Chapter 1

Summary

1.1 The need for Occam's razor

There are countless problems in science, statistics and technology which require that, given a limited data set, preferences be assigned to alternative models of differing complexities. For example, two alternative hypotheses accounting for planetary motion are the geocentric 'epicyclic' model, and the simpler Copernican model of the solar system. In the less theologically contentious but similar problem of fitting a curve to data, alternative models assign different functional forms to the curve, for example 'a linear function with two free parameters', 'a quadratic with three', or 'a cubic function with four parameters'. It would be nice if we could just rank the models by how well they 'fit' the data, but it is a familiar difficulty that a more complex model typically fits the data better: when we fit a curve to data, a quadratic curve with three parameters can always fit the data better than a linear model with two parameters, and a polynomial with a hundred terms fits the data even better; preferring the 'best fit' model leads us to choose implausibly detailed and over–parameterised models, which interpolate and generalise poorly. 'Occam's razor' is the principle that states that unnecessarily complex models should not be preferred to simpler ones. How can we quantify this intuitive principle so as to make it an objective part of our modelling method?

Bayesian probability theory provides a framework for inductive inference which has been called 'common sense reduced to calculation'; it is a poorly known fact that Bayesian methods actually embody Occam's razor automatically and quantitatively [26, 38]. Bayesian model comparison is the central theme of this thesis. In particular, the power of the Bayesian Occam's razor is demonstrated on 'neural networks'. Neural networks are novel modelling tools capable of 'learning from examples'. These currently popular models are notorious for their lack of an objective grounding; the main goal of this thesis is to provide an objective and practical framework for the use of neural network techniques by applying the methods of Bayesian model comparison. In the process several enhancements to current neural network methods arise.

1.2 What is Bayesian modelling?

Bayesian methods for inductive inference were first developed in detail early this century by the Cambridge geophysicist, Sir Harold Jeffreys [38]. At that time, Jeffreys' ideas were opposed by Fisher and others, and since then a debate has persisted between the 'orthodox' view of statistics and the minority Bayesian camp. I will not dwell here on the details of

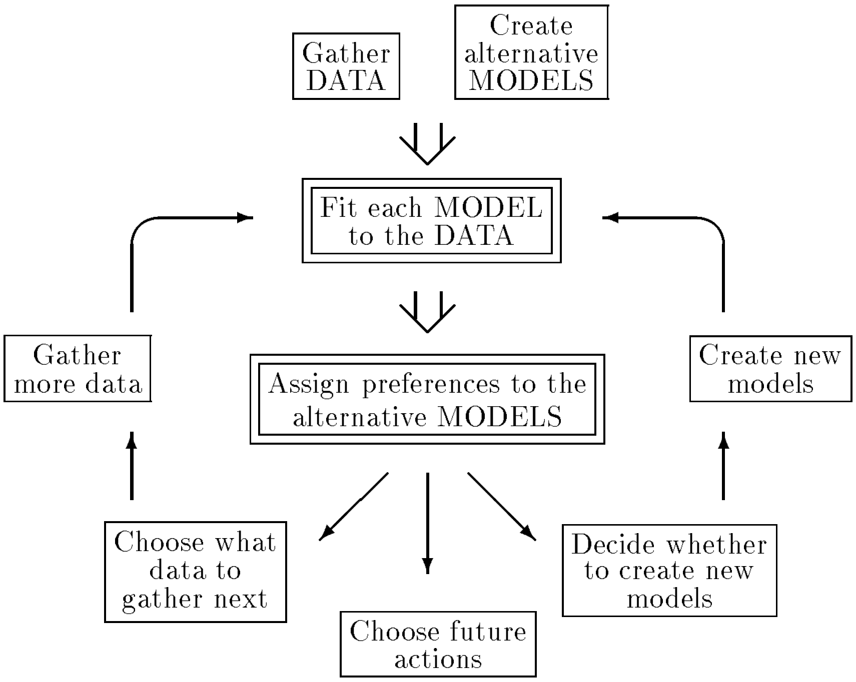

Figure 1.1: **Abstraction of the data modelling process.**
The two central boxes are the *inference* steps, where Bayesian methods can be applied.

the philosophical argument, which goes deep down to the meaning of a probability [17, 36]; rather, this thesis will demonstrate that it is possible using Bayesian methods to solve problems in neural networks which have otherwise been found laborious or impossible. Since the 1960's, the Bayesian minority has been steadily growing, especially in the fields of economics [89] and pattern processing [20]. At this time, the state of the art for the problem of speech recognition is a Bayesian technique (Hidden Markov Models), and the best image reconstruction algorithms are also based on Bayesian probability theory (Maximum Entropy), but Bayesian methods are still viewed with mistrust by the orthodox statistics community; the framework for model comparison is especially poorly known, even to most people who call themselves Bayesians. This thesis therefore takes some time to thoroughly review the flavour of Bayesianism that I am using. To some, the word Bayesian denotes a decision strategy that minimises the expectation of a cost [24]; to others, a Bayesian is someone who tries to incorporate prior knowledge into their inference and decision process [8]. In fact, according to Good, there are 46656 varieties of Bayesian![1] This thesis presents a flavour of Bayesianism in which decisions are not involved. Inference and decision are cleanly separated. The terms 'Bayes risk' and 'Bayes optimal' are not in the vocabulary of this thesis. The genealogy of this flavour is Laplace–Jeffreys–Cox–Jaynes–Gull [80, 38, 17, 36, 26]. A further difference between this approach and other work known as Bayesian is that the emphasis is on inverse rather than forward probability. Forward probability uses probabilities and priors, but it does not make use of Bayes' rule. Forward probability is used for example to evaluate the typical performance of a modelling procedure averaged over different data sets from a defined ensemble [82, 32]. Here the philosophy is, using inverse probability, to evaluate the relative plausibilities of several alternative *models* in the light of the *single* data set that we actually observe.

[1] Good was unaware of the Bayesian Occam's razor.

Where inference fits into the data modelling process

Figure 1.1 illustrates an abstraction of the data modelling process; this summary applies for example to the tasks of fitting a curve to data, reconstructing a blurred image, and making an automatic pattern recognition system; the figure is also descriptive of the general scientific method.

We start by gathering data and creating models to account for those data. There are then two levels of *inference*, which are marked by the double–framed boxes. At the first level, 'fitting each model to the data', the task is to infer what the free parameters of each model might be given the data. The second level of inference is the task of model comparison. Here, we wish to rank how plausible the alternative models are in the light of the data.

Having fitted the models and compared them, we can then decide to gather more data or to invent new models for the data, and we can repeat the inference process. We can also use the knowledge we have gained from the data to make decisions about our future actions in the world.

Bayesian methods can be used to solve the two inductive inference problems, which are the two central boxes in the figure; the other tasks in the modelling process are not directly addressed by Bayes' rule, which applies to inductive inference problems only. The first level of inference, fitting each model to the data, is usually a straightforward task, and differences between Bayesian and non–Bayesian solutions are often not pronounced at this level. This thesis will especially emphasise the second level of inference, the task of model comparison. This inference problem is not straightforward because a quantitative Occam's razor is needed to penalise over–complex models. The other boxes in this diagram will also be visited during the thesis.

Bayes' rule

The fundamental concept of Bayesian analysis is that the plausibilities of alternative hypotheses are represented by probabilities, and inference is performed by evaluating those probabilities. Suppose that we have a collection of models, $\mathcal{H}_1, \mathcal{H}_2, \ldots \mathcal{H}_L$ competing to account for the data we gather. Our initial beliefs about the relative plausibility of these models are quantified by a list of probabilities, $P(\mathcal{H}_1), P(\mathcal{H}_2), \ldots P(\mathcal{H}_L)$, which sum to 1. Each model $\mathcal{H}_i$ makes predictions about how likely different data sets 'D' are, if that model is true. These predictions are described by a probability distribution $P(D|\mathcal{H}_i)$ ('the probability of D *given* $\mathcal{H}_i$'). When we observe the actual data D, **Bayes' rule** describes how we should update our beliefs in the models in the light of the data. The plausibility of model $\mathcal{H}_i$, given that we have observed D, written $P(\mathcal{H}_i|D)$, is obtained by multiplying together two quantities: first, $P(\mathcal{H}_i)$, *i.e.*, how plausible we thought $\mathcal{H}_i$ was *before* the data arrived; and second, $P(D|\mathcal{H}_i)$, *i.e.*, how much the model $\mathcal{H}_i$ *predicted* the data. In symbols, Bayes' rule is written:

$$P(\mathcal{H}_i|D) = \frac{P(\mathcal{H}_i)P(D|\mathcal{H}_i)}{P(D)}.$$

The denominator $P(D)$ is a normalising constant which makes our final beliefs $P(\mathcal{H}_i|D)$ add up to 1. A Bayesian addresses any inference problem by using this equation. The hard line Bayesian position is that the Cox axioms [17] prove that consistent inference can *only* be Bayesian, and no other inference methods should be used, on pain of inconsistency [75]. However, I will develop the more moderate position that the Bayesian method is an important tool which should be used alongside other pragmatic modelling tools. I will demonstrate that the simultaneous application of Bayesian and non–Bayesian methods leads to insights that could not be obtained by using either tool alone.

1.3 What are neural networks and why do they need Occam's razor?

Research in neural–like networks is motivated by the observation that the brain has a 'connectionist' computational architecture: the brain is composed of many simple devices (neurons) which are massively interconnected with each other; the computational abilities of the brain are an 'emergent phenomenon' arising from the cooperative interactions of these simple components. Workers in the field of neural networks create novel connectionist devices so as to try and understand 'how the brain does it', and to try to create new and useful tools for such tasks as speech recognition, character recognition, and robotics.

The most popular neural network algorithm is 'backpropagation', which is capable of 'learning from examples' [66]. In this case, a neural network can be viewed as a black box which produces an output when we give it an input. How the output depends on the input is controlled by some tens or thousands of knobs on the black box, which we, the teacher, are able to twiddle. The object of the 'learning' process is to adjust these knobs so as to get the black box to give a desired output in response to each input. What is inside the black box is not essential to this discussion: usually it contains a network of simple 'neurons' feeding from the inputs to the outputs, and the 'knobs' are the strengths of the 'synapses' between the 'neurons'.

Imagine that what we feed to the inputs of the black box is a simple encoding of a piece of English text; and imagine that we want the outputs of the black box to be the pronunciation of that piece of text, in a simple code we have defined. When we present an untrained black box with a piece of English text, its outputs are very likely to be complete garbage, compared with the coded pronunciation that we wanted it to produce. What we would like to do is adjust the knobs on the black box a little, so that the next time we give the same piece of text as input, the output of the box will be *a little closer* to what it should have been. The backpropagation learning algorithm is a prescription for how to tweak all the knobs on the black box to achieve precisely this goal. (Backpropagation performs gradient descent on the error function.) Now the perhaps surprising outcome of this procedure is that after repeated 'training' on a dictionary of 50,000 English words, a black box consisting of 200 'neurons' can learn not only to pronounce correctly a large fraction of the words it was trained on, but also to perform equally well on other words which were not in the training set. Thus the device is able to extract the underlying structure in the examples it was trained on and 'generalise' from them.

The backpropagation algorithm has been applied to many other tasks (the text pronunciation example above is one of the earliest successes), and a performance equalling the ability of human experts is often obtained. Recently, especially impressive results have been obtained for adaptive optics [4].

However, the performance of these algorithms depends on a considerable number of design choices, most of which are currently made by rules of thumb and trial and error. For example, in designing the neural network for text pronunciation, one has to decide how many 'neurons' there should be in the architecture of the black box, how they should be connected to each other, and what constraints should be imposed on the parameters of the network. The problem of Occam's razor rears its head repeatedly when we try to make these design choices because a more complex and unconstrained neural network will nearly always learn the examples in the training set better than a simpler one; however the simpler neural network may actually be a better model of the problem, and generalise better to new examples.

The fact that we cannot use the performance on the training set to choose between different solutions would not matter if we had plenty of data and limitless computational resources: we could generate solutions using thousands of different models with different complexities and rank them by evaluating the test error on some reserved test data. But since we have limited resources we would like to be able to use *all* our data both to fit all the models and also to rank them. We would furthermore like to find a technique for automatically optimising the choice of model design, without having to perform massive computational searches through 'design space'.

The Bayesian framework presented in this thesis satisfies these desiderata.

Overview

The thesis consists of four papers. The first paper (Chapter 2) reviews in detail the Bayesian framework for model comparison and regularisation due to Gull and Skilling, by studying the problem of interpolating a noisy data set with traditional linear models. This chapter demonstrates that Bayesian methods do indeed embody Occam's razor in a consistent, intuitive and quantitative way.

In the second paper (Chapter 3) this framework is applied to 'neural networks', and it is demonstrated that (at least for the toy problem studied) Bayesian probability theory chooses between alternative solutions found using networks with different architectures in a way that succesfully embodies Occam's razor. Another enhancement to neural network training methods concerns regularisation. Neural networks sometimes perform poorly because the parameters ('weights') in the network blow up to implausibly large values in order to fit the details of the training set. To prevent this it is popular to use a procedure called 'weight decay' during the training. However, no objective procedure previously existed for setting the weight decay *rate* (apart from the computationally expensive option of testing multiple decay rates in parallel experiments). The Bayesian framework for neural network learning yields a simple prescription for optimising the weight decay rate, which is interpreted as a regularisation constant. This prescription can be easily approximated and implemented 'on line', and it may be one of the most useful practical tools to emerge from this research. In Chapter 3 we also see how the combination of Bayesian and non–Bayesian model assessment techniques can draw attention to defects in our hypothesis space, helping us traverse the loop to the right of figure 1.1, in which we invent new models.

In the third paper (Chapter 4), information–based utility functions are discussed for the purpose of data selection, the left–hand loop in figure 1.1. The evaluation of data utility is a problem relevant to a scientist whose data measurements are expensive, and to an autonomous robot which has to decide where to explore next so as to satisfy a pre–programmed curiosity about its environment; we also need to evaluate data utility in situations where data is so abundant that we have to decide which data to throw away. The information–based criteria derived in this chapter have promising properties, but I do not believe that they are the final solution to the data selection problem, because artefacts may result when these criteria are applied to poor models.

The fourth paper (Chapter 5) applies the methods developed in the first three papers to neural networks solving classification problems, rather than regression problems. One of the simplest but most important results in this chapter is a demonstration that careful incorporation of error bar information into the outputs of a classifier can give improved predictions. As in Chapter 3, the Bayesian Occam's razor does its job surprisingly well.

Chapter 6 is a short note extending the framework of Chapters 2 and 3 to allow modelling of an input–dependent noise level. A maximum likelihood solution to this problem would have singularities where the interpolant fits the data exactly; the Bayesian solution naturally avoids these problems.

In the final chapter I reflect on the strengths and weaknesses of the Bayesian approach to adaptive modelling, and the open questions and frontiers facing this framework.

Relevance to Biology

This work is not intended to shed any direct light on the functioning of biological neural networks. But it is clear that biological neural networks have solved the Occam's razor problem — we are expert adaptive modelling systems. I believe that if we are ever to understand the brain, a prerequisite will be that we should understand the problems that it has solved. We need to understand how to model, and how to infer. Of course, I do not expect that the brain embodies any of the equations in this thesis; I am sure that Nature has found far more elegant solutions to these problems. But I hope that the Bayesian normative theory of learning will serve as a guide in trying to elucidate how learning is performed by natural systems.