Chapter 3

Statistical Ideas

In chapter one I explained that there were two parts to Pattern Recognition; the first was finding a suitable system of *measurement*, so that each object got turned into a vector, a point in $\mathbb{R}^n$. The second part consisted of comparing the results of doing this on a new object, with some set of data where we know the category to which the object belongs, thus comparing vectors with vectors.

The second chapter looked at ways of measuring objects- mostly pixel arrays, since these arise from pointing cameras at the world. In this chapter and the next, I shall start on the task of doing the actual recognition using statistical methods. In the following chapter I shall discuss neural nets, and in later chapters I shall treat syntactic methods.

We assume then that the robot has sensed the world, maybe by pointing a camera at it, possibly by other means, and that some measuring process has been applied to obtain a vector of real numbers. The robot has seen other such vectors in the past and has also been told what category of object they describe: now it has to make up its own mind about the latest vector. An industrial robot looking at something to be sorted, or possibly welded, may easily be in this situation.

I find myself, yet again, in a difficult position. It is not hard to give recipes which are easy to implement in programs, and to pass lightly over the issues concerning why they work, or indeed if they work. This goes against the grain; it is necessary to give a survey of what is standard practice, but it is also necessary to look at standard practice with a cold, critical and careful eye if there is to be any progress. The situation is particularly acute in Statistics. Statistics is often taught as recipes, with the deplorable consequences I have mentioned in chapter one, and it is about time this repellent habit was abandoned.

On the other hand, sorting out fundamental ideas in Probability theory carefully is a very technical business indeed, and many issues are still in dispute. So this chapter starts off somewhere between Scylla and Charybdis, trying to get fundamental ideas examined in informal language. This may show more courage than judgement; you can decide whether or not you feel illuminated.

I shall in this chapter, then, go into foundational issues with more enthusiasm than is common in Pattern Recognition texts. When we come to investigate Syntactic Pattern Recognition we shall come up against conceptual difficulties related to the kinds of models of neural processing which these methods imply. We might as well start the way we intend to continue and clear away the undergrowth from the beginning. In the spirit of critical reflection which I propose to stimulate, I shall draw attention to the underlying assumptions and conceptual underpinnings of contemporary practice. It is hard to change your own ideas, and even harder if you don't know what they are. Those who feel uncomfortable with theoretical issues should appreciate that there is nothing so practical as a good theory, and nothing so dangerous as a bad one.

The present chapter then is about the general ideas of statistics and probability theory which have a bearing on pattern recognition; the next chapter will give the recipes. I have broken the material up in this way because painful experience has compelled me to recognise that many people who claim to be of a practical disposition object to generalities and find ideas confusing. 'Don't give me all that blather, just give me the formulae', they say. Thought hurts their heads and gives them doubts about their certainties, so they want nothing to do with it. I have taken pity on them; this chapter therefore, may be skipped by those for whom thinking is an unnatural act. They may, of course, regret the decision later.

3.1 History, and Deep Philosophical Stuff

3.1.1 The Origins of Probability: random variables

Probability Theory originated in attempts to answer the practical questions of whether it is better to discard one card to fill a straight or two to fill a flush, and similar questions regarding dice. No branch of Applied Mathematics has better credentials as a practical and necessary part of a gentleman's education. If the theory worked, the customer made a buck, and if it didn't he tried to get his losses back from the theoretician; seldom has the motivation for honest theorising been so pointed.

The problem referred to arises in draw poker where you are one of some number of players, say four for definiteness, who have each been dealt five cards. You look at your hand and see, let us suppose, $\heartsuit 4, 5, 6; \clubsuit 7; \spadesuit Q$. You are allowed to discard one or two cards and get replacements off the top of what is left of the pack. The aim is, as all but the youngest are aware, to get a 'good' hand. Two possibilities immediately suggest themselves: to discard the Queen of Spades and hope to fill the straight by picking either a 3 or an 8; or to discard the club and spade and draw two cards in the hope of filling the flush by getting two more hearts.

The argument proceeds as follows: There are 47 cards not in my hand, and we know nothing about them beyond the plausible assumption that they are from a standard pack, and in a random order- whatever that means. In which case the 10 hearts left are somewhere in those

47 cards you haven't got. So there is a probability of 10/47 that the next card on the stack is a heart, and of 9/46 that the card after that is also a heart. So the probability of getting your flush is 90/2162, approximately 0.0416. If you aim for the straight, there are eight possible cards which can do the job, a 3 or 8 of any suit will do, and there are 47 left, so the probability of the top card being acceptable is 8/47 or approximately 0.1702. In rough terms, its about a one in six shot that you'll fill the straight, and only about one in twenty four that you'll fill the flush.

Which suggests you should go for the straight, but a flush beats a straight, so maybe you should go for the flush and improve your chances of winning more money. Who is to tell?

Now the analysis is very standard but contains a lot of assumptions, such as that the deck is not rigged and the other players will not shoot you under the table if you win. As everybody who has played poker knows, there may be more useful information in the momentary smug look on the face opposite than in the above calculation. The assumptions are not usually much thought about, which is why, perhaps, probabilists are not noticeably richer than the rest of us. [1]

The great Australian contribution to gambling is the game of *two-up*, which consists of throwing two coins in the air. If they land with the same face showing, two heads or two tails, then one party wins, otherwise he doesn't. While they are in the air, an attempt is made to affect the outcome by shouting such advice as '*Come in Spinner*' at the coins. Contrary to what one might expect, this works. But only about half the time.

Now there is nothing much random about the next two cards on the pack, either they are both hearts or they aren't, and likewise it is rather strange that throwing coins in the air should have anything random

[1] It is rumored that Probabilists are good people with whom to play poker, because they focus on the probabilities and not the personalities. I am unable to confirm this from personal experience as none of the Probabilists I know will play poker with me. I can't imagine why.

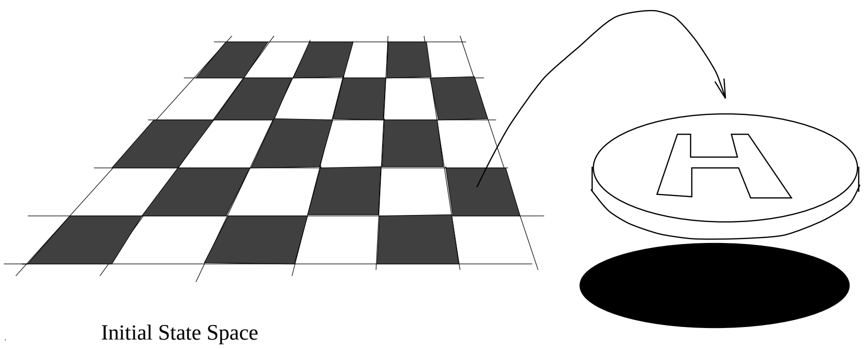

Figure 3.1: A random variable.

about it, since Newton's laws apply and the initial state determines the final state completely. The fact is, we don't have much useful information about the initial state, and the dynamical calculations on bouncing are somewhat beyond most players. Persi Diaconis, a statistician who started academic life somewhat non-canonically (much to the joy of his students and most of his colleagues), can reliably toss a coin so as to produce whatever outcome he wants. Score one for Isaac and Persi. But most of us will take it that initial states which are, to us, indistinguishable, can and do produce different outcomes.

Now suppose we take the space of initial positions, orientations, momenta and angular momenta for the coin and assign, in accordance with Newton's laws, a colour to each point in this space, black if the coin comes down heads and white if it comes down tails. Then the twelve dimensional space of initial states is divided up into a sort of checker-board, alternate black and white regions, separated by rather thin bits corresponding to the cases where the coin finishes on its edge. We cannot say much about the actual initial state, because the regions of black and white are so small that we see only a grey blur; but symmetry arguments lead us to the belief that there is as much black as there is white. The hypervolume or *measure* of the black points is equal to the measure of the white points. If we take the total measure to be 1, for convenience only, then we can assign the measure of the Heads outcome as being pretty much 1/2- assuming, in the face of the evidence, that the coin is truly symmetric. We summarise this by saying that if the coin is fair then the probability of getting heads is 0.5. And what this means is that the measure of all the regions in which the coin starts its throw which will lead by the inexorable action of natural law to its coming up Heads, is one half.

Intuitively, in the case of a finite set, the measure of the set is proportional to the number of elements in it; in the case of a subset of the real line, the measure of a set is the length of the set; in two dimensions it is the set's area, in three its volume. It is possible to abstract a collection of properties that all these things have in common, and then we list these properties and call them the axioms for a *measure space*. Then anything having these properties is an example, including the ones we started with. This is exactly like taking out the common properties of $\mathbb{R}^n$ and calling them the axioms for a Vector Space; it is standard Mathematical practice and allows us to focus our minds on the essentials. We thereupon forget about the dimension of any space of initial states, and indeed almost all properties of it, except the fact that it has some collection of subsets each of which has a measure, and that it has total measure 1. The unit square in the plane is simply a particular example of such a space.

It is thought along these lines which leads us to the idea of a *random variable* (rv). We imagine a map from some space (of initial conditions?) to a space of outcomes, and we suppose that nothing is known about the domain of the map except the measure of the set which produces each type of outcome. At least, this is what we assume it means if there are only finitely many outcomes; if there is a continuum of outcomes, as when someone throws a dart at a board, we suppose that there is a measure for any measurable set in which the dart might land. By putting sensible conditions on what a measure ought to be, making it behave in the same way as area, volume, *et cetera*, and similar conditions on how maps ought to treat measures, we come out with a definition of a random variable as a map from some inscrutable space having a measure and nothing much else, into $\mathbb{R}^n$ for some n, sometimes with restrictions such as taking values only in 0,1 with $n = 1$. Now we can define an *event* as a set of possible outcomes, and the *probability of an event A, p(A)*, to be the measure of all those points in the domain of the *rv* which lead to us getting an outcome in A. This is why probability behaves mysteriously like area of sets, and why those little diagrams of circles intersecting, which figure in the elementary texts on Probability Theory, actually work.

In short we imagine that any random variable is much like throwing a die or a dart. This imagery keeps probabilists going long after the logic has run dry.

Similarly, your ignorance of the state of the stack of cards from which you will be dealt the next one or two you ask for, is, we shall assume, close to total. Of all the possible arrangements in the stack of cards left, you deem them equally likely, by virtue of your innocent faith in the shuffling process to which the pack was subjected. To obtain the given 'probabilities' by the arithmetic process is a trivial combinatorial argument which most people find convincing, both as providing some relative degree of belief in the two outcomes, and also an estimate of what might happen if you were in this situation a large number of times and kept count of the results.

Experiments have been done on packs of cards to see if the results are what the arguments predict. Some anomalies have been reported, but mainly by Psychic investigators with an axe to grind. It is within the capacity of all of us to toss a coin a hundred times. (Persi Diaconis has done it much more often!) I suggest you try the experiment and see how persuasive you find the argument in the light of the experimental data.

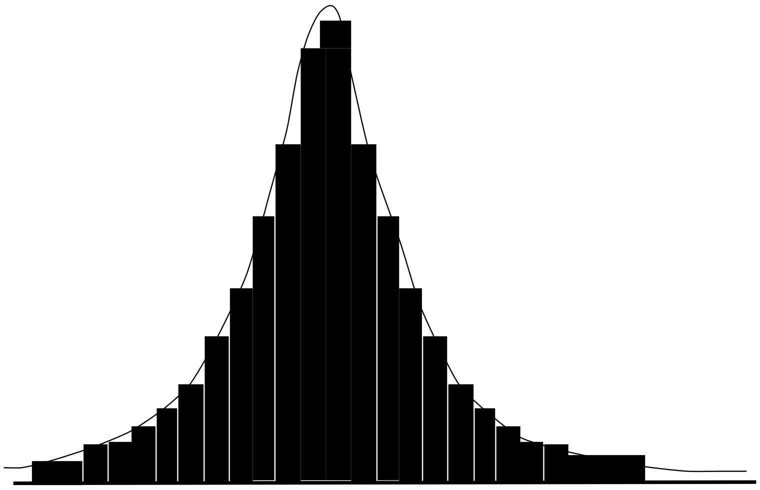

Figure 3.2: Histogram/PDF for a RV taking values in $\mathbb{R}$.

3.1.2 Histograms and Probability Density Functions

If there are only a finite number of possible things we might have as outcomes, we can enumerate them, 1 to k, and, if the random variable is really given to us explicitly, compute the measure for each of them. We could draw a histogram of outputs and *probabilities* associated with them. Such a thing is called a *probability distribution* and we are all familiar with them. For a coin which can only come down Heads or Tails, we can let 0 represent Tails, 1 Heads, and erect a little pole of height 0.5 over each of these two numbers.

If the random variable takes a continuum of values, as when someone hurls darts at a board, [2] we cannot assign a probability to any single

[2] It is usual to model dart throwing as selecting a point at random inside a disk. This might shock genuine darts players. Of course, real dartboards are made of atoms, and space might come in little lumps for all we know, so this model is only one of several.

outcome, since under normal circumstances this will be zero. But we can still draw a histogram for any choice of boxes partitioning the space, $\mathbb{R}^n$, of values of the *rv*. If we normalise, so that the area, volume or in general measure under the histogram is one, and then do it again with a finer partition, we can get closer to a continuous distribution. In the limit, with some technical conditions satisfied that you can safely ignore because they are no more than mathematical book-keeping, you may get a continuous non-negative function over $\mathbb{R}^n$, with integral 1, and this is known as a *probability density function, pdf* for short, and again they are exceedingly familiar. There are some niceties; the *pdf* may not be continuous and the values may be mixed discrete and continuous, but we need not contemplate these issues in our applications. The usual derivation of the *pdf* from the measure is different from the present hint that you do it by limits of histograms, and has little to recommend it unless you are an analyst.

It is worth noting that since we can only measure vectors to some finite precision and we absolutely never get a data set which is of uncountably infinite cardinality, the distinction between a very fine histogram and a real *pdf* is somewhat metaphysical. It is also worth noting that a hypothetical measure space and map from it, about which nothing can be said except the relative meaures of bits corresponding to sets of outputs from the map, is also a touch metaphysical. If you can use one model for the *rv* and I can use another with a different domain space, and if we agree on every calculation about what to expect, then it must cross one's mind that we might manage to do very well without having an *rv* at all. All we really need is a measure on the space of outcomes, the *sample space*. If we reflect that to specify a measure on a sample space which is some region in $\mathbb{R}^n$ the simplest way is by giving a density function, we see that it might have been simpler to start with the *pdf.*

3.1.3 Models and Probabilistic Models

Of course, nobody comes along and gives you a random variable. What they do usually is to give you either a description of a physical situation or some data and leave it to you to model the description by means of an *rv*. The cases of the cards and the coins are examples of this. With the coins, for example, you may take the 12-dimensional state space of physics if you wish, but it suffices to have a two point space with measure 0.5 on each point, the map sending one to Heads and the other to Tails. For *two-up*, you can make do with a two point space, one labelled 'same', the other labelled 'different', or you can have two copies of the space of Heads and Tails with four points in it, a pair of them labelled 'same', the other pair 'different', or a twenty-four dimensional state space with half the space black and the other half white- it doesn't much matter. The histograms over the space of outcomes are the same.

Statisticians sometimes say that the *rv*, or the histogram or *pdf*, is a *model* for the actual data. In using this terminology they are appealing to classical experience of mathematical models such as Newtonian Dynamics. There are some differences: in a classical mathematical model the crucial symbols have an interpretation in terms of measurables and there are well defined operations, such as weighing, which make the interpretation precise. In the case of probabilistic models, we can measure values of a physical variable, but the underlying mechanism of production is permanently hidden from us. Statistical or probabilistic models are not actually much like classical mathematical models at all, and a certain kind of confidence trick is being perpetrated by using the same terminology. Let us look at the differences.

Newton modelled the solar system as a collection of point masses called planets revolving around another point mass called the sun, which was fixed in space. The reduction of a thing as big as a planet to a single point had to be carefully proved to be defensible, because we live on one of them and have prejudices about the matter. The Earth doesn't look much like a point to us. It can be shown that the considerable simplification will have no effect on the motion provided the planets are small enough not to bump into each other, and are spheres having a density function which is radial. Now this is not in fact true. Planets and suns are oblate, and the Earth is not precisely symmetric. But the complications this makes are not large, so we usually forget about them and get on with predicting where the planets will be at some time in the future. The results are incredibly good, good enough to send space craft roaring off to their destiny a thousand million miles away and have them arrive on schedule at the right location. Anybody who doesn't thrill to this demonstration of the power of the human mind has rocks in his head. Note that the model is not believed to be *true*, it is a symbolic description of part of the universe, and it has had simplifications introduced. Also, we don't have infinite precision in our knowledge of the initial state. So if the spacecraft is out by a few metres when it gets to Saturn, we don't feel too bad about it. Anybody who could throw darts with that precision could put one up a gnat's bottom from ten kilometres.

If we run any mathematical model (these days often a computer simulation) we can look at the measurements the model predicts. Then we go off and make those measurements, and look to see if the numbers agree. If they do, we congratulate ourselves: we have a good model. If they disagree but only by very small amounts that we cannot predict but can assign to sloppy measurement, we feel moderately pleased with ourselves. If they differ in significant ways, we go into deep depression and brood about the discrepancies until we improve the model. We don't have to believe that the model is *true* in order to use it, but it has to agree with what we measure when we measure it, or the thing is useless. Well, it might give us spiritual solace or an aesthetic buzz, but it doesn't actually do what models are supposed to do.

Probabilistic models do not generate numbers as a rule, and when they do they are usually the wrong ones. That is to say, the average behaviour may be the same as our data set, but the actual values are unlikely to be the same; even the model will predict that. Probabilistic models are models, not of the measured values, but of their distribution and density. It follows that if we have very few data, it is difficult to reject any model on the grounds that it doesn't fit the data. Indeed, the term 'data' is misleading. There are two levels of 'data', the first is the set of points in $\mathbb{R}^n$, and the second is the distribution and density of this set which may be described by a probabilistic model. Since we get the second from the first by counting, and counting occurences in little cells to get a histogram as often as not, if we have too few to make a respectable histogram we can't really be said to have any data at the level where the model is trying to do its job. And if you don't have a measurement, how can you test a theory? This view of things hasn't stopped people cheerfully doing hypothesis testing with the calm sense of moral superiority of the man who has passed all his exams without thinking and doesn't propose to start now.

Deciding whether a particular data set is plausibly accounted for by a particular probabilistic model is not a trivial matter therefore, and there are, as you will see later, several ways of justifying models. At this point, there is an element of subjectivity which makes the purist and the thoughtful person uneasy. Ultimately, any human decision or choice may have to be subjective; the choice of whether to use one method or another comes easily to the engineer, who sees life as being full of such decisions. But there is still an ultimate validation of his choice: if his boiler blows up or his bridge falls down, he goofed. If his program hangs on some kinds of input, he stuffed up. But the decision as to whether or not the probabilistic advice you got was sound or bad is not easily taken, and if you have to ask the probabilist how to measure his success, you should do it before you take his advice, not after.

A probabilistic model then does not *generate* data in the same way that a causal model does; what it can do, for any given measurement it claims to model, is to produce a number saying how likely such a measurement is. In the case of a discrete model, with finitely many outcomes say, the number is called the *probability* of the observation. In the case of a continuous *pdf* it is a little more complicated. The continuous pdf is, recall, a limit of histograms, and the probability of getting any specified value is zero. If we want to know the probability of getting values of the continuous variable within some prescribed interval, as when we want to know the probability of getting a dart within two centimetres of the target's centre, we have to take the limit of adding up the appropriate rectangular areas: in other words we have to integrate the *pdf* over the interval. For any outcome in the continuum, the *pdf* takes, however, some real value. I shall call this value the *likelihood* of the outcome or event, according to the model defined by the *pdf*. [3]

If we have two data, then each may be assessed by the model and two probabilities or likelihoods output (depending on whether the model is discrete or continuous); multiplying these numbers together gives the probability or likelihood of getting the pair on independent runs of the model. It is important to distinguish between a model of events in a space of observables applied twice, and a model where the observables are pairs. The probability of a pair will not in general be the product of the probabilities of the separate events. When it is, we say the events are *independent*. For example, I might assert that any toss of a coin is an atomic phenomenon, in which case I am asserting that the probability of any one toss producing heads is the same as any other. I am telling you how to judge my model: if you found a strict alternation of heads and tails in a sequence of tosses, you might reasonably have some doubts about this model. Conversely, if I were to model the production of a sequence of letters of the alphabet by asserting that there is some probability of getting a letter 'u' which depends upon what has occurred in the preceding two letters, the analysis of the sequence of letters and the inferences which might be drawn from some data as to the plausibility of the model would be a lot more complicated than the model where each letter is produced independently, as though a die were being thrown to generate each letter. Note that this way of looking at things supposes that probabilities are things that get

[3] There is a school of thought which regards this as a very wicked choice of terminology. This is because probabilists have persuaded themselves that the right way to use the term 'likelihood' is to mentally drive a nail through the data and then regarding the numbers obtained as a function of the parameters used for specifying a *pdf*. *This* is the likelihood. They are, however, making rather a pointless distinction, since they are talking about a function of two variables, data, and parameters of a family of *pdf*s. What I describe as pointless, they would describe as subtle. My objection to this particular subtlety is that it seems to be concerned with the semantics rather than the syntax: they use the term so as to tell you what they plan to do with the number, not how they compute it. Conversely, when offered my definition above, probabilists have assured me disdainfully that there aren't any parameters, so my definition makes no sense. But there is a perfectly good parameter space, it just happens to consist of a single point. I have discussed this matter with distinguished probabilists: usually, each of us winds up thinking the other rather crass. I may return to this point later if my conscience gets to me.

assigned to events, things that happen, by a model. Some have taken the view that a model is a collection of sentences about the world, each of which may often contain a number or numbers between 0 and 1; others, notably John Maynard Keynes, have taken the view that the sentence doesn't contain a number, but its truth value lies between 0 and 1. The scope for muddle when trying to be lucid about the semantics of a subject is enormous.

Another difference between probabilistic models and causal models is the initial conditions. Most models are of the form: *if* condition A is observed or imposed, *then* state B will be observed. Here condition A and state B are specified by a set of measurements, i.e by the values of vectors obtained by prescribed methods of measurement. Now it is not at all uncommon for probabilistic models to assume that a system is 'random'. In the poker calculation for example, all bets are off if the cards were dealt from a pack which had all the hearts at the top, for my having three hearts means that so do two of the other players and the last has four. So the probability of a flush being filled is zero. Now if you watched the dealer shuffle the cards, you may believe that such a contingency is unlikely, but it was the observation of shuffling that induced you to feel that way. If you'd seen the dealer carefully arranging all the hearts first, then the spades and clubs in a disorganised mess, and then the diamonds at the end, you might have complained. Why? Because he would have invalidated your model of the game. Now you probably have some loose notion of when a pack of cards has been randomised, but would you care to specify this in such a way that a robot could decide whether or not to use the model? If you can't, the inherent subjectivity of the process is grounds for being extremely unhappy, particularly to those of us in the automation business.

The terminology I have used, far from uncommon, rather suggests that some orders of cards in a pack are 'random' while others are not, and shuffling is a procedure for obtaining one of the random orders. There are people who really believe this. Gregory Chaitin is perhaps the best known, but Solomonoff and Kolmogorov, also take seriously the idea that some orders are random and others are not. The catch is that they define 'random' for sequences of cards as, in effect, 'impossible to describe briefly, or more briefly than by giving a list of all the cards in order'. This makes the sequence of digits consisting of the decimal expansion of π from the ten thousandth place after the decimal point to the forty thousandth place very much non-random. But if you got them printed out on a piece of paper it would not be very practical to see the pattern. They would almost certainly pass all the standard tests that statisticians use, for what *that* is worth.

There was a book published by Rand Corporation once which was called ' One Million Random Digits', leading the thoughtful person to enquire whether they were truly random or only appeared to be. How can you tell? The idea that some orders are more random than others is distinctly peculiar, and yet the authors of ' One Million Random Digits' had no hesitation in rejecting some of the sequences on the grounds that they failed tests of randomness. Would the observation that they can't be random because my copy of the book has *exactly the same digits* as yours, allowing me to predict the contents of yours with complete accuracy, be regarded as reasonable? Probably not, but what if an allegedly random set of points in the plane turned out to be a star map of some region of the night sky? All these matters make rather problematic the business of deciding if something is random or not. Nor does the matter of deciding whether something is realio-trulio random allow of testing by applying a fixed number of procedures. And yet randomness appears to be a necessary 'condition A' in lots of probabilistic models, from making decisions in a card game to sampling theory applied to psephologists second guessing the electorate. These points have been made by Rissanen, but should trouble anybody who has been obliged to use probabilistic methods. Working probabilists and statisticians can usually give good value for money, and make sensible judgements in these cases. Hiring one is relatively safe and quite cheap. But if one were to contemplate automating one, in even a limited domain, these issues arise.

The notion of repeatability of an experiment is crucial to classical, causal models of the world, as it is to probabilistic models. There is a problem with both, which is, how do you know that all the other things which have changed in the interval between your so called replications are indeed irrelevant? You never step into the same river twice, and indeed these days most people don't step into rivers at all, much preferring to drive over them, but you never throw the same coin twice either, it got bashed when it hit the ground the first time, also, last time was Tuesday and the planet Venus was in Caries, the sign of the dentist, and how do you know it doesn't matter? If I collect some statistics on the result of some measurements of coin tossing, common sense suggests that if the coin was run over by a train half way through the series and severely mangled, then this is enough to be disinclined to regard the series before and after as referring to the same thing. Conversely, *my* common sense assures me that if another series of measurements on a different coin was made, and the first half were done on Wednesdays in Lent and the last half on Friday the thirteenth, then this is not grounds for discounting half the data as measuring the wrong thing. But there are plenty of people who would earnestly and sincerely assure me that my common sense is in error. Of course, they probably vote Green and believe in Fairies at the bottom of the garden, but the dependence on subjectivity is disconcerting. In assigning some meaning to the term 'probability of an event', we have to have a clear notion of the event being, at least in principle, repeatable with (so far as we can determine) the same initial conditions. But this notion is again hopelessly metaphysical. It entails at the very least appeal to a principle asserting the irrelevance of just about everything, since a great deal has changed by the time we come to replicate any experiment. If I throw a coin twice, and use the data to test a probabilistic model for coins, then I am asserting as a necessary part of the argument, that hitting the ground the first time didn't change the model for the coin, and that the fact that the moons of Jupiter are now in a different position is irrelevant. These are propositions most of us are much inclined to accept without serious dispute, but they arise when we are trying to automate the business of applying probabilistic ideas. If we try to apply the conventional notions to the case of a horse race, for instance, we run into conceptual difficulties: the race has never been run before, and will never be run again. In what sense then can we assign a probability to a horse winning? People do in fact do this, or they say they do and behave as if they do, to some extent at least, so what is going on here? The problem is not restricted to horse races, and indeed applies to every alleged replication. Anyone who would like to build a robot which could read in the results of all the horse races in history, inspect each horse in a race, examine the course carefully, and taking into account whatever data was relevant produce estimates of the probabilities of each horse winning, will see the nature of the difficulties. There is no universally accepted procedure which could do this for the much simpler case of coin tossing.

So the question of what is an allowable measurement gives us a problem, since specifying what is irrelevant looks to be a time consuming job. If the cards are marked, all bets are off on the matter of whether its better to draw two to fill the flush or one for the straight. And how do you list all the other considerations which would make you believe that the argument given above was rendered inapposite? The classical case is much simpler because we can tell whether or not something is being measured properly if we have been trained in the art of recognising a proper measurement, and we can build machines which can implement the measurements with relatively little difficulty. But how do we get a machine to tell if the other guys are cheating in a game of poker? How does it decide if the shuffling process has produced enough randomness? And when does it give up on the model because the data is not in accord with it? Subjectivity may be something engineers are used to, but they haven't had to build it into robots before. But whenever you test a model you have to ensure that the input conditions are satisfied and the measuring process properly conducted.

A fundamental assumption in classical systems, seldom stated,[4] is the stability of the model under variations in the data and the model parameters. If you cast your mind back to the curious discussions of weights joined by zero thickness, weightless threads, hanging over pulleys, which took place in applied mathematics classes in olden times, you may recall having been troubled by the thought that if the threads were of zero width then the pressure underneath them must have been infinite and they would have cut through the pulley like a knife through butter. And they would have to be infinitely strong not to snap, be-

[4]But see Thom's *Structural Stability and Morphogenesis*, and Abraham's *Foundations of Mechanics*.

cause the force per unit cross sectional area would also have been infinite. What was being glossed over, as you eventually came to realize, was that although the idealised model made no physical sense, it is possible to approximate it with bits of string sufficiently well to allow the results of calculations to be useful. Idealisations are never satisfied in the real world, but sometimes it doesn't much matter.

What it boils down to then is this: you postulate a model and you simultaneously postulate that if the input data and model parameters are varied just a little bit it won't make any great deal of diference to the output of the model. If you are right about this last assumption, you have a classical model, while if not you have a probabilistic model and have to to be less ambitious and settle for knowing only the relative probabilities of the outcomes.

This is complicated by the platonic, metaphysical component of probabilistic modelling: you may be given a set of conditions under which you are told it is safe to use the model, but the conditions are not such as can be confirmed by any operational procedure. For example, you are allowed to make inferences about populations from samples providing your sample is 'random', but there is no procedure for ensuring that a given selection scheme *is* random. Or you are assured that using a normal or gaussian distribution is warranted when the process generating the data has a 'target' value and is influenced by a very large number of independent factors all of which may add a small disturbing value and satisfy some conditions. But you are not told, for a particular data set, how to set about investigating these factors and conditions. This situation, in which models are 'justified' provided they satisy certain conditions, but there is no operational way of deciding whether the conditions are satisfied or not, is common. The extent to which Statistical methodology is a matter of training people to tell the same story, as in theological college, as opposed to giving the only possible story, as in science, is far from clear, but the assurances one gets from working probabilists are not particularly comforting.

3.2 Probabilistic Models as Data Compression Schemes

Many of the above problems evaporate if one regards probabilistic models in a different light. Instead of seeing them as competitors of Classical mathematical models, we may regard them as simply data compression schemes. I can, and shall, regard a gaussian *pdf* as simply a compressed way of describing a set of points in $\mathbb{R}^n$. This follows Rissanen's approach, see the references for details. This approach is relatively new, but very appealing to those of us who can almost believe in information theoretic ideas.

As has been pointed out by Penttila,[5] another Finn, the Babylonians kept careful records of eclipses and other astronomical events. They also kept careful records of city fires and floods and famines. The records allowed them to eventually become able to predict eclipses but not to predict fires, floods or famines. Thus they only got good results on the things they didn't actually care about. What is done in practice is to follow the example of the Babylonians: you try hard to use the data to make predictions by any method you can devise. If you can't find any pattern at all, you say to yourself 'Sod it, the bloody stuff is random'. This is essentially a statement about your intellectual limitations. If I am much cleverer than you, I might be able to figure out a pattern you cannot. What the moron describes as 'bad luck', the smarter guy identifies as incompetence. One man's random variable is another's causal system, as Persi Diaconis can demonstrate with a coin.

This leads to one taking a class of models and asking how much information can each member of the class extract from the given data set? We may distinguish between a class of models which we can write down and where we are able to do the sums to answer the question *how much does this model compress the data*, and a class of models which might

[5] In the common room of the Mathematics Department at the University of Western Australia. The coffee is awful, but the conversation sometimes makes up for it.

be implemented in the neural tissue of a particular human being, concerning which, it must be admitted, little can be said. For each class of models we can imagine a robot implementing that class, and using the data to select a model or subset of models from the class, possibly by maximising the amount of compression of the data. Presumably, human beings, jelly-fish and other elements of creation are also busily trying to compress the data of life's experience into some family of models. It so happens we can say very little about these machines because we didn't build them; the robots we can say a little about. And a robot of a given class will say that some data is random precisely when that robot is unable to effect any significant amount of compression with the family of models at its disposal.

Once one has abandoned attempts to determine whether anything is 'realio-trulio' random and observed that randomness is relative to some system trying to extract information, much clarity is gained. The belief that there is some absolute sense of randomness, is one of those curious bits of metaphysics to which certain kinds of impractical, philosophical temperaments are vulnerable, but it makes no particular sense. The definitions of randomness in terms of description length which were initiated by Solomonoff, Kolmogorov and Chaitin gives no practical decision procedure for declaring something to be random, it only allows you, sometimes, to observe that it isn't. I shall expand upon this later. See the writings of Rissanen for the words of the prophet on these matters.

3.2.1 Models and Data: Some models are better than others

Given a data set and a (generally infinite) set of models, the problem of finding the 'best' model in the set to describe the data arises. This is known as *point* or *parameter estimation* in the literature. It is not a very well posed problem, as the term 'best' is not defined, and there are several possible meanings it might have. In the next sections we look at the best known contenders for 'best' models.

3.3 Maximum Likelihood Models

Given a probabilistic model and a set of data, we can calculate the likelihood that the model 'produced' each output value and indeed the likelihood that it produced the whole lot in independent applications of a somewhat mythical generating process. Given two such models, we can find out which is more likely to have produced the actual data. For example, if we threw a coin 10 times and got 8 Heads (H) and 2 Tails (T) in some particular order, we might try the family of models

$$\{q \in [0,1] : p(H) = q\}$$

and decide that the p(H) = 0.5 model is not so likely to have produced the outcome as the model with p(H) = 0.7. The computation is rather simple: The p(H) = 0.5 model has p(T) = 0.5 also, and so the probability of the actual data is $(0.5)^8(0.5)^2$. This is, of course, the same as for any other sequence of outcomes, although there are many different ways of getting 8 Heads and two Tails. The probability of having generated the same sequence if p(H) = 0.7 is $(0.7)^8(0.3)^2$, and some messing with a calculator gives 0.0009765 for the former and 0.005188 for the latter. So we get more than five times the probability of obtaining 8 Heads and 2 Tails on the second model as on the first.(In any order.)

In fact it is rather likely to cross your mind that the most attractive model is the one with p(H) = 0.8. It is not hard to prove that this particular model gives a bigger value of the probability of getting that output than any other model from that family. I shall call it the *Maximum Likelihood Model.* [6] Note that in this case we have a nice little continuous space of models, each one specified by a number between 0

[6]This terminology is more used by engineers than probabilists, for whom the term 'model' is more often used to denote what I have called a family of models. Thus a probabilist would commonly say that what I have called a family of models indexed by the numbers between 0 and 1 is just the binomial model. Since it is clearly a good idea to have a distinction between the space of things and the individual things in the space, I shall follow what seems to be a *de facto* tradition among engineers and such roughnecks, that is less confusing than the tradition followed by statisticians and probabilists. Beware of this point when reading the works of the latter.

and 1. I shall refer to this as a one parameter family of models. More generally, it is often the case that we have one model for each point in a manifold or manifold with boundary, and such families of models are referred to as *parametric models*. In some cases it is considered necessary by statisticians to deal with a set of models which cannot be, or at least have not yet been, finitely parametrised and such sets of models are called *non-parametric*. This invites critical comment based on the observation that life provides only finite amounts of data and the conclusion that one cannot hope to distinguish between too many models in such a case, but I am trying to be good.

For any parametric family of models, that is to say one where we have a manifold each point of which is a model, and for any fixed data set, we can compute a likelihood of each model having produced that data. Thus we have a map from the cartesian product of the manifold and the possible data sets into the Reals. In many cases, for a fixed data set, this function has a unique maximum, that is, there is a unique model for which the likelihood of a fixed data set is a maximum. There is a strong intuitive feeling that induces many people to show a particular fondness for the Maximum Likelihood model, and there are routine ways of computing it for several families of models. For example, given a family of gaussians on $\mathbb{R}^n$ parametrised by the centres and the covariance matrices, computing the centroid and the covariance for the data gives the maximum likelihood model.

3.3.1 Where do Models come from?

It is reasonable to ask in the case of coin tossing, where the model comes from. It may well come out of a hat or the seething subconscious of the statistician analysing the data. If I collect some data and you give me two discrete models, I can find which gives me the larger probability. If they are continuous models, I can, using the idea of the *pdf* as a limit of histograms, calculate the likelihood, the value of the *pdf* at the data value, for each of the data points, and multiply the numbers together to get a total likelihood for the whole data set, on that model.

Or as statisticians prefer to think of it, the likelihood of those model parameters for that particular data set. This was discussed, somewhat casually, in the particular case of gaussian models in chapter one. The heuristic rule of picking the larger of the likelihoods was advanced, without anything other than an appeal to intuition to justify it. In the case where the models were all gaussians it is possible to find that model, in the now infinite manifold of all such models, which gives the maximum likelihood. But then we can ask, where does this family of models come from? Maybe gaussians are daft.

There is no algorithm for producing a good family of models; given two models we can produce rules for choosing between them. When the two models are different members of a family with some fixed number of parameters, we can imagine ourselves swanning about the manifold of all such models, looking for the one for which the likelihood of getting our data is maximal. When the models are of very different types, we can still choose the model which gives the maximum likelihood, but this could be a very stupid thing to do. There is always one model which produces exactly the data which was obtained, and no other, with probability one. In the case of the coin which came down Heads 8 times out of 10, there is always the theory that says this was bound to happen. It has certainly been confirmed by the data! This 'predestination' model[7], easily confused with the data set, has probability 1, if it can be assigned a probability at all, but you are unlikely to feel kindly disposed towards it, if only because it says absolutely nothing about what might happen if you tossed the coin again.

3.4 Bayesian Methods

Consider a situation such as that of Exercise 1.1, i.e. the first exercise at the end of chapter one. We imagined that we were going to get lots of pictures of half naked women and about ten times as many trees. Your job was to classify a new image. Now you already have odds of

[7]Known to followers of E.T Jaynes as the 'SureThing' model.

9 to 1 that it's a tree. Shouldn't you use this information in doubtful cases? In this situation, you are treating the models themselves as objects on which to put probabilities. Doing this systematically gives the Bayesian solution, which is an attempt to rationally utilise other sources of information about any one model being the 'best'.

More generally, we may have prejudices about the world based on other experiences upon which we are inclined to rely more heavily than a single small amount of data. In the case of the coins, I may have a predisposition to believe that the probability of Heads is much closer to 0.5 than the 0.8 which this single experiment would suggest. This may reflect my having thrown very similar looking coins more than ten times in the past and having got much less than 80% of them coming Heads. I do not approach this experiment in a total vacuum. These considerations lead to Bayesian ideas about how to choose the 'best' model.

3.4.1 Bayes' Theorem

Suppose we are given the *codomain* of a discrete random variable, that is to say the set of k possible outcomes. Each such outcome will be referred to as an *atomic event*. I shall use the labels 1 to k to specify each of the outcomes.

If we imagine that the *rv* is used to generate data sequentially, as with succeeding throws of a coin, then we will obtain for a sequence of n such repetitions a space of k^n distinct outcomes. The fact that these are repetitions of the same *rv*, means that each of the n repetitions is a separate event in no way contingent upon the results of earlier or later repetitions. The same histogram should be applied to each of them separately when it comes to computing probabilities. Then this defines us a particular kind of new *rv* which is from the cartesian product of n copies of the domain of the basic *rv*, and goes to the cartesian product of n copies of the codomain, by the obvious product map. It follows that the probability of an atomic event in the product *rv* is the product of the probabilities of the components.

There may be a different *rv* on the same domain and into the same codomain(k^n) in the case where what happens at the j^{th} stage in the sequence depends upon what happens at earlier (or later) stages in the sequence. In such a case the probability of an event in the codomain will not usually be expressible as such a product. In order to deal with such cases, and for other situations also, the idea of *conditional probability* is useful.

For any *rv*, an *event* will be used to mean a non-empty element of the power set of the space of atomic events, i.e. a non-empty subset of the codomain of the *rv*. For example, if the *rv* is the obvious model for the throwing of a six sided cubic die, the atomic events can be labelled by the integers 1 to 6, and the set of such points $\{1, 3, 5\}$ might be described as the event where the outcome is an odd number. Then events inherit probabilities from the atomic events comprising them: we simply take the measure of the inverse image of the set by the map which is the *rv*. Alternatively, we take the probability of each of the atomic events in the subset, and add up the numbers.

Suppose A and B are two events for a discrete random variable. Then $p(A)$ makes sense, and $p(B)$ makes sense, and so does $p(A \cup B)$ and $p(A \cap B)$. It is clear that the formula

$$p(A \cup B) = p(A) + p(B) - p(A \cap B)$$

simply follows from the fact that p is a measure, i.e. behaves like an area or volume. In addition, I can make up the usual definitions:

Definition: A and B are *independent* iff $p(A \cap B) = p(A)p(B)$

It might not be altogether trivial to decide if two events which are observed are produced by a random variable for which they are independent. Usually one says to oneself 'I can't think of any reason why these events might be causally related, so I shall assume they aren't.' As for example when A is the event that a coin falls Heads up and B is the event that it is Tuesday. It seems a hit or miss process, and sometimes it misses.

Definition: If A and B are any two events for a discrete *rv* the *conditional probability of A given B*, $p(A|B)$, is defined when $P(B) \neq 0$ by

$$p(A|B) = \frac{p(A \cap B)}{p(B)}$$

There is a clear enough intuitive meaning to this: Given that event B has occurred, we really have to look at the measure of those points in the domain of the *rv* which can give rise to an outcome in the event A from among those which can give rise to the event B. In short we have what a category theorist would call a *subrandom variable*, were category theorists to be let loose in probability theory. A straightforward example would be when B is the event that a die is thrown and results in a value less than four, and A is the event that the die shows an odd number. Then $p(A|B)$ is the probability of getting one or three divided by the probability of getting one or two or three, in other words it is $\frac{2}{3}$. We can also calculate $p(B|A)$, which is the probability of getting one or three divided by the probability of getting one or three or five, which is also $\frac{2}{3}$. In general the two probabilities, $p(A|B)$ and $p(B|A)$, will be different. In fact we obviously have the simple result:

Bayes' Theorem $\quad p(B|A) = \frac{p(A|B)p(B)}{p(A)}$

This follows immediately from the rearrangements:

$$p(A|B)p(B) = p(A \cap B) = p(B|A)p(A)$$

It would be a mistake to imagine that Bayes' theorem is profound, it is simply linguistic. For some reason, Bayes got his name on a somewhat distinctive approach to deciding, *inter alia* which model is responsible for a given datum. For now let us suppose that this extends to the case of continua, and we have the situation of *Fig.1.10* of chapter one. Here, two models, g_m and g_f perhaps, are candidates for having been responsible for the point x. We can calculate without difficulty the numbers $g_m(x)$ and $g_f(x)$. We decide to regard these two numbers as something like the conditional probability of having got x given model m and the conditional probability of having got x given model

f, respectively. We may write this, with extreme sloppiness, as $p(x|m)$ and $p(x|f)$ respectively. These would be interpreted by the sanguine as measuring the probability that the datum x will be observed given that model m (respectively, f) is true. Now what we want is $p(m|x)$ and $p(f|x)$ respectively, the probabilities that the models are true given the observation x. By applying Bayes Theorem in a spirit of untrammelled optimism, we deduce that

$$p(m|x) = \frac{p(x|m)p(m)}{p(x)}$$

with a similar expression for $p(f|x)$.

Suppose we are in the situation of the first problem at the end of chapter one; you will doubtless recall the half naked ladies and possibly the trees from which they were to be distinguished. Now it could be argued that in the absence of any data from an actual image, we would rate the probability of the image being of a tree as 0.9 and of it being a naked lady as 0.1, on the grounds that these are the ratios of the numbers of images. Bayesians refer to these as the *prior probabilities* of the events. So in the above formulae we could put $p(m)$ and $p(f)$ in as numbers if we had some similar sort of information about the likelihoods of the two models. This leaves only $p(x)$ as a number which it is a little hard to assign. Happily, it occurs in both expressions, and if we look at the *likelihood ratio*, it cancels out. So we have:

$$\frac{p(m|x)}{p(f|x)} = \frac{p(x|m)p(m)}{p(x|f)p(f)}$$

and the right hand side, known as the *likelihood ratio* is computable. Well, sort of.

If you are prepared to buy this, you have a justification for the rule of thumb of always choosing the bigger value, at least in the case where $p(m)$ and $p(f)$ are equal. In this case, $p(m|x)$ is proportional to $p(x|m)$ and $p(f|x)$ is proportional to $p(x|f)$ and with the same constant of proportionality, so choosing whichever model gives the bigger answer is the Bayes Optimal Solution. More generally than the crude rule of thumb, if it is ten times as likely to be m as f which is responsible for a datum in a state of ignorance of what the actual location of the datum is, then we can use this to obtain a bias of ten to one in favour of m by demanding that the ratio of the likelihoods be greater than ten to one before we opt for f as the more likely solution.

The blood runs cold at the thought of what must be done to make the above argument rigorous; the step from probabilities, as used in the formulae, to likelihoods (since things like $p(x|m)$ interpreted as $g_m(x)$ are distinctly shonky) requires some thought, but reasonable arguments about *pdf*s being really just the limits of histograms can carry you through this. What is more disturbing is the question of what the random variable is, that has events which are made up out of points in $\mathbb{R}^n$ together with gaussian (or other) models for the distribution of those points. Arguments can be produced. A *Bayesian* is one for whom these (largely philosophical) arguments are convincing. The decision arrived at by this method is known as the *Bayes Optimal decision.* Sticking the word 'optimal' in usually deceives the gullible into the conviction that no critical thought is necessary; it is the best possible decision according to Bayes, who was a clergyman and possibly divinely inspired, so to quarrel with the Bayes optimal decision would be chancing divine retribution as well as the certainty of being sneered at by an influential school of Statisticians. Rather like the calculation of ' 99% confidence intervals' which suggest to the credulous that you can be 99% confident that the results will always be inside the intervals. This is true only if you are 100% confident in the model you are using. And if you are, you shouldn't be.

Using power words as incantations is an infallible sign of ignorance, superstition and credulity in the part of the user. It should be left to politicians and other con-men.

3.4.2 Bayesian Statistics

In the decision case of telling the guys from the gals, we assumed that the models m and f were given and non-negotiable and that the problem was to choose between them as the most probable explanation for the occurrence of the datum x. In many problems, we have the data and the issue is to find the best or most probable *pdf* to account for the whole lot. *Point or parameter estimation* is the business of finding the 'best' model for a given data set. The question, of course, is, what does 'best' mean? We have discussed the idea of the Maximum Likelihood estimator, but there are cases when it is obvious even to the most partisan that it is not a particularly good choice. In the case of the coin which was thrown ten times and came down Heads eight times, the Maximum Likelihood model leads you to believe it will come down 800 times, give or take a few, if it is tossed a thousand times. This might surprise someone who looked hard at the coin and concluded that it looked like any other coin. Such a person might be prejudiced in favour of something closer to 0.5 and unwilling to go all the way to 0.8 on the strength of just ten throws.

Let m be the model that asserts $p(H) = m$; then there is a model for each $m \in [0, 1]$. Let x be the observation consisting of a particular sequence of 8 Heads and 2 Tails. Then $p(x|m) = m^8(1-m)^2$ for any such m. If we maximise $p(x|m)$ for all values of m, we get the maximum likelihood model, which by a bit of easy calculus occurs when

$$8m^7(1-m)^2 - m^8(2)(1-m) = 0$$

i.e. for $m = 0.8$, as claimed in *3.1.5.*

Now let us suppose that you have a prejudice about the different values for m. What I shall do is to assume that there is a probability distribution over the different values which m can have. If I try the distribution $6m(1-m)$ then I am showing a preference for the value of m at 0.5 where this *pdf* has a maximum, but it is not a very powerful commitment.

Now looking at the Bayes formula

$$p(m|x) = \frac{p(x|m)p(m)}{p(x)}$$

we obtain immediately that

$$p(m|x) = \frac{m^8(1-m)^2 6m(1-m)}{p(x)} = \frac{6m^9(1-m)^3}{p(x)}$$

Since $p(x)$ is some prior probability of x which, I shall suppose, does not depend upon the model, m, I can use calculus to obtain the maximum of this expression. A little elementary algebra gives a value of $m = \frac{3}{4}$. This is between the maximum likelihood estimate and the prior prejudice of 0.5. Since my commitment to 0.5 was not very strong, as indicated by the *pdf* I chose, I have gone quite a long way toward the ML model. Had I used $30m^2(1-m)^2$ as my measure of prejudice, I should have been closer to 0.5.

The above argument assumes that $p(x)$ does not depend on m. But what *is* $p(x)$? If x has just been observed, then it ought to be 1; so it doesn't mean that. The most sensible answer is that it is the so called *marginal* probability, which is obtained by adding up all the $p(x|m)$s for all the different m, weighted by the probability of m. Since these are all the values between 0 and 1, we have

$$p(x) = \int_{m=0}^{m=1} p(x|m)p(m)dm$$

which means that dividing by $p(x)$ simply amounts to normalising the numerator in the definition of $p(m|x)$ above, thus ensuring that it is a *bona fide pdf*. So we have not just a most probable value for the estimated value of the parameter m given the observation data x, we have a *pdf* for m. This *pdf* known, not unreasonably, as the *posterior pdf*, and is written, as above, as $p(m|x)$, and may be compared with $p(m)$ as the *prior pdf*. It is easy to compute it explicitly in the case of the coin tossing experiment, and the reader is urged to do this as it is soothing, easy and helps burn the idea into the brain in a relatively painless manner. The maximum of the posterior *pdf* is at some places in the literature referred to as the MAP probability, because acronyms are much less trouble to memorise than learning the necessary latin, and

Maximum A Posteriori takes longer to say than 'MAP', even though it can be difficult for the untrained ear to hear the capital letters.

In the general case where data is coming in sequentially, we can start off with some prior probability distribution, and for each new datum use this method to predict what the datum ought to be, and when we find out what it actually was, we can update the prior. This is clearly a kind of learning, and it is not beyond the range of belief that something similar may occur in brains. We talk, at each stage of new data acquisition, of the *a priori* and *a posteriori pdf*s for the data, so we obtain an updated estimate of the 'best' *pdf* all the time. This will not in general be the same as the Maximum Likelihood estimate, as in the example.

3.4.3 Subjective Bayesians

There are people who feel in their bones that it makes sense to assign a prior distribution to such things as the different models for the coin tossing in the last section, so as to represent their own internal states of belief. Such a person might say 'Well, I think that $p(H) = 0.5$ is reasonable as a result of looking at the coin and its symmetry. I suppose I could be persuaded that $p(H) = 0.4$ or $p(H) = 0.6$ are reasonably likely, while I am much more sceptical about values outside that region. The preceding sentences express my prejudices in English, which is a very scruffy and imprecise language, what I *really* mean is that my degree of belief in the model $p(H) = m$ is precisely $6m(1-m)$ for m between 0 and 1. '

This kind of statement may strike one as rather odd. It suggests, at the very least, a good deal of confidence in one's powers of introspection. Most people can barely make up their minds what to have for dinner tonight, while a subjective Bayesian appears to have his preferences expressible by analytic functions. The idea that beliefs can be expressed with such precision and certainty, to as many decimal places as one wishes to use, has been criticised severely by, for instance, Peter Walley.

We may doubt that an individual's beliefs can be known to him with such precision, but if we regard any animal as a learning machine which at any time has amassed a certain amount of knowledge of how the world works, then any situation in which it finds itself might be said to be a generator of expectations of what will happen next. These expectations are, one presumes, the result of the animal's brain having stored counts of what happened next, in roughly similar circumstances, in the past.[8] How the brain stores and uses this information is of course still largely conjectural, but it would appear to be what brains are for. And it is not beyond belief that the different possible outcomes of a situation may be assigned different degrees of confidence in some representation accomplished by neurons, so there may indeed be some discrete approximation to a Bayesian prior *pdf* sitting in some distributed fashion in the brain of the animal. And if the animal happens to be a subjective Bayesian, he may feel competent to tell us what this *pdf* feels like, even unto using the language of analytic functions. This last is the hardest part to believe, but dottier ideas have come good in Quantum Mechanics, as my old Mother used to say.

When the animal is fresh out of the egg, it may have rather bland preferences; these may be expressed by uniform priors, for example the belief that all values of m are equally likely. Or we may suppose that far from having a *tabula rasa* for a brain, kindly old evolution has ensured that its brain has in fact many pre-wired expectations. If an animal is born with wildly wrong expectations, then kindly old evolution kills it stone dead before it can pass on its loony preconceptions to junior.

It is not absolutely impossible then that at some future time, when the functioning of brains is much better understood than it is at present, by some careful counting of molecules of transmitter substances at synapses using methods currently undreamt of, it might be possible to actually measure somebody's prior *pdf* for some event. Whether this might be expected to give results in good agreement with the opinion

[8] What 'roughly similar' means is, of course, intimately tied up with transfer of learning and the crucial business of what precisely constitutes a replication of an experiment? These issues are slithered over in the foundations of probabilistic statistics because they are too hard.

of the owner of the brain on the matter is open to debate. Subjective Bayesians propose to carry on as if the experiments had been carried out and given the answers they want to hear; this seems foolhardy, but one can admire their courage.

Objective Bayesians are in Schism with Subjective Bayesians; just as the Fascists were in schism with the Communists. And it is pretty hard for outsiders to tell the difference in both cases, although the insiders feel strongly about them. Jaynes has written extensively about sensible priors used to describe ones ignorance or indifference in a way which should be carefully studied by anyone with a leaning toward Fuzzy Sets. An objective Bayesian believes that one can work out not what prior is in his head, but what prior *ought* to be in his head, given the data he has. Thus he has a commitment to the view that there really is a best thing to do given one's data and the need for decision. If the subjective Bayesian believes he is modelling his own belief structures, the objective Bayesian is modelling what he thinks he ought to believe. Presumably he believes he ought to believe in objective Bayesianism; the possibility of an objective Bayesian concluding that there is insufficient evidence for objective Bayesianism but deciding to believe in it anyway, ought not to be ruled out. The idea that there is a best prior, a right and reasonable prior, but that it might be very difficult to find out what it *is*, leads to *sensitivity analysis* which aims to determine if a particular choice of prior is likely to lead you into terrible trouble, or if it doesn't much matter which prior you choose from a class.

It is much easier to philosophise in a woolly manner, and get nowhere, than to come to clear conclusions on matters such as these. This makes the subject fascinating for some and maddening for others.

I.J. Good has written on the philosopical issues involved, if you like that sort of thing.

3.5 Minimum Description Length Models

So far we have considered the issue of selecting, from some family of models, that model which is 'best' by considering which model assigns the data the higher probability, possibly bearing in mind prior prejudices about reasonable results. Another issue in comparing two models is not just the likelihood of the data having arisen from the model, but the model complexity.

Simple models which give reasonably close results are more attractive that very complicated models which give exactly the right answer. In fact we may decline to believe a sufficiently right model, and we always feel uncomfortable with very complicated ones. So model complexity is a consideration, and the question of how we measure it arises naturally. This is where Rissanen comes in.

Rissanen has argued roughly as follows: a model of a data set allows you to compress the data for purposes of transmission or storage. If, for example you had a data set consisting of a sequence of 100 digits which went 314159.... and agreed with π until the end of the sequence, sending the information coded as 'first hundred digits of π, forget the decimal point' would compress the sequence rather considerably. As long as the guy at the other end knew what π meant. If you don't feel safe about this, you may feel obliged to tell him, by, for example, also sending the algorithm for generating π. This extra, I shall call the 'overhead'. Now one model is better than another in Rissanen's sense if it does more compression, taking into account the overhead. If you reason that the more you compress the data, the more information you are extracting from it, this sounds very plausible. This principle allows us to reject the maximum likelihood 'raw data' model in favour of a gaussian model when the number of data points exceeds some quite small number. It also allows us to make sensible decisions on how many gaussians one ought to use in modelling a point set as a mixture of gaussians, a matter of practical importance when we get to syntactic pattern recognition. The ideas are related to Chaitin's attempts to characterise randomness.

See Chaitin (1975) and Chaitin (1987) for an antidote to some of the cynicism of the present and an earlier section.

In this section I sketch the recently developed and indeed still developing theory of Rissanen on *Stochastic Complexity*, as described in his book Rissanen(1989) and papers (see the bibliography). It offers a philosophically quite different approach to finding the 'best' model to account for the data, and an approach which avoids some of the objections I have pointed out to the classical theory. In particular it appears to solve elegantly the issue of how many gaussians one ought to have in a gaussian mixture model for a particular data set. (The answer may be 'none', if the data set is small enough!) This will concern us in the next chapter, where we meet such beasts.

The general question as to order of a model occurs in other settings also. At the simple level of fitting a polynomial to a set of data points, the question as to what degree of polynomial is appropriate arises. At one extreme one can have the degree so high that every point is fitted, at the other we can assume it is constant. The same problem arises in the case of Neural Nets, as we shall see in a later chapter.

The gaussian mixture modelling problem, how many gaussians should we use, makes the problem quite stark. There will be an increase in the log-likelihood of the data with respect to the model whenever we increase the number of gaussians, until the degenerate case of non-invertible covariance matrices makes it infinite. But there is a penalty to be paid in terms of the number of parameters needed to specify the increasing number of gaussians. It is sensible to try to penalise the model in some way by offsetting the increase in log-likelihood with a corresponding increase in the number of parameters, and seeking to minimise the combination. But then the question arises, what should be the rate of exchange between log-likelihood and the parameter count? Rissanen gives us a natural and plausible rate of exchange. Because the ideas have not yet percolated through to the Pattern Recognition community in its entirety, I shall explain them in elementary terms.

I would like to be able to assume a moderate familiarity with Shannon's *Information Theory*, which is by now an essential part of the mental furniture of anybody with pretensions to an education, but it seems unsafe to do so. Typically, an engineers mathematical education seems to consist increasingly of having memorised some recipes and then forgotten them, and computer scientists, so called, have left out most mathematics except a little finitary material such as logic and search algorithms. So I shall explain the basics as I proceed. If I insult your intelligence and knowledge then I put my trust in your tolerance.

3.5.1 Codes: Information theoretic preliminaries

Suppose I have a piece of English text to transmit down a line or store on disk. I shall suppose that the conventional ASCII character set and its usual representation by seven bits is well known to you and everybody who might take an interest in computers and text. Let us assume that the text consists of the sentence

```
the cat sat on the mat
```

which has just 22 characters. It therefore takes up 154 bits with the usual ascii coding. If we use 8 bits to a character, with say a parity check bit, then it goes up to 176 bits.

Let us examine the best compression we can think of, and ignore the noise and corruption which the parity check bit is designed to detect.

Most letters do not occur at all, and of those that do, the frequency varies. The frequencies of occurrence and the corresponding probabilities (divide by 22) rounded are:

```
t 5  0.22727
^ 5  0.22727
a 3  0.13636
h 2  0.09091
```

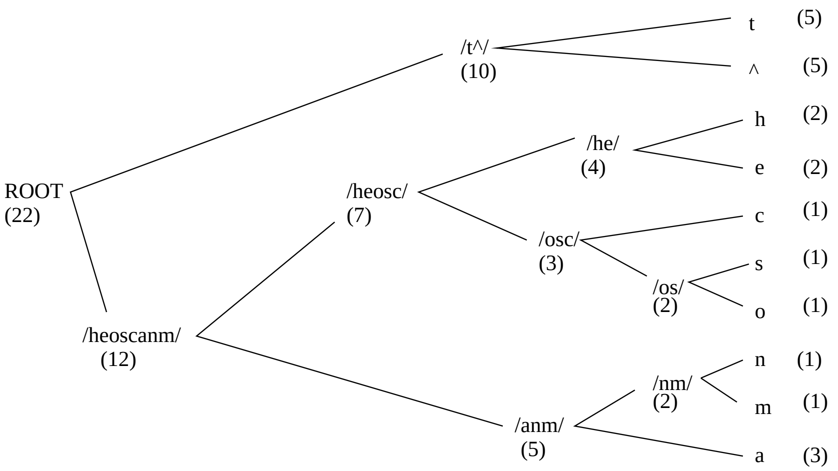

Figure 3.3: A Huffman tree to code 'the cat sat on the mat'.

e	2	0.09091
c	1	0.04545
s	1	0.04545
o	1	0.04545
n	1	0.04545
m	1	0.04545

I have put the characters in decreasing order of frequency, with a space denoted by ^ . Now I shall recode the characters by giving a new dictionary chosen so that the symbols which occur most often get the shortest binary coding. There are several well known ways of doing this, Shannon-Fano and Huffman being the best known.

In Huffman coding (Huffman, 1952) we imagine that we have the letters and probabilities at the leaves of a tree and proceed to join them together pairwise until we have a single node, the root of the tree. The simplest rule for joining nodes, and the one I shall use in this example, is to pick the two with the lowest summed probability, and to choose arbitrarily when there are several such.

In the above case we join together m and n as these have smallest probabilities and call the resulting node/mn/. Next we join o and s to get /os/. Each of /mn/ and /os/ have sum probability 0.09091 (after allowing for rounding). Now we join /os/ to c to get /osc/ with probability 0.13636. We could have chosen to join /mn/ to c, it doesn't make any difference to the end compression. We continue in this way until we have the whole tree, as indicated in *Fig.3.3.*

Now we obtain a code for each symbol by starting off from the ROOT and proceeding to the symbol. Every time we turn left at a fork we write '0' and every time we turn right we write '1'. Thus the symbol t gets coded as 00, while a space is 01. 'a' becomes 111, and the rest are as shown:

```
t 5  0.22727   00
^ 5  0.22727   01
a 3  0.13636   111
h 2  0.09091   1000
e 2  0.09091   1001
c 1  0.04545   1010
s 1  0.04545   10110
o 1  0.04545   10111
n 1  0.04545   1100
m 1  0.04545   1101
```

Now suppose we code using this system. We get:

```
00 1000 1001 01 1010 111 00 01 10110 111 00 01
t   h    e    ^   c   a   t  ^    s    a   t ^

10111 1100 01 00  1000 1001 01  1101 111 00
  o     n  ^   t    h    e   ^    m    a   t
```

Several points merit attention. First, the total number of bits has dropped from 154 to 67, a considerable saving. Second, the code is a *prefix code* which is to say it can be decoded left to right without putting spaces in, no codeword is a left segment of another codeword. Third, the saving is illusory since I also have to send the alphabet of ten symbols (in ascii, 7 bits per symbol) followed by the code word for it, a total of 70 + 37 bits, which gives 174 bits. On the other hand, we didn't consider the overhead involved in the ascii code being sent!

The need to send the coding scheme I shall refer to as the *overhead* associated with the scheme. Of course, the coding scheme need be sent only once, and it is unreasonable to expect much compression in such a short string anyway. The actual example is fairly silly of course, but using the same idea, it is easy to get respectable levels of string compression for natural language text.

A little thought suggests some improvements. The 'model' we are tacitly applying here is for English text being a random sequence of letters each with a predetermined frequency. English is pretty bad, but not as bad as that. If we were to store the frequency of consecutive pairs of symbols, the 'bigrams', to use Shannon's terminology, we would get a more accurate model of English. For large enough samples of English text, an accurate bigram count would surely do a better job than the singleton count used in the example. After all, most bigrams don't occur at all, and some redundancies in English (such as u after q except in QANTAS) can be exploited. And why not move on to trigrams and tetragrams and so on? Given large enough samples of text from which to extract the $n$grams, we could extract, you might think, every scrap of redundancy in English.[9] It is true, as one's intuitions suggest, that there are improvements of coding which, by treating blocks of text, get closer to the ideal coding implied by the model, and also that one cannot do better without a better model.

There are several ways of doing better than Huffman coding, but the first point to which I wish to draw your attention is that there are two components to compression of symbols: the first is a *probabilistic*

[9] It isn't that easy, but this is another subject.

model which gives frequencies of occurences which may be expected of symbols or blocks of symbols. The second is the exploitation of this model by assigning strings of bits to the symbols or blocks so that the high frequency symbols or blocks get the shortest strings.

The second point to observe is that if x is a symbol in an alphabet $\mathcal{X}$ of cardinality M, and if $p(x)$ denotes the probability of getting the symbol x in the stream we wish to transmit (or store) according to a model, then Huffman coding gets us reasonably close to being able to code x with a string of bits of length $l(x)$, where $l(x) \approx \log_2(\frac{1}{p(x)})$. It is easy to see that if we deal with $n$grams, the increase in the alphabet size allows us to get closer to this ideal coding length. And given any coding in bits of length $l(x)$, then the code gives a 'probability' to x of $(\frac{1}{2})^{l(x)}$. This is shown for the silly example, above, to indicate how one gets reasonably close despite the smallness of the data set.

t	5	0.22727	00	0.25
^	5	0.22727	01	0.25
a	3	0.13636	111	0.125
h	2	0.09091	1000	0.0625
e	2	0.09091	1001	0.0625
c	1	0.04545	1010	0.0625
s	1	0.04545	10110	0.03125
o	1	0.04545	10111	0.03125
n	1	0.04545	1100	0.0625
m	1	0.04545	1101	0.0625

If we code $\mathcal{X}$ in the obvious, 'blind' way, we need $\log_2(|\mathcal{X}|)$ bits per symbol and if the message is of length L, it takes up $L\log_2(|\mathcal{X}|)$ bits to send it, neglecting overhead. If we code it using the model, we have, assuming we can get $l(x)$ close to ideal,

$$L\sum_{x\in\mathcal{X}} p(x)\log_2(\frac{1}{p(x)})$$

which is usually less. In fact it cannot be more, and the two are equal in the 'maximum entropy' case where $p(x) = \frac{1}{|\mathcal{X}|}$. Proving this is a simple bit of calculus. Note that we take the value of $p(x)\log_2(\frac{1}{p(x)})$ to be 0 when $p(x)$ is zero- which makes sense if you draw the graph of $p(x)\log_2(\frac{1}{p(x)})$ for x between 0 and 1.

The third point to observe is that if you poured water at 1 litre per second into the root of the tree in *Fig.3.6* and the water splits up at the first node so that half goes each way, and similarly for every subsequent place where the pipe bifurcates, then the water coming out at the leaf labelled t emerges at $\frac{1}{4}$ litres per second, and if x is a leaf, then the amount of water coming out at x is $\frac{1}{2^{l(x)}}$. Now if you add up the amount of water coming out of all the leaves, equivalent to adding up all the numbers in the last column of the above table, you get the amount of water going in, which is 1. It is clear that you couldn't have more water coming out than going in, but the above counting neglects the possibility that we might have some code words left over with no symbols to be assigned to them. We therefore deduce the rather trivial result:

$$\sum_{x \in \mathcal{X}} \frac{1}{2^{l(x)}} \leq 1$$

It is easy to see that there is a generalisation to the case where we use more than two symbols in place of our binary code. We leave it to the reader to write it down. This result is the well known *Kraft Inequality.* [10] It is plain that any code for $\mathcal{X}$ must satisfy the Kraft inequality, and this sometimes comes in useful.

A fourth point to note is that if the model, which asserts something about the probability of getting the symbol x, is wrong, then it is easy to see that the coding based upon the model will be less effective at compressing the data stream than if the model were right. This follows

[10] Proofs done using water are frowned upon in some quarters, usually by people to whom proofs are part of life's mystery, and who want to keep life as mysterious as possible. You may have read some of their books, or tried to.

because we can compare the length of the coded data stream using the actual frequencies, say $q(x)$, when we get

$$L \sum_{x \in \mathcal{X}} q(x) \log_2(\frac{1}{p(x)})$$

for the length of the bitstream coded on model p and

$$L \sum_{x \in \mathcal{X}} q(x) \log_2(\frac{1}{q(x)})$$

for the length of the bitstream coded on model q, and the second is less than the first by an easy little exercise in constrained optimisation. It is obviously a good idea to get as good a model as possible for what actually happens, if you want the maximum compression.

A fifth point to make is the rather obvious one that if we can predict the rest of the string past some point with absolute certainty, then the rest of the string is not informative to anyone who knows our prediction system. All we need to do is to send our prediction system and that suffices to allow the receiver to reconstruct the rest of the sequence. This, of course, is the idea behind sending the algorithm for π instead of the first ten thousand digits. Our probabilistic model has some probability 1, and the length of the compressed string falls to zero. We have to send only the overhead.

Summarising, compression arises from redundancy in the string which allows us to model the string and the different frequencies of the components of the string. The more we know about the string the shorter the resulting coded version. The belief that there is some intrinsic amount of redundancy in the string is simple minded, what we have is the possibility of better and better models. The better your model, the more compression. We can extract redundancy from the data, relative to the model, by a coding scheme. The difference

$$L \left(\log_2(|\mathcal{X}|) - \sum_{x \in \mathcal{X}} p(x) \log_2(\frac{1}{p(x)}) \right)$$

measures the amount of information extracted from a stream of symbols from the alphabet $\mathcal{X}$ of length L by the model which gives the probabilities over the alphabet. In the case where the model fits the data perfectly with probabilities being 1 or 0, this is all of the information there is.

In practice, in order to use any such scheme, we have to transmit the model itself, which overhead may make it simpler to transmit the data instead. For large amounts of data, compression would seem to be a good investment.

This gives us an unambiguous sense of when one model is better than another: we can measure the amounts of compression we get on the data which the model is supposed to be modelling. The one which gives higher compression has been extracting more information (that is the only sensible way to measure information) from the data. In making this comparison, we may have to consider the overhead of the models. If the models have the same number of parameters, then the overheads are the same and may be ignored, but if the model is complex, the overhead may be unacceptable- and this may be determined unambiguously by measuring the total compression, taking into account the overhead.

You will be, I hope, familiar with the practicalities of *Information Theory*, or *The Mathematical Theory of Communication*, as Shannon prefers to call it. In order to think about what is going on, as opposed to simply knowing how to do the sums, it is useful to think of the basics as follows.

First one argues that information is passed in *messages* which consist of temporal sequences of *symbols*. We may as well take it that these symbols come from some finite alphabet, because if they came from an infinite alphabet, either we should have to wait an infinite length of time to get just one of them or we should have some practical limit on our ability to distinguish when two of them are different. So we model the whole business of communication as an exchange of symbol strings from some finite alphabet. Now the sender and receiver of these strings, not necessarily different people, will have, at any point in the string, some kind of expectation about the possibilities of what symbol can come next, which we propose to model as a probability distribution over the alphabet. This probability distribution will, of course, change as we go down the string of symbols.

Now at any point, for any symbol in the alphabet, I shall take the probability of the symbol and take its reciprocal and call it the *improbability*. Then I take the logarithm of this improbability and call it the *surprise* of the symbol turning up at this location. This seems a reasonable name, since if the probability is 1, so it is certain, then the improbability is also 1 and the logarithm is zero, so I shall get zero surprise. If the probability is 0 so that I firmly believe it can't possibly happen, then I get an infinite amount of surprise, which serves me right. Not so much a surprise as a paralysing shock. And if the probability is 0.5, then the improbability is 2, and if I take logarithms to the base 2, I get one unit of surprise. Such a unit is called a *bit*. Since probabilities for independent events multiply, so do improbabilities, and surprises simply sum. So if I get one bit of surprise from one throw of a coin (as I will if it is a fair coin, or if I believe it is) and then I throw it again and get another bit of surprise, I get altogether two bits of surprise.

Note that this is the surprise produced by the data *relative to the model*. It doesn't make sense to talk of the amount of surprise in the data *per se*. If you and I have different models, we might experience quite different amounts of surprise at different parts of the string of symbols.

The average amount of surprise I expect is easily calculated: the symbol x will occur with frequency $p(x)$ (which will be conditional on all sorts of things) and each time it occurs I get $\log_2(\frac{1}{p(x)})$ bits of surprise, so the total is the sum of this over all symbols in the alphabet, weighted by the relative frequency of occurrence. This gives the well known formula for the *entropy* of the probability distribution:

$$\sum_{x \in \mathcal{X}} p(x) \log_2(\frac{1}{p(x)})$$

This is clearly a property of the model and has nothing at all to do with the data. It is, of course, known as the *entropy* of the model. I can also compute, by summing the amount of surprise as I go along, the total amount of surprise I get from any given message, and the average surprise per symbol requires that I divide this by the number of symbols in the message. If the model is a good model of the message, then these two numbers ought to be fairly close. Given two models, I can measure the exent to which they expect, on average, the same symbols with similar probabilities, and thus get a measure of how different the models are, the *Kullback-Leibler* measure. What I call the 'surprise' is more commonly called the *information* associated with the symbol. I guess it sounds classier.

Most of this may be found, in improved and extended form, in the book by Shannon and Weaver, which should be on everybody's bookcase. What isn't may be found in the book *Elements of Information Theory* by Cover and Thomas of the bibliography.

In conclusion, a thought for the day.

Brains may be regarded as devices for figuring out what is likely to happen next so as to enable the owner of the brain to improve his chances of survival. A brain has to compress the experience of the owner's life so far, and possibly also some of the experience of his ancestors. The amount of data is huge, and compressing it is essential. Happily, the better at extracting information from that experience, the more compression. So a device for figuring out what is likely to happen next will have the best chance of surviving in a hostile world when it is best at extracting information and hence of compressing its experience.

3.5.2 Compression for coin models

Suppose we conduct the experiment with tossing a coin ten times and get the sequence of 8 Heads and 2 Tails, as described earlier. Again contemplate two models, the p(H) = 0.8 model and the p(H) = 0.5 model. I want to compute the compression I can obtain by using these models to describe the actual data.

Using model m=0.5, because the arithmetic is easier, on the first throw, the probability of Heads (according to the model, models are things that have probabilities, not coins) is 0.5. The improbability is the reciprocal of this which is 2, and the information I get when a Head actually turns up is $\log_2(2) = 1$. This happens for the first 8 throws of the coin, and I get 1 bit of information or surprise each time. On the last two throws, I get a Tail which I also expect with probability 0.5, so I get another 1 bit of information for each of these. This adds up to a total of ten bits, and that is the most uninformative model I could have for the process.

Using model m = 0.8, I get a total amount of surprise of

$$8\log_2(\frac{1}{0.8}) + 2\log_2(\frac{1}{0.2}) = 8(0.3219) + 2(2.3219) = 7.219$$

which is less. So the amount of compression of this string of digits on this second model could reduce it from 10 bits to just over 7 bits. Huffman coding won't help, there are too few symbols, but *Arithmetic coding* can help. I describe this now.

In writing a number between 0 and 1 in binary, I imagine representing my number as a point on the line. I divide the line into two equal bits, with $\frac{1}{2}$ in the middle. If the number is in the left half of this partition, I write 0, if in the right half I write 1. Now I take the half containing the point and blow it up so that it looks the same as the original interval, and again I subdivide it in the middle. If my point is in the left half I write 0, if in the right half I write 1. So far I have two digits, .00 if the point is in the left quarter, .01 if it lies between $\frac{1}{4}$ and $\frac{1}{2}$, and so on. I continue in this way until I have specified the position of the point to a sufficient precision, unless I am a pure mathematician in which case I continue indefinitely. This gives pure mathematicians something to do.

Now there was no necessity to chop up the interval into two subintervals; had I used ten and labelled them from 0 to 9, I should have obtained the decimal expansion of the number by the same process. Nor is there any necessity to chop the interval up into *equal* intervals. I shall now chop it up into a left hand portion between 0 and 0.8, and a right hand portion from 0.8 to 1.0. If the first coin is Heads, I put it in the left portion, if tails in the right portion. If the second coin is Heads, I do the same on the expanded portion which got used last time. I continue until all ten coin throws have been described. The result is an interval. For the case of ten throws with the first 8 being Heads, I actually find that the interval is from $0.96(0.8)^8$ to 0.8^8. In decimal this comes out as between 0.16106127 and 0.16777216 approximately. Now to transmit the result in arithmetic code, I simply choose a number in binary which is in this interval. If I receive such a number and know that there are only ten results and that the 0.8/0.2 partition has been used, I can deduce all the results. In fact the binary number 0.0010101 with only seven bits comes out as 0.1630625 exactly, which is inside the interval. So I can actually send seven bits to code the sequence- using the model m = 0.8 to do it. This involves a small amount of cheating which I leave you to ponder. Beating the ideal of 7.219 looks suspicious and is.

Some innocent experimenting with other models and a calculator may lead you to enlightenment faster than any amount of prohibited substances.

It is easy to see that the m = 0.8 model does better than any other. The reason is obvious: when we add up the logarithms of the reciprocals of the probabilities we are multiplying the improbabilities, and the minimum of this number is the maximum likelihood. So the maximal compression model in this case is the good old Maximum Likelihood model. It is not too hard to believe that in the case where a Bayesian convinces you that the MAP model is the best one for fitting the data that hasn't yet come in, then your best bet for compressing this data, if he is right, is the MAP model.

The overhead of sending the compression scheme, i.e. the model and the procedure for Arithmetic coding is, presumably, the same in both cases and may be ignored.

3.5.3 Compression for *pdf*s

Although the ideas are clear enough in the case of discrete distributions, it is not quite so easy to see how to use a gaussian *pdf*, for example, to compress a set of data. And yet in real life we always specify our real numbers to some precision, and the use of an infinite string of digits is a pretence. It simplifies some aspects of thinking about the world if we decline to say how many digits following the decimal point are going to be used, but it brings us some formidable complications in the language at a later date. (All the paradoxes of infinite sets, the existence of uncountable sets, arise from a desire to go beyond the confines of a finite language for essentially aesthetic reasons.) It is clear that to send one real number alone would require, for almost all real numbers, an infinite amount of information. So let us choose, from the beginning, some sensible bound on the precision of all the numbers we are going to transmit.

Now suppose we decide to work to one part in 2^P for some positive integer P. And suppose we are dealing with a set of points on the real line, in order to keep the example simple, and let us choose the five points $\{0.999, 0.099, 0.009, 0.001, 0.000\}$.

Suppose we decide to send these points to three decimal places; each place is a digit which takes four bits to code, since there are ten of them. So there are $(5)(4)(3) = 60$ bits. If we found a more efficient coding of the digits, we would be able to get $\log_2(10) = 3.32193$ bits per digit, giving 49.8289 as the total number of bits. Can we do better with an improved model which takes into account the fact that most of the density is near 0?

Suppose we define a *pdf*, $f(x)$, over the interval $[0, 1]$ by having the height of $f(x)$ as

$\frac{1}{3.7}$ for $x \in (0.1, 1.0]$,
$\frac{10}{3.7}$ for $x \in (0.01, 0.10]$,
$\frac{100}{3.7}$ for $x \in (0.001, 0.01]$,
$\frac{1000}{3.7}$ for $x \in [0, 0.001]$.

Then let us divide the unit interval up into the two parts from 0 to 0.1 and from 0.1 to 1, and for each point, write $< .0?????? >$ if the point is in the left tenth and $< .1??????? >$ if it is in the right 90%. The question marks have to be filled in by repeating the process a few times.

It is clear that repeating it three times to get $< .000 >$ locates the point as being in the interval from 0.00000.. to 0.001 (using decimal notation) so we have correctly specified the first three digits of 0.000. To write $< .001 >$ specifies a number in the interval from 0.001000... to 0.001999.. in decimal, which adequately specifies the first three digits of 0.001.

We can specify then in binary the points 0.000 and 0.001 using the bit strings $< .000 >$ and $< .001 >$ respectively, thus getting two of the points in just 6 bits. The point 0.009 starts off in binary as $< .001??? >$ which tells us that it lies between 0.001 and 0.01. Since the *pdf* is uniform in this region, I simply code the interval in straight, equal interval, binary. I need to divide the region into 10 intervals to get 0.009. This can be done with an extra four bits, with a little wastage, Thus 0.009 is coded as $< .0011001 >$. So 7 bits suffices for the third point out from the origin.

0.099 is the next point, and we start off with $< .01??? >$ to say it lies within the interval from 0.01 to 0.1. This has to be divided uniformly into 100 intervals which can be done using an extra 7 bits, again using equal interval binary, giving a total of 9 bits for this point.

Finally we have to code 0.999 and this starts off $< .1???? >$ to tell us it lies in the interval from 0.1000.. to 1.0000... If we divide this interval into 1000 intervals we can specify the location to the required precision with an extra 10 bits. This gives 11 bits for the last number. The total number of bits is thus $3 + 3 + 7 + 9 + 11 = 33$ bits. I have used a sneaky convention to cut down on the bit lengths, and a pessimist might throw in another few bits, but this is a saving of over 13 bits in anybody's language.

If we consider the uniform probability distribution, recalling that we quantised to a precision of three figures, we see that the uniform distribution would give each of the five points a probability of $\frac{1}{1000}$ and so the information is $5 \log_2(1000)$ which is the 49.8289 of the starting point. Using the assumed *pdf* we get that for the point $< .000 >$ for example, the probability is the height of the function multiplied by the width, which I shall generously put at $\frac{1}{2000}$, and the improbability is the reciprocal of this which is $(3.7)(\frac{2000}{1000}) = 7.4$, so the surprise or information content is $\log_2(7.4) = 2.8875$. We get the same for the string $< .001 >$ while for the point at 0.009 we get $\log_2(74) = 6.20945$. These correspond to the 3 bit and 7 bit codings of the first three points. Continuing we get:

$$2 \log_2(7.4) + \log_2(74) + \log_2(740) + \log_2(7400) = 31.48167$$

which is a little better than we can realise with our coding scheme but shows the principle at work.

This, of course, all assumes that the overhead is negligible, which is hardly reasonable. On the other hand, the overhead can be specified by giving the *pdf*, and in transmitting the data in any other system there is *some* convention of coding employed. If I send it all in base 10, I ought to send also a translation of the ten digits in binary and the convention for using decimals if I want to compare like with like. I need to say something about the intended precision and the number of data points, too, so you will not need separators in the code to tell you when one stops and another starts. These issues apply to both the uniform coding and the non-uniform coding, so neglecting the overhead is unreasonable only if you take the view that there is a unique way to code real numbers and it presupposes a uniform density function.

It is plain, on a little reflection, that sending a list of N real numbers in the interval $[0,1]$ by a process such as the above to a precision of 2^{-P}, for positive integers N and P, will require, if we use the uniform *pdf*, NP bits. If we have a *pdf*, $f(x)$, defined on the interval and replace it with an approximation uniform on intervals of size 2^{-P}, and if we suppose that the model is a good representation of the distribution of the data, then in any interval on the number x, the fraction of data points in that interval will be $Nf(x)(2^{-P})$ and they will be able to be coded using $\log_2(\frac{1}{f(x)(2^{-P})})$ bits each, i.e. $P + \log_2(\frac{1}{f(x)})$ each, so the total number of bits will be

$$\sum_{i=1}^{2^P} \frac{N}{2^P} f(i)(P + \log_2(\frac{1}{f(i)}))$$

Since $\sum_{i=1}^{2^P} f(i) = 2^P$ this simplifies to:

$$NP + N\sum_{i=1}^{2^P} f(i) log_2(\frac{1}{f(i)})\frac{1}{2^P}$$

which in the limit for a finer and finer approximation to an integrable function f is:

$$NP + N\int_0^1 f(x)\log_2(\frac{1}{f(x)})dx$$

If we compute $\int_0^1 f(x)\log_2(\frac{1}{f(x)})dx$ for the *pdf* of the example, we get :

$$\left(\frac{1000}{3.7}\frac{1}{1000}\log_2(\frac{3.7}{1000})\right)+\left(\frac{100}{3.7}\frac{9}{1000}\log_2(\frac{3.7}{100})\right)+\cdots+\left(\frac{1}{3.7}\frac{9}{10}\log_2(\frac{3.7}{1})\right)$$

which comes out as

$$-2.183313 - 1.156945 - 0.348909 + 0.4591278 = -3.2300396$$

which multiplied by the number of points, 5, gives -16.1502, a fair approximation to the number of bits saved by our choice of *pdf*. Purists might say I have been overly generous to myself here and there, but I merely wish to illustrate the principle.

The formula

$$NP + N\int_0^1 f(x)\log_2(\frac{1}{f(x)})dx$$

or equivalently

$$NP - N\int_0^1 f(x)\log_2(f(x))dx$$

is of some use in estimating potential savings. It should be noted that all the savings come from the case where the value of $f(x)$ is greater than 1, since here the contribution is negative so there is a saving. It is essential to remember that this is the minimum possible bit length for representing the data *on the assumption that the model is the best possible probabilistic model for the data.*

Everyone will, of course, recognise the expression for the integral as defining the *entropy* of the *pdf*, f. If we use natural logarithms instead of logarithms to base 2, we merely change the units of measurement from bits to *nats* (or for some, less serious writers, *nits*).

If instead of working over the interval $[0,1]$ we were to work over, say, $[-1,1]$, we would need an extra bit to determine the sign of the number. On the other hand, if we are prepared to work to the same *relative* precision, we can give up a bit on the location, since knowing where things are on a region of twice the size means we can go to fewer binary places. It really rather depends therefore on what we mean by the precision of specification of a point. This in turn hangs rather on why we want to send a data set in the first place.

If you ask me to measure the diameter of some coins and to send you the results, I might give results in centimetres or inches, and it is likely to matter which. So the choice of a unit is vital and I have to send this information somehow. If you received a value in centimetres of 1.823456789, it is idle to pretend that you are going to care equally about the different digits. You will care, perhaps about the first digit, the 1, more than the second digit, unless as a result of having some crude measurements yourself you have already guessed the first will be 1. And you are likely to care less and less about succeeding digits until you don't care at all about the digits past some point. In fact you probably wouldn't believe them. If you discovered I had been using interferometric methods to get the eighth and ninth decimal places, you might think I was pretty daft, because those extra decimal places cost money and effort to obtain and normally do not carry a good deal of information, in the sense that you probably don't care too much about the values. Let us assume that you really do need the answers correct to one part in 2^P. If some third party were to choose units which were half the size of mine, he would be producing answers which were numerically twice as large, and in order to give the same precision relatively, he would need one less binary place than me. My interval of points will be, say between 0 and 1, his will be between 0 and 2, and if our intervals are divided up into 2^P subintervals, his will be twice as big as mine to convey the same amount of information. This has to be taken into account when specifying a *pdf*, f, because part of the specification is the domain space.

Suppose I consider a set of points given by a triangular *pdf* on the interval $[0, 1]$,

$$f(x) = 4x \; if \; 0 \leq x \leq 0.5; f(x) = 4 - 4x \; if \; 0.5 \leq x \leq 1$$

Suppose we choose to specify N points each to P binary places. I shall measure the information in *nats* to make the integration simpler, where 1 bit $= \ln(2) \approx 0.6931472$ nats

Then the best possible compression on this model is

$$NP\ln(2) - N\int_0^{\frac{1}{2}} 4x\ln(4x)dx - N\int_{\frac{1}{2}}^1 (4-4x)\ln(4-4x)dx$$

nats.

This becomes

$$NP\ln(2) - N\frac{2}{4}\left[\frac{u^2}{2}\ln(u) - \frac{u^2}{4}\right]_{u=0}^{u=2}$$

which is $NP\ln(2) + \frac{1}{2}N(1 - 2\ln(2))$ nats. Which is a rather small saving.

Now suppose the measurement units are changed so that the interval is halved. The required precision to specify the data is increased to $N(P+1)\ln(2)$ nats. The function $f(x)$ now becomes

$$f(x) = 16x\ if\ 0 \leq x \leq 0.25; f(x) = 8 - 16x\ if\ 0.25 \leq x \leq 0.5$$

This still has to have integral 1 over the interval which is half the size. The total saving then comes out to be $N(P+1)\ln(2) + N\frac{1}{2}(1 - 2\ln(4))$ nats. So we have lost, for every point, one bit on the first term, but gained it on the second, giving no net change. It is easy to persuade yourself by a generalisation of this reasoning, that linear scaling does not change the compression resulting from using the model f, which is moderately comforting.

It follows that for a model defined over any bounded interval $[a, b]$ of $\mathbb{R}$, it is safe to use the formula

$$NP + N\int_0^1 f(x)\log_2(\frac{1}{f(x)})dx$$

as an estimate of a bound on the compression attainable by using model f, provided we simply change the limits of the integral and bear in mind the implicit conditions on the precision with which we expect to specify the data points.

Intuitively, one would expect that there would always *be* a saving, that the worst case is when the function f is constant when the saving is clearly zero. In other words, reflecting on the function f as distorting the density of points in the space and on coding this information suitably as a way of expressing the data more efficiently, leads one to the belief that $\int_a^b f(x)\log(f(x))dx$ will be non-negative for any function f such that $\int_a^b f(x)dx = 1$, and for any logarithm base. This is indeed the case, and I shall now prove it.

Proposition 3.1 *If data is generated in a bounded interval with density given by a pdf f and if the pdf is used to code the data, then for* *sufficiently large amounts of data f may be used to compress it except when the distribution is uniform.*

Proof By the earlier observations, without loss of generality, we may suppose the interval is $[0,1]$. First, if the interval $[0,1]$ is divided up into M equal subintervals and used to represent a finite probability distribution $\{p(i) : 1 \leq i \leq M\}$ it is easy to see that the expression $-\sum_{i=1}^{M} p(i) \log p(i)$ is maximised when $\forall i, p(i) = \frac{1}{M}$. It has a value in this case of $\log(M)$. It may be minimised by making any one of the $p(i)$ 1 and the rest zero, when the minimum value of 0 is attained. If we transfer this to a function f on the unit interval, we have the same result for $f(i)\frac{1}{M} = p(i)$. Thus the value of $\sum p(i) \log(p(i))$ lies between $-log(M)$ and 0:

$$- \log(M) \leq \sum_{i=1}^{M} p(i) \log p(i) \leq 0$$

and since we can model this by the piecewise constant function f with $p(i) = f(i)\frac{1}{M}$,

$$- \log(M) \leq \sum_{i=1}^{M} f(i) \log(\frac{f(i)}{M}) \frac{1}{M} \leq 0$$

which is to say

$$- \log(M) \leq \sum_{i=1}^{M} f(i) \log(f(i)) \frac{1}{M} - \sum_{i=1}^{M} f(i) \log(M) \frac{1}{M} \leq 0$$

which becomes

$$- \log(M) \leq \sum_{i=1}^{M} (f(i) \log(f(i))) \frac{1}{M} - \log(M) \leq 0$$

i.e.

$$0 \leq \sum_{i=1}^{M} (f(i) \log(f(i))) \frac{1}{M} \leq \log(M)$$

And in the limit, when we replace the sum by an integral and the $\frac{1}{M}$ by dx we get

$$0 \leq \int_0^1 f(x)\log(f(x))dx < \infty$$

The zero value is attained only for the uniform distribution. □

The cheerful, and some would say irresponsible, treatment of limits will give a severe case of the uglies to analysts, who may divert themselves by showing at length how it should be done.

It is easy to see that the value of $\int_0^1 f(x)\log(f(x))dx$ can be made arbitrarily large by concentrating f at some point. The wonderful Dirac delta function would allow an infinite amount of compression, if it only existed. Of course, if we agree to consider finite precision in the specification of the data, it makes little sense to worry overmuch about such things as limits or Dirac delta functions, they are accidents of the language to which we have become committed for historical reasons that no longer apply. An obsession with adventitious aspects of the linguistic apparatus makes sense only for lunatic Platonists. [11]

It is plain that you never get the total number of bits reduced below zero, but it nevertheless follows that *any pdf* on a bounded interval which is not constant does some good in allowing compression *of a data set described closely by the pdf.* Moreover, any reasonable scheme for compressing a set of real numbers in a bounded interval into binary strings, which makes sense before you know what the numbers are, must assign binary strings differentially to intervals and hence determines a *pdf*.

If we want to work over the whole real line, the situation is somewhat more complicated. For points that are a long way from the origin, the number of bits necessary to get just the integer part goes up without bound. This, of course, means that it makes sense to shift the origin to

[11] Of whom there are a fair number around. No doubt a cure will be found in due course.

where the points are and store the shift; this amounts to subtracting off the centroid and this obviously gets you some compression, although it is less than clear what model this corresponds to.

It is easy to compute $\int_{-\infty}^{\infty} f(x)\ln(\frac{1}{f(x)})dx$ for the case where f(x) is the gaussian or normal function $g_{[0,\sigma]}(x)$ and obtain $\ln(\sigma\sqrt{2\pi e})$. This is negative when $\sigma\sqrt{2\pi e} < 1$, i.e. when $\sigma < \frac{1}{\sqrt{2\pi e}}$. The conclusion is that for fat gaussians there is no advantage to be gained over just sending the data, and a fat gaussian is one where $\sigma > \frac{1}{\sqrt{2\pi e}} \approx 0.2419707$. This is quite different from the case where the *pdf* has compact support. It is easy to persuade yourself that the trouble is that unbounded intervals provide you, for any given size of data set, some residual probability of occurrence of a datum that nobody would believe for a minute belongs in with the rest of the data set. Pragmatic treatments of outliers has almost always been to pretend they didn't occur, but rationales for this conduct tend to lack something.

A gaussian *pdf* for the height of adult males fits the data quite well over several standard deviations, but the total number of male adults of negative height will always be zero, no matter how large the population. The reduction of the overhead in specifying f in a simple formula will have a cost in compression.

In practice, one can cheat quite sensibly by putting the same number of bits available for coding every data point past some distance from the origin, which means that for points which are very far from the origin, there may be too few bits to specify them at all- a sort of overflow condition. This is equivalent to making a decision to simply ignore some data if it is simply too far from the rest. Alternatively, I can decide on the precision I need by looking at the biggest distance from the origin of any of the data points, or I can have different precisions for different points and in this case I need a *pdf* on $\mathbb{R}$ which falls off as the number of bits needed to specify the integer part. This may be done rather neatly: See Rissanen's *universal pdf* on $\mathbb{R}$.

The fact that we get a sensible scale invariance for finite precision on compact intervals and no such result when the precision is allowed to become infinite and the domain of the *pdf* unbounded, suggests that the simplifications which we introduced into our lives many years ago in order to have very general theorems (of a rather impractical sort)[12] may have been a horrible mistake. Maybe we should never have messed with infinite sets at all. Certainly we might argue for finite precision results, since nobody except a lunatic or a platonist philosopher thinks he is going to measure anything *exactly* with a ruler or any other sort of meter. And there is something metaphysical about unbounded *pdf*s in that no finite amount of data is going to demand them, and finite data is all we'll ever get.

My solution to the problem is going to be to choose some number N of standard deviations of a gaussian and use it to compress data within N standard deviations of the centre. If I ever get a datum outside my limit of N standard deviations, I shall simply ignore it. I shall choose N so that I get some compression from using the gaussian. How much compression I require will depend on how sober I am and other such matters contingent on the problem details.

This is $\mathcal{SIN}$ to platonist philosophers, and may give them conniption fits, which is good for them and stimulates their livers.

The extension to higher dimensions than one is essentially straightforward; in dimension n you need to send n numbers instead of just 1. The reader is invited to decide whether it is ever worth storing multivariate gaussians. Note that provided the entropy of the *pdf* is negative, then there is some number of points at which the saving beats the overhead on sending the *pdf*. Which we have not, so far, considered.

[12]The Intermediate Value Theorem tells you that if you have a continuous function defined on an interval which takes values which are positive at one location and negative at another location, then it must be defined at some other location where it takes the value zero. This codifies an intuitive picture of the real line as resembling a ruler made out of chewing gum and a continuous function as stretching it but not tearing it, and it means that when we want to find the zero of a function by some iterative process, we have usually got grounds for thinking it is there to be found. The price paid for this simplicity and adherence to naive intuitions includes the uncountability of the reals and non-constructive existence proofs which give confidence only to the more naive Applied Mathematicians.

3.5.4 Summary of Rissanen Complexity

Suppose we have two models for some set of data; we may think of the data as coming in sequentially for our purposes, in which case we can imagine the two models, q_1 and q_2, as predictors. The amount of information, which I called the *surprise*, which the data supplies to each model, is simply the sum over the data set of the logarithm of the improbability of each datum, as assessed by each model. Then all other things being equal, the less surprise or information supplied by the data to the model, the better the model. In the extreme case where one model is quite certain of its predictions and always correct, the data supplies no information at all. In the other extreme case the information supplied on each occasion is the logarithm of the number of possible states the datum point can occupy.

We suppose that there is some genuine random process specified by a *pdf*, p operating to produce the data, and that the model we have is represented by q. The sum $\sum_{\mathbf{x}\in D} p(\mathbf{x}) \log_2(\frac{1}{q(\mathbf{x})})$ gives the idealised value of the information supplied to the model q by data generated in fact by process p. This is larger than $\sum_{\mathbf{x}\in D} p(x) \log_2(\frac{1}{p(x)})$ the idealised value of the information supplied to the model by data generated from the model, the entropy of p, except in the case when $p = q$. The difference,

$$\sum_{\mathbf{x}\in D} p(\mathbf{x}) \log_2(\frac{1}{q(\mathbf{x})}) - \sum_{\mathbf{x}\in D} p(x) \log_2(\frac{1}{p(x)}) = \sum_{\mathbf{x}\in D} p(\mathbf{x}) \log_2 \left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right)$$

known as the *Kullback-Leibler* distance, is therefore always non-negative, and is used as a natural measure of the goodness of fit of one *pdf* to another. It is not symmetric. In our case we are obliged to estimate it since we do not know p, and we can estimate it only up to the unknown entropy of the actual distribution p. At least this allows us to compare two *pdf*s q_1 and q_2 on a given data set to see which is better.

It is fairly easy to see that if the two models q_1 and q_2 were predicting repeatedly from a finite list of events, and if they were to gamble with each other by making bets based on their separate predictions, using their probabilities to establish the odds for the bet and the stake size in any one of a number of sensible ways, then the better predictor, in the information sense, will make money from the worse one. The result in practical terms is simply to maximise the log-likelihood of the data for each model. If instead of two models we have a manifold of models, we simply maximise the log-likelihood over the manifold.

If other things are not equal, we may choose to penalise a model which has more parameters than another by some rate of exchange between the mean information supplied to the model by the data per data point, and the number of parameters in the model. The essential reason for this is our desire to use the model to predict what is going to happen next, and our confidence in the ability of our model to extract information from the data set maximally, while not being unduly affected by noise, is at issue. Occams Razor is the doctrine 'Entities are not to be multiplied unnecessarily', or more popularly: 'Given two theories, both of which account for the data, choose the simpler'. Many centuries after William of Occam, we may be in a position to say it in algebra with some precision instead of in Latin.

Why *should* we penalise a model with more parameters? One's intuitions, if one has messed around with models and data to any serious extent, should tell one that estimating too many parameters from too little data is unlikely to give anything very effective at predicting what is going to happen next. It seems reasonable that this state of affairs should be formulated as a penalty for number of parameters, the only question that remains to be solved is the rate of exchange between log likelihood (which will almost always improve for more parameters) and the number of parameters themselves.

The ideal rate of exchange would be determined by testing the model on later data to see how well it performs. After all, that is what we want models for[13]. The measure of better performance ought to be the same as the measure above; the amount of information supplied to

[13]Well, it's what *I* want models for. If you have some alternative use for them, good luck to you.

the model by later data, generated by the same process that gave the original data, will be less for the better model. What grounds have we for believing that the model with fewer parameters will do rather better than expected, while the model with more parameters will do rather worse? The model with too many parameters will be trying to model the randomness of any particular data set as if it were significant, when it isn't. It has actually extracted too much information from the data, finding pattern and structure when there isn't any. This is easily done; brains do it when they see butterflies or body parts in Rorschach blots, faces in flames and so on. This line of thought can lead to the *Akaike Information Criterion.* Here, we ask the question, if the data is a sample, D, generated by a *pdf* p and if we model it by a *pdf* q, what is the expected log-likelihood of the data relative to q? It turns out that in many cases the maximum log-likelihood choice of q is a biassed estimate of the mean expected log-likelihood of p itself, and subtracting off the number of parameters removes the bias. See Akaike and also Sakamoto, *et al.* for an elucidation of these somewhat technical points. See also Shibata.

Alternatively, we can argue that since we know nothing of the model class which actually generated the data, we should abandon this attempt and go to more robust considerations. We can argue that compression is pertinent.

A *pdf* over $\mathbb{R}^n$ can be regarded as a method of compressing a set of points in $\mathbb{R}^n$. In doing this we simply think of the *pdf* as telling us how much to stretch bits of the space or alternatively to re-measure it so that we can code points efficiently, just as with Shannon coding. The amount of compression is greatest when the *pdf* is a good model of the distribution of the set. We can turn this on its head and define a 'good model' as one which gives good compression. Certainly for two different models of the same family, the better model is the one with greater log likelihood. The 'same family' of models means that there is some finite set of parameters which specify the model within the family, which is to say the 'same family' of models comprise a manifold. So far, then, this is merely a new way of writing Maximum Likelihood models. But in the case where the class is larger than this and comprises, say, several manifolds with different dimension, we can also code the model parameters, and store the extra *overhead.* This allows us to compute the total length of the data as coded using a model, plus the overhead of specifying the model. It is not particularly difficult to compute, for such things as gaussian mixture models with different numbers of gaussians what the net saving is. There is an advantage in having a better model, offset by the drawback of having to count extra parameters. Penalising the number of parameters is an obvious thing to do, but Rissanen offers a coherent explanation of precisely how to do this. The Rissanen approach, under certain circumstances, agrees with the Akaike approach asymptotically. Some might find this comforting, others mysterious.

Other work has been done from a Bayesian point of view on attempting to put in an Occams Razor, that is to say a device for penalising a large number of parameters or entities. The idea here is that we can penalise a model family of higher complexity, that is to say with more parameters, by observing that the normalising constant $\int p(x|m)p(m)dm$ will be bigger when the integration is over a higher dimensional manifold of models m thus reducing $p(m|x)$ in comparison with some alternative and 'smaller' such manifold. The fact that the integral depends on the parametrisation of the manifold of models is grounds for some degree of nervousness among the innocent, but one can be persuaded this is not fatal. The result is that we favour a model from a class when the particular model accounts for the data well, when we have no strong prior prejudice against the model, and when on average the other members of the model class do fairly badly, possibly because there are so many of them. This has a certain charm. See MacKay's Doctoral Thesis in the bibliography for a well written propaganda job and some good references.

In the case of gaussian mixture modelling, the method implicitly recommended in the next chapter for dealing with pattern recognition problems if we opt for statistical methods, the choice of how many gaussians to use can be solved using the Akaike Information Criterion (AIC), the MDL or stochastic complexity criterion of Rissanen, or others, notably the BIC. Experiments, conducted on data sets actually generated by a known number of gaussians, suggest that the Rissannen criterion works tolerably well over a range of dimensions and numbers of gaussians, and performs better than the AIC. A rough guess of a sensible number may often be made by eyeballing the data under projection provided the dimension is not too high, and more sophisticated methods of automating the eyeballing to some extent will occur to the reasonably ingenious. Then some experiments with different numbers of gaussians and some arithmetic along the lines indicated above can yield a sensible number of gaussians to use in the model.

I shall return to the matter of compression and optimal modelling when we come to Neural Net models, which are notorious for having enormous dimensions for representing the data and consequently large numbers of parameters.

3.6 Summary of the chapter

The basic assumption behind the probabilistic analysis of data is that there is some process or processes operating with the property that even when the input to the process is more or less known, the outputs differ. Rather like the drunk dart throwers of chapter one, the process *ought*, in a neat newtonian, causal, clockwork universe, to replicate its last output this time, but it generally does not. The classic examples, and the origins of probability theory, are in playing cards, throwing dice and tossing coins, where although we have a convincing causal model (Newtonian Dynamics) for the process, our ignorance of the initial states leaves us only 'average' behaviour as effective data in deciding what to expect next.

The assumptions of probability theory are that when you replicate an experiment or repeat something like tossing a coin or throwing a die, what you actually do is to reapply a map with some generally different and unknown and indeed unknowable initial state. All we know about the initial states is the measure of those subsets of them that lead to different sets of outcomes. We can simplify then to having a measure on the space of outcomes. From this model of random processes as non-random processes but with ignorance of the precise initial state, all of probability theory derives, and on this basis, much modern statistics is founded. We rapidly[14] deduce the existence of probability density functions over the space of outcomes as measuring the results of repeating the application of the random variable an infinite (usually uncountably infinite) number of times. When there are only finitely many possible outcomes, we assign a probability distribution to the elementary outcomes. The practicalities of trying to decide whether one state of affairs is a repetition of another state of affairs are ignored on the grounds that it is too difficult to lay down rules. So what a replication *is*, nobody knows, or them as knows ain't saying. To highlight this point, consider a robot programmed with classical mechanics required to throw or catch a cricket ball or some similar mechanical task.[15] It might be a programming problem, but the principles are clear enough. Now contrast this with the case of a robot which is intended to play a reasonable game of poker. To what extent is the information acquired in one game transferable to another? This is the problem of replication; it involves what psychologists call 'transfer of learning', and we do not understand it. When we do, you will be able to play a poker game with two human beings and a robot, and the robot will catch you if you cheat. Don't hold your breath until it happens.

Model families, often parametrised as manifolds, spring out of the collective unconscious whenever a suitable data set is presented, and there are no algorithmic procedures for obtaining them. Again, this is bad news for the innocent, such as me, who wants to build a robot to do it. There are procedures for taking the manifold of models and choosing the point of it which best fits or describes the data. If there were only one we might feel some faith in it, but there are several. There is no consensus as to which procedure gives the best model, because there are competing ideas of what 'best' means. There are three well

[14] Well, fairly rapidly. Doing this with axiomatic precision requires careful avoidance of pathologies which trouble the sleep of Pure Mathematicians but which most people wouldn't think of. I shan't ask you to think of them either.

[15] Robots *are* programmed to do similar sorts of things, e.g. shoot down enemy aircraft, and they do it tolerably well.

known, and occasionally competing, procedures for picking, from some set of models, the best model for a given data set, and they all demand that someone, somehow, went and worked out what the choice of models had to be. (Take a model, any model...). These three procedures, which we have discussed at some length, give rise to the Maximum Likelihood model, the Bayesian optimal model and the Minimum Description length model. Sometimes these are all different, sometimes not. Arguments to persuade you that one is better than another are crude, heuristic and possibly unconvincing. Tough luck, that's the way the subject is.

Thus we face some rather hairy problems. The state of the art is something like this:

We cannot in general evolve a model of any useful sort from the data as yet, we have to rely on people looking at the data, and then inspecting the entrails of chickens or gallivanting about in their subconsciouses or wherever, and bringing back a model family.

When we *have* a set of models described by finitely many parameters so that the models comprise a smooth manifold, there are several different ways of picking the best one from the set. We can compute a figure of merit for the pair consisting of a model and a set of data which the model might or might not have generated. This figure of merit is a real number called the *Likelihood*, and hence for a fixed data set there is a function defined from the manifold of models into $\mathbb{R}$. This function has a unique maximum often enough to make it tempting to use Maximum Likelihood as the natural meaning of the 'best' model to account for the data. However, this pays no attention to the possible existence of other information which might predispose us to some other model; in particular, there might be a (prior) probability density function associated with the manifold of models. This, many people might feel in their bones, ought to be used in choosing the 'best' model.[16] Where the prior *pdf* comes from is a matter seldom discussed, but presumably it comes from some other source of data about the system under in-

[16]There is altogether too much of people feeling things in their bones in this business.

vestigation: if we are into coin tossing then presumably it derives from having tossed other, different but similar, coins in the past.

Finally, if we have feelings in our bones about information theory as the right place to found statistical reasoning, and if we also feel in our bones a preference for simple models rather than complicated ones, we may be able to fall back on Rissanen style arguments if we are lucky, but many statisticians don't accept Rissanen's ideas. Rissanen gives us a chance to reject models which give a high likelihood but seem to be too complex, and to prefer simpler models with a lower degree of likelihood for the data. I discussed philosophical issues, I drew morals and extracted, some would say extorted, principles. I then went on to *pdf*s, starting with a coin model, and showing how the ML model for the coin compressed the results of tossing it. Then I compressed a set of points in the unit interval, using a *pdf* over $[0, 1]$. Various useful and intuitive results were proved in a manner that no modern analyst would tolerate but that was good enough for Gauss.

The subjectivity which allows Statisticians to spend uncommon amounts of time in disputation means that choosing the answer that pleases you most is what is commonly done.

One should not be deterred from considering a theory just because of the air of dottiness about it which appears once one has stripped away the technicalities and described it in plain English. Theories are to be judged by their consequences, and on that criterion Probability Theory has been extremely successful. There are, nevertheless, quite serious problems with the semantics of the theory. For a start, the repetitions of the application of the *rv* mean that the different initial states have to be equally likely, but this is part of what we are trying to define by the apparatus of random variables. In applications, Rissanen has pointed out that there is no operational procedure for deciding if something is 'random', and experts argue over the legitimacy of various techniques while the trusting just use them. Formalisation of these ideas via the Kolmogorov axioms has meant that while the Mathematics may be impeccable, it isn't at all clear how reality gets into the picture. The innocent engineer who wants recipes not rationales can be a party to his own deception and often has been.

You can see why so many people dislike Stats. Just as you've finished learning some complicated methods of coping with uncertainty and doubt and finally come to believe you've got a stranglehold on the stuff, it leers at you and tells you it's all a matter of opinion as to whether it's OK to use them. For the innocent wanting certainty, it isn't as good a buy as religion.

3.7 Exercises

These little exercises are of the simplest possible sort, and involve mostly tossing coins. They should therefore be easy for anyone who has even the smallest knowledge of probability theory. Try them on your friendly neighbourhood probabilist if you can't do them yourself.

1. You are asked by a friend to show him the coins you have in your pocket or purse and you find two ten cent coins. He selects one of them, and then proceeds to toss it ten times, obtaining eight heads and two tails in some order. He then offers to bet on the results of tossing the coin again. Feeling that the data is too restricted to allow you to estimate odds sensibly, you toss the other coin one thousand times (rather quickly) and keep count of the results. You get 503 Heads and 497 Tails, but you note that the variation if you partition them into groups of ten is considerable and accords well with the predictions of a binomial model with $p(H) = 0.5$. Present a careful argument to determine what odds you would offer your friend in a bet for a dollar on the next throw of his coin coming heads. Repeat for the case where the bet is one hundred dollars. Would it make a difference if the friend was called Persi? If you were allowed to be the one to toss the coin? Could the order of the results make a difference, for example if the two tails came first? Last? What about the possibility that the number of throws was chosen not because ten is a nice round number but

because your friend threw two tails at the end because he was getting tired and thought that two bad throws meant it was time to stop?

2. The same situation as in the last exercise occurs, but this time you have one ten cent coin and one twenty cent coin. Your friend takes the ten and you do your experiment with the twenty. Do you feel that this makes no substantive difference to the process of throwing your coin one thousand times and applying the result to expectations about *his* coin? Would there be some measure of relevance which you were using implicitly? How about if there was one ten cent coin and (a) a saucer (b) a ten dollar bill and (c) a small piece of green cheese, in your pocket? Would you feel that the relevance of tossing the ten dollar bill in the air so small as to be useless, but a twenty cent coin might give you some information? If so, explain how to program a robot to calculate the relevance. What would you do with the green cheese?[17]

3. Suppose a martian or some visitor from a distant planet, or maybe an intelligent octopus, were to be watching you and your friend, and suppose the kibitzer had never seen a coin tossed before or anything remotely like it. Would it, do you feel, be inclined to feel that tossing the second coin or possibly the ten dollar bill might be a good idea? Could it conceivably examine the dynamics of the processes and make any inferences as to relevance? If any of the parties present were a subjective Bayesian, would he have have any explanation of where the kibitzer got his, her or its prior from before it did the experiment with a different coin?

4. You are given 2000 points in the unit interval to a precision of 10 bits. There is thus quite a good chance[18] that several of the points will coincide, up to the given precision. Suppose they have in fact been generated by a uniform distribution; can one expect, on average, to compress them by rather better than by sending 20,000 bits, how might one do this, and ought one to do it?

[17] Do *not* try to use it to prove Kraft's inequality.
[18] Actually, probability 1.

5. You work to one bit precision on the unit interval and you have three data points, $\{0, 0, 1\}$. You wish to find a good model for the data. The maximum likelihood model says that it's twice as likely to be a 0 as a 1. You note however that with three data points the only possibilities are three 0s, two 0s and a 1, two 1s and a 0, and three 1s. Since you wish to be able to get good predictions on the next 1000 points, it seems absurd to restrict yourself to one of a class of four models, the only ones maximum likelihood *could* provide you with, if you assume a binomial model family to start from and have only three data. You therefore use bayesian methods and assume a uniform prior on the grounds that it contains least prejudice about the way things are. Try the calculation and see how much good it does you when you take the MAP estimate. Are there any grounds for a prior prejudice in favour of a binomial model with equal probabilities for the two outcomes? If the data came from tossing a coin and assigning a zero to Heads and a 1 to Tails, would you alter your answer to the last part? If the bits represented polarisation of a signal from outer space, would you feel differently about things? Do you think learning a bit of Physics might help in the last case? Just how ignorant *are* you?

6. Suppose that in the last example, the data comes in sequentially after the first three arrive in a lump, and that we proceed by updating our estimate of the posterior distribution continuously. This procedure amounts to getting a sequence of points in the space of posterior distributions, which bears a passing resemblance to the unit interval. If the data were in fact generated by a binomial process with probability of getting 0 equal to p, there ought to be some grounds for thinking that the sequence will almost surely converge to this value. By formalising what the terms mean, prove that this is indeed the case.

7. With the assumptions of the last question, how many points are needed in order to justify the use of the appropriate binomial model as a compression device?

8. If the sequence in fact consists of the bit sequence consisting of $(0, 0, 1)$ repeated indefinitely, how many points are required before the repetition model beats the binomial model? How many would *you* need?

9. Suppose that you have a gaussian mixture model of some data, where the data was in fact generated by simulating k gaussians in the plane. Are there circumstances in which you would expect a really effective method for penalising too many coefficients to (a) underestimate or (b) overestimate k? In other words, could underestimating (overestimating) k ever be the sensible thing to do?

3.8 Bibliography

1. Albert Morehead and Geoffrey Mott-Smith, *Hoyles Rules of Games* New American Library, 1983.

2. Jorma Rissanen, *Stochastic Complexity in Statistical Inquiry* World Scientific Pub.Co. 1989.

3. Jorma Rissanen, *Stochastic Complexity* J.Roy.Statist.Soc.**B** (1987)**49**, No. 3, pp223-239 and 252-265.

4. Jorma Rissanen, *Stochastic Complexity and the Maximum Entropy Principle*, Jorma Rissanen, IBM Almaden Research Center, K52/802, San Jose, Ca. 95120-6099, USA.

5. Peter Walley, *Statistical Reasoning with Imprecise Probabilities*, Chapman and Hall, 1991.

6. E.T. Jaynes, *Papers on Probability, Statistics, and Statistical Physics* (Ed. R.D. Rosenkrantz. Kluwer Boston, 1983.

7. E.T. Jaynes *Probability Theory: The Logic Of Science* ftp site : bayes.wustl.edu on subdirectory *Jaynes.book*.

8. R. Thom, *Structural Stability and Morphogenesis* Translated from the French ed., as updated by the author, by D. H. Fowler. W. A. Benjamin, 1975.

9. Ralph Abraham, *Foundations of Mechanics* Benjamin, 1967.

10. Gregory Chaitin, *Information, Randomness and Incompleteness* World Scientific 1987.

11. Gregory Chaitin, *Randomness and Mathematical Proof*, **Scientific American**, 232 pp 47-52, May 1975.

12. Irving John Good, *Good thinking : the foundations of probability and its applications* University of Minnesota Press, 1983.

13. Irving John Good, (Ed.) *The scientist speculates / an anthology of partly-baked ideas.* London : Heinemann, 1962. Here for fun.

14. Irving John Good, *The estimation of probabilities : an essay on modern Bayesian methods* Cambridge, Mass : M.I.T. Press, 1965.

15. Claude Shannon *A Mathematical Theory of Communication* Bell Systems Technical Journal, **47** pp143-157.

16. Claude Shannon and W. Weaver *The Mathematical Theory of Communication*, Univ. of Illinois Press, 1949.

17. Thomas Cover, Joy Thomas *Elements of Information Theory* Wiley 1991.

18. Y.Sakamoto, M Ishiguro and G. Kitagawa *Akaike Information Statistics* D. Reidel 1986.

19. H. Akaike *A new look at the statistical model identification* IEEE Trans. Aut. Control. **19** pp716-723 1974.

20. G. Schwartz *Estimating the dimension of a model* Ann. Stat. **6** 2, pp461-464 1978.

21. R. Shibata *Asymptotically efficient selection of the order of the model for estimating parameters of a linear process.* Ann. Stats. **8** pp147-167 1980.

22. David J.C. MacKay *bayesian Methods for Adaptive Models* (Doctoral Thesis from CalTech) 1992.

23. T.P. Speed and Bin Yu *Model Selection and Prediction: Normal Regression* Ann. Inst. Staist. Math. **45** No.1, pp35-54. 1993.