a source bit is flipped. Are parity bits or source bits more likely to be among these three flipped bits, or are all seven bits equally likely to be corrupted when the noise vector has weight two? The Hamming code is in fact completely symmetric in the protection it affords to the seven bits (assuming a binary symmetric channel). [This symmetry can be proved by showing that the role of a parity bit can be exchanged with a source bit and the resulting code is still a $(7, 4)$ Hamming code; see below.] The probability that any one bit ends up corrupted is the same for all seven bits. So the probability of bit error (for the source bits) is simply three sevenths of the probability of block error.

$$p_{\rm b} \simeq \frac{3}{7} p_{\rm B} \simeq 9f^2. \tag{1.48}$$

Symmetry of the Hamming $(7, 4)$ *code*

To prove that the $(7, 4)$ code protects all bits equally, we start from the parity-check matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{1.49}$$

The symmetry among the seven transmitted bits will be easiest to see if we reorder the seven bits using the permutation $(t_1 t_2 t_3 t_4 t_5 t_6 t_7) \rightarrow (t_5 t_2 t_3 t_4 t_1 t_6 t_7)$. Then we can rewrite $\mathbf{H}$ thus:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}. \tag{1.50}$$

Now, if we take any two parity constraints that $\mathbf{t}$ satisfies and add them together, we get another parity constraint. For example, row 1 asserts $t_5 + t_2 + t_3 + t_1 = \text{even}$, and row 2 asserts $t_2 + t_3 + t_4 + t_6 = \text{even}$, and the sum of these two constraints is

$$t_5 + 2t_2 + 2t_3 + t_1 + t_4 + t_6 = \text{even}; \tag{1.51}$$

we can drop the terms $2t_2$ and $2t_3$, since they are even whatever t_2 and t_3 are; thus we have derived the parity constraint $t_5 + t_1 + t_4 + t_6 = \text{even}$, which we can if we wish add into the parity-check matrix as a fourth row. [The set of vectors satisfying $\mathbf{Ht} = \mathbf{0}$ will not be changed.] We thus define

$$\mathbf{H}' = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}. \tag{1.52}$$

The fourth row is the sum (modulo two) of the top two rows. Notice that *the second, third, and fourth rows are all cyclic shifts of the top row.* If, having added the fourth redundant constraint, we drop the first constraint, we obtain a new parity-check matrix $\mathbf{H}''$,

$$\mathbf{H}'' = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}, \tag{1.53}$$

which still satisfies $\mathbf{H}''\mathbf{t} = 0$ for all codewords, and which looks just like the starting $\mathbf{H}$ in (1.50), except that all the columns have shifted along one to the right, and the rightmost column has reappeared at the left (a cyclic permutation of the columns).

This establishes the symmetry among the seven bits. Iterating the above procedure five more times, we can make a total of seven different $\mathbf{H}$ matrices for the same original code, each of which assigns each bit to a different role.

We may also construct the super-redundant seven-row parity-check matrix for the code,

$$\mathbf{H}''' = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} . \tag{1.54}$$

This matrix is 'redundant' in the sense that the space spanned by its rows is only three-dimensional, not seven.

This matrix is also a *cyclic* matrix. Every row is a cyclic permutation of the top row.

Cyclic codes: if there is an ordering of the bits $t_1 \ldots t_N$ such that a linear code has a *cyclic* parity-check matrix, then the code is called a *cyclic code*.

The codewords of such a code also have cyclic properties: any cyclic permutation of a codeword is a codeword.

For example, the Hamming $(7, 4)$ code, with its bits ordered as above, consists of all seven cyclic shifts of the codewords 1110100 and 1011000, and the codewords 0000000 and 1111111.

Cyclic codes are a cornerstone of the algebraic approach to error-correcting codes. We won't use them again in this book, however, as they have been superceded by sparse-graph codes (Part VI).

Solution to exercise 1.7 (p.13). There are fifteen non-zero noise vectors which give the all-zero syndrome; these are precisely the fifteen non-zero codewords of the Hamming code. Notice that because the Hamming code is *linear*, the sum of any two codewords is a codeword.

Graphs corresponding to codes

Solution to exercise 1.9 (p.14). When answering this question, you will probably find that it is easier to invent new codes than to find optimal decoders for them. There are many ways to design codes, and what follows is just one possible train of thought. We make a linear block code that is similar to the $(7, 4)$ Hamming code, but bigger.

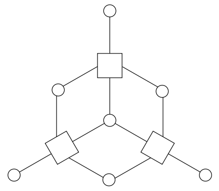


Figure 1.20. The graph of the $(7, 4)$ Hamming code. The 7 circles are the bit nodes and the 3 squares are the parity-check nodes.

Many codes can be conveniently expressed in terms of graphs. In figure 1.13, we introduced a pictorial representation of the $(7, 4)$ Hamming code. If we replace that figure's big circles, each of which shows that the parity of four particular bits is even, by a 'parity-check node' that is connected to the four bits, then we obtain the representation of the $(7, 4)$ Hamming code by a *bipartite graph* as shown in figure 1.20. The 7 circles are the 7 transmitted bits. The 3 squares are the parity-check nodes (not to be confused with the 3 parity-check *bits*, which are the three most peripheral circles). The graph is a 'bipartite' graph because its nodes fall into two classes – bits and checks

– and there are edges only between nodes in different classes. The graph and the code's parity-check matrix (1.30) are simply related to each other: each parity-check node corresponds to a row of **H** and each bit node corresponds to a column of **H**; for every 1 in **H**, there is an edge between the corresponding pair of nodes.

Having noticed this connection between linear codes and graphs, one way to invent linear codes is simply to think of a bipartite graph. For example, a pretty bipartite graph can be obtained from a dodecahedron by calling the vertices of the dodecahedron the parity-check nodes, and putting a transmitted bit on each edge in the dodecahedron. This construction defines a parity-check matrix in which every column has weight 2 and every row has weight 3. [The weight of a binary vector is the number of 1s it contains.]

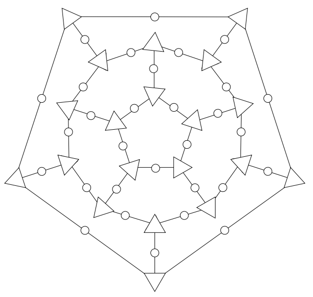


Figure 1.21. The graph defining the $(30, 11)$ dodecahedron code. The circles are the 30 transmitted bits and the triangles are the 20 parity checks. One parity check is redundant.

This code has $N = 30$ bits, and it appears to have $M_{\text{apparent}} = 20$ parity-check constraints. Actually, there are only $M = 19$ *independent* constraints; the 20th constraint is redundant (that is, if 19 constraints are satisfied, then the 20th is automatically satisfied); so the number of source bits is $K = N - M = 11$. The code is a $(30, 11)$ code.

It is hard to find a decoding algorithm for this code, but we can estimate its probability of error by finding its lowest-weight codewords. If we flip all the bits surrounding one face of the original dodecahedron, then all the parity checks will be satisfied; so the code has 12 codewords of weight 5, one for each face. Since the lowest-weight codewords have weight 5, we say that the code has distance $d = 5$; the $(7, 4)$ Hamming code had distance 3 and could correct all single bit-flip errors. A code with distance 5 can correct all double bit-flip errors, but there are some triple bit-flip errors that it cannot correct. So the error probability of this code, assuming a binary symmetric channel, will be dominated, at least for low noise levels f, by a term of order f^3, perhaps something like

$$12 \binom{5}{3} f^3 (1-f)^{27}. \tag{1.55}$$

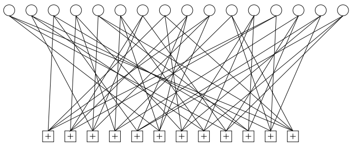


Figure 1.22. Graph of a rate-$1/4$ low-density parity-check code (Gallager code) with blocklength $N = 16$, and $M = 12$ parity-check constraints. Each white circle represents a transmitted bit. Each bit participates in $j = 3$ constraints, represented by ⊞ squares. The edges between nodes were placed at random. (See Chapter 47 for more.)

Of course, there is no obligation to make codes whose graphs can be represented on a plane, as this one can; the best linear codes, which have simple graphical descriptions, have graphs that are more tangled, as illustrated by the tiny $(16, 4)$ code of figure 1.22.

Furthermore, there is no reason for sticking to linear codes; indeed some nonlinear codes – codes whose codewords cannot be defined by a linear equation like $\mathbf{Ht} = \mathbf{0}$ – have very good properties. But the encoding and decoding of a nonlinear code are even trickier tasks.

Solution to exercise 1.10 (p.14). First let's assume we are making a linear code and decoding it with syndrome decoding. If there are N transmitted bits, then the number of possible error patterns of weight up to two is

$$\binom{N}{2} + \binom{N}{1} + \binom{N}{0}. \tag{1.56}$$

For $N = 14$, that's $91 + 14 + 1 = 106$ patterns. Now, every distinguishable error pattern must give rise to a distinct syndrome; and the syndrome is a list of M bits, so the maximum possible number of syndromes is 2^M. For a $(14, 8)$ code, $M = 6$, so there are at most $2^6 = 64$ syndromes. The number of possible error patterns of weight up to two, 106, is bigger than the number of syndromes, 64, so we can immediately rule out the possibility that there is a $(14, 8)$ code that is 2-error-correcting.

The same counting argument works fine for nonlinear codes too. When the decoder receives $\mathbf{r} = \mathbf{t} + \mathbf{n}$, his aim is to deduce both $\mathbf{t}$ and $\mathbf{n}$ from $\mathbf{r}$. If it is the case that the sender can select any transmission $\mathbf{t}$ from a code of size $S_{\mathbf{t}}$, and the channel can select any noise vector from a set of size $S_{\mathbf{n}}$, and those two selections can be recovered from the received bit string $\mathbf{r}$, which is one of at most 2^N possible strings, then it must be the case that

$$S_{\mathbf{t}} S_{\mathbf{n}} \leq 2^N. \tag{1.57}$$

So, for a (N, K) two-error-correcting code, whether linear or nonlinear,

$$2^K \left[\binom{N}{2} + \binom{N}{1} + \binom{N}{0} \right] \leq 2^N. \tag{1.58}$$

Solution to exercise 1.11 (p.14). There are various strategies for making codes that can correct multiple errors, and I strongly recommend you think out one or two of them for yourself.

If your approach uses a linear code, e.g., one with a collection of M parity checks, it is helpful to bear in mind the counting argument given in the previous exercise, in order to anticipate how many parity checks, M, you might need.

Examples of codes that can correct any two errors are the $(30, 11)$ dodecahedron code on page 20, and the $(15, 6)$ pentagonful code to be introduced on p. 221. Further simple ideas for making codes that can correct multiple errors from codes that can correct only one error are discussed in section 13.7.

Solution to exercise 1.12 (p.16). The probability of error of R_3^2 is, to leading order,

$$p_{\mathrm{b}}(\mathrm{R}_3^2) \simeq 3 \left[p_{\mathrm{b}}(\mathrm{R}_3)\right]^2 = 3(3f^2)^2 + \cdots = 27f^4 + \cdots, \tag{1.59}$$

whereas the probability of error of R_9 is dominated by the probability of five flips,

$$p_{\mathrm{b}}(\mathrm{R}_9) \simeq \binom{9}{5} f^5 (1-f)^4 \simeq 126 f^5 + \cdots. \tag{1.60}$$

The R_3^2 decoding procedure is therefore suboptimal, since there are noise vectors of weight four that cause it to make a decoding error.

It has the advantage, however, of requiring smaller computational resources: only memorization of three bits, and counting up to three, rather than counting up to nine.

This simple code illustrates an important concept. Concatenated codes are widely used in practice because concatenation allows large codes to be implemented using simple encoding and decoding hardware. Some of the best known practical codes are concatenated codes.

2

Probability, Entropy, and Inference

This chapter, and its sibling, Chapter 8, devote some time to notation. Just as the White Knight distinguished between the song, the name of the song, and what the name of the song was called (Carroll, 1998), we will sometimes need to be careful to distinguish between a random variable, the value of the random variable, and the proposition that asserts that the random variable has a particular value. In any particular chapter, however, I will use the most simple and friendly notation possible, at the risk of upsetting pure-minded readers. For example, if something is 'true with probability 1', I will usually simply say that it is 'true'.

▶ 2.1 Probabilities and ensembles

An ensemble X is a triple $(x, \mathcal{A}_X, \mathcal{P}_X)$, where the *outcome* x is the value of a random variable, which takes on one of a set of possible values, $\mathcal{A}_X = \{a_1, a_2, \ldots, a_i, \ldots, a_I\}$, having probabilities $\mathcal{P}_X = \{p_1, p_2, \ldots, p_I\}$, with $P(x\!=\!a_i) = p_i$, $p_i \geq 0$ and $\sum_{a_i \in \mathcal{A}_X} P(x\!=\!a_i) = 1$.

The name $\mathcal{A}$ is mnemonic for 'alphabet'. One example of an ensemble is a letter that is randomly selected from an English document. This ensemble is shown in figure 2.1. There are twenty-seven possible letters: a–z, and a space character '-'.

i	a_i	p_i
1	a	0.0575
2	b	0.0128
3	c	0.0263
4	d	0.0285
5	e	0.0913
6	f	0.0173
7	g	0.0133
8	h	0.0313
9	i	0.0599
10	j	0.0006
11	k	0.0084
12	l	0.0335
13	m	0.0235
14	n	0.0596
15	o	0.0689
16	p	0.0192
17	q	0.0008
18	r	0.0508
19	s	0.0567
20	t	0.0706
21	u	0.0334
22	v	0.0069
23	w	0.0119
24	x	0.0073
25	y	0.0164
26	z	0.0007
27	-	0.1928

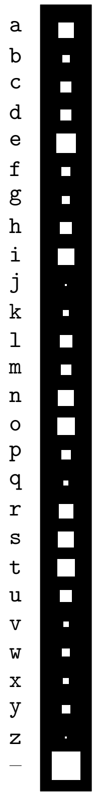


Figure 2.1. Probability distribution over the 27 outcomes for a randomly selected letter in an English language document (estimated from *The Frequently Asked Questions Manual for Linux*). The picture shows the probabilities by the areas of white squares.

Abbreviations. Briefer notation will sometimes be used. For example, $P(x\!=\!a_i)$ may be written as $P(a_i)$ or $P(x)$.

Probability of a subset. If T is a subset of $\mathcal{A}_X$ then:

$$P(T) = P(x\!\in\!T) = \sum_{a_i \in T} P(x\!=\!a_i). \tag{2.1}$$

For example, if we define V to be vowels from figure 2.1, $V = \{\mathtt{a}, \mathtt{e}, \mathtt{i}, \mathtt{o}, \mathtt{u}\}$, then

$$P(V) = 0.06 + 0.09 + 0.06 + 0.07 + 0.03 = 0.31. \tag{2.2}$$

A joint ensemble XY is an ensemble in which each outcome is an ordered pair x, y with $x \in \mathcal{A}_X = \{a_1, \ldots, a_I\}$ and $y \in \mathcal{A}_Y = \{b_1, \ldots, b_J\}$.

We call $P(x, y)$ the joint probability of x and y.

Commas are optional when writing ordered pairs, so $xy \Leftrightarrow x, y$.

N.B. In a joint ensemble XY the two variables are not necessarily independent.

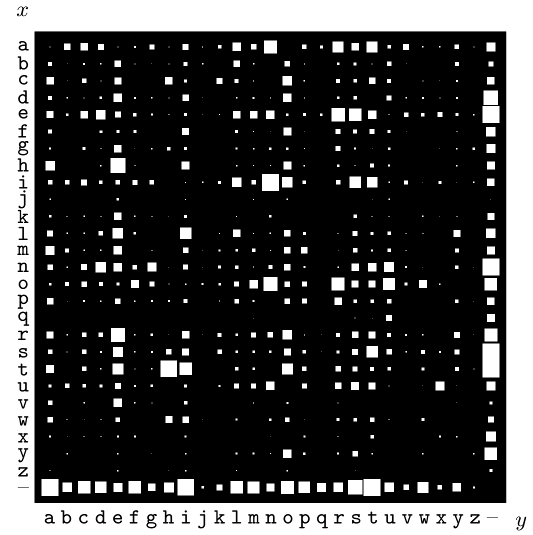


Figure 2.2. The probability distribution over the 27×27 possible bigrams xy in an English language document, *The Frequently Asked Questions Manual for Linux.*

Marginal probability. We can obtain the marginal probability $P(x)$ from the joint probability $P(x, y)$ by summation:

$$P(x\!=\!a_i) \equiv \sum_{y \in \mathcal{A}_Y} P(x\!=\!a_i, y). \tag{2.3}$$

Similarly, using briefer notation, the marginal probability of y is:

$$P(y) \equiv \sum_{x \in \mathcal{A}_X} P(x, y). \tag{2.4}$$

Conditional probability

$$P(x\!=\!a_i \,|\, y\!=\!b_j) \equiv \frac{P(x\!=\!a_i, y\!=\!b_j)}{P(y\!=\!b_j)} \text{ if } P(y\!=\!b_j) \neq 0. \tag{2.5}$$

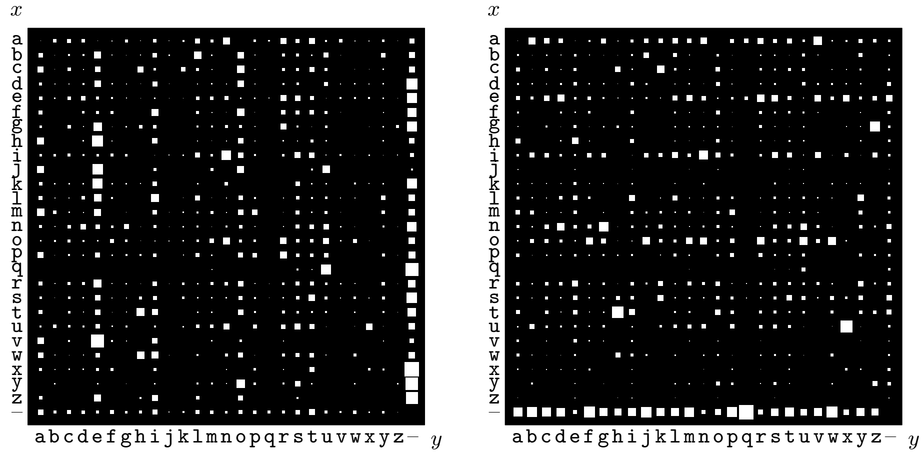
[If $P(y\!=\!b_j) = 0$ then $P(x\!=\!a_i \,|\, y\!=\!b_j)$ is undefined.]

We pronounce $P(x\!=\!a_i \,|\, y\!=\!b_j)$ 'the probability that x equals a_i, given y equals b_j'.

Example 2.1. An example of a joint ensemble is the ordered pair XY consisting of two successive letters in an English document. The possible outcomes are ordered pairs such as `aa`, `ab`, `ac`, and `zz`; of these, we might expect `ab` and `ac` to be more probable than `aa` and `zz`. An estimate of the joint probability distribution for two neighbouring characters is shown graphically in figure 2.2.

This joint ensemble has the special property that its two marginal distributions, $P(x)$ and $P(y)$, are identical. They are both equal to the monogram distribution shown in figure 2.1.

From this joint ensemble $P(x, y)$ we can obtain conditional distributions, $P(y \,|\, x)$ and $P(x \,|\, y)$, by normalizing the rows and columns, respectively (figure 2.3). The probability $P(y \,|\, x\!=\!\mathtt{q})$ is the probability distribution of the second letter given that the first letter is a `q`. As you can see in figure 2.3a, the two most probable values for the second letter y given



(a) $P(y \mid x)$

(b) $P(x \mid y)$

Figure 2.3. Conditional probability distributions. (a) $P(y \mid x)$: Each *row* shows the conditional distribution of the second letter, y, given the first letter, x, in a bigram xy. (b) $P(x \mid y)$: Each *column* shows the conditional distribution of the first letter, x, given the second letter, y.

that the first letter x is `q` are `u` and `-`. (The space is common after `q` because the source document makes heavy use of the word `FAQ`.)

The probability $P(x \mid y = \texttt{u})$ is the probability distribution of the first letter x given that the second letter y is a `u`. As you can see in figure 2.3b the two most probable values for x given $y = \texttt{u}$ are `n` and `o`.

Rather than writing down the joint probability directly, we often define an ensemble in terms of a collection of conditional probabilities. The following rules of probability theory will be useful. ($\mathcal{H}$ denotes assumptions on which the probabilities are based.)

Product rule – obtained from the definition of conditional probability:

$$P(x, y \mid \mathcal{H}) = P(x \mid y, \mathcal{H}) P(y \mid \mathcal{H}) = P(y \mid x, \mathcal{H}) P(x \mid \mathcal{H}). \tag{2.6}$$

This rule is also known as the chain rule.

Sum rule – a rewriting of the marginal probability definition:

$$\begin{aligned} P(x \mid \mathcal{H}) &= \sum_y P(x, y \mid \mathcal{H}) && (2.7) \\ &= \sum_y P(x \mid y, \mathcal{H}) P(y \mid \mathcal{H}). && (2.8) \end{aligned}$$

Bayes' theorem – obtained from the product rule:

$$\begin{aligned} P(y \mid x, \mathcal{H}) &= \frac{P(x \mid y, \mathcal{H}) P(y \mid \mathcal{H})}{P(x \mid \mathcal{H})} && (2.9) \\ &= \frac{P(x \mid y, \mathcal{H}) P(y \mid \mathcal{H})}{\sum_{y'} P(x \mid y', \mathcal{H}) P(y' \mid \mathcal{H})}. && (2.10) \end{aligned}$$

Independence. Two random variables X and Y are *independent* (sometimes written $X \perp Y$) if and only if

$$P(x, y) = P(x) P(y). \tag{2.11}$$

Exercise 2.2.[*1*, p.40] Are the random variables X and Y in the joint ensemble of figure 2.2 independent?

I said that we often define an ensemble in terms of a collection of conditional probabilities. The following example illustrates this idea.

Example 2.3. Jo has a test for a nasty disease. We denote Jo's state of health by the variable a and the test result by b.

$$\begin{array}{ll} a=1 & \text{Jo has the disease} \\ a=0 & \text{Jo does not have the disease.} \end{array} \tag{2.12}$$

The result of the test is either 'positive' ($b=1$) or 'negative' ($b=0$); the test is 95% reliable: in 95% of cases of people who really have the disease, a positive result is returned, and in 95% of cases of people who do not have the disease, a negative result is obtained. The final piece of background information is that 1% of people of Jo's age and background have the disease.

OK – Jo has the test, and the result is positive. What is the probability that Jo has the disease?

Solution. We write down all the provided probabilities. The test reliability specifies the conditional probability of b given a:

$$\begin{array}{ll} P(b\!=\!1\,|\,a\!=\!1) = 0.95 & P(b\!=\!1\,|\,a\!=\!0) = 0.05 \\ P(b\!=\!0\,|\,a\!=\!1) = 0.05 & P(b\!=\!0\,|\,a\!=\!0) = 0.95; \end{array} \tag{2.13}$$

and the disease prevalence tells us about the marginal probability of a:

$$P(a\!=\!1) = 0.01 \qquad P(a\!=\!0) = 0.99. \tag{2.14}$$

From the marginal $P(a)$ and the conditional probability $P(b\,|\,a)$ we can deduce the joint probability $P(a,b) = P(a)P(b\,|\,a)$ and any other probabilities we are interested in. For example, by the sum rule, the marginal probability of $b\!=\!1$ – the probability of getting a positive result – is

$$P(b\!=\!1) = P(b\!=\!1\,|\,a\!=\!1)P(a\!=\!1) + P(b\!=\!1\,|\,a\!=\!0)P(a\!=\!0). \tag{2.15}$$

Jo has received a positive result $b\!=\!1$ and is interested in how plausible it is that she has the disease (i.e., that $a\!=\!1$). The man in the street might be duped by the statement 'the test is 95% reliable, so Jo's positive result implies that there is a 95% chance that Jo has the disease', but this is incorrect. The correct solution to an inference problem is found using Bayes' theorem.

$$\begin{aligned} P(a\!=\!1\,|\,b\!=\!1) &= \frac{P(b\!=\!1\,|\,a\!=\!1)P(a\!=\!1)}{P(b\!=\!1\,|\,a\!=\!1)P(a\!=\!1) + P(b\!=\!1\,|\,a\!=\!0)P(a\!=\!0)} && (2.16) \\ &= \frac{0.95 \times 0.01}{0.95 \times 0.01 + 0.05 \times 0.99} && (2.17) \\ &= 0.16. && (2.18) \end{aligned}$$

So in spite of the positive result, the probability that Jo has the disease is only 16%. □

▶ 2.2 The meaning of probability

Probabilities can be used in two ways.

Probabilities can describe *frequencies of outcomes in random experiments*, but giving noncircular definitions of the terms 'frequency' and 'random' is a challenge – what does it mean to say that the frequency of a tossed coin's

Notation. Let 'the degree of belief in proposition x' be denoted by $B(x)$. The negation of x (NOT-x) is written $\overline{x}$. The degree of belief in a conditional proposition, 'x, assuming proposition y to be true', is represented by $B(x \,|\, y)$.

Axiom 1. Degrees of belief can be ordered; if $B(x)$ is 'greater' than $B(y)$, and $B(y)$ is 'greater' than $B(z)$, then $B(x)$ is 'greater' than $B(z)$.

[Consequence: beliefs can be mapped onto real numbers.]

Axiom 2. The degree of belief in a proposition x and its negation $\overline{x}$ are related. There is a function f such that

$$B(x) = f[B(\overline{x})].$$

Axiom 3. The degree of belief in a conjunction of propositions x, y (x AND y) is related to the degree of belief in the conditional proposition $x \,|\, y$ and the degree of belief in the proposition y. There is a function g such that

$$B(x, y) = g\left[B(x \,|\, y), B(y)\right].$$

Box 2.4. The Cox axioms. If a set of beliefs satisfy these axioms then they can be mapped onto probabilities satisfying $P(\text{FALSE}) = 0$, $P(\text{TRUE}) = 1$, $0 \le P(x) \le 1$, and the rules of probability:

$$P(x) = 1 - P(\overline{x}),$$

and

$$P(x, y) = P(x \,|\, y)P(y).$$

coming up heads is $1/2$? If we say that this frequency is the average fraction of heads in long sequences, we have to define 'average'; and it is hard to define 'average' without using a word synonymous to probability! I will not attempt to cut this philosophical knot.

Probabilities can also be used, more generally, to describe *degrees of belief* in propositions that do not involve random variables – for example 'the probability that Mr. S. was the murderer of Mrs. S., given the evidence' (he either was or wasn't, and it's the jury's job to assess how probable it is that he was); 'the probability that Thomas Jefferson had a child by one of his slaves'; 'the probability that Shakespeare's plays were written by Francis Bacon'; or, to pick a modern-day example, 'the probability that a particular signature on a particular cheque is genuine'.

The man in the street is happy to use probabilities in both these ways, but some books on probability restrict probabilities to refer only to frequencies of outcomes in repeatable random experiments.

Nevertheless, degrees of belief *can* be mapped onto probabilities if they satisfy simple consistency rules known as the Cox axioms (Cox, 1946) (figure 2.4). Thus probabilities can be used to describe assumptions, and to describe inferences given those assumptions. The rules of probability ensure that if two people make the same assumptions and receive the same data then they will draw identical conclusions. This more general use of probability to quantify beliefs is known as the *Bayesian* viewpoint. It is also known as the *subjective* interpretation of probability, since the probabilities depend on assumptions. Advocates of a Bayesian approach to data modelling and pattern recognition do not view this subjectivity as a defect, since in their view,

> you cannot do inference without making assumptions.

In this book it will from time to time be taken for granted that a Bayesian approach makes sense, but the reader is warned that this is not yet a globally held view – the field of statistics was dominated for most of the 20th century by non-Bayesian methods in which probabilities are allowed to describe only random variables. The big difference between the two approaches is that

Bayesians also use probabilities to describe *inferences.*

▶ 2.3 Forward probabilities and inverse probabilities

Probability calculations often fall into one of two categories: *forward probability* and *inverse probability*. Here is an example of a forward probability problem:

Exercise 2.4.$^{[2,\ \text{p.40}]}$ An urn contains K balls, of which B are black and $W = K - B$ are white. Fred draws a ball at random from the urn and replaces it, N times.

(a) What is the probability distribution of the number of times a black ball is drawn, n_B?

(b) What is the expectation of n_B? What is the variance of n_B? What is the standard deviation of n_B? Give numerical answers for the cases $N = 5$ and $N = 400$, when $B = 2$ and $K = 10$.

Forward probability problems involve a *generative model* that describes a process that is assumed to give rise to some data; the task is to compute the probability distribution or expectation of some quantity that depends on the data. Here is another example of a forward probability problem:

Exercise 2.5.$^{[2,\ \text{p.40}]}$ An urn contains K balls, of which B are black and $W = K - B$ are white. We define the fraction $f_B \equiv B/K$. Fred draws N times from the urn, exactly as in exercise 2.4, obtaining n_B blacks, and computes the quantity

$$z = \frac{(n_B - f_B N)^2}{N f_B (1 - f_B)}. \tag{2.19}$$

What is the expectation of z? In the case $N = 5$ and $f_B = 1/5$, what is the probability distribution of z? What is the probability that $z < 1$? [Hint: compare z with the quantities computed in the previous exercise.]

Like forward probability problems, *inverse probability problems* involve a generative model of a process, but instead of computing the probability distribution of some quantity *produced* by the process, we compute the conditional probability of one or more of the *unobserved variables* in the process, *given* the observed variables. This invariably requires the use of Bayes' theorem.

Example 2.6. There are eleven urns labelled by $u \in \{0, 1, 2, \ldots, 10\}$, each containing ten balls. Urn u contains u black balls and $10 - u$ white balls. Fred selects an urn u at random and draws N times with replacement from that urn, obtaining n_B blacks and $N - n_B$ whites. Fred's friend, Bill, looks on. If after $N = 10$ draws $n_B = 3$ blacks have been drawn, what is the probability that the urn Fred is using is urn u, from Bill's point of view? (Bill doesn't know the value of u.)

Solution. The joint probability distribution of the random variables u and n_B can be written

$$P(u, n_B \,|\, N) = P(n_B \,|\, u, N) P(u). \tag{2.20}$$

From the joint probability of u and n_B, we can obtain the conditional distribution of u given n_B:

$$\begin{aligned} P(u \,|\, n_B, N) &= \frac{P(u, n_B \,|\, N)}{P(n_B \,|\, N)} && (2.21) \\ &= \frac{P(n_B \,|\, u, N) P(u)}{P(n_B \,|\, N)}. && (2.22) \end{aligned}$$

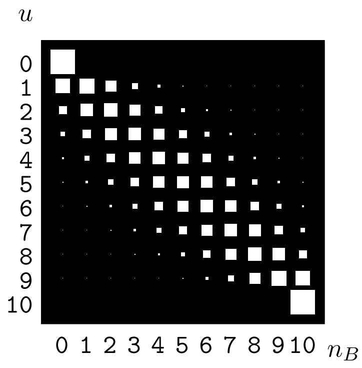


Figure 2.5. Joint probability of u and n_B for Bill and Fred's urn problem, after $N = 10$ draws.

The marginal probability of u is $P(u) = \frac{1}{11}$ for all u. You wrote down the probability of n_B given u and N, $P(n_B \mid u, N)$, when you solved exercise 2.4 (p.27). [You *are* doing the highly recommended exercises, aren't you?] If we define $f_u \equiv u/10$ then

$$P(n_B \mid u, N) = \binom{N}{n_B} f_u^{n_B} (1 - f_u)^{N - n_B}. \tag{2.23}$$

What about the denominator, $P(n_B \mid N)$? This is the marginal probability of n_B, which we can obtain using the sum rule:

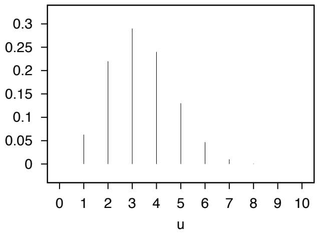
$$P(n_B \mid N) = \sum_u P(u, n_B \mid N) = \sum_u P(u) P(n_B \mid u, N). \tag{2.24}$$

So the conditional probability of u given n_B is

$$P(u \mid n_B, N) = \frac{P(u) P(n_B \mid u, N)}{P(n_B \mid N)} \tag{2.25}$$

$$= \frac{1}{P(n_B \mid N)} \frac{1}{11} \binom{N}{n_B} f_u^{n_B} (1 - f_u)^{N - n_B}. \tag{2.26}$$

This conditional distribution can be found by normalizing column 3 of figure 2.5 and is shown in figure 2.6. The normalizing constant, the marginal probability of n_B, is $P(n_B = 3 \mid N = 10) = 0.083$. The posterior probability (2.26) is correct for all u, including the end-points $u = 0$ and $u = 10$, where $f_u = 0$ and $f_u = 1$ respectively. The posterior probability that $u = 0$ given $n_B = 3$ is equal to zero, because if Fred were drawing from urn 0 it would be impossible for any black balls to be drawn. The posterior probability that $u = 10$ is also zero, because there are no white balls in that urn. The other hypotheses $u = 1$, $u = 2$, … $u = 9$ all have non-zero posterior probability. □



u	$P(u \mid n_B = 3, N)$
0	0
1	0.063
2	0.22
3	0.29
4	0.24
5	0.13
6	0.047
7	0.0099
8	0.00086
9	0.0000096
10	0

Figure 2.6. Conditional probability of u given $n_B = 3$ and $N = 10$.

Terminology of inverse probability

In inverse probability problems it is convenient to give names to the probabilities appearing in Bayes' theorem. In equation (2.25), we call the marginal probability $P(u)$ the *prior* probability of u, and $P(n_B \mid u, N)$ is called the *likelihood* of u. It is important to note that the terms likelihood and probability are not synonyms. The quantity $P(n_B \mid u, N)$ is a function of both n_B and u. For fixed u, $P(n_B \mid u, N)$ defines a *probability* over n_B. For fixed n_B, $P(n_B \mid u, N)$ defines the *likelihood* of u.