JMLR: Workshop and Conference Proceedings 4: 148-162 New challenges for feature selection

Best Agglomerative Ranked Subset for Feature Selection

Roberto Ruiz ROBERTORUIZ@UPO.ES

Jesús S. Aguilar–Ruiz AGUILAR@UPO.ES
School of Engineering, Pablo de Olavide University
Ctra. Utrera km. 1, 41013 Seville, Spain.

José C. Riquelme RIQUELME@LSI.US.ES
Department of Computer Science, University of Seville
Avda. Reina Mercedes s/n, 41012 Seville, Spain.

Editor: Saeys et al.

Abstract

The enormous increase of the size in databases makes finding an optimal subset of features extremely difficult. In this paper, a new feature selection method is proposed that will allow any subset evaluator -including the wrapper evaluation method- to be used to find a group of features that will allow a distinction to be made between the different possible classes. The method, *BARS* (Best Agglomerative Ranked Subset), is based on the idea of relevance and redundancy, in the sense that a ranked feature (or set) is more relevant if it adds information when it is included in the final subset of selected features. This heuristic method reduces dimensionality drastically and leads to improvements in the accuracy, in comparison to a complete set and as opposed to other feature selection algorithms.

1. Introduction

A feature selection algorithm combines two processes: one is the search for an optimum and the other is an evaluation of sets of features. The evaluation measure estimates in the latter process will guide the search. Therefore, a feature selection algorithm is simply a search algorithm that should optimize a measure that shows how good a subset of features is (or an individual feature). There are many possible combinations of search methods and feature evaluation measures (1). However, search methods can be too costly in high-dimensional databases, particularly if a learning algorithm is applied as an evaluation criterion.

There are two ways to group feature selection algorithms, depending on the chosen evaluation measure: one, according to the model used (filter or wrapper) and two, according to the way in which the features are evaluated (individually or by subsets).

The filter model evaluates features according to heuristics based on overall data characteristics, notwithstanding the classification method applied, whereas the wrapper uses the behaviour of a classification algorithm as a feature evaluation criterion. The wrapper model chooses the features that show the best classification and help to improve a learning algorithm's behaviour. The downside is that its computational cost (2; 3) is higher than the filter model. A key factor in wrapper methods is the way in which a search is made in the feature subset space (4). It appears next to such search strategies as sequential greedy

search, best-first search and genetic algorithms (1). Most of them have a temporal $O(n^2)$ complexity and cannot be applied in databases with tens of thousands of features.

On the other hand, FR (Feature Ranking) methods evaluates features individually, whereas FSS (Feature Subset Selection) evaluates the benefits of each candidate subset. In the FR algorithm category, the first k features will make up the final subset. This is a good approach for high-dimensional databases, given their linear cost in relation to the number of features. However, in algorithms capable of feature subset selection, some sort of search strategy is used to generate the candidate subsets. There are many different search strategies: exhaustive, heuristic and random, combined with various types of measures that make up a large number of algorithms. The temporal complexity is exponential in relation to the dimensionality of the data in an exhaustive search and quadratic in a heuristic search. In a random search, the complexity may be linear to the number of iterations (5), but experience shows that the number of iterations needed to find an optimal subset is at least quadratic in relation to the number of features (6).

The aim of this paper is to study and propose a feature selection method that can be applied to high-dimensional databases in a supervised learning framework, concretely for classification purposes. Two classification learning algorithms will be used to compare the effects of feature selection: one probabilistic (Naïve Bayes) and a second one based on decision trees (C4.5).

The paper is structured as follows: after an introduction on the particularities of feature subset selection in large databases, a review will be made of the related literature and the general concepts of relevance and redundancy. *BARS* algorithm is described in Section 3, and the results obtained are shown in Section 4. Finally, Section 5 gives some of the more interesting conclusions.

2. Related work

The limitations of both approaches, FR and FSS, clearly suggest the need for a hybrid model. Lately, a new framework (7) for feature selection is used which includes several of the methods given above, as well as the concepts of feature relevance and redundancy. There are many definitions of relevance. We have chosen the one offered by (8) that is considered especially suited to obtain a predictive feature subset. With respect to the redundancy, it is usually expressed in terms of a correlation between features. Between pairs of variables we can distinguish linear and non-linear correlation. However, it is not so clear how to know when a feature is correlated with a set of features. (9) apply a technique based on cross-entropy (KL-distance, (10)), called *Markov blanket* filtering, to eliminate redundant features.

In databases with a large number of features, the selection process usually is divided into two stages: In stage one, features are evaluated individually, providing a ranking based on a filtering criterion. In stage two, a feature subset evaluator (filter or wrapper) is applied to a certain number of features in the previous ranking (the ones that pass a threshold, or the first k), following a search strategy. (11), (7), (12) and (13) are among the most cited works that follow this path. Another work employed a linear sequential search over a ranking (14), and any type of criteria could be used in ranking and in generating a feature subset.

When databases with a lot of features are ranked, there are normally many features with similar scores. The frequent selection of redundant features in the final subset is often criticized. However, according to (15), taking into account presumably redundant features can reduce noise and, therefore, a better separation between the various classes can be obtained. Moreover, a very high correlation (in absolute value) between variables does not mean that they do not complement each other. Consequently, the idea of redundancy in this paper is not based on the measure of correlation between two features. Rather, it is based on any subset evaluation criterion, which may be a filter or wrapper approach. In this sense, a feature (or set) is selected if additional information is obtained when it is added to the previously selected feature subset, and rejected in the opposite case because the information provided is already contained (redundant) in the previous subset.

3. Agglomerative search for feature subset selection

In this work, a new method is proposed. It is called agglomerative due to the way it constructs the final subset of selected features. The method begins by generating a ranking. Then, pairs of features are obtained with the ranking's first features, in combination with each one of the remaining features on the list. The pairs of features are ranked according to the value of the evaluation, and the process is repeated, that is, the subsets made up by the first sets on the new list are compared with the rest of the sets. The process ends when only one feature subset is left, or when combining the subsets no longer causes an improvement.

Our approach uses a fast search through the attribute space and any subset evaluation measure, classifier approach included, can be embedded into it as evaluator. Therefore, a feature subset evaluator, named *SubEvaluator*, is used to select a small group of features. Thus, given a *SubEvaluator* and given a feature subset X, a search is made in the $\mathcal{P}(\mathcal{X})$ space for the feature subset with the best evaluation result, using the value to compare the behaviour of the *SubEvaluator* on the test subset. Before continuing, we shall establish the concepts of accuracy and measure value:

The result of dividing the number of correct classifications by the total number of samples examined is wide known as accuracy (Γ). Then, given a set tagged E of m instances $(\overline{x_j}, y_j)$, where $j = 1, \ldots, m$, each one composed of n input values $x_{j,i}$ with $(i = 1, \ldots, n)$ and one output y_j, and given the classifier L, in the following expression, if $\mathsf{L}(\overline{x_j}) = y_j$ (example well classified) then 1 is counted, and 0 in any other case.

$$\Gamma(\mathsf{E}, \mathsf{L}) = \frac{1}{m} \sum_{j=1}^{m} (\mathsf{L}(\overline{x_j}) = y_j)$$

Considering that, in this paper, the selection algorithms are subsequently applied to a classification task, the definition of accuracy given for the total set of data, applied to a feature subset S, would be:

$$\Gamma(\mathsf{E}/\mathsf{S}, \mathsf{L}) = \frac{1}{m} \sum_{j=1}^{m} (\mathsf{L}(\mathsf{S}(\overline{x_j})) = y_j)$$

Therefore, $\Gamma(\mathsf{E}/\mathsf{S}, \mathsf{L})$ is the accuracy, applying classifier L to the database with the features that belong to subset S. In addition to a wrapper measure to evaluate a feature subset,

we can use a filter type measure that also returns a real value on the goodness of the said subset. This value, which we will call a measure value, can be defined as:

Definition 1 (Measure value) *Let* E *be a set of tagged data;* S *a feature subset of* E *data; the* $\Upsilon(\mathsf{E}/\mathsf{S}, \not\mathsf{L})$ *measure value is the result of applying the evaluator type subset filter* $\not\mathsf{L}$ *considering only the* S *data subset.*

Therefore, if $\not\mathsf{L}$ is a subset evaluator, we can use two similar expressions: on the one hand $\Gamma(\mathsf{E}/\mathsf{S}, \mathsf{L})$, which is valid for evaluating a subset by means of a wrapper measure; on the other hand, the expression $\Upsilon(\mathsf{E}/\mathsf{S}, \not\mathsf{L})$, that evaluates the S subset by means of an $\not\mathsf{L}$ filter.

To make it easier, and to unify the notation, hereinafter the evaluation of a feature subset S will be annotated by $\Gamma(\mathsf{E}/\mathsf{S}, \mathsf{L})$ assuming a dual role (wrapper or filter).

Let $\Psi_1 = \{A_1^1, A_2^1, \ldots, A_n^1\}$ be the initial list of of candidate subsets, where $A_i^1 = \{X_i\}$, that is, each subset on the list has a feature. Let Ψ_1^k and Ψ_1^ϵ be two ranked sequences obtained from the set Ψ_1:

$$\Psi_1^k =< A_{(1)}^1, \ldots, A_{(k)}^1 >$$

the k first subsets of Ψ_1 ranked by L descendingly and

$$\Psi_1^\epsilon =< A_{(1)}^1, \ldots, A_{(\epsilon)}^1 >$$

the first ϵ of Ψ_1 ranked likewise by L, $k \leq \epsilon \leq n$, and n being the total number of features.

T_1 will be the result of applying the L subset evaluator to the subset best placed on the Ψ_1^k list, in this case, the most valued feature:

$$T_1 = \Gamma(\mathsf{E}/A_{(1)}^1, \mathsf{L})$$

As indicated above, Γ encompasses the accuracy value of a L wrapper as well as the measure value obtained with a filter type measure $\Upsilon(\mathsf{E}/A_{(1)}^1, \not\mathsf{L})$.

Sets based on the two above ranked sequences (Ψ_1^k and Ψ_1^ϵ) are constructed below in such a way that each set of the first sequence is joined to each set of the second sequence. Of the new sets generated, we are only interested in the ones that improve the best result obtained with the subsets of Ψ_1, that is, the ones with a more favourable evaluation than T_1. We will call this new set Ψ_2:

$$\Psi_2 = \{A_i^2 | A_i^2 = A_{(j)}^1 \cup A_{(l)}^1$$

$\forall j : 1..k$ and $\forall l : 1..\epsilon$ with $A_{(j)}^1 \in \Psi_1^k$ and $A_{(l)}^1 \in \Psi_1^\epsilon$ and with $\Gamma(\mathsf{E}/A_i^2, \mathsf{L}) > T_1\}$.

Likewise, Ψ_2^k and Ψ_2^ϵ will be two ranked sequences obtained from the Ψ_2 set. In general, the list of solutions Ψ_p is defined as the candidate feature subset. It is made up of each subset by the joining of two subsets from the above list of solutions, and it obtained a more favourable evaluation than the best subset on that list. That is,

$$\Psi_p = \{A_i^p | A_i^p = A_{(j)}^{p-1} \cup A_{(l)}^{p-1}\} \ \wedge \ \Gamma(\mathsf{E}/A_i^p, \mathsf{L}) > T_{p-1}\}$$

The process continues until no new subsets are generated, or until the ones generated do not exceed the value of the goodness of the best subset on the above list of solutions (let

Algorithm 1 *BARS–Best Agglomerative Ranked Subset.*

Require: E–data set, U–ranking criterion, L–*SubEvaluator*, k–No. of initial sets in the ranking, ϵ–limits the number of subsets in the ranking.
Ensure: *BestSub*–feature subset

```
1: R ⇐ generateRanking(U, E))
2: p ⇐ 1
3: SolutionsListΨ_p ⇐ ∅
4: for i = 1 until n do
5:    SolutionsListΨ_p ⇐ SolutionsListΨ_p ∪ {A_i^1} (A_i^1 = {X_i} ∧ X_i ∈ R)
6: end for
7: while #SolutionsListΨ_p > 1 do
8:    T ⇐ Γ(E/A_1^p, L)
9:    p ⇐ p + 1
10:   SolutionsListΨ_p ⇐ ∅
11:   i ⇐ 1
12:   for j = 1 until k do
13:      for l = j + 1 until ε do
14:         A_i^p ⇐ A_(j)^(p−1) ∪ A_(l)^(p−1)
15:         if Γ(E/A_i^p, L) > T then
16:            SolutionsListΨ_p ⇐ SolutionsListΨ_p ∪ A_i^p
17:            i ⇐ i + 1
18:         end if
19:      end for
20:   end for
21:   rank SolutionsListΨ_p by Γ(E/A_i^p, L)
22: end while
23: BestsSub ⇐ A_1^(p−1)
```

Ψ_p be $\varnothing$). Therefore, the solution is established as $\mathsf{S} = A_{(1)}^{p-1}$, the most relevant subset on the last list with available solutions.

Unlike the conventional forward sequential search, where the best feature, best pair, best set of three and so on are obtained until no improvement occurs, the *BARS* method makes a search at a lower cost because it covers a smaller part of the features space. Furthermore, the search path developed by *BARS* is done around the most relevant subsets at each given moment, choosing the best k subsets in each cycle of the algorithm, and expanding the search to other relevance subsets in the ranking (ranking percentage ϵ). This way we can avoid getting caught in local minima. This approach provides the possibility of efficiently applying any evaluation measure, wrapper models included, in high-dimensional domains. The final subset is obviously not the optimum, but it is unfeasible to search for every possible subset of attributes through the search space.

3.1 Algorithm

The process followed until the final subset is obtained is (see algorithm 1): Step one generates a feature ranking (line 1) ranked from best to worst according to an evaluation measure (U). Next, a list of solutions is generated (line 2-6, $SolutionsList\Psi_p$), in such a way that a solution for each individual feature is created and the same ranking order is maintained. The steps required to make an agglomerative search is shown on lines 7-22. At the end, the algorithm returns the best positioned feature subset of all the subsets evaluated.

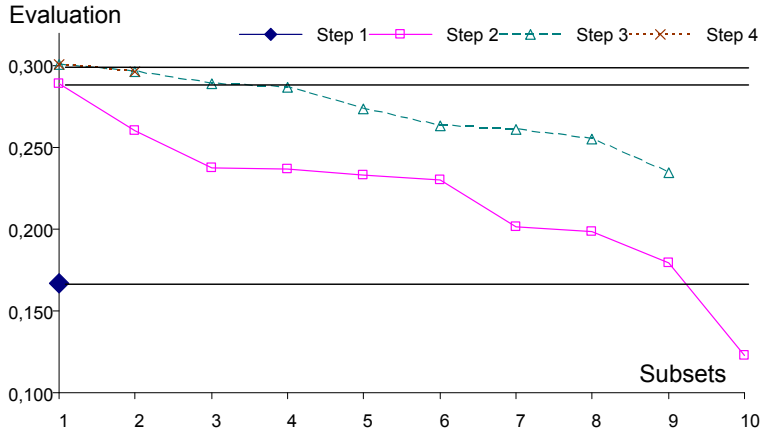

Figure 1: Example of the reduction process followed by *BARS*. The horizontal lines represent the limits at the end of each heuristic stage.

The agglomerative search consists in making a subset of relevant features by joining subsets with a lower number of features.

Each iteration of the repetitive `while` structure generates a new list of solutions from the previous structure. Each candidate set, made by joining two sets from the previous list of solutions, will become part of the next list of solutions if, when the subset evaluator L is applied to it, it gives back a higher measure value (Γ) than the one obtained with the best (or first) subset from the previous list of solutions (T). To prevent the algorithm from having a prohibitive time cost, new sets of features are generated by joining the first sets to the remaining previous list of solutions. That is, the first set on the list is joined to the second set, next the first set is joined to the third set, and so on until the end of the list. Next, the second set of the list is joined to the third set, the second set and the fourth set, and so on until the last set on the list. This process of combining a set of features with the rest of the sets on the list is carried out with the best k feature sets from the previous list of solutions (line 12).

In line 13, the input parameter ϵ can limit the number of new subsets. In data sets with an extremely high dimensionality, or when a high computational cost *SubEvaluator* is used, ϵ can be fixed to a percentage of the previous solutions number.

Figure 1 shows an example of the feature selection process with *BARS*, using the non-linear correlation *CFS* (Correlation–based Feature Selection algorithm (16)) as a subset evaluation measure (L). The figure represents the evaluation of the feature subsets in the different algorithm iterations. The numbers that can be seen on the abscissa axis represent the order of the subset in the corresponding ranking, and the ordinate axis shows the evaluation obtained for each subset. The horizontal lines set the limit at the end of each stage of the algorithm.

The reduction process followed in this example, for k equals three and ϵ equal to 100% of the ranking (nine features), is given below:

1. An initial feature ranking is generated. In this case, the non-linear correlation (CFS) is used as an evaluation measure, obtaining:
$\Psi_1 = [x_1, x_7, x_4], x_5, x_2, x_3, x_6, x_9, x_8$.

2. The evaluation of the first feature of the previous ranking (x_1) is used to set the limit. In this example, the threshold is set at 0.167, obtained by applying the CFS algorithm to the feature x_1, $\mathsf{L}(\mathsf{X_1}) = 0.167$.

3. Next, two subsets of features are made with the first three ($k = 3$) of the previous ranking ($\Psi_1^3 = [x_1, x_7, x_4]$) and the ranking ($\Psi_1^{100\%} = x_1, x_7, x_4, x_5, x_2, x_3, x_6, x_9, x_8$) with everything ($\epsilon = 100\%$) and they are evaluated with L (CFS). The sets with the evaluation in bold type have passed the threshold that was set beforehand with a feature (0.167).

   - With feature x_1 the following combinations are obtained: $(x_1, x_7 - \mathbf{0.261})$, $(x_1, x_4 - \mathbf{0.237})$, $(x_1, x_5 - 0.083)$, $(x_1, x_2 - 0.083)$, $(x_1, x_3 - \mathbf{0.202})$, $(x_1, x_6 - \mathbf{0.179})$, $(x_1, x_9 - 0.083)$, $(x_1, x_8 - 0.083)$
   - With feature x_7: $(x_7, x_4 - \mathbf{0.289})$, $(x_7, x_5 - 0.123)$, $(x_7, x_2 - 0.123)$, $(x_7, x_3 - \mathbf{0.234})$, $(x_7, x_6 - \mathbf{0.230})$, $(x_7, x_9 - 0.123)$, $(x_7, x_8 - 0.123)$
   - And with feature x_4: $(x_4, x_5 - 0.101)$, $(x_4, x_2 - 0.101)$, $(x_4, x_3 - \mathbf{0.237})$, $(x_4, x_6 - \mathbf{0.198})$, $(x_4, x_9 - 0.101)$, $(x_4, x_8 - 0.101)$

   Ranking the subsets that have improved to the previous best subset ($x_1 - 0.167$) leaves:
   $\Psi_2 = [(x_7, x_4 - 0.289), (x_1, x_7 - 0.261), (x_1, x_4 - 0.237)], (x_4, x_3 - 0.237), (x_7, x_3 - 0.234), (x_7, x_6 - 0.230), (x_1, x_3 - 0.202), (x_4, x_6 - 0.198), (x_1, x_6 - 0.179)$

4. The evaluation of the first subset in the ranking sets the new threshold ($x_7, x_4 - 0.289$). Once again, subsets are made with the three first sets of the last ranking generated ($\Psi_2^3 = [(x_7, x_4), (x_1, x_7), (x_1, x_4)]$) with the remaining pairs ($\Psi_2^{100\%}$), and they are evaluated with L. As in the previous step, the subsets that pass a new limit (0.289) are in bold type.

   - The combinations given below are obtained with the set (x_7, x_4): $(x_7, x_4, x_1 - \mathbf{0.296})$, $(x_7, x_4, x_1 - 0.296)$, $(x_7, x_4, x_3 - 0.289)$, $(x_7, x_4, x_3 - 0.289)$, $(x_7, x_4, x_6 - 0.273)$, $(x_7, x_4, x_1, x_3 - \mathbf{0.301})$, $(x_7, x_4, x_6 - 0.273)$, $(x_7, x_4, x_1, x_6 - 0.287)$
   - With the set (x_1, x_7): $(x_1, x_7, x_4 - 0.296)$, $(x_1, x_7, x_4, x_3 - 0.301)$, $(x_1, x_7, x_3 - 0.261)$, $(x_1, x_7, x_6 - 0.255)$, $(x_1, x_7, x_3 - 0.261)$, $(x_1, x_7, x_4, x_6 - 0.287)$, $(x_1, x_7, x_6 - 0.255)$
   - And with (x_1, x_4): $(x_1, x_4, x_3 - 0.264)$, $(x_1, x_4, x_7, x_3 - 0.301)$, $(x_1, x_4, x_7, x_6 - 0.287)$, $(x_1, x_4, x_3 - 0.264)$, $(x_1, x_4, x_6 - 0.234)$, $(x_1, x_4, x_6 - 0.234)$

   Ranking the subsets that pass the current threshold (0.289), we have: $\Psi_3 = [(x_7, x_4, x_1, x_3 - 0,301), (x_7, x_4, x_1 - 0,296)]$

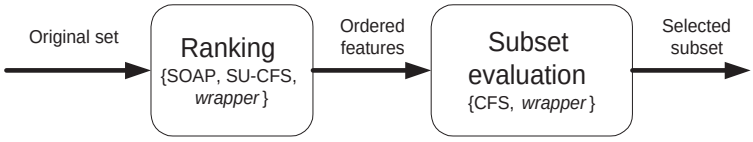

Figure 2: Parts of a $BARS$ selection algorithm.

5. In the next step of the example, the limit is set at 0.301 ($\mathsf{L}(x_7, x_4, x_1, x_3)$), which is not passed by any combination of subsets in the remaining ranking. Therefore, the process ends because the new list of solutions Ψ_4 (line 7) is empty. The previous list of solutions Ψ_3 is ranked according to subset relevance. The selected subset will be A_1^3, which occupies the first position of Ψ_3.

Generating sets that were already evaluated occurs very frequently in the process of combining two subsets. Therefore, the evaluated subsets will be controlled to prevent the evaluation from being repeated.

4. Experiments and results

The aim of this section is to evaluate our approach in terms of classification accuracy, degree of dimensionality and speed in selecting features, in order to see how good $BARS$ is in situations where there is a large number of features and instances.

The comparison was performed with two representative groups of data sets: Twelve data sets were selected from the UCI Repository (Table 1) and five from the NIPS 2003 feature selection benchmark (17). In this group (Table 2), the data sets were chosen to span a variety of domains (cancer prediction from mass-spectrometry data, handwritten digit recognition, text classification, and prediction of molecular activity). One data set is artificial. The input variables are continuous or binary, sparse or dense. In this second group all data sets are two-class classification problems. The full characteristics of all the data sets are summarized in Tables 1 and 2. We chose two different learning algorithms, C4.5 and Naïve Bayes, to evaluate the accuracy on selected features for each feature selection algorithm.

Figure 2 can be considered to illustrate the two blocks that always make up a $BARS$ algorithm. Therefore, this selection algorithm needs a ranking and a feature subset measure. Several versions of $BARS$ selection algorithms could be made by combining the criteria for each group of measures (individual and of subsets). Whenever the $BARS$ algorithm appears, the criteria for generating the ranking is given, as well as how feature subset quality was evaluated in the selection process. To clarify the components that $BARS$ uses in each case as much as possible, a subscript is put before BA that indicates the ranking method used, and a superscript is placed after it to indicate the *SubEvaluator*. In the experiments made for $BARS$, the same evaluation measure was used to prepare the ranking as the one used in the second part of the algorithm employed in the feature subset search. Two subset evaluation measures are used, one for each type of approach (wrapper and filter-CFS). For instance, ${}_{CF}BA^{CF}$ shows that CFS will be used as an individual measure in the first part and CFS as a subset in the second part, and ${}_{CF}BA^{WR}$ shows that NB or $C4$ classifier will be used as a subset evaluator in the second part.

Table 1: UCI data sets. Acron-acronym. Atts-number of attributes. Inst-number of instances.

Data	Acron.	Atts.	Inst.	Class
ads	ADS	1558	3279	2
arrhythmia	ARR	279	452	16
hypothyroid	HYP	29	3772	4
isolet	ISO	617	1559	26
kr vs kp	KRV	36	3196	2
letter	LET	16	20000	26
multi feat.	MUL	649	2000	10
mushroom	MUS	22	8124	2
musk	MUK	166	6598	2
sick	SIC	29	3772	2
splice	SPL	60	3190	3
waveform	WAV	40	5000	3

Table 2: NIPS data sets. Acron-acronym. Atts-number of attributes. %Ran-percentage of random Atts.

Data	Acron.	Atts.	Inst.	%Ran.
Arcene	ARC	10000	100	30
Dexter	DEX	20000	300	50
Dorothea	DOR	100000	800	50
Gisette	GIS	5000	6000	30
Madelon	MAD	500	2000	96

In the tests, *BARS* generated subsets from the three best subsets in the solutions list it used at each stage, limiting the range to 50% of the solutions list in wrapper type approaches and going to the end in the filter approaches. That is, $k = 3$ and ϵ=50% and 100%, respectively.

Due to the high dimensionality of data, we limited our comparison to sequential forward (SF) techniques, fast correlation–based filter ($FCBF$) algorithm (7) and the incremental ranked (BI) algorithm (14). We chose two representative subset evaluation measures in combination with SF search engine. One, denoted by SF_{WR}, uses a target learning algorithm (NB or $C4$) to estimate the worth of feature subsets; the other, denoted by SF^{CF}, is a subset search algorithm which exploit sequential forward search and use correlation

measures (CFS, Correlation–based Feature Selection algorithm (16)) to guide the search. BI use the same nomenclature as $BARS$.

Tables 3, 5, 4 and 6 show the results obtained with NB and C4 classifiers. By columns: the results obtained with $BARS$, BI and SF algorithms using the wrapper as a subset evaluator and using CFS; $FCBF$ algorithm; and the results obtained with the complete database. The columns headed by Ac and At$^{\%}$ show the success rate and the reduction percentage respectively. On NIPS data, the reduction percentage is too low to be comparable, therefore the number of features (At$^{\#}$) is shown. In each case, the results were obtained by calculating the mean of five executions of two crossed validations (5×2CV). Two reductions were made in each execution, one for each training set, to prevent the selection algorithm from being over-adjusted to the data used. A paired t-Student test was applied to evaluate whether the difference between the wrapper approach of the proposed algorithm and the other results was statistically significant at a confidence level of 0.05.

As it is possible to observe in Tables 3 and 5 obtained on data from Table 1, chosen subsets by $BARS$ are considerably smaller than the other compared techniques and drastically less than the original set. With NB classifier, the reduction percentage of $BARS$ was 9.8%, while the other methods obtained 15.8%–$_{NB}BI^{NB}$, 16.5%–SF^{NB}, 14.1%–SF^{CF} and 18.1%–$FCBF$, although with a light loss of precision. With C4.5 classifier the differences are greater $BARS$ was 10.4%, while the other methods obtained 16.3%–$_{C4}BI^{C4}$, 18.2%–SF^{NB}, 14.1%–SF^{CF} and 18.1%–$FCBF$, without significant lost of accuracy.

Notice that in two cases with C4 classifier (ISO and MUK) SF_{WR} did not report any results after three weeks running, therefore, there are not selected attributes nor success rates. Although it is supposed that the missed results would favor to $BARS$, since SF has not finished because of the inclusion of many attributes.

Actually, $BARS$ demonstrates its relevance on very high–dimensional data as it can be seen in Table 4 and 6. It is possible to be observed that in comparison with $FCBF$ the number of selected attributes is drastically smaller, without lost of precision in the classification.

Tables 7 and 8 report the running time for each feature selection algorithm on UCI and NIPS data sets respectively, showing three different results, two for wrapper approaches (depending on the learning algorithm chosen) and one for filter approaches. In the wrapper cases, we can observe that $BARS$ need a little more time than BI on average (see Table 7, columns 2–3 and 5–6). However, the time savings of $BARS$ with respect to BI became more obvious when the number of features of the data increased (for example ADS data set). This assertion is confirmed in Table 8 where $BARS$ takes half (columns 2-3) for NB classifier or quarter (columns 5-6) for C4.5. Therefore, the more data is high-dimensional the more $BARS$ is much more efficient.

On the other hand, the advantage of $BARS$ with respect to the SF (sequential forward search) is clear. $BARS$ is between 5 and 7 times faster (see Table 7, columns 2–4 and 5–7), and we must take into account that SF did not report any results on several data sets.

Finally, filter approaches are compared (columns from 8 to 11 in Table 7). $BARS$ is faster than the rest of filter options: BI, 25% on average; SF^{CF} (SF with CFS), 3 times faster; and $FCBF$ over 60%. On NIPS data (columns from 6 to 8 in Table 8) $BARS$ is faster than BI algorithm again, but 3 times slower than $FCBF$. This result can be explained by the way like $BARS$ search the minimum subset of attributes, while $FCBF$ does not fit

Table 3: Results obtained with NB for each feature selection algorithm on UCI Data. Ac-accuracy; At%- percentage of attributes retained. The symbols ∘ and • respectively identify statistically significant (at 0.05 level) wins or losses over the second column (${}_{NB}BA^{NB}$).

Data	${}_{NB}BA^{NB}$		${}_{NB}BI^{NB}$		SF^{NB}		${}_{CF}BA^{CF}$		${}_{CF}BI^{CF}$		SF^{CF}		FCBF		Orig.
	Ac	At%	Ac	At%	Ac	At%	Ac	At%	Ac	At%	Ac	At%	Ac	At%	Ac
ADS	95.80	0.5	95.42	0.7	95.83	1.1	94.61•	0.3	95.38	0.4	95.81	0.6	95.64	5.3	96.38
ARR	68.94	2.3	68.01	5.5	67.70	3.0	67.30	4.1	66.5	4.1	68.05	6.2	63.98	2.9	60.13
HYP	94.92	10.3	95.10	15.9	95.32	29.3	94.15•	3.4	94.15•	3.4	94.15•	3.4	94.90	18.3	95.32
ISO	77.41	3.4	83.30∘	11.1	82.28	4.7	66.95•	3.8	77.61	11.1	80.79	15.4	74.62	3.7	80.42
KRV	94.09	11.4	94.27	13.9	94.32∘	14.4	84.41	7.2	90.43•	8.3	90.43•	8.3	92.50	18.1	87.50•
LET	55.74	39.4	65.67∘	68.8	65.67∘	72.5	64.28∘	56.3	64.28∘	56.3	64.28∘	56.3	65.06∘	64.4	63.97∘
MUL	96.80	2.1	97.21	3.4	96.87	2.4	96.55	3.9	97.04	4.3	96.72	13.9	96.19	18.7	94.37•
MUS	98.68	7.3	98.78	9.5	99.01	13.6	98.52	4.5	98.52	4.5	98.52	4.5	98.52	16.4	95.10•
MUK	84.60	1.0	84.59	0.6	84.59	0.0	74.54	7.5	79.94	3.9	69.78•	9.8	72.29	1.7	83.56
SIC	93.88	3.4	94.55	8.3	93.88	0.0	93.89	3.4	93.89	3.4	93.89	3.4	96.25∘	16.6	92.41
SPL	94.65	15.3	94.85	21.8	94.91	24.7	93.63•	10.0	93.63•	10.0	93.60•	10.2	95.49	36.3	95.26
WAV	80.38	21.3	80.85	30.5	81.55	32.3	80.34	35.3	81.01	31.0	80.12	37.0	78.42	15.3	80.02
Av.	86.32	9.8	90.36	15.8	89.19	16.5	85.08	11.6	86.24	11.7	87.70	14.1	86.26	18.1	85.44

Table 4: Similar to Table 3 on NIPS Data. At#- number of attributes retained.

Data	${}_{NB}BA^{NB}$		${}_{NB}BI^{NB}$		SF^{NB}		${}_{CF}BA^{CF}$		${}_{CF}BI^{CF}$		SF^{CF}		FCBF		Orig.
	Ac	At#	Ac	At#	Ac	At#	Ac	At#	Ac	At#	Ac	At#	Ac	At#	Ac
ARC	65.40	4.6	64.60	15.3	60.60	3.8	66.00	22.5	63.20	39.2	60.20	42.6	61.20	35.2	65.40
DEX	79.13	15.3	81.33	30.2	75.33	13.2	80.67	7.5	82.47	11.3	87.73	35.5	85.07	25.1	86.47∘
DOR	93.25	2.3	93.23	10.5	N/A		93.25	2.1	93.80	11.9	N/A		92.38	75.3	90.68
GIS	91.17	9.2	92.66	35.3	93.55∘	24.2	87.26•	8.6	90.83	30.2	92.64	62.2	87.58	31.2	91.88
MAD	60.99	4.9	59.00	11.8	60.12	5.8	60.37	6.3	60.56	5.8	60.17	9.9	58.20	4.7	58.24

Table 5: Results obtained with C4 for each feature selection algorithm on UCI Data. Ac-accuracy; At$^{\%}$- percentage of attributes retained. The symbols ∘ and • respectively identify statistically significant (at 0.05 level) wins or losses over the second column ($_{NB}BA^{NB}$).

Data	$_{C4}BA^{C4}$		$_{C4}BI^{C4}$		SF^{C4}		$_{CF}BA^{CF}$		$_{CF}BI^{CF}$		SF^{CF}		FCBF		Orig.
	Ac	At$^{\%}$	Ac	At$^{\%}$	Ac	At$^{\%}$	Ac	At$^{\%}$	Ac	At$^{\%}$	Ac	At$^{\%}$	Ac	At$^{\%}$	Ac
ADS	96.42	0.5	96.55	0.5	96.85	0.8	95.30	0.3	96.43	0.4	96.39	0.6	95.85	5.3	96.46
ARR	67.92	2.0	68.01	2.4	67.39	3.1	66.46	4.1	66.42	4.1	67.04	6.2	64.87	2.9	64.29
HYP	98.90	10.7	99.07	14.5	99.30	20.3	96.56•	3.4	96.56•	3.4	96.56•	3.4	98.03	18.3	99.36
ISO	68.15	2.7	69.43	3.6	N/A		67.29	3.8	72.68	11.1	71.94	15.4	66.63	3.7	73.38
KRV	94.09	11.1	95.11	17.2	94.26	13.6	84.41	7.2	90.43•	8.3	90.43•	8.3	94.07	18.1	99.07∘
LET	80.50	45.0	84.99	68.8	85.17	63.1	84.21	56.3	84.21	56.3	84.21	56.3	84.84	64.4	84.45
MUL	93.74	1.5	92.42	3.2	93.11	2.1	92.77	3.9	93.17	4.3	93.12	13.9	92.29	18.7	92.74
MUS	99.41	9.1	99.91∘	18.6	100.00∘	22.3	98.52•	4.5	98.52•	4.5	98.52•	4.5	98.84	16.4	100.00∘
MUK	95.71	4.9	95.43	5.8	N/A		94.44	7.5	94.06•	3.9	94.60	9.8	91.19•	1.7	95.12
SIC	96.33	7.2	98.28∘	20.3	98.19∘	19.0	96.33	3.4	96.33	3.4	96.33	3.4	97.50∘	16.6	98.42∘
SPL	92.73	12.2	93.05	16.3	93.04	18.3	92.54	10.0	92.54	10.0	92.61	10.2	93.17	36.3	92.92
WAV	75.93	17.5	76.20	24.0	75.44	19.8	76.65	35.3	76.46	31.0	76.56	37.0	74.52	15.3	74.75
Av.	88.32	10.4	87.03	16.3	88.07	18.2	85.04	11.6	84.78	11.7	85.87	14.1	86.31	18.1	85.94

Table 6: Similar to Table 5 on NIPS Data. At$^{\#}$- number of attributes retained.

Data	$_{C4}BA^{C4}$		$_{C4}BI^{C4}$		SF^{C4}		$_{CF}BA^{CF}$		$_{CF}BI^{CF}$		SF^{CF}		FCBF		Orig.
	Ac	At$^{\#}$	Ac	At$^{\#}$	Ac	At$^{\#}$	Ac	At$^{\#}$	Ac	At$^{\#}$	Ac	At$^{\#}$	Ac	At$^{\#}$	Ac
ARC	63.60	2.7	65.80	7.9	62.40	3.7	61.60	22.5	59.00	39.2	56.60	42.6	58.80	35.2	57.00
DEX	78.30	5.8	80.27	18.9	90.47	8.7	80.40	7.5	81.47	11.3	80.13	35.5	79.00	25.1	73.80
DOR	93.20	2.6	92.13	7.2	N/A		93.20	2.1	91.63	11.9	N/A		90.33	75.3	88.73
GIS	93.00	11.5	93.29	26.9	N/A		89.60•	8.6	90.92	30.2	93.07	62.2	90.99•	31.2	92.68
MAD	68.40	4.3	73.02	17.0	72.99	12.4	69.30	6.3	69.77	5.8	69.29	9.9	61.11	4.7	57.73•

well to the minimum subset as it can be seen in column $\text{At}^{\#}$ (number of features) below $FCBF$, Tables 4 and 6.

5. Conclusions

The success of many learning schemes in the attempt to construct data models depends on their ability to identify a small subset of highly predictive features. During the model's construction stage, the inclusion of features that are irrelevant, redundant or have noise may cause poor predictive behaviour and a computational increase. Applying the more popular search methods in databases with a great many features may be prohibitive. However, in this paper, a new feature selection method has been presented that allows any subset evaluator -including the wrapper evaluation model- to be used to find a good set of features for classification. Our technique $BARS$ chooses a very small subset of features from the original set with similar predictive performance to other methods. For massive data sets, wrapper–based methods might be computationally unfeasible, so $BARS$ turns out a fast technique that provides good performance in prediction accuracy.

Acknowledgements

The research was supported by the Spanish Research Agency CICYT under grant TIN2007-68084-C02.

References

[1] Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. on Knowledge and Data Eng. **17**(3), 1–12 (2005)

[2] Langley, P.: Selection of relevant features in machine learning. In: Procs. Of the AAAI Fall Symposium on Relevance, 140–144 (1994)

[3] Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence **1-2**, 273–324 (1997)

[4] Inza, I., Larrañaga, P., Blanco, R., Cerrolaza, A.: Filter versus wrapper gene selection approaches in dna microarray domains. Artificial Intelligence in Medicine **31**, 91–103 (2004)

[5] Liu, H., Setiono, R.: A probabilistic approach to feature selection: a filter solution. In: 13th Int. Conf. on Machine Learning, 319–327. Morgan Kaufmann (1996)

[6] Dash, M., Liu, H., Motoda, H.: Consistency based feature selection. In: Pacific-Asia Conf. on Knowledge Discovery and Data Mining, 98–109 (2000)

[7] Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. Journal of machine learning research **5**, 1205–1224 (2004)

[8] Caruana, R., Freitag, D.: Greedy attribute selection. In: 11th Int. Conf. on Machine Learning, 28–36 (1994)

Table 7: Running time (seconds) for each feature selection algorithm on UCI data.

	Wrapper						Filter			
Datos	${}_{NB}BA^{NB}$	${}_{NB}BI^{NB}$	SF^{NB}	${}_{C4}BA^{C4}$	${}_{C4}BI^{C4}$	SF^{C4}	${}_{CF}BA^{CF}$	${}_{CF}BI^{CF}$	SF^{CF}	FCBF
ADS	165.1	217.5	2541.2	838.3	1895.2	18465.9	4.9	7.0	14.3	37.6
ARR	54.4	53.6	370.7	77.6	119.3	730.0	0.4	0.4	0.8	0.2
HYP	7.9	9.9	67.6	10.3	14.6	69.0	0.1	0.1	0.1	0.1
ISO	6761.2	4363.5	37293.2	20591.0	11632.0	N/A	17.4	21.1	58.2	9.5
KRV	4.0	2.5	8.9	4.5	7.3	19.3	0.0	0.0	0.1	0.0
LET	144.0	134.4	629.6	651.7	540.0	2703.8	1.5	1.5	1.3	0.9
MUL	2359.8	1251.4	8416.8	4727.7	2503.6	16975.3	9.2	12.0	49.7	15.9
MUS	4.5	4.2	9.4	5.3	5.8	18.4	0.1	0.1	0.1	0.1
MUK	52.6	40.7	16.2	2038.8	897.2	N/A	5.3	5.5	7.0	2.6
SIC	2.3	2.3	1.1	4.9	12.3	66.3	0.1	0.1	0.1	0.1
SPL	11.3	7.8	65.1	26.0	32.6	218.7	0.1	0.1	0.2	0.3
WAV	43.0	24.5	200.4	319.3	253.6	830.9	0.7	0.7	0.8	0.3
Total	9610,1	6112,3	49620,2	6665,6*	5384,3*	40097,6	39,8	48,6	132,7	67,6

* Results on ISO and MUK are not included.

Table 8: Similar to Table 7 on NIPS Data.

	Wrapper				Filter		
Datos	${}_{NB}BA^{NB}$	${}_{NB}BI^{NB}$	${}_{C4}BA^{C4}$	${}_{C4}BI^{C4}$	${}_{CF}BA^{CF}$	${}_{CF}BI^{CF}$	FCBF
ARC	191.1	266.4	267.2	384.5	70.0	96.0	2.8
DEX	714.1	1540.4	683.1	4460.4	53.9	259.7	8.9
DOR	6342.7	7137.2	6988.3	16525.1	3657.8	7525.5	1079.8
GIS	1511.2	9075.2	5854.8	34630.7	296.7	440.7	118.0
MAD	76.6	137.5	87.9	1269.1	2.5	3.5	1.4
Total	8835,7	18156,7	13881,3	57269,8	4080,9	8325,4	1210,9

[9] Koller, D., Sahami, M.: Toward optimal feature selection. In: 13th Int. Conf. on Machine Learning, 284–292 (1996)

[10] Kullback, S., Leibler, R.: On information and sufficiency. Annals of Mathematical Statistics **22**(1), 79–86 (1951)

[11] Xing, E., Jordan, M., Karp, R.: Feature selection for high-dimensional genomic microarray data. In: Proc. 18th Int. Conf. on Machine Learning, 601–608. Morgan Kaufmann, San Francisco, CA (2001)

[12] Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. In: IEEE Computer Society Bioinformatics, 523–529 (2003)

[13] Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machine. Machine Learning **46**(1-3), 389–422 (2002)

[14] Ruiz, R., Riquelme, J., Aguilar-Ruiz, J.: Incremental wrapper-based gene selection from microarray expression data for cancer classification. Pattern Recognition **39**, 2383–2392 (2006)

[15] Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research **3**, 1157–1182 (2003)

[16] Hall, M.: Correlation-based feature selection for discrete and numeric class machine learning. In: 17th Int. Conf. on Machine Learning, 359–366. Morgan Kaufmann, San Francisco, CA (2000)

[17] Guyon, I., Gunn, S., Ben-Hur, A., Dror, G.: Result analysis of the nips 2003 feature selection challenge. In: Advances in Neural Information Processing Systems, 545–552. MIT Press (2005)