Chapter 1

Introduction

We live in a world where information increasingly exists in an electronic form, stored on computer systems which through the internet are accessible across the planet. Recent estimates suggest that there are over 11.5 billion pages in the indexable portion of the world wide web [30], and a number of projects are digitising large quantities of material which currently only exists in printed form [102] [101].

The availability of such large quantities of information presents opportunities and challenges. The work presented in this thesis covers three areas:

- **Acquisition of information:** This area includes topics such as text entry and speech recognition which are used to directly create new works, or areas such as optical character recognition which convert existing information into an electronic form.
- **Processing of information:** For example, interpretation, summarisation and searching for electronic resources.
- **Presentation of information:** When a user has requested information, how should it be presented to them?

The unifying theme will be the use of *probability theory* to perform these tasks.

1.1 Why probabilistic models?

The subjective interpretation of probability views probabilities as statements of *degrees of belief* in hypotheses [24]. Probability theory can be seen as an extension of classical Boolean logic [34] — a set of axioms for making consistent statements about our beliefs.

Probability theory allows simple models to be constructed which are able to capture many of the interesting properties of complicated systems. Most documents are the product of human thought, complex social interactions and the workings of large physical systems. Attempting to understand such documents by building models which mimic these processes is well beyond our current abilities, so the power afforded by simple probabilistic models is invaluable.

Perhaps an even more important advantage of probabilistic modelling is the possibility of *learning*. In other words the details of a model need not be fully specified when it is created and behaviour can be learned by example. In the probabilistic framework, learning is achieved by jointly modelling all observations. Statistical correlations are assumed to exist between what has been observed in the past and what will be observed in the future. This framework gives rise to the idea of *prior* and *posterior* distributions, the latter of which encapsulates current beliefs in the underlying operation of the world, and can be updated using Bayes' theorem.

1.2 Fundamentals of probability theory

Let x be a random variable. The probability distribution over x is written $\Pr(x)$, which is non-negative for all x and is normalised so that

$$\int \mathrm{d}x \Pr(x) = 1 \tag{1.1}$$

Two random variables, x and y can be expressed in terms of a *joint probability distribution*, $\Pr(x, y)$, which can encapsulate any correlations between their values. Given such a distribution, the marginal distribution over x is given by

$$\Pr(x) = \int \mathrm{d}y \Pr(x, y) \tag{1.2}$$

with an equivalent definition for the marginal distribution over y. The conditional probability distribution over x when y is observed is given by

$$\Pr(x \mid y) = \frac{\Pr(x, y)}{\Pr(y)} \tag{1.3}$$

If $\Pr(x \mid y) = \Pr(x)$ then x and y are said to be independent. The joint probability can be decomposed in two ways,

$$\Pr(x, y) = \Pr(x \mid y) \cdot \Pr(y) \tag{1.4}$$

and

$$\Pr(x, y) = \Pr(y \mid x) \cdot \Pr(x) \tag{1.5}$$

Equating these gives

$$\Pr(x \mid y) \cdot \Pr(y) = \Pr(y \mid x) \cdot \Pr(x) \tag{1.6}$$

$$\Pr(x \mid y) = \frac{\Pr(y \mid x) \cdot \Pr(x)}{\Pr(y)} \tag{1.7}$$

Which is Bayes' famous rule. One particularly useful application is the case where multiple observations, $\{x_i\}_{i=1}^N$ are jointly distributed with some parameters, y. In this case,

$$\Pr(x_N \mid x_1, \ldots, x_{N-1}) = \int \mathrm{d}y \Pr(x_N, y \mid x_1, \ldots, x_{N-1}) \quad (1.8)$$

$$= \int \mathrm{d}y \Pr(x_N \mid y, x_1, \ldots, x_{N-1}) \cdot \Pr(y \mid x_1, \ldots, x_{N-1}) \quad (1.9)$$

If $\{x_i\}_{i=1}^N$ are independent given y, then

$$\Pr(x_N \mid y, x_1, \ldots, x_{N-1}) = \Pr(x_N \mid y) \quad (1.10)$$

which gives

$$\Pr(x_N \mid x_1, \ldots, x_{N-1}) = \int \mathrm{d}y \Pr(x_N \mid y) \cdot \Pr(y \mid x_1, \ldots, x_{N-1}) \quad (1.11)$$

De Finetti's theorem states that this conditional independence assumption holds if and only if $\{x\}$ are exchangeable, that is, if their probability is invariant under permutation. Equation (1.11) is the basis of Bayesian machine learning. The term $\Pr(y \mid x_1, \ldots, x_{N-1})$ is the *posterior* distribution over y given the data seen so far, and can be calculated using Bayes' theorem,

$$\Pr(y \mid x_1, \ldots, x_{N-1}) = \frac{\Pr(x_1, \ldots, x_{N-1} \mid y) \cdot \Pr(y)}{\Pr(x_1, \ldots, x_{N-1})} \quad (1.12)$$

where $\Pr(x_1, \ldots, x_{N-1} \mid y)$ is the *likelihood* and $\Pr(y)$ is the *prior*. Intuitively speaking, the prior represents our belief about y before any data is seen, and the posterior represents the belief after seeing the data. As more data is observed, the posterior will typically become tighter as more is learnt.

1.3 Applications of document modelling

The subject of this thesis is the creation and application of models of documents expressed in terms of probability distributions. There are many areas in which this approach has been shown to be fruitful, some of which are briefly introduced in the following sections.

1.3.1 Language modelling

The majority of the work in this thesis concerns plain text documents, with the aim being to produce language models consisting of probability distributions over strings of tokens. These distributions represent our belief in the likely content of newly observed documents. Chapters 2 and 3 of this thesis consider language modelling from a Bayesian viewpoint.

Application	Observation	Source
Speech recognition	Acoustic signal	Spoken text
Handwriting recognition	Digitised pen strokes or bitmap image	Written text
Optical character recognition	Bitmap image	Written text
Automatic translation	Text in source language	Text in target language
Spelling correction	Mis-spelt word	Correct spelling
Ambiguous text entry	Keystroke sequence	Entered text

Table 1.1: Natural language applications of channel modelling.

Channel models

Consider a noisy communications channel, which transmits some stochastically generated source signal $\boldsymbol{y}$. An observation, $\boldsymbol{x}$, is made of the resulting output of the channel, with the goal being to reconstruct $\boldsymbol{y}$. In order to perform such a reconstruction it is necessary both to have a model of the channel, representing beliefs in the transformations which may have occurred, and a model of the source.

Formally speaking, the posterior distribution over $\boldsymbol{y}$ can be found using Bayes' rule,

$$\Pr(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{\Pr(\boldsymbol{x} \mid \boldsymbol{y}) \Pr(\boldsymbol{y})}{\Pr(\boldsymbol{x})} \tag{1.13}$$

$\Pr(\boldsymbol{x} \mid \boldsymbol{y})$ is the model of the channel, which is application dependent. $\Pr(\boldsymbol{y})$ is the prior distribution, representing our belief in the input in the absence of any observations.

This framework may be applied to a number of problems involving natural languages, for example speech recognition [36] [70], handwriting recognition [69], optical character recognition, automatic translation [15], spelling correction [41] and ambiguous text entry. Examples of the channels which are used are given in Table 1.1. In all of these applications the source is a string in a natural language, so the prior distribution is a language model.

Data compression

Data compression is the task of converting input strings into codewords with the aim of reducing the average length of transmitted information. Arithmetic coding makes use of a probabilistic model of input strings, $\hat{p}(x)$, giving an estimate of the probability that x will be selected [98]. By ordering the possible strings in dictionary order, this model can be used to assign non-overlapping intervals on the real line $[0, 1)$.

To specify codewords, intervals on the line are expressed as strings of binary digits. Treating the strings as binary fractions after the decimal point, .0 and .1 divide the line in half, .00, .01, .10 and .11 divide it into quarters and so on. After n binary digits the line is divided

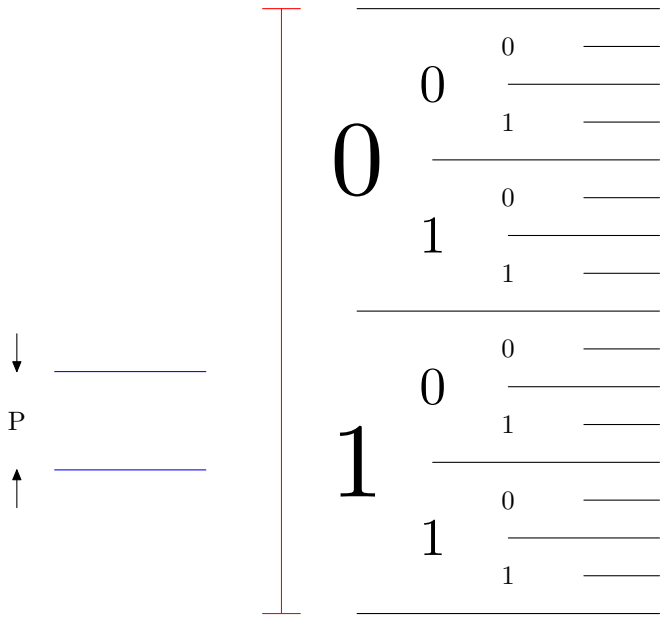

Figure 1.1: An illustration of arithmetic coding. The interval shown in blue on the left hand side corresponds to the string being coded, which has probability P. The right hand side shows the division of the real line (shown in red) into intervals specified by binary codewords. The codeword is the largest binary interval entirely enclosed by the string, in this case 101.

into 2^n intervals of length 2^{-n}. The codeword is the shortest string which is entirely enclosed by the interval defined by the input. Figure 1.1 illustrates the process.

Assuming that the interval corresponding to the string being compressed, x, is exactly aligned with the binary code word intervals,

$$2^{-n} = \hat{p}(x) \tag{1.14}$$

$$n = -\log_2(\hat{p}(x)) \tag{1.15}$$

Taking the expectation with respect to the true distribution over inputs gives

$$\langle n \rangle = -\sum_x p(x) \log_2(\hat{p}(x)) \tag{1.16}$$

where the notation $\langle \cdots \rangle$ is used to represent the expected value of a random variable. If the model exactly matches the source distribution, this expression is equal to the entropy of the source [56], which is the theoretical limit on the compression which can be achieved.

In practice the intervals will not precisely align with the input strings. By considering the worst case, it can be shown that the maximum overhead is 2 bits. As this result holds for arbitrarily long strings, arithmetic compression is able to come very close to ideal performance.

Arithmetic coding essentially reduces the problem of data compression to that of source modelling. If the data to be compressed is a plain text document then the source model is a

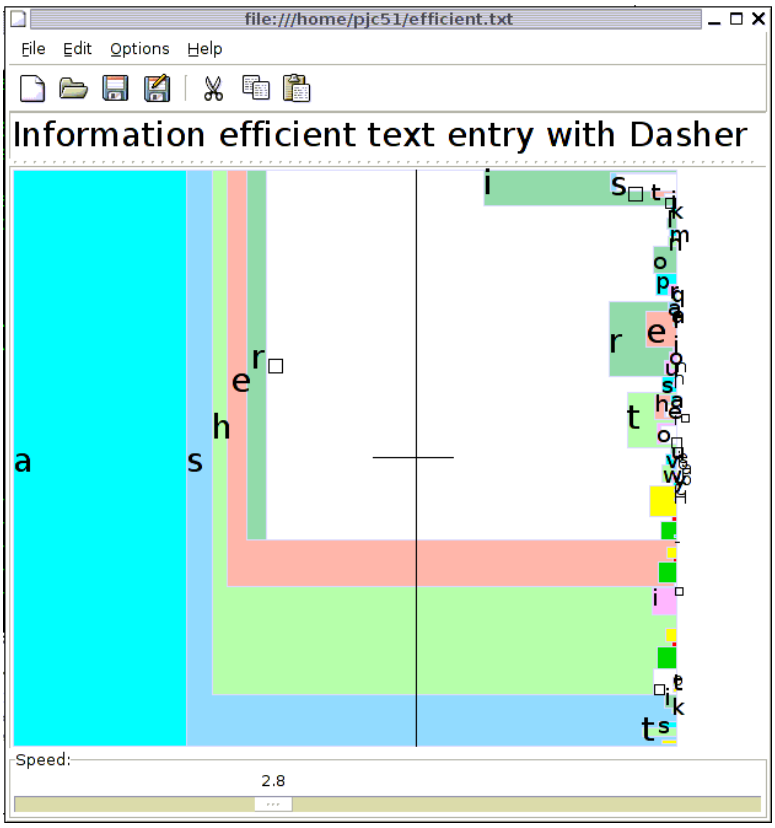

Figure 1.2: The Dasher text entry system. Possible strings are represented as a sequence of nested boxes, with the size of each box being proportional to the conditional probability of each symbol given the parent. The display zooms continuously towards a point on the real line indicated by the user, resulting in the selection of the string corresponding to that point. As a language model is used, more probable strings represent larger intervals on the line, and therefore can be entered more rapidly.

language model of the sort considered in this thesis [6].

Dasher

Dasher is a text entry system originally developed by Ward and MacKay [96] [105] [95]. A screen capture of the Dasher interface is shown in Figure 1.2.

Dasher is based on arithmetic decompression. The Dasher interface evolves dynamically to zoom in on a point on the line, which is equivalent to specifying a codeword. A probabilistic language model is used to model the input strings, which are represented by a series of nested boxes adjoining the line. The user of Dasher can therefore navigate one symbol at a time, with more probable strings being assigned more screen space and therefore being easier to enter.

Dasher is targeted primarily at two areas: firstly, text entry on small devices such as Personal Digital Assistants (PDAs) which are too small to include a usable full size keyboard,

and secondly as a communication aid for people who are prevented from using a normal keyboard by disability. Various adaptations have been used to enable input from eye or head tracking [97], one dimensional input devices such as breath controllers, and button input. Dasher has been adapted for use in a large number of different languages, and facilities exist for using Dasher to control other applications running on the computer. Dasher is also the basis for a novel multi-modal text entry system which combines speech recognition [93]. Dasher is Free software, available under the GNU General Public License. Chapter 5 considers a new use for Dasher — as an efficient method for performing search.

Document classification

The goal of document classification is to group documents which belong to the same class. In many cases the number of classes is fixed and is specified in advance. Let $\{\mathcal{H}_i\}$ be a set of hypotheses concerning the correct classification of a document. By producing a model for the data conditioned on each of these hypotheses, Bayes' rule can be used to compute the posterior probability of class membership,

$$\Pr\left(\mathcal{H}_i \mid \boldsymbol{x}\right) \propto \Pr\left(\boldsymbol{x} \mid \mathcal{H}_i\right) \cdot \Pr\left(\mathcal{H}_i\right) \tag{1.17}$$

Again, we have introduced here a prior, $\Pr\left(\mathcal{H}_i\right)$, on class membership. By comparing the probabilities of membership of each class, decision theory can be used to optimally classify the document.

A topical example of an application of this technique is the task of identifying unsolicited bulk e-mail messages (spam) [80]. By training a model on a large number of examples, reliable classification can be achieved. Further information can be included by adding a vector of features, for example representing details of the message headers. These can simply be treated as part of the document text and modelled in the same way. A typical approach is to use a naïve Bayes classifier, which models documents as 'bags of words', ignoring correlations between terms.

Information retrieval

The task of information retrieval is to search a collection of documents for those which are relevant to a query supplied by the user. A variety of methods have been used for information retrieval, including many which do not make use of probabilistic methods. However, recent research has re-cast the problem in terms of document modelling. Information retrieval has become a significant area for the application of document modelling techniques. Search engines such as Google [107], Yahoo! [114] and MSN Search [110] are now the first port of call for many web users. Information retrieval as an application of language modelling is considered in Chapter 4.

1.3.2 Multimedia documents

Although language modelling is an important aspect of document modelling, it is not the whole story. Many documents contain information beyond that which is contained in the text itself, ranging from the layout of text on the page, through graphical elements to audio and video content.

A particular application beyond language modelling which is considered in this thesis is the analysis of hand-drawn electronic ink diagrams. These diagrams are typically produced using a system which incorporates a pen-based interface, such as a Table PC or a Personal Digital Assistant. The task which will be addressed is the process of identifying perceptually relevant objects in these diagrams. This analysis potentially permits more natural interaction when editing the diagram and allows automatic construction of a more formally laid-out version of a rough sketch. Chapter 6 considers this problem, developing a model based on the conditional random field [48] which is able to simultaneously group ink fragments and label the objects which they represent.