# Goal-Directed Inductive Learning: Trading off Accuracy for Reduced Error Cost

Foster John Provost

Computer Science Department
University of Pittsburgh
foster@cs.pitt.edu

**Abstract**

In this paper I present issues and results pertaining to goal-directed inductive machine learning, in particular, inductive learning that takes into account the cost of the errors made when the learned concept description is used. Previous work introduced the notion that learning programs should be able to take as input different policies, so that they can learn under different pragmatic considerations. This paper shows that inductive learning can trade off classification accuracy for a reduction in cost, when the learning program uses a cost function to evaluate its learned knowledge. In particular, I discuss costs related to risks, in the classic mushroom and breast-cancer domains, and monetary costs in a domain of diagnosing errors in the local loop of the telephone network.

## 1 Introduction

In inductive machine learning it is usually assumed that the goal of learning is to maximize the classification accuracy of the resultant concept description (sometimes along with some concern for the comprehensibility of the concept description). However, in some cases it is not possible to achieve perfect classification because of insufficient data or computational resources, or an inadequate description language. A policy that treats all misclassification errors identically may run into trouble in domains where different types of errors have different costs, especially if there is a large disparity in the costs.

In previous work [Provost & Buchanan, 1992a], [Provost, 1992] we introduced the notion of *inductive policy* as the strategy for selecting inductive bias based on pragmatic considerations and the underlying assumptions of the domain. In this paper I discuss a particular pragmatic consideration: different errors may have different costs. I show that a standard rule learner can be run with a policy that allows it to trade off accuracy for reduced cost of errors, by criticizing the rules learned based on a user-defined cost function. (The decision to learn with sensitivity to prediction cost is similar to the decision to be sensitive to the cost of measuring features, as in [Tan & Schlimmer, 1990].)

Section 2 describes the approach I used in the experiments below. Sections 3 and 4 describe results in three domains. Section 3 summarizes results (previously reported in [Provost & Buchanan, 1992a]) on the mushroom domain (stored in the UCI repository) where accuracy is traded for reduced risk. Similar results are observed in the breast-cancer domain (also in the UCI repository). Section 4 discusses preliminary results showing that similar tradeoffs can be made with respect to monetary costs, when learning to diagnose errors in the telephone local loop. Finally, Section 5 discusses lessons learned and future directions.

## 2 One Approach

For this work I took a standard machine learning algorithm and built policies around it for learning with sensitivity to the cost of errors

(I discuss alternative approaches below). The learning program is RL [Clearwater & Provost, 1990], a rule-learning program that performs a general-to-specific heuristic search of the space of condition-action rules defined by a domain-specific vocabulary. Each rule's condition is a conjunction of features; a set of rules is viewed as a disjunction. (Domain-specific evidence-gathering techniques can be used to make classifications from a rule set.) The search is guided by an explicit bias describing such things as: criteria for good rules, a complexity limit on the rules, thoroughness of the search, *etc.* RL is described in detail in [Provost, 1992].

The policies built around the learning algorithm have a simple, iterative structure: *learn, criticize, modify, learn, criticize, modify, learn, etc.* The *learn* phase is a run (or several runs) of the learning system. The *criticize* phase analyzes the results of the learning and goes hand-in-hand with the *modify* phase. The concept description may be modified and/or the bias of the learning system may be modified (based on the critique). Finally, the system iterates until some convergence criteria are satisfied. Subsequent learning can use the description formed (so far) as prior knowledge to guide and restrict further learning.

The results summarized in Section 3 and presented in Section 4 were generated with different high-level systems. The results in Section 3 were generated with the SBS Testbed, a system that allows different policies to be specified under different pragmatic constraints, by giving the system different sets of bias selection operators and a bias evaluation function. The SBS Testbed is described in detail in [Provost, 1992], and in [Provost & Buchanan, 1994]. It uses bias-selection operators to create tentative candidate biases, learns rules with each, combines these with the previously learned rule set, and chooses the best of the tentative sets with respect to the evaluation function. The current bias is set to the bias with which the best rule set was learned, and the process iterates until one of several stopping criteria is met. The specific set of operators used below performs a hill-climbing search in RL's bias space, constructing a concept description across biases and using the learned knowledge to guide and restrict its search.

To construct the concept description across biases, the policy combines the most recently learned (tentative) rule set with the currently held rule set. In short, it takes the union of the current set and the tentative set and prunes it, using a greedy heuristic, to find a subset of the rules that performs maximally well with respect to the bias-evaluation function (which could be called *non-monotonic theory construction*). The heuristic sorts the rules based on a certainty factor (as in [Quinlan, 1987]), and then iterates through the rule set removing the first rule (on each iteration) that does not help the performance (*cf.* the greedy rule set pruning heuristic from [Quinlan, 1987]). The evaluation functions used in Section 3 combined accuracy with risk, and are described below.

For the results in Section 4 regarding diagnosing errors in the telephone local loop, it was clear from previous analysis (described in [Danyluk & Provost, 1993b]) that a learning bias would be required that would allow many, very small disjuncts to be learned. The policy used in this domain was to bias RL appropriately for learning small disjuncts, and start with a reasonably quick search of the rule space (using a narrow beam search in RL). The policy was iterative, as above, pruning learned rule sets with a greedy heuristic to remove rules that hurt (or did not help) the performance, and increasing the beam width when no improvement was achieved. In both Sections 3 and 4, the learned rule sets were used to restrict further learning from considering new rules that did not cover at least one new example (see [Provost & Buchanan, 1992b]). The evaluation functions used in Section 4 combined accuracy with monetary costs, and are described below.

This is but one approach to learning with sensitivity to the cost of errors. Unfortunately, machine learning work usually treats error costs as equal and concentrates solely on predictive accuracy. One exception to this is the CRL system [Tcheng, Lambert, Lu, & Rendell, 1989], which allows the user to specify domain dependent error metrics, which can then be used to guide the learning; however, results as to the method's efficacy are not presented. Another alternative technique would be to learn a probabilistic concept description and apply it in combination with

decision-analytic techniques (which take into account costs, benefits and likelihoods). This is similar to the work of Etzioni [Etzioni, 1991], which studies the introduction of decision-analytic techniques into an agent's control policy (and the use of learning to aid estimation). Cost-sensitive-classification work from statistics should also be considered in such a comparison, an overview is given in [James, 1985].

## 3 Risks as Costs (or Better Safe than Sorry)

In many domains the cost of making mistakes is asymmetric—especially where significant risks are involved. For example, in the classic mushroom domain [Schlimmer, 1987], under normal circumstances no harm is done when an edible mushroom is classified as poisonous. In contrast, classifying a poisonous mushroom as edible can have dangerous (sometimes mortal) consequences. The assumption that a certain prediction is more risky than another should lead to a different learning policy than that taken when one can assume that all mistakes can be weighted equally. Obviously, for the mushroom domain a completely safe policy would be not to even use a learning program; instead use a concept description that always predicts a mushroom is poisonous—dangerous predictions would never be made. However, this approach would never allow any mushroom to be eaten. The policy used by mushroom experts varies from expert to expert. A very conservative policy requires considerably more evidence, for example, than a less conservative one [Spear, 1992].

[Provost & Buchanan, 1992a] showed that in the mushroom domain, a learning program can be biased to learn a risk-averse rule set, at the expense of a reduction in the classification accuracy. Manual selection of learning bias achieved as low as 0.04% dangerous errors, keeping an accuracy of 84.5% (averages over 10 runs, training on 100 examples, testing on 915). This work also showed that a learning system that can learn with different policies under different pragmatic considerations (described above) can trade off accuracy for reduced risk automatically. When the system was given a policy to maximize accuracy, it achieved 96.3% accuracy (average 10 runs, *etc.*), but 2.7% of its predictions were risky errors. Factoring in a penalty for risky predictions allowed the risk rate to be reduced to 0.16%, while the accuracy was degraded to 82.2%.

Predicting the edibility of mushrooms is not the only classification task where the risk associated with different errors differs. In many medical diagnosis domains the "cost" of different errors is also different. In the classic breast-cancer domain [Michalski, *et al.*, 1986], it is not possible (given the information provided) to come up with a concept description that makes no errors. However, the two different types of errors in this domain do not have equivalent costs. One type of error would be to predict a recurrence of cancer, when in fact it would not occur. The other type would be to predict that the cancer would not occur, when in fact it would. For this section, I will assume that in terms of safety the second type of error (falsely saying "everything's ok") is more costly than the first (falsely saying "we'd better watch out for recurrence"). Note that this does not tell the whole story, because the medical procedures that stem from a false prediction of recurrence might be costly both in terms of money and safety.

For this data set, previous learning work has reported accuracies in the 65%–75% range. It is interesting to note that the simple rule, *predict the most commonly occurring class from the training set*, performs in this range since approximately 70% of the examples are from one class (no recurrence); however, all of its errors (30%) are of the risky type. This, taken together with the specialists' reported accuracy of only 64% [Michalski, *et al.*, 1986], lends credence to the belief that accuracy alone is not a sufficient basis upon which to judge the results of learning.

Running the cost-sensitive system with a policy for maximizing accuracy achieved 70.3% (averaged over 10 runs, using 100 examples for training, 100 examples for evaluation, and 86 for testing), with a risky prediction rate of 18.1%. Penalizing the system for risky errors allowed a reduction in the risky prediction rate to 4.4%, with a corresponding average accuracy of 44.5%. (Note that the "always safe" rule yields 0% risk in this domain, with an accuracy of 30%.) These results are presented in detail in [Provost, 1992].

The explicit representation of how the system should deal with the accuracy/safety tradeoff allows the system to select biases to learn a concept description that gives a good score with respect to this function, and thereby performs well with respect to the tradeoff. The problem of specifying the bias-evaluation function (in light of a particular inductive policy) remains. However, the first functions chosen in the previous studies performed very well. I assert that specifying this function and letting the system select the bias automatically is easier than manual bias selection (such as that mentioned above), because its specification is more directly related to the assumption in question (*i.e.*, the relative importance of the different prediction errors and predictive accuracy).

## 4 Monetary Costs

MAX is an expert system developed by NYNEX Science and Technology that performs high-level diagnosis of customer-reported telephone troubles [Rabinowitz, Flamholz, Wolin, & Euchner, 1991]. Due to the volume of troubles handled by MAX, even a small improvement in its accuracy is extremely valuable—each dispatch typically involves at least one hour of time by a highly trained worker. Because the process of tuning and updating the MAX system is very tedious and time intensive, and since the MAX system must be modified to take into account the differences that exist with each new site, and must be modified to take into consideration changes over time, it is interesting to consider whether machine-learning techniques can be useful to help in automating such modifications (see [Danyluk & Provost, 1993a]). As mentioned in Section 2, previous work has shown that if a learning system is biased to learn very small disjuncts (as well as larger ones), it can learn high-accuracy concept descriptions from the data used in this study. Now I will consider adding monetary costs to the equation.

### 4.1 A Quick Max Overview

When a customer has a problem with his (or her) phone/line, he calls 611 to report the trouble. A NYNEX representative takes a report of the trouble and also initiates electrical tests on the line (a "mechanized loop test" or MLT[1]). The representative sends the information from the trouble report and the electrical tests to a *maintenance administrator (MA)*, who evaluates the trouble and determines how the company should take care of ("dispatch") the trouble. MAX (Maintenance Administrator eXpert) plays the role of an MA, *i.e.*, it uses the MLT test results, together with other information, to make a screening diagnosis. The only exception is that MAX has the option of referring difficult problems to a human MA. MAX diagnoses a problem based on the following information: results of the MLT, knowledge about the customer's line, and general knowledge about equipment. MAX can "dispatch the trouble" in any of the following ways (costs indicated[2]): to cable repair technician (~$158), to internal repair technician (*i.e.*, to customer's home) (~$158), to central office (~$79), request further testing (~$5), send to human MA (~$0).

### 4.2 Saving Money

As mentioned above, MAX can save large sums of money due to the volume of dispatches. Since one primary goal is to save money, high accuracy should not be the sole consideration when building the system (manually or automatically). In this section, I present a case study of learning with sensitivity to monetary costs. For these results, I concentrated on 1500 examples of dispatches. For the following experiments, 500 examples were used for training, 500 for evaluating concept descriptions during learning, and 500 for final testing.

As in the previous section, the system judged potential concept descriptions based on a linear polynomial that factored in both accuracy and cost—in this case, the sum of the dollar costs of the errors made by the set of rules on the evaluation data. Below I will refer to the

---

[1] MLT was developed, and is maintained, by AT&T.

[2] These costs are reasonable values that have been selected from baseline studies. These are not the actual costs to NYNEX, which are still under debate (*e.g.*, sending a problem to a human MA is not really free).

coefficients in this function as *A* and *B*, based on the function:

score = *A* * accuracy - *B* * cost.

For these experiments I concentrated solely on the cost of errors made, ignoring the cost of correct dispatches. This is based on the assumption that the correct dispatch would have to be made eventually after an erroneous dispatch was detected, so the difference between the costs is the cost of the error. This assumption is arguable for some special cases, but seems to be a good first approximation.

The following experiments deal with two additional issues: the default class and the redundancy level of the pruned rule sets. For previous work, the most frequently occurring class was used as the default class (as is common in inductive machine learning). However, in this domain the most frequently occurring class (dispatch to the customer's home) is a high-cost dispatch. If the objective is to reduce cost, a more rational default would be to send the problem to a human maintenance administrator—considered a freebie for this study.

The second additional issue is that of redundancy in the pruned rule sets. As described above, the criticize/modify step in the iterative policy prunes rules from the currently held rule set. There are two intuitive options for such pruning: (i) prune rules that decrease the evaluation score of the rule set, or (ii) prune rules that do not increase the score of the rule set. For brevity I will call option (i) *redundant pruning*, because it leaves a rule set with some redundancy, and option (ii) *non-redundant pruning*.

### 4.3 Results

When the system was run with A=1 and B=0 (consider accuracy alone), with "dispatch to home" as the default and redundant pruning, the results were as expected: high accuracy (89.2%) and moderate error cost ($6839 for the 500 test cases). Learning with a default of "send to human MA" produced small reductions in the accuracy (84.8%) and the cost ($5258). It is interesting to note, however, that it is better to take the rule set learned with "dispatch to home" as the default, and switch the default to "send to human MA" when it is used on the final test set; this yielded an accuracy of 84.8%, but a cost of only $4626. This phenomenon can be explained by the fact that the reduced cost is not due to the system optimizing the rule set for cost. It is due to the fact that with the latter method the errors made by the default rule have lower costs. More specifically, when pruning for higher accuracy with the default "dispatch to home," the system discarded all rules whose class was "dispatch to home" that made even one erroneous prediction, because the default rule covered for them. This reduced the number of high cost errors of commission in the resultant rule set.

Perhaps counterintuitively, when running the system with "dispatch to home" as the default and redundant pruning, factoring in the cost of errors had (almost) no effect on the performance of the rule sets. Even with A=0 and B=1 (consider cost alone), the accuracy was still 89.2% and the cost around $6700. The reason for this effect was that by using a high-cost default, the best the system could do to minimize the cost of errors was to maximize the accuracy (*i.e.*, the best way to avoid costly errors is not to make errors).

Factoring in the cost of errors was much more effective when the default was set to "send to human MA," since now errors of omission were not costly (and the system could concentrate on reducing the number of costly errors of commission). Table 1 shows that by changing the relative weights of accuracy and cost in the evaluation function, different levels of tradeoff can be achieved (in Table 1, B=1 for all experiments). The tradeoff spectrum is fairly well behaved, ranging from high accuracy and high cost when the function is dominated by the accuracy term, to much lower accuracy with significantly reduced cost when the function is dominated by the cost term.

An interesting question arises from the results summarized in Table 1. One would like the cost to be minimized when cost is the sole factor in evaluation. However, for the experiment with A=0 and B=1, there was still a substantial cost of errors (albeit, one-third of the cost of errors for the highest accuracy case, above). This is due to the redundant pruning. As mentioned above, this domain is one where special cases play a very large role. For a selection of 500 examples, there will be many

Table 1: Accuracy can be traded off for reduced cost (redundant pruning). A is the relative weight of accuracy (as a percent) to cost. Rules are learned on 500 examples, and pruned on a separate 500; accuracy and cost are computed on a third set of 500 examples.

| A | # rules | accuracy | cost | | A | # rules | accuracy | cost |
|---|---|---|---|---|---|---|---|---|
| 0 | 162 | 59.2 | $2330 | | 50 | 175 | 75.2 | $3357 |
| 1 | 162 | 59.2 | $2330 | | 100 | 174 | 75.2 | $3436 |
| 10 | 162 | 59.2 | $2330 | | 150 | 245 | 83.6 | $3446 |
| 30 | 163 | 60.4 | $2335 | | 200 | 244 | 82.8 | $4078 |

special cases that do not appear at all in a disjoint selection of 500 examples. Thus, when the rules are evaluated (on a set of evaluation examples), many rules that would make errors on a different set of examples, instead have no effect on the score (and are kept by the redundant pruning). This has important benefits when trying to maximize accuracy (non-redundant pruning throws away good rules, and degrades accuracy), but hurts when trying to minimize cost.

Table 2 shows the effects of factoring in the cost of errors when non-redundant pruning is used to discard rules that do not seem to affect the performance in a positive manner (otherwise, the experiments are identical to those described in Table 1). We see that now the system shows the intuitively satisfying effect of finding a zero-cost set of rules (a null set of rules) when the objective is to minimize cost. Also, the performance of the learned rule sets behaves reasonably well as the relative weight of accuracy to cost is increased.

## 5 Conclusions and Future Directions

What lessons can we learn from the results presented above? First, it is clear that inductive learning can be effectively directed by different goals—here goals regarding tradeoffs between the accuracy of a learned concept description and the cost of using it—and that an iterative learn-prune-learn cycle is an effective means of directing the learning.

In addition, there are important factors to consider, namely: what default is to be used and what type of pruning is to be used. With respect to defaults, a high-cost default seems to force the system to maximize accuracy, even when directed to minimize cost. This is because with a high-cost default, errors of omission become very costly, so pruning error-prone rules is not efficacious. With respect to pruning, for domains such as NYNEX MAX where special cases play a large role, non-redundant pruning degrades accuracy.

Table 2: Accuracy can be traded off for reduced cost (non-redundant pruning). A is the relative weight of accuracy (as a percent) to cost. Rules are learned on 500 examples, and pruned on a separate 500; accuracy and cost are computed on a third set of 500 examples.

| A | # rules | accuracy | cost | | A | # rules | accuracy | cost |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 27.6 | $0 | | 60 | 42 | 68.6 | $1396 |
| 1 | 31 | 51.4 | $661 | | 70 | 48 | 75.8 | $1875 |
| 10 | 35 | 51.8 | $661 | | 80 | 53 | 79.0 | $2191 |
| 20 | 39 | 52.2 | $996 | | 90 | 53 | 79.0 | $2191 |
| 30 | 33 | 52.8 | $680 | | 100 | 53 | 79.0 | $2191 |
| 40 | 50 | 63.2 | $1242 | | 150 | 56 | 79.8 | $2195 |
| 50 | 42 | 68.6 | $1396 | | 200 | 56 | 80.2 | $2670 |

This is because special case rules seem to have no effect on the evaluation if their corresponding special cases do not appear in the evaluation data. On the other hand, this same phenomenon makes non-redundant pruning effective for reducing the cost of using a rule set, because it throws away rules that do not increase the score (and may make costly mistakes on another set of examples). We saw in Section 4 that with non-redundant pruning, if the objective is to minimize cost (alone), the learner learned a zero-cost rule set.[3]

It is possible that a system that selects inductive bias automatically can select an appropriate default class and pruning strategy (which both fit into an extended definition of inductive bias, as in [Provost & Buchanan, 1994]). I have not yet experimented with this idea.

The precise relationship between the specific evaluation function given to the system and the specific tradeoff achieved needs to be investigated further. At this point, achieving a precise tradeoff is more a matter of intuition and experimentation in the specification of the inductive policy (*viz.*, the evaluation function) than a matter of firm theory. One possibility is: if a given minimum level of accuracy and/or maximum cost is specified, use the criticize/modify phase of the learning process to attempt to achieve these specific levels. Other issues for future investigation include: does the iterative process really provide substantial benefits, or would learning a very large, redundant rule set and pruning (once) be better? And, can we incorporate better techniques for using learned knowledge and goals to guide subsequent search?

## References


Clearwater, S., & Provost, F. (1990). RL4: A Tool for Knowledge-Based Induction. In *Proceedings of the Second International IEEE Conference on Tools for Artificial Intelligence*, p. 24-30. IEEE C.S. Press.

Danyluk, A. P., & Provost, F. J. (1993a). Adaptive Expert Systems: Applying Machine Learning to NYNEX MAX. In *Proceedings of the AAAI-93 Workshop: AI in Service and Support: Bridging the Gap between Research and Applications.*

Danyluk, A. P., & Provost, F. J. (1993b). Small Disjuncts in Action: Learning to Diagnose Errors in the Telephone Network Local Loop. In *Proceedings of the Tenth International Conference on Machine Learning (ML-93).* Morgan Kaufmann.

Etzioni, O. (1991). Embedding Decision-analytic Control in a Learning Architecture. *Artificial Intelligence*, **49**(1991), p. 129-159.

James, M. (1985). *Classification Algorithms.* New York: John Wiley & Sons.

Michalski, R., Mozetic, I., Hong, J., & Lavrac, N. (1986). The Multi-purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, p. 1041-1045. AAAI-Press.

Provost, F., & Buchanan, B. (1992a). Inductive Policy. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, p. 255-261.

Provost, F., & Buchanan, B. (1992b). Inductive Strengthening: the effects of a simple heuristic for restricting hypothesis space search. In K. P. Jantke (ed.), *Analogical and Inductive Inference (Proceedings of the 3rd International Workshop on Analogical and Inductive Inference).* Berlin: Springer-Verlag.

Provost, F. J. (1992). *Policies for the Selection of Bias in Inductive Machine Learning.* Ph.D. Thesis, University of Pittsburgh.

Provost, F. J., & Buchanan, B. G. (1994). Inductive Policy: The Pragmatics of Bias Selection. *Submitted to: Machine Learning.*


[3] Thanks, in part, to the existence of a zero-cost error.

Quinlan, J. (1987). Generating Production Rules from Decision Trees. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, p. 304-307.

Rabinowitz, H., Flamholtz, J., Wolin, E., & Euchner, J. (1991). NYNEX MAX: A Telephone Trouble Screening Expert. In *Innovative Applications of Artificial Intelligence 3*, p. 213-230.

Schlimmer, J. (1987). *Concept Acquisition Through Representational Adjustment.* Ph.D. Thesis, Department of Information and Computer Science, University of California at Irvine.

Spear, M. (1992). V. P. of Research, Sylvan Spawn Laboratories. Personal Comm.

Tan, M., & Schlimmer, J. (1990). Two Case Studies in Cost-Sensitive Concept Acquisition. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, p. 854-860. AAAI Press.

Tcheng, D., Lambert, B., Lu, S., & Rendell, L. (1989). Building Robust Learning Systems by Combining Induction and Optimization. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, p.806-812.