becomes a relevant concern.

For our purposes, measure theory is more useful for describing theorems that apply to most points in $\mathbb{R}^n$ but do not apply to some corner cases. Measure theory provides a rigorous way of describing that a set of points is negligibly small. Such a set is said to have "*measure zero*". We do not formally define this concept in this textbook. However, it is useful to understand the intuition that a set of measure zero occupies no volume in the space we are measuring. For example, within $\mathbb{R}^2$, a line has measure zero, while a filled polygon has positive measure. Likewise, an individual point has measure zero. Any union of countably many sets that each have measure zero also has measure zero (so the set of all the rational numbers has measure zero, for instance).

[3]The Banach-Tarski theorem provides a fun example of such sets.

Another useful term from measure theory is "*almost everywhere*". A property that holds almost everywhere holds throughout all of space except for on a set of measure zero. Because the exceptions occupy a negligible amount of space, they can be safely ignored for many applications. Some important results in probability theory hold for all discrete values but only hold "almost everywhere" for continuous values.

One other detail we must be aware of relates to handling random variables that are deterministic functions of one another. Suppose we have two random variables, $\boldsymbol{x}$ and $\boldsymbol{y}$, such that $\boldsymbol{y} = g(\boldsymbol{x})$. You might think that $p_y(\boldsymbol{y}) = p_x(g^{-1}(\boldsymbol{y}))$. This is actually not the case.

Suppose $\boldsymbol{y} = \frac{\boldsymbol{x}}{2}$ and $\boldsymbol{x} \sim U(0, 1)$. If we use the rule $p_y(\boldsymbol{y}) = p_x(2\boldsymbol{y})$ then p_y will be 0 everywhere except the interval $[0, \frac{1}{2}]$, and it will be 1 on this interval. This means

$$\int p_y(\boldsymbol{y}) d\boldsymbol{y} = \frac{1}{2}$$

which violates the definition of a probability distribution.

This common mistake is wrong because it fails to account for the distortion of space introduced by the function $g(\boldsymbol{x})$. Recall that the probability of $\boldsymbol{x}$ lying in an infinitesimally small region with volume $\delta\boldsymbol{x}$ is given by $p(\boldsymbol{x})\delta\boldsymbol{x}$. Since g can expand or contract space, the infinitesimal volume surrounding $\boldsymbol{x}$ in $\boldsymbol{x}$ space may have different volume in $\boldsymbol{y}$ space. To correct the problem, we need to preserve the property

$$|p_y(g(\boldsymbol{x}))dy| = |p_x(\boldsymbol{x})dx|.$$

Solving from this, we obtain

$$p_y(\boldsymbol{y}) = p_x(g^{-1}(\boldsymbol{y}))|\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{y}}|$$

or equivalently

$$p_x(\boldsymbol{x}) = p_y(g(\boldsymbol{x}))|\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}|. \tag{3.5}$$

In higher dimensions, the absolute value of the derivative generalizes to the determinant of the *Jacobian matrix* — the matrix with $J_{i,j} = \frac{\partial x_i}{\partial y_j}$.

3.14 Structured Probabilistic Models

Machine learning algorithms often involve probability distributions over a very large number of random variables. Often, these probability distributions involve direct interactions between relatively few variables. Using a single function to describe the entire joint probability distribution can be very inefficient (both computationally and statistically).

Instead of using a single function to represent a probability distribution, we can split a probability distribution into many factors that we multiply together. For example, suppose we have three random variables, a, b, and c. Suppose that a influences the value of b and b influences the value of c, but that a and c are independent given b. We can represent the probability distribution over all three variables as a product of probability distributions over two variables:

$$p(\mathrm{a}, \mathrm{b}, \mathrm{c}) = p(\mathrm{a})p(\mathrm{b} \mid \mathrm{a})p(\mathrm{c} \mid \mathrm{a}).$$

These factorizations can greatly reduce the number of parameters needed to describe the distribution. Each factor uses a number of parameters that is exponential in the number of variables in the factor. This means that we can greatly reduce the cost of representing a distribution if we are able to find a factorization into distributions over fewer variables.

We can describe these kinds of factorizations using graphs. Here we use the word "graph" in the sense of graph theory, i.e. a set of vertices that may be connected to each other with edges. When we represent the factorization of a probability distribution with a graph, we call it a *structured probabilistic model* or *graphical model*.

There are two main kinds of structured probabilistic models: directed and undirected. Both kinds of graphical models use a graph in which each node in the graph corresponds to a random variable, and an edge connecting two random variables means that the probability distribution is able to represent direct interactions between those two random variables.

Directed models use graphs with directed edges, and they represent factorizations into conditional probability distributions, as in the example above. Specifically, a directed model contains one factor for every random variable x_i in the distribution, and that factor consists of the conditional distribution over x_i given the parents of x_i:

$$p(\mathbf{x}) = \prod_i p\left(\mathrm{x}_i \mid Pa_{\mathcal{G}}\left(\mathrm{x}_i\right)\right).$$

See Fig. 3.5 for an example of a directed graph and the factorization of probability distributions it represents.

Undirected models use graphs with undirected edges, and they represent factorizations into a set of functions; unlike in the directed case, these functions are usually not probability distributions of any kind. Any set of nodes that are all connected to each other in $\mathcal{G}$ is called a clique. Each clique $\mathcal{C}^{(i)}$ in an undirected model is associated with a factor $\phi^{(i)}(\mathcal{C}^{(i)})$. These factors are just functions, not probability distributions. The output of each factor must be non-negative, but there is no constraint that the factor must sum or integrate to 1 like a probability distribution.

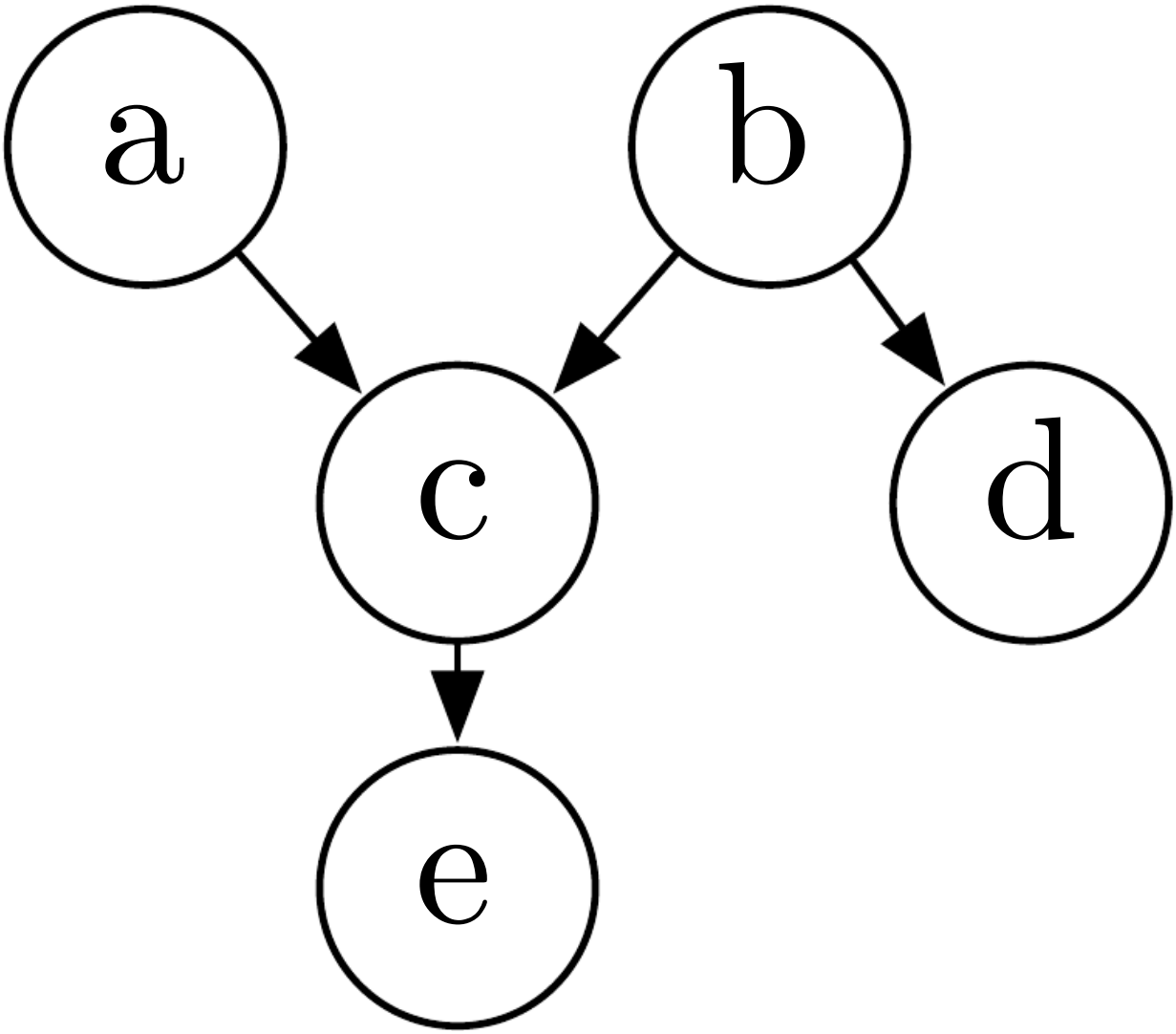

Figure 3.5: A directed graphical model over random variables a, b, c, d and e. This graph corresponds to probability distributions that can be factored as $p(\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}, \mathrm{e}) = p(\mathrm{a})p(\mathrm{b})p(\mathrm{c} \mid \mathrm{a}, \mathrm{b})p(\mathrm{d} \mid \mathrm{b})p(\mathrm{e} \mid \mathrm{c})$. This graph allows us to quickly see some properties of the distribution. For example, a and c interact directly, but a and e interact only indirectly via c.

The probability of a configuration of random variables is *proportional* to the product of all of these factors—assignments that result in larger factor values are more likely. Of course, there is no guarantee that this product will sum to 1. We therefore divide by a normalizing constant Z, defined to be the sum or integral over all states of the product of the ϕ functions, in order to obtain a normalized probability distribution:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_i \phi^{(i)} \left(\mathcal{C}^{(i)} \right) .$$

See Fig. 3.6 for an example of an undirected graph and the factorization of probability distributions it represents.

Keep in mind that these graphical representations of factorizations are a language for describing probability distributions. They are not mutually exclusive families of probability distributions. Being directed or undirected is not a property of a probability distribution; it is a property of a particular *description* of a probability distribution, but any probability distribution may be described in both ways.

Throughout part I and part II of this book, we will use structured probabilistic models merely as a language to describe which direct probabilistic relationships different machine learning algorithms choose to represent. No further understanding of structured probabilistic models is needed until the discussion of research topics, in part III, where we will explore structured probabilistic models in much greater detail.

3.15 Example: Naive Bayes

We now know enough probability theory that we can perform some simple applications with a probabilistic model. In this example, we will show how to infer the probability that a patient has the flu using a simple probabilistic model. For now, we will assume that we just know the correct model somehow. Later chapters will cover the concepts needed to learn the model from data.

The *Naive Bayes* model is a simple probabilistic model that is often used to recognize patterns. The model consists of one random variable c representing a category and a set of random variables $\mathbb{F} = \{\mathrm{f}^{(1)}, \ldots, \mathrm{f}^{(n)}\}$ representing features of objects in each category. In this example, we'll use Naive Bayes to diagnose patients as having the flu or not. The random variable c can thus have two values: c_0 representing the category of patients who do not have the flu, and c_1 representing the category of patients who do. Suppose $\mathrm{f}^{(1)}$ is the random variable representing whether the patient has a sore throat, with $f_0^{(1)}$ representing no sore

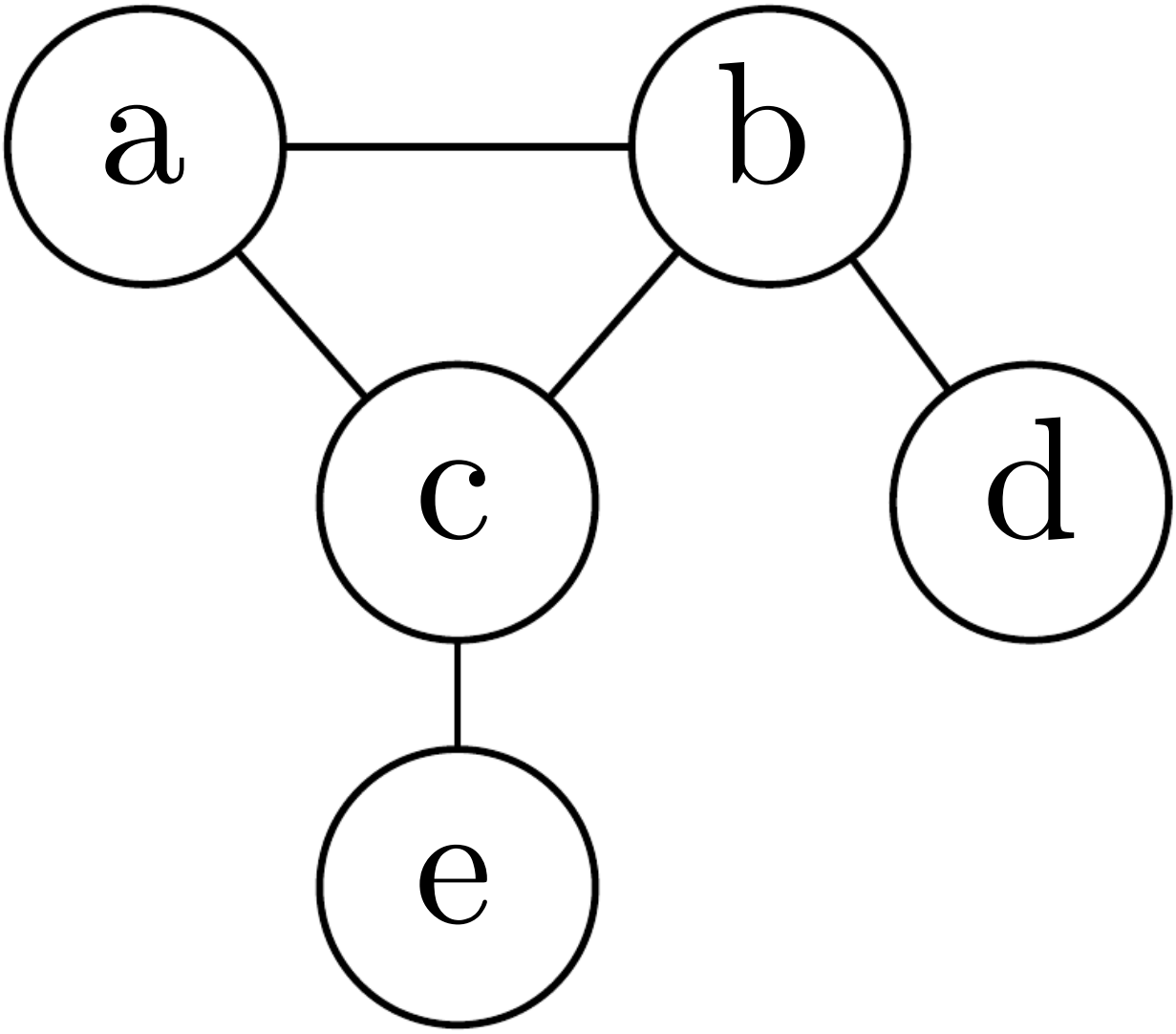

Figure 3.6: An undirected graphical model over random variables a, b, c, d and e. This graph corresponds to probability distributions that can be factored as $p(\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}, \mathrm{e}) = \frac{1}{Z}\phi^{(1)}(\mathrm{a}, \mathrm{b}, \mathrm{c})\phi^{(2)}(\mathrm{b}, \mathrm{d})\phi^{(3)}(\mathrm{c}, \mathrm{e})$. This graph allows us to quickly see some properties of the distribution. For example, a and c interact directly, but a and e interact only indirectly via c.

throat, and $f_1^{(1)}$ representing a sore throat. Suppose $\mathrm{f}^{(2)} \in \mathbb{R}$ is the patient's temperature in degrees Celsius.

When using the Naive Bayes model, we assume that all of the features are independent from each other given the category:

$$P(\mathrm{c}, \mathrm{f}^{(1)}, \ldots, \mathrm{f}^{(n)}) = P(\mathrm{c}) \prod_i P(\mathrm{f}^{(i)} \mid \mathrm{c}).$$

See Fig. 3.7 for a directed graphical model that expresses these conditional independence assumptions. These assumptions are very strong and unlikely to be true in naturally occuring situations, hence the name "naive". Surprisingly, Naive Bayes often produces good predictions in practice (even though the assumptions do not hold precisely), and is a good baseline model to start with when tackling a new problem.

Beyond these conditional independence assumptions, the Naive Bayes framework does not specify anything about the probability distribution. The specific choice of distributions is left up to the designer. In our flu example, let's make $P(\mathrm{c})$ a Bernoulli distribution, with $P(\mathrm{c} = c_1) = \phi^{(\mathrm{c})}$. We can also make $P(\mathrm{f}^{(1)} \mid \mathrm{c})$ a Bernoulli distribution, with

$$P(\mathrm{f}^{(1)} = f_1^{(1)} \mid \mathrm{c} = c) = \phi_c^{\mathrm{f}}.$$

In other words, the Bernoulli parameter changes depending on the value of c. Finally, we need to choose the distribution over $\mathrm{f}^{(2)}$. Since $\mathrm{f}^{(2)}$ is real-valued, a normal distribution is a good choice. Because $\mathrm{f}^{(2)}$ is a temperature, there are hard limits to the values it can take on—it cannot go below $0K$, for example. Fortunately, these values are so far from the values measured in human patients that we can safely ignore these hard limits. Values outside the hard limits will receive extremely low probability under the normal distribution so long as the mean and variance are set correctly. As with $\mathrm{f}^{(1)}$, we need to use different parameters for different values of c, to represent that patients with the flu have different temperatures than patients without it:

$$\mathrm{f}^{(2)} \sim N(\mathrm{f}^{(2)} \mid \mu_c, \sigma_c^2).$$

Now we are ready to determine how likely a patient is to have the flu. To do this, we want to compute $P(\mathrm{c} \mid \mathbb{F})$, but we know $P(\mathrm{c})$ and $P(\mathbb{F} \mid \mathrm{c})$. This suggests that we should use Bayes' rule to determine the desired distribution. The word "Bayes" in the name "Naive Bayes" comes from this frequent use of Bayes' rule in conjunction with the model. We begin by applying Bayes' rule:

$$P(\mathrm{c} \mid \mathbb{F}) = \frac{P(\mathrm{c})P(\mathbb{F} \mid \mathrm{c})}{P(\mathbb{F})}. \tag{3.6}$$

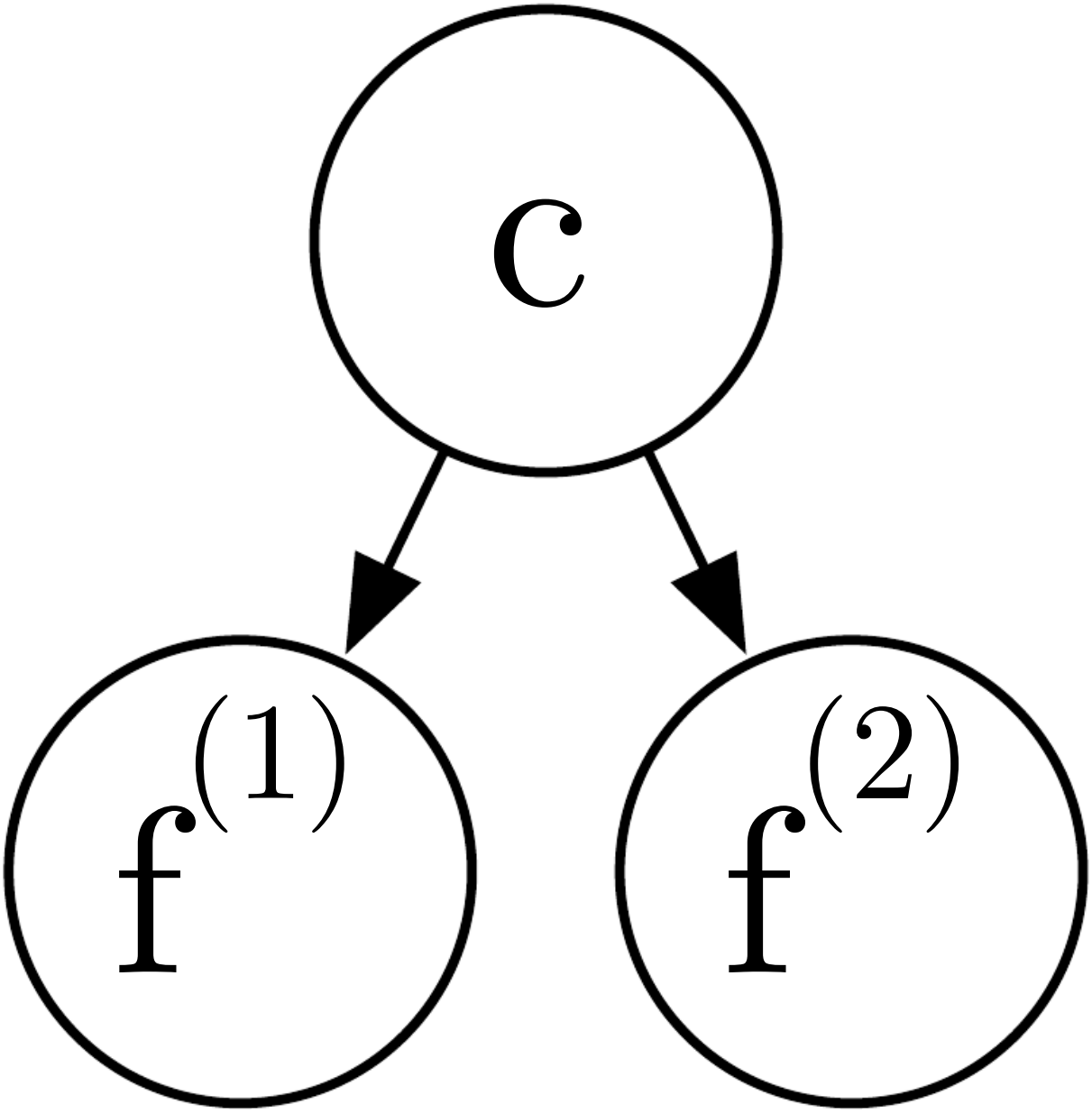

Figure 3.7: A directed graphical model depicting the conditional independence assumptions used by the Naive Bayes model.

We do not know $P(\mathbb{F})$. Fortunately, it is easy to compute:

$$P(\mathbb{F}) = \sum_{c \in \mathrm{c}} P(\mathrm{c} = c, \mathbb{F}) \text{ (by the sum rule)}$$

$$= \sum_{c \in \mathrm{c}} P(\mathrm{c} = c) P(\mathbb{F} \mid \mathrm{c} = c) \text{ (by the chain rule).}$$

Substituting this result back into equation 3.6, we obtain

$$P(\mathrm{c} \mid \mathbb{F}) = \frac{P(\mathrm{c}) P(\mathbb{F} \mid \mathrm{c})}{\sum_{c \in \mathrm{c}} P(\mathrm{c} = c) P(\mathbb{F} \mid \mathrm{c} = c)}$$

$$= \frac{P(\mathrm{c}) \Pi_i P(\mathrm{f}^{(i)} \mid \mathrm{c})}{\sum_{c \in \mathrm{c}} P(\mathrm{c} = c) \Pi_i P(\mathrm{f}^{(i)} \mid \mathrm{c} = c)}$$

by the Naive Bayes assumptions. This is as far as we can simplify the expression for a general Naive Bayes model.

We can simplify the expression further by substituting in the definitions of the particular probability distributions we have defined for our flu diagnosis example:

$$P(\mathrm{c} = c \mid \mathrm{f}^{(1)} = f_1, \mathrm{f}^{(2)} = f_2) = \frac{g(c)}{\sum_{c' \in \mathrm{c}} g(c')}$$

where

$$g(c) = P(\mathrm{c} = c) P(\mathrm{f}^{(1)} = f^{(1)} \mid \mathrm{c} = c) P(\mathrm{f}^{(2)} = f^{(2)} \mid \mathrm{c} = c).$$

Since c only has two possible values in our example, we can simplify this to:

$$P(\mathrm{c} = 1 \mid \mathrm{f}^{(1)} = f_1, \mathrm{f}^{(2)} = f_2) = \frac{g(1)}{g(0) + g(1)}$$

$$= \frac{1}{1 + \frac{g(0)}{g(1)}}$$

$$= \frac{1}{1 + \exp\left(\log g(0) - \log g(1)\right)}$$

$$= \sigma\left(\log g(1) - \log g(0)\right). \tag{3.7}$$

To go further, let's simplify $\log g(i)$:

$$\log g(i) = \log \quad \phi^{(\mathrm{c})i} (1 - \phi^{(\mathrm{c})})^{1-i} \phi_1^{(\mathrm{f})f_1} (1 - \phi_1^{(\mathrm{f})})^{1-f_1} \quad \sqrt{\frac{1}{2\pi\sigma_i^2}} \exp \quad -\frac{1}{2\sigma_i^2} (f_2 - \mu_i)^2$$

$$= i \log \phi^{(\mathrm{c})} + (1-i) \log \left(1 - \phi^{(\mathrm{c})}\right) + f_1 \log \phi_i^{(\mathrm{f})} + (1-f_1) \log(1-\phi_i^{(\mathrm{f})}) + \left(\frac{1}{2} \log \frac{1}{2\pi\sigma_i^2} - \frac{1}{2\sigma_i^2}\right) (f_2 - \mu_i)^2.$$

Substituting this back into equation 3.7, we obtain

$$P(\mathrm{c} = c \mid \mathrm{f}^{(1)} = f_1, \mathrm{f}^{(2)} = f_2) =$$

$$\sigma\Bigg(\log \phi^{(\mathrm{c})} - \log(1 - \phi^{(\mathrm{c})}) + f_1 \log \phi_1^{(\mathrm{f})} + (1 - f_1) \log(1 - \phi_1^{(\mathrm{f})}) \\ - f_1 \log \phi_0^{(\mathrm{f})} + (1 - f_1) \log(1 - \phi_0^{(\mathrm{f})}) \\ - \frac{1}{2} \log 2\pi\sigma_1^2 + \frac{1}{2} \log 2\pi\sigma_0^2 - \frac{1}{2\sigma_1^2} (f_2 - \mu_1)^2 + \frac{1}{2\sigma_0^2} (f_2 - \mu_0)^2 \Bigg).$$

From this formula, we can read off various intuitive properties of the Naive Bayes classifier's behavior on this example problem, regarding the *inference* that can be drawn from a trained model. The probability of the patient having the flu grows like a sigmoidal curve. We move farther to the left as f_2, the patient's temperature, moves farther away from μ_1, the average temperature of a flu patient.