Empirical Analysis of the General Utility Problem in Machine Learning

Lawrence B. Holder
Department of Computer Science Engineering
University of Texas at Arlington
Box 19015, Arlington, TX 76019-0015
holder@cse.uta.edu

Abstract

The overfit problem in inductive learning and the utility problem in speedup learning both describe a common behavior of machine learning methods: the eventual degradation of performance due to increasing amounts of learned knowledge. Plotting the performance of the changing knowledge during execution of a learning method (the performance response) reveals similar curves for several methods. The performance response generally indicates an increase to a single peak followed by a more gradual decrease in performance. The similarity in performance responses suggests a model relating performance to the amount of learned knowledge. This paper provides empirical evidence for the existence of a general model by plotting the performance responses of several learning programs. Formal models of the performance response are also discussed. These models can be used to control the amount of learning and avoid degradation of performance.

Introduction

As machine learning methods acquire increasing amounts of knowledge based on imperfect (e.g., sparse, noisy, low probability) instances, the amount of low-utility knowledge increases, and performance degrades. The *general utility problem* in machine learning refers to the degradation of performance due to increasing amounts of learned knowledge [Holder, 1990]. This term derives from the *utility problem* used by Minton [1988] to describe this phenomenon in speedup learning, but generalizes to other machine learning paradigms. Other researchers have observed the ubiquity of the utility problem in machine learning paradigms and compare the utility problem in speedup learning to the problems of noise and overfit in inductive learning [Yoo and Fisher, 1991]. This work suggests that individual methods for avoiding the general utility problem may derive from a general model of the relationship between learned knowledge and performance that applies to several learning paradigms.

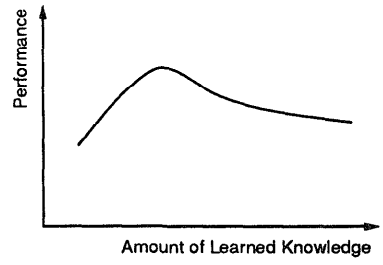

Figure 1: Performance response indicative of the general utility problem.

Identifying this model would provide a general mechanism for preventing performance degradation due to the general utility problem. The analysis in this paper reveals some important properties of such a model.

A useful tool for analyzing the general utility problem in machine learning is the performance response: the performance of the learned knowledge measured during the course of learning (see Figure 1). The units along the horizontal axis represent a simple transformation in the learner's hypothesis. For example, the transformation performed by a splitting method is a single split. Since a knowledge transformation may not always increase the amount of learned knowledge in terms of the size of the knowledge, an increase along this axis represents a refinement of existing knowledge. The vertical axis of the performance response measures the performance of the learned knowledge after each transformation. The classification accuracy of inductive learners and the problem-solving time of speedup learners are the focus of this work.

Figure 1 illustrates the typical performance response of a learning method that suffers from the general utility problem: an initial performance increase to a single peak followed by a more gradual decrease. The next two sections reveal this common trend in the performance responses of inductive and speedup learners. A model of this trend can be used to avoid the performance degradation by controlling the amount of learned knowledge to coincide with the peak of the performance response. Holder [1991a] describes the

MBAC (Model-Based Adaptive Control) system that uses an empirical model of the performance response to control learning. MBAC adapts a parabolic model of the performance response peak by sampling the actual performance response of the learner. Although the parabolic model forces MBAC to adhere to the trend in Figure 1, several samples are necessary to insure identification of the true peak. A separate model is maintained for each learning method/performance dimension pair. MBAC uses the models to select an appropriate learning method according to the model's predicted peak performance. MBAC then invokes the learner, performing the number of knowledge transformations necessary to reach the peak of the performance response model. Experimentation with MBAC shows that the parabolic model is capable of choosing an appropriate learning method and controlling that method [Holder, 1991a]. However, a more formal model of the performance response is necessary to improve the MBAC approach. The section following the empirical results discusses some preliminary formal models.

Inductive Learning

The general utility problem in inductive learning relates to the overfit problem. Overfit occurs when the learning method identifies errant patterns in the training data. Errant patterns may arise due to noise in the training data or inadequate stopping criteria of the method. As demonstrated below, the overfit behavior of splitting, set-covering and neural network learning methods follow the general utility problem trend in Figure 1.

Splitting Methods

Splitting methods recursively split the set of training data by choosing an appropriate feature or feature-value pair. The knowledge produced by a splitting method can be represented as a decision tree. The learned knowledge changes every time the method makes a split; therefore, one choice for the x-axis of the performance response is the number of splits. The y-axis (performance) measures the classification accuracy of the knowledge after each split, as measured using a separate set of test data.

Figure 2 illustrates three performance responses obtained from the ID3 inductive learner [Quinlan, 1986] on the DNF2 domain [Pagallo and Haussler, 1990]. Each performance response in Figure 2 represents a different decision tree node expansion order. Each performance response is an average over ten trials. Each trial consists of selecting random training and testing sets, generating the decision tree using the training set, and measuring accuracy after each split using the testing set. As Figure 2 reveals, the order of the knowledge transformations is important for perceiving the desired performance response trend in Figure 1. The effects of overfit increase as the decision tree becomes deeper; therefore, a breadth-first traversal of the tree defers

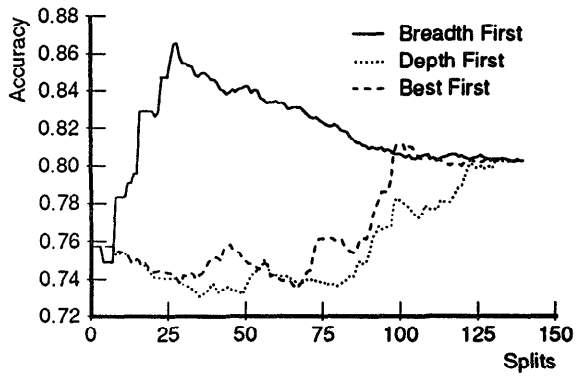

Figure 2: Performance responses of ID3 on the DNF2 domain for three different orders of decision tree expansion.

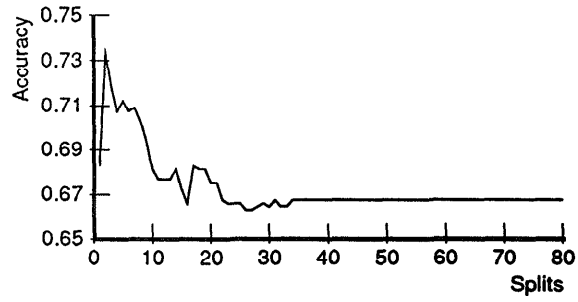

Figure 3: Performance response of ID3 on the Flag domain.

overfit to later splits. A general-to-specific ordering along the amount of learned knowledge axis is necessary for perceiving the performance response trend in most learning methods suffering from the general utility problem. Figure 3 shows the performance response of ID3 on the Flag[1] domain using the breadth-first splitting order. Both figures illustrate a performance response that follows the trend of Figure 1.

The chi-square pre-pruning [Quinlan, 1986] and reduced-error post-pruning [Quinlan, 1987] techniques help to alleviate overfit, but on average the accuracy of the resulting tree is still less than the peak accuracy of the performance response. Similar results were obtained with the PLS1 splitting method [Rendell, 1983], which uses an increase in the t_α parameter to increase pruning. Table 1 shows that these pruning techniques do not completely alleviate the overfit problem.

Set-Covering Methods

A set-covering method for inductive learning constructs a hypothesis which describes a subset of the training instances, and then applies the same method

[1]The Flag domain is available from the UC Irvine machine learning databases.

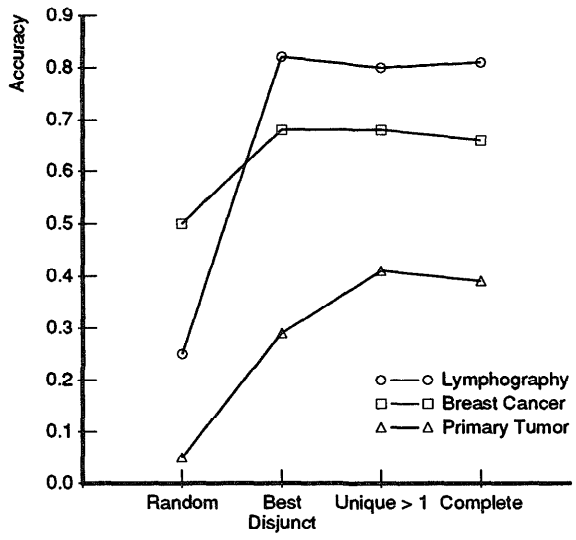

Figure 4: Performance response of AQ for three medical domains.

on the remaining training instances. Since set-covering methods typically learn disjunctive normal form (DNF) expressions for the hypotheses, the dimension used to measure the amount of learned knowledge is the number of disjuncts in the induced hypothesis.

During experimentation with the AQ set-covering method, [Michalski, 1989] found that repetitive application of AQ can yield less accurate hypotheses than a more conservative application strategy combined with a more flexible inference mechanism than exact matching. Michalski compared the accuracy of the *complete* DNF hypothesis produced by AQ to truncated versions of the same hypothesis. The first truncated version of the hypothesis consists of the single disjunct covering the most examples (*best disjunct*). The second truncated version of the hypothesis consists of only those disjuncts covering more than one unique example (*unique > 1*). The truncated hypotheses use a simple matching procedure for classifying uncovered and multiply-covered examples.

Although based on only four points, Figure 4 approximates the performance response of AQ in three medical domains (Lymphography, Breast Cancer and Primary Tumor) averaged over four trials.[2] Figure 4 demonstrates that AQ also suffers from the general utility problem with increasing numbers of disjuncts, and the response curves indicate the same trend as in Figure 1. Holte *et al.* [1989] alludes to similar behavior in the CN2 set-covering method.

[2] Data from individual trials was not available for significance testing.

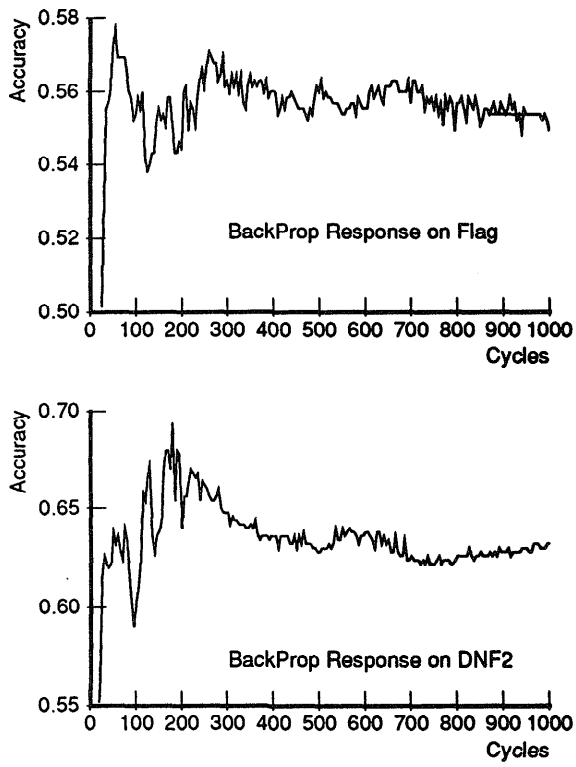

Figure 5: BackProp performance response.

Neural Network Methods

The multilayer perceptron using error back- propagation [Rumelhart *et al.*, 1986] updates the weights of the network according to errors in classifying the training instances. Each pass through the set of training instances is called a *cycle*. As the number of cycles increases, the network more accurately classifies the training instances. However, overfit eventually occurs as the network learns the training instances too precisely, degrading accuracy on the testing data. To analyze the overfit of the back-propagation neural network, the performance response measures accuracy of the network after every five cycles.

Figure 5 shows the performance response of the error back-propagation neural network (BackProp) on the Flag and DNF2 domains. The networks contained on hidden layer with four units. The BackProp response on the Flag and DNF2 domains follows the general utility problem trend as in Figure 1. Table 1 reveals that on average the network at the initial peak performs better than the final network. Geman *et al.* [1992] found similar behavior in the domain of handwritten number recognition.

Speedup Learning

Although the utility problem has been verified in several speedup learning systems [Minton, 1988; Tambe

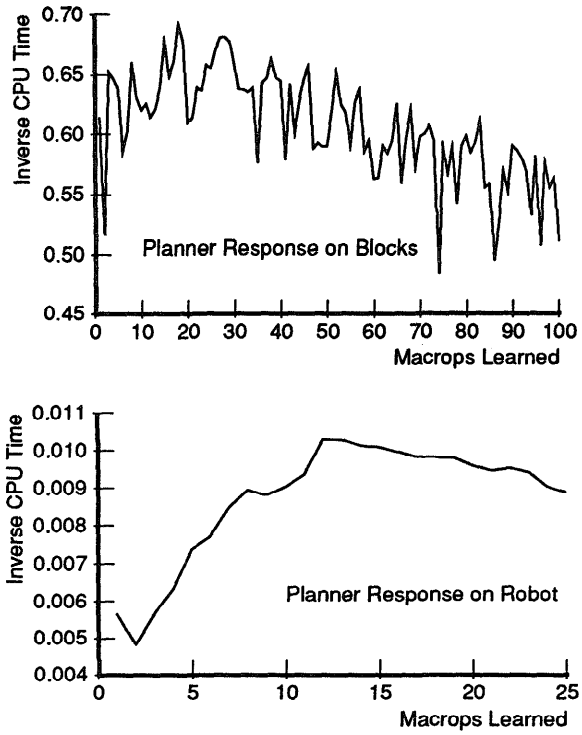

Figure 6: Planner performance response.

and Newell, 1988; Mooney, 1989; Markovitch and Scott, 1989], the experiments typically do not show the performance response of the system.[3] Figure 6 plots the performance response of a macro-operator learner consisting of a forward-chaining planner and a STRIPS-like plan generalizer [Fikes *et al.*, 1972]. Two domains are used in the experimentation. The blocks domain consists of four operators for stacking and unstacking blocks. The robot domain consists of eight operators allowing the robot to move boxes within a layout of connected rooms.

The experiments proceed by solving a training problem in the domain, generalizing the resulting plan, adding the generalized plan to the set of available operators, and then measuring the amount of CPU time needed to solve a separate set of test problems using the augmented set of operators. The x-axis of the performance response is the number of learned macrops. The y-axis measures the inverse CPU time needed to solve the set of test problems. Although erratic in the blocks domain, the performance response plots in Figure 6 follow the trend of the general utility problem.

[3]Cohen [1990] plots control-rule learning response curves for several planning domains.

Table 1: Percentage final performance of peak for inductive learners.

Method	Domain				
	BC	Flag	Flare	Vote	DNF2
ID3	91.2	88.2	95.0	97.6	93.6
ID3 Chi 99.0	89.0	88.5	94.4	98.1	94.4
ID3 Chi 99.9	90.8	89.9	96.1	97.0	97.2
ID3 Red-Err	98.6	95.4	98.7	99.7	100.3
PLS1 $t_\alpha = 1.5$	92.4	97.6	98.5	98.9	92.8
PLS1 $t_\alpha = 2.0$	94.6	98.4	98.5	99.3	95.6
BP4	82.8	89.8	88.2	92.6	91.1

Table 2: Percentage final performance of peak for Planner.

Method	Domain	
	Blocks	Robot
Planner	67.4	76.1

Trends

Previous sections verify the existence of the general utility problem in several learning methods. The performance responses of these methods follow the trend illustrated in Figure 1. Adopting a model of this trend permits the control of the general utility problem by constraining the amount of learned knowledge to reside at the point corresponding to the peak performance.

Tables 1 and 2 quantify the possible performance gains by using this model-based control of the amount of learned knowledge. Each entry in the tables is the percentage final performance of peak performance ($\frac{\text{final}}{\text{peak}} * 100$) averaged over ten performance response curves. Table 1 lists entries for several of the previously described inductive learning methods[4] on five different domains[5]. Table 2 lists entries for the Planner speedup learner on two domains. Note that the entries in Table 2 can be arbitrarily deflated by allowing the speedup learner to acquire more macrops.

As shown in Tables 1 and 2, the final performance is less than the peak performance for all but one case. A majority of the values are statistically significant, and in the cases where the significance is low, the peak of the performance response is no worse than the final performance. Thus, the ability to constrain the amount of learned knowledge to the point corresponding to peak performance will improve the performance of the learner. Although individual methods exist for alleviating the general utility problem in each particular learning method, the performance response model offers a general method for avoiding the general utility problem in many machine learning methods.

[4]BP4 is error back-propagation with one hidden layer containing four units.

[5]The Breast Cancer (BC), Flare and Voting (Vote) domains are from the UC Irvine machine learning databases.

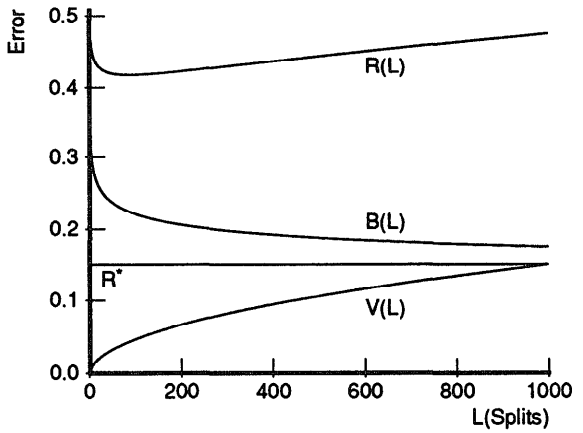

Figure 7: Performance response derived by Breiman et al. [1984] for a decision tree induction method.

Formal Models

Breiman *et al.* [1984] derive a formal model of the performance response for splitting methods. The shape of the performance response is the result of a tradeoff between bias and variance. *Bias* expresses the degree of fit of the decision tree to the training data. A low bias (many small hyper-rectangles) is preferred to a high bias (few large hyper-rectangles), because low bias allows a more precise fit to the data. However, a low bias increases the likelihood that hyper-rectangles produce classification error (*variance*) due to fewer points on which to base the classification.

The analysis expresses the bias and variance in terms of the number of leaves L in the decision tree. Assuming binary splits at each node of the tree, the number of splits is $L - 1$. Therefore, the behavior of the bias and variance as the number of splits increase will be similar to the behavior as L increases. The expression for the classification error $R(L)$ in terms of the bias $B(L)$ and the variance $V(L)$ is

$$R(L) = B(L) + V(L) + R^* \qquad (1)$$

where R^* is the Bayes optimal classification error. Breiman *et al.* derive the following constraints on the bias $B(L)$ and the variance $V(L)$:

$$B(L) \leq \frac{C}{L^{2/M}}, \quad V(L) \leq \sqrt{\frac{L}{N}}, \quad V(L \simeq N) \leq R^*$$

where C is a constant, M is the dimension of the instance space (i.e., number of features used to describe the training instances), and N is the number of training instances. Equation 1 is an expression of the classification error response curve. Figure 7 plots the bias $B(L)$, variance $V(L)$, Bayes error R^*, and estimated classification error $R(L)$ from Equation 1, where $C = 0.35$, $M = 20$, $N = 1000$ and $R^* = 0.15$. The plot extends from $L = 0$ to 1000. Subtracting this error curve from one would yield the accuracy response curve. The similarity of this performance response to that of Figure 1 supports the existence of a single peak and the inevitability of overfit in splitting algorithms without appropriate stopping criteria or post-pruning techniques. Maximizing performance while avoiding overfit requires the determination of the number of splits L corresponding to the minimum of the error response curve.

A similar analysis applies to set-covering methods, where the number of disjuncts in the DNF hypothesis replaces the number of splits in the decision tree. The corresponding expressions for bias and variance as a function of the number of disjuncts have a similar behavior as those depending on the number of splits, and the set-covering response curve follows the behavior in Figure 7. Geman *et al.* [1992] describe a similar model for two-layer networks in terms of the number of hidden units, and nearest neighbor methods in terms of the number of neighbors. Holder [1991a] describes a possible relationship between the number of cycles and the number of hidden layers in a multilayer network.

These models assume that an increase in the amount of learned knowledge corresponds to an increase in the complexity of the resulting hypothesis. One definition of complexity is the degree of the function represented by the hypothesis. Given that the candidate hypotheses have a sufficient degree of complexity to allow overfit, ordering the amount of learned knowledge in terms of increasing complexity insures the presence of the general utility problem trend.

A speedup learner is similar to an inductive learner in that both seek a concept that maximizes performance. The concept sought by a speedup learner is a set of macro-operators or control rules minimizing the time taken by the problem solver to solve problems from some domain. If the set of problems used to train the learner is not representative of the distribution of problems in the domain, then the performance obtained for the training examples may degrade performance on the testing examples for reasons similar to overfit in inductive learners. However, the factors underlying the performance degradation are different from those affecting inductive learners. Minton [1990] identifies three ways in which macro-learning affects the problem-solving performance of a speedup learner. A simple quantification of these three components in terms of branching factors behaves similarly to the general utility problem trend [Holder, 1991a], but the exact relationship between macro-operator (or control rule) learning and performance is not fully understood.

Conclusions

Both inductive and speedup learning methods suffer from the general utility problem: the eventual degradation of performance due to increasing amounts of learned knowledge. The performance response curves of these methods indicate a common trend depicted in

Figure 1. A model of this trend can be used to control the amount of learned knowledge to achieve peak performance, which is typically greater than the final performance of the learning method (see Table 1). The model could also predict the achievable performance of the learning method as a means of selecting an appropriate method for a learning task [Holder, 1991b].

The MBAC approach uses and empirical model of the performance response. However, an empirical model requires samples of the actual performance response (runs of the learning method) and suffers from inaccuracies due to discrepancies between the empirical and true model of the performance response. Therefore, the MBAC approach (or any approach to controlling and estimating the performance of a learning method) would benefit from a formal model of the performance response that depends on properties of the current learning task, such as number of instances and dimension of the instance space. The formal models discussed earlier represent preliminary progress towards this goal. The underlying forces of bias and variance and the constraints on the order of knowledge transformations serve to unify several inductive and, at a high level, speedup learning methods. Continued refinement of models of the general utility problem will provide a general framework for controlling and comparing different learning paradigms.

Acknowledgements

I would like to thank Larry Rendell, members of the Inductive Learning Group at University of Illinois, and the reviewers for their helpful suggestions. Thanks also to Ray Mooney, Jude Shavlik and Carl Kadie for their implementations of the learning methods. This research was partially funded by the National Science Foundation under grant IRI 8822031.

References

Breiman, L.; Friedman, J. H.; Olshen, R. A.; and Stone, C. J. 1984. *Classification and Regression Trees.* Wadsworth.

Cohen, W. W. 1990. Learning approximate control rules of high utility. In *Proceedings of the Seventh International Conference on Machine Learning.* 268–276.

Fikes, R. E.; Hart, P. E.; and Nilsson, N. J. 1972. Learning and executing generalized robot plans. *Artificial Intelligence* 4(3):189–208.

Geman, S.; Bienenstock, E.; and Doursat, R. 1992. Neural networks and the bias/variance dilemma. *Neural Computation* 4(1):1–58.

Holder, L. B. 1990. The general utility problem in machine learning. In *Proceedings of the Seventh International Conference on Machine Learning.* 402–410.

Holder, L. B. 1991a. *Maintaining the Utility of Learned Knowledge Using Model-Based Adaptive Control.* Ph.D. Dissertation, Department of Computer Science, University of Illinois at Urbana-Champaign.

Holder, L. B. 1991b. Selection of learning methods using an adaptive model of knowledge utility. In *Proceedings of the First International Workshop on Multistrategy Learning.* 247–254.

Holte, R. C.; Acker, L. E.; and Porter, B. W. 1989. Concept learning and the problem of small disjuncts. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence.* 813–818.

Markovitch, S. and Scott, P. D. 1989. Utilization filtering: A method for reducing the inherent harmfulness of deductively learned knowledge. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence.* 738–743.

Michalski, R. S. 1989. How to learn imprecise concepts: A method based on two-tiered representation and the AQ15 program. In Kodratoff, Y. and Michalski, R. S., editors 1989, *Machine Learning: An Artificial Intelligence Approach, Vol III.* Morgan Kaufmann Publishers.

Minton, S. 1988. *Learning Search Control Knowledge: An Explanation-Based Approach.* Kluwer Academic Publishers.

Minton, S. 1990. Issues in the design of operator composition systems. In *Proceedings of the Seventh International Conference on Machine Learning.* 304–312.

Mooney, R. J. 1989. The effect of rule use on the utility of explanation-based learning. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence.* 725–730.

Pagallo, G. and Haussler, D. 1990. Boolean feature discovery in empirical learning. *Machine Learning* 5(1):71–100.

Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning* 1(1):81–106.

Quinlan, J. R. 1987. Simplifying decision trees. *International Journal of Man-Machine Studies* 27:221–234.

Rendell, L. A. 1983. A new basis for state-space learning systems and a successful implementation. *Artificial Intelligence* 20(4):369–392.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning internal representations by error propagation. In *Parallel Distributed Processing, Volume 1.* MIT Press. chapter 8, 318–362.

Tambe, M. and Newell, A. 1988. Some chunks are expensive. In *Proceedings of the Fifth International Conference on Machine Learning.* 451–458.

Yoo, J. and Fisher, D. 1991. Concept formation over problem-solving experience. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence.* 630–636.