Human Issues in the use of Pattern Recognition Techniques

Alan Dix *

October 1991

1 Introduction

The purpose of this chapter is to emphasise that when including neural nets or similar techniques in systems with a human component, the technological issues are far easier to address than the attendant human ones. It highlights the need for a thorough theoretical understanding of the behaviour of the computer-based techniques in order to be able to assess the human consequences of their use.

The chapter focuses on two applications of pattern recognition. One is an innovative example based method of query construction and the other is the more established use of neural nets for routine decision making such as credit vetting.

In the latter example the 'user' of the system is seen as not just the operative who directly uses the computer, but also the client who is the target of the process. This wide view of human-computer interaction means we have to deal not 'just' with the usability of systems but also the entailing ethical and legal responsibilities.

Range of systems covered

This chapter concerns the use of example based or taught pattern recognition techniques. This includes most neural net or connectionist approaches and also inductive learning. These techniques all operate by being given a set of examples and from them generalising to unseen data. They are essentially

*work funded by SERC Advanced Fellowship B/89/ITA/220

black-box techniques in that the user of the algorithms is not expected to peer too deeply into their workings, but simply to give the examples and accept the outputs. I realise that not all pattern recognition systems fall into this category, but for want of a better term, I shall simply say 'pattern recognition' as a blanket term for such techniques.

Range of applications

This chapter is considering only a particular class of applications which incorporate pattern recognition techniques. Specifically it is those systems where the intelligence is 'up front'. This would include intelligent front-ends, adaptive interfaces, and some natural language interfaces. The two examples dealt with in the chapter, an intelligent data base query mechanism and automated credit vetting are specific examples.

We are not concerned with systems where the underlying computation technique is hidden. That is, if it is completely hidden. In these cases the user is unaware of the particular characteristics of the technique and is interacting purely with the results of the technique. Examples where this might be the case are speech, image and handwriting recognition. A more specific case would be an industrial robot which uses a neural net to identify chocolates on the production line for automatic packing.

One might initially think that it is the latter systems, which lie outside the scope of the chapter, which are more prevalent. However, it is rare that the underlying computation mechanism lies completely hidden in any system. Anyone who is versed in software production will have seen odd behaviour in systems and been able to trace it to the particular implementation techniques – the underlying algorithms have a way of 'bleeding out'.

This 'bleeding out' is especially prevalent when the system is operating at the boundaries of its specification or when errors occur. In the case of the industrial robot, we may want to be certain of its behaviour if a misshapen chocolate, or even dead mouse, should happen to pass. Unfortunately it is often these breakdown situations which are most critical.

So this chapter should not be read as a blanket critique of pattern recognition systems, but it does have an extensive domain of application.

Structure

The central focus of the chapter is on the two examples. Each problem area will be presented, a solution suggested making use of computer pattern

recognition techniques, and finally the various human issues, of usability and ethics, discussed. After looking at these examples, we shall summarise some of the lessons we have learnt from them. Finally, we shall compare the analyses of pattern recognition with traditional computation and analysis techniques, including statistical methods and human reasoning. Do they suffer from similar problems, and if so can the accumulated experience with these established methods guide us in our use of more recent techniques.

2 Application – Query by Browsing

Relational Query By Example (RQBE) is now a standard feature of many data base systems. However, its 'by example' nature is only partial. The emphasis is still on the query itself, the process is still one of first constructing a query then getting a list of the selected records. If the listing is not what was expected you have to go back and rethink your query. RQBE supports query construction, but not the whole querying cycle.

Query by Browsing aims to address this by focussing on the list of required records rather than the query itself. The interaction will then be more 'direct' as it focusses on the goal, the listing, rather than the means of of achieving the goal, the query.

Concept

As with RQBE, Query-by-Browsing starts with a template of the listing with the headings of the various columns and their association with the database relations specified. However, unlike RQBE which presents the user with an initially blank screen, Query-by-Browsing fills in the listing completely with all the records in the database.

The user then goes through this listing marking those records that are of interest and which should be in the final listing and rejecting those which aren't. In Figure 2 we have a simulated screenshot, the user's interest is indicated by a tick and rejection with a cross. After a while the system guesses what the user's criterion is and highlights all the records which it thinks should be in the final listing (Fig. 3).

The user must then evaluate the system's response. If the user agrees with the system's listing she indicates this, and all the irrelevant records are hidden. Perhaps at this point a hard listing is produced or the query is stored away for future reference. If she disagrees with the system's guess, she can continue the process of adding more positive and negative examples.

RQBE

In standard query-by-example, the start point is a template for the eventual listing. This consists of the headings for the listing and their association to the fields in various database relations. The construction of this template is not trivial, and might well benefit from some 'intelligent' assistance.
Given this template, the user begins to fill in the slots producing a sort of archetypal line of listing. For example, say the headings in the template are `Department`, `Name` and `Salary`. You could fill in the first column as `accounts`, leave the second column blank, and put `>15000` in the third. This would denote the query:

```
SELECT Department, Name, Salary
WHERE Department = ''accounts''
and Salary > 15000
```

There are additional conventions to represent more complex conditions involving logical connectives.
There is obviously not a great deal of difference between the two representations and the understanding needed. The main improvement is in the syntactic ease of RQBE, the form of the representation suggests the form of the query. Compared with the standard form of the query, RQBE looks far more similar to the final listing. However, there is little difference in the difficulty predicting the exact records which will be listed.

Figure 1: Standard query by example

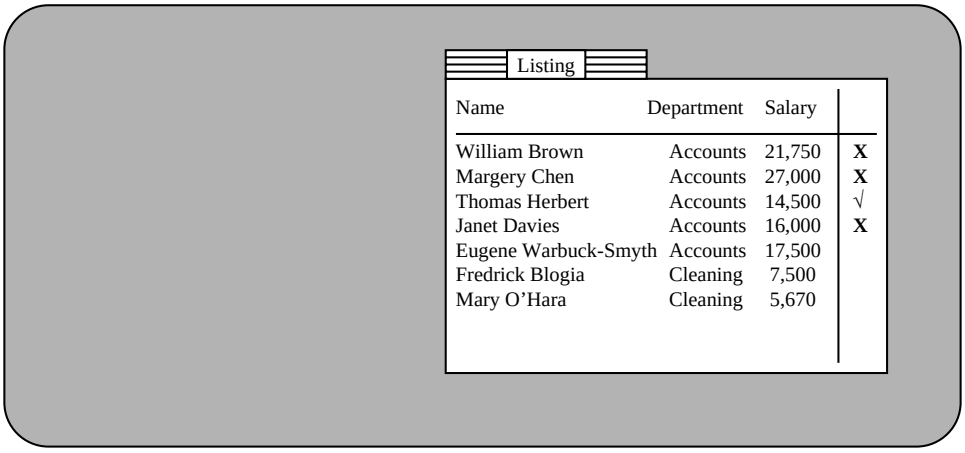


Figure 2: Query-by-Browsing – user ticks interesting records

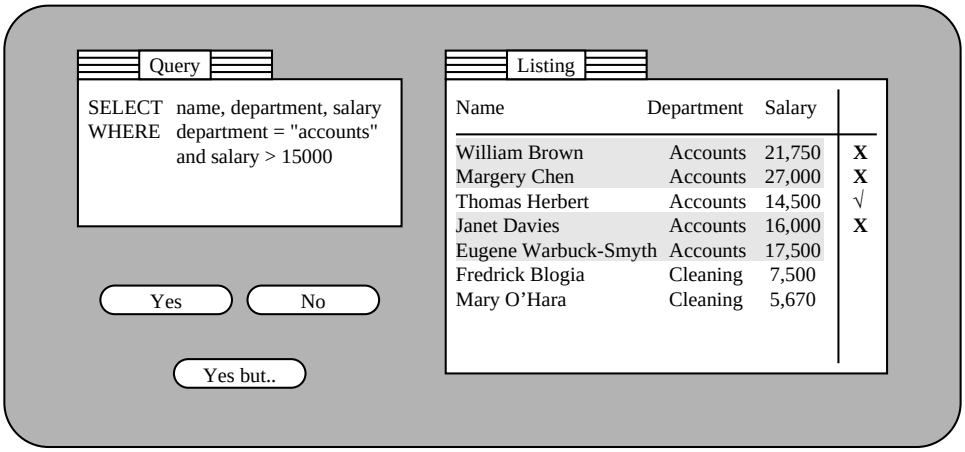


Figure 3: Query-by-Browsing – system highlights inferred selection

Alternatively, she can indicate partial acceptance of the system's choice "yes all those but more as well" or "yes only those but not all of them".

Computer solution – pattern recognition

The application of pattern recognition is quite straightforward. We train the system using the records the user has selected as interesting or uninteresting and then let it generalise to the rest of the records in the data base. Either neural nets or inductive learning algorithms could be employed. For an inductive learning system the mapping is particularly simple, each field becomes an attribute with type information from the schema telling us whether fields are to be treated as numeric, enumerated etc.

Human issues

So far so good, it is easy to apply the technology of pattern recognition to Query by Browsing. However, how easy would it be to use such a system? Even assuming that the system correctly infers the user's intention, how does the user know it has? She can see that those system selected records which are displayed on the current screen are correct, but how does she know that the generalisation is correct everywhere?

For example, in the example screen shot, the user has selected people with high salaries, but is it just the people with high salaries who are in the accounts department, or all employees? Either generalisation would be consistent with the information on the screen as the only other department visible has no highly paid employees in it. Obviously she could browse through verifying sections of the listing. Alternatively the system could display the inferred query, say in a SQL-like representation or alternatively using a RQBE tableau. Although the purpose of Query by Browsing is to lessen the focus on the query itself, we may assume that the user is more adept at reading queries than composing them and thus that it is a useful feedback on the generalisation process. This would mimic the action of the human expert aide, when shown several examples the aide may well say to the user "Oh, you mean everyone in the accounts department earning over £15,000". Displaying the inferred query would be straightforward if we chose to use inductive learning but difficult if not impossible with neural net-based approaches.

Assuming we are able to describe the form of the inferred query, is it likely to be in a form the user understands? It may be logically what the

user intended, but the form of expression may not be comprehensible. For instance, a complex query could be represented in disjunctive normal form or as a bracketed logical formula using common sub-expressions. Which form is best, and which is the best order to present terms depends on the particular application.

Similarly, domain knowledge is important in the actual generalisation process. Some sorts of query are more likely than others. It is possible to 'tune' inductive learning algorithms to generate certain types of decision tree in preference to others. For instance, Quinlan has investigated ways of producing deep narrow trees; these represent queries consisting mainly of 'and' terms.

When the user disagrees with the system (assuming she can work out whether or not she does!) she will want to enter into a dialogue to hone the system's inferred query until it meets her requirements. However, the user will have to be able to understand the system's logic sufficiently in order to know what additional information to supply. For instance, in the example, she could find a high earning employee in another department in order to show the system whether or not she meant all employees or just those in accounts. If the user does not have sufficient understanding of the system's inference, or to put it another way, if the inference method is not sufficiently comprehensible, then this supposed 'intelligent' system will be far more difficult to use than standard data base querying.

3 Application – automation of routine decision making

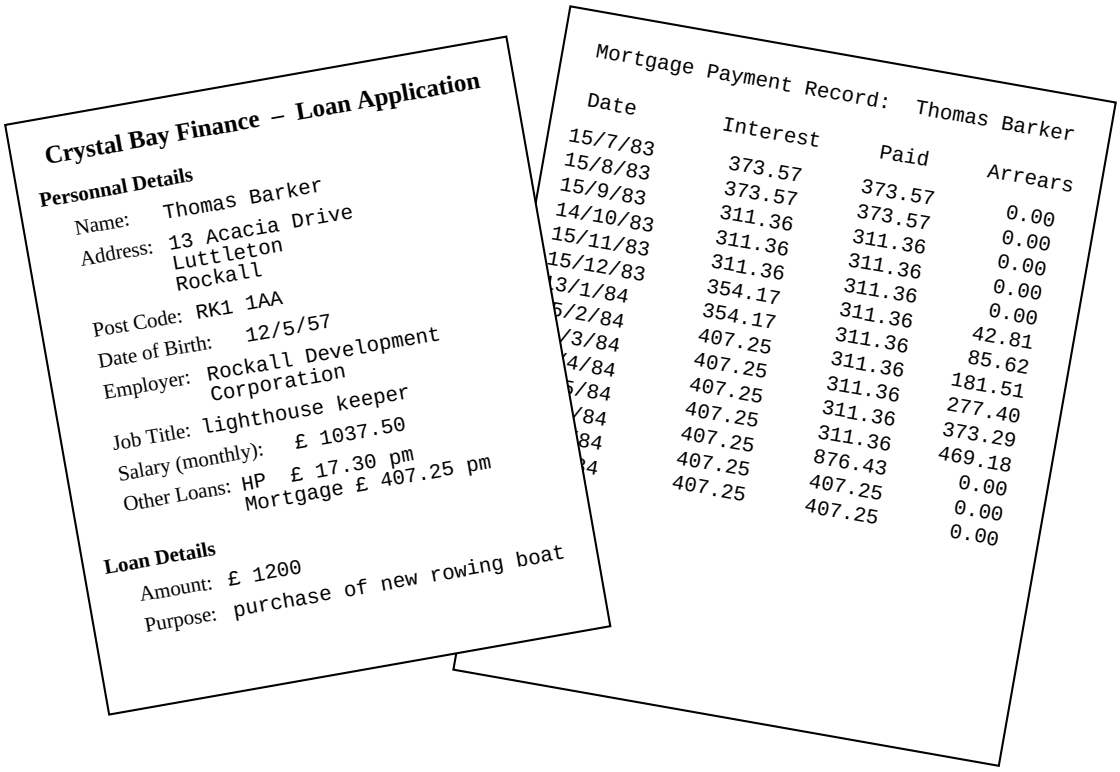
Many clerical decisions are routine. If they are made by reference to a rule book then it is a simple matter (assuming the required data is available on-line) to automate the process. However, there are a large number of situations where the decisions require a degree of judgement; even where a rule book exists it is used as a guide to the judgement process.

Example – credit vetting

Credit vetting of loan applicants is a good example of this. Various relevant factors are reasonably obvious: salary, security of employment, previous credit record. However, how these factors weigh against one another is far from obvious. On the one hand we may have a client who has a high salary

in a secure job, but who has had a history of bad debt. Alternatively we may have a client who has a low salary and is frequently out of work, but despite this has an exemplary record. My feeling would be to trust the judgement and integrity of the client who has coped well under adverse circumstances. Certainly any crude rule system such as adding up credit points could not cope with such decisions.

A similar example involving the complex weighing up of many factors is the allocation of council housing. This works precisely on an adding up points system!



Crystal Bay Finance – Loan Application

Personnal Details

Name: Thomas Barker
Address: 13 Acacia Drive
Luttleton
Rockall
Post Code: RK1 1AA
Date of Birth: 12/5/57
Employer: Rockall Development Corporation
Job Title: lighthouse keeper
Salary (monthly): £ 1037.50
Other Loans: HP £ 17.30 pm
Mortgage £ 407.25 pm

Loan Details

Amount: £ 1200
Purpose: purchase of new rowing boat

Mortgage Payment Record: Thomas Barker

Date	Interest	Paid	Arrears
15/7/83	373.57	373.57	0.00
15/8/83	373.57	373.57	0.00
15/9/83	311.36	311.36	0.00
14/10/83	311.36	311.36	0.00
15/11/83	311.36	311.36	0.00
15/12/83	354.17	311.36	42.81
3/1/84	354.17	311.36	85.62
5/2/84	407.25	311.36	181.51
/3/84	407.25	311.36	277.40
4/84	407.25	311.36	373.29
5/84	407.25	311.36	469.18
/84	407.25	876.43	0.00
84	407.25	407.25	0.00
4	407.25	407.25	0.00

Figure 4: Decisions are based on many complex factors

Computer solution – neural nets

This sort of problem is ideal for pattern recognition techniques: we have a good idea of the relevant factors and can use past records of applicants against their associated profits or losses (a slow payer who nevertheless eventually pays all debts with interest may be more profitable than the good payer!). A neural net or similar system can thus learn which patterns of factors are associated with profitable loans and hence be used to process future applications.

Not only are such systems possible but neural nets are already being used for credit vetting. So it must work, mustn't it?

Human issues – the clerk

There are two sets of users to consider here: the clerical workers in the credit company whose interaction with the system is via VDU screens, and the clients whose interaction is via paper application forms, standard letters and (if they are lucky!) via the clerical workers themselves.

The situation for the clerical workers is very similar to that of the Query by Browsing user. Faced with angry phone calls from clients asking why their loan applications have been turned down, they must either answer "because the computer says so" or have some understanding of the way the system makes decisions. They do not need to understand the complete reasoning process, but need to know on a case by case basis what were the relevant factors affecting a decision. This might enable them for instance to give a more informative reply "because of the time you were £500 in arrears on your mortgage". The client would then be in a position either to have some understanding of the decision or even to offer additional information: "that was the time they were sending all their letters to the wrong address".

Human issues – the client

We can already see that the ultimate user, the client, is affected by the comprehensibility of the system. However, there are far deeper issues. When information is processed some of it is ignored as irrelevant and other disparate elements are brought together and combined. In "Information processing, context and privacy" [3] I show how items of information which we are happy to divulge in isolation may be regarded as private when combined, further we may regard some items as private unless presented in the context of other information. The example above is a case in point. The client is

happy that the credit company knows that he was once in arrears *so long* as the reason for this is taken into account.

In a traditional data processing system these elements of filtering and aggregation are explicit and open for interpretation. With a neural net based system the precise way it processes the information at its disposal is diffuse, spread between weights in the network. Thus it is very difficult to do an information audit on a net-based decision system. This has some frightening consequences.

It has been a practice of some British credit companies that if the occupier of a house has defaulted, then *the address* is black-listed. Thereafter subsequent occupiers of the house may have loan requests refused because of the financial history of their predecessor. This practice has recently been the subject of public controversy and has been officially condemned. In a similar vein some public utilities may demand a deposit before connecting their services, the decision being based on the postal area of the property – if you're in a rough area you're obviously up to no good!

Figure 5: Is an address an appropriate way to assess creditworthiness?

Such practices can only be held up to public scrutiny if they are explicit. It would not have been possible to attack the address black-listing practice if the system had been automated using neural networks. This would of course have saved the credit companies a lot of hassle, but ...

Decision making about people involves issues of privacy and equity. That is no reason to eschew automated techniques entirely but it does demand

that we understand how the decisions are made.

Legal implications

For a similar example, imagine a company using a neural network for recruitment. How could they ensure that the decisions of the net did not contravene sexual or racial discrimination laws?

If we are using a neural net to help filter candidates for employment, we may deliberately withhold information concerning a candidate's race or sex and thus avoid the possibility of discrimination. This is not sufficient. In Britain, many recent prosecutions have involved indirect discrimination, where some secondary aspect, which is closely related to race or sex, has been used as factor in choosing a candidate for employment. So, even if we don't tell the neural net the race or sex explicitly, it may still implicitly use that information and thus act illegally.

It has been put to me that when the system makes a discriminatory decision, it may be right. The use of the word 'right' in this context being not legally or ethically right, but logically right. That is, it may be that race or sex is a good guide to the suitability of a candidate. We can easily see how this might arise. It has generally been the case in Britain that women have received a less effective scientific education than men. So, in the absence of specific educational details, sex might well be a good predictor of scientific training. Thus, without a policy of positive discrimination, we might argue that sex is good way of choosing candidates. This highlights the (rather obvious) fact that the sort of information we give a decision system, influences the type of decision it will make.

In the above case the solution is very obvious and we would seek explicit educational details. In other cases the chain of connections may be less obvious and an unhindered system might well choose sex as the best determiner of future performance. In fact, it may even be the case that, taking into account differential access to education and all social pressures, women are better than men at certain jobs and *visa versa*. However, we as a society[1] have decided that, even if this is the case, sex is *not* an acceptable attribute upon which to base such decisions.

The issue above is that quite possibly women *in general* are better or worse than men at some jobs. In credit vetting it may well be the case that persons who live in a certain area are *in general* more likely to be good

[1] Here I am referring specifically to the UK although the same could be said of many countries.

or bad debtors, or that if you get an application from the same address as a previous bad debtor, *in general*, the application is fraudulently from the original debtor. The problem occurs when these general results are applied to *particular* persons.

The whole job of a decision system is to make such generalisations. It is impossible for an employer or a credit company to know thoroughly and completely the individuals with which they deal. What is more, we wouldn't want them to know us that well! The ethical and, upon occasions, legal issue is: what sorts of generalisation are acceptable? Given appropriate data and goals, an automated decision system can make logically 'right' decisions, but it is not an ethical agent and cannot choose to 'do right'. We must either know or be able to control such decision systems sufficiently to make these ethical choices ourselves.

4 Lessons

Human issues are more complex than computer ones

In both examples it is easy to see how neural networks could be or are being used. However, in each case, the human problems of usability and suitability are paramount. The computer question is "can we do it?" and the answer is "yes". The human question is "can we understand it and use it?"; the answer is less clear.

Understanding of neural networks

One of the problems is how users understand the decisions made by the system. This is especially difficult for neural nets. Comparable techniques, for instance statistical methods, have a model whereby we can formulate precisely what the algorithm is supposed to achieve. Neural nets are often seen as a black-box, you put data in and you get answers out. However, from the point of view of understanding the opposite is the case: there is a very clear model of *how* they work, but often none of *what* they do.

To some extent inductive learning system are already better from this point of view: at least the resulting decision tree is relatively comprehensible. However, there are often many decison trees which are equally good at classifying a given set of examples. Each tree will have different generalisation properties. Why the inductive learning system chooses a particular tree may not be at all clear to the user.

Assessment of reliability

If the system produces some level of reliability for its results this can be used to improve both systems. In the case of Query by Browsing doubtful records are good candidates for presenting to the user for clarification. For credit vetting we may want to process problem cases manually.

Some neural nets produce indications of confidence, however, the measure of uncertainty yielded by the algorithm need not correspond to 'significant' features of the problem. This is a problem well known in statistics where the algorithm's measure of statistical 'significance' is not necessarily related to importance. Some understanding of the problem semantics seems essential.

Semantics and meta-knowledge

In order to behave in ways acceptable and comprehensible to the user the pattern recognition system needs to embody some knowledge about the domain. This might be implicit, for instance with Quinlan's deep narrow trees, or explicit, for instance some form of knowledge base about preferred attributes for decision making.

In the case of Query by Browsing this would enable the system to produce queries which reflect more closely the user's understanding of the problem. In the case of credit vetting it could be used to avoid decision rules which are ethically unacceptable.

This application of meta-knowledge requires a symbiosis between, on the one hand, knowledge-based methods embodying the meta-knowledge about the desired forms of decision rule, and on the other hand, more loosely structured pattern recognition systems which infer the rules within the allowable decision space.

In short ...

This all seems a tall order at the current state of the art, but unless we can produce acceptable systems perhaps we ought not be using black-box techniques in areas where understanding is crucial.

5 Comparisons with other techniques

Are the above problems purely ones for taught pattern recognition systems or are they common to other technologies. We can look at some more long-

standing technologies and see how they compare. We shall start with the youngest.

Standard data processing

In the Query-by-Browsing example, there were two main problems. One was just the design of appropriate dialogue and interaction techniques. This is equally applicable to any computer system and is one of the aspects of HCI. The second centred around the user's understanding of the system's inferences. If we were to attempt to produce such a system using standard programming techniques we would almost certainly end up with a very simple algorithm. It might not act in the way people do, but it would probably be reasonably predictable. However, when neural nets are used, it is the combination of intelligent behaviour with the fact that this intelligence is 'alien' which causes the additional problems.

Thinking about decision systems, it is certainly the case that one is frequently presented with a *fait accompli*: "that's what the computer says". This refers almost always to a traditionally programmed system. Incomprehensibility is not the sole reserve of the neural net (although they do do a particularly good job at it). When it comes to privacy and ethical issues, the paper previously cited [3] was initially aimed at just these standard data processing systems. There are certainly plenty of problems with existing systems. The one advantage that standard programs have over neural networks is the level of possible explanation. Although some programs are quite complex, it is usually the case that with a bit of effort one is able to comprehend their logic. With a neural net such comprehension is not usually possible except at an empirical level.

Despite the possibility of analysing and understanding standard software, like neural nets, it may be hard to assess reliability. Software is designed to be correct; however, we all know that most software is not. Unforeseen combinations of factors may produce a result not expected by the developer. Measuring the reliability of software is a major headache, various software metrics have been developed, and testing and validation methodologies proposed, yet no one would claim a definitive answer. As with nets the failure modes are often odd and unpredictable, not easy to find via normal testing. The odd thing about software is that there is an implicit claim of 100% reliability. Most (not all) programs demand complete input data. Without the full input data they cannot give a correct answer. This implicitly claims that *with* complete data a correct answer *can* be given! Formal verification

may help a program aim towards correctness, but it can only address bugs in implementation, not those in design or conception. At worst, it may fallaciously increase confidence and take emphasis away from the more fundamental faults in design.[2] At least systems with elements of fuzzy pattern recognition make this fundamental lack of reliability more obvious.

Some years ago I was shown a police picture which had been processed by a (traditional) image processing system.[3] The original picture, of a motor car, was blurred and out of focus, but, in the enhanced image, the number plate was sharp and clearly readable. The image processing system had been designed for astronomical images, and was merely designed to highlight sharp edged objects. Now, one might think the system would have been even better if it had been trained to recognise numbers and letters only, and specifically in the combinations found on number plates. However, the system actually used could just as easily have produced a sharp picture of Egyptian hieroglyphics, or of quasars, or complete rubbish. That fact that it need not have produced a number plate increased one's confidence in the correctness of the enhancement when it did. In other words there was a clear indication of the reliability of the result. The reliability required of an algorithm depends on the purpose to which the results will be put. For court evidence, maximum reliability is desired ("beyond all reasonable doubt"), but possibly for initial police investigations any information, however obtained, is of use.

So reliability is a problem for systems old and new, assessing that reliability is crucial.

Statistical methods

I have already mentioned comparisons with statistical methods when looking at the lessons learnt from the examples. Statistical methods have many similarities to example-based pattern recognition. Sets of data are examined and models created based upon them. The model may be used in its own right, or may be used to classify or analyse fresh data. The simplest example is linear regression which produces a straight line fit through data points. Having used it to fit a set of example (x,y) points, the resulting model can be used to predict the expected value of y for a given x, or *visa*

[2] See [4] for a discussion of the good and bad reasons for using formal methods in interface design

[3] This system also used reasonably complex statistical techniques, so belongs partly with the next section.

versa. Statistical decision theory has comparable behaviour to the neural net and inductive learning approaches adopted by Finlay and Beale (this volume) in their user modelling work. Examples of correctly classified data are given to the algorithms which then produce a decision rule for unseen data. Furthermore, factor analysis, clustering and multi-dimensional scaling all exhibit *emergent* or self-organising behaviour now often associated with connectionist approaches.

Certainly these models can be hard to understand or complex, and thus can easily suffer the same problems of comprehensibility as more modern pattern recognition techniques. However, all these statistical techniques are framed upon well understood models of behaviour. In particular, it is usually easy to state what the expected results of the algorithms are. For instance, linear regression produces the line which has the least sum-of-squares distance from the set of points. This obviously needs a little unpacking and statistical knowledge to understand terms like 'sum-of-squares', but at least such a definition exists.

Neural nets rarely have such a model. One can describe the algorithm which the net performs but have difficulty in describing what the results of the activity are. Contrast this with, say, factor analysis. This has a relatively easy description of what it does and the sort of data over which it is optimal. However, the algorithms to perform it are more complex and it requires quite sophisticated mathematics to derive them and prove that they perform the desired analysis.

One of the few neural techniques for which one can frame such a model is the Kohonenian net [2]. With a small proviso about the number of bits in each pattern being roughly similar, one can frame its results as dividing the surface of a n-dimensional hypersphere into a set of clusters enclosed by $(n - -1)$-dimensional hypercircles. Such a statement (again with a bit of unpacking!) both gives one an understanding of the output one expects from the net and also suggests what sort of data it is appropriate for. (In particular, that it is wise to pre-process data so that the number of bits set is approximately constant!). Surprisingly, even where such models are possible for neural nets, they are rarely described in this way. On the other hand, a statistical method without such a model would be unlikely to be taken seriously.

The importance of understanding has been emphasised above as one of the lessons learnt from the example systems. However, it would be unfair to leave the reader with the impression that statistical methods are completely without problems and that neural nets virtually unusable based on this

criteria!

Firstly, the types of data that can be analysed are totally different. The diffuse binary or discrete data which are handled by many neural nets are virtually impossible to analyse by traditional statistical methods. Possibly statistics has chosen the 'easy' data sets.

Also, even though the statistical methods have sound underlying models, the data upon which they are actually used rarely precisely match the required conditions. The experienced practitioner will use judgement and accumulated experience about the techniques and the data in order to decide whether this mismatch 'matters' or not. For instance, standard analysis of variance (ANOVA) is based on an assumption of normally distributed data. However, it is well known to be 'robust' and can be used on a wide range of data sets which are far from normal. There is no reason why the experienced neural net practitioner cannot (and probably does) develop a similar experience and intuition. This is clearly easier in statistics where one can say precisely where one's data deviates from the ideal, but there is not a fundamental difference. Nevertheless, it is worth repeating the enormous value of basic research on understanding the behaviour of neural nets so that they can be used with at least some of the same confidence as statistical methods. Beale's thesis work on the ADAM network is a good example of such research [1].

Finally, we must remember that neither statistical methods nor neural techniques will always be used by experienced professionals. Anyone with a reasonable knowledge of statistics will be constantly shocked at the cavalier use of statistical techniques in industrial, academic and political publications (although to be fair, in the latter the misuse is probably deliberate). A level of 5% significance is routinely used as statistical proof, when in reality it represents a 1 in 20 chance of being wrong. If the relevant experiment is to prove the safety of a drug this may be far from satisfactory. Again, the lack of statistically significant differences is often used as a proof of equality whereas it may equally well merely indicate a weak or noisy experiment. On the other hand highly statistically significant results may seem to be very exciting, but may correspond to real but tiny differences which are practically meaningless.

Given the above, the use of neural networks or similar techniques could hardly make this dire situation worse. Indeed, once while I was denigrating neural nets in favour of statistics, it was pointed out to me that existing staff recruitment procedures use statistically derived aptitude tests. These tests are deemed by their originators totally unsuitable for employee selection, yet

are used for precisely that purpose. The problems with such tests include ... the likelihood of racial discrimination, *touché*.

Human reasoning

When considering Query-by-Browsing, the analogy of a human aide was used. In the second example, the pattern recogniser was explicitly replacing human decision makers. Is this to good or ill? One origin and aim of systems which learn is to emulate aspects of human reasoning at the logical (inductive learning) or neurological (connectionist) level. In particular, they would like to adopt some of our ability to deal with partial data and reach decisions ... by intuition. Many of the criticisms leveled in this chapter against pattern recognition systems could be (and often have been) levelled against the humans which they replace. Can we learn from this?

Starting with the Query-by-Browsing. This is typical of many expert-novice situations. In some way the computer system is trying to play the same role as a system analyst: produce the listing that the relatively computer naive user requires. This is no easy matter.

Some years ago, while working in Local Government, I was asked to modify the year end listing of the authority's pensioners. After talking to the head of the pensions department for well over an hour, we agreed on the form of listing which he required. I was still left a little unhappy as I had never really got to the bottom of exactly how he used the listing. I produced an example listing and went to see him at his office. He was very pleased with the format and would have been happy to leave it as it was, but I pressed him to show me how the listing was used. Partly it was retained as a record in its own right, however there was also a parallel paper record for each pensioner. For each pensioner various figures from the listing were added up to fill in slots in the paper record. "Wouldn't you like me to calculate the exact figures you use?", I suggested. It had never occurred to him that this could be done. Presumably during our previous hour-long discussion over what was really a pretty basic listing, neither of us had succeeded in effectively communicating its purposes and possibilities.

If effectively communicating the requirements of a listing was so difficult for my colleague and I, no wonder there seem to be potential pitfalls for the automated system.

Since Aristotle, logicians have tried to generate descriptions of the way people reason. However, when humans make decisions they balance factors in ways which defy logical description. Given partial or contradictory infor-

mation, they seem able to jump to judgements. Of course, these judgements are often wrong. They may be logically wrong (many Greek and subsequent constructions to square the circle) or proved wrong by later experience (e.g. geocentricism). Furthermore, they may be ethically wrong. The only reason that I have been able to use a term like "discrimination " is because it is a recognised fault of human judgement.

This remarkable but uncertain judgement is both our glory and our weakness, and we have developed various ways to restrict the latter without stifling the former. One way in which we test our intuitive judgements is by *post hoc* rationalisation. That is, after the event we justify our actions or judgements. It is well known that, even when we try to examine our reasons, these explanations rarely account fully for our behaviour. However, they have a purpose in communicating and in checking for seriously flawed intuition. A (not typical) example is mathematical proof. The process by which a mathematician discovers a new result is ill understood and cannot be taught except as a craft discipline. But having made the discovery a rigorous proof is required. The proof and discovery processes have different purposes, and both are necessary.

Can we adopt the same approach for pattern recognition techniques? To some extent inductive learning algorithms partly satisfy this. The results of the algorithm are given as decision tree. This may be large and difficult to understand as a whole, but, given a particular decision, one can trace through the actual choice points and use this as a form of rationalisation: "your credit application was refused because you are £200 overdrawn and your birthday falls on a Tuesday". However, this only gives a trace of the decision process and does not help us to understand just why your birthday is a relevant factor. On the other hand this form of decision trace could be of help in a case of possible discrimination.

Drawing suitable rationalisations from neural nets is fundamentally more difficult. However, Kozato and De Wilde's work (this volume) is very encouraging. They use a neural net as a heuristic to drive a standard expert system. They can thus use the normal explanatory facilities of the expert system to provide a rational reconstruction of why the decision is a good one. This exactly mirrors the roles of discovery and proof in mathematics. Unfortunately, I think they intend to replace the expert system at some stage with another net – I hope not.

At a social, legal and constitutional level, the ability of humans to make gross mistakes is controlled by various checks and balances: important decisions may need ratifying by several individuals or bodies, victims of injustice

may be able to appeal through various levels of authority, and the discretion of individual decision makers may be limited by fixed rules and regulations. However, these systems, as are all human systems, are flawed, and it has been known for The best of men to be condemned by the society he lived in. We can adopt similar approaches for automated decision making: allow referral to human authority or enclose an efficient but unreliable pattern recognition system within a robust set of limiting rules. Such a system will not be perfect, but such mechanisms re-establish an element human understanding and control.

So there are various similarities and differences between human and automated reasoning. However, it is worth repeating the most fundamental difference, mentioned earlier. Humans are responsible, ethical beings – they can do right and do wrong. Computers are not. Where there is any possibility that an automated decision process can make decisions which have an impact on people, that process must be understood and controlled sufficiently by humans who can then make the final ethical choices.

6 Summary – to use or not to use

The tone of this chapter has been somewhat negative. Having read many exciting accounts of the successful application of pattern recognition it is perhaps wise to take a side-long glance. It is important for any technology that one makes a *sober* estimate of its benefits and dangers. The intention is not to eschew a method because of its pitfalls, but, because we are aware of them, to tread boldly on.

Neural nets and other pattern recognition techniques are able to analyse and process data that would be otherwise impossible, and in some applications their attendant problems are not an issue. In particular, if the pattern recognition is deeply embedded in the system it may have no immediate effect on the user. I have described above how such systems may be less prevalent than one might imagine.

A major determinant in the acceptability of a pattern recognition technique is the level of security and correctness that is required of the system. Neural nets have been used to scan news items in order to find those which may be of interest to an individual. The process is nearly identical to that proposed for Query-by-Browsing. However, with an electronic news system, it is not too critical if an interesting news item is missed or an irrelevant one is posted. For a database query the user wants *exactly* the set of records she

is thinking of, the odd extra or missed record is not acceptable.

This is even more obvious when we look at the wider decision making tasks. In the pattern of things it is probably not that important if a particular company refuses credit; there are other companies with differently trained nets (or even people!). However, when we look at staff recruitment and promotion, the issues become somewhat sharper. Finally, along this line, would we allow a conviction on the basis of a neural net's assessment of guilt? At a formal level the problems are similar, and yet the technologies we employ cannot be the same.

There are, of course, the whole gamut of process control type tasks. For instance, a neural net is to be used to control the plasma in the JET experimental fusion reactor [5]. A new improved design for the reactor coils would not have been possible with traditional control software. One assumes that such uses will multiply. I have not discussed these here, but the potential dangers are *I hope* well understood. It is worth remembering that the failure of such systems is rarely a purely technical issue.

Pattern recognition techniques make possible new applications with enormous potential benefits to individuals and to society. However, the very properties of pattern recognition which make it valuable are often those which pose problems: ease of generalisation implies potential unreliability and discriminatory decisions, diffuseness of knowledge implies lack of comprehensibility. Thus adopting such techniques entails a certain responsibility. To exercise that responsibility we must obtain a level of understanding of the techniques we employ, so that the decisions made will be *our* decisions – for good or ill.

References

[1] R Beale. *The Theory and Applications of Associative Neural Architectures*. PhD thesis, University of York, Department of Computer Science, 1991.

[2] R. Beale and T. Jackson. *Neural Computing: an introduction*. Adam Hilger, 1990.

[3] A.J. Dix. Information processing, context and privacy. In D. Diaper, D. Gilmore, G. Cockton, and B. Shakel, editors, *Human–Computer Interaction – INTERACT'90*, pages 15–20. North-Holland, 1990.

[4] A.J. Dix. *Formal Methods for Interactive Systems.* Academic Press, 1991.

[5] Elizabeth Greake. Neural network keeps fusion plasma in shape. *New Scientist*, page 27, 12th October 1991.