# PROCESSING ISSUES IN COMPARISONS OF SYMBOLIC AND CONNECTIONIST LEARNING SYSTEMS[1]

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

**Raymond Mooney**
Department of Computer Sciences
Taylor Hall
University of Texas
Austin, TX 78712

**Jude W. Shavlik**
**Geoffrey Towell**
Computer Sciences Department
University of Wisconsin
1210 West Dayton Street
Madison, WI 53706

## ABSTRACT

Symbolic and connectionist learning strategies are receiving much attention. Comparative studies should qualify the advantages of systems from each paradigm. However, these systems make differing assumptions along several dimensions, thus complicating the design of 'fair' experimental comparisons. This paper describes our comparative studies of ID3 and back-propagation and suggests experimental dimensions that may be useful in cross-paradigm experimental design.

## Introduction

Symbolic and connectionist (or neural-net) learning methods often address similar tasks, but they may differ considerably in their processing assumptions. Thus, there is impetus for investigating the relative advantages and limitations of systems of each paradigm. A task that has commonly been explored in each paradigm is *learning from examples* (or tutored learning): from a set of classified observations, a learning system abstracts a 'rule' or mapping that facilitates classification of new observations. This paper reviews experimental comparisons of a symbolic learning system, ID3, and a connectionist learning system, back-propagation in a feed-forward net. However, independent investigations (Fisher & McKusick, 1989; Mooney, Shavlik, Towell, & Gove, 1989; Shavlik, Mooney, & Towell, 1989; Weiss & Kapouleas, 1989) differ considerably in the strategies used to train back-propagation. We will not tightly couple our results, but suggest a framework for evaluating our respective strategies and for guiding future experimental comparisons.

Following Schlimmer and Fisher (1986), we characterize ID3 and back-propagation in terms of three primary dimensions: the time complexity of processing a single observation or *cost per observation*, the *number of observations* required to attain specified prediction accuracy levels, and inversely the *accuracy* of learned knowledge with respect to novel (i.e., not used for training) *test* observations. We may also characterize each system by the *total cost* (i.e., *number of observations* × *cost per observation*) required to achieve specified (e.g., asymptotic) accuracy levels. Collectively, these measures enable strong statements about relative system accuracy and efficiency. Our analysis is motivated by tentative findings that back-propagation achieves slightly higher accuracy levels than ID3, but requires much more time to do so.

## ID3 and Back-Propagation

ID3 and back-propagation were chosen for initial study because of their widespread use and well-known successes. ID3 (Quinlan, 1986) is a simple, but effective symbolic method for learning from examples. The system constructs a *decision tree* from a set of training objects. At each node the training objects are partitioned by their value along a single, most-informative attribute. The training set is recursively

[1] This research was partially supported by grants from the Vanderbilt University Research Council, University of Wisconsin-Madison Graduate School, and the University of Texas at Austin's Department of Computer Sciences and Artificial Intelligence Laboratory.

decomposed in this manner until no remaining attribute improves prediction in a statistically-significant manner. We assume *nominal* attributes which have a finite set of values (e.g., Color $\in$ {red, blue, green}).

Back-propagation (Rumelhart, Hinton, & Williams, 1986) assumes that input nodes of a network record observed features from the environment and pass 'activation' forward through a single intermediate layer of 'hidden' nodes to an output layer. The total activation of a hidden or output node is a weighted sum of its inputs. We encode nominal attributes using a set of units, one dedicated to each value (Sejnowski & Rosenberg, 1987). 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. Each output node corresponds to one class; the object is classified by the class whose output node has the highest activation. Back-propagation adjusts network weights so as to improve the match between actual and ideal output. In principle, with 'sufficient' hidden units, backpropagation learning can converge on correct classification for any domain (assuming classes contain unique objects), but it is always possible that local minima will lead learning astray.

## Training conventions

ID3 is a *nonincremental* learning system: all training observations must be present from the outset of system execution. This 'batch' approach to training ID3 was consistently maintained across our experimental studies. After training, a disjoint test set of observations was presented for classification. Classification accuracy afforded by the learned decision tree was recorded. Our experiments (Fisher & McKusick, 1989; Shavlik, Mooney & Towell, 1989) also varied training set size. However, whenever a larger (more encompassing) training set was investigated, decision trees of prior experiments were not exploited. Rather, a new decision tree was generated from 'scratch'.

While ID3 training was relatively standard across our individual studies, there are three conventions collectively found in our ongoing work with back-propagation. Mooney, Shavlik, Towell, and Gove (1989) used a *batch* strategy for training back-propagation. In particular, a training set was repeatedly cycled through the net until a threshold-level (i.e., 99.5%) of the set was correctly classified. After (near)convergence on the training set, the net was used to classify a disjoint test set. Shavlik, Mooney, and Towell (1989) tested this batch strategy for varying training set sizes. As with ID3, the net weights from prior training were not used (i.e., the net was reinitialized) for each more inclusive training set.

In contrast, an *incremental* strategy was explored by Fisher and McKusick (1989). Training observations were drawn randomly (with replacement) and presented once to the network for each draw. Accuracy tests were conducted at intermittent points in training. Network weights were not reinitialized between tests; new training observations incrementally modified weights that were previously accumulated.

## Behavioral Characterizations

### Cost per Instance

This section develops a very rough measure of the computational cost of processing an individual observation for ID3 and the two back-propagation training strategies. To divide a decision tree node ID3 requires that the informativeness of each attribute be determined. The cost of the step is proportional to $(a \times i) + (a \times |V_{ave}|)$, where $a$ is the number of attributes not used to previously divide the tree, $|V_{ave}|$ is the average number of values per attribute, and $i$ is the number of instances to be partitioned at the node; each instance must be examined for its (single) value along each attribute, after which counts of each attribute value must be examined to determine attribute informativeness. Typically, $|V_{ave}|$ is much smaller than $I$, so we consider only $a \times |I|$. The final decision tree can be at most $|A|$ levels deep, where $|A|$ is the total number of attributes. Moreover, at each tree level at most $|I|$ instances (collectively taken across all nodes at that level) can be examined to determine attribute informativeness. Total tree building cost is proportional to

$$\sum_{a=0}^{|A|} a|I| = |I|\frac{|A|(|A|+1)}{2}.$$

The asymptotic cost of each instance is proportional to

$$\text{Cost per observation (ID3): } |A|^2.$$

Because tree depth is also bounded by $|I|$ and total tree depth rarely reaches $|A|$, our experience suggests that average cost is linear in the number of attributes. In addition, there are a fixed number of multiplications per step that will contribute a constant factor.

An analysis of back-propagation is complicated by our various training strategies. In particular, an observation may be presented many times. Thus, we must distinguish between cost per *presentation* and cost per *observation*. Given our encoding, a net contains $|A||V_{ave}|$ input lines. In addition, we have explored various numbers of hidden units. We assume that a reasonable number of hidden units (with no *a priori* knowledge of the domain) is some function of the cardinality of the inputs. Shavlik, Mooney, and Towell (1989) empirically found that one-tenth the number of input units proved an adequate number of hidden units.[2] This is roughly consistent with other empirical studies (e.g., Sejnowski & Rosenberg, 1987). Given a fully-interconnected network this convention yields a cost per presentation proportional to the number of interconnections (weights): $\frac{|A|^2|V_{ave}|^2}{10}$. This is also the cost per observation using incremental training.

$$\text{Cost per observation (incremental backprop): } \tfrac{1}{10}(|A||V_{ave}|)^2.$$

Assuming a constant upperbound on the number of presentations per observation, this figure is also proportional to the cost per observation for the batch method. However, this upperbound may be very high ($10^2$ to $10^3$ presentations), thus significantly increasing the cost per observation.

$$\text{Cost per observation (batch backprop): } 10(|A||V_{ave}|)^2 \text{ to } 10^2(|A||V_{ave}|)^2.$$

In summary, the cost complexity per *observation* of ID3 and back-propagation's cost per *presentation* appears the same, but in practice the considerably smaller constant term of back-propagation makes it relatively cheaper. However, the constant factor contributed by repeated presentation results in greater costs per observation of batch versions of back-propagation than for ID3.

## Asymptotic Accuracy

Our independent studies have compared ID3 and back-propagation in a wide variety of natural and artificial domains. Fisher and McKusick (1989) report slight differences between ID3 and back-propagation (using the incremental training strategy) in asymptotic accuracy. However, back-propagation tends to equal or better ID3 across all those domains tested. Similarly, Mooney, Shavlik, Towell, and Gove (1989) and Shavlik, Mooney, & Towell (1989) report that back-propagation tends to equal or (slightly) better ID3 in terms of asymptotic accuracy. Regardless of training strategy, our asymptotic accuracy results suggest that back-propagation consistently equals or slightly betters ID3. Our data suggests that training strategy (incremental and batch) does not significantly impact back-propagation's asymptotic accuracy.

## Training Amount

ID3 is trained in a strict batch manner; of interest is the number of training objects (i.e., $|I|$) that must be batched in order to reasonably assure asymptotic or close-to-asymptotic accuracy. In contrast, the number of training observations required by back-propagation depends on training strategy. Fisher and McKusick (1989) report that incremental back-propagation requires one to three orders of magnitude more training

[2] Actually, the number of hidden units was 1/10 the number of input and output units. However, the number of output lines is typically less than the number of inputs; consideration of outputs will not add asymptotically to cost, but it will increase the constant term.

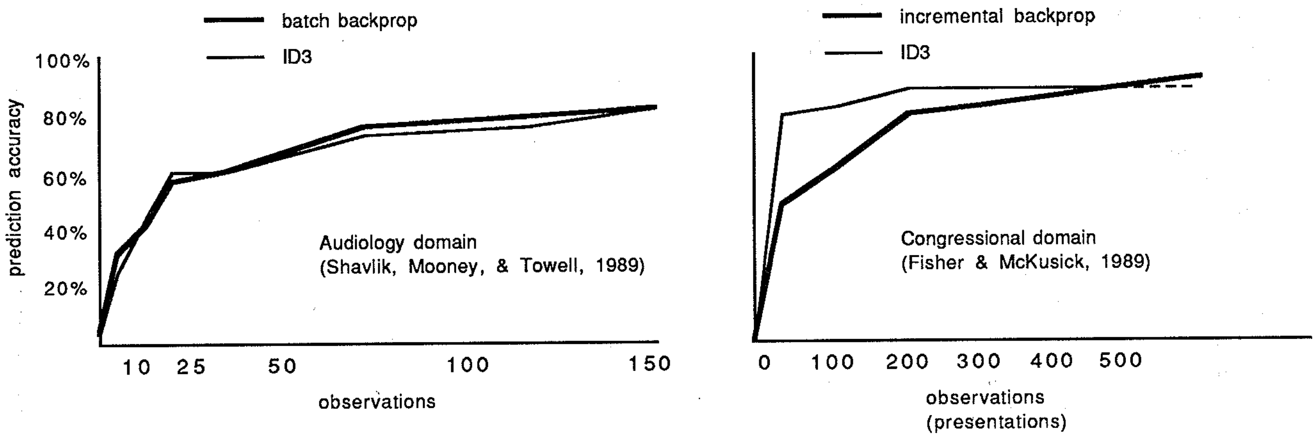

Figure 1: Accuracy as a function of training amount.

objects than ID3 (i.e., $10^3|I|$) to achieve a level of performance that equals ID3, although it may eventually surpass ID3 with more training (but not by another order of magnitude of training). In contrast, Shavlik, Mooney, & Towell (1989) show that batch-trained back-propagation and ID3 require approximately the same number of observations to achieve similar performance.

$$\text{Training observations (ID3): } |I|.$$

$$\text{Training observations (incremental backprop): } 10^2|I| \text{ to } 10^3|I|.$$

$$\text{Training observations (batch backprop): } |I|.$$

Finally, Hinton (in press) suggests that the total number of *presentations* to achieve asymptotic performance is $O(N^2)$, where $N$ is the total number of weights or roughly $|A|^2|V_{ave}|^2$. Based on our experience this is a reasonable upperbound that depends on domain characteristics (i.e., $|A|$).

## Summary of Primary Dimensions

Graphs (a) and (b) of Figure 1 illustrate the variance between ID3, incremental back-propagation, and batch back-propagation. Back-propagation tends to achieve slightly higher accuracy levels, but at a cost of more observations (incremental) or more presentations per observation (batch).

## Total Work

We are now in a position to make initial guesses as to total effort required to converge on asymptotic (or any intermediate level) of performance. In each case this is simply the product of the cost per observation and total number of observations. The following use ID3 as a baseline (assuming $|I|$ observations) and approximate the total work required by back-propagation to achieve similar accuracy levels.

$$\text{Total work (ID3): } |I||A|^2.$$

$$\text{Total work (incremental backprop): } 10|I|(|A||V_{ave}|)^2 \text{ to } 10^2|I|(|A||V_{ave}|)^2.$$

$$\text{Training work (batch backprop): } 10|I|(|A||V_{ave}|)^2 \text{ to } 10^2|I|(|A||V_{ave}|)^2.$$

Back-propagation takes considerably longer to achieve similar results. The required number of presentations more than offsets the relatively cheap cost per presentation. Second, while we have not conducted head-to-head comparisons of cpu-time expended by the two versions of back-propagation, our analysis suggests an equal amount of total effort is required in the two cases. The relative advantage of one approach over another lies in the constraints of a particular task environment. For example, in an environment requiring rapid response (e.g., real-time control) and assuming an organism of limited memory, the incremental alternative may be the best or only alternative. In other environments, the number of instances may be a critical constraint, but processing per observation is not a limiting factor. In these environments as much information as possible should be extracted from each observation; a batch approach is most suitable.

## Concluding Remarks

We have framed our comparisons of ID3 and back-propagation along three primary dimensions (i.e., cost per observation, training amount, and accuracy) and one composite dimension (i.e., total work). Our analysis reveals that incremental and batch back-propagation reach slightly higher accuracy levels than ID3, but require considerably more work than ID3 to do so. In terms of total work there appears to be no advantage to one version of back-propagation over the other; preference must depend on other factors.

We plan to use our dimensions to frame ongoing and future experiments. Our ongoing work focuses on a hybridization of the incremental and batch back-propagation training strategies. This hybrid assumes that 'subbatches' are processed until near convergence (on the subbatch). However, processing is incremental between subbatches: a subbatch is processed with respect to the evolving network (vice a reinitialized network). Our initial results suggest that there is an 'optimal' subbatch size with respect to total work. For example, in the congressional domain a subbatch size of 3 yields 97% accuracy after 453 total *presentations.* Total presentations increase with larger subbatch sizes (e.g., 1700 presentations for a subbatch size of 100), but with no improvement in accuracy. Smaller subbatch sizes (e.g., 1) yield slightly lower accuracy and slightly fewer presentations. In other domains this effect is even more pronounced; there is an intermediate size that appears to optimize a tradeoff of total work and asymptotic accuracy. While our analysis reveals that our individual training conventions are equivalent along a *total work* dimension, a hybrid methodology offers some promise that substantive reductions in work can be realized without detrimentally impacting accuracy. In addition, this methodology does not require a full memory of past observations; apparently, very few instances need be remembered at any one time. We are continuing our investigations into the effects of subbatch size.

### References

D. Fisher & K. McKusick. An Empirical Comparison of ID3 and Back-propagation. IJCAI Proc. Detroit (1989). Morgan-Kaufmann.

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

R. Mooney, J. Shavlik, G. Towell, & A. Gove. An Experimental Comparison of Symbolic and Connectionist Learning Algorithms. IJCAI Proc. Detroit (1989). Morgan-Kaufmann.

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

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

J. Schlimmer & D. Fisher. A Case Study of Incremental Concept Learning. AAAI Proc. Philadelphia (1986). Morgan Kaufmann.

T. Sejnowski & C. Rosenberg. Parallel Networks that Learn to Pronounce English Text. *Complex Systems, 1* (1987).

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

S. Weiss & I. Kapouleas. An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods. IJCAI Proc. Detroit (1989). Morgan Kaufmann.