Hypercuboid-Formation Behaviour of Two Learning Algorithms

Chris Thornton

1987

Cognitive Studies Research Papers

Serial No. CSRP. 067

HYPERCUBOID-FORMATION BEHAVIOUR OF TWO LEARNING ALGORITHMS

Chris Thornton

Cognitive Studies Programme,
Arts Building D,
University of Sussex,
Brighton,
England

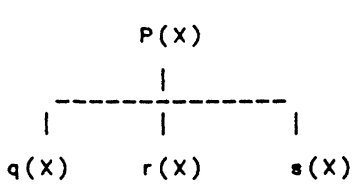
Tel. Brighton 606755

ABSTRACT

Bundy et al (1985) have provided an analytical comparison of a number of rule-learning programs including the Focussing algorithm and the Classification algorithm. They analyse the behaviour of these algorithms in the case where the description space consists of a set of relation trees. However, it is possible to add some interesting footnotes to their analysis if we take the step of re-construing the description space as a geometrical space. Under this construal, the behaviour of both the Focussing algorithm and the Classification algorithm is analysed in terms of the construction of hypercuboids. This analysis leads to a number of observations: (i) that there is at least one, novel generalisation of the Focussing algorithm, (ii) that a heuristic-based strategy for coping with the disjunctive-concept problem will, in general, involve the exploitation of a valid metric over the description space and (iii) that the class of disjuntive rules (or concepts) which can be constructed by the Classification algorithm (Hunt et al, 1966) is characterised by some quite specific geometrical constraints.

1. Introduction

The analysis provided by Bundy et al (ibid.) shows that a number of rule-learning programs are quite closely related and that many of them are subsumed by the concept learning algorithm associated with Young et al (1977). This algorithm, called the Focussing algorithm, is described in some detail. The description concentrates on the case where the description space consists of a set of relation trees. A relation tree is characterised as follows.



Implies that

p(X) <-> Exactly one of {q(X), r(X), s(X)}

It is noted that a description space consisting of such trees allows a partially specified rule to be represented. The most general form of a partially specified rule is defined by associating an "upper marker" with a specific node in each relation tree. The most specific form of a partially specified rule is defined by associating a "lower marker" with a specific node in each relation tree. If upper and lower markers in a single tree coincide, then the tree is described as "firmed-up". If upper and lower markers in all trees coincide then the rule (or concept) is completely specified.

The Focussing algorithm, as described by Bundy et al, consists basically of two processes: generalisation and specialisation (which Bundy et al call "discrimination"). Specialisation becomes possible following the presentation of a negative training instance; i.e. an instance for which the rule being learnt should fail, or to which the concept being learnt is not applicable. It entails moving upper markers downwards in their respective relation trees so that the most general form of the rule is not satisfied by the negative training instance. Similarly, generalisation becomes possible following the presentation of a positive training instance and entails moving lower markers upwards so that the most specific form of the rule is satisfied by the training instance.

As Bundy et al point out, this basic algorithm generalises a number of learning programs, e.g. (Waterman, 1970), (Langley, 1981) and (Brazdil, 1978), and is similar to the program described in (Mitchell et al, 1983). It has been noted that the idea underlying the Focussing algorithm resonates with certain ideas in epistemology. Charniak and McDermott have in fact, referred to (a version of) it as the "empiricist algorithm" in recognition of the fact that it has roots in theories of the early empiricist philosophers (Charniak and McDermott, 1985, p. 614). The algorithm is capable of producing quite interesting learning behaviour; the most well-known example is perhaps Winston's learning system which was able to learn appropriate descriptions for a variety of blocksworld concepts (Winston, 1975).

2. Focussing in a geometrical description space

Consider the case in which we construe the description space described above as a

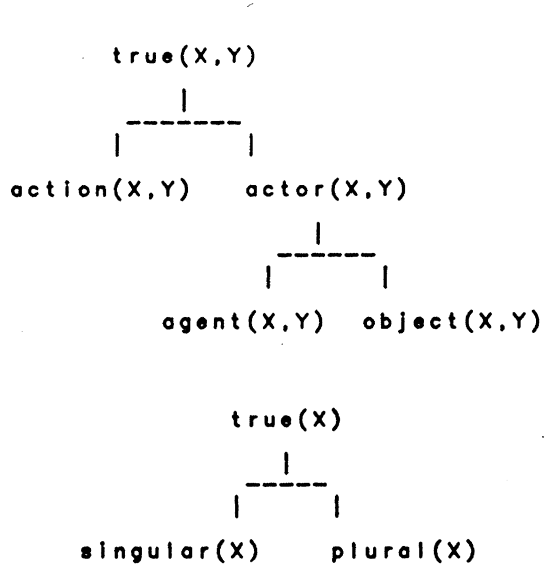


Fig.1.

Consider the description space consisting of the two relation trees depicted in Fig. 1, (part of an example used by Bundy et al, 1985). This description space translates into a geometrical space of three dimensions. One of these dimensions is a special case. It ranges over the set of values {positive, negative}; I call it the P/N dimension below.

The second dimension ranges over the set of values {singular(X), plural(X)}. This entire set of values is associated with the value true(X). The third dimension ranges over the set of values {action(X,Y), agent(X,Y), object(X,Y)}. The subset of contiguous values {agent(X,Y), object(X,Y)} is associated with the value actor(X,Y) while the set of values {action(X,Y), agent(X,Y), object(X,Y)} is associated with the value true(X,Y). Values from different relation trees having the same lexical form are assumed to be distinct.

In this construal the placing of markers in relation trees corresponds to the formation of hypercuboids. (A hypercuboid is defined as the n-dimensional generalisation of a rectangle.) Note that any marker (either upper or lower) must be associated with a node in a relation tree. It must therefore be associated with a set of contiguous values in the corresponding dimension. In the case where a marker is placed at the root node, it is associated with the entire set of values in the dimension.

For each marker therefore (call it m1) we can derive two values which are the extremal elements of the set of contiguous values associated with m1's host node. These extremal values define the position of two boundaries of the hypercuboid for the relevant dimension.

The application of the Focussing algorithm to markers placed in relation trees corresponds, roughly, to a process in which an "outer" hypercuboid is shrunk, and an "inner" hypercuboid expanded until they coincide. Note that the presentation of a positive training instance corresponds to the instantiation of a point in space whose coordinate in the P/N dimension is the value "positive". Obviously, the response of the Focussing algorithm corresponds to an expansion of the inner hypercuboid so that it encloses the new point. Conversely, the presentation of a negative training instance corresponds to the instantiation of a point whose coordinate in the P/N dimension is the value "negative". The response of the Focussing algorithm corresponds to a shrinking of the outer hypercuboid so that it excludes the new point (but see below).

We can depict the description space described above using a two-dimensional figure such as Fig. 2. In this figure, the two "real" dimensions correspond to the two major axes and specific training instances correspond to the "P" and "N" characters appearing in the different cells of the figure. All instances represented as Ps are points having the value "positive" in the P/N dimension while all instances represented as Ns have the value "negative" in this dimension. Positive and negative training instances are associated with values of the real dimensions in the obvious way. Note that below, we will refer to positive training instances as "Ps" and to negative training instances as "Ns".

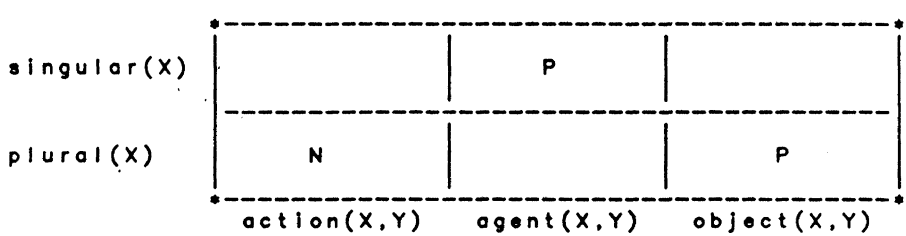


Fig. 2

Consider the behaviour of the Focussing algorithm in the case where it is presented with the following sequence of training instances.

plural(X) & object(X,Y)	(positive)
singular(X) & agent(X)	(positive)
plural(X) & action(X,Y)	(negative)

These training instances are in fact the ones represented in Fig. 2.

In the standard construal, the Focussing algorithm is initialised via the presentation of the initial positive instance. This leads to upper and lower markers being associated with nodes in relation trees in the appropriate way. Following presentation of the second positive instance generalisation will take place. That is to say lower markers will be raised so as to cover the new case. Following the presentation of a

of the inner hypercuboid and presentation of the final, negative instance leads to a shrinking of the outer hypercuboid so as to exclude the negative instance. At this point the inner and outer hypercuboid are identical and the algorithm terminates. The form of the final hypercuboid is represented by the asterisks in Fig. 3.

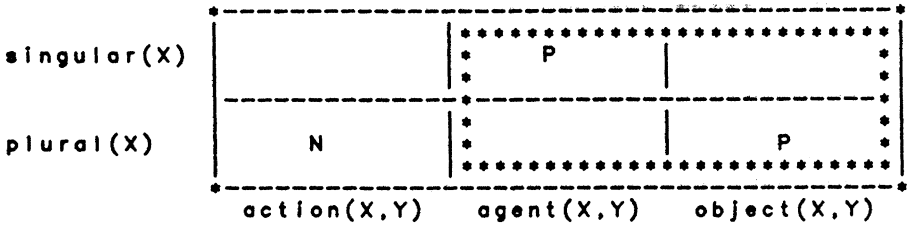


Fig. 3

3. Generalised Focussing

Our construal leads us to distinguish between the standard Focussing algorithm and an interesting generalisation of it. Recall that the standard Focussing algorithm involves the movement of markers. As we have noted, this behaviour can be construed in terms of the manipulation of hypercuboid boundaries. However, it is important to note that the Focussing algorithm is only capable of implementing a subset of the possible hypercuboid boundary manipulations.

In effect, it can only move boundaries between positions enclosing the extremal elements of the set of contiguous values associated with a pair of nodes having a parent-child relationship. This means that it cannot, for instance, position boundaries so as to enclose the two values action(X,Y) and agent(X,Y) in Fig. 3.

It is credible that a Focussing algorithm capable of implementing arbitrary boundary movements might have interesting properties. Consider the dimension corresponding to a relation tree of depth one whose leaf nodes are the values past(Y), present(Y) and future(Y). The standard Focussing algorithm can only generalise the first two values by moving a lower marker to the root node. In effect, the generalisation is thereby forced to embrace the future(Y) value in addition to the past(Y) and present(Y) values. Obviously, there might be situations in which a generalisation of past(Y) and present(Y) might advantageously exclude the future(Y) value. We therefore infer, that a generalised Focussing algorithm, capable of implementing arbitrary boundary movements, might have an improved performance.

4. The far-miss problem

In the standard construal of the Focussing algorithm it is the case that specialisation (or discrimination) can take place in a variety of different ways. This is due to the fact that moving an upper marker down in any one relation tree will generally be

non-determinism can lead to unfortunate consequences.

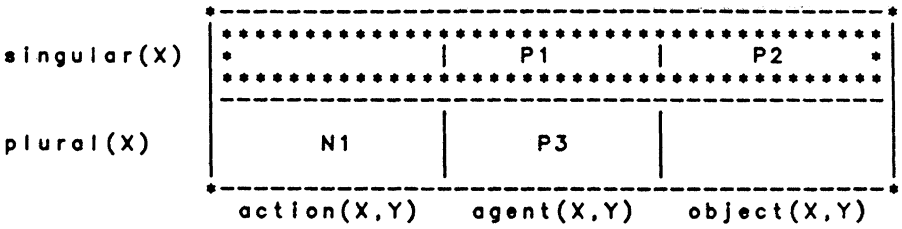


Fig. 4

Fig. 4 depicts the form of the outer hypercuboid following the presentation of two positive instances (P1 and P2) and one negative instance (N1). The outer boundary has been shrunk to exclude N1. This shrinking involved the movement of the lower boundary so as to exclude the plural(X) value. The next instance presented is a positive instance of the form plural(X) & agent(X,Y) (P3). Clearly, if we expand the inner hypercuboid so as to include P3 an anomolous situation will result in which the inner hypercuboid extends beyond the outer hypercuboid. This situation cannot be dealt with by the basic algorithm and seems to require exhaustive searching of the different possibilities. The general conclusion is that an incorrect, deterministic shrinking of the outer hypercuboid may lead to a failure of the Focussing algorithm (see also Bundy et al, 1985, p. 159).

5. The disjunctive-concept problem

The analysis provided by Bundy et al includes a description of the way in which the Focussing algorithm can fail in the case where the rule being learnt has a disjunctive form. They show that when a certain sequence of positive and negative instances are presented to the algorithm, situations will arise in which either upper markers are moved below lower markers or in which lower markers are moved above upper markers. The rule corresponding to the resulting configuration of markers is clearly meaningless: its most specific form will be more general than its most general form.

When we construe the description space as a geometric space, this problem (the so-called disjunctive-concept problem) appears as a static property of a set of instances, which is not affected by the presentation schedule.

Consider Fig. 5. It is obvious that we cannot form a hypercuboid in this space which both encloses all the Ps and excludes all the Ns so we can infer that the Focussing algorithm cannot produce a completely specified rule for this collection of instances. One way out is to assume that either the negative instance or one of the positive instances corresponds to noisy data and can therefore be ignored. However, if we accept all the instances as valid, we have to conclude that the concept corresponding to the positive instances must have a disjunctive description; e.g. (plural(X) & object(X)) OR (singular(X) & agent(X)).

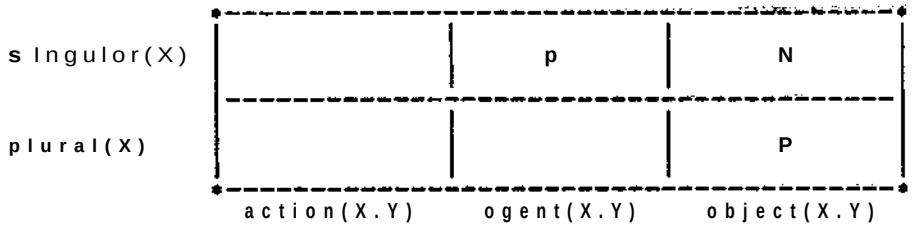


Fig. 5

The general conclusion is as follows: whenever points corresponding to negative instances fall inside the smallest hypercuboid enclosing all the positive instances, it is the case that the Focussing algorithm cannot produce a completely specified rule describing«the instances. Note that the sequence in which instances are presented makes no difference.

6. Shell Creation

Bundy et al show that a number of Focussing-related learning programs try to implement solutions to the disjunctive-concept problem (ibid.). They also note that none of the solutions are particularly satisfactory. One technique discussed is "shell creation". This technique is described as a modification of the Candidate Elimination Algorithm reported in Mitchell et al (1983). The process of Shell Creation involves the construction of what Bundy et al call "rule-shells[11] and is introduced as follows.

> **To cope with inconsistencies caused by disjunctive rules it is necessary to introduce a new rule-shell and to divide the positive instances between the old and the new shells. Negative instances should apply to both shells (p. 167).**

Clearly, under our construal. new rule-shells correspond to new hypercuboids. The interpretation of Bundy et al*s conclusion follows automatically. To cope with the case where it is not possible to construct a single hypercuboid which encloses all Ps and excludes all Ns. it is essential to introduce multiple hypercuboids. Having done this, it then becomes necessary to decide which Ps should be considered to belong to (and therefore expand) which hypercuboids. Ns are easily processed since any N can be used to shrink any hypercuboid.

Bundy et al suggest that all solutions to the disjunctive-concept problem which use techniques similar to Shell Creation are flawed. They sum things up as follows.

> **these "solutions" are far from perfect, and rely on very favourable training orders. It seems to us that any adaption of Focussing to learn disjunctive concepts correctly ... must include storage of all the training instances. This is due to the problem of deciding which instances belong to which disjunct. We know of no way of ensuring that the division of instances is**

might exist through which the division (or, in general, the assignment) of instances might be performed correctly, first time?

Consider Fig. 6. This figure depicts a situation in which five positive training instances have been presented. Two inner hypercuboids are represented. Each one encloses two positive instances. The hypercuboid enclosing P1 and P2 will be referred to as the leftmost hypercuboid. The hypercuboid enclosing P3 and P4 will be referred to as the rightmost hypercuboid.

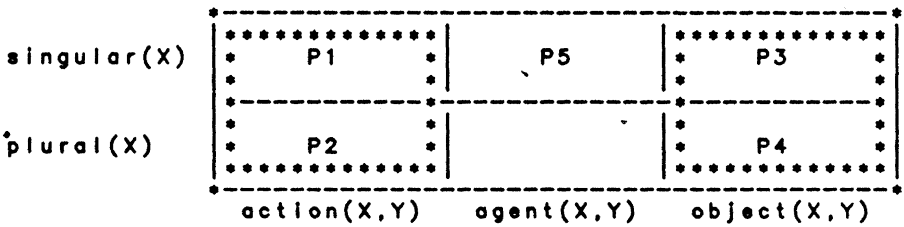


Fig. 6

Clearly, if we have a heuristic for performing correct assignments first time, then we should be able to use it to decide whether P5 should be assigned to the leftmost or the rightmost hypercuboid. The question is could such a heuristic exist? and if so, how would it work?

Note that we will expect the heuristic function to satisfy a number of criteria: (i) it must produce definite and consistent results; (ii) it must not refer to any implicit properties of the particular instances presented, or to any implicit properties of the particular description space involved; (iii) it must compute the same result, regardless of the way in which we pose the problem.

In addition we will expect that it will generate preferences which are intrinsically coherent. This means that we will expect that the generated preference for assigning some specific instance (call it P1) to some specific hypercuboid (call it H1) will not be "outweighed" by the generated preference for assigning P1 first to some intermediate hypercuboid and then on to H1. If generated preferences were of this sort then we would be able to increase our level of preference for any particular assignment simply by devising ever more elaborate assignment schedules. Finally, we will expect our heuristic to function normally in the special case where hypercuboids always enclose single points.

Clearly, any heuristic satisfying these criteria must exploit what is, in effect, a valid metric over the description space. Presented with any two points in the description space, the heuristic must be capable of generating a distinct level of preference for assigning one of the two points to a hypercuboid enclosing the other. Note that this level of preference, interpreted as a distance measure, satisfies all the axioms for a valid metric, including triangle inequality.

8. The Classification algorithm

Bundy et al describe an algorithm, called the Classification algorithm, which, they say. can deal flawlessly with disjunctive concepts. The particular example they discuss is Quinlan's ID3 program. The basic behaviour of this algorithm is well-documented, e.g. (Quinlan. 1983). (Bundy et al, 1985). (Hunt et al, 1966).

The processing of positive and negative instances by the Classification algorithm leads to the construction of a decision tree. Each node of this tree" corresponds to an attribute and its branches correspond to the possible values of the attribute. The leaf nodes of the tree are associated with sets of instances which are either empty or contain only positive or negative instances.

The initial step in the Classification algorithm involves choosing an attribute on which to split the instances up into "classes". This choice is made from a set of attributes (call it A).

Initially A is simply the set of all attributes. Any attribute can be chosen from A and an information-theoretic criterion is used to implement the choice. This states that the attribute selected will be the one which achieves the greatest reduction of the entropy of the distribution of instances. Intuitively, it states that an attribute in which positive and negative instances are less mixed-up will be preferred to one in which they are more mixed-up.

If an attribute with K possible values is selected, then K classes will be formed each of which is associated with one of the values of the dimension. Each of these classes contains all those instances which have the value associated with the class, The partitioning of instances into classes according to the values of an attribute X is referred to as "splitting on X" (see Bundy et al. 1985. p. 173). The set of attributes corresponding to the set of classes which contain a mixture of positive and negative instances is then derived. This set is now assigned to be the new value of A and the process continues recursively.

9. Hypercubes formed by the Classification algorithm

It is interesting to consider the behaviour of the Classification algorithm in terms of hypercuboid formation. To do this we have to construe the description space described above as a geometrical space.

As before, we assume that the description space has one special dimension whict ranges over the values "positive" and "negative" (P and N) and K real dimensions We assume that each of these K real dimensions corresponds to a specific attribute and ranges over the attribute's set of possible values. In the case where there are two attributes we will obviously be able to represent positive and negative traininj instances using the familiar two-dimensional diagram, e.g. Fig. 7.

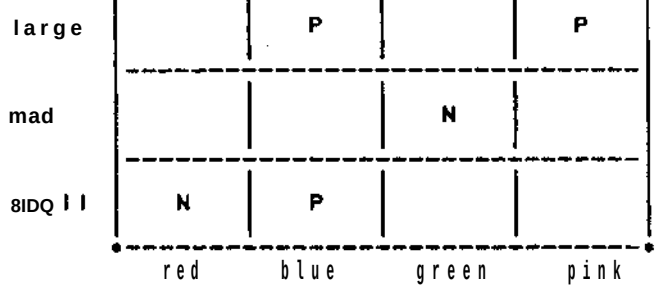


Fig. 7

Fig. 7 depicts the geometrical description space corresponding to the case where there are just two attributes under consideration: a "size" attribute and a "colour" attribute. The possible values of the "size" attribute are {large, med, small}, while the possible values of the "colour" attribute are {red, blue, green, pink}. The diagram depicts the "size" attribute and the "colour" attribute as dimensions in the usual way: three positive training instances and two negative training instances are shown. We now consider the way in which the Classification algorithm would process this collection of instances and how we should construe its behaviour in terms of hypercuboid formation.

Note that the dimension (attribute) along which the Ps and the Ns are least mixed-up is the "colour" dimension (attribute). This dimension will therefore be chosen first for splitting. The classes that result from this splitting contain only Ps or Ns. The algorithm therefore terminates.

The disjunctive concept constructed by this algorithm in this example corresponds to the decision tree depicted in figure 6. It corresponds to a disjunctive rule for the given positive instances: blue(X) OR pink(X).

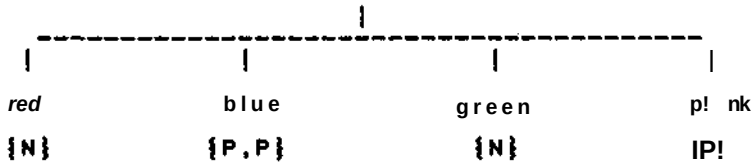


Fig. 8

Note that each time the Classification algorithm splits on an attribute it creates a set of K hypercuboids boundaries, where K - 1 is simply the number of values over which the dimension (attribute) ranges. Each of these boundaries separates two possible values of the selected dimension.

In the initial case, a set of K hypercuboids is formed as a result. The boundaries of these hypercuboids in the other dimensions enclose the extremal values in these dimensions. If the algorithm terminates after the first splitting then the struture of the decision tree will correspond exactly to this set of N hypercuboids. Fig. 9 depicts the set of hypercuboids constructed by the Classifiction algorithm for the exam-

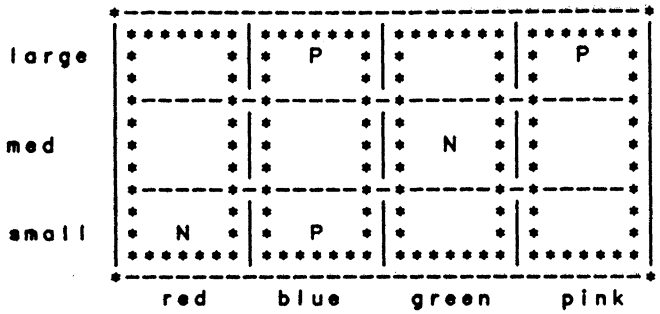


Fig. 9

If the algorithm does not terminate after the first splitting then the next best dimension will be selected and classes derived in the described way. As in the initial case, this process effectively identifies K - 1 boundaries. However in contrast, these boundaries only apply to hypercuboids which contain both Ps and Ns. In effect all hypercuboids which contain both Ps and Ns are split up into K "sub-hypercuboids" (K being the number of values in the selected dimension). The process continues until all hypercuboids contain only Ps or Ns.

Construing the behaviour of the Classification algorithm in terms of hypercuboid formation enables us to directly compare the (hypercuboid formation) behaviour of the Focussing algorithm with the described behaviour of the Classification algorithm. We note immediately that the Classification algorithm differs from the Focussing algorithm in that it is able to construct more than one hypercuboid. This is of course the property that underlies its ability to form disjunctive concepts.

We should also note that whereas the Focussing algorithm can, in principle, form arbitrarily shaped hypercuboids the Classification algorithm is restricted to forming hypercuboids which are, geometrically speaking, highly constrained. It is in fact the case that the Classification algorithm can only form hypercuboids whose two boundaries in any one dimension enclose either a single value of the dimension, or its extremal values. In simple terms, the boundaries in any one dimension of a hypercuboid formed by the Classification algorithm must, definitionally, be either as close as they can possibly be, or as distant as they can possibly be.

If we consider the entire set of possible hypercuboids which can be constructed in any given geometrical space of a reasonable size, and compare it with the set of hypercuboids which can be constructed in the same space using boundaries which are of the described form, we will conclude that the size of the former set is overwhelmingly greater than the size of the latter set. The implication is that the Classification algorithm can only construct a small proportion of possible hypercuboids in any given space and must therefore be assumed to be only capable, in general, of constructing a small proportion of the possible disjunctive rules for a given set of instances.

"generalized graph" description spaces into geometrical spaces. However, our arguments revolve around a reconstrual rather than a mapping; therefore they are not invalidated by Mitchell's observation.

We have shown that under the described interpretation a number of possibilities emerge: (i) a generalisation of the Focussing algorithm, which might have useful properties, can be envisaged; (ii) a very simple account of the disjunctive-concept problem can be provided which does not refer to the details of a presentation schedule and which can be supported with a simple graphical illustration; (iii) an argument can be constructed suggesting that any systematic solution to the disjunctive-concept problem will involve the exploitation of what is. in effect, a valid metric over the description space and (iv) it can be demonstrated that the Classification algorithm is only capable of forming a small subset of the possible disjunctive concepts for any given description space.

ACKNOWLEDGEMENTS

I would like to thank Allan Ramsay. Guy Scott and Steve Draper for providing extremely helpful comments on this paper.

REFERENCES

Brazdil, P.. "Experimental learning model" *AISB/GI,* Society for the Study of Artificial Intelligence and Simulation of Behaviour. 1978. pp. 46-50.

Bundy. A.. Silver. B. and Plummer, D., "An Analytical Comparison of Some Rule-Learning Programs" *Artificial Intelligence* 27, 1985. pp. 137-181.

Charniak. E. & McDermott. D.. *Introduction to Aritificial Intelligence.* Addison-Wesley. 1985.

Hunt. E.B.. Marin, J. and Stone. P.J.. *Experiments in Induction,* New York: Academic Press. 1966.

Langley. P.. "Language acquisition through error recovery" CD? Working Paper 432, Carnegie-Mellon University. Pittsburgh. PA. June 1981.

Mitchell. T.M. "Generalization as Search" *Artificial Intelligence* 18. 1982. pp. 203-226.

Mitchell. T.M.. Utgoff, P.E. & Banerji. R.. "Learning by Experimentation: acquiring and modifying problem-solving heuristics" in R.S. Michalski. J.G. Carbonell and T.M. Mitchell (Eds.) *Machine Learning* Tioga, Palo Alto. CA, 1983. pp. 163-190.

Quinlan. J.R.. "Learning Efficient Classification Procedures and their Application to Chess End Games" in R.S. Michalski. J.G. Carbonell and T.M. Mitchell (Eds.) *Machine Learning* Tioga, Palo Alto. CA. 1983. pp. 463-482.

Waterman. D.A.. "Generalization learning techniques for automating the learning of heuristics." *Artificial Intelligence* 1, 1970. pp. 121-170.

Winston, P.H.. "Learning Stuctural Descriptions from Examples" in *The Psychology of Computer Vision.* P. H. Winston (Ed.). Mcgraw-Hill. 1975.

Young. R.M.. Plotkin, G.D. and Linz. R.F.. "Analysis of an extended concept-learning task" in R. Eddy (Ed.). *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* Cambridge. MA. 1977. p. 285.