2 Representation

Computationally accounting for uncertainty requires a formal representation. This chapter discusses how to represent uncertainty. We begin by introducing the notion of degree of belief and show how a set of axioms results in our ability to use probability distributions to quantify our uncertainty.[1] Because many important problems involve probability distributions over a large number of variables, we discuss a way to represent joint distributions efficiently that takes advantage of conditional independence between variables.

[1] For a more comprehensive elaboration, see E. T. Jaynes, *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

2.1 Degrees of Belief and Probability

In problems involving uncertainty, it is essential to be able to compare the plausibility of different statements. We would like to be able to represent, for example, that the plausibility of a proposition A is greater than the plausibility of another proposition B. If A represents ''I will wear a coat'', and B represents ''it will rain'', then we would write $A \succ B$. If we believe A and B are equally plausible, then we write $A \sim B$.

$$A \succ B \text{ if and only if } A \text{ is more likely than } B \tag{2.1}$$

$$A \sim B \text{ if and only if } A \text{ is as likely as } B \tag{2.2}$$

$$A \prec B \text{ if and only if } A \text{ is less likely than } B \tag{2.3}$$

We want to make certain assumptions about the relationships induced by the operators $\succ$, $\sim$, and $\prec$. The assumption of *universal comparability* requires exactly one of the following to hold: $A \succ B$, $A \sim B$, or $A \prec B$. The assumption of *transitivity*[2] requires that if $A \succeq B$ and $B \succeq C$ then $A \succeq C$. Universal comparability and transitivity assumptions lead to an ability to represent plausibility by a real-

[2] Here, $A \succeq B$ means A is at least as plausible as B in the same way that $a \geq b$ for two real values a and b means a is at least as large as b.

valued function. In other words, we can use a function P that has the following two properties:

$$P(A) > P(B) \text{ if and only if } A \succ B \tag{2.4}$$

$$P(A) = P(B) \text{ if and only if } A \sim B \tag{2.5}$$

If we make a set of additional assumptions[3] about the form of P, then we can show that P must satisfy the basic *axioms of probability* (appendix A.2). If we are certain of A, then $P(A) = 1$. If we believe A is impossible, then $P(A) = 0$. Uncertainty in the truth of A is represented by values in between the two extrema. Hence, probability masses must lie between 0 and 1 with $0 \leq P(A) \leq 1$.

[3] The axiomatization of subjective probability is given by P. C. Fishburn, "The Axioms of Subjective Probability," *Statistical Science*, vol. 1, no. 3, pp. 335–345, 1986. A more recent axiomatization is contained in M. J. Dupré and F. J. Tipler, "New Axioms for Rigorous Bayesian Probability," *Bayesian Analysis*, vol. 4, no. 3, pp. 599–606, 2009.

2.2 Probability Distributions

A *probability distribution* assigns probabilities to different outcomes.[4] There are different ways to represent probability distributions depending on whether they involve discrete or continuous outcomes.

[4] For an introduction to probability theory, see D. P. Bertsekas and J. N. Tsitsiklis, *Introduction to Probability*. Athena Scientific, 2002.

2.2.1 Discrete Probability Distributions

A *discrete probability distribution* is a distribution over a discrete set of values. We can represent such distributions as a *probability mass function*, which assigns a probability to every possible assignment of its input variable to a value. For example, suppose we have a variable X that can take on one of n different values: $1, \ldots, n$, or, using *colon notation*, $1 : n$.[5] A distribution associated with X specifies the n probabilities of the various assignments of values to that variable, in particular $P(X = 1), \ldots, P(X = n)$. Figure 2.1 shows an example of a discrete distribution.

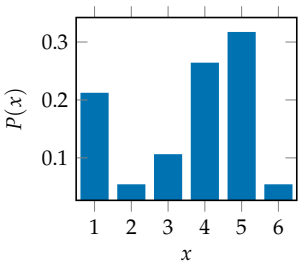


Figure 2.1. A probability mass function for a distribution over $1 : 6$.

[5] We will often use this colon notation for compactness. Other texts sometimes use the notation $[1 \,..\, n]$ for integer intervals from 1 to n. We will also use this colon notation to index into vectors and matrices. For example $x_{1:n}$ represents $x_1, \ldots, x_n$. The colon notation is sometimes used in programming languages, such as Julia and MATLAB.

There are constraints on the probability masses associated with discrete distributions. The masses must sum to one:

$$\sum_{i=1}^{n} P(X = i) = 1 \tag{2.6}$$

and $0 \leq P(X = i) \leq 1$ for all i.

For notational convenience, we will use lowercase letters and superscripts as shorthand when discussing the assignment of values to variables. For example, $P(x^3)$ is shorthand for $P(X = 3)$. If X is a *binary variable*, it can take on the value true or false.[6] We will use 0 to represent false and 1 to represent true. For example, we use $P(x^0)$ to represent the probability X is false.

The *parameters* of a distribution govern the probabilities associated with different assignments. For example, if we use X to represent the outcome of a roll of a six-sided die, then we would have $P(x^1) = \theta_1, \ldots, P(x^6) = \theta_6$, with $\theta_{1:6}$ being the six parameters of the distribution. However, we need only five *independent parameters* to uniquely specify the distribution over the outcomes of the roll because we know that the distribution must sum to 1.

[6] Julia, like many other programming languages, similarly treats Boolean values as 0 and 1 in numerical operations.

2.2.2 Continuous Probability Distributions

A *continuous probability distribution* is a distribution over a continuous set of values. Representing a distribution over a continuous variable is a little less straightforward than for a discrete variable. For instance, in many continuous distributions, the probability that a variable takes on a particular value is infinitesimally small. One way to represent a continuous probability distribution is to use a *probability density function* (see figure 2.2), represented with lowercase letters. If $p(x)$ is a probability density function over X, then $p(x)dx$ is the probability X falls within the interval $(x, x + dx)$ as $dx \to 0$. Similar to how the probability masses associated with a discrete distribution must sum to 1, a probability density function $p(x)$ must integrate to 1:

$$\int_{-\infty}^{\infty} p(x)\,\mathrm{d}x = 1 \tag{2.7}$$

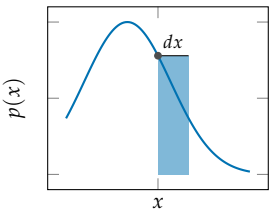


Figure 2.2. Probability density functions are used to represent continuous probability distributions. If $p(x)$ is a probability density, then $p(x)dx$ indicated by the area of the blue rectangle is the probability that a sample from the random variable falls within the interval $(x, x + dx)$ as $dx \to 0$.

Another way to represent a continuous distribution is with a *cumulative distribution function* (see figure 2.3), which specifies the probability mass associated with values below some threshold. If we have a cumulative distribution function P associated with variable X, then $P(x)$ represents the probability mass associated with X taking on a value less than or equal to x. A cumulative distribution function can be defined in terms of a probability density function p as follows:

$$\mathrm{cdf}_X(x) = P(X \leq x) = \int_{-\infty}^{x} p(x')\,\mathrm{d}x' \tag{2.8}$$

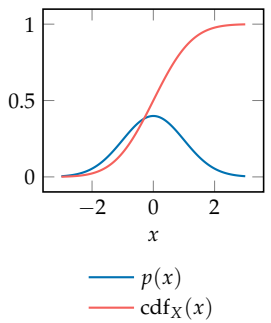


Figure 2.3. The probability density function and cumulative distribution function for a standard Gaussian distribution.

Related to the cumulative distribution function is the *quantile function*, also called the *inverse cumulative distribution function* (see figure 2.4). The value of $\text{quantile}_X(\alpha)$ is the value x such that $P(X \leq x) = \alpha$. In other words, the quantile function returns the minimum value of x whose cumulative distribution value exceeds α. Of course, we have $0 \leq \alpha \leq 1$.

There are many different parameterized families of distributions. We outline several in appendix B. A simple distribution family is the *uniform distribution* $\mathcal{U}(a, b)$, which assigns probability density uniformly between a and b, and zero elsewhere. Hence, the probability density function is $p(x) = 1/(b - a)$ for x in the interval $[a, b]$. We can use $\mathcal{U}(x \mid a, b)$ to represent the density at x.[7] The *support* of a distribution is the set of values that are assigned non-zero density. In the case of $\mathcal{U}(a, b)$, the support is the interval $[a, b]$. See example 2.1.

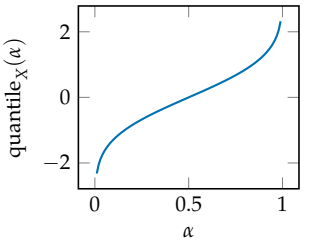


Figure 2.4. The quantile function for a standard Gaussian distribution.

[7] Some texts use a semicolon to separate the parameters of the distribution. For example, we would write $\mathcal{U}(x; a, b)$.

Example 2.1. An example uniform distribution with a lower bound of 0 and an upper bound of 10.

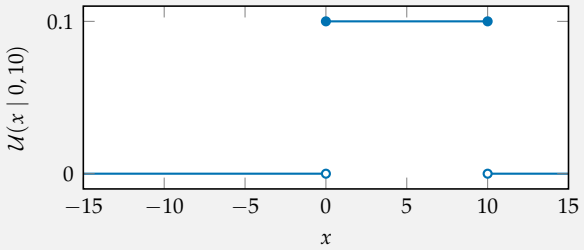
The uniform distribution $\mathcal{U}(0, 10)$ assigns equal probability to all values in the range $[0, 10]$ with a probability density function:

$$\mathcal{U}(x \mid 0, 10) = \begin{cases} 1/10 & \text{if } 0 \leq x \leq 10 \\ 0 & \text{otherwise} \end{cases} \tag{2.9}$$

The probability that a random sample from this distribution is equal to the constant π is essentially zero. However, we can define non-zero probabilities for samples being within some interval, say $[3, 5]$. For example, the probability that a sample lies between 3 and 5 given the distribution plotted below is:

$$\int_3^5 \mathcal{U}(x \mid 0, 10)\, \mathrm{d}x = \frac{5 - 3}{10} = \frac{1}{5} \tag{2.10}$$

The support of this distribution is the interval $[0, 10]$.



Another common distribution for continuous variables is the *Gaussian distribution* (also called the *normal distribution*). The Gaussian distribution is parameterized by a mean μ and variance σ^2:

$$p(x) = \mathcal{N}(x \mid \mu, \sigma^2) \tag{2.11}$$

Here, σ is the *standard deviation*, which is the square root of the variance. The variance is also commonly denoted by v. We use $\mathcal{N}(\mu, \sigma^2)$ to represent a Gaussian distribution with parameters μ and σ^2 and $\mathcal{N}(x \mid \mu, \sigma^2)$ to represent the probability density at x as given by

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right) \tag{2.12}$$

where ϕ is the *standard normal density function*:

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \tag{2.13}$$

Appendix B shows plots of Gaussian density functions with different parameters.

Although a Gaussian distribution is often convenient because it is defined by only two parameters and makes computation and derivation easy, it has some limitations. It assigns non-zero probability to large positive and negative values, which may not be appropriate for the quantity we are trying to model. For example, we might not want to assign non-zero probabilities for aircraft flying below the ground or at infeasible altitudes. We can use a *truncated Gaussian distribution* (see figure 2.5) to bound the *support* of possible values, that is, the range of values assigned non-zero probabilities. The density function is given by

$$\mathcal{N}(x \mid \mu, \sigma^2, a, b) = \frac{\frac{1}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)} \tag{2.14}$$

when x is within the interval (a, b).

The function Φ is the *standard normal cumulative distribution function* as given by

$$\Phi(x) = \int_{-\infty}^{x} \phi(x')\,\mathrm{d}x' \tag{2.15}$$

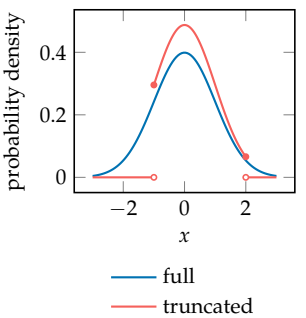


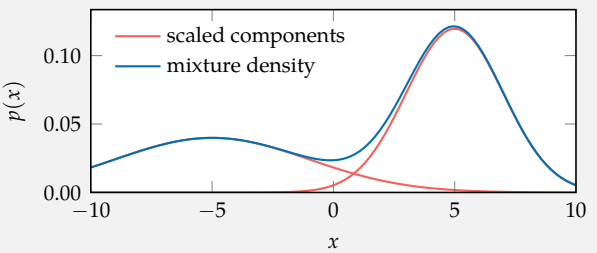
Figure 2.5. The probability density functions for a unit Gaussian distribution and the same distribution truncated between −1 and 2.

The Gaussian distribution is *unimodal*, meaning that there is a point in the distribution at which the density increases on one side and decreases on the other side. There are different ways to represent continuous distributions that are *multimodal*. One way is to use a *mixture model*, which is a mixture of multiple distributions. We mix together a collection of unimodal distributions to obtain a multimodal distribution. A *Gaussian mixture model* is a mixture model that is simply a weighted average of different Gaussian distributions. The parameters of a Gaussian mixture model include the parameters of the Gaussian distribution components $\mu_{1:n}, \sigma^2_{1:n}$ as well as their weights $\rho_{1:n}$. The density is given by

$$p(x \mid \mu_{1:n}, \sigma^2_{1:n}, \rho_{1:n}) = \sum_{i=1}^{n} \rho_i \, \mathcal{N}(x \mid \mu_i, \sigma_i^2) \tag{2.16}$$

where the weights must sum to 1. Example 2.2 shows a Gaussian mixture model with two components.

We can create a Gaussian mixture model with components $\mu_1 = 5$, $\sigma_1 = 2$ and $\mu_2 = -5$, $\sigma_2 = 4$, weighted according to $\rho_1 = 0.6$ and $\rho_2 = 0.4$. Below we plot the density of two components scaled by their weights.



Example 2.2. An example Gaussian mixture model.

Another approach to representing multimodal continuous distributions is through discretization. For example, we can represent a distribution over a continuous variable as a *piecewise-uniform density*. The density is specified by the bin edges, and a probability mass is associated with each bin. Such a piecewise-uniform distribution is a type of mixture model where the components are uniform distributions.

2.3 Joint Distributions

A *joint distribution* is a probability distribution over multiple variables. A distribution over a single variable is called a *univariate distribution*, and a distribution over multiple variables is called a *multivariate distribution*. If we have a joint distribution over two discrete variables X and Y, then $P(x, y)$ denotes the probability that both $X = x$ and $Y = y$.

From a joint distribution, we can compute a *marginal* distribution of a variable or a set of variables by summing out all other variables using what is known as the *law of total probability*:[8]

$$P(x) = \sum_y P(x, y) \tag{2.17}$$

This property is used throughout this book.

[8] If our distribution is continuous, then we integrate out the other variables when marginalizing. For example,

$$p(x) = \int p(x, y)\, \mathrm{d}y$$

Real-world decision making often requires reasoning about joint distributions involving many variables. Sometimes there are complex relationships between the variables that are important to represent. We may use different strategies to represent joint distributions depending on whether the variables involve discrete or continuous values.

2.3.1 Discrete Joint Distributions

If the variables are discrete, the joint distribution can be represented by a table like the one shown in table 2.1. That table lists all possible assignments of values to three variables. Each variable can only be 0 or 1, resulting in $2^3 = 8$ possible assignments. As with other discrete distributions, the probabilities in the table must sum to 1. It follows that although there are eight entries in the table, only seven of them are *independent*. If θ_i represents the probability in the ith row in the table, then we only need the parameters $\theta_1, \ldots, \theta_7$ to represent the distribution because we know $\theta_8 = 1 - (\theta_1 + \ldots + \theta_7)$.

Table 2.1. Example joint distribution involving binary variables X, Y, and Z.

X	Y	Z	$P(X, Y, Z)$
0	0	0	0.08
0	0	1	0.31
0	1	0	0.09
0	1	1	0.37
1	0	0	0.01
1	0	1	0.05
1	1	0	0.02
1	1	1	0.07

If we have n binary variables, then we need as many as $2^n - 1$ independent parameters to specify the joint distribution. This exponential growth in the number of parameters makes storing the distribution in memory difficult. In some cases, we can assume that our variables are *independent*, which means that the realization of one does not affect the probability distribution of the other. If X and Y are independent, which is sometimes written $X \perp Y$, then we know $P(x, y) = P(x)P(y)$ for all x and y. Suppose we have binary variables $X_1, \ldots, X_n$ that are all independent

of each other, resulting in $P(x_{1:n}) = \prod_i P(x_i)$. This factorization allows us to represent this joint distribution with only n independent parameters instead of the $2^n - 1$ required when we cannot assume independence. Independence can result in an enormous savings in terms of representational complexity, but it is often a poor assumption. See table 2.2 for an example.

We can represent joint distributions in terms of factors. A *factor* ϕ over a set of variables is a function from assignments of those variables to the real numbers. Algorithm 2.1 provides an implementation for discrete factors, and example 2.3 demonstrates how they work. In order to represent a probability distribution, the real numbers in the factor must be non-negative. A factor with non-negative values can be normalized such that it represents a probability distribution. Algorithm 2.2 normalizes a factor.

Table 2.2. If we know the variables in table 2.1 are independent, we can represent $P(x, y, z)$ using the product $P(x)P(y)P(z)$. This representation requires only one parameter for each of the three univariate distributions.

X	$P(X)$
0	0.85
1	0.15

Y	$P(Y)$
0	0.45
1	0.55

Z	$P(Z)$
0	0.20
1	0.80

```julia
const FactorTable = Dict{NamedTuple,Float64}

struct Variable
    name::Symbol
    m::Int # number of possible values
end

struct Factor
    vars::Vector{Variable}
    table::FactorTable
end

variablenames(ϕ::Factor) = [var.name for var in ϕ.vars]

function assignments(vars::AbstractVector{Variable})
    n = [var.name for var in vars]
    return [namedtuple(n, v) for v in product((1:v.m for v in vars)...)]
end
```

Algorithm 2.1. The `Factor` type over discrete variables. Each row in a factor table is represented by a named tuple. Each `Variable` has a name and can take on the values $1 : m$. We also include functions for returning a vector of variable names associated with a factor and an exhaustive list of assignments associated with a collection of variables. As discussed in appendix G.3.3, `product` produces the Cartesian product of a set of collections. It is imported from `Base.Iterators`.

Another approach to reduce the storage required to represent joint distributions with repeated values is to use a *decision tree*. A decision tree involving three discrete variables is shown in example 2.4. Although the savings in this example in terms of number of parameters may not be significant, it can become quite substantial when there are many variables and many repeated values.

We can instantiate the table from table 2.1 using the `Factor` type using the following code:

```
X = Variable(:x, 2)
Y = Variable(:y, 2)
Z = Variable(:z, 2)
ϕ = Factor([X, Y, Z], FactorTable(
    (x=1, y=1, z=1) ⇒ 0.08,
    (x=1, y=1, z=2) ⇒ 0.31,
    (x=1, y=2, z=1) ⇒ 0.09,
    (x=1, y=2, z=2) ⇒ 0.37,
    (x=2, y=1, z=1) ⇒ 0.01,
    (x=2, y=1, z=2) ⇒ 0.05,
    (x=2, y=2, z=1) ⇒ 0.02,
    (x=2, y=2, z=2) ⇒ 0.07,
))
```

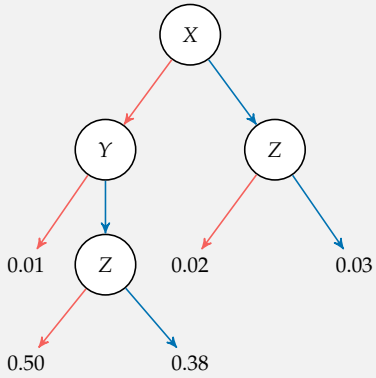
Example 2.3. Constructing a discrete factor using the `Factor` type.

```
function normalize!(ϕ::Factor)
    z = sum(p for (a,p) in ϕ.table)
    for (a,p) in ϕ.table
        ϕ.table[a] = p/z
    end
    return ϕ
end
```

Algorithm 2.2. Normalization of a factor `ϕ`, which divides all of the entries in the factor by the same scalar value such that they sum to 1, which is useful when working with factors that represent joint probability distributions.

Suppose we have the following table representing a joint probability distribution. We can use the decision tree to the right of it to more compactly represent the values in the table. Red arrows are followed when a variable is 0, and blue arrows are followed when a variable is 1. Instead of storing eight probabilities, we store only five along with a representation of the tree.

X	Y	Z	$P(X,Y,Z)$
0	0	0	0.01
0	0	1	0.01
0	1	0	0.50
0	1	1	0.38
1	0	0	0.02
1	0	1	0.03
1	1	0	0.02
1	1	1	0.03



Example 2.4. A decision tree can be a more efficient representation of a joint distribution than a table.

2.3.2 *Continuous Joint Distributions*

We can also define joint distributions over continuous variables. A rather simple distribution is the *multivariate uniform distribution*, which assigns a constant probability density everywhere there is support. We can use $\mathcal{U}(\mathbf{a}, \mathbf{b})$ to represent a uniform distribution over a *box*, which is a Cartesian product of intervals with the ith interval being $[a_i, b_i]$. This family of uniform distributions is a special type of *multivariate product distribution*, which is a distribution defined in terms of the product of univariate distributions. In this case,

$$\mathcal{U}(\mathbf{x} \mid \mathbf{a}, \mathbf{b}) = \prod_i \mathcal{U}(x_i \mid a_i, b_i) \tag{2.18}$$

We can create a mixture model from a weighted collection of multivariate uniform distributions, just as we can with univariate distributions. If we have a joint distribution over n variables and k mixture components, we need to define $k(2n + 1) - 1$ independent parameters. For each of the k components, we need to define the upper and lower bounds for each of the variables in addition to its weight. We can subtract 1 because the weights must sum to 1. Figure 2.6 shows an example that can be represented by five components.

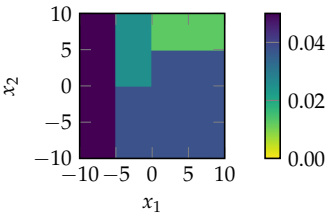


Figure 2.6. A density function for a mixture of multivariate uniform distributions.

It is also common to represent piecewise constant density functions by discretizing each of the variables independently. The discretization is represented by a set of bin edges for each variable. These bin edges define a grid over the variables. We then associate a constant probability density with each grid cell. The bin edges do not have to be uniformly separated. In some cases, it may be desirable to have increased resolution around certain values. Different variables might have different bin edges associated with them. If there are n variables and m bins for each variable, then we need $m^n - 1$ independent parameters to define the distribution—in addition to the values that define the bin edges.

In some cases, it may be more memory efficient to represent a continuous joint distribution as a decision tree in a manner similar to what we discussed for discrete joint distributions. The internal nodes compare variables against thresholds and the leaf nodes are density values. Figure 2.7 shows a decision tree that represents the density function in figure 2.6.

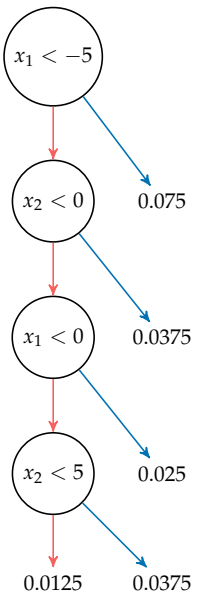


Figure 2.7. An example of a decision tree that represents a piecewise constant joint probability density defined over x_1 and x_2 over the interval $[-10, 10]^2$.

Another useful distribution is the *multivariate Gaussian distribution* with the density function

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) \tag{2.19}$$

where $\mathbf{x}$ is in $\mathbb{R}^n$, $\boldsymbol{\mu}$ is the *mean vector*, and $\boldsymbol{\Sigma}$ is the *covariance matrix*. We require that $\boldsymbol{\Sigma}$ be positive semidefinite (definition is reviewed in appendix A.5). The number of independent parameters is equal to $n + (n+1)n/2$, the number of components in $\boldsymbol{\mu}$ added to the number of components in the upper triangle of matrix $\boldsymbol{\Sigma}$.[9] Appendix B shows plots of different multivariate Gaussian density functions. We can also define *multivariate Gaussian mixture models*. Figure 2.8 shows an example of one with three components.

[9] If we know the parameters in the upper triangle of $\boldsymbol{\Sigma}$, we know the parameters in the lower triangle as well because $\boldsymbol{\Sigma}$ is symmetric.

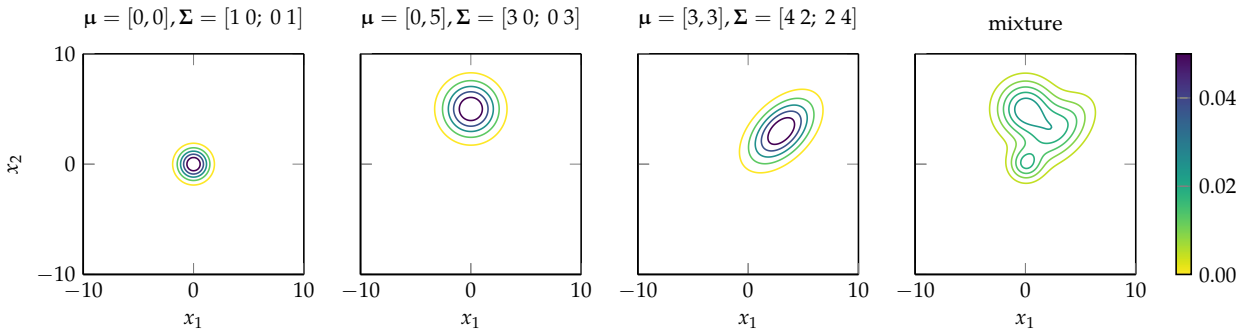


Figure 2.8. Multivariate Gaussian mixture model with three components. The components are mixed together with weights 0.1, 0.5, and 0.4, respectively.

If we have a multivariate Gaussian with all of the variables independent, then we can assume that the covariance matrix $\boldsymbol{\Sigma}$ is diagonal with only n independent parameters. In fact, we can write the density function as a product of univariate Gaussian densities:

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_i \mathcal{N}(x_i \mid \mu_i, \Sigma_{ii}) \tag{2.20}$$

2.4 Conditional Distributions

The previous section introduced the idea of independence, which can help reduce the number of parameters used to define a joint distribution. However, as was mentioned, independence can be too strong of an assumption. This section will introduce the idea of conditional independence, which can help reduce the number of independent parameters without making assumptions that are as strong. Before discussing conditional independence, we will first introduce the notion of a *conditional distribution*, which is a distribution over a variable given the value of one or more others.

The definition of *conditional probability* states that

$$P(x \mid y) = \frac{P(x, y)}{P(y)} \tag{2.21}$$

where $P(x \mid y)$ is read as ''probability of x given y.'' In some contexts, it is common to refer to y as *evidence*.

Since a conditional probability distribution is a probability distribution over one or more variables given some evidence, we know that

$$\sum_x P(x \mid y) = 1 \tag{2.22}$$

for a discrete X. If X is continuous, it integrates to 1.

We can incorporate the definition of conditional probability into equation (2.17) to obtain a slightly different form of the law of total probability:

$$P(x) = \sum_y P(x \mid y)P(y) \tag{2.23}$$

for a discrete distribution.

Another useful relationship that follows from the definition of conditional probability is *Bayes' rule*:[10]

$$P(x \mid y) = \frac{P(y \mid x)P(x)}{P(y)} \tag{2.24}$$

If we have a representation of a conditional distribution $P(y \mid x)$, we can apply Bayes's rule to swap the y and x to obtain the conditional distribution $P(x \mid y)$.

We will now discuss a variety of ways to represent conditional probability distributions over discrete and continuous variables.

[10] Named for the English statistician and Presbyterian minister Thomas Bayes (c. 1701–1761) who provided a formulation of this theorem.

2.4.1 Discrete Conditional Models

A conditional probability distribution over discrete variables can be represented using a table. In fact, we can use the same discrete factor representation that we used in section 2.3.1 for joint distributions. Table 2.3 shows an example of a table representing $P(X \mid Y, Z)$ with all binary variables. In contrast with a joint table (e.g, table 2.1), the column containing the probabilities need not sum to 1. However, if we sum over the probabilities that are consistent with what we are conditioning on, we must get 1. For example, conditioning on y^0 and z^0 (the evidence), we have

$$P(x^0 \mid y^0, z^0) + P(x^1 \mid y^0, z^0) = 0.08 + 0.92 = 1 \tag{2.25}$$

Conditional probability tables can become quite large. If we were to create a table like table 2.3 where all variables can take on m values and we are conditioning on n variables, there would be m^{n+1} rows. However, since the m values of the variable we are not conditioning on must sum to 1, there are only $(m-1)m^n$ independent parameters. There is still an exponential growth with the number of variables on which we condition. When there are many repeated values in the conditional probability table, a decision tree (introduced in section 2.3.1) may be a more efficient representation.

Table 2.3. An example conditional distribution involving the binary variables X, Y, and Z.

X	Y	Z	$P(X \mid Y, Z)$
0	0	0	0.08
0	0	1	0.15
0	1	0	0.05
0	1	1	0.10
1	0	0	0.92
1	0	1	0.85
1	1	0	0.95
1	1	1	0.90

2.4.2 *Conditional Gaussian Models*

A *conditional Gaussian* model can be used to represent a distribution over a continuous variable given one or more discrete variables. For example, if we have a continuous variable X and a discrete variable Y with values $1:n$, we could define a conditional Gaussian model as follows:

$$p(x \mid y) = \begin{cases} \mathcal{N}(x \mid \mu_1, \sigma_1^2) & \text{if } y^1 \\ \vdots & \\ \mathcal{N}(x \mid \mu_n, \sigma_n^2) & \text{if } y^n \end{cases} \tag{2.26}$$

with parameter vector $\theta = [\mu_{1:n}, \sigma_{1:n}]$. All $2n$ of those parameters can be varied independently. If we want to condition on multiple discrete variables, we just need to add more cases and associated parameters.

2.4.3 *Linear Gaussian Models*

The *linear Gaussian* model of $P(X \mid Y)$ represents the distribution over a continuous variable X as a Gaussian distribution with the mean being a linear function of the value of the continuous variable Y. The conditional density function is

$$p(x \mid y) = \mathcal{N}(x \mid my + b, \sigma^2) \tag{2.27}$$

with parameters $\theta = [m, b, \sigma]$. The mean is a linear function of y defined by parameters m and b. The variance is constant. Figure 2.9 shows an example.

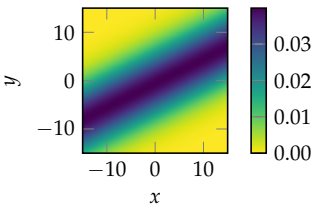


Figure 2.9. A linear Gaussian model with

$$p(x \mid y) = \mathcal{N}(x \mid 2y + 1, 10^2)$$

2.4.4 Conditional Linear Gaussian Models

The *conditional linear Gaussian* model combines the ideas of conditional Gaussian and linear Gaussian models to be able to handle conditioning a continuous variable on both discrete and continuous variables. Suppose we want to represent $p(X \mid Y, Z)$, where X and Y are continuous and Z is discrete with values $1 : n$. The conditional density function is then

$$p(x \mid y, z) = \begin{cases} \mathcal{N}(x \mid m_1 y + b_1, \sigma_1^2) & \text{if } z^1 \\ \vdots & \\ \mathcal{N}(x \mid m_n y + b_n, \sigma_n^2) & \text{if } z^n \end{cases} \tag{2.28}$$

Above, the parameter vector $\theta = [m_{1:n}, b_{1:n}, \sigma_{1:n}]$ has $3n$ components.

2.4.5 Sigmoid Models

We can use a *sigmoid*[11] model to represent a distribution over a binary variable conditioned on a continuous variable. For example, we may want to represent $P(x^1 \mid y)$, where x is binary and y is continuous. Of course, we could just set a threshold θ and say $P(x^1 \mid y) = 0$ if $y < \theta$ and $P(x^1 \mid y) = 1$ otherwise. However, in many applications, we may not want to have such a hard threshold that results in assigning zero probability to x^1 for certain values of y.

[11] A sigmoid is an "S"-shaped curve. There are different ways to define such a curve mathematically, but we will focus on the logit model.

Instead of a hard threshold, we could use a *soft threshold* that assigns low probabilities when below a threshold and high probabilities when above a threshold. One way to represent a soft threshold is to use a *logit model*, which produces a sigmoid curve:

$$P(x^1 \mid y) = \frac{1}{1 + \exp\left(-2\frac{y - \theta_1}{\theta_2}\right)} \tag{2.29}$$

The parameter θ_1 governs the location of the threshold, and θ_2 controls the "softness" or spread of the probabilities. Figure 2.10 shows an example plot of $P(x^1 \mid y)$ with a logit model.

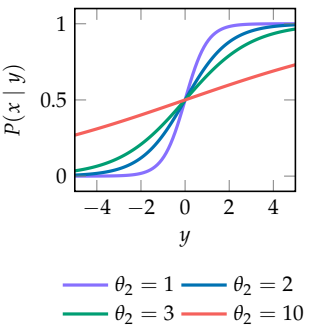


Figure 2.10. The logit model with $\theta_1 = 0$ and different values for θ_2.

2.4.6 Deterministic Variables

Some problems may involve a *deterministic variable* whose value is fixed given evidence. In other words, we assign probability 1 to a value that is a deterministic function of its evidence. Using a conditional probability table to represent

a discrete deterministic variable is possible, but it is wasteful. A single variable instantiation will have probability 1 for each parental instantiation, and the remaining entries will be 0. Our implementation can take advantage of this sparsity for a more compact representation. Algorithms in this text using discrete factors treat any assignments missing from the factor table as having value 0, making it so that we only have to store the assignments that have non-zero probability.

2.5 Bayesian Networks

A *Bayesian network* can be used to represent a joint probability distribution.[12] The structure of a Bayesian network is defined by a *directed acyclic graph* consisting of nodes and directed edges. Each node corresponds to a variable. Directed edges connect pairs of nodes, with cycles in the graph being prohibited. The arrows indicate direct probabilistic relationships. Associated with each node X_i is a conditional distribution $P(X_i \mid \text{Pa}(X_i))$, where $\text{Pa}(X_i)$ represents the parents of X_i in the graph. Algorithm 2.3 provides an implementation of a Bayesian network data structure. Example 2.5 illustrates the application of Bayesian networks to a satellite-monitoring problem.

[12] For an in-depth treatment of Bayesian networks and other forms of probabilistic graphical models, see the textbook by D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

```
struct BayesianNetwork
    vars::Vector{Variable}
    factors::Vector{Factor}
    graph::SimpleDiGraph{Int64}
end
```

Algorithm 2.3. A discrete Bayesian network representation in terms of a set of variables, factors, and a graph. The graph data structure is provided by `LightGraphs.jl`.

The *chain rule* for Bayesian networks specifies how to construct a joint distribution from the local conditional probability distributions. Suppose we have the variables $X_{1:n}$ and want to compute the probability of a particular assignment of all these variables to values $P(x_{1:n})$. The chain rule says

$$P(x_{1:n}) = \prod_{i=1}^{n} P(x_i \mid \text{pa}(x_i)) \tag{2.30}$$

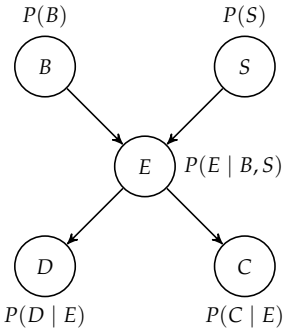
where $\text{pa}(x_i)$ is the particular assignment of the parents of X_i to their values. Algorithm 2.4 provides an implementation for Bayesian networks with conditional probability distributions represented as discrete factors.

Below the caption is a Bayesian network for a satellite-monitoring problem involving five binary variables. Fortunately, battery failure and solar panel failures are both rare, although solar panel failures are somewhat more likely than battery failures. Failures in either can lead to an electrical system failure. There may be causes of electrical system failure other than battery or solar panel failure, such as a problem with the power management unit. An electrical system failure can result in trajectory deviation, which can be observed from the earth by telescope, as well as a communication loss that interrupts the transmission of telemetry and mission data down to various ground stations. Other anomalies not involving the electrical system can result in trajectory deviation and communication loss.

Associated with each of the five variables are five conditional probability distributions. Because B and S do not have any parents, we only need to specify $P(B)$ and $P(S)$. The code below creates a Bayesian network structure with example values for the elements of the associated factor tables. The tuples in the factor tables index into the domains of the variables, which is $\{0, 1\}$ for all of the variables. For example, `(e=2,b=1,s=1)` corresponds to (e^1, b^0, s^0).

```
B = Variable(:b, 2); S = Variable(:s, 2)
E = Variable(:e, 2)
D = Variable(:d, 2); C = Variable(:c, 2)
vars = [B, S, E, D]
factors = [
    Factor([B], FactorTable((b=1,) ⇒ 0.01, (b=2,) ⇒ 0.99)),
    Factor([S], FactorTable((s=1,) ⇒ 0.02, (s=2,) ⇒ 0.98)),
    Factor([E,B,S], FactorTable(
        (e=1,b=1,s=1) ⇒ 0.90, (e=1,b=1,s=2) ⇒ 0.04,
        (e=1,b=2,s=1) ⇒ 0.05, (e=1,b=2,s=2) ⇒ 0.01,
        (e=2,b=1,s=1) ⇒ 0.10, (e=2,b=1,s=2) ⇒ 0.96,
        (e=2,b=2,s=1) ⇒ 0.95, (e=2,b=2,s=2) ⇒ 0.99)),
    Factor([D, E], FactorTable(
        (d=1,e=1) ⇒ 0.96, (d=1,e=2) ⇒ 0.03,
        (d=2,e=1) ⇒ 0.04, (d=2,e=2) ⇒ 0.97)),
    Factor([C, E], FactorTable(
        (d=1,e=1) ⇒ 0.98, (d=1,e=2) ⇒ 0.01,
        (d=2,e=1) ⇒ 0.02, (d=2,e=2) ⇒ 0.99))
]
graph = SimpleDiGraph(5)
add_edge!(graph, 1, 3); add_edge!(graph, 2, 3)
add_edge!(graph, 3, 4); add_edge!(graph, 3, 5)
bn = BayesianNetwork(vars, factors, graph)
```

Example 2.5. A Bayesian network representing a satellite-monitoring problem.



B battery failure
S solar panel failure
E electrical system failure
D trajectory deviation
C communication loss

```
function probability(bn::BayesianNetwork, assignment)
    subassignment(ϕ) = select(assignment, variablenames(ϕ))
    probability(ϕ) = get(ϕ.table, subassignment(ϕ), 0.0)
    return prod(probability(ϕ) for ϕ in bn.factors)
end
```

Algorithm 2.4. A function for evaluating the probability of an assignment represented as a named tuple (e.g., `(a=2, b=3, c=1)`) given a Bayesian network `bn`. The `select` function is provided by the `NamedTupleTools.jl` package, summarized in appendix G.4.1.

In the satellite example, suppose we want to compute the probability that nothing is wrong, that is, $P(b^0, s^0, e^0, d^0, c^0)$. From the chain rule,

$$P(b^0, s^0, e^0, d^0, c^0) = P(b^0)P(s^0)P(e^0 \mid b^0, s^0)P(d^0 \mid e^0)P(c^0 \mid e^0) \tag{2.31}$$

If we had fully specified a joint distribution over the five variables B, S, E, D, and C, then we would have needed $2^5 - 1 = 31$ independent parameters. The structure assumed in our Bayesian network allows us to specify the joint distribution using only $1 + 1 + 4 + 2 + 2 = 10$ independent parameters. The difference between 10 and 31 does not represent an especially significant savings in the number of parameters, but the savings can become enormous in larger Bayesian networks. The power of Bayesian networks comes from their ability to reduce the number of parameters required to specify a joint probability distribution.

2.6 Conditional Independence

The reason that a Bayesian network can represent a joint distribution with fewer independent parameters than would normally be required is due to the *conditional independence* assumptions encoded in its graphical structure.[13] Conditional independence is a generalization of the notion of independence introduced in section 2.3.1. Variables X and Y are conditionally independent given Z if and only if $P(X, Y \mid Z) = P(X \mid Z)P(Y \mid Z)$. The assertion that X and Y are conditionally independent given Z is written $(X \perp Y \mid Z)$. It is possible to show from this definition that $(X \perp Y \mid Z)$ if and only if $P(X \mid Z) = P(X \mid Y, Z)$. Given Z, information about Y provides no additional information about X, and vice versa. Example 2.6 provides an example.

[13] If the conditional independence assumptions made by the Bayesian network are invalid, then we run the risk of not properly modeling the joint distribution, as will be discussed in chapter 5.

We can use a set of rules to determine whether the structure of a Bayesian network implies that two variables must be conditionally independent given a set of other evidence variables.[14] Suppose we want to check whether $(A \perp B \mid \mathcal{C})$ is implied by the network structure, where $\mathcal{C}$ is a set of evidence variables. We have

[14] Even if the structure of a network does not imply conditional independence, there may still be conditional independence due to the choice of conditional probability distributions. See exercise 2.10.

Example 2.6. Conditional independence in the satellite-tracking problem.

Suppose the presence of satellite trajectory deviation (D) is conditionally independent of whether we have a communication loss (C) given knowledge of whether we have an electrical system failure (E). We would write this ($D \perp C \mid E$). If we know that we have an electrical system failure, then the fact that we observe a loss of communication has no impact on our belief that there is a trajectory deviation. We may have an elevated expectation that there is a trajectory deviation, but that is only because we know that an electrical system failure has occurred.

to check all possible undirected paths from A to B for what is called *d-separation*. A path between A and B is d-separated by $\mathcal{C}$ if any of the following are true:

1. The path contains a *chain* of nodes, $X \to Y \to Z$, such that Y is in $\mathcal{C}$.
2. The path contains a *fork*, $X \leftarrow Y \to Z$, such that Y is in $\mathcal{C}$.
3. The path contains an *inverted fork* (also called a *v-structure*), $X \to Y \leftarrow Z$, such that Y is *not* in $\mathcal{C}$ and no descendant of Y is in $\mathcal{C}$.

We say that A and B are d-separated by $\mathcal{C}$ if all paths between A and B are d-separated by $\mathcal{C}$. This d-separation implies that $(A \perp B \mid \mathcal{C})$.[15]

Sometimes the term *Markov blanket*[16] of a node X is used to refer to the minimal set of nodes that, if their values were known, makes X conditionally independent of all other nodes. A Markov blanket of a particular node turns out to consist of its parents, its children, and the other parents of its children.

[15] An algorithm for efficiently determining d-separation is a bit complicated. See algorithm 3.1 in D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[16] Named after the Russian mathematician Andrey Andreyevich Markov (1856–1922). J. Pearl, *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, 1988.

2.7 Summary

- Representing uncertainty as a probability distribution is motivated by a set of axioms related to the comparison of the plausibility of different statements.
- There are many different families of both discrete and continuous probability distributions.
- Continuous probability distributions can be represented by density functions.
- Probability distribution families can be combined together in mixtures to result in more flexible distributions.

- Joint distributions are distributions over multiple variables.
- Conditional distributions are distributions over one or more variables given values of evidence variables.
- A Bayesian network is defined by a graphical structure and a set of conditional distributions.
- Depending on the structure of the Bayesian network, we can represent joint distributions with fewer parameters due to conditional independence assumptions.

2.8 Exercises

Exercise 2.1. Consider a continuous random variable X that follows the exponential distribution parameterized by λ with density $p(x \mid \lambda) = \lambda \exp(-\lambda x)$ with nonnegative support. Compute the cumulative distribution function of X.

Solution: We start with the definition of the cumulative distribution function. Since the support of the distribution is lower-bounded by $x = 0$, there is no probability mass in the interval $(-\infty, 0)$, allowing us to adjust the lower bound of the integral to 0. After computing the integral, we obtain $\text{cdf}_X(x)$:

$$\begin{aligned}
\text{cdf}_X(x) &= \int_{-\infty}^{x} p(x')\,\mathrm{d}x' \\
\text{cdf}_X(x) &= \int_{0}^{x} \lambda e^{-\lambda x'}\,\mathrm{d}x' \\
\text{cdf}_X(x) &= -e^{-\lambda x'}\Big|_0^x \\
\text{cdf}_X(x) &= 1 - e^{-\lambda x}
\end{aligned}$$

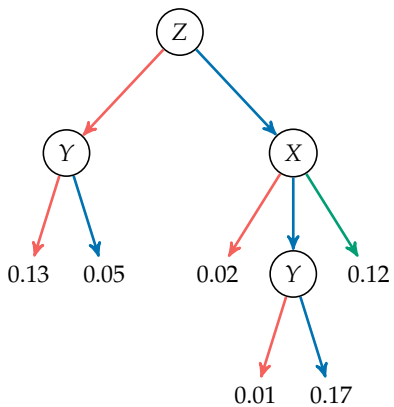
Exercise 2.2. For the density function in figure 2.6, what are the five components of the mixture? (There are multiple valid solutions.)

Solution: One solution is $\mathcal{U}([-10,-10],[-5,10])$, $\mathcal{U}([-5,0],[0,10])$, $\mathcal{U}([-5,-10],[0,0])$, $\mathcal{U}([0,-10],[10,5])$, and $\mathcal{U}([0,5],[10,10])$.

Exercise 2.3. Given the following table representation of $P(X, Y, Z)$, generate an equivalent compact decision tree representation.

X	Y	Z	$P(X, Y, Z)$
0	0	0	0.13
0	0	1	0.02
0	1	0	0.05
0	1	1	0.02
1	0	0	0.13
1	0	1	0.01
1	1	0	0.05
1	1	1	0.17
2	0	0	0.13
2	0	1	0.12
2	1	0	0.05
2	1	1	0.12

Solution: We start with the most common probabilities, 0.13, which occurs when $Z = 0$ and $Y = 0$, and 0.05, which occurs when $Z = 0$ and $Y = 1$. We choose to make Z the root of our decision tree and when $Z = 0$ we continue to a Y node. Based on the value of Y we branch to either 0.13 or 0.05. Next, we continue with cases when $Z = 1$. The most common probabilities are 0.02, which occurs when $Z = 1$ and $X = 0$, and 0.12, which occurs when $Z = 1$ and $X = 2$. So, when $Z = 1$, we choose to continue to an X node. Based on the whether X is 0, 1, or 2, we continue to 0.02, a Y node, or 0.12, respectively. Finally, based on the value of Y, we branch to either 0.01 or 0.17.



Exercise 2.4. Suppose we want to specify a multivariate Gaussian mixture model with three components defined over four variables. We require that two of the three Gaussian distributions assume independence between the four variables, while the other Gaussian distribution is defined without any independence assumptions. How many independent parameters are required to specify this mixture model?

Solution: For a Gaussian distribution over four variables ($n = 4$) with independence assumptions, we need to specify $n + n = 2n = 8$ independent parameters; there are four parameters for the mean vector and four parameters for the covariance matrix (which is equivalent to the mean and variance parameters of four independent univariate Gaussian distributions). For a Gaussian distribution over four variables without independence assumptions, we need to specify $n + n(n+1)/2 = 14$ independent parameters; there are four parameters for the mean vector and ten parameters for the covariance matrix. Additionally, for our three mixture components ($k = 3$), we need to specify $k - 1 = 2$ independent parameters for the weights. Thus, we need $2(8) + 1(14) + 2 = 32$ independent parameters to specify this mixture distribution.

Exercise 2.5. We have three independent variables $X_{1:3}$ defined by piecewise-constant densities with 4, 7, and 3 bin edges, respectively. How many independent parameters are required to specify their joint distribution?

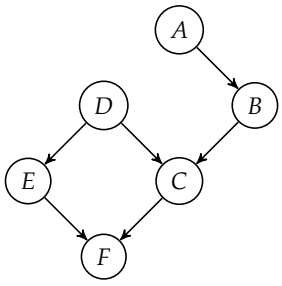
Solution: If we have a piecewise-constant density with m bins edges, then there are $m - 1$ bins and $m - 2$ independent parameters. For this problem, there will be $(4 - 2) + (7 - 2) + (3 - 2) = 8$ independent parameters.

Exercise 2.6. Suppose we have four continuous random variables, X_1, X_2, Y_1, and Y_2, and we want to construct a linear Gaussian model of $X = X_{1:2}$ given $Y = Y_{1:2}$, i.e. $p(X \mid Y)$. How many independent parameters are required for this model?

Solution: In this case, our mean vector for the Gaussian distribution is two-dimensional and requires four independent parameters for the transformation matrix $\mathbf{M}$ and two independent parameters for the bias vector $\mathbf{b}$. We also require three independent parameters for the covariance matrix $\mathbf{\Sigma}$. In total, we need $4 + 2 + 3 = 9$ independent parameters to specify this model:

$$p(\mathbf{x} \mid \mathbf{y}) = \mathcal{N}(\mathbf{x} \mid \mathbf{M}\mathbf{y} + \mathbf{b}, \mathbf{\Sigma})$$

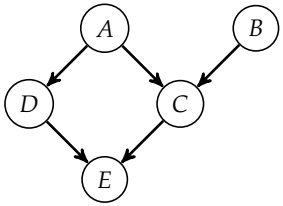
Exercise 2.7. Given the following Bayesian network where each node can take on one of four values, how many independent parameters are there? What is the percent reduction in the number of independent parameters required when using the following Bayesian network compared to using a full joint probability table?



Solution: The number of independent parameters for each node is equal to $(k-1)k^m$ where k is the number of values the node can take on and m is the number of parents that the node has. The variable A has 3, B has 12, C has 48, D has 3, E has 12, and F has 48 and independent parameters. There are 126 total independent parameters for this Bayesian network.

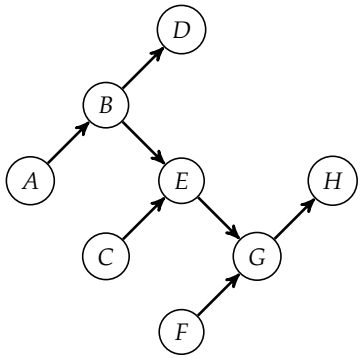
The number of independent parameters required to specify a joint probability table over n variables that can take on k values is equal to $k^n - 1$. Therefore, specifying a joint probability table would require $4^6 - 1 = 4096 - 1 = 4095$ independent parameters. The percent reduction in the number of independent parameters required is $(4095 - 126)/4095 \approx 96.9\,\%$.

Exercise 2.8. Given the following Bayesian network, is A d-separated from E given C?



Solution: There are two paths from A to E: $A \to D \to E$ and $A \to C \to E$. There is d-separation along the second path, but not the first. Hence, A is not d-separated from E given C.

Exercise 2.9. Given the following Bayesian network, determine the Markov blanket of B.



Solution: Paths from B to A can only be d-separated given A. Paths from B to D can only be d-separated given D. Paths from B to E and simultaneously F, G, and H can be efficiently d-separated given E. Paths from B to C are naturally d-separated due to a v-structure; however, since E must be contained in our Markov blanket, paths from B to C given E can only be d-separated given C. So, the Markov blanket of B is $\{A, C, D, E\}$.

Exercise 2.10. In a Bayesian network with structure $A \rightarrow B$, is it possible for A to be independent of B?

Solution: There is a direct arrow from A to B, which indicates that independence is not implied. However, this does not mean that they are not independent. Whether or not A and B are independent depends on the choice of conditional probability tables. We can choose the tables so that there is independence. For example, suppose both variables are binary and $P(a) = 0.5$ is uniform and $P(b \mid a) = 0.5$. Clearly, $P(A)P(B \mid A) = P(A)P(B)$, which means they are independent.