WORKING NOTE AC64

Alan Creak
12 June 1995

AN ORDERLY APPROACH TO PATTERN PERCEPTION

PREPREAMBLE.

I think I started writing this in 1987, while Lex Miller was working on his thesis[1]. The intention was to clarify my mind on the subject of pattern perception, but it got out of hand. Now I'm not so much tying up loose ends as cutting them off, just to get the Note out of draft form and into a state of presentability which will justify my letting it die. It may be no more than a collection of half-baked ideas; please don't try to read more than that into it. If it means anything to you, I'm glad – it probably means that you've perceived a pattern that I didn't. Tell me about it.

PREAMBLE.

These ideas grew from an attempt to follow the consequence of this proposition :

> *if all artificial intelligence can be regarded as search in some appropriate problem space, and if the activity of perceiving patterns in collections of data is a proper part of artificial intelligence, then there must be a corresponding problem space, and there must be ways of moving about within that space to approach a goal.*

(There must, for that matter, be a goal, but for the moment I'll take that as self-evident. Later on, I'll have to try to define it.) One can quibble about details in the proposition : "must" may be too strong a word, perhaps not all artificial intelligence can be represented as a problem of search – but that isn't the point. The point at issue is that it's a plausible proposition, and it's a fruitful proposition, and for present purposes that's probably better than being right.

The proposition was put forward in the particular context of Lex's thesis[1]. We were confronted with a large volume of recorded data, which happened to be about sailing boats (that shouldn't be important, though it's useful to have a concrete representation to talk about). It is essentially certain that there are patterns of some sort within the data – if the wind gets stronger, the boat will move faster in the direction of the wind – but we don't know what the patterns are. We want to write a programme which will inspect the data in search of patterns.

I am being careful to write of "*perceiving* patterns" rather than "*recognising* patterns" because I want to emphasise that the problem is to discern any patterns that may be present in a mass of data, not to search the data for occurrences of some predefined pattern.

A final qualification : I'm not discussing neural network methods. They're different. I would like to think that they will turn out to be good at perceiving patterns, but most of the work I'm aware of at the moment has more to do with recognising patterns. Some of the unsupervised learning techniques could be seen as pattern perception, but those I've read about don't in any sense look for patterns – they'll learn anything, rubbish or not. It seems to me that a pattern perceiver must have some idea of *significance* with which it can judge its perceived input, distinguishing significant patterns from noise. Neural networks don't seem to do this particularly well at present. John Jensen put some work into this approach, but without conspicuous success[2].

I am ready to believe that this activity of discerning patterns is reasonably described as intelligent, if only because it's something that people do startlingly well, and I think I'm being intelligent when I do it. We are even able to discern patterns that simply aren't there – which is where the Rorschach test comes from. (Though maybe that's really a matter of pattern recognition ? No matter – the two are quite closely connected. I suppose ?) Unfortunately, that's no help : introspection gives me no hint on how I do it. I'm certainly not conscious of any search activity; but then, I can say the same of most, if not all, of my experience at solving problems tackled by artificial intelligence programmes.

What is the goal ? To find some patterns. What does that mean ? Er … So how can I define a problem space ? I can't. So where do we go from here ?

I must start by seeking definitions. A pattern-discerning programme requires certain basic operations appropriate to moving in the problem space. It requires some way of determining whether whatever it finds is significant. To define such ideas more clearly, I need answers to questions like these :

What sorts of data are presented ?
What sorts of pattern can be expected ?
How do you look for them ?
How do you recognise them ?
How do you compare them ?

In the rest of this note, I ruminate on these questions, and finally come back to the original question : what is the goal ?

WHAT SORTS OF DATA ?

If we're looking for a pattern, we have to start with a set of things to inspect. If you only have one thing, there's no non-trivial sense in which you can discern a pattern; if you only have two things, then the most you can say is that you can find some coincidences. ("To lose one parent, Mr Worthing, may be regarded as a misfortune; to lose both looks like carelessness."[3]) Indeed, you need a sufficiently large number of things to be able to say that certain statements can be made about more of the things than you would expect under some form of null hypothesis. I am not at all clear just what form the null hypothesis should take, but it must be able to cope with quite qualitative descriptions, if only because the patterns I'm interested in are unlikely to be well quantified. If I have a lot of samples each of which shows about three points, then "about three points" is a sort of pattern, unless there's some mechanism which makes three points inevitable. (Perhaps the experiment is just designed to count three points for each sample.) "The three points always define a triangle" isn't a pattern, because that's inevitable. "The three points always define something quite close to an equilateral triangle" is a pattern, because it isn't inevitable, and one might expect triangles of all shapes and sizes – which is a sort of qualitative null hypothesis, but "quite close" is decidedly fuzzy.

It seems to me that "Three points always make a triangle" isn't a pattern, but "all cats have four legs" is a pattern. What's the difference ? The triangle assertion is inevitable, but the cat assertion isn't far from inevitable. I think I give up.

What do I mean by "thing" ? From the first paragraph in this section, it must be able to contain or represent or otherwise include approximately one pattern. I think it's something like "context in which a pattern might be found". There could therefore be many things in a stream of observations or a picture. The "definition" implies that the sorts of pattern we find are to some extent governed by the things we inspect. (That's obvious, really : you can't find a big pattern in a small thing, and it's probably almost as hard to find a small pattern in a big thing.)

So the minimum we can get away with in the way of data is a set of things, and some expectations about the things on which we can base our assessments of likelihood and the null hypothesis. On the whole, the sets are easy; if we didn't have a set of something, we wouldn't be looking for patterns anyway. Notice that there is a sense of statistics – or, at least, probability – creeping in. Is that inevitable ? I suspect that it may be.

The expectations are harder. If you accept the very-null-indeed hypothesis that everything is equally likely, you will find all manner of patterns – mainly patterns of omission – such as that there are no green cats, or houses built of jelly. (Long, long ago, I tried to find patterns in properties of chemical materials. After a great deal of very hard work, my programme came up with its single result : that acids were not alkaline. I was delighted.) Even these observations have their own peculiar significance, but we don't really want them, as they contribute nothing to what we know, and waste a lot of time. If, on the other hand, we derive our expectations from the sample itself, then everything will appear to be normal.

People seem to see patterns in a set of things by appealing to their general knowledge of how things of that sort behave. If 10% of a sample of dogs have only three legs, one would regard it as noteworthy; we know that dogs normally have four legs, and that they don't drop off easily. If 10% of a sample of animals of a hitherto unknown species have only three legs while the rest have four, one would probably still regard it as noteworthy, this time basing the expectation on knowledge that animals of a species usually have the same structure, at least so far as easily countable appendages are concerned. If our programme is to do this, we have to give it our expectations about the field before it starts work on the set of data. But unless we can somehow encode our whole knowledge about the objects concerned, we could omit the critical fact that makes a pattern noteworthy. (Though does the noteworthiness have any connection with the patternness ? I really don't see that it does.)

The first sentence of the preceding paragraph is worth a second look. What is one's "general knowledge of how things of that sort behave" ? Well, it's a pattern. "Almost all dogs have four firmly attached legs" is certainly a pattern we discern in our observation of the world around us, and a 10% incidence of three-legged dogs in some sample, while part of a pattern in any circumstances, becomes significant because of the expectations raised by the broader pattern. Clearly, this collection of patterns which we call our general knowledge is of great importance in assessing the significance of our observations, which is one of the reasons why Lenat and others[4] have been working on the Cyc project, aiming to codify general knowledge in a way which will make it accessible to artificial intelligence programmes.

While this is an eminently desirable thing to have, it may not be what we want in our pattern recognition systems. The trouble with general knowledge is that it isn't general. For all the talk about global villages, the general knowledge of a North American is rather different from that of a New Zealander, and most probably even more different from that of an African or a Chinese. (I'm sure that Lenat and co. will have thought of that, though I don't know what, if anything, they propose to do about it.) There is, of course, a common subset – days and nights alternate, things drop if not supported – though it's by no means synonymous with, and may even contradict, qualitative physics. That's the point of Galileo's demonstration, apocryphal or not, in dropping weights from the tower of Pisa. Probably more significant for our immediate purpose, general knowledge is hierarchic. Within any field of expertise, there is a sort of specialist general knowledge shared by all practitioners and present as a background in their reasoning. Chemists, but not necessarily biologists, have a collection of general knowledge about atoms and chemical bonds; physical chemists, though not necessarily natural products chemists, have general knowledge about thermodynamics; and so on, perhaps through several levels of increasing specialisation. What we need is a way to describe the patterns we expect to find, so that the programme can use these to evaluate any patterns that it manages to perceive. (Having said all that, though, it must be said that one would be very happy with a programme which managed to identify any pattern whatever !)

In practice, we seem to get round this to some degree by only writing programmes which look for patterns of a certain kind. In effect, we build into our programmes a potentially interesting subset of our expectations. That's better than nothing – but people can do that, too, and have been known to miss very obvious features by concentrating on one thing[5].

> "*His observation in matters botanical was what the inferior sort of scientific people call a 'trained observation' – you look for certain things and neglect everything else.*" [6]

But how do we decide what's likely to be interesting ? Is it guesswork, or wishful thinking – or are we, in turn, discerning a pattern in something or other, and using that for guidance ? Whatever the answer to this question, it seems very likely that to include any such explicit knowledge of the field in the programme will introduce bias of one sort or another, perhaps making it harder for the programme to find certain patterns which we didn't expect.

Some examples, and a sort of taxonomy.

Enough of speculation; let's look at some realish examples. Here are some examples of the things which populate the sets which we might study – or, more precisely, observed properties of the things, as there are few things which we can study directly inside a computer. Some notes :

- I have tried, not too hard, to stick to some conventions in my use of symbols. For values, x stands for anything, and a for something sortable; for subscripts, i distinguishes members of a set, while n implies that the order is significant.

- If you want to be careful, watch out for the distinction between sets and bags : I haven't worried about it, so many things I call sets can contain more than one copy of a value.

- It is interesting that many of the items listed as "properties of interest" (and occasionally under other headings) can be regarded as properties of derived sets rather than as properties of the sets ostensibly under discussion. I shall return to this observation later, as I think that it holds the key to the nature of the problem space with which I began this note. For the moment, just observe the derived sets as they appear.

Single independent observations : x_i

The observations are atomic; they have no internal structure. They are purely qualitative (green, apple), and cannot be ordered in any significant way. (So even "green" won't do if you define a spectrum.)

Examples : I can't think of any convincing collection of primary data which comes into this category. The sets are nevertheless significant as derived sets from more complex collections; for example, "the set of colours of all cars in the sample".

Topics of interest : counting.

Counting in fact gives us a new set of pairs, each pair containing an element of the original set (in this case, certainly a bag) and its frequency of occurrence. I think that's the only one. If we have a set of 25 reds, 4 greens, and 3 blues, then we can say that things seem likely to be red, but not whether that's at all remarkable. The only relationship between items on which we can build patterns is similarity; two items are either the same, or they are different.

Unordered observations are likely to be expressed in terms of classifications : red, blue or green. The only patterns in such an environment are the collections of observations themselves, which we regard as indicative of the populations from which the observations were taken. "The things are most likely to be red."

Methods : No method is required.

Single observations, sortable : a_i

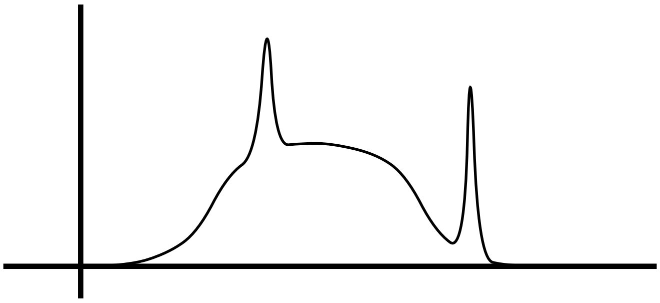
Like the sets we studied in the previous category, the items here have no internal structure; but they are quantitative, at least to the extent of defining an order (35, small, old). In consequence, as well as possible identity with other items, each item now has neighbours – items identified as being in some sense "closer" to it than others. Notice that we have no idea whether or not the sorting is likely to be significant.

Examples : examination marks, numbers of passengers per vehicle.

Topics of interest : distribution. (Counting, too, of course : generally, things to do with simple classes apply to more complicated ones, and I won't discuss them again explicitly.)

Here again, we are looking at properties of a derived set of pairs of values and corresponding frequencies. This time, the values are typically the midpoints of ranges of values of the observations, and the frequencies are the numbers of observations found in the ranges.

For ordered observations, the same applies; but people perceive any departure from a "simple" distribution as an "interesting" pattern. Consider a sample with a fequency distribution something like this :



We are likely to describe it as "two strongly preferred values superimposed on a normal distribution", and forthwith start looking for reasons for the two special values. In practice, that seems to work rather well – the whole of spectroscopy is based on it, to give only one example – but that itself is a pattern ! ("If we look for deviations from simple behaviour, we can find reasons for them.") To justify that one, we have to make assumptions about the universe by default behaving in ways which we consider to be "simple".

We are getting into deeper water than I'd intended.

If we have something that looks rather like a normal distribution, but with two strong sharp peaks, then we see the peaks as the patterns.

If they are numbers, we can also say *how* similar they are.

It is not necessarily easy even to decide whether or not the observations fall into significant groups. Lex Miller[1] addressed this problem, and found that a method based on successive local averages led to a useful classification.

Methods : clustering theory.

Collections of observations : $\{ x_i, y_i, z_i \dots \}$

Each observation contains several items; all observations normally contain items of the same types; missing values are possible.

Examples : experimental results, survey results.

Topics of interest : functional relationships, distribution of values.

Now we have some structure within the observations, and we can look for internal regularities. More than that – we can always find some; there are always infinitely many mathematical (including logical) equations which are satisfied by all members of the set. Most such functions are not significant – we shall discuss examples later – but patterns correspond to more useful relationships.

Generally we seek relationships between the measurements within an observation. In their simplest form, these are just statements of (perhaps statistical) fact : "Number of heads = number of bodies = (number of legs) / 4"; "If that's a face, there should be eyes there and there, and a nose there, and ..."; "Red sky at night, shepherd's delight"; "Absence makes the heart grow fonder".

We can (try to) write them as functions (mathematical, logical, etc.) to make them more precise, and include measures of probability.

Methods : Bacon, ID3.

Collections of observations, with a distinguished variable : $\{ x_i, y_i \dots, t_i \}$

Collections of this sort constitute an important special case of the previous class. One of the observations is of special significance, for some reason connected with the nature of the observations. Typically it will be the independent variable; in a very common special special case, the independent variable is time, which is why I've called it *t*.

An ordered set of collections of observations falls into this category, perhaps requiring the inclusion of an ordinal value within each collection.

The ordering observation may make it possible to determine whether there are missing values. I'm not sure that that means a lot, except that there may be patterns in the omissions, but they would be a species of metapattern.

Examples : Observations of a variable over time. (Strictly, I don't suppose that there's anything special about time so far as patterns are concerned, but time series are common things.). Seismograms, tides, binary signals – Morse code, spectra of related substances, variables measured in parallel, data logging, traffic lights, speech, writing. Examples not involving time : outline grammars, prescribing the order of features encountered as an outline is followed; character groups and sequences in text.

Topics of interest : relationships between different variables, trends.

Each sequence may be a candidate for study under the previous heading – or we may already know what its interesting features are. Notice too that we may not be able to identify features in one stream without cues from another- or, equivalently, that patterns spread over several collections and their neighbours in "time".

Methods : As for the previous group, but with special attention to the distinguished variable. It is generally of interest to look for functional dependencies of the dependent variables, or combinations thereof, on the distinguished variable.

Repeated patterns of behaviour are of considerable interest. This is what Lex Miller was doing\nc<Lex>. The repetitions need not be regular : consider prime pairs on the number line.

Collections with several distinguished variables : $\{ x_{i,} y_i, .. t_i, u_i, .. \}$

I include these as an obvious extension of the previous category, but I don't propose to discuss them in detail. While basically similar to the previous class, they differ in that there may be relationships between the independent variables which define important special cases – so, for example, unusual behaviour may not be associated with any specific values of t_i and u_i but with any example in which $t_i = u_i$.

Examples : Contour maps, phase diagrams.

Sets of observations $\{ x_{ij}, j = 1, n_i \}$

Each set is just a collection of things which are associated in some way, each thing has its own properties. These sets differ from the collections of variables in that there can be any number of instances in a set; the collections of variables always contain the same number of components.

Examples : plants found growing together within 1m squares.

Topics of interest : which things are associated, under what circumstances.

Regularities observed under this heading may become bases for further investigations of the "Collection of observations" sort – so, having found that certain plants are commonly found together, we may proceed to focus attention on those plants and gather more information on them alone.

"Where there's a will, there's a way"; "Two's company, three's a crowd".

Methods : statistics ? clustering ?

WHAT SORTS OF PATTERN ?

The patterns in data are superficially of different sorts, but there is a common thread, which reduces everything to the behaviour of sets of single observations. What we can say in each of the more complicated cases is that – if we can find a pattern at all – we can define some function (formal or informal) of each thing in such a way that the set of function values shows some significant pattern – some predicate is almost always true, or some function has only three different values. (Yes, that begs the question of what constitutes a "significant pattern". My "acids are not alkaline" falls into this category, and plenty of experimental measurements have been coerced into straight-line graphs, often by rejecting points which didn't fit as "obviously erroneous". There are reports that *automatic* rejection of "obviously erroneous" measurements delayed recognition of the upper atmospheric "ozone hole" for some years.) We can then collect the function values for the members of the set into an auxiliary set of single observations, and deal with that in ways appropriate to such sets.

Another sort of derived set is exemplified by the (n_i, n_{i+1}) set which can be derived from a set in which order is significant : as well as combining elements within an observation, we can draw comparisons between observations. If there is a distinguished variable, it may give guidance as to which comparisons are likely to be significant – or it may not. That depends on what sort of behaviour underlies the patterns. If the behaviour is continuous, then presumably local correlations will be most informative; if the pattern is caused by some spasmodic event, then the interesting feature is the repetition of local structures over comparatively long, and possibly arbitrary, intervals. If there is no distinguished variable, then we may have to consider all possible pairs if we wish to give an honest treatment.

There is, of course, no sense in which the derived set is unique; we can define such a set for any function defined over the original set, and unless we have reason to believe otherwise the derived sets are presumably of comparable significance.

For example, if all the variables in a set of collections of observations are numeric, then for each collection evaluate the sum, S_i. Now every collection satisfies the equation

$$\Pi_i(x + y + z + ... - S_i) = 0.$$

If the variables are not numeric, then for each collection construct an expression in which each variable is equated to its value in that collection, and conjoin the resulting terms; then the disjunction of all the expressions so defined is satisfied by each member of the collection :

$$\bigcup_i((x = x_i) \& (y = y_i) \& (z = z_i) \& ...).$$

Of course, such expressions are vacuous, as they carry no more information than was present in the original set of data. They are not quite useless; they will reliably recognise members of the set if they appear for testing, but there is no reason to expect that newly generated collections will follow the same pattern. In short, they are (probably) not reliable generalisations of the patterns exhibited in the original set. I shall say more about this topic later while discussing the sorts of pattern which can sensibly be sought.

- It is sometimes possible to demonstrate that expressions of the sort suggested are necessarily rubbish, provided that we have more information about the natures of the variables concerned. A good example is the requirement that addition and subtraction is only sensible if the quantities added and subtracted have the same dimensions. If x is a distance, y a speed, and z a time, then no expression of the form $x + y + z$ can have any meaning at all. The best we can do is to use an expression which includes cooking constants – $ax + by + cz$ – to fix the dimensions – thereby introducing yet more opportunities for engineering meaningless coincidences.

- I said I wasn't going to talk about neural networks, but it's interesting that just this sort of behaviour is a problem in that area. If you train a large network on a comparatively small number of examples, it may learn to recognise the patterns individually in just this sort of way. It is therefore better to use a smaller network, to force it to seek a more economical representation of the data which may turn out to be an effective generalisation. I have no idea whether anyone has checked whether neural network function generators do sensible things with dimensions.

The question is : how can we be sure that an automatic procedure which we unleash on our set of data will not return some inane function of the sort I have described ? Empirically, we do so by insisting on a *simple* relationship defining the regular behaviour, so all we have to do now is define what we mean by "simple". The equation for the numeric collections above would fail because it has too many adjustable constants, and the expression for the logical case would fail because of its complexity. (I know that begs the question, but you try to explain it. You can devise measure of the complexity of an expression in terms of the length of the shortest bit string required to define it, but I'd rather avoid that if I can.)

A difficulty here is that our own ideas of simplicity are not very well defined. There is an old example which goes like this :

Q : What is the next number in the sequence { 1 2 4 8 16 ... } ?

A : 32.

Response : No.

1. 32 is the next element of the exponential sequence 2^n;

2. there is a quartic polynomial through the points (1, 1), (2, 2), (3, 4), (4, 8), (5, 16), (6, 31);

3. a polynomial is generally regarded as a simpler function than an exponentiation;

4. therefore the correct answer is 31.

I shall shortly explain why I think that 32 is indeed the correct answer, but before that I want to remark on some interesting features of the argument.

First, each step of the argument is acceptable in itself, so either the argument must be accepted, or it must have missed something. What's missing ?

Second, it regards the sequence { 1 2 4 8 16 } as the same thing as the set of pairs (n_i, p_i), where the added item p_i is something which determines the order of the elements – and in this case p_i has been made equal to i, which certainly works, and gives the set { (1, 1), (2, 2), (4, 3), (8, 4), (16, 5) }. That's why I didn't include ordered sequences in the earlier, single variable, classes. The assertion that the order of the elements is significant is information over and above that contained in the values themselves. The same information is captured explicitly in the set representation, which identifies sequences as falling into the "collections of observations" category. (However, there is more to say about that, so bear it in mind.) Is there, after all, much difference between that sort of sequence and a sequence of observations of, say, the size of a population at a series of times, which we would quite happily accept as a set of pairs (n_i, t_i) ?

Well, there's perhaps not much difference, but there is certainly some difference, and it lies in the interpretation of the second item in the pairs. In the case of the time series, t_i is clearly numeric; there is no particular reason why it should increase in regular steps, provided that we know what the appropriate values of t_i are. The second items of the series representing the powers-of-two sequence, p_i, are only numbers by accident; their purpose is solely to establish the order of the first items, and we could have managed just as well in principle (though rather less conveniently) with { (1, +), (2, /=+), (4, /=+-), (8, /=+-++), (16, +/=+-++) }. And that is why the answer is 32; the flaw in the argument above is in the unstated assumption that the numbers in the sequence must be determined by some function of their ordinal positions in the sequence, whereas all that is actually given is the order of numbers, so all we can really expect to find is some relationship between each number and the preceding numbers.

In other words, what we really want to find is some regular property in the pairs (n_i, n_{i+1}) – that is, { (1, 2), (2, 4), (4, 8), (8, 16) }. Can we find such a property ? Yes, and without too much difficulty; the pattern is rather obviously determined by the equation

$$n_{i+1} = 2\, n_i,$$

and you can fill in the bits that lead to 32.

What would we do if we didn't find such a pattern ? We'd try triplets (n_i, n_{i+1}, n_{i+2}) – think of Fibonacci numbers. If that didn't work, we'd try quadruplets – and so on. We wouldn't get anything like the quartic until we tried quintuplets, but then we only have the single example (1, 2, 4, 8, 16), so any relationship we find is, like the hopeful ones we began with, vacuous. 31 is wrong, for we have exactly no evidence that it conforms to any pattern at all.

HOW DO YOU LOOK FOR PATTERNS ?

People recognise patterns primarily with their eyes and their ears. Ears (and a lot of associated neural machinery) recognise words in speech, rhythm and melodic and harmonic patterns in music, and can separate significant sound patterns from meaningless background. We use an aural metaphor for meaningless signals in general – we call them *noise*.

Nevertheless, we are even better at discerning patterns with our eyes, and even the sonic patterns can very easily be represented in graphical form. Perhaps that's why so much of pattern recognition is commonly assumed to be concerned with visual patterns, even when it isn't. There is a danger in relying on this visual metaphor, in that there may be patterns which have no visual representation. I can't think of any. If there are any, they will probably be abstract patterns rather than physical ones (which we can always draw); even with these, we indefatigably seek visual representations, and are commonly successful. Consider the periodic table in chemistry, prime pairs marked on a number line in mathematics, Venn diagrams (or Euler's Circles[7]) and other patterns[8] in logic.

Assuming, then, that we can always draw some sort of picture, however symbolic, of our data, what does a pattern look like ? It always concerns (at least) two things.

Patterns of more than two things exist, but they're much harder to see.

The "standard" artificially intelligent method for exploring an unknown problem space is (or, maybe, was) the agenda method, as exemplified most dramatically in AM and Eurisko[9]. These programmes are experimenters and explorers; given a set of things they can do, they search for interesting ways of combining their possible operations, and follow up leads which seem promising. Indeed, they are doing something very close to the very pattern perception I require, though that isn't the way they're usually described.

The problem space explored by AM and Eurisko is unknown, but it is not in any sense undefined. The consequences of applying an operator to some state are straightforward to compute; the clever trick is to reach a sequence of states which is deemed in some way to be interesting. It is not possible to associate an interestingness gradient with an operator, so we must have some way of estimating the interestingness of the states themselves, and we proceed by arranging known states in order of interestingness (the agenda) and generating a new state by applying some suitable operator to the most interesting state. Repeated application of this sequence (with a few elaborations which make it work but don't change the idea) leads us along interesting paths until either they peter out or some state of sufficient interest to be noteworthy is attained.

Perhaps the major difference between this technique and other ways of searching problem spaces is in the absence of any systematic search of the possible transitions from any individual state. It is quite possible for the method to miss whole areas of exciting behaviour simply because the first few states on the way looked rather boring. This possibility is accepted as preferable to the alternative of missing whole areas of exciting behaviour simply because of an insistence on following up even the most boring paths on the off chance that something good might happen.

How does the agenda method fit into pattern perception ? Pattern perception isn't quite the same as number theory (AM) or war games (Eurisko), in that the space to be explored isn't just abstract; instead, all proposals must conform, in some sense, to the data provided. I think that means we are exploring a two-component world. One component is the set of allowed operations, analogous to the abstract rules used by AM and Eurisko; but the other is the collection of data which constitute the raw material on which the programme must work. AM could generate examples of some concept by doing simple arithmetic; but the pattern perceiver can only "generate" examples of a hypothesis by actually finding them in the data.

HOW CAN PATTERNS BE RECOGNISED ?

The methods I have discussed are essentially those which I think I use when looking at various sorts of graphical representation of data. If these are to work, we must find a way to represent our data graphically, and it must be a way that works. It's unlikely to be useful just to type a list of facts as they come to mind and then to draw a circle round each one; that's a picture of the data, but we have no reason to believe that it will show up any order that might exist.

Instead, we have to begin by representing any order we can see in the data. Then we hope that new regularities will become apparent. That's why we invented graphs (the "plot *x* against *y*" sort); we exploit some order we know about in the hope of finding order we don't know about. We plot *x* against *y*, and look at the graph.

How not to do it.

A paper[10] presents another way of deciding whether experimental results conform to a pattern or are caused by random fluctuations. Unfortunately, that's all it does, because it's the most astonishing collection of tripe I've read in a "respectable" journal for some time. The author is a crank who doesn't believe in statistics; one can only guess what the referees are. A prime piece of "evidence" is gained by analysis of two bit strings, one of which is given by someone called Gleick, and purports to be a representation in Morse code of the message

"All form is formless",

while the other is generated as a psuedo-random (sic) string by a very peculiar looking Basic programme, and, of course, means nothing at all.

In fact, the Gleick string is a concatenation of the Morse representations of the characters of the given string, with 0 for a dot and 1 for a dash, with no punctuation for gaps between characters or between words. It therefore means all sorts of things, such as

"Entire item ten mess normless";
"Entire item kegs stem tap these".

Further, the random string can be read as :

"Men gut madam I set meek note at me";
"Men gut madam heat tin gone on".

(There were more, but I'd wasted enough time. I didn't have to begin with "Entire item".) If you want to quibble that they don't make sense, look at the original interpretation.

The method depended on the accident that the Gleick string contained two sequences of at least seven adjacent zeros, which he thinks is unlikely in a truly random system. His "random" string, when split at 5-bit boundaries (which just happened to be the way I copied it) contains the substrings 11000, 11100, and 11110 twice each – I haven't checked any other sort of partitioning.

It's the March/April issue. Is it a joke ? It isn't witty enough. This man is working on medical statistics.

HOW CAN PATTERNS BE COMPARED ?

We want to compare patterns so that we have a way to choose the "best" of a set of possibilities. We want to do that as a guide to the artificially intelligent programme which is supposed to be searching for the patterns. Eurisko needs some basis for arranging ideas in order of interestingness, and any reasonably plausible measure is better than none.

What makes a pattern good ? There are several ways of looking at the satisfactoriness of a pattern imposed on a set of data. Here are some.

The pattern is interesting in itself.
The pattern conforms to some previous expectation.
The pattern contradicts some previous expectation.
The pattern happens a lot.
The pattern fits the observations very well.

WHAT IS THE GOAL ?

I am now prepared to return to the original question, and to give a fairly formal definition of the goal of the search : it is *to find an interesting set.*

An interesting set is a set of pairs of the sort which I called the "new set" in discussing the simplest set – the set of single independent observations. The new set contains pairs, each comprising a value and a frequency. The interestingness of the set depends on the set of frequencies. Suppose that there are n pairs in the "new set" and m elements in the parent set of single observations. Then the set of pairs is interesting if one or more of the frequencies in its elements significantly exceeds m/n – meaning that the values are distributed non-randomly over the range. (It's also interesting if n is very small, for then we have found a function which classifies all the examples into a small number of sets. I've left that out because I can't work out what I mean by "small".)

That's hardly a rigorous definition, but it's a start on a definition of what might be called the intrinsic interestingness of the set. I pointed out earlier, though, that the actual interestingness depended on expectations too, so we have to assess this result against our general knowledge. This may convert an interesting set into an uninteresting one, or vice versa – or it may even convert an interesting set into a set which is interesting for a different reason ! An example is our set of dogs, 10% of which were three-legged. The set is intrinsically interesting, because far more than half the dogs had four legs; the real interest lies in the unduly large proportion of dogs with three legs. I suppose that the point of that example is that the intrinsic interestingness is really pretty useless; general knowledge is very important, and we must find ways to use it wherever possible.

On the other hand, the idea that interestingness is associated with sets of simple things is interesting. So to speak. It gives us a measure of interestingness which we can use to guide our development, and to grade operators. This is something not available to the AM and Eurisko programmes. Of course, this doesn't mean that only operators which reduce the sizes of set elements are interesting; the most interesting operator we found in the powers-of-2 example was that which associated adjacent pairs of elements to produce a set of pairs from the original sequence of single items, and in many circumstances it may be necessary to form more elaborate combinations before regularities can become apparent.

The operations are clearly pretty well any operation which converts a set, or some sets, into another set. I have mentioned a number of such operations in this note; clearly, there are many sorts. Here are some :

Projection : $\{ x_i, y_i, ... \} \rightarrow \{ x_i \}$

Counting : $\{ x_i \} \rightarrow \{ X_j, |\{ i \mid x_i = X_j \}| \}$, where X_j is a value taken by x_i, and the second term is the number of instances of that value in the original set.

Functions : $\{ x_i, y_i, ... \} \rightarrow \{ f(x_i, y_i, ...) \}$ For the powers-of-2 pairs, $f(x_i, y_i) = x_i / y_i$.

Association : $\{ x_n \} \rightarrow \{ x_n, x_{n+1} \}$

Merging : $\{ x_i, y_i, ... \} \cup \{ x_i, z_i, ...) \rightarrow \{ x_i, y_i, z_i, ... \}$

Selection : $\{ x_i, ... \} \rightarrow \{ x_i, ... \mid x_i \in S \}$

We can now see the pattern perception process as a succession of set transformations chosen to isolate exceptional phenomena, and then to reduce the resulting set to some simple measure of interestingness. Clearly, we must guard against vacuous combinations of properties which have only accidental significance, and we must take into account our expectations gained from general knowledge.

That does not constitute an algorithm for the job. It is far from clear how to choose "good" transformations, or how to ensure that we don't accidentally end up with a vacuous function. Even general knowledge has its traps; it's no use simply eliminating all the four-legged dogs from a sample without first checking their other attributes. (Perhaps they all speak fluent English.) In other words, there's lots of work to do yet, but at least I now have some idea of what it is.

Well, that's what I thought in 1987 – 1991. Rereading it in 1995, I'm not sure that I'm quite so confident, but I still think that there's something in there somewhere if I could only find it.

(And I read some other things which I haven't discussed[11, 12, 13], but this sentence gives me a hook on which I can hang references in case anyone's interested.)

REFERENCES.

1 : L.C. Miller : *LOVE – learning from observations converted into examples*, M.Sc. Thesis, Auckland University Computer Science Department, 1988.

2 : J.T. Jensen : 07.501 Project report, 07.483 Project report, Auckland University Computer Science Department, 1990 and 1991.

3 : O. Wilde : *The importance of being earnest* (1895), Act 1.

4 : D.B. Lenat, R.V. Guha, K. Pittman, D. Pratt, M. Shepherd : "CYC : towards programs with common sense", *Comm.ACM* **33#8**, 30 (August, 1990); D.B. Lenat, R.V. Guha : "The evolution of CycL, the Cyc representation language", *Sigart Bulletin* **2#3**, 84 (June, 1991).

5 : G.A. Creak : *Comp.Bull.Ser.4* **3#1**, 27 (February 1991).

6 : H.G. Wells : *The food of the gods*, Collins Library of Classics, undated, page 158.

7 : L.S. Stebbing : *A modern elementary logic*, Methuen, fifth edition, 1957, page 25.

8 : L. Carroll (C.L. Dodgson) : *Symbolic logic and The game of logic*, (Dover Publications, 1958).

9 : E. Charniak, D. McDermott : *Introduction to artificial intelligence* (Addison-Wesley, 1985), page 642.

10 : A.D. Allen : "A less arbitrary method for inferring cause and effect : generalization of a medical model", *IEEE Trans.Sys.Man Cyb.* **21**, 339 (1991).

11 : R.S. Michalski : "Pattern recognition as rule-guided inductive inference", *IEEE Trans. Pat. Anal. Mach. Int.* **2**, 349 (1980)

12 : J. Hallam : *Artificial intelligence and signal understanding*, DAI Research paper No. 384, Department of Artificial Intelligence, University of Edinburgh, 1988.

13 : P. Coad : "Object-oriented patterns", *CommACM* **35#9**, 152 (September, 1992).