Published in 1994, William W. Cohen & Haym Hirsh, eds., *Machine Learning: Proceedings of the Eleventh International Conference*, 121-129, Morgan Kaufmann Publishers, San Francisco, CA.

Irrelevant Features and the Subset Selection Problem

George H. John
Computer Science Dept.
Stanford University
Stanford, CA 94305
gjohn@CS.Stanford.EDU

Ron Kohavi
Computer Science Dept.
Stanford University
Stanford, CA 94305
ronnyk@CS.Stanford.EDU

Karl Pfleger
Computer Science Dept.
Stanford University
Stanford, CA 94305
kpfleger@CS.Stanford.EDU

Abstract

We address the problem of finding a subset of features that allows a supervised induction algorithm to induce small high-accuracy concepts. We examine notions of relevance and irrelevance, and show that the definitions used in the machine learning literature do not adequately partition the features into useful categories of relevance. We present definitions for irrelevance and for two degrees of relevance. These definitions improve our understanding of the behavior of previous subset selection algorithms, and help define the subset of features that should be sought. The features selected should depend not only on the features and the target concept, but also on the induction algorithm. We describe a method for feature subset selection using cross-validation that is applicable to any induction algorithm, and discuss experiments conducted with ID3 and C4.5 on artificial and real datasets.

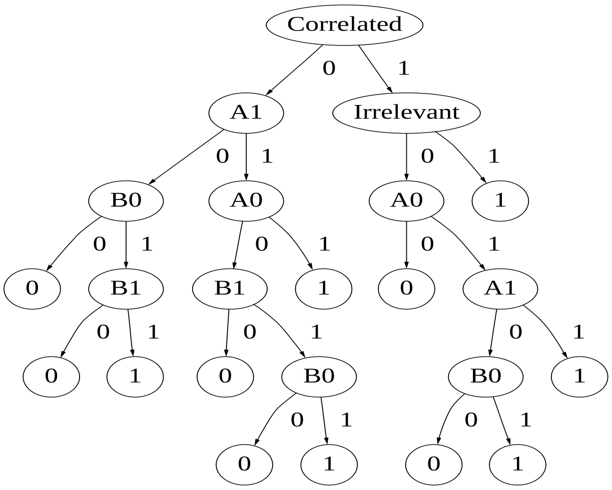

Figure 1: An example where ID3 picks a bad relevant feature (correlated) for the root, and an irrelevant feature (irrelevant).

1 INTRODUCTION

In supervised learning, one is given a training set containing labelled instances. The instances are typically specified by assigning values to a set of features, and the task is to induce a hypothesis that accurately predicts the label of novel instances. Following Occam's razor (Blumer *et al.* 1987), minimum description length (Rissanen 1986), and minimum message length (Wallace & Freeman 1987), one usually attempts to find structures that correctly classify a large subset of the training set, and yet are not so complex that they begin to overfit the data. Ideally, the induction algorithm should use only the subset of features that leads to the best performance.

Since induction of minimal structures is NP-hard in many cases (Hancock 1989; Blum & Rivest 1992), algorithms usually conduct a heuristic search in the space of possible hypotheses. This heuristic search may lead to induced concepts which depend on irrelevant features, or in some cases even relevant features that hurt the overall accuracy. Figure 1 shows such a choice of a non-optimal split at the root made by ID3 (Quinlan 1986). The Boolean target concept is $(A0 \wedge A1) \vee (B0 \wedge B1)$. The feature named "irrelevant" is uniformly random, and the feature "correlated" matches the class label 75% of the time. The left subtree is the correct decision tree, which is correctly induced if the "correlated" feature is removed from the data. C4.5 (Quinlan 1992) and CART (Breiman *et al.* 1984) induce similar trees with the "correlated" feature at the root. Such a split causes all these induction algorithms to generate trees that are less accurate than if this feature is completely removed.

The problem of *feature subset selection* involves finding a "good" set of features under some objective function. Common objective functions are prediction accuracy, structure size, and minimal use of input features (*e.g.*, when features are tests that have an associated cost). In this paper we chose to investigate the possibility of improving prediction accuracy or decreasing the size of the structure without significantly decreasing prediction accuracy. This specific problem has been thoroughly investigated in the statistics literature, but under assumptions that do not apply to most learning algorithms(see Section 5).

We begin by describing the notions of relevance and irrelevance that have been previously defined by researchers. We show that the definitions are too coarse-grained, and that better understanding can be achieved by looking at two degrees of relevance. Section 3 looks at two models for feature subset selection: the filter model and the wrapper model. We claim that the wrapper model is more appropriate than the filter model, which has received more attention in machine learning. Section 4 presents our experimental results, Section 5 describes related work, and Section 6 provides a summary and discussion of future work.

2 DEFINING RELEVANCE

In this section we present definitions of relevance that have been suggested in the literature. We then show a single example where the definitions give unexpected answers, and we suggest that two degrees of relevance are needed: weak and strong.

The input to a supervised learning algorithm is a set of n training instances. Each instance $\mathbf{X}$ is an element of the set $F_1 \times F_2 \times \cdots \times F_m$, where F_i is the domain of the ith feature. Training instances are tuples $\langle \mathbf{X}, Y \rangle$ where Y is the label, or output. Given an instance, we denote the value of feature X_i by x_i. The task of the induction algorithm is to induce a structure (*e.g.*, a decision tree or a neural net) such that, given a new instance, it is possible to accurately predict the label Y. We assume a probability measure p on the space $F_1 \times F_2 \times \cdots \times F_m \times Y$. Our general discussion does not make any assumptions on the features or on the label; they can be discrete, continuous, linear, or structured, and the label may be single-valued or a multi-valued vector of arbitrary dimension.

2.1 EXISTING DEFINITIONS

Almuallim and Dietterich (1991, p. 548) define relevance under the assumption that all features and the label are Boolean and that there is no noise.

Definition 1 *A feature X_i is said to be* **relevant** *to a concept $\mathcal{C}$ if X_i appears in every Boolean formula that represents $\mathcal{C}$ and irrelevant otherwise.*

Gennari *et al.* (1989, Section 5.5) define relevance as[1]

[1]The definition given is a formalization of their statement: "Features are relevant if their values vary systematically with category membership."

Definition	Relevant	Irrelevant
Definition 1	X_1	X_2, X_3, X_4, X_5
Definition 2	None	All
Definition 3	All	None
Definition 4	X_1	X_2, X_3, X_4, X_5

Table 1: Feature relevance for the Correlated XOR problem under the four definitions.

Definition 2 *X_i is relevant iff there exists some x_i and y for which $p(X_i = x_i) > 0$ such that*

$$p(Y = y \mid X_i = x_i) \neq p(Y = y) \ .$$

Under this definition, X_i is relevant if knowing its value can change the estimates for Y, or in other words, if Y is conditionally dependent of X_i. Note that this definition fails to capture the relevance of features in the parity concept, and may be changed as follows.

Let S_i be the set of all features except X_i, *i.e.*, $S_i = \{X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_m\}$. Denote by s_i a value-assignment to all features in S_i.

Definition 3 *X_i is relevant iff there exists some x_i, y, and s_i for which $p(X_i = x_i) > 0$ such that*

$$p(Y = y, S_i = s_i \mid X_i = x_i) \neq p(Y = y, S_i = s_i) \ .$$

Under the following definition, X_i is relevant if the probability of the label (given all features) can change when we eliminate knowledge about the value of X_i.

Definition 4 *X_i is relevant iff there exists some x_i, y, and s_i for which $p(X_i = x_i, S_i = s_i) > 0$ such that*

$$p(Y = y \mid X_i = x_i, S_i = s_i) \neq p(Y = y \mid S_i = s_i) \ .$$

The following example shows that all the definitions above give unexpected results.

Example 1 (Correlated XOR)
Let features $X_1, \ldots, X_5$ be Boolean. The instance space is such that X_2 and X_3 are negatively correlated with X_4 and X_5, respectively, i.e., $X_4 = \overline{X_2}$, $X_5 = \overline{X_3}$. There are only eight possible instances, and we assume they are equiprobable. The (deterministic) target concept is

$$Y = X_1 \oplus X_2 \qquad (\oplus \text{ denotes XOR}) \ .$$

Note that the target concept has an equivalent Boolean expression, namely, $Y = X_1 \oplus \overline{X_4}$. The features X_3 and X_5 are irrelevant in the strongest possible sense. X_1 is indispensable, and one of X_2, X_4 can be disposed

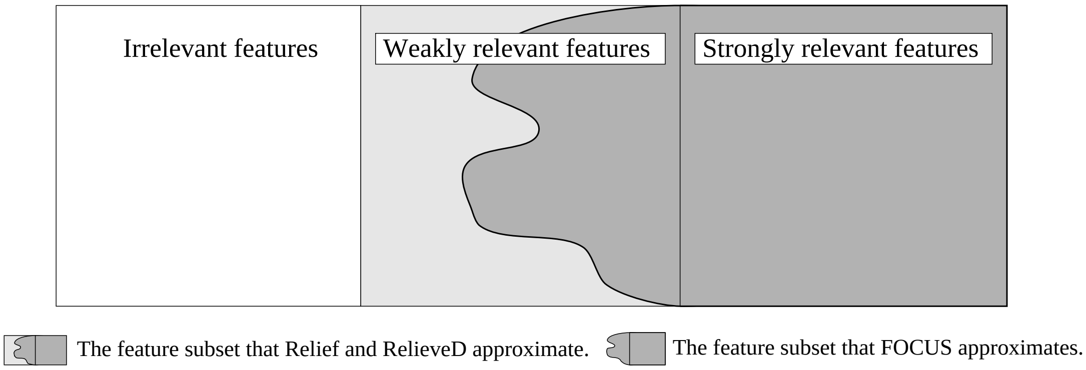

Figure 2: A view of feature relevance.

of, but we must have one of them. Table 1 shows for each definition, which features are relevant, and which are not.

According to Definition 1, X_3 and X_5 are clearly irrelevant; both X_2 and X_4 are irrelevant because each can be replaced by the negation of the other. By Definition 2, all features are irrelevant since for any output value y and feature value x, there are two instances that agree with the values. By Definition 3, every feature is relevant, because knowing its value changes the probability of four of the eight possible instances from 1/8 to zero. By Definition 4, X_3 and X_5 are clearly irrelevant, and both X_2 and X_4 are irrelevant, since they do not add any information to S_4 and S_2, respectively.

Although such simple negative correlations are unlikely to occur, domain constraints create a similar effect. When a nominal attribute such as color is encoded as input to a neural network, it is customary to use a *local encoding*, where each value is represented by an indicator variable. For example, the local encoding of a four-valued nominal $\{a, b, c, d\}$ would be $\{0001, 0010, 0100, 1000\}$. Under such an encoding, any single indicator variable is redundant and can be determined by the rest. Thus most definitions of relevancy will declare all indicator variables to be irrelevant.

2.2 STRONG AND WEAK RELEVANCE

We now claim that two degrees of relevance are required. Definition 4 defines strong relevance. Strong relevance implies that the feature is indispensable in the sense that it cannot be removed without loss of prediction accuracy.

Definition 5 (Weak relevance)
A feature X_i is weakly relevant iff it is not strongly relevant, and there exists a subset of features S'_i of S_i for which there exists some x_i, y, and s'_i with $p(X_i = x_i, S'_i = s'_i) > 0$ such that

$$p(Y = y \mid X_i = x_i, S'_i = s'_i) \neq p(Y = y \mid S'_i = s'_i)$$

Weak relevance implies that the feature can sometimes contribute to prediction accuracy. Features are *relevant* if they are either strongly or weakly relevant, and are *irrelevant* otherwise. Irrelevant features can never contribute to prediction accuracy, by definition.

In Example 1, feature X_1 is strongly relevant; features X_2 and X_4 are weakly relevant; and X_3 and X_5 are irrelevant. Figure 2 shows our view of relevance.

Algorithms such as FOCUS (Almuallim & Dietterich 1991) (see Section 3.1) find a minimal set of features that are sufficient to determine the concept. Given enough data, these algorithms will select all strongly relevant features, none of the irrelevant ones, and a smallest subset of the weakly relevant features that are sufficient to determine the concept. Algorithms such as Relief (Kira & Rendell 1992a; 1992b; Kononenko 1994) (see Section 3.1) attempt to efficiently approximate the set of relevant features.

3 FEATURE SUBSET SELECTION

There are a number of different approaches to subset selection. In this section, we claim that the *filter model*, the basic methodology used by algorithms like FOCUS and Relief, should be replaced with the *wrapper model* that utilizes the induction algorithm itself.

3.1 THE FILTER MODEL

We review three instances of the filter model: FOCUS, Relief, and the method used by Cardie (1993) .

The FOCUS algorithm (Almuallim & Dietterich 1991), originally defined for noise-free Boolean domains, exhaustively examines all subsets of features, selecting the minimal subset of features that is sufficient to determine the label. This is referred to as the MIN-FEATURES bias.

This bias has severe implications when applied blindly without regard for the resulting induced concept. For

Figure 3: The feature filter model, in which the features are filtered independent of the induction algorithm.

example, in a medical diagnosis task, a set of features describing a patient might include the patient's social security number (SSN). (We assume that features other than SSN are sufficient to determine the correct diagnosis.) When FOCUS searches for the minimum set of features, it will pick the SSN as the only feature needed to uniquely determine the label[2]. Given only the SSN, any induction algorithm will generalize very poorly.

The Relief algorithm (Kira & Rendell 1992a; 1992b) assigns a "relevance" weight to each feature, which is meant to denote the relevance of the feature to the target concept. Relief is a randomized algorithm. It samples instances randomly from the training set and updates the relevance values based on the difference between the selected instance and the two nearest instances of the same and opposite class (the "near-hit" and "near-miss").

The Relief algorithm does not attempt to determine *useful* subsets of the weakly relevant features:

> Relief does not help with redundant features. If most of the given features are relevant to the concept, it would select most of them even though only a fraction are necessary for concept description (Kira & Rendell 1992a, page 133).

In real domains, many features have high correlations, and thus many are (weakly) relevant, and will not be removed by Relief[3].

Cardie (1993) uses subset selection to remove irrelevant features from a dataset to be used with the nearest-neighbor algorithm. As a metric of an attribute's usefulness, C4.5 was used to induce a decision tree from a training set, and those features that did not appear in the resulting tree were removed. The resulting performance of the nearest-neighbor classifier was higher than with the entire set of features.

Figure 3 describes the feature filter model, which characterizes these algorithms. In this model, the feature subset selection is done as a preprocessing step. The disadvantage of the filter approach is that it totally ignores the effects of the selected feature subset on the performance of the induction algorithm.

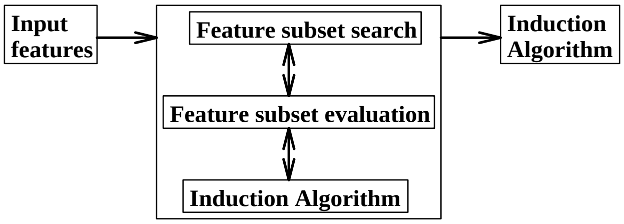

Figure 4: The **wrapper** model. The induction algorithm is used as a "black box" by the subset selection algorithm.

We claim that to determine a useful subset of features, the subset selection algorithm must take into account the biases of the induction algorithm in order to select a subset that will ultimately result in an induced structure with high predictive accuracy on unseen data. This motivated us to consider the the following approach which does employ such information.

3.2 THE WRAPPER MODEL

In the wrapper model that we propose, the feature subset selection algorithm exists as a wrapper around the induction algorithm (see Figure 4). The feature subset selection algorithm conducts a search for a good subset using the induction algorithm itself as part of the evaluation function.

3.2.1 Subset Evaluation

Given a subset of features, we want to estimate the accuracy of the induced structure using only the given features. We propose evaluating the subset using n-fold cross validation (Breiman *et al.* 1984; Weiss & Kulikowski 1991). The training data is split into n approximately equally sized partitions. The induction algorithm is then run n times, each time using $n-1$ partitions as the training set and the other partition as the test set. The accuracy results from each of the n runs are then averaged to produce the estimated accuracy.

Note that no knowledge of the induction algorithm is necessary, except the ability to test the resulting structure on the validation sets.

3.2.2 Searching the space of subsets

Finding a good subset of features under some measure requires searching the space of feature subsets. Many common AI search algorithms may be employed for this task, and some have been suggested in the

[2] This is true even if SSN is encoded in ℓ binary features as long as more than ℓ other features are required to uniquely determine the diagnosis.

[3] In the simple parity example used in (Kira & Rendell 1992a; 1992b), there were only strongly relevant and irrelevant features, so Relief found the strongly relevant features most of the time.

statistics literature under various assumptions about the induction algorithm (see Section 5). These assumptions do not hold for most machine learning algorithms, hence heuristic search is used.

One simple greedy algorithm, called *backward elimination*, starts with the full set of features, and greedily removes the one that most improves performance, or degrades performance slightly. A similar algorithm, called *forward selection* starts with the empty set of features, and greedily adds features.

The algorithms can be improved by considering both addition of a feature and deletion of a feature at each step. For example, during backward elimination, consider adding one of the deleted features if it improves performance. Thus at each step the algorithm greedily either adds or deletes. The only difference between the backward and forward versions is that the backward version starts with all features and the forward version starts with no features. The algorithms are straightforward and are described in many statistics books (Draper & Smith 1981; Neter, Wasserman, & Kutner 1990) under the names *backward stepwise elimination* and *forward stepwise selection*. One only has to be careful to set the degradation and improvement margins so that cycles will not occur.

The above heuristic increases the overall running time of the black-box induction algorithm by a multiplicative factor of $O(m^2)$ in the worst case, where m is the number of features. While this may be impractical in some situations, it does not depend on n, the number of instances. As noted in Cohen (1993) , divide and conquer systems need much more time for pruning than for growing the structure (by a factor of $O(n^2)$ for random data). By pruning after feature subset selection, pruning may be much faster.

4 EXPERIMENTAL RESULTS

In order to evaluate the feature subset selection using the wrapper model we propose, we ran experiments on nine datasets. The C4.5 program is the program that comes with Quinlan's book (Quinlan 1992); the ID3 results were obtained by running C4.5 and using the unpruned trees. On the artificial datasets, we used the "-s -m1" C4.5 flags, which indicate that subset splits may be used and that splitting should continue until purity. To estimate the accuracy for feature subsets, we used 25-fold cross validation. Thus our feature subsets were evaluated solely on the basis of the training data without using data from the test set. Only after the best feature subset was chosen by our algorithm did we use the test set to give the results appearing in this section.

In our experiments we found significant variance in the relevance rankings given by Relief. Since Relief randomly samples instances and their neighbors from the training set, the answers it gives are unreliable without a very high number of samples. We were worried by this variance, and implemented a deterministic version of Relief that uses all instances and all near-hits and near-misses of each instance. This gives the results one would expect from Relief if run for an infinite amount of time, but requires only as much time as the standard Relief algorithm with the number of samples equal to the size of the training set. Since we are no longer worried by high variance, we call this deterministic variant RelieveD. In our experiments, features with relevancy rankings below 0 were removed.

The real-world datasets were taken from the UC-Irvine repository (Murphy & Aha 1994) and from Quinlan (1992) . Figures 5 and 6 summarize our results. We give details for those datasets that had the largest differences either in accuracy or tree size.

Artificial datasets

CorrAL This is the same dataset and concept described in the Introduction (Figure 1), which has a high Correlation between one Attribute and the Label, hence "CorrAL."

Monk1*,Monk3* These datasets were taken from Thrun *et al.* (1991). The datasets have six features, and both target concepts are disjunctive. We created 10 random training sets of the same size as was given in Thrun *et al.* (1991), and tested on the full space.

Parity 5+5 The target concept is the parity of five bits. The dataset contains 10 features, 5 uniformly random (irrelevant). The training set contained 100 instances, while all 1024 instances were used in the test set.

Real-world datasets

Vote This dataset includes votes for U.S. House of Representatives Congresspersons on the 16 key votes identified by the Congressional Quarterly Almanac Volume XL. The data set consists of 16 features, 300 training instances and 135 test instances.

Credit (or CRX) The dataset contains instances for credit card applications. There are 15 features and a Boolean label. The dataset was divided by Quinlan into 490 training instances and 200 test instances.

Labor The dataset contains instances for acceptable and unacceptable contracts. It is a small dataset with 16 features, a training set of 40 instances, and a test set of 17 instances.

Our results show that the main advantage of doing subset selection is that smaller structures are created.

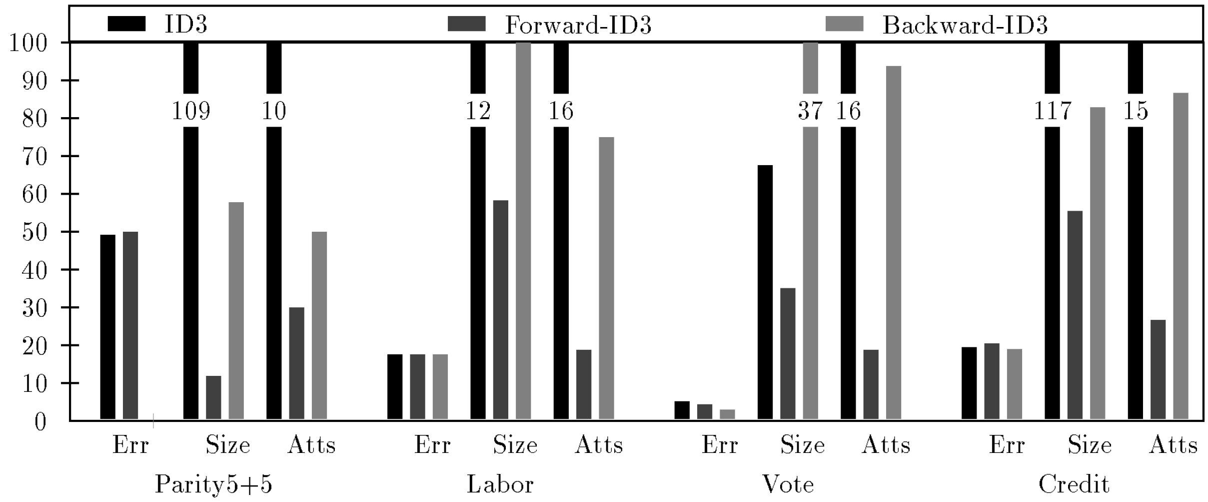

Figure 5: Results for subset selection using the ID3 Algorithm. For each dataset and algorithm we show the error on the test set, the relative size of the induced tree (as compared with the largest of the three, whose absolute size is given), and the relative number of features in the training set.

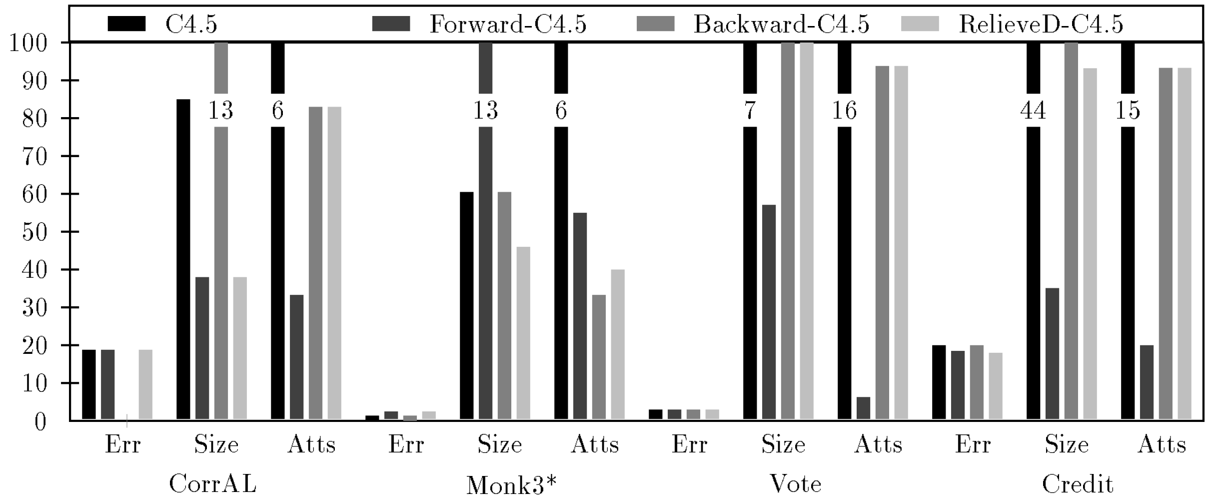

Figure 6: Results for the C4.5 Algorithm.

Smaller trees allow better understanding of the domain, and are thus preferable if the error rate does not increase significantly. In the Credit database, the size of the resulting tree after forward stepwise selection with C4.5 decreased from 44 nodes to 16 nodes, accompanied by a slight improvement in accuracy.

Feature subset selection using the wrapper model did not significantly change generalization performance. The only significant difference in performance was on parity5+5 and CorrAL using stepwise backward elimination, which reduced the error to 0% from 50% and 18.8% respectively. Experiments were also run on the Iris, Thyroid, and Monk1* datasets. The results on these datasets were similar to those reported in this paper.

We observed high variance in the 25-fold cross-validation estimates of the error. Since our algorithms depend on cross-validation to choose which feature to add or remove, a single "optimistic" 25-CV estimate caused premature stopping in many cases. Such an estimate of low error could not be improved, and the algorithm stopped.

RelieveD performed well in practice, and reduced the number of features. The number of features deleted, however, is low compared to our forward stepwise selection.

Although in most cases the subset selection algorithms have found small subsets, this need not always be the case. For example, if the data has redundant features but also has many missing values, a learning algorithm should induce a hypothesis which makes use of these redundant features. Thus the best feature subset is not always the minimal one.

5 RELATED WORK

Researchers in statistics (Boyce, Farhi, & Weischedel 1974; Narendra & Fukunaga 1977; Draper & Smith 1981; Miller 1990; Neter, Wasserman, & Kutner 1990) and pattern recognition (Devijver & Kittler 1982; Ben-Bassat 1982) have investigated the feature subset selection problem for decades, but most work has concentrated on subset selection using linear regression.

Sequential backward elimination, sometimes called sequential backward selection, was introduced in Marill & Green (1963). Kittler generalized the different variants including forward methods, stepwise methods, and "plus ℓ–take away r." Branch and bound algorithms were introduced by Narendra & Fukunaga (1977). Finally, more recent papers attempt to use AI techniques, such as beam search and bidirectional search (Siedlecki & Sklansky 1988), best first search (Xu, Yan, & Chang 1989), and genetic algorithms (Vafai & De Jong 1992).

Many measures have been suggested to evaluate the subset selection (as opposed to cross validation), such as adjusted mean squared error, adjusted multiple correlation coefficient, and the C_p statistic (Mallows 1973). In Mucciardi & Gose (1971), seven different techniques for subset selection were empirically compared for a nine-class electrocardiographic problem.

The search for the best subset can be improved by making assumptions on the evaluation function. The most common assumption is monotonicity, that increasing the subset can only increase the performance. Under such assumptions, the search space can be pruned by the use of dynamic programming and branch-and-bound techniques. The monotonicity assumption is not valid for many induction algorithms used in machine learning (see for example Figure 1).

The terms weak and strong relevance are used in Levy (1993) to denote formulas that appear in one minimal derivation, or in all minimal derivations. We found the analog for feature subset selection helpful. Moret (1982) defines *redundant features* and *indispensable features* for the discrete case. The definitions are similar to our notions of irrelevance and strong relevance, but do not coincide on some boundary cases. Determinations were introduced by Russel (1986; 1989) under a probabilistic setting, and used in a deterministic, non-noisy setting in Schlimmer (1993), and may help analyze redundancies.

In the machine learning literature, the most closely related work is FOCUS and Relief which we have described. The PRESET algorithm described in Modrzejewski (1993) is another filter algorithm that uses the theory of *Rough Sets* to heuristically rank the features, assuming a noiseless Boolean domain. Littlestone (1988) introduced the WINNOW family of algorithms that efficiently learns linear threshold functions with many irrelevant features in the mistake bound model and in Valiant's PAC model.

Recently the machine learning community has shown increasing interest in this topic. Moore and Lee (1994) present a set of efficient algorithms to "race" competing subsets until one outperforms all others, thus avoiding the computation involved in fully evaluating each subset. Their method is an example of the wrapper model using a memory-based (instance-based) algorithm as the induction engine, and leave-one-out cross validation (LOOCV) as the subset evaluation function. Searching for feature subsets is done using backward and forward hill-climbing techniques similar to ours, but they also present a new method—schemata search—that seems to provide a four-fold speedup in some cases. Langley and Sage (1994) have also recently used LOOCV in a nearest-neighbor algorithm.

Caruana and Freitag (1994) test the forward and backward stepwise methods on the calendar apprentice domain, using the wrapper model and a variant of ID3 as the induction engine. They introduce a caching scheme to save evaluations of subsets, which speeds up the search quite a bit, but it seems to be specific to ID3.

Skalak (1994) uses the wrapper model for feature subset selection and for decreasing the number of prototypes stored in instance-based methods. He shows that this can sometimes increase the prediction accuracy in some cases.

6 DISCUSSION AND FUTURE WORK

We defined three categories of feature relevance in order to clarify our understanding of existing algorithms, and to help define our goal: find all strongly relevant features, no irrelevant features, and a useful subset of the weakly relevant features that yields good performance. We advocated the wrapper model as a means of identifying useful feature subsets, and tested two greedy search heuristics—forward stepwise selection and backward stepwise elimination—using cross validation to evaluate performance.

Our results show that while accuracy did not improve significantly (except for the parity5+5 and CorrAL datasets) the generated trees induced by ID3 and C4.5 were generally smaller using the wrapper model. We also tested C4.5 on several datasets using RelieveD as a feature filter, and observed that while it removes some features, it does not remove as many features as did our forward selection method.

We included the results for forward and backward search methods separately to illustrate the different biases of the two greedy strategies, but one can easily imagine combining the two methods to achieve the best behavior of both. In the simplest approach, we could run both methods separately and select the best of the two results, based on our evaluation method. This should yield the same positive results as forward search in most cases while retaining the reasonable behavior of backward search for problems with high feature interaction, such as parity5+5. In all but one experiment, the smaller of the two trees produced by forward stepwise selection and backward stepwise elimination was smaller than the tree induced by ID3 or C4.5, and it was not larger in the last case.

Note that even the better of the backward and forward results should not be taken as the best performance possible from the wrapper model. More comprehensive search strategies could search a larger portion of the search space and might yield improved performance. The feature relevance rankings produced by RelieveD could be used to create a set of initial states in the space from which to search.

One possible reason for the lack of significant improvement of prediction accuracy over C4.5 is that C4.5 does quite well on most of the datasets tested here, leaving little room for improvement. This seems to be in line with with Holte's claims (Holte 1993). Harder datasets might show more significant improvement. Indeed the wrapper model produced the most significant improvement for the two datasets (parity5+5 and CorrAL) on which C4.5 performed the worst.

Future work should address better search strategies, better evaluation estimates, and should test the wrapper model with other classes of learning algorithms. Research aimed at improving the evaluation estimates for subsets should attempt to find a method of reducing the problem of high variance in the cross validation estimates. We believe this may be possible by averaging a number of separate cross validation runs (shuffling data between runs), and by using stratified cross validation.

Feature subset selection is an important problem that has many ramifications. Our introductory example (Figure 1) shows that common algorithms such as ID3, C4.5, and CART, fail to ignore features which, if ignored, would improve accuracy. Feature subset selection is also useful for constructive induction (Pagallo & Haussler 1990) where features can be constructed and tested using the wrapper model to determine if they improve performance. Finally, in real world applications, features may have an associated cost (*i.e.*, when the value of a feature is determined by an expensive test). The feature selection algorithms can be modified to prefer removal of high-cost tests.

Acknowledgements

We have benefitted from the comments and advice of Wray Buntine, Tom Dietterich, Jerry Friedman, Igor Kononenko, Pat Langley, Scott Roy and the anonymous reviewers. Richard Olshen kindly gave us access to the CART software. Thanks to Nils Nilsson and Yoav Shoham for supporting the $\mathcal{MLC}$++ project, and everyone working on $\mathcal{MLC}$++, especially Brian Frasca and Richard Long. George John was supported by a National Science Foundation Graduate Research Fellowship. The $\mathcal{MLC}$++ project is partly funded by ONR grant N00014-94-1-0448.

References

Almuallim, H., and Dietterich, T. G. 1991. Learning with many irrelevant features. In *Ninth National Conference on Artificial Intelligence*, 547–552. MIT Press.

Ben-Bassat, M. 1982. Use of distance measures, information measures and error bounds in feature evaluation. In Krishnaiah, P. R., and Kanal, L. N., eds., *Handbook of Statistics*, volume 2. North-Holland Publishing Company. 773–791.

Blum, A. L., and Rivest, R. L. 1992. Training a 3-node neural network is NP-complete. *Neural Networks* 5:117–127.

Blumer, A.; Ehrenfeucht, A.; Haussler, D.; and Warmuth, M. K. 1987. Occam's razor. *Information Processing Letters* 24:377–380.

Boyce, D.; Farhi, A.; and Weischedel, R. 1974. *Optimal Subset Selection.* Springer-Verlag.

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

Cardie, C. 1993. Using decision trees to improve case-based learning. In *Proceedings of the Tenth International Conference on Machine Learning*, 25–32. Morgan Kaufmann.

Caruana, R., and Freitag, D. 1994. Greedy attribute selection. In Cohen, W. W., and Hirsh, H., eds., *Machine Learning: Proceedings of the Eleventh International Conference.* Morgan Kaufmann.

Cohen, W. W. 1993. Efficient pruning methods for separate-and-conquer rule learning systems. In *13th International Joint Conference on Artificial Intelligence*, 988–994. Morgan Kaufmann.

Devijver, P. A., and Kittler, J. 1982. *Pattern Recognition: A Statistical Approach.* Prentice-Hall International.

Draper, N. R., and Smith, H. 1981. *Applied Regression Analysis.* John Wiley & Sons, 2nd edition.

Gennari, J. H.; Langley, P.; and Fisher, D. 1989. Models of incremental concept formation. *Artificial Intelligence* 40:11–61.

Hancock, T. R. 1989. On the difficulty of finding small consistent decision trees. Unpublished Manuscript, Harvard University.

Holte, R. C. 1993. Very simple classification rules perform well on most commonly used datasets. *Machine Learning* 11:63–90.

Kira, K., and Rendell, L. A. 1992a. The feature selection problem: Traditional methods and a new algorithm. In *Tenth National Conference on Artificial Intelligence*, 129–134. MIT Press.

Kira, K., and Rendell, L. A. 1992b. A practical approach to feature selection. In *Proceedings of the Ninth International Conference on Machine Learning*. Morgan Kaufmann.

Kononenko, I. 1994. Estimating attributes: Analysis and extensions of Relief. In *Proceedings of the European Conference on Machine Learning*.

Langley, P., and Sage, S. 1994. Oblivious decision trees and abstract cases. In *Working Notes of the AAAI94 Workshop on Case-Based Reasoning*. In press.

Levy, A. Y. 1993. *Irrelevance Reasoning in Knowledge Based Systems*. Ph.D. Dissertation, Stanford University.

Littlestone, N. 1988. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning* 2:285–318.

Mallows, C. L. 1973. Some comments on c_p. *Technometrics* 15:661–675.

Marill, T., and Green, D. M. 1963. On the effectiveness of receptors in recognition systems. *IEEE Transactions on Information Theory* 9:11–17.

Miller, A. J. 1990. *Subset Selection in Regression*. Chapman and Hall.

Modrzejewski, M. 1993. Feature selection using rough sets theory. In Brazdil, P. B., ed., *Proceedings of the European Conference on Machine Learning*, 213–226.

Moore, A. W., and Lee, M. S. 1994. Efficient algorithms for minimizing cross validation error. In Cohen, W. W., and Hirsh, H., eds., *Machine Learning: Proceedings of the Eleventh International Conference*. Morgan Kaufmann.

Moret, B. M. E. 1982. Decision trees and diagrams. *ACM Computing Surveys* 14(4):593–623.

Mucciardi, A. N., and Gose, E. E. 1971. A comparison of seven techniques for choosing subsets of pattern recognition properties. *IEEE Transactions on Computers* C-20(9):1023–1031.

Murphy, P. M., and Aha, D. W. 1994. UCI repository of machine learning databases. For information contact ml-repository@ics.uci.edu.

Narendra, M. P., and Fukunaga, K. 1977. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers* C-26(9):917–922.

Neter, J.; Wasserman, W.; and Kutner, M. H. 1990. *Applied Linear Statistical Models*. Irwin: Homewood, IL, 3rd edition.

Pagallo, G., and Haussler, D. 1990. Boolean feature discovery in empirical learning. *Machine Learning* 5:71–99.

Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning* 1:81–106. Reprinted in Shavlik and Dietterich (eds.) Readings in Machine Learning.

Quinlan, J. R. 1992. *C4.5: Programs for Machine Learning*. Los Altos, California: Morgan Kaufmann.

Rissanen, J. 1986. Stochastic complexity and modeling. *Ann. Statist* 14:1080–1100.

Russel, S. J. 1986. Preliminary steps toward the automation of induction. In *Proceedings of the National Conference on Artificial Intelligence*, 477–484.

Russel, S. J. 1989. *The Use of Knowledge in Analogy and Induction*. Morgan Kaufmann.

Schlimmer, J. C. 1993. Efficiently inducing determinations: A complete and systematic search algorithm that uses optimal pruning. In *Proceedings of the Tenth International Conference on Machine Learning*, 284–290. Morgan Kaufmann.

Siedlecki, W., and Sklansky, J. 1988. On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence* 2(2):197–220.

Skalak, D. B. 1994. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In Cohen, W. W., and Hirsh, H., eds., *Machine Learning: Proceedings of the Eleventh International Conference*. Morgan Kaufmann.

Thrun, S. B., et al. 1991. The monk's problems: A performance comparison of different learning algorithms. Technical Report CMU-CS-91-197, Carnegie Mellon University.

Vafai, H., and De Jong, K. 1992. Genetic algorithms as a tool for feature selection in machine learning. In *Fourth International Conference on Tools with Artificial Intelligence*, 200–203. IEEE Computer Society Press.

Wallace, C., and Freeman, P. 1987. Estimation and inference by compact coding. *Journal of the Royal Statistical Society (B)* 49:240–265.

Weiss, S. M., and Kulikowski, C. A. 1991. *Computer Systems that Learn*. San Mateo, CA: Morgan Kaufmann.

Xu, L.; Yan, P.; and Chang, T. 1989. Best first strategy for feature selection. In *Ninth International Conference on Pattern Recognition*, 706–708. IEEE Computer Society Press.