An Empirical Comparison of ID3 and Back-propagation*

Douglas H. Fisher and Kathleen B. McKusick
Department of Computer Science
Box 67, Station B
Vanderbilt University
Nashville, TN 37235

Abstract

AI and connectionist approaches to learning from examples differ in knowledge-base representation and inductive mechanisms. To explore these differences we experiment with a system from each paradigm: 1D3 and back-propagation. We compare the systems on the basis of both prediction accuracy and length of training. The systems show distinct performance differences across a variety of domains. We identify aspects of each system that may account for these performance differences. Finally, we suggest paths for cross-paradigm interaction.

1 Introduction

Research in machine learning has grown rapidly in recent years. A primary focus of study has been methods of *learning from examples:* a system accepts object descriptions (e.g., patient case histories) that are pre classified (e.g., hypothyroid disease). Based on this training, the system forms a knowledge base that can accurately classify new objects. Historically, the dominant approach in AI assumes that the knowledge base is a flat or tree-structured set of concept, descriptions. Typically, each concept is a logical rule that defines class membership.

In contrast, *connectionist* methods (Hinton, in press; McClelland & Rumelhart, 1988) assume a knowledge base of interconnected nodes, each of which computes a weighted sum of its inputs. External inputs (i.e., object features) are arithmetically combined and propagated through the network. This process terminates with the computation of external outputs that represent an object's classification. Learning alters weights so that classification correctness is improved.

AI and connectionist approaches typically differ in object and knowledge-base representation, as well as the inductive mechanisms employed. This paper explores some implications of these differences by experimenting with a system from each paradigm: ID3 and back-propagation. Sections 2 and 3 describe ID3 and back-propagation, respectively. Section 4 compares the systems in terms of the prediction accuracy attained in natural and artificially-constructed domains, and the amount of training required to achieve these accuracy levels. Section 5 describes processing and representation differences between ID3 and back-propagation, as well as between these systems and others of their respective paradigms. This discussion qualifies our study as it might relate to paradigm-wide comparisons and suggests foundations for work on hybrid systems.

*This work was supported by a grant from the Vanderbilt University Research Council.

2 ID3

ID3 (Quinlan, 1986) is a simple and effective AI method for learning from examples. The system constructs a *decision tree* from a set of training objects. At each node of the tree the training objects are partitioned by their value along a single attribute. An information theoretic measure is used to select the attribute whose values improve prediction of class membership above the accuracy expected from a random guess. The training set is recursively decomposed in this manner until no remaining attribute improves prediction in a statistically-significant manner by a user-supplied parameter of'confidence' (e.g., 90%). In our experiments we assume *nominal* attributes: those with a finite set of values (e.g., Color £ {red, blue, green}).

ID3 decision trees are equivalent to tree-structured Disjunctive Normal Form (DNF) concepts. Each path to a leaf is a conjunction of values, joined at the root by disjunction. Quinlan and others have verified that this approach attains high levels of accuracy in an absolute sense and relative to other systems. These studies report favorable results in several natural domains, under idealized and noisy conditions.

3 Back-propagation

In a feed-forward connectionist net, input nodes record observed features from the environment and pass 'activation' forward through an intermediate layer of 'hidden* nodes to an output layer. We assume that each node is

linked to every node at the next layer via weighted interconnections. The total activation of a node is a weighted sum of its inputs. We encode nominal attributes using a set of input units, one dedicated to each value. For a particular object description, one of these units (e.g., 'red') will be 1.0 and the rest (e.g., 'blue', 'green') will be 0.0, representing feature presence and absence respectively. A set of such input units is allocated for each attribute. This representation has been used by Sejnowski and Rosenberg (1988) and has advantages for nominal attribute encodings. Our convention is that each output node corresponds to one class; the object is classified by the class whose output node has the highest activation. Ideally, for any particular object the activation of one output node should be 1.0 and the others should be 0.0.

Back-propagation (Rumelhart, Hinton, & Williams, 1986; McClelland & Rumelhart, 1988) adjusts weights so as to improve the match between actual and ideal output. If there are 4 classes (output units) and an object belongs to class 1, then the ideal output is (1 0 0 0). If the actual output is (0.3 0.2 0.5 0.3) then the error for each output unit (0.7 -0.2 -0.5 -0.3) is back-propagated through the network. Weight adjustment is proportional to the size (and sign) of the error, and the degree to which the lower-level node contributed to the output node's node's activation error. A user-supplied parameter of 'learning rate' is used to vary weight adjustment.

With 'sufficient' hidden units, back-propagation can converge on perfect classification (assuming no noise), but this number will vary with domain. When no hidden units are present, *linearly separable* classes are recognizable, which properly include 'X of N' functions: for a selected subset of the input features (where the subset is of size N), *at least* X must be present in order to qualify for class membership. Logical conjunction and inclusive disjunction are special cases of the X of N function: X=N and X=1 represent the conjunctive case where all features must be present and the disjunctive case where only one feature need be present.

4 Experimental Comparisons

This section describes empirical comparisons between ID3 and back-propagation. Comparisons of any kind, whether cross-paradigmatic or not, require that we justify system-dependent parameter settings, choose domains and encodings that are fair to both systems, and accept that systems may be designed for disparate applications. Even if we agree on domains that allow the approximation of fair comparison, we must realize that systems may be superior along different dimensions (e.g., cost versus correctness). This section compares the behavior of two systems. Hopefully, we and the reader can avoid unfounded generalizations with respect to the paradigms more generally. Section 5 analyzes system performance in light of paradigm-wide assumptions in order to qualify our comparative results and to suggest paths for cross-paradigm fertilization.

4.1 Experimental Design

ID3 and back-propagation were tested in the natural domains of thyroid disease case histories, soybean disease case histories (Stepp, 1984), and congressional voting records. In addition to the basic domains, we systematically introduce noise to each domain. Finally, we test each system in a number of artificial domains, including *exclusive-or* which is not linearly separable, but which may be described by a simple logical expression.

For each domain, back-propagation was tested with varying numbers of hidden units (0, 10, 20) and learning rates (0.05, 0.10, 0.20). Of these, we report results with 10 hidden units and a learning rate of 0.05, which consistently optimizes or comes close to optimal asymptotic prediction accuracy and learning speed over all domains. ID3 was tested with varying confidence levels (0%, 90%, 95%, 99%). In general, 90% confidence does as well as others for domains, training schedules, and noise levels that we are investigating.[1]

All features were nominal and were given a binary encoding for back-propagation as described in Section 3. Class membership was similarly encoded for outputs. In all of the natural domains, back-propagation required far more object 'presentations' to converge on asymptotic accuracy than there were unique objects. Thus, training objects were drawn randomly (with replacement) from a fixed pool of objects. For 1D3 and back-propagation a disjoint object subset was reserved to test (but not update) the knowledge base at intermittent points in training. Back-propagation is incremental; after each testing point, learning resumes with the network weights derived by previous training. In contrast, 1D3 is nonincremental; at each testing point 1D3 is constructed anew with the training set used previously, plus newly presented objects. ID3 may see each object at most once for any particular trial.

4.2 Natural Domains

The graphs of Figure 1 show the learning curves' of ID3 and back-propagation in the congressional and thyroid domains. In the congressional domain ID3 achieves and maintains 94% accuracy after approximately 200 objects, while back-propagation asymptotes at 97% accuracy after 1000 objects. In the thyroid domain, ID3 requires 135 objects to attain 88% accuracy. Rack-propagation reaches and maintains 91% accuracy after 1000 object presentations. In the soybean domain ID3 averaged 99%

[1] However, we report exceptions to these findings when they significantly improve performance (e.g., 0% versus 90% confidence; 0 versus 10 hidden units).

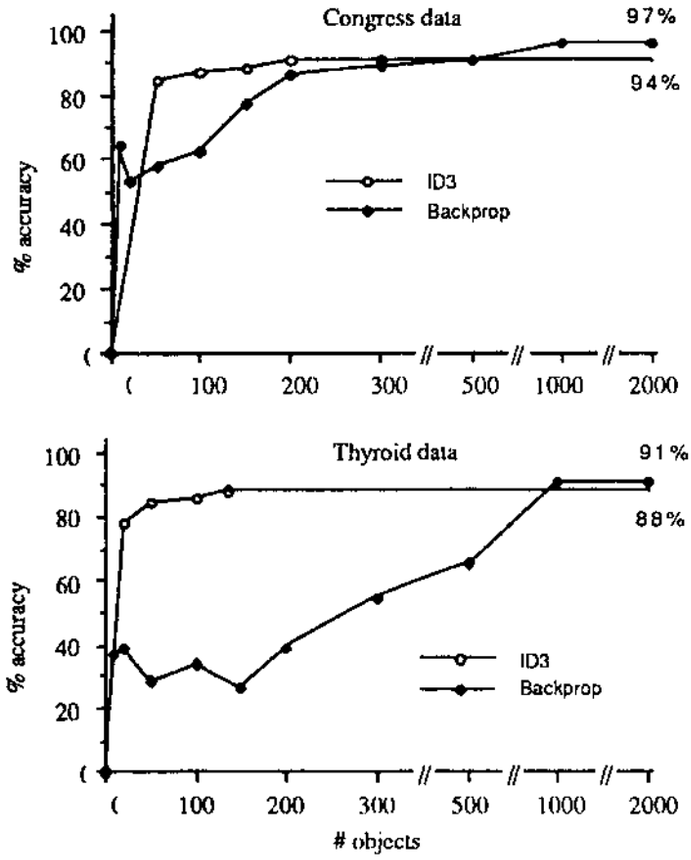

Figure 1: Accuracy as a function of training.

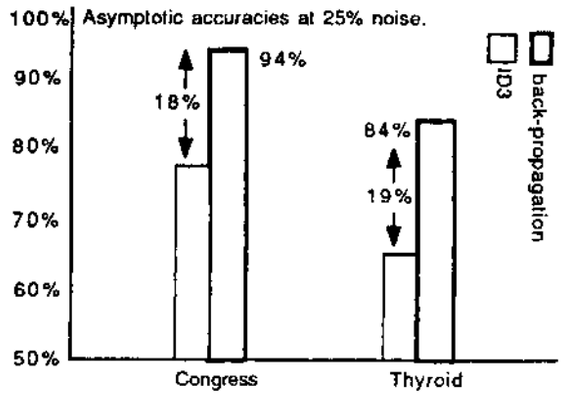

Figure 2: Asymptotic accuracy at 25% noise (differences significant by t test: $\alpha = 0.05$).

accuracy over the test set. Back-propagation reached perfect accuracy after 500 object presentations.

In these domains back-propagation reaches slightly higher accuracy levels, but not significantly so. Moreover, back-propagation requires many more training presentations. Another difference relates to the shape of the 'learning curves': ID3 quickly achieves high levels of accuracy (e.g., after 10 training objects or so) and then gradually converges on its asymptotic value. In contrast, the slope of back-propagation's curve is more gradual and uniform.

Our training conventions for back-propagation assume that an object *presentation* is the primary unit of cost. In contrast to our *incremental* approach, Shavlik, Mooney, and Towell (1989) assume that the training objects are repeatedly presented until the network converges to near perfect prediction of this set. Only then is the test set presented for classification. This *batch* convention assumes that each object (regardless of the number of times that it is repeated) is the basic unit of cost. Their finding is that approximately the same number of objects are required to achieve similar accuracy levels. We (Fisher, McKusick, Mooney, Shavlik, & Towell, 1989) have reconciled these conventions and found them basically equivalent. In the incremental approach the cost per presentation (observation) is inexpensive, but many observations are needed. In the batch approach the cost per observation is more expensive, but fewer observations are needed. In either case the empirically-observed *time* until convergence for back-propagation is up to several orders of magnitude greater than ID3, a property that our graphs reflect.

4.3 Noise

The effect of *noise* was also explored. In a manner suggested by Quintan (1986) attribute values were randomly replaced according to a probability (e.g., 25%) that reflected the noise level. Asymptotic accuracy for the congressional and thyroid domains under conditions of 25% noise are graphed in Figure 2. Regardless of noise level (i.e., 25% or 50%) or domain, back-propagation always attained accuracy levels greater than ID3. However, back-propagation again requires significantly more observations to achieve these results (but this is not revealed by the bar graphs). The learning curves' are similar in shape to those of Figure 1: ID3 rapidly peaks and levels off, while back-propagation rises more slowly and uniformly.

4.4 Artificial Domains

Artificial domains were constructed so that comparisons could be made under controlled circumstances. Our experiments systematically vary the degree that attribute values are sufficient and necessary for class membership: [4.4.1] individual values are necessary and sufficient; [4.4.2] values are necessary, but not sufficient; [4.4.3] values are not individually necessary or sufficient (X of N); [4.4.4] the necessity of a value's presence or *absence* is conditioned on the presence of other attributes (exclusive-or), in which no *set* of values are necessary or sufficient.

4.4.1 Sufficiency

Each of four (4) artificial domains contained three (3) classes of objects. Objects were described over 10 attributes. Each class was describable as a conjunction of attribute values that were *unique* to that class. Each such value was singly sufficient to distinguish the class. Each of the four domains varied in the number of singly-sufficient values (i.e., domain 1: 10 of 10 attributes were individually sufficient; domain 2: 6 of 10 were sufficient; domain 3: 4 of 10 were sufficient; domain 4: 1 of 10 were sufficient). Values of those attributes that were not sufficient were generated randomly, and thus were irrelevant to classification.

1D3 uniformly reached perfect performance by 10 objects, regardless of the number of individually-sufficient values. Back-propagation is more sensitive to the number of sufficient values. When all attributes distinguish membership, convergence to perfect prediction required 200 objects, but when no hidden units were used only 10 objects were required. When 4 attributes were sufficient, 300 objects were required for perfect performance (100 objects were required with no hidden units). When only 1 attribute was sufficient (and thus necessary too) 500 observations were required. As the number of sufficient values decreased, back-propagation required more observations to reach asymptotic accuracy.

4.4.2 Necessity

Domains in the second set of experiments added to the complexity of the 'sufficiency' domains. Each class was still describable as a conjunction of attribute values, but the values were not unique to that class: no value was singly sufficient to distinguish class membership.[2] Rather, only the entire conjunctive expression was sufficient (and necessary) to distinguish membership. Once again, the size of the conjunctive expression varied from 1 to 9.

Neither ID3 or back-propagation averaged perfect performance in all domains for the maximum allotted training. Back-propagation remained just below perfect prediction (about 98.5%) for conjunctions of size 4 and 6; ID3 reached perfect accuracy, but required 150 training objects. Back-propagation achieved perfect prediction after 500 objects for the conjunction of size 9; 1D3 averaged 95%) asymptotic accuracy. As the size of the conjunctive target concept grew, it became more difficult for ID3 to spot; each attribute individually transmitted less information.

[2] Except in the case where the conjunction is of size 1; note that this extreme is identical to the lone sufficient (and necessary) condition of 4.4.1.

4.4.3 X of N

In these experiments there was no conjunctive description for any class. Rather, a class was defined by an *X of N* function. More specifically, each domain contained only two classes *(C* and ¬C). *C* is associated with 10 'preferred' values (one for each attribute); an object was a class member iff it contained *at least X* of the 10 values. In contrast to 4.4.2, our 'X of N' experiments disallow any single value *set* to be necessary. The size of X was varied between 1 and 9.

1D3 consistently attained accuracy levels in the vicinity of 85%, to 90% after 200 objects, but about 75%. accuracy was achieved after 10 objects. Back-propagation reached average levels of 99%, to 100% within 2000 observations.

4.4.4 Exclusive-Or

A final set of artificial domains insists that two attributes exhibit the exclusive-or relation (one and only one of two selected values are present). Exclusive-or, like X of N, has no values or value combinations that are necessary for membership. However, 'X of N' allows X *or more* selected values to be present, but exclusive-or requires *exactly* X values to be present. As such, exclusive-or is not linearly separable.

The version of 1D3 that we used (Quinlan, 1986) was not capable of learning this function, averaging between 50% and 75%, accuracy depending on our confidence threshold. However, Section 5 describes a new 1D3 descendenf (Quinlan, 1988) that can undoubtedly reach perfect prediction in this domain. Exclusive or also presents problems to back-propagation: it can not be learned without hidden units. With 10 hidden units, it required approximately 5000 observations to achieve 100% accuracy (approximately 50%, accuracy after 1000 observations.)

5 Discussion

Experiments indicate that back-propagation achieves higher asymptotic accuracy levels under noisy conditions and selected artificial domains, but requires considerably more object presentations. This section identifies several principles of each system that may account for performance differences. To the extent possible, we tie our discussion to distinctions between AI and connectionist paradigms more generally. This discussion qualifies our results as they might apply to paradigm-wide comparisons and encourages the exploration of AI and connectionist hybrid learning systems.[3]

[3] In particular, we avoid the symbolic/subsymbolic distinction. This has proved an unhelpful distinction in that it does not promote short-term progress. This distinction artificially segregates research programs because it is often conveyed as a *prescriptive* (and unknown) boundary that each paradigm *must* observe. Rather, we believe that useful distinctions are *descriptive* and pro-

5.1 Representation and Bias

Differences between ID3 and back-propagation may be attributable to the size and form of the search space explored by each system. The primitive evidence combination function of back-propagation (i.e., X of N) generalizes ID3's primitive logical combinators (conjunction and inclusive disjunction). Finer granularity enables back propagation to converge on logical concepts and others with less hardware, but each primitive must be specialized, which requires greater training. The DNF equivalent of X of N (for arbitrary X) is quite complex. The course primitives of ID3 also suggests that it takes bigger steps in a uniform search space: it approximates the final solution more quickly, but it may accept less than optimal solutions because it 'oversteps' or 'understeps' the optimum.

The subsumtion relation between representation languages suggests that logical descriptions can be harnessed as a sort of 'admissible' heuristic that allow rapid approximation followed by slower refinement. This approach is taken by Utgoff's (1988b) *perceptron trees,* a hybrid of decision trees with linear threshold units as leaves. The decision tree brings about large cuts in the search space, with final convergence left to the leaves. Evaluation of this particular approach must await further experimentation, but nonetheless it represents an important conceptual advance towards the development of hybrid systems.

5.2 Probabilistic versus Logical Classification

A recent trend in machine concept learning is towards *probabilistic* representations (Smith *fa* Medin, 1981). Probabilistic concepts typically classify observations by a. summation of evidence, just as do connectionist network nodes. For example, Fisher's (1987) COBWEB can be viewed as constructing a 'decision' tree of evidence summation units (nonlinear) with a variable threshold (i.e., an object is placed in the node with the highest summation). Arithmetic evidence combination has traditionally distinguished AI and connectionist learning methods, but COBWEB illustrates that this should not be taken as a prescriptive difference.

5.3 Monothetic versus Polythetic Classification

A weakness of ID3 is that it is *monothetic:* learning considers the utility of a single attribute at a time. The predictive merits of attribute value combinations are not explicitly consider, presumably to the detriment of prediction accuracy. This seems evident in the comparisons of 4.4.2 and 4.4.4. In contrast, back-propagation is *polythetic,* in that the values of multiple attributes are simultaneously considered (summed). However, the monothetic property of ID3 is not a general assumption of the field. Traditionally, AI concept learning methods have been search intensive precisely because they simultaneously consider the utility of many attributes. Recently, Quinlan (1988) has introduced a polythetic extension to ID3 that builds a monothetic decision tree (without using confidence measures to terminate decomposition), and then converts it to a set of production rules. Each rule is a polythetic concept that is 'massaged' in order to improve its accuracy. That this extension was prompted by Quinlan's comparison of ID3 and a genetic classifier (which shares certain characteristics with connectionist nets), adds impetus to continued comparisons with this extension. Quinlan's comparisons only occurred in an artificial domain similar to our exclusive-or function. Our methodology promises to characterize the ID3 extension across a wide range of domains. A determination of whether logic-based, but polythetic learning systems can overcome problems of granularity discussed in 5.1 must await experimentation. However, polythetic classification in AI learning systems need not preclude probabilistic representations as noted in 5.2.

5.4 Incremental versus Nonincremental Processing

ID3 assumes that all observations are simultaneously available for processing, while back propagation processes observations as they become available. This distinction is somewhat true of the two paradigms more generally. However, ID3 has recently spawned two incremental variants, 1D4 (Schlirnmer & Fisher, 1986) and IDS (UtgofT, 1988a). Incremental learning more generally is becoming popular in AJ learning research. The common thread in these systems (including connectionist methods and COBWEB) is the *use* of 'probabilistic' representations. Finer granularity allows more conservative steps through a search space. This conservatism is necessary; early in training, many observations are inconsistent with the evolving concept description. No observation should irrevocably impact the incomplete con cept description.

5.5 Constructive versus Convergent Search

Perhaps the most overt distinction between AI and connectionist systems is the manner in which they explore their respective search spaces. AI systems typically reconstruct the space upon demand, which is only defined implicitly by operators and an initial state to begin with. In contrast, most connectionist systems preenumerate a subset of the space, which is implicit in the number and interconnections between nodes. Problems arise if too much (e.g., slow convergence) or too little (e.g., the concept cannot be learned) of the space is preenurnerated. Important steps in reconciling these strategies have been explored by Schlirnmer *fa* Granger (1986) and Hampson *fa* Volper (1987). Schlirnmer *fa* Granger's STAGGER

mote interaction.

system adapts connectionist evidence combination procedures to the 'constructive' (upon demand) approach: the emphasis is on technology transfer to AI systems. In contrast, Hampson *k* Volper stress transfer to connectionist research: specialized disjunctive nodes can be enumerated and integrated on demand, thus facilitating rapid convergence. Picking up on the discussion of 5.1, AI systems appear well suited to enumerating an appropriate subspace that may then be refined.

6 Concluding Remarks

Empirical comparisons have uncovered advantages and disadvantages of two specific learning systems. That these systems are from different paradigms is impetus for interaction. Research of this ilk is being pursued by several researchers, some of whom we have discussed. In addition, future comparisons must increase the scope of our study to other learning systems and learning strategies. Currently, we (Fisher, McKusick, Mooney, Shavlik, *k* Towell, 1989) are exploring an alternative training strategy for back-propagation that combines the incremental and batch approaches: small batches of training objects are incrementally presented and are processed until convergence (on the subbatch). Our initial results suggest that using very small batch sizes (1 to 4) significantly reduces the total number of presentations required until asymptotic accuracy is achieved. We are pursuing these experiments and hope to flesh out an explanation of the phenomena in terms AI concepts; most notably, can we view this process as simulating case-based reasoning and/or a *specific to general* search for the best classifier? Either interpretation is a departure from the usual training strategy of moving from general (indiscriminate) towards greater specialization. Hopefully, these explorations will improve back-propagation training time without detrimentally impacting accuracy.

Acknowledgements

We thank Ray Mooney, Jude Shavlik, and Geoff Towell for influential discussions and debates. Comments by Richard Sutton improved the correctness and clarity of discussion, as did comments by IJCAI reviewers.

Bibliography

Fisher, D. (1987). Knowledge Acquisition Via Incremental Conceptual Clustering. *Machine Learning, 2,* 139-172.

Fisher, D., McKusick, K., Mooney, R., Shavlik, J. & Towell, G. (1989). Processing Issues in Comparisons of Symbolic and Connectionist Learning Systems. *Proceedings of the Sixth International Machine Learning Workshop.* Ithaca, NY: Morgan Kaufmann.

Hinton, G. (in press). Connectionist Learning Procedures. *Artificial Intelligence.*

Hampson, S. *k* Volper, D. (1987). Disjunctive models of Boolean Category Learning, *Biological Cybernetics, 56,* 121-137.

McClelland, J. *k* Rumelhart, D. (1988). *Explorations in Parallel Distributed Processing,* MIT Press, Cambridge, MA.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning, 1.*

Quinlan, J. R. (1988). *Proceedings of the Fifth International Machine Learning Conference,* (135-141), Ann Arbor, MI: Morgan Kaufmann.

Rumelhart, D., Hinton, G., & Williams, J. (1986). Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing, Vol. 1* (D. Rumelhart *k* J. McClelland, eds.). MIT Press.

Schlimrner, J. & Fisher, D. (1986). A Case Study of Incremental Concept Learning. *Proceedings of the Fifth National Conference on Artificial Intelligence.* Philadelphia, PA: Morgan Kaufmann.

Schlimrner, J. *k* Granger, R. (1986). Incremental Learning from Noisy Data. *Machine Learning, I,* 317 334.

Sejnowski, T. *k* Rosenberg, C. (1987). Parallel Networks that Learn to Pronounce English Text. *Complex Systems, L*

Shavlik, J., Mooney, R., *k* Towell, G. (1989). Symbolic and Neural Net Learning Algorithms: An Experimental Comparison. Technical Report, University of Wisconsin, Madison.

Smith, E. & Medin, D. (1981). *Categories and Concepts,* Cambridge, MA: Harvard University Press.

Stepp, R. (1984). *Conjunctive Conceptual Clustering: A Methodology and Experimentation.* Doctoral dissertation. University of Illinois, Urban a-Champaign, IL.

Utgoff, P. (1988a). ID5: An Incremental 1D3. *Proceedings of the Fifth International Machine Learning Conference,* (107 120), Ann Arbor, MI: Morgan-Kaufman n.

Utgoff, P. (1988b). Percetron Trees: A Case Study in Hybrid Concept Representations. *Proceedings of the Seventh National Conference on Artificial Intelligence,* St. Paul, MN: Morgan Kaufmann.