Fast Feature Ranking Algorithm*

Roberto Ruiz, José C. Riquelme, and Jesús S. Aguilar-Ruiz

Departamento de Lenguajes y Sistemas, Universidad de Sevilla.
Avda. Reina Mercedes S/N. 41012 Sevilla, España
{rruiz,riquelme,aguilar}@lsi.us.es

Abstract. The attribute selection techniques for supervised learning, used in the preprocessing phase to emphasize the most relevant attributes, allow making models of classification simpler and easy to understand. The algorithm has some interesting characteristics: lower computational cost (O(m n log n) m attributes and n examples in the data set) with respect to other typical algorithms due to the absence of distance and statistical calculations; its applicability to any labelled data set, that is to say, it can contain continuous and discrete variables, with no need for transformation. In order to test the relevance of the new feature selection algorithm, we compare the results induced by several classifiers before and after applying the feature selection algorithms.

1 Introduction

It is advisable to apply to the database preprocessing techniques to reduce the number of attributes or the number of examples in such a way as to decrease the computational time cost. These preprocessing techniques are fundamentally oriented to either of the next goals: feature selection (eliminating non-relevant attributes) and editing (reduction of the number of examples by eliminating some of them or calculating prototypes [1]). Our algorithm belongs to the first group.

Feature selection methods can be grouped into two categories from the point of view of a method's output. One category is about ranking feature according to same evaluation criterion; the other is about choosing a minimum set of features that satisfies an evaluation criterion. In this paper we present a new feature ranking algorithm by means of Projections and the hypothesis on which the heuristic is based is: "place the best attributes with the smallest number of label changes (NLC)".

2 Feature Evaluation

2.1 Description

To describe the algorithm we will use the well-known data set IRIS, because of the easy interpretation of their two-dimensional projections.

* This work has been supported by the Spanish Research Agency CICYT under grant TIC2001-1143-C03-02.

V. Palade, R.J. Howlett, and L.C. Jain (Eds.): KES 2003, LNAI 2773, pp. 325–331, 2003.

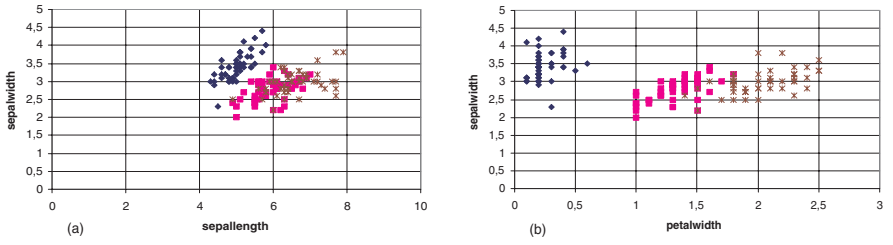

Fig. 1. Representation of Attributes (a) *Sepalwidth-Sepallength* and (b) *Sepalwidth-Petalwidth*

Three projections of IRIS have been made in two-dimensional graphs. In Figure 1(a) it is possible to observe that if the projection of the examples is made on the abscissas or ordinate axis we can not obtain intervals where any class is a majority, only can be seen the intervals [4.3,4.8] of Sepallength for the Setosa class or [7.1,8.0] for Virginica. In Figure 1(b) for the Sepalwidth parameter in the ordinate axis clear intervals are not appraised either. Nevertheless, for the Petalwidth attribute is possible to appreciate some intervals where the class is unique: [0,0.6] for Setosa, [1.0,1.3] for Versicolor and [1.8,2.5] for Virginica.

SOAP is based on this principle: to count the label changes, produced when crossing the projections of each example in each dimension. If the attributes are in ascending order according to the NLC, we will have a list that defines the priority of selection, from greater to smaller importance. Finally, to choose the more advisable number of features, we define a reduction factor, RF, in order to take the subset from attributes formed by the first of the aforementioned list.

Before formally exposing the algorithm, we will explain with more details the main idea. We considered the situation depicted in Figure 1(b): the projection of the examples on the abscissas axis produces an ordered sequence of intervals (some of then can be a single point) which have assigned a single label or a set of them: [0,0.6] Se, [1.0,1.3] Ve, [1.4,1.4] Ve-Vi, [1.5,1.5] Ve-Vi, [1.6,1.6] Ve-Vi, [1.7,1.7] Ve-Vi, [1.8,1.8] Ve-Vi, [1.9,2.5] Vi. If we apply the same idea with the projection on the ordinate axis, we calculate the partitions of the ordered sequences: Ve, R, R, Ve, R, R, R, R, R, R, R, R, R, Se, R, Se, R, Se, where R is a combination of two or three labels. We can observe that we obtain almost one subsequence of the same value with different classes for each value from the ordered projection. That is to say, projections on the ordinate axis provide much less information that on the abscissas axis.

In the intervals with multiple labels we will consider the worst case, that being the maximum number of label changes possible for a same value.

The number of label changes obtained by the algorithm in the projection of each dimension is: Petalwidth 16, Petallength 19, Sepallenth 87 and Sepalwidth 120. In this way, we can achieve a ranking with the best attributes from the point

Table 1. Main Algorithm

```
Input: E training (N examples, M attributes)
Output: E reduced (N examples, K attributes)
  for each attribute A_i ∈ 1..M
    QuickSort(E,i)
    NLC_i ← NumberChanges(E,i)
  NLC Attribute Ranking
  Select the k first
```

Table 2. NumberChanges function

```
Input: E training (N examples, M attributes), i
Output: number of label changes
 for each example e_j ∈ E with j in 1..N
    if att(u[j],i) ∈ subsequence of the same value
      changes = changes + ChangesSameValue()
    else
      if lab(u[j]) <> lastLabel)
        changes = changes + 1
 return(changes)
```

of view of the classification. This result agrees with what is common knowledge in data mining, which states that the width and length of petals are more important than those related to sepals.

2.2 Algorithm

The algorithm is very simple and fast, see Table 1. It has the capacity to operate with continuous and discrete variables as well as with databases which have two classes or multiple classes. In the ascending-order-task for each attribute, the QuickSort [5] algorithm is used. This algorithm is O(n log n), on average. Once ordered by an attribute, we can count the label changes throughout the ordered projected sequence. NumberChanges in Table 2, considers whether we deal with different values from an attribute, or with a subsequence of the same value (this situation can be originated in continuous and discrete variables). In the first case, it compares the present label with that of the following value. Whereas in the second case, where the subsequence is of the same value, it counts as many label changes as are possible (function ChangesSameValue).

After applying QuickSort, we might have repeated values with the same or different class. For this reason, the algorithm firstly sorts by value and, in

case of equality, it will look for the worst of the all possible cases (function ChangesSameValue).

We could find the situation as depicted in Figure 2(a). The examples sharing the same value for an attribute are ordered by class. The label changes obtained are two. The next execution of the algorithm may find another situation, with a different number of label changes. The solution to this problem consists of finding the worst case. The heuristic is applied to obtain the maximum number of label changes within the interval containing repeated values. In this way, the ChangesSameValue method would produce the output shown in Figure 2(b), seven changes. This can be obtained with low cost. It can be deduced counting the class' elements. ChangesSameValue stores the relative frequency for each class within the interval. It is possible to be affirm that:

$$if\ rfi\ >\ (nelem/2)\quad them\ (nelem - rfi) * 2\quad else\ nelem - 1 \tag{1}$$

rfi: relative frequency for each class, with i in {1,. . . ,k} classes.
nelem: number of elements within the interval.

In Figure 2(a) we can observe a subsequence of the same value with eight elements: three elements are class A, four class B and one C. Applying formula 2 there is no relative frequency greater than half of the elements. Then, the maximum number of label changes is nelem-1, seven. In Figure 2(b) we verify it.

Ranking algorithms produce a ranked list, according to the evaluation criterion applied. The methods need an external parameter to take the subset from attributes formed by the first features of the aforementioned list. This parameter produces different results with different data sets. Therefore, in order to establish the number of attributes in each case, we put the range of value of the ranked lists between [0,1], i.e. the punctuation of the first attribute of the list will be 1, and the last attribute 0. Then, we select attributes over the parameter named Reduction Factor (RF). We do not realize an especial analyzed on each data set.

3 Experiments

In order to compare the effectiveness of SOAP as a feature selector for common machine learning algorithms, experiments were performed using sixteen

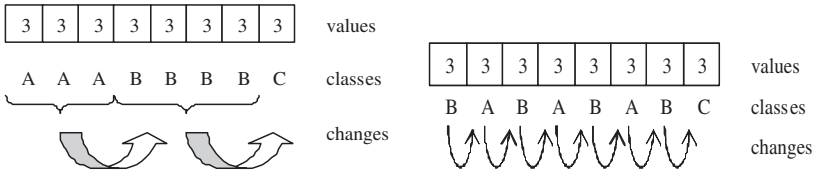

Fig. 2. Subsequence of the same value (a) two changes (b) seven changes

Table 3. Data sets, number of selected features, the percentage of the original features retained and time in milliseconds

DATA				SOAP			CFS			RLF		
Data Set	Inst.	Atts	N°Cl.	Atts	(%)	t-ms	Atts	(%)	t-ms	Atts	(%)	t-ms
autos	205	25	7	2.9	(11.8)	15	5.3	(21.3)	50	10.9	(43.7)	403
breast-c	286	9	2	1.5	(16.7)	4	4.1	(45.9)	6	3.7	(41.6)	174
breast-w	699	9	2	5.2	(57.6)	6	9.0	(99.7)	35	8.1	(89.4)	1670
diabetes	768	8	2	2.8	(34.9)	6	3.1	(38.9)	39	0.0	(0.0)	1779
glass2	163	9	2	3.2	(35.7)	2	4.0	(43.9)	9	0.3	(3.6)	96
heart-c	303	13	5	6.3	(48.2)	6	6.4	(49.1)	10	6.9	(53.4)	368
heart-st	270	13	2	5.4	(41.8)	4	6.3	(48.2)	12	6.3	(48.2)	365
hepatit.	155	19	2	2.6	(13.6)	4	8.7	(45.6)	9	13.3	(70.0)	135
horse-c.	368	27	2	2.3	(8.6)	16	2.0	(7.4)	43	2.3	(8.6)	941
hypothy.	3772	29	4	1.7	(5.7)	180	1.0	(3.4)	281	5.2	(18.0)	94991
iris	150	4	3	2.0	(50.0)	3	1.9	(48.3)	3	4.0	(100.0)	44
labor	57	16	2	4.3	(27.0)	1	3.3	(20.8)	3	8.8	(55.3)	21
lymph	148	18	4	1.8	(9.9)	3	8.9	(49.2)	7	11.8	(65.8)	109
sick	3772	29	2	1.0	(3.4)	120	1.0	(3.4)	252	7.1	(24.5)	93539
sonar	208	60	2	3.0	(5.0)	21	17.8	(29.7)	90	3.9	(6.5)	920
vote	435	16	2	1.6	(10.0)	9	1.0	(6.3)	4	15.5	(96.9)	651
Average					(23.7)			(35.1)			(45.3)	

standard data sets from the UCI repository [4]. The data sets and their characteristics are summarized in Table 3. The percentage of correct classification with C4.5, averaged over ten ten-fold cross-validation runs, were calculated for each algorithm-data set combination before and after feature selection by SOAP (RF 0.75), CFS and ReliefF (threshold 0.05). For each train-test split, the dimensionality was reduced by each feature selector before being passed to the learning algorithms. The same fold were used for each feature selector-learning scheme combination.

To perform the experiment with CFS and ReliefF we used the Weka[1] (Waikato Environment for Knowledge Analysis) implementation.

Table 3 shows the average number of features selected and the percentage of the original features retained. SOAP is a specially selective algorithm compared with CFS and RLF. If SOAP and CFS are compared, only in one data set (labor) is the number of characteristics significantly greater than those selected by CFS. In six data sets there are no significant differences, and in nine, the number of features is significantly smaller than CFS. Compare to RLF, only in glass2 and diabetes, SOAP obtains more parameters in the reduction process (threshold 0.05 is not sufficient). It can be seen that SOAP retained 23,7% of the attributes on average.

Table 4 shows the results for attribute selection with C4.5 and compares the size (number of nodes) of the trees produced by each attribute selection scheme

[1] http://www.cs.waikato.ac.nz/~ml

Table 4. Result of attribute selection with C4.5. Accuracy and size of trees. ○, • Statistically significant improvement or degradation (p=0.05)

	DATA		**SOAP**		**CFS**		**RLF**	
Set	Ac.	Size	Ac.	Size	Ac.	Size	Ac.	Size
autos	82.54	63.32	73.37 •	45.84 ○	74.54 •	55.66 ○	74.15 •	85.74 •
breast-c	74.37	12.34	70.24 •	6.61 ○	72.90	18.94 •	70.42 •	11.31
breast-w	95.01	24.96	94.64	21.28 ○	95.02	24.68	95.02	24.68
diabetes	74.64	42.06	74.14	7.78 ○	74.36	14.68 ○	65.10 •	1.00 ○
glass2	78.71	24.00	78.96	14.88 ○	79.82	14.06 ○	53.50 •	1.70 ○
heart-c	76.83	43.87	77.06	34.02 ○	77.16	29.35 ○	79.60 ○	28.72 ○
heart-stat	78.11	34.58	80.67 ○	19.50 ○	80.63 ○	23.84 ○	82.33 ○	14.78 ○
hepatitis	78.97	17.06	80.19	5.62 ○	81.68 ○	8.68 ○	80.45	11.26 ○
horse-c.OR.	66.30	1.00	66.30	1.00	66.30	1.00	66.28	1.36 •
hypothyroid	99.54	27.84	95.02 •	4.30 ○	96.64 •	5.90 ○	93.52 •	12.52 ○
iris	94.27	8.18	94.40	8.12	94.13	7.98	94.40	8.16
labor	80.70	6.93	78.25	3.76 ○	80.35	6.44	80.00	5.88 ○
lymph	77.36	28.05	72.84 •	7.34 ○	75.95	20.32 ○	74.66	24.10 ○
sick	98.66	49.02	93.88 •	1.00 ○	96.32 •	5.00 ○	93.88 •	1.00 ○
sonar	74.28	27.98	70.05 •	7.00 ○	74.38	28.18	70.19 •	9.74 ○
vote	96.53	10.64	95.63 •	3.00 ○	95.63 •	3.00 ○	96.53	10.64
Average	82.93	26.36	80.98	11.94	82.24	16.73	79.38	15.79

against the size of the trees produced by C4.5 with no attribute selection. Smaller trees are preferred as they are easier to interpret, but accuracy is generally degraded. The table shows how often each method performs significantly better (denoted by ○) or worse (denoted by •) than when performing no feature selection (column 2 and 3). Throughout we speak of results being significantly different if the difference is statistically at the 5% level according to a paired two-sided t test. Each pair of points consisting of the estimates obtained in one of the ten, ten-fold cross-validation runs, for before and after feature selection. For SOAP, feature selection degrades performance on seven data sets, improves on one and it is equal on eight. The reason for why the algorithm is not as accurate is the number of attribute selected, less than three feature. Five of these seven data sets obtain a percentage less than 10% of the original features. The results are similar to ReliefF and a little worse than those provided by CFS. Analyzing the data sets in which SOAP lost to CFS, we can observe breast-c, lymph and sonar, where the number of feature selected by SOAP is 25% of CFS (breast-c 4,1 to 1,5 with SOAP, lymph 8,9-1,8 and sonar 17,8-3). Nevertheless the accuracy reduction is small: breast-c 72,9 (CFS) to 70,24 with SOAP, lymph 75,95-72,84 and sonar 74,38-70,05.

It is interesting to compare the speed of the attribute selection techniques. We measured the time taken in milliseconds to select the final subset of attributes. SOAP is an algorithm with a very short computation time. The results shown

in Table 3 confirm the expectations. SOAP takes 400 milliseconds[2] in reducing 16 data sets whereas CFS takes 853 milliseconds and RLF more than 3 minutes. In general, SOAP is faster than the other methods and it is independent of the classes number. Also it is possible to be observed that ReliefF is affected very negatively by the number of instances in the data set, it can be seen in "hypothyroid" and "sick". Even though these two data sets were eliminated, SOAP is more than 3 times faster than CFS, and more than 75 times than ReliefF.

4 Conclusions

In this paper we present a deterministic attribute selection algorithm. It is a very efficient and simple method used in the preprocessing phase. A considerable reduction of the number of attributes is produced in comparison to other techniques. It does not need distance nor statistical calculations, which could be very costly in time (correlation, gain of information, etc.). The computational cost is lower than other methods O(m n log n).

References

[1] Aguilar-Ruiz, Jesús S., Riquelme, José C. and Toro, Miguel. Data Set Editing by Ordered Projection. Intelligent Data Analysis Journal. Vol. 5, n°5, pp. 1-13, IOS Press (2001). 325
[2] Almuallim, H. and Dietterich, T. G. Learning boolean concepts in the presence of many irrelevant features. Artificial Intelligence, 69(1-2):279-305 (1994).
[3] Blake, C. and Merz, E. K. UCI Repository of machine learning databases (1998).
[4] Hall M. A. Correlation-based feature selection for machine learning. PhD thesis, Department of Computer Science, University of Waikato, Hamilton, New Zealand (1998).
[5] Hoare, C. A. R. QuickSort. Computer Journal, 5(1):10-15 (1962). 327
[6] Kira, K. and Rendell, L. A practical approach to feature selection. In Proceedings of the Ninth International Conference on Machine Learning. pp. 249-256, Morgan Kaufmann (1992).
[7] Kohavi, R. and John, G. H. Wrappers for feature subset selection. Artificial Intelligence, 97, 273-324 (1997).
[8] Kononenko, I. Estimating attibutes: Analisys and extensions of relief. In Proceedings of the Seventh European Conference on Machine Learning. pp. 171-182, Springer-Verlag (1994).
[9] Quinlan, J. C4.5: Programs for machine learning. Morgan Kaufmann (1993).
[10] Robnik-Šikonja, M. And Kononenko, I. An adaption of relief for attribute estimation in regression. In Proceedings of the Fourteenth International Conference on Machine Learning. pp. 296-304, Morgan Kaufmann (1997).
[11] Setiono, R., and Liu, H. A probabilistic approach to feature selection-a filter solution. In Proceedings of International Conference on Machine Learning, 319-327 (1996).

[2] This is a rough measure. Obtaining true cpu time from within a Java program is quite difficult.