FEATURE SELECTION VIA THE DISCOVERY OF SIMPLE CLASSIFICATION RULES

G. Holmes & C.G. Nevill-Manning
Department of Computer Science
University of Waikato, Hamilton, New Zealand

ABSTRACT

It has been our experience that in order to obtain useful results using supervised learning of real-world datasets it is necessary to perform feature subset selection and to perform many experiments using computed aggregates from the most relevant features. It is, therefore, important to look for selection algorithms that work quickly and accurately so that these experiments can be performed in a reasonable length of time, preferably interactively. This paper suggests a method to achieve this using a very simple algorithm that gives good performance across different supervised learning schemes and when compared to one of the most common methods for feature subset selection.

KEYWORDS

Feature subset selection; supervised learning; 1R; filter model; wrapper model.

INTRODUCTION

There is growing evidence that feature subset selection can substantially improve the task of performing supervised learning. The algorithms that perform feature subset selection have been studied in a variety of fields from the 1960’s and have employed many different techniques ranging from genetic algorithms (Kelly & Davis, 1991) to conditional probabilities (Creecy et al, 1992). Each algorithm can, however, be characterised in terms of its connection with the induction algorithm used to perform the supervised learning. If the input features are selected prior to induction then the feature selection algorithm is said to employ a *filter* model. If, on the other hand, the induction algorithm is bound to the process of searching, evaluating and selecting features then it is said to employ a *wrapper* model; the feature selection algorithm exists as a wrapper around the induction algorithm (John et al, 1995).

Feature subset selection is generally achieved against some form of objective function. In our case we choose classification accuracy as an objective function; our goal being to improve (or not dramatically reduce) classification accuracy while reducing the number of features in the original dataset.

The objective function is used by a search strategy to find the “best” subset. If there are d features then the size of the search space of all possible features is 2^d. It is not practical to exhaustively search this space and so some form of hill-climbing or optimization technique is used to guide the search. Subsets found using non-exhaustive search strategies do not guarantee to find optimal solutions, and that is the sense in which “best” subsets are found. It is the search strategy that accounts for the cost of performing feature subset selection. This cost and the accuracy of the resulting subset of features are useful measures for comparing the performance of different algorithms.

The approach we use is an extension of Robert Holte's 1R system (Holte, 1992), and can be used as either a filter or wrapped around an induction algorithm. The 1R system was originally written to demonstrate the “weakness” of some of the standard datasets (specifically those in the repository at the University of California, Irvine) used to test new induction algorithms. 1R builds rules based on a single feature (called 1-rules) for each feature in a dataset. By splitting the dataset into training and test sets it is possible to compute a classification accuracy score for each feature. Holte selects the highest scoring feature and shows that for most of the UCI datasets the rule associated with this single feature performs comparably with state-of-the-art techniques in machine learning.

In theory then, 1R could be viewed as an extremely powerful filter, reducing all datasets to one feature. This view, however, is not likely to enhance the performance of supervised learning schemes that require a search space of greater complexity to work through. 1R can be used to select those features that contribute, in the simplest sense (i.e. those with low error rates), to classification accuracy. Holte uses 1R as a classifier, comparing it with the likes of C4.5. We view it as a feature selector that can be used to enhance the performance of programs like C4.5.

Our hypothesis is that classification accuracies of individual features are good indicators of feature relevance. We run 1R over a dataset forming 1-rules for each feature acquiring, at the same time, a ranked list of features based on classification accuracy scores. Feature subset selection can then be achieved by either selecting a pre-determined number of features from the list or by iteratively adding the best features from the list to an initially empty set and evaluating each set using an induction algorithm.

In this paper we investigate two questions related to the use of 1R as a feature subset selection algorithm. Firstly, can 1R be used successfully with a variety of machine learning schemes? Secondly, how does 1R compare as a feature subset selection algorithm against more expensive algorithms?

In the sections that follow we describe Holte’s 1R system, our experimental methodology designed to answer the questions posed above, the results of those experiments, and finally, we discuss the implications of those results.

HOLTE’S 1-RULE INDUCER

Here we present a shortened version of the 1R algorithm and a worked example to illustrate how it can be used for feature selection. A fuller description of the 1R algorithm can be found in Holte (1992).

Algorithm

```
For each feature $f$,
    For each value $v$ from the domain of $f$
        Select the set of instances where feature $f$ has value $v$
        Let $c$ = the most frequent class in that set.
        Add the clause “if feature $f$ has value $v$ then the class is $c$” to the rule for feature $f$
Output the rule with the highest classification accuracy.
```

Note that numeric features are quantised using a simple heuristic, and that missing values are treated as a separate value in the enumeration.

Example

Figure 1a is a two feature version of Quinlan’s golf dataset (Quinlan, 1992), which uses weather information to decide whether or not to play golf. The dataset has two nominal features: *outlook* (with values *sunny*, *overcast* or *rain*) and *windy* (with values *true* or *false*). The classification of each instance is either *play* or *don’t play*.

Figure 1b shows the number of times that each class occurs for each feature-value pair, and highlights the most common class in grey. Note that for windy = true, an arbitrary decision must be made because both classes occur with the same frequency. Figure 1c shows the rules derived from these tables—the predicted classes correspond to the highlighted classes in 1b. Summing the frequencies of the predicted classes for each feature-value gives the number of correct predictions in the training set, and dividing by the number of instances gives the memorisation accuracy (Kohavi et al, 1994). The rule for feature *outlook* has a memorisation accuracy of 71.4%, while the rule for feature *windy* has an accuracy of 64.3%. Sorting the features by the accuracy of their rules gives the ranking which we use for feature selection.

(a)

rain	false	play
rain	true	don't play
overcast	false	play
sunny	false	play
rain	false	play
sunny	false	don't play
overcast	true	play
sunny	true	don't play
sunny	true	play
overcast	false	play
sunny	false	don't play
overcast	true	play
rain	true	don't play
rain	false	play

(b)

outlook	play	dont play
overcast	4	0
sunny	2	3
rain	3	2

windy	play	dont play
true	3	3
false	6	2

(c)

outlook

if overcast then	play	(4/4)
if sunny then	don't play	(3/5)
if rain then	play	(3/5)

Accuracy = 10/14 (71.4%)

windy

if true then	don't play	(3/6)
if false then	play	(6/8)

Accuracy = 9/14 (64.3%)

Figure 1: (a) a two-feature version of the golf dataset, (b) a table of frequencies of each class for each value of the two features, (c) the rules derived from 1b, and their accuracy.

EXPERIMENTS

We designed two experiments to see if 1R was useful as a feature selection algorithm. The first experiment attempts to see if 1R can select relevant features across a variety of different supervised learning algorithms. The second compares 1R with a wrapper model which employs a popular search algorithm called forward sequential selection (FSS) and C4.5 as the inducer.

Experiment 1

We chose three different induction algorithms: FOIL (Quinlan, 1990) which learns first order logical predicates, IB1 an instance based learning algorithm (Aha et al, 1991) and two variants of C4.5 (Quinlan, 1992)—pruned and unpruned. The two variants were chosen to observe the difference in performance when C4.5 uses its pruning algorithm, which is itself a form of feature selection. Default settings of all parameters were used for these schemes. Thirteen datasets from the UCI repository were tested in the following manner (Holte, 1993).

New datasets were created which included only the best feature (1-set), the top two features (2-set) and so on, using the ranking given by 1R.

Each dataset was randomly split into a training set (2/3 of the data) and a test set. The machine learning scheme was trained on the training set, and then its accuracy was measured on the test set. The splitting, training and testing was repeated 25 times, and the accuracy was averaged.

	Full	**1-set**	**2-set**	**3-set**	**4-set**	**5-set**	**8-set**	**11-set**	**15-set**	**Default**
BC	66.1	58.8	58.8	64.6	65	66.1	67.8	66.1	66.1	66.1
CH	89.6	59.1	67.1	87	87.7	87.7	90.5	93.7	95.3	94.1
G2	76.4	65.2	78.9	75.4	78.2	84.6	76.4	76.4	76.4	76.4
GL	67.8	45.3	60.2	71.3	71.6	72.1	73.3	67.8	67.8	67.8
HD	75.5	68.7	73	75.8	76.7	76.9	78.5	79.1	75.5	76.2
HE	80.8	76.7	79.2	79.5	77.5	80.3	77.6	79	81.4	80.8
HO	77.4	72.4	73.3	75.7	78.4	77.9	76.4	78.4	75	75.4
HY	97.7	96.7	96.8	97.5	97.5	97.6	97.6	97.8	97.9	97.7
IR	95.3	92.2	96.1	95.4	95.3	95.3	95.3	95.3	95.3	95.3
LA	84.2	73.5	81.2	79.8	88.8	88.8	87.8	85.5	85.5	84.6
SE	93.8	88.6	91.3	91.3	91.4	91.8	92.1	92	92.3	93.8
V1	87.3	81.1	86.4	87.3	87.6	87.7	87.3	86	87.3	88.5
VO	91.9	90.7	91.4	93.4	93.4	93.5	94.2	92.1	92.7	92.8
Avg diff		-8.83	-3.85	-0.75	0.41	1.27	0.85	0.42	0.36	0.44

Table 1: Accuracy of IB1 with varying numbers of features chosen by 1R

The results for IB1 are given in Table 1. *Full* means that all the features were present in the dataset, and *Default* means that all features which ranked above the default accuracy for the dataset were included. The *Avg diff* row computes the average difference between each column and the *Full* column. For example, the *average difference* figure 1.27 in the 5-set column means that averaged over all the datasets, using the top five features from a dataset results in an increase in accuracy of 1.27% over using all the features.

Scheme	**1set**	**2set**	**3set**	**4set**	**5set**	**8set**	**11set**	**15set**	**Default**
IB1	-8.8	-3.9	-0.8	0.4	1.3	0.9	0.4	0.4	0.4
FOIL	-15.3	-5.8	0.6	1.6	2.1	2.5	3.0	3.7	4.8
C4.5pruned	-3.5	-2.7	-0.4	-0.1	-0.5	-0.5	-0.3	-0.3	-0.1
C4.5 unpruned	-3.6	-1.7	0.3	0.4	-0.1	-0.3	-0.4	-0.5	-0.3
Average Overall	-7.8	-3.5	-0.0	0.6	0.7	0.7	0.7	0.8	1.2
% reduction of features	94.0	88.1	82.1	76.2	70.6	54.1	40.4	24.8	34.4

Table 2: The average difference results for each of the schemes

Discussion of Experiment 1

The results in Table 2 show encouraging signs for 1R as a feature selector. If we use 1R as a filter then we can say that, based on these preliminary findings, on average the best three or more features will be as accurate as using the full complement of features. These results are consistent with Holte’s conclusions about the UCI repository. Using three features minimises the number of features we need to use without any significant loss of accuracy. Thus, we could use 1R as a filter which simply selected the best three features and then passed these on to an induction algorithm. We show the results of using 1R this way in our next experiment.

Experiment 2

In order to compare our approach with one of the most common approaches to feature selection, we used the FSS algorithm built into the MLC++ system (Kohavi et al, 1994) to select features, and then ran similar experiments to the one above to obtain a comparison between FSS, 1R and choosing features randomly. We chose C4.5 (pruned) as the induction algorithm because this was the only algorithm which we could guarantee to be the same across MLC++ and our WEKA system (Holmes et al, 1994).

Dataset	**FSS** **O(d^2)**	**1R best** **O(d)**	**1R pre-det** **O(1)**	**Random** **O(1)**	**Default** **Accuracy**
BC	73.6	72.1	70.4	69.6	70.3
CH	97.6	97	90.3	58.9	52.2
G2	76.7	77	77	72.4	53.4
GL	69.9	68.2	65.9	58.8	35.5
HD	82.3	83	83	65.6	54.5
HE	81.8	84	80.3	78.1	79.3
HO	84.3	82.6	81.4	70.9	63.0
HY	99.3	99.3	98.9	96.7	95.2
IR	94.3	95.5	94.4	92.6	33.3
LA	83.8	82.3	76.8	63.8	64.9
SE	97.4	96.2	96	93.2	90.7
V1	89.4	89.4	88.4	84.7	61.4
VO	95.5	95.4	95.3	79.7	61.4
Average	86.6	86.3	84.5	75.8	62.7
Avg # features	3.9	5.8	3	3.9	

Table 3: Comparison of feature selection methods

The number of features selected randomly was chosen to be the same number as chosen by the FSS strategy. The 1R best strategy is a less expensive form of wrapper model. We use 1R to form a ranking of the features and then enumerate all sets of combinations as per experiment 1; namely, the best, the best plus next best, and so on. We then determine an average accuracy score, using an induction algorithm, for each of these sets and choose the best. This is a form of wrapper model without hill-climbing, hence the saving in search time.

Discussion of Experiment 2

The results in Table 3 again show encouraging signs for 1R. It is difficult to draw firm conclusions without a full statistical analysis, but it would appear to be true that on average 1R performs comparably with FSS both in wrapper and filter modes. The default accuracies of some of the datasets (HY, SE) are very high, and so only a fine distinction can be drawn from them, but others (CH, HD, LA, VO) show large differences between the default and random accuracies and the accuracies obtained using FSS and 1R.

The compromise between FSS and 1R best is interesting. FSS is an order of magnitude slower than 1R best but on average uses fewer features. For datasets with ten or fewer features this difference is not great when taken together. However, as the number of features increases the ability of FSS to complete its selection in a reasonable time tends to diminish. By way of an example, it took FSS approximately eight hours on a SparcServer 1000 to make its selection for the CH dataset (36 features).

CONCLUSION

We have presented an algorithm for feature subset selection which in preliminary experiments appears to work well both across differing supervised learning schemes and when compared with more common, and more expensive, approaches to the problem. In order to make firm conclusions about this approach, however, it will be necessary to conduct more experiments over a greater number of datasets and to perform a more rigorous statistical analysis of the results.

REFERENCES

Aha, D.W. and Bankert, R.L. (1994); A Comparative Evaluation of Sequential Feature Selection Algorithms; In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, pp 1-7. Fort Lauderdale, Florida.

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

Creecy, T.M., Masand, B.M., Smith, S.J., and Waltz, D.L. (1992); Trading MIPS and memory for knowledge engineering; *Communications of the ACM*, Vol. 35, pp 48-64.

Holmes, G., Donkin, A., and Witten, I.H. (1994); WEKA: A Machine Learning Workbench; *Tech Report 94/9*, Dept. Computer Science., Waikato University, New Zealand.

Holte, R. C. (1993); Very Simple Classification Rules Perform Well on Most Commonly Used Datasets; *Machine Learning*, Vol. 11, pp 63-90.

John, G.H., Kohavi, R., and Pfleger, K. (1994); Irrelevant Features and the Subset Selection Problem; In *Proceedings of the Eleventh International Machine Learning Conference*, pp 121-129. New Brunswick, NJ: Morgan Kaufmann.

Kelly, J.D. Jr., and Davis, L. (1991); A hybrid genetic algorithm for classification; In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pp 645-650, Sydney, Australia; Morgan Kaufmann.

Kohavi, R., John, G., Long, R., Manley, D., and Pfleger, K. (1994); MLC++: A Machine Learning Library in C++; *Tech Report*, Computer Science Dept., Stanford University.

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

Quinlan, J.R. (1990); Learning logical definitions from relations; *Machine Learning* Vol. 5, pp 239-266.