Heuristic Search over a Ranking for Feature Selection

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

Department of Computer Science, University of Seville, Spain
{rruiz, riquelme, aguilar}@lsi.us.es

Abstract. In this work, we suggest a new feature selection technique that lets us use the wrapper approach for finding a well suited feature set for distinguishing experiment classes in high dimensional data sets. Our method is based on the relevance and redundancy idea, in the sense that a ranked-feature is chosen if additional information is gained by adding it. This heuristic leads to considerably better accuracy results, in comparison to the full set, and other representative feature selection algorithms in twelve well–known data sets, coupled with notable dimensionality reduction.

1 Introduction

In recent years, there has been an explosion in the rate of acquisition of data in several domains. A typical data set may contain thousands of features. Theoretically, having more features should give us more discriminating power. However, this can cause several problems: increase computational complexity and cost; too many redundant or irrelevant features; and estimation degradation in the classification error.

Most of the feature selection algorithms approach the task as a search problem, where each state in the search specifies a distinct subset of the possible attributes [1]. The search procedure is combined with a criterion in order to evaluate the merit of each candidate subset of attributes. There are a lot of possible combinations between each procedure search and each attribute measure [2]. Feature selection is grouped in two ways according to the attribute evaluation measure: depending on the type (filter or wrapper techniques) or on the way that features are evaluates (individual or subset evaluation). The filter model relies on general characteristics of the data to evaluate and select feature subsets without involving any mining algorithm. The wrapper model requires one predetermined mining algorithm and uses its performance as the evaluation criterion. It searches for features better suited to the mining algorithm, aiming to improve mining performance, but it also is more computationally expensive [3] than filter model. Feature ranking (FR), also called feature weighting [1, 4], assesses individual features and assigns them weights according to their degrees of relevance, while the feature subset selection (FSS) evaluates the goodness of each found feature subset. (Unusually, some search strategies in combination

with subset evaluation can provide a ranked list). In the FR algorithms category, a subset of features is often selected from the top of a ranking list. This approach is efficient for high–dimensional data due to its linear time complexity in terms of dimensionality. In the FSS algorithms category, candidate feature subsets are generated based on a certain search strategy. Different algorithms address theses issues distinctively. In [2], a great number of selection methods are categorized. We found different search strategies, namely exhaustive, heuristic and random search, combined with several types of measures to form different algorithms. The time complexity is exponential in terms of data dimensionality for exhaustive search and quadratic for heuristic search. The complexity can be linear to the number of iterations in a random search [5], but experiments show that in order to find best feature subset, the number of iterations required is usually at least quadratic to the number of features [6]. The most popular search methods in machine learning ([7, 8]) can not be applied to these data sets due to the large number of features. One of the few used search techniques in these domains is sequential forward [9, 10, 11] (also called hill–climbing or greedy).

The limitations of both approaches, ranking and subset selection, clearly suggest that we should pursue a hybrid model. Recently, a new framework for feature selection has been used, where several above–mentioned approaches are combined. The process of selection involves two phases due to the high number of attributes: Algorithms begin with a phase where attributes are individually evaluated, and provide a ranking according to a filter criterion. In the next step, a feature subset evaluator (filter or wrapper) is applied to a fixed number of attributes from the previous ranking (greater than a threshold value, or the first k features) following a search strategy. The method proposed by Xing et al. [12], the one proposed by Yu and Liu [9], and another by Guyon et al. [13] are among the most referenced works at present following this framework.

Our paper is organized as follows. In section 2, we present the concept of relevance and redundancy at the same time used in our wrapper approach. Algorithm is described in section 3. Experimental results are shown in Section 4, and the most interesting conclusions are summarized in section 5.

2 Wrapper Approach over Feature Ranking

Feature ranking makes use of a scoring function $S(i)$ computed from the values $x_{k,i}$ and y_k ($k = 1, \ldots, m$ examples and $i = 1, \ldots, n$ features). By convention, we assume that a high score is indicative of high relevance and that features are sorted in decreasing order of $S(i)$. We consider ranking criteria defined for individual features, independently of the context of others. In feature subset selection, it is a fact that two types of attributes are generally perceived as being unnecessary: attributes that are irrelevant to the target concept, and attributes that are redundant given other attributes. We now formally define incremental ranked (IR) usefulness in order to devise an approach to explicitly identify relevant features and do not take into account redundant features. In other words,

learning can be achieved more efficiently and effectively with just relevant and non–redundant features.

Definition 1. *Let R be a set of M features sorted in decreasing order of $S(i)$, given a sample of data D, a learning algorithm L , and a subset of selected features F, feature F_i is incrementally useful to L with respect to F if the accuracy of the hypothesis that L produces using the group of features $\{F_i\} \cup F$ is better significantly (denoted by $>$) than the accuracy achieved using just the subset of features F, in this case F_i is added to F. Note that the process starts from the first feature in R, and continues with the next ranked attribute.*

Wrapper subset evaluates attribute sets by using a learning scheme. Five cross validation is used to estimate the accuracy of the learning scheme for a set of features. We conduct Student's paired two–tailed t–test in order to evaluate the statistical significance (at 0.1 level) of the difference between the previous best subset and the candidate subset. This last definition allows us to select features from the ranking, but only those that increase the classification rate significantly. Although the size of the sample is small (5 cross validation), our search method use a t–test. We want to obtain an heuristic not to do an accurate population study. However, on the one hand it must be noted that it is an heuristic based on an objective criterion, to determine the statistical significance degree of difference between the accuracy of each subset. On the other hand, the confidence level has been relaxed from 0.05 to 0.1 due to the small size of the sample. Statistically significant differences at the $p < 0.05$ significance level would not permitted us to add more features, because it would be difficult to obtain significant differences between the accuracy of each subset by the test. Obviously, if the confidence level is increased, more features can be selected, and vice versa. Then, the user can adjust the confidence level.

3   Algorithm

There are two phases in the algorithm shown in Figure 3: Firstly, the features are ranked according to some evaluation measure (line 1–4). In second place, we deal with the list of attributes once, crossing the ranking from the beginning to the last ranked feature (line 5–12).

Consider the situation depicted in Figure 2; an example of feature selection process by IR. It shows the attributes ranked according to some evaluation measure. We obtain the classification accuracy with the first feature in the list (f5). In the second step, we run the classifier with the first two features of the ranking (f5,f7), and a paired t–test is performed to determine the statistical significance degree of the differences. As it is lower than 0.1, f7 is not selected. The same occurs with the two next subsets (f5,f4 and f5,f3), but feature f1 is added, because the accuracy obtained is significatively better than that obtained with only f5, and so on. In short, the classifier is run nine times to select, or not, the ranked features (f5,f1,f2): once with only one feature, four times with two features, three with three features and once with four features. The same situation occurs in high–dimensional data.

```
Input: E training, U--measure, W--classifier
Output: BestSubset
1   list l = {}
2   for each F_i ∈ F
3       S(i) = compute(f_i, U)
4       position F_i into l according to S(i)
5   BestClassification = 0
6   BestSubset = ∅
7   for each F_i ∈ l
8       TempSubset = BestSubset ∪ F_i
9       TempClassification = WrapperClassification(TempSubset, W)
10      if (TempClassification > BestClassification)
11          BestSubset = TempSubset
12          BestClassification = TempClassification
```
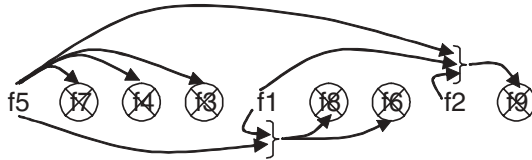
Fig. 1. IR Algorithm

Fig. 2. Example of feature selection process by IR

4   Experiments and Results

The aim of this section is to evaluate our approach in terms of classification accuracy, degree of dimensionality and speed on selected features, in order to see how IR fare in situations where there are large numbers of features. The comparison was performed with two representative groups of high–dimensional data sets: Three data sets are selected from the UCI Repository[1], and three selected from the NIPS 2003[2] feature selection benchmark. The main characteristic of these data sets is the great number of features. The full characteristics of all the data sets are summarized in Table 1. In order to compare the effectiveness of feature selection, attribute sets chosen by each technique were tested with two learning algorithms, a probabilistic (naive Bayes) and a decision tree learner (c4.5). These two algorithms were chosen because they represent two quite different approaches to learning.

As already mentioned, the proposed search is realized over a ranking of attributes, and any evaluation measure can be used for it. In the experiments, we use two criterions: one belongs to wrapper model and one to filter model. In the wrapper approach, denoted by IR_W, we order attributes according to their indi-

---

[1] http://www.ics.uci.edu/ mlearn/MLRepository.html
[2] http://clopinet.com/isabelle/Projects/NIPS2003/

Table 1. Data sets

Data Acr.	Feat.	Inst.	Classes
Musk[1] MK	166	6598	2
Arrhythmia[1] AR	279	452	16
Madelon[2] MA	500	2000	2
Multi–feature[1] MF	649	2000	10
Arcene[2] AC	10000	100	2
Dexter[2] DE	20000	300	2

Table 2. Accuracy of nb on selected features. The symbol " + " and " − " respectively identify statistically significant, at 0.1 level, wins or losses over IR_W

Data	Wrapper					Filter				Full
	IR_W		IR_F		SF_W		CFS_{SF}		$FOCUS_{SF}$	
	Acc	Att	Acc	Att	Acc	Att	Acc	Att	Acc	Att	Acc
MK	84.59	1	84.59	1	N/A		65.75$^-$	10	83.37$^-$	11	83.86
AR	73.01	7	73.02	8	74.35	15	69.69$^-$	25	69.03$^-$	21	61.74$^-$
MA	63.00	4	62.65	3	62.75	6	60.90	6	59.15$^-$	15	58.40$^-$
MF	97.30	15	97.85	19	N/A		97.10	86	93.65$^-$	7	93.35$^-$
AC	90.00	22	93.00	19	83.00	4	N/A		60.00$^-$	4	70.00$^-$
DE	88.67	14	88.00	15	84.67	11	N/A		90.33	23	88.67

vidual predictive power, using as criterion the performance of the target classifier built with a single feature. In the filter approach, a ranking is provided using non–linear correlation measure. We choose symmetrical uncertainty (denoted by IR_F), based on entropy and information gain concepts.

Due to the high–dimensional data, we limit our comparison to sequential forward (SF) techniques (see Introduction section). We choose three representative subset evaluation measures in combination with SF search engine. One, denoted by SF_W, uses a target learning algorithm to estimate the worth of attribute subsets; the other two are subset search algorithms which exploit sequential forward search and utilize correlation measure (variation of CFS algorithm [8]) or consistency measure (variation of FOCUS [7]) to guide the search, denoted by CFS_{SF} and $FOCUS_{SF}$ respectively (both of them used in [9]).

The experiments are conducted using the WEKA's implementation of all these existing algorithms and our algorithm is also implemented in the WEKA environment [14]. For each data set, we run CFS_{SF} and $FOCUS_{SF}$ algorithms (both of them are independent of the learning algorithm), and for each data set and each classifier, we run the wrapper feature selection algorithms, IR_W, IR_F and SF_W. We record the running time and the number of selected features for each algorithm. We then apply the two classifiers (nb and c4) on the original data set as well as on each newly obtained data set containing only the selected features from each algorithm and record overall accuracy by a 10–fold cross–validation.

Table 3. Accuracy of c4 on selected features

Data	Wrapper						Filter				Full
	IR_W		IR_F		SF_W		CFS_{SF}		$FOCUS_{SF}$		
	Acc	Att	Acc	Att	Acc	Att	Acc	Att	Acc	Att	Acc
MK	96.83	7	96.30	7	96.44	6	95.54^-	10	95.04^-	11	96.88
AR	74.32	6	73.02	5	74.10	8	69.04^-	25	71.67	21	64.38^-
MA	83.50	9	80.20^-	23	80.80^-	11	74.55^-	6	78.20^-	15	70.35^-
MF	95.70	13	94.55	10	95.70	17	94.45^-	86	91.40^-	7	94.75
AC	91.00	6	94.00	9	95.00	7	N/A		77.00^-	4	74.00^-
DE	88.00	12	88.33	17	90.33	12	N/A		89.33	23	76.00^-

Tables 2 and 3 report accuracy and number of features selected from nb and c4 respectively by each feature selection algorithm and the full set. We conduct an Students paired two–tailed t–test in order to evaluate the statistical significance of the difference between two averaged accuracy values: one resulted from IR_W and the other resulted from one of IR_F, SF_W, CFS_{SF}, $FOCUS_{SF}$ and the full set. The symbol " + " and " − " respectively identify statistically significant, at 0.1 level, wins or losses over IR_W. And Table 4 records the running time for each feature selection algorithm, showing two results for each wrapper approach, depending on the learning algorithm chosen.

Before we compare our technique with the others. Note the similarity between the results obtained with the two approaches of our algorithm, one based on a ranking–wrapper (IR_W) and the other on a ranking–filter (IR_F). As we can see from Table 2 and 3, in all the cases, except for one data set (MA) with c4 classifier, these accuracy differences are not statistically significant. the number of attributes selected are similar but IR_F is a little bit faster than IR_W because of the time needed to build the ranking for the wrapper–ranking approach.

Apart from the previous comparison, we study the behavior of IR_W comparing in three way: with respect to a whole set of features; with respect to another wrapper approach; and with respect to two filter approaches.

Classification accuracies obtained with the whole feature set are statistically lower than those obtained with our wrapper approach. As we can see from the last column in Table 2 and 3, IR_W wins in most of the cases, except in two data sets (MK and DE) and two data sets (MK and MF) for nb and c4 respectively. These accuracy differences are especially relevant in two data sets (AR and AC) and four (AR, MA, AC and DE) for nb and c4 respectively. We notice that the number of selected features is drastically low as regards the whole set.

For the two classifiers, no statistical significant differences are shown, except for c4 in MA data set, between the accuracy of our wrapper approach and the accuracy of the sequential forward wrapper procedure (SF_W). On the other hand, the advantage of IR_W with respect to the SF_W for nb and c4 is clear. We can observe (see Table 4) that IR_W is consistently faster than SF_W. The time savings from IR_W become more obvious when the computer–load necessities of the mining algorithm increases. In many cases the time savings are in

Table 4. Running time (seconds) for each feature selection algorithm

Data	Wrapper						Filter	
	nb			c4				
	IR_W	IR_F	SF_W	IR_W	IR_F	SF_W	CFS_{SF}	$FOCUS_{SF}$
MK	334	72	N/A	2400	2700	10277	10	77
AR	251	140	4089	291	245	2400	2	14
MA	156	96	825	2460	5100	18000	5	52
MF	1984	2643	N/A	6502	5280	72000	73	45
AC	1020	660	1027	1121	945	5820	N/A	35
DE	3300	2622	20280	9240	20880	86400	N/A	1320

degrees of magnitude, and in two cases, SF_W did not report any results: for nb in MF data set SF_W did not produce any results after forty eight hours running (hence, neither selected features nor accuracy results); and in MK data set for nb classifier, results are not shown because the accuracy obtained with each individual feature is lower than without feature. These results verify the superior computational efficiency of incremental search applied by IR_W over greedy sequential search applied by SF_W, with similar number of attributes and without statistical significant differences.

In general, the computer–load necessities of filter procedures can be considered as negligible with respect to wrapper ones (Table 4), except for $FOCUS_{SF}$ in DE data set. However, accuracies obtained with IR_W are notably better for nb and c4. Firstly, for the last two data sets (AC and DE) results were not produced by CFS_{SF} because the program ran out of memory after a period of considerably long time due to its quadratic space complexity. Secondly, in the rest of data sets, IR_W either improves or maintains the accuracy of both CFS_{SF} and $FOCUS_{SF}$. From Table 2 and 3, it can be seen that apart from the two last data sets, IR_W improves CFS_{SF} on two and four data sets for nb and c4 respectively. And IR_W improves $FOCUS_{SF}$ on five and four data sets and no statistical significant differences on the rest.

5   Summary and Future Work

The success of many learning schemes, in their attempts to construct data models, hinges on the reliable identification of a small set of highly predictive attributes. The inclusion of irrelevant, redundant and noisy attributes in the model building process phase can result in poor predictive performance and increased computation. The most popular search methods in machine learning can not be applied to these data sets due to the large number of features. However, in this paper, we have proposed a new feature selection technique that lets us use a wrapper approach for finding a well suited feature set for classification. We use the incremental ranked usefulness definition to decide at the same time, whether a feature is relevant and non–redundant or not (non–relevant or redundant). The technique extracts the best non–consecutive features from the ranking, trying to

statistically avoid the influence of unnecessary attributes on the later classification. This new heuristic, named IR, shows an excellent performance comparing to the traditional sequential forward search technique, not only regarding the classification accuracy with respect to filter approaches, but also the computational cost with respect to the wrapper approach. By way of comparison, a rough estimate of the time required by the SF wrapper approach to choose this many features is on the order of thousands of hours, assuming the method does not get caught in a local minima first and prematurely stops adding attributes as a result.

Acknowledgements

References

1. Blum, A., Langley, P.: Selection of relevant features and examples in machine learning. In Greiner, R., Subramanian, D., eds.: Artificial Intelligence on Relevance. Volume 97. (1997) 245–271
2. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. on Knowledge and Data Engineering **17** (2005) 1–12
3. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intalligence **1-2** (1997) 273–324
4. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research **3** (2003) 1157–1182
5. Liu, H., Setiono, R.: A probabilistic approach to feature selection: a filter solution. In: 13th Inter. Conf. on Machine Learning, Morgan Kaufmann (1996) 319–327
6. Dash, M., Liu, H., Motoda, H.: Consistency based feature selection. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. (2000) 98–109
7. Almuallim, H., Dietterich, T.: Learning boolean concepts in the presence of many irrelevant features. Artificial Intelligence **69** (1994) 279–305
8. Hall, M.: Correlation-based feature selection for discrete and numeric class machine learning. In: 17th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA (2000) 359–366
9. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. Journal of machine learning research **5** (2004) 1205–24
10. Inza, I., Larrañaga, P., Blanco, R., Cerrolaza, A.: Filter versus wrapper gene selection approaches in dna microarray domains. Artificial Intelligence in Medicine **31** (2004) 91–103
11. Xiong, M., Fang, X., Zhao, J.: Biomarker identification by feature wrappers. Genome Res **11** (2001) 1878–87
12. Xing, E., Jordan, M., Karp, R.: Feature selection for high-dimensional genomic microarray data. In: Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA (2001) 601–608
13. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machine. Machine Learning **46** (2002) 389–422
14. Witten, I., Frank, E.: Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, San Francisco (2000)