Appears in the AAAI Fall Symposium on Relevance, 1994

Feature Subset Selection as Search with Probabilistic Estimates

Ron Kohavi
Computer Science Department
Stanford University
Stanford, CA 94305
ronnyk@CS.Stanford.EDU
http://robotics.stanford.edu/~ronnyk

Abstract

Irrelevant features and weakly relevant features may reduce the comprehensibility and accuracy of concepts induced by supervised learning algorithms. We formulate the search for a feature subset as an abstract search problem with probabilistic estimates. Searching a space using an evaluation function that is a random variable requires trading off accuracy of estimates for increased state exploration. We show how recent feature subset selection algorithms in the machine learning literature fit into this search problem as simple hill climbing approaches, and conduct a small experiment using a best-first search technique.

1 Introduction

Practical algorithms in supervised machine learning degrade in performance (prediction accuracy) when faced with many features that are not necessary for predicting the desired output. An important question in the field of machine learning, statistics, and pattern recognition, is how to select a good subset set of features.

From a theoretical standpoint, the question is not of much interest. A Bayes classifier is monotonic, *i.e.*, adding features cannot decrease performance, and hence restricting the induction algorithm to a subset of features is never advised. Practical algorithms, however, are not ideal, and the monotonicity assumption rarely holds. Notable exceptions that do satisfy monotonicity assumption are discriminant functions and distance measures such as the Bhattacharyya distance and divergence. For these functions branch and bound techniques can be used to prune the search space (Narendra & Fukunaga 1977).

Common machine learning algorithms, including top-down of decision tree algorithm, such as ID3 and C4.5 (Quinlan 1993), and instance based algorithms, such as IB3 (Aha, Kibler, & Albert 1991), are known to suffer from irrelevant features. For example, running C4.5 without special flags on the Monk 1 problem (Thrun etal. 1991), which has three irrelevant features, generates a tree with 15 interior nodes, five of which test irrelevant features. The generated tree has an error rate of 24.3%, which is reduced to 11.1% if only the three relevant features are given. Aha (1992) noted that "IB3's storage requirement increases exponentially with the number of irrelevant attributes."

Following the definitions in John, Kohavi, & Pfleger (1994), features can be divided into relevant and irrelevant. The relevant features can be further divided into strong and weak relevances (see Section 2 for the formal definitions). Irrelevant features are features that have no relation to the target concept; weakly relevant features have some bearing to the target concept, but are not essential; and strongly relevant features are indispensable. A good subset of features that would be used by an ideal classifier includes all the strongly relevant features, none of the irrelevant features, and a subset of the weakly relevant features.

In the next section we give the basic definitions for the rest of the paper. In Section 3, we describe the wrapper model. In Section 4, we abstract the subset selection into a search problem in which the evaluation function is probabilistic. In Section 5, we show how some recent suggestions for feature selection fit into the search framework. Section 6 describes a small experiment using best-first search instead of hill-climbing, and Section 7 concludes with a summary and future work.

2 Definitions

The following definitions closely follow those defined in John, Kohavi, & Pfleger (1994). 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, a neural net, or simply a list of instances), such that given a new instance, it is possible to accurately predict the label Y. We assume a probability measure p on the space $F_1 \times F_2 \times \cdots \times F_m \times Y$.

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 1 (Strong relevance)
X_i is strongly 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

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

Definition 2 (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

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

A feature is **relevant** if it is either weakly relevant or strongly relevant. A feature is **irrelevant** if it is not relevant.

3 The Wrapper Model

A good subset of features for an inductive learning algorithm should include a subset of the relevant features that optimizes some performance function, usually prediction accuracy.

The pattern recognition literature (Devijver & Kittler 1982), statistics literature (Miller 1990; Neter, Wasserman, & Kutner 1990), and recent machine learning papers (Almuallim & Dietterich 1991; Kira & Rendell 1992; Kononenko 1994) consist of many such measures that are all based on the data alone. Most measures in the pattern recognition and statistics literature are monotonic, *i.e.*, for a sequence of nested feature subsets $F_1 \supseteq F_2 \supseteq \cdots \supseteq F_k$, the measure f obeys $f(F_1) \geq f(F_2) \geq \cdots \geq f(F_k)$. Monotonic measures allow pruning the search space using a branch and bound algorithm, but most machine learning induction algorithms do not obey the monotonic restriction. Even when branch and bound can be used, the space is usually too big when there are more than 20 features, and suboptimal methods are used in practice.

All of the above measures and algorithms, however, ignore the fact that induction algorithms are not optimal, and that most induction algorithms conduct a very limited search in the space of possible structures. Ignoring these limitations can lead to feature subsets which are inappropriate for the induction algorithm used. As was shown in by John, Kohavi, & Pfleger (1994), even features with high predictive power may impair the overall accuracy in some cases. Selecting a subset of features must, therefore, not be based solely on the intrinsic discriminant properties of the data, but should be made relative to a given algorithm.

In the wrapper model, shown in Figure 1, the feature subset selection is done using the induction algorithm as a black box. The feature subset selection algorithm conducts a search for a good subset using the induction algorithm itself as part of the evaluation function.

In order to evaluate the prediction accuracy of the induced structure, k-fold cross validation (Breiman *et al.* 1984) can be used. The training data is split into k approximately equally sized partitions. The induction algorithm is then run k times, each time using $k-1$ partitions as the training set and the other partition as the test set. The accuracy results from each of the k runs are then averaged to produce the estimated accuracy.

4 Feature Subset Selection as Search

The problem of feature subset selection is basically a problem of state space search. Each state represents a subset of features, and the goal is to find the state with the best performance measure.

The wrapper model, which uses cross validation to estimate accuracy, complicates the search problem further. The fact that k-fold cross validation returns an estimate that is a random variable for $k < n$, implies that there is uncertainty in the returned value.

One way to decrease the variance is to run k-fold cross validation more than once and average the results, shuffling the data before each k-fold cross validation run. Averaging the results will yield a mean, such that the variance of the mean depends on the number of iterations conducted. Increasing the number of iterations shrinks the confidence interval for the mean, but requires more time. The tradeoff between more accurate estimates and more extensive exploration of the search space leads to the following abstract search problem.

Search with Probabilistic Estimates Let $\mathcal{S}$ be a state space with operators between states. Let $f : \mathcal{S} \mapsto \mathbb{R}$ be an unbiased probabilistic evaluation function that maps a state to a real number, indicating how good the state is. The number returned by $f(s)$ comes from a distribution $\mathcal{D}(s)$ with mean $f^*(s)$, which is the actual (unknown) value of the state. The goal is to find the state s with the maximal value of $f^*(s)$.

The mapping to the feature subset selection problem is as follows. The states are the subsets, and the operators are "add one feature," "delete one feature," *etc.* The evaluation function is the cross validation accuracy.[1]

Searching in the space of feature subsets has been studied for many years. Sequential backward elimination, sometimes called sequential backward selection, was introduced by Marill & Green in 1963. Kittler generalized the different variants including forward methods, stepwise methods, and "plus ℓ–take away

[1]Evaluation using cross validation is pessimistically biased due to the fact that only part of the data is used for training. The estimate from each fold is an unbiased estimator for that fold, which contains only $n \cdot (k-1)/k$ of the instances. For model selection, this pessimism is of minor importance.

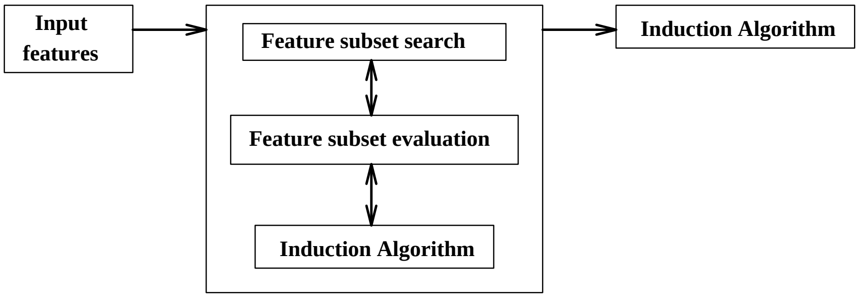

Figure 1: The **wrapper** model. The induction algorithm is used as a "black box" by the subset selection algorithm.

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). All the algorithms described above assume that the evaluation function is deterministic. When the evaluation function is a random variable, the search becomes more complicated.

Greiner (1992) describes how to conduct a hill-climbing search when the evaluation function is probabilistic. The algorithm stops at a node that is a local optimum with high probability. Yan & Mukai (1992) analyze an algorithm based on simulated annealing and show that it will find the global optimum if given enough time.

5 Instantiations of the Abstract Search Problem

In this section we look at three instantiations of the abstract search problem.

5.1 Hill climbing using the mean value

One simple approach used by John, Kohavi, & Pfleger (1994) is to do a k-fold cross validation and use the mean value as the estimate. This approach was used with forward stepwise selection and backward stepwise elimination.

Backward stepwise elimination is a hill-climbing approach that starts with the full set of features and greedily removes or adds one feature that improves performance, or degrades performance slightly. Forward stepwise selection is a similar algorithm that starts from the empty set of features.

The main disadvantage of this algorithm is that it does not take into account the uncertainty in the estimated accuracy. In the empirical observations it was noted that the values returned from the cross validation estimates had a large variance. This variance causes the algorithm to stop prematurely both during forward stepwise selection and during backward stepwise elimination.

5.2 Hoeffding races

Maron & Moore (1994) in an approach very similar to Greiner (1992), attempt to shrink the confidence interval of the accuracy for a given set models, until one model can be proven to be optimal with high probability. The evaluation function is a single step in leave-one-out cross validation, *i.e.*, the algorithm is trained on randomly chosen $n-1$ instances and tested on the one that is left.

The idea in the above paper is to *race* competing models, until one is a clear winner. Models drop out of the race when the confidence interval of the accuracy does not overlap with the confidence interval of the accuracy of the best model (this is analogous to imposing a higher and lower bound on the estimation function in the B^* algorithm). The race ends when there is a winner, or when all n steps in the leave-one-out cross validation have been executed. The confidence interval is defined according to Hoeffding's formula (Hoeffding 1963):

$$\Pr\left(\left|f^*(s) - \widehat{f}(s)\right| > \epsilon\right) < 2e^{-2m\epsilon^2/B^2}$$

where $\widehat{f}(s)$ is the average of m evaluations and B bounds the possible spread of point values. Given a confidence level, one can determine ϵ, and hence a confidence interval for $f^*(s)$, from the above formula.

The paper, however, does not discuss any search heuristic, and assumes that a fixed set of models is given by some external source.

5.3 Hill climbing utilizing races

Moore & Lee (1994) describe an algorithm for feature subset selection that has both ingredients of the abstract problem—it has a search heuristic, and it uses the probabilistic estimates in a non-trivial manner.

The algorithm does a forward selection and backward elimination (see Section 5.1), but instead of estimating the accuracy of each added (deleted) feature using leave-one-out cross validation, all the features that can be added (deleted) are raced in parallel. Assuming that the distribution of $f(s)$ is normal, confidence intervals are used to eliminate some features from the race.

Dataset	Baseline Acc.	C4.5 Acc.	C4.5-HCS Acc.	C4.5-BFS Acc.	BFS Cpu time (sec)	Subset
Breast-cancer	73.7%	74.7%	73.7%	74.7%	873	1, 4, 6, 8
Chess	53.2%	99.5%	93.9%	97.4%	27289	0,5,9, 13,14, 20, 31, 32, 34
Glass	30.6%	63.9%	61.1%	62.5%	937	0, 1, 2, 3
Glass2	58.2%	72.7%	80.0%	80.0%	515	0, 1, 3, 7
Heart-disease	62.4%	74.3%	79.2%	79.2%	787	8, 11, 12
Hepatitis	86.5%	80.8%	82.7%	84.6%	2100	4, 6, 16, 17
Horse-colic	60.3%	80.9%	85.3%	85.3%	2073	0, 2, 9, 21
Hypothyroid	94.8%	99.2%	99.2%	99.2%	10826	13, 14, 22
Iris	30.0%	94.0%	92.0%	92.0%	68	3
Labor	64.7%	82.4%	82.4%	82.4%	401	1, 10
Lymphography	60.0%	76.0%	78.0%	78.0%	923	0, 8, 12, 16
Mushroom	31.1%	100.0%	100.0%	100.0%	19937	4,7,11,14,19
Sick-euthyroid	90.4%	97.7%	97.8%	97.8%	13125	9,14,16,22
Soybean-small	31.1%	100.0%	100.0%	100.0%	937	20, 21
Vote	64.8%	95.2%	95.2%	95.2%	534	3
Vote1	64.8%	88.3%	89.7%	89.7%	751	2, 3, 5
Average	59.78%	86.2%	86.9%	87.4%		

Table 1: Comparison of C4.5 and C4.5 with best-FS feature subset selection.

Schemata search is another search variant that allows taking into account interactions between features. Instead of starting with the empty or full set of features, the search begins with the unknown set of features. Each time a feature is chosen and raced between being "in" or "out." All combinations of unknown features are used in equal probability, thus a feature that should be "in" will win the race, even if correlated with another feature.

Although this method uses the probabilistic estimates in a Bayesian setting, the basic search strategy is simple hill-climbing.

6 Experimental Results

In order to estimate the utility of broadening the search, we used a best-first search in the space of feature subsets. The initial node was the empty set of features and the evaluation function was a single 10-fold CV.[2] At each expansion step, best-first search chooses to expand an unexpanded node with the highest estimated accuracy. The search stops when five node expansions do not yield improved performance of more than 0.1%.

The datasets shown in Table 1 are the same ones used in Holte's paper (Holte 1993) from the UC Irvine repository (Murphy & Aha 1994). For all datasets that did not have a test set, we generated an independent sample of one third of the instances for testing.[3]

The table shows the baseline accuracy, *i.e.*, a majority predictor; C4.5's accuracy; C4.5's accuracy for the subset selected by a hill-climbing search (C4.5-HCS); C4.5's accuracy for the subset selected by a best-first search (C4.5-BFS); the CPU time for C4.5-BFS on a Sparc 10 512; and the subset selected by the best-first search (feature numbers starting from zero).

The results show that the hill-climbing search for a good subset improve C4.5's average accuracy by 0.7%, and that the best-first search strategy improves it by 1.2%. By themselves, these improvements may not seem significant, but it is well known that it is very hard to improve on C4.5's performance, and in some cases (*e.g.*, glass2, heart-disease, horse-colic), the improvements are substantial.

On some artificial datasets, we have seen more dramatic examples of the improvement of a good search strategy. For example, on the monk1 dataset (Thrun etal. 1991), C4.5's accuracy is 75.7%, C4.5-HCS's accuracy is 75.0%, and C4.5-BFS's accuracy is 88.9%. On the Corral dataset (John, Kohavi, & Pfleger 1994), C4.5's accuracy is 81.2%, C4.5-HCS's accuracy is 75%, and C4.5-BFS's accuracy is 100.0%.

7 Summary and Future Work

We have abstracted the feature subset selection using cross validation into a search problem with a probabilistic evaluation function. We have shown (Section 5) how three different instantiations of the abstract algorithm differ in their treatment of the evaluation function and search. While one algorithm ignores the fact that the evaluation is probabilistic and uses the mean value of a series of evaluations (k-fold cross validation), the other two use confidence intervals to aid in finding the best state (subset) fast. The two search algorithms examined are basic hill-climbing algorithms.

Preliminary experiments using best-first search and

[2] The same 10-way split was done for all subsets.

[3] Note that the accuracies are from a single randomly selected test set, not averaging over multiple runs as was done by Holte.

simple 10-fold cross validation for evaluation, show that broadening the search may indeed help. A more extensive experiment utilizing the fact that the evaluation function is probabilistic is now being conducted.

The algorithms discussed attempted to improve prediction accuracy. In many cases comprehensibility is very important, even when resulting in a small loss of accuracy. Biasing the algorithms towards smaller subsets may be important in such cases.

The search for a good subset is conducted in a very large space. All algorithms mentioned in this paper start the search either from the empty set of features, or from the full set of features. Since an optimal classifier should include all strongly relevant features, it might be beneficial to estimate which features are strongly relevant, and start the search from this subset.

Acknowledgments We wish to thank George John and Karl Pfleger for our initial work on feature subset selection. George John has been of great help in this ongoing research. We wish to thank Pat Langley for his comments. The experimental section was done using $\mathcal{MLC}$++, partly funded by ONR grant N00014-94-1-0448 and NSF grant IRI-9116399.

References

Aha, D. W.; Kibler, D.; and Albert, M. K. 1991. Instance-based learning algorithms. *Machine Learning* 6(1):37–66.

Aha, D. W. 1992. Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies* 36(1):267–287.

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

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

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

Greiner, R. 1992. Probabilistic hill climbing : Theory and applications. In Glasgow, J., and Hadley, R., eds., *Proceedings of the Ninth Canadian Conference on Artificial Intelligence*, 60–67. Morgan Kaufmann.

Hoeffding, W. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association* 58:13–30.

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

John, G.; Kohavi, R.; and Pfleger, K. 1994. Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the Eleventh International Conference*, 121–129. Morgan Kaufmann. Available by anonymous ftp from: `starry.Stanford.EDU:pub/ronnyk/ml94.ps`.

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

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

Maron, O., and Moore, A. W. 1994. Hoeffding races: Accelerating model selection search for classification and function approximation. In *Advances in Neural Information Processing Systems*, volume 6. Morgan Kaufmann.

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

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.

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.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning.* Los Altos, California: Morgan Kaufmann.

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

Thrun etal. 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.

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.

Yan, D., and Mukai, H. 1992. Stochastic discrete optimization. *Siam J. Control and Optimization* 30(3):594–612.