19

Parallel and Serial Methods of Pattern Matching

D. J. Willshaw and O. P. Buneman
Theoretical Psychology Unit
University of Edinburgh

1. INTRODUCTION

This paper is concerned with two aspects of the 'exact match' problem which is that of searching amongst a set of stored patterns to find those specified by a given partial description. We describe how to design simple content-addressable memories, functioning in parallel, which can do this and which, in some sense, can generalise about the stored data. Secondly, we consider how certain graphical representations of data may be suitable for use in efficient serial search strategies. We indicate how such structures can be used in diagnosis when the availability or cost of tests to be applied cannot be determined in advance.

The type of parallel system to be considered is to store descriptions of a set of patterns, and is then to be used to supplement an incomplete description of a newly presented pattern by matching it against those in store. If this partial description matches one or more of the stored patterns then we would like the memory to provide us with the partial description that these patterns share. If the new pattern does not match any in store then we expect that the information supplied will be according to the relationships between the pattern presented and those in store. The information that we require our memory to provide when given an incomplete description as an address is therefore more than just the response 'yes' or 'no'. In this respect our type of system differs from content-addressable parallel memories used in computer technology, and for the same reason its capabilities exceed that of a switching network which is designed to respond positively when the states of its input channels attain one of a number of combinations of binary values (Richards 1971, Renwick and Cole 1971).

This paper generalises the work of Willshaw (1972) to ensembles which conform to few or no logical constraints. The graphical generalisation of the *multitree* which is used by Willshaw to represent the ensemble will be seen to have properties which suggest its use as a method of data retrieval in conventional storage systems. This method is applicable to the problem of reducing the amount of serial search needed to identify a pattern in store from a given partial description.

2. THE INDUCTIVE NET

We now review the properties of a parallel device called the Inductive Net (Willshaw 1972), which is closely related to the Associative Net (Willshaw, Buneman and Longuet-Higgins 1969). This is designed to store a selection of patterns chosen from a special type of ensemble. When presented with an incomplete description of an ensemble member it is able to use the information in store to supplement the description of that member. The structure imposed on the ensemble enables the Net to have the additional property of generalisation as it can augment descriptions of ensemble members with which it was not explicitly provided.

Each pattern with which the Net deals is in the form of a binary vector of fixed length, each component of the vector representing the value the pattern takes on a particular binary feature. The restriction placed on the ensemble is that no two features are logically independent – that is, for each pair of features not all the four possible combinations of feature values '++', '+−', '−+', and '−−' are possessed by members of the ensemble, which is said to obey the *four point condition* (Buneman 1971).

The Inductive Net is made up of a set of horizontal lines (input lines) crossing at right angles a set of vertical lines (output lines), binary switches being placed at the intersections so formed. Each possible feature value has one horizontal line and one vertical line identified with it. A pattern is stored in the Net by exciting the horizontal lines and the vertical lines identified with its feature values, and turning on each switch which receives excitation along both the lines on which it is placed. In the retrieval mode, the pattern to be used as the address or *cue* (which is usually incomplete) is input by exciting the appropriate horizontal lines. Each binary switch whose horizontal line is excited and which has been turned on sends a pulse of unit strength down its vertical line which then fires if the number of pulses sent down it equals the number of excited input lines. An inhibitory mechanism placed across each pair of output lines identified with the same feature prevents any signal emerging if both lines are simultaneously active. The set of feature values associated with the output produced in this manner is regarded as the response of the Inductive Net to the given cue.

As an example of how the Inductive Net functions figure 1 shows the Net which has stored the following patterns:

A:	+1	−2	+3	−4
B:	+1	+2	−3	−4
C:	+1	+2	−3	+4
D:	−1	−2	−3	−4.

Each pattern is represented by a binary vector of length 4. To assign a value '$+x$' to a pattern means that this pattern takes value '+' on feature x. Similarly for '$-x$'. From this list of feature values we note, for example, that a pattern with feature value $+2$ also has values $+1$ and -3. This simple correlation has been noted in the Net, for along the horizontal line labelled $+2$, switches $+1$ and -3 have been turned on and switches -1 and $+3$ have not. This line is therefore storing the 'rule' that a pattern with feature value $+2$ also has feature values $+1$ and -3. Similarly, we observe that a pattern with feature value $+3$ also has feature values $+1$, -2 and -4, so this 'rule' is stored in horizontal line $+3$. More complex rules such as 'the presence of $+1$ *and* $+2$ implies the presence of -3' have not been stored in the Net because each horizontal line is identified with only one feature value.

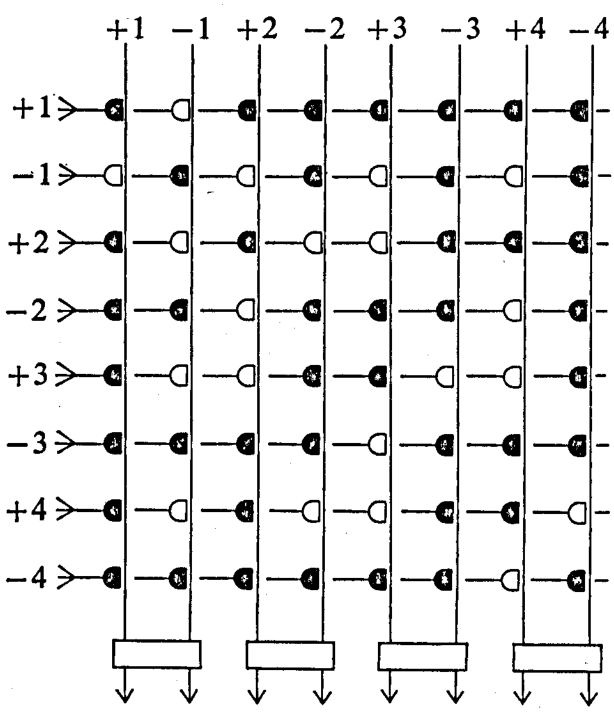

Figure 1. The Inductive Net which has stored patterns A, B, C and D. Switches which are on are coloured black.

In order to discuss the behaviour of the Inductive Net we need the fact that all the information that a selection of patterns from a four point ensemble can tell us about the ensemble itself can be represented by a family of trees which we can draw as a single structure called a *multitree*, in which each node represents a possible member of the ensemble and each link corresponds to the alteration of one or more features.

The multitree constructed from patterns A, B, C and D is shown in figure 2. Each of the four stored patterns is represented by a node and there is a node, which we label X, not identified with any pattern. The links are directed, and the arrows placed on them point towards that region of the multitree having value '+' on the associated feature, the remainder of the multitree taking the value '−'. The reader may like to verify that the multitree of figure 2 is the correct one by checking that its links assign the correct set of feature values to nodes A, B, C and D. (In this simple case each link is identified with only one feature, so that the multitree represents just one tree. In general the multitree represents more than one tree.)

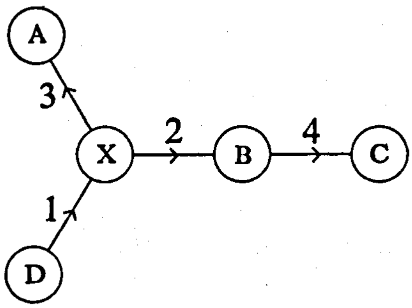

Figure 2. The multitree constructed from patterns A, B, C and D.

Let us imagine that we are given the descriptions of a number of patterns which we know were selected from a four point ensemble, and we are now asked to supplement incomplete descriptions of ensemble members. The appropriate multitree is a very useful tool for this task. First of all, it contains descriptions of all the members of the ensemble whose presence can be inferred from the given sample. That is the reason why the extra node X appears in our example – the four point ensemble from which patterns A, B, C and D were selected must also contain pattern X. Secondly, the multitree has the property that any incomplete set of feature values common to one or more ensemble members specifies that connected region of the multitree containing the nodes associated with them, and so the complete set of feature values common to these ensemble members can be read off from the multitree. For example, if we were asked to supplement the partial description $+4$, reference to figure 2 shows that feature value $+4$ specifies node C, and its complete description $+1+2-3+4$ can be read off from the multitree. Similarly the partial description $-2-3$ specifies nodes D and X which, as the multitree tells us, share feature values $-2-3-4$.

It is a straightforward exercise to prove that the Inductive Net can supplement partial descriptions just as well as we would expect from an examination of the multitree constructed from the set of patterns given to the Net to store (Willshaw 1972). Returning to our example once more, this means that if we input the description +4 to the Net of figure 1 then the output is +1+2−3+4, and the response to the cue −2−3 is −2−3−4.

These are two illustrations of the general theorem which reads:

> If the set of feature values used as input to the Inductive Net specifies a connected region of the multitree then the output is the complete set of feature values common to that region.

The most important property of the Inductive Net is that it is able to supplement descriptions which it was not explicitly given to store. This asset becomes undesirable if we do not wish the Net to generalise in this way. If, for example, we had defined our ensemble to comprise the patterns A, B, C and D and no more, the Net of figure 1 would still supplement partial descriptions on the assumption that X was present as well.

The Inductive Net is therefore not able to augment descriptions of arbitrarily chosen sets of patterns. It is convenient to divide this problem into two parts and consider each separately. We shall show in the next section that we can always produce a parallel system able to supplement descriptions of any set of patterns placed in store. However, if the system is to be able to generalise then the ensemble from which the stored set of patterns is taken must have one of a finite number of types of structure. Some of these types of ensemble do not in fact obey the four point condition. However, if the only hypothesis about the ensemble is that it is not four point (which is in effect a non-hypothesis) then no generalisations can be made. The possible generalisations are prescribed by graphical representations of data of the type discussed later, so we shall delay discussion about the making of generalisations until these representations have been introduced.

3. COMPLEX NETS WHICH DO NOT MAKE GENERALISATIONS

Let us suppose that we would like to design an Inductive Net to supplement the descriptions of A, B, C and D of figure 2, but not X, as it does at the moment. If, for example, we supply as input to the Net the partial description −2−3, this should be enough to distinguish D (−1−2−3−4) from A, B and C. However, reference to the multitree tells us that feature values −2−3 specify nodes D and X, and so the output from the Net is the set of feature values −2−3−4, no information about the value of feature 1 having been recorded on input lines −2 or −3.

The solution is to construct a parallel structure, which we call a *Complex Net*, to record more complex combinations of input and output feature values than the Inductive Net does. We do this by giving the Inductive Net additional horizontal lines, each of which has a mask placed on the front of it which causes the line to fire when a particular combination of feature values occur together in the input pattern. We say that a Complex Net is of size S if each mask looks at no more than S feature values to decide whether or not it should fire. In our example, if we add in a horizontal line which fires only when feature values -2 and -3 occur in the input pattern, this ensures that the output for the partial description $-2-3$ is now $-1-2-3-4$, as we require. In fact, for this Complex Net to supplement all possible descriptions of A, B, C and D we need the extra masks $+1-3$, $+1-2$, $-2-3$, so this Net is of size 2.

As a second example of the construction of Complex Nets we add to our patterns A, B, C and D two more called E and F. The descriptions of these six patterns are:

A:	+1	−2	+3	−4
B:	+1	+2	−3	−4
C:	+1	+2	−3	+4
D:	−1	−2	−3	−4
E:	+1	+2	+3	+4
F:	−1	−2	+3	−4

This set of patterns violates the four point condition; for example, the four pairs of feature values $+3+4$, $+3-4$, $-3+4$ and $-3-4$ all occur in the above list of feature values.

In order to find out what masks our Complex Net should have so that it may supplement descriptions of just these six patterns, and no more, we use an algorithm which is outlined in section 5 along with the other mathematical propositions. In fact, for our example we need a Complex Net of size 3 with 13 masks of size 2* or higher. These masks are listed below in table 1.

Table 1. Feature values prescribing the masks of size 2 or higher for the Complex Net storing patterns A to F.

+1−2	+1+2+3
+1−3	+1+2−4
−1+2	+1+3−4
−1−2	+2+3−4
−1+3	
−1−3	
+2−3	
−2+3	
−2−3	

This ends our discussion of how to design a Complex Net able to answer questions about any set of patterns placed in store. We shall show how Nets of this type can make generalisations after we have discussed certain graphical representations of data.

* The size of a mask is the number of feature values that it must look at in order to decide whether it should fire.

4. GRAPHICAL REPRESENTATIONS AND DIAGNOSTIC KEYS

Burkhard and Keller (1972) have shown that, in the 'closest match' problem, the construction of a graph on the patterns in store can cut down the amount of searching. This problem is that of finding those patterns in a conventional (computer) store which have least Hamming distance from a given 'test' pattern. Their method is to construct that graph on the patterns in store in which pairs of patterns are linked which are separated by a Hamming distance less than some number α. In the search for the patterns closest to the test pattern, if we have already found a pattern distance d from the test pattern we can subsequently discard from the set of closest patterns any pattern we find whose distance is greater than $d+\alpha$ from the test pattern and, without further calculation, we can reject all its neighbours in this graph. The degree to which the search is speeded up depends critically on the choice of α.

In this paper, we have been concerned with the 'exact match' problem, that of identifying a pattern or some of its properties from a partial description of that pattern. Is there a method of decreasing the search time which similarly employs some graph on the patterns? For the time being we shall assume these patterns are to be stored conventionally in either a list or an array and that graphs on these patterns can be constructed either by use of a two-dimensional binary array, or by the use of more complicated pointer structures. Call the n patterns to be stored $O_1, O_2, \ldots, O_n$ each consisting of k binary-valued components or features. A graph on these patterns will be called an F-graph if that graph, when restricted to those patterns which possess any given combination of feature values, is a connected graph. Put otherwise, a graph is an F-graph if for each conjunctive predicate $P = P_1 \cap P_2 \cap P_3 \ldots \cap P_n$ on a subset of the features we can find a connected subgraph whose points are just those patterns which satisfy P. Given an F-graph, we need search only along its links to achieve an exact match and the search can be conducted in such a way that, if we have found a pattern which satisfies some components of the predicate P, then we need move only along links on which these components remain unchanged. Thus a hill climbing procedure on an F-graph which optimises the match with P is bound to achieve an exact match if one exists.

This process is of no use unless the F-graph is such that each pattern is linked, on average, to only a fraction of the others. The complete graph is, trivially, an F-graph but is useless as a device for restricting the search. There is however, for any set of patterns, a unique minimal F-graph on those patterns, that is an F-graph whose lines are lines in any F-graph on those patterns. This graph can be constructed as follows: link every pair of patterns which are Hamming distance one apart by a line, and associate with this line a length of one. Call this graph G_1. Augment G_1 with lines of length two by joining points of Hamming distance two apart, provided these points are not already graphical distance two apart in G_1. Call this augmented graph G_2 and measure distances in G_2 by the shortest length of path (if one exists) joining two points. Now augment G_2 in the same way with lines of length three, and continue until the largest Hamming distance has been dealt with.

Although the minimal F-graph on the given patterns is well defined, it may be possible, by adding extra patterns, to construct an F-graph with fewer lines. One example of this is the tree-like data considered earlier, where by adding patterns we can achieve an F-graph which is a tree. In this case each new pattern has the property that for every *pair* of features there is some given pattern that has the same values on those features. As we shall see, properties of other F-graphs bear a close relationship to the properties of the networks of the previous section.

It is interesting to compare the use of these graphs with that of another essentially graphical construction, a diagnostic key. This is a number of instructions telling the user how to apply a set of tests to a specimen in order to place it in one of a number of previously determined classes. (Niemela, Hopkins and Quadling 1968; Pankhurst 1970). Such keys depend for their use on the assumption that the result of any test (feature value) can be established if required, and it is also in general impossible to use any partial description that might be available at the outset. While diagnostic keys are usually prepared for their efficiency of operation in terms of the number and cost of the tests necessary to identify an object, their drawback is their extreme inflexibility. We feel that computer diagnosis might benefit from the use of more descriptive data structures; the problem of deciding which tests to perform, when there is a choice, could be computed as the diagnosis proceeds, for it may only be then that the cost or availability of tests can be established.

5. COMBINATORIAL RESULTS AND GENERALISATIONS

We here summarise some useful definitions. Each *pattern* is a binary vector of length k. Associated with the ith component of a set of patterns is the ith *feature* f_i which carries each pattern onto its boolean value. An *elementary predicate* is a predicate of the form f_i or $\neg f_i$ for some feature and an *elementary disjunction* (or *conjunction*) is a disjunction (or conjunction) of elementary predicates.

An elementary disjunction (conjunction) D_1 *contains* an elementary disjunction (conjunction) D_2 if each elementary predicate in D_2 occurs also in D_1. A conjunction D is called a mask if $D \Rightarrow P_i$ for some elementary predicate P_i and there is no smaller conjunction D' contained in D for which $D' \Rightarrow P_i$. The order of a disjunction or conjunction is the number of elementary predicates it contains.

The following propositions hold:

(1) The masks are determined by the minimal elementary disjunctions which are true on every pattern. Equivalently, they are determined by the minimal false disjunctions.

(2) The set of patterns determine and are determined by their masks.

(3) Given a set Ω of patterns, a pattern O' satisfies all the minimal disjunctions for Ω of order less than or equal to some j if for each subset S of the features of order j there is a pattern O in Ω such that

$$f_i(O) = f_i(O') \text{ for all } f_i \text{ in } S.$$

The first of these propositions means that the masks can be calculated from the minimal true disjunctions. The problem of finding these is the same as finding the set of minimal covers contained in a given cover of a set. (A cover of a set S is a set of subsets whose union contains S.) There is a serial algorithm for finding these masks, details of which will be published elsewhere. It is an attractive possibility that the masks might also be calculated by an adaptive mechanism operating on an Inductive Net, but we know of no such method.

The second result ensures, first of all, that if a Complex Net contains all the masks for a set of patterns then those patterns alone are the only complete outputs that this Net can give. However, if an upper limit is placed on the size of the masks that this Net can have, there may be other patterns which have been effectively stored (Proposition 3). Returning to our example of the storage of patterns A to F, if the Net can only have masks of size 1 this means that no minimal disjunctions of order greater than 2 can be used to determine which patterns are in store. Therefore some minimal disjunctions are disallowed and so extra patterns will have been stored in the Net. As proposition 3 indicates, these are found by taking those of the set of 16 possible patterns which were not originally stored. For each of these, if we find that all of the $\binom{4}{2}$ pairs of feature values it possesses occur in the patterns originally stored, we regard this as being a stored pattern too. Having applied this procedure to each of these 10 patterns we are then able to draw the F-graph on the new set of stored patterns. This is shown in figure 3 and we observe that it contains many extra nodes. There are other F-graphs that we can construct from the patterns A to F. If our Complex Net were allowed to have masks of size 2 or less then we check *triplets* instead of *doublets* of feature values in order to find the extra patterns in store. By this means we produce the F-graph shown in figure 4 which contains one extra node. Finally, by checking *quadruplets* we obtain the F-graph which is constructed on the patterns themselves (figure 5). This is as it should be because in this case no minimal disjunctions are disallowed. We say that an F-graph is of order S if it was produced by allowing only minimal disjunctions of order S or less to determine the patterns represented in it.

The point about constructing graphs of this form is that a Complex Net of size $S-1$ – that is, each mask is allowed to look at no more than $S-1$ components of the input – will supplement incomplete descriptions of the patterns represented in the F-graph of order S which may mention patterns other than those given to the Net to store. It is in this sense that a Complex Net is able to perform generalisations. If a Net has had built into it an explicit assumption about the structure of the data it stores, this assumption

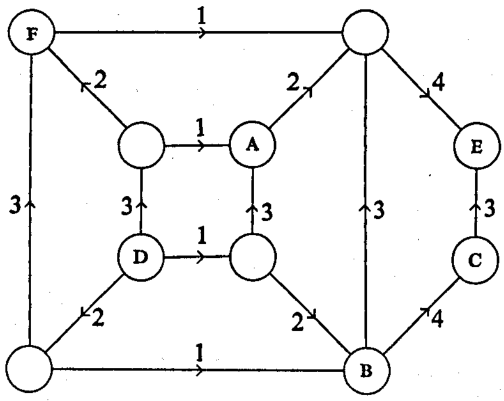

Figure 3. The *F*-graph of order 2 for patterns A to F.

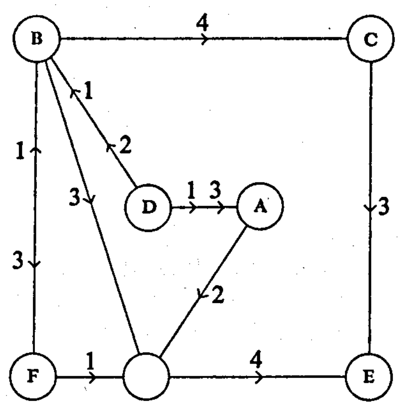

Figure 4. The *F*-graph of order 3 for patterns A to F.

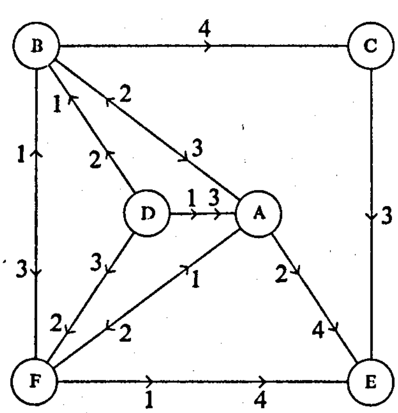

Figure 5. The *F*-graph of order 4 for patterns A to F.

taking the form of an upper limit to the size of the mask it can possess, then the Net will be able to generalise on this basis. On the other hand, if no assumption has been made (that is, there is no limit to the size of mask the Complex Net may have) then it will always supplement descriptions according to the *F*-graph constructed from the patterns given to the Net to store, and so will not be able to generalise.

The family of *F*-graphs that we can construct in this way may also have some application to the diagnosis problem. If we examine the three graphs we have just constructed, the *F*-graph of figure 4 contains fewest links. Assuming that we can mark invented nodes so that they can be discarded when required, it would seem that the *F*-graph with the smallest number of links would be the one to use in diagnosis, where links may be the expensive items. However, we know of no method for producing from a set of patterns the *F*-graph which is minimal in this respect.

We have programs for constructing *F*-graphs and masks for reasonably large data sets, and we hope to put these into practice in some diagnostic application.

Acknowledgements

The authors would like to thank Professor H.C. Longuet-Higgins and Mr C.J. Stephenson for many helpful discussions.

This work was financed by grants from the Science Research Council and the Royal Society.

REFERENCES

Buneman, O.P. (1971) The recovery of trees from measures of dissimilarity. *Mathematics in the Archaeological and Historical Sciences* pp. 387–95 (eds Hodson, F.R., Kendall, D.G. & Tăutu, P.). Edinburgh: Edinburgh University Press.

Burkhard, W.A. & Keller, R.M. (1972) Heuristic approaches to a file searching Problem. *Comp. Science Lab., Princeton Univ. Technical Report 103.*

Niemela, S.I., Hopkins, J.W. & Quadling, C. (1968) Selecting an economical binary test battery for a set of microbial cultures. *Canadian J. Microbiology*, **14**, 271–9.

Pankhurst, R.J. (1970) A computer program for generating diagnostic keys. *Comput. J.*, **13**, 145–51.

Renwick, W. & Cole, A.J. (1971) *Digital Storage Systems.* London: Chapman & Hall.

Richards, R.K. (1971) *Digital Design.* New York: Wiley.

Willshaw, D.J. (1972) A simple model capable of inductive generalisation. *Proc. Roy. Soc. B.*, **182**, 233–47.

Willshaw, D.J., Buneman, O.P. & Longuet-Higgins, H.C. (1969) Non-holographic associative memory. *Nature*, **222**, 960–2.