Appears in the First International Conference on Knowledge Discovery and Data Mining (KDD-95)

Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology

Ron Kohavi and Dan Sommerfield
Computer Science Department
Stanford University
Stanford, CA. 94305
{ronnyk,sommda}@CS.Stanford.EDU

Abstract

In the wrapper approach to feature subset selection, a search for an optimal set of features is made using the induction algorithm as a black box. The estimated future performance of the algorithm is the heuristic guiding the search. Statistical methods for feature subset selection including forward selection, backward elimination, and their stepwise variants can be viewed as simple hill-climbing techniques in the space of feature subsets. We utilize best-first search to find a good feature subset and discuss overfitting problems that may be associated with searching too many feature subsets. We introduce *compound operators* that dynamically change the topology of the search space to better utilize the information available from the evaluation of feature subsets. We show that compound operators unify previous approaches that deal with relevant and irrelevant features. The improved feature subset selection yields significant improvements for real-world datasets when using the ID3 and the Naive-Bayes induction algorithms.

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 fields of machine learning, knowledge discovery, statistics, and pattern recognition is how to select a good subset of features. The problem is especially severe when large databases, with many features, are searched for patterns without filtering of important features by human experts or when no such experts exist.

Common machine learning algorithms, including top-down induction of decision trees, such as CART, ID3, and C4.5 (Breiman, Friedman, Olshen & Stone 1984, Quinlan 1993), and nearest-neighbor algorithms, such as IB1, are known to suffer from irrelevant features. Naive-Bayes classifiers, which assume independence of features given the instance label, suffer from correlated and redundant features. A good choice of features may not only help improve performance accuracy, but also aid in finding smaller models for the data, resulting in better understanding and interpretation of the data.

In the filter approach to feature subset selection, a feature subset is selected as a preprocessing step where features are selected based on properties of the data itself and independent of the induction algorithm. In the wrapper approach, the feature subset selection is found using the induction algorithm as a black box. The feature subset selection algorithm conducts a search for a good feature subset using the induction algorithm itself as part of the evaluation function.

John, Kohavi & Pfleger (1994) used the wrapper method coupled with a hill-climbing search. Kohavi (1994) showed that best-first search improves the accuracy. One problem with expanding the search (*i.e.*, using best-first search and not hill-climbing) is that of overfitting: the accuracy estimation (cross-validation in both papers) guides the search toward feature subsets that will be good for the specific cross-validation folds; however, overusing the estimate can lead to overfitting, a problem we discuss in Section 4.

In the common organization of the state space search, each node represents a feature subset, and each operator represents the addition or deletion of a feature. The main problem with this organization is that the search must expand (*i.e.*, generate successors of) every node from the empty subset or from the full subset on the path to the best feature subset, which is very expensive. In Section 5 we introduce a way to change the search space topology by creating dynamic operators that directly connect to nodes considered promising given the evaluation of the children. These operators better utilize the information available in all the evaluated children. Our experimental results, shown in Sections 5 and 6, indicate that compound operators help identify better feature subsets faster and that feature subset selection can significantly improve the performance of induction algorithms.

2 Relevant and Optimal Features

The input to a supervised learning algorithm is a training set D of m labelled instances independently and identically distributed (i.i.d.) from an unknown distribution $\mathcal{D}$ over the labelled instance space. An unlabelled instance $\mathbf{X}$ is an element of the set $F_1 \times F_2 \times \cdots \times F_n$, where F_i is the domain of the ith feature. Labelled instances are tuples $\langle \mathbf{X}, Y \rangle$ where Y is the label, or output.

Let $\mathcal{I}$ be an induction algorithm using a hypothesis space $\mathcal{H}$; thus $\mathcal{I}$ maps D to $h \in \mathcal{H}$ and $h \in \mathcal{H}$ maps an unlabelled instance to a label. The prediction accuracy of a hypothesis h is the probability of correctly classifying the label of a randomly selected instance from the instance space according to the probability distribution $\mathcal{D}$. The task of the induction algorithm is to choose a hypothesis with the highest prediction accuracy.

We now define relevance of features in terms of a Bayes classifier—the optimal classifier for a given problem. A feature X is *strongly relevant* if removal of X alone will result in performance deterioration of an optimal Bayes classifier. A feature X is *weakly relevant* if it is not strongly relevant and there exists a subset of features, S, such that the performance of a Bayes classifier on S is worse than the performance on $S \cup \{f\}$. A feature is *irrelevant* if it is not strongly or weakly relevant. The set of strongly relevant features is called the *core*. Formalized versions of the above definitions can be found in John et al. (1994).

There are three main problems with these definitions that make them hard to use in practice. First, many hypothesis spaces are parametric (*e.g.*, perceptrons, monomials) and the best hypothesis approximating the target concept from the family may not even use all the strongly relevant features. Second, practical learning algorithms are not always consistent: even with an infinite amount of data they might not converge to the best hypothesis. Third, even consistent learning procedures may be improved for finite samples by ignoring relevant features. These reasons motivated us to define the optimal features, which depend not only on the data, but also on the specific induction algorithm.

An *optimal feature subset, $\mathcal{S}^*$*, for a given induction algorithm and a given training set is a subset of the features, $\mathcal{S}^*$, such that the induction algorithm generates a hypothesis with the highest prediction accuracy. The feature subset need not be unique.

The relation between relevant and optimal features is not obvious. In Section 5, we show how compound operators improve the search for optimal features using the ideas motivated by the above definitions of relevance.

3 Feature Subset Selection as Heuristic Search

The statistical and pattern recognition literature on feature subset selection dates back a few decades, but the research deals mostly with linear regression. We refer the reader to the related work section in John et al. (1994) for key references. Langley (1994) provides a survey of recent feature subset selection algorithms, mostly in machine learning.

Most criteria for feature subset selection from the statistics and pattern recognition communities are algorithm independent and do not take into account the differences between the different induction algorithms. For example, as was shown in John et al. (1994), features with high predictive power may impair the overall accuracy of the induced decision trees.

The task of finding a feature subset that satisfies a given criteria can be described as a state space search. Each state represents a feature subset with the given criteria used to evaluate it. Operators determine the partial ordering between the states.

In this paper, we use the *wrapper method* wherein the criteria to optimize is the estimated prediction accuracy. Methods that wrap around the induction algorithm, such as holdout, bootstrap, and cross-validation (Weiss & Kulikowski 1991) are used to estimate the prediction accuracy. To conduct a search, one needs to define the following:

Search Space Operators The operators in the search space are usually either "add feature" or "delete feature" or both. In the statistics literature, the term *forward selection* refers to a space containing only the "add feature" operator; the term *backward elimination* refers to a space containing only the "delete feature" operator. The stepwise methods use both operators. In our experiments, we used both operators.

Accuracy Estimation The heuristic function in the wrapper approach is the estimated prediction accuracy. In our experiments, we used ten-fold cross-validation as the accuracy estimation function.

Search Algorithm Any heuristic search algorithm can be used to conduct the search. In our experiments, we used best-first search, which at every iteration generates the successors of the the best unexpanded node (the node with the highest estimated accuracy). The termination condition was five consecutive non-improving nodes. The initial node determines the general direction of the search. One typically starts forward selection from the empty set of features and backward elimination from the full set of features.

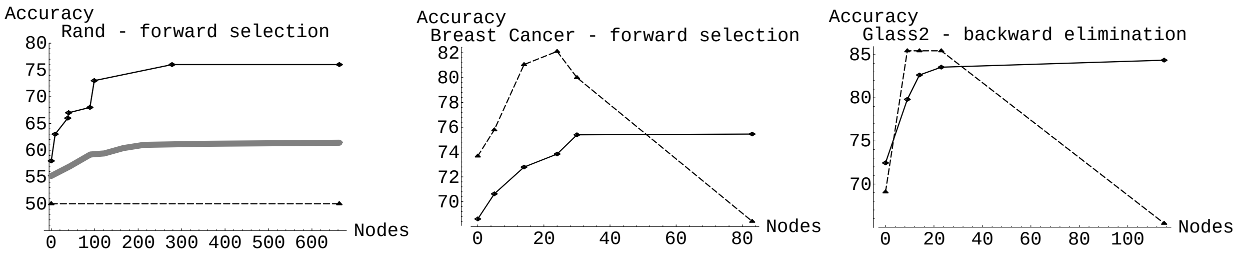

Figure 1: Overfitting in feature subset selection using ID3. The left graph shows accuracies for a random dataset. The solid line represents the estimated accuracy for a training set of 100 instances, the thick grey line for a training set of 500 instances, and the dotted line shows the real accuracy. The middle and right graphs show the accuracy for real-world datasets. The solid line is the estimated accuracy and the dotted line is the accuracy on an independent test set.

4 Overfitting

An induction algorithm *overfits* the dataset if it models the given data too well and its predictions are poor. An example of an over-specialized hypothesis, or classifier, is a lookup table on all the features. Overfitting is closely related to the bias-variance tradeoff (Geman & Bienenstock 1992, Breiman et al. 1984): if the algorithm fits the data too well, the variance term is large, and hence the overall error is increased.

Most accuracy estimation methods, including cross-validation, evaluate the predictive power of a given hypothesis over a feature subset by setting aside instances (holdout sets) that are not shown to the induction algorithm and using them to assess the predictive ability of the induced hypothesis. A search algorithm that explores a large portion of the space and that is guided by the accuracy estimates can choose a bad feature subset: a subset with a high accuracy estimate but poor predictive power.

If the search for the feature subset is viewed as part of the induction algorithm, then overuse of the accuracy estimates may cause overfitting in the feature-subset space. Because there are so many feature subsets, it is likely that one of them leads to a hypothesis that has high predictive accuracy for the holdout sets. A good example of overfitting can be shown using a *no-information* dataset (Rand) where the features and the label are completely random. Figure 1 (left) shows the estimated accuracy versus the true accuracy for the best node the search has found after expanding k nodes. One can see that especially for the small sample of size 100, the estimate is extremely poor (26% optimistic), indicative of overfitting. The middle and right graphs in the figure show overfitting in small real-world datasets.

Recently, a few machine learning researchers have reported the cross-validation estimates that were used to guide the search as a final estimate of performance, thus achieving overly optimistic results. Experiments using cross-validation to guide the search must report the accuracy of the selected feature subset on a *separate* test set or on holdout sets generated by an external loop of cross-validation that were never used during the feature subset selection process.

The problem of overfitting in feature subset space has been previously raised in the machine learning community by Wolpert (1992) and Schaffer (1993), and the subject has received much attention in the statistics community (cf. Miller (1990)).

Although the theoretical problem exists, our experiments indicate that overfitting is mainly a problem when the number of instances is small. For our experiments, we chose reasonably large datasets and our accuracies are estimated on unseen instances. In our reported experiments, there were 70 searches for feature subsets. Ten searches were optimistically biased by more than two standard deviations and one was pessimistically biased by more than two standard deviations.

5 Compound Operators

In this section we introduce *compound operators*, a method that utilizes the accuracy estimation computed for the children of a node to change the topology of the search space.

The motivation for compound operators comes from Figure 2 that partitions the feature subsets into core features (strongly relevant), weakly relevant features, and irrelevant features. An optimal feature subset for a hypothesis space must be from the relevant feature subset (strongly and weakly relevant features). A backward elimination search starting from the full set of features (as depicted in Figure 2) that removes one feature at a time, will have to expand all the children of each node before removing a single feature. If there are i irrelevant features and f features, $(i \cdot f)$ nodes must be evaluated. In domains where feature subset selection might be most useful, there are many features

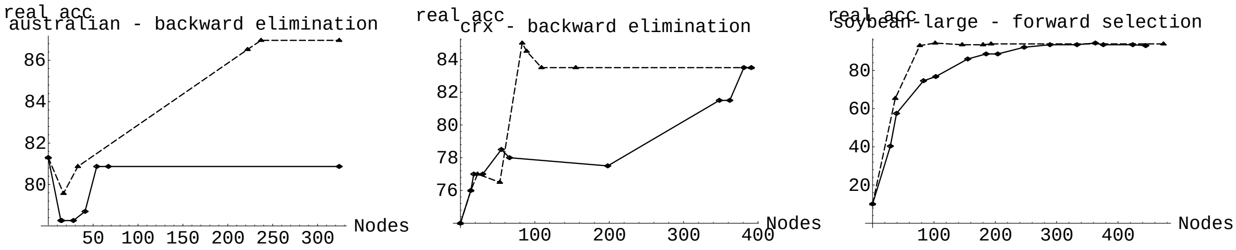

Figure 3: Comparison of compound (dotted line) and non-compound (solid line) searches. The accuracy (y-axis) is that of the best node on an independent test set after a given number of node evaluations (x-axis).

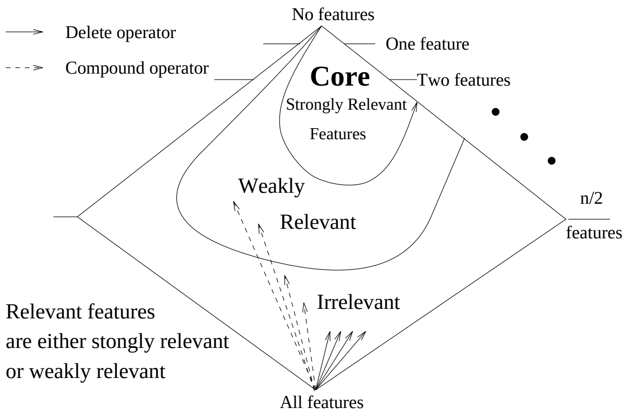

Figure 2: The state space. If a feature subset contains an irrelevant feature, it is in the irrelevant area; if it contains only strongly relevant features it is in the core region; otherwise, it is in the relevant region. The dotted arrows indicate compound operators.

but such a search may be prohibitively expensive.

Compound operators are operators that are dynamically created *after* the standard set of children, created by the add and delete operators, have been evaluated. Intuitively, there is more information in the evaluation of the children than just the node with the maximum evaluation. Compound operators combine operators that led to the best children into a single dynamic operator. If we rank the operators by the estimated accuracy of the children, then we can define compound operator c_i to be the combination of the best $i+1$ operators. For example, the first compound operator will combine the best two operators.

The compound operators are applied to the parent, thus creating children nodes that are farther away in the state space. Each *compound node* is evaluated and the generation of compound operators continues as long as the estimated accuracy of the compound nodes improves.

Compound operators generalize a few suggestions previously made. Kohavi (1994) suggested that the search might start from the set of strongly relevant features (the core). If one starts from the full set of features, removal of any single strongly relevant feature will cause a degradation in performance, while removal of any irrelevant or weakly relevant feature will not. Since the last compound operator connects the full feature subset to the core, the compound operators from the full feature subset plot a path leading to the core. The path is explored by removing one feature at a time until estimated accuracy deteriorates. Caruana & Freitag (1994) implemented a SLASH version of feature subset selection that eliminates the features not used in the derived decision tree. If there are no features that improve the performance when deleted, then (ignoring orderings due to ties) one of the compound operators will lead to the same node that slash would take the search to. While the SLASH approach is only applicable for backward elimination, compound operators are also applicable to forward selection.

In order to compare the performance of the feature subset selection algorithm with and without compound nodes, we ran experiments comparing them on different datasets. Figure 3 compares a search with and without compound operators. Compound operators improve the search by finding nodes with higher accuracy faster; however, whenever it is easy to overfit, they cause overfitting earlier.

6 Experimental Results

In order to compare the feature subset selection, we used ID3 and Naive-Bayes, both implemented in $\mathcal{MLC}$++ (Kohavi, John, Long, Manley & Pfleger 1994). The ID3 version does no pruning by itself; pruning is thus achieved by the feature subset selection mechanism. The Naive-Bayes algorithm assumes the features are independent given the instance label. The use of feature subset selection in Naive-Bayes was first suggested in Langley & Sage (1994). The data for Naive-Bayes was discretized using the discretization algorithm presented in Fayyad & Irani (1993) and implemented in $\mathcal{MLC}$++.

Dataset	Features	Train sizes	Test	Majority Accuracy	Dataset	Features	Train sizes	Test	Majority Accuracy
anneal	24	898	CV-5	76.17±1.4	australian	14	690	CV-5	55.51±1.9
breast (L)	9	286	CV-5	70.28±2.7	breast (W)	10	699	CV-5	65.52±1.8
chess	36	2130	1066	52.22±0.9	cleve	13	303	CV-5	54.46±2.9
crx	15	690	CV-5	55.51±1.9	DNA	180	3186	2000	51.91±0.9
heart	13	270	CV-5	55.56±3.0	horse-colic	22	368	CV-5	63.04±2.5
hypothyroid	25	2108	1055	95.23±0.4	mushroom	22	5416	2708	51.80±0.6
pima	8	768	CV-5	65.10±1.7	sick-euthyroid	25	2108	1055	90.74±0.5
soybean-lrg	35	683	CV-5	13.47±1.3	vehicle	18	846	CV-5	25.77±1.5
vote	16	435	CV-5	61.38±2.3	vote1	15	435	CV-5	61.38±2.3

Table 1: Datasets and baseline accuracy (majority). CV-5 indicates accuracy estimation by 5-fold cross-validation. The number after the ± denotes one standard deviation of the accuracy.

Dataset	ID3	ID3-FSS	p-val	C4.5	Naive-Bayes	NB-FSS	p-val
anneal	99.55±0.2	99.33±0.2	0.23	91.65±1.6	97.66±0.4	96.66±1.0	0.18
australian	80.43±1.0	85.94±1.7	1.00	85.36±0.7	86.09±1.1	85.90±1.6	0.47
breast (L)	68.20±2.9	73.43±2.3	0.92	71.00±2.3	70.99±2.3	70.63±2.1	0.45
breast (W)	94.42±0.8	94.28±0.8	0.45	94.71±0.4	97.14±0.5	96.57±0.4	0.19
chess	98.69±0.3	98.87±0.3	0.65	99.50±0.3	87.15±1.0	94.28±0.7	1.00
cleve	71.99±3.2	77.87±2.0	0.94	73.62±2.3	82.87±3.1	83.20±2.6	0.53
crx	79.86±1.7	84.35±1.6	0.97	85.80±1.0	86.96±1.2	85.07±0.8	0.08
DNA	90.39±0.9	92.50±0.8	0.97	92.70±0.8	93.34±0.7	93.42±0.7	0.53
heart	72.22±3.0	81.48±2.8	0.99	77.04±2.8	81.48±3.3	84.07±2.0	0.75
horse-colic	75.32±3.8	84.79±2.0	0.99	84.78±1.3	80.96±2.5	83.70±1.2	0.84
hypothyroid	98.58±0.4	98.77±0.3	0.65	99.20±0.3	98.58±0.4	99.24±0.3	0.93
mushroom	100.00±0.0	100.00±0.0	0.50	100.00±0.0	96.60±0.3	99.70±0.1	1.00
pima	71.75±2.1	68.36±3.0	0.18	72.65±1.8	75.51±1.6	73.56±2.2	0.24
sick-euth	96.49±0.6	95.83±0.6	0.22	97.70±0.5	95.64±0.6	97.35±0.5	0.98
soybean-lrg	91.94±1.0	93.27±1.3	0.80	88.28±2.0	91.36±2.0	93.41±0.8	0.83
vehicle	73.76±2.0	69.86±0.9	0.04	69.86±1.8	59.22±1.6	61.23±1.3	0.84
vote	94.02±0.4	95.63±0.8	0.97	95.63±0.4	90.34±0.9	94.71±0.6	1.00
vote1	84.60±1.2	86.44±1.2	0.87	86.67±1.1	87.36±2.1	90.80±2.0	0.88
Average	85.68	87.83		87.01	86.63	87.97	

Table 2: The accuracies for ID3, ID3 with feature subset selection (FSS), C4.5, Naive-Bayes, and Naive-Bayes with FSS. The numbers after the ± indicate the standard deviation of the reported accuracy. The first p-val column indicates the probability that FSS improves ID3 and the second column indicates the probability that FSS improves Naive-Bayes. The p-values were computed using a one-tailed t-test.

Because small datasets are easier to overfit using our approach, we chose real-world datasets from the U.C. Irvine repository (Murphy & Aha 1994) that had at least 250 instances. For datasets with over 1000 instances, a separate test set with one-third of the instances was used; for datasets with fewer than 1000 instances, 5-fold cross-validation was used. Table 1 describes general information about the datasets used.

The initial node for our search was the empty set of features mainly because the search progresses faster and because in real-world domains one would expect many features to be irrelevant or weakly relevant. The best-first search is able to overcome small local maxima caused by interacting features, whereas hill-climbing cannot.

Table 2 shows that feature subset selection significantly (over 90% confidence) improves ID3 on eight out of the eighteen domains and significantly degrades the performance only on one domain. Performance of Naive-Bayes significantly improves on five domains and significantly degrades on one domain. The average error rate for the datasets tested decreased (relatively) by 15% for ID3 and by 10% for Naive-Bayes. Both ID3 and Naive-Bayes were inferior to C4.5, but both outperformed C4.5 after feature subset selection.

A similar experiment (not shown) with C4.5 showed that C4.5 with feature subset selection slightly improved C4.5: the average accuracy went up from 87.01% to 87.60%, a 4.5% reduction in error.

The execution time on a Sparc20 for feature subset selection using ID3 ranged from under five minutes for breast-cancer (Wisconsin), cleve, heart, and vote to about an hour for most datasets. DNA took 29 hours, followed by chess at four hours. The DNA run took so long because of ever increasing estimates that did not really improve the test-set accuracy.

7 Conclusions

We reviewed the wrapper method and discussed the problem of overfitting when the search through the state space is enlarged through the use of best-first search. While overfitting can occur, the problem is less severe for large datasets, so we have restricted our experiments to such datasets. One possible way to deal with overfitting is to reevaluate the best nodes using different cross-validation folds (*i.e.*, shuffle the data). Initial experiments indicate that re-evaluation of the best nodes indeed leads to lower estimates for those nodes, partially overcoming the overfitting problem.

We introduced compound operators that change the search topology based on information available from the evaluation of children nodes. The approach generalizes previous suggestions and was shown to speed up discovery of good feature subsets. Our results indicated significant improvement both for ID3 and Naive-Bayes and some improvement for C4.5. The average error rate for the datasets tested decreased (relatively) by 15% for ID3, by 10% for Naive-Bayes, and by 4.5% for C4.5.

An issue that has not been addressed in the literature is whether we can determine a better starting point for the search. For example, one might start with the feature subset used by a learning algorithm when the subset is easy to identify, such as when using decision trees.

Acknowledgments The work in this paper was done using the $\mathcal{MLC}$++ library, partly funded by ONR grant N00014-94-1-0448 and NSF grants IRI-9116399 and IRI-9411306. We thank George John and Pat Langley for their comments. The reviewers comments were excellent, but many are not addressed due to lack of space.

References

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

Caruana, R. & Freitag, D. (1994), Greedy attribute selection, *in* W. W. Cohen & H. Hirsh, eds, "Machine Learning: Proceedings of the Eleventh International Conference", Morgan Kaufmann.

Fayyad, U. M. & Irani, K. B. (1993), Multi-interval discretization of continuous-valued attributes for classification learning, *in* "Proceedings of the 13th International Joint Conference on Artificial Intelligence", Morgan Kaufmann, pp. 1022–1027.

Geman, S. & Bienenstock, E. (1992), "Neural networks and the bias/variance dilemma", *Neural Computation* **4**, 1–48.

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

Kohavi, R. (1994), Feature subset selection as search with probabilistic estimates, *in* "AAAI Fall Symposium on Relevance", pp. 122–126.

Kohavi, R., John, G., Long, R., Manley, D. & Pfleger, K. (1994), MLC++: A machine learning library in C++, *in* "Tools with Artificial Intelligence", IEEE Computer Society Press, pp. 740–743. Available by anonymous ftp from: `starry.Stanford.EDU:pub/ronnyk/mlc/toolsmlc.ps`.

Langley, P. (1994), Selection of relevant features in machine learning, *in* "AAAI Fall Symposium on Relevance", pp. 140–144.

Langley, P. & Sage, S. (1994), Induction of selective bayesian classifiers, *in* "Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence", Morgan Kaufmann, Seattle, WA, pp. 399–406.

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

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

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

Schaffer, C. (1993), "Selecting a classification method by cross-validation", *Machine Learning* **13**(1), 135–143.

Weiss, S. M. & Kulikowski, C. A. (1991), *Computer Systems that Learn*, Morgan Kaufmann, San Mateo, CA.

Wolpert, D. H. (1992), "On the connection between in-sample testing and generalization error", *Complex Systems* **6**, 47–94.