Chapter 8

Efficient Incremental-Ranked Feature Selection in Massive Data

Roberto Ruiz

Pablo de Olavide University

Jesús S. Aguilar-Ruiz

Pablo de Olavide University

José C. Riquelme

University of Seville

8.1 Introduction

In recent years, there has been an explosion in the growth of databases in all areas of human endeavor. Progress in digital data acquisition and storage technology has resulted in the growth of huge databases. In this work, we address the feature selection issue under a classification framework. The aim is to build a classifier that accurately predicts the classes of new unlabeled instances. Theoretically, having more features and instances should give us more discriminating power. However, this can cause several problems: increased computational complexity and cost; too many redundant or irrelevant features; and estimation degradation in the classification error.

The problem of feature selection received a thorough treatment in pattern recognition and machine learning. 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 [2]. 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 [17, 4, 16]. However, search methods can be prohibitively expensive in massive datasets, especially when a data mining algorithm is applied as an evaluation function.

There are various ways in which feature selection algorithms can be grouped according to the attribute evaluation measure, depending on the type (filter or wrapper technique) or on the way that features are evaluated (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 [15, 13] than filter models. Feature ranking (FR), also called feature weighting [2, 8], 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 order to compare the effectiveness of feature selection, feature sets chosen by each technique are tested with three well-known learning algorithms: a probabilistic learner (naïve Bayes), an instance-based learner (IB1), and a decision tree learner (C4.5). These three algorithms have been chosen because they represent three quite different approaches to learning, and their long-standing tradition in classification studies.

The chapter is organized as follows. In the next two sections, we will review previous work, and notions of feature relevance and redundancy, respectively. In Section 8.4, we will present our proposed measures of feature relevance and redundancy using a wrapper or filter approach, and describe our algorithm. Experimental results are shown in Section 8.5, and the most interesting conclusions are summarized in Section 8.6.

8.2 Related Work

Traditional feature selection methods in some specific domain often select the top-ranked features according to their individual discriminative powers [7]. This approach is efficient for high-dimensional data due to its linear time complexity in terms of dimensionality. They can only capture the relevance of features to the target concept, but cannot discover redundancy and basic interactions among features. In the FSS algorithms category, candidate feature subsets are generated based on a certain search strategy. Different algorithms address these issues distinctively. In [17], a great number of selec-

tion methods are categorized. We found different search strategies, namely *exhaustive*, *heuristic*, and *random* searches, combined with several types of measures to form different algorithms. The time complexity is exponential in terms of data dimensionality for exhaustive searches and quadratic for heuristic searches. The complexity can be linear to the number of iterations in a random search, but experiments show that in order to find the best feature subset, the number of iterations required is usually at least quadratic to the number of features [5]. The most popular search methods in pattern recognition and machine learning cannot be applied to massive datasets due to the large number of features and instances (sometimes tens of thousands). One of the few used search techniques in these domains is sequential forward (SF, also called hill-climbing or greedy search). Different subset evaluation measures in combination with an SF search engine can be found. We are specially interested in the wrapper approach.

A key issue of wrapper methods is how to search into the space of subsets of features. Although several heuristic search strategies exist such as greedy sequential search, best-first search, and genetic algorithm, most of them are still computationally expensive $O(N^2)$ (with N the number of features of the original dataset), which prevents them from scaling well to datasets containing thousands of features. A rough estimate of the time required by most of these techniques is in the order of thousands of hours, assuming that the method does not get caught in a local minima first and stops prematurely. For example, if we have chosen 50 features from 20,000 (0.0025% of the whole set) through a greedy search, the subset evaluator would be run approximately one million times (N times to find the best single feature, then it tries each of the remaining features in conjunction with the best to find the most suited pair of features $N - 1$ times, and so on, more or less $20,000 \times 50$ times). Assuming 4 seconds on average by each evaluation, the results would take more than 1,000 hours.

The limitations of both approaches, FR and FSS, clearly suggest that we should pursue a hybrid model. Recently, a new framework of feature selection has been used, where several of the above-mentioned approaches are combined. [21] proposed a fast correlation-based filter algorithm (FCBF) that uses correlation measure to obtain relevant features and to remove redundancy. There are other methods based on relevance and redundancy concepts. Recursive feature elimination (RFE) is a proposed feature selection algorithm described in [10]. The method, given that one wishes to find only r dimensions in the final subset, works by trying to choose the r features that lead to the largest margin of class separation, using an SVM classifier. This combinatorial problem is solved in a greedy fashion at each iteration of training by removing the input dimension that decreases the margin the least until only r input dimensions remain (this is known as backward selection). The authors in [6] have used mutual information for gene selection that has maximum relevance with minimal redundancy by solving a simple two-objective optimization, and [20] proposes a hybrid of filter and wrapper approaches to feature selection.

In [12], the authors propose a rank search method to compare feature selection algorithms. Rank search techniques rank all features, and subsets of increasing size are evaluated from the ranked list (i.e., the first attribute, the two first ones, etc.). The best attribute set is reported. The authors apply the wrapper approach to datasets up to 300 attributes and state that for the ADS dataset (1,500 attributes) the estimated time to only generate the ranking in a machine with a 1.4GHz processor would be about 140 days and to evaluate the ranked list of attributes would take about 40 days. In contrast, our method can be tested on datasets with 20,000 features on a similar machine in a few hours.

This chapter presents a feature selection method, named *BIRS* (Best Incremental Ranked Subset), based on the hybrid model, and attempts to take advantage of all of the different approaches by exploiting their best performances in two steps: First, a filter or wrapper approach provides a ranked list of features, and, second, ordered features are added using a wrapper or filter subset evaluation ensuring good performance (the search algorithm is valid for any feature ranked list). This approach provides the possibility of efficiently applying any subset evaluator, wrapper model included, in large and high-dimensional domains, obtaining good results. The final subset is obviously not the optimum, but it is unfeasible to search for every possible subset of features through the search space. The main goal of our research is to obtain a few features with high predictive power. The wrapper version of this algorithm has been proved to be efficient and effective in microarray domains [18].

8.3 Preliminary Concepts

8.3.1 Relevance

The purpose of a feature subset algorithm is to identify relevant features according to a definition of relevance. However, the notion of relevance in machine learning has not yet been rigorously defined in common agreement [1]. Reference [13] includes three disjointed categories of feature relevance: strong relevance, weak relevance, and irrelevance. These groups are important to decide what features should be conserved and which ones can be eliminated. The strongly relevant features are, in theory, important to maintain a structure in the domain, and they should be conserved by any feature selection algorithm in order to avoid the addition of ambiguity to the sample. Weakly relevant features could be important or not, depending on the other features already selected and on the evaluation measure that has been chosen (accuracy, simplicity, consistency, etc.). Irrelevant attributes are not necessary at all. Reference [1] makes use of information theory concepts to define the en-

tropic or variable relevance of a feature with respect to the class. Reference [2] collects several relevance definitions. The above notions of relevance are independent of the specific learning algorithm being used. There is no guarantee that just because a feature is relevant, it will necessarily be useful to an algorithm (or vice versa). The definition of incremental relevance in [3] makes it explicit, since it is considered especially suited to obtain a predictive feature subset.

DEFINITION 8.1 Incremental usefulness *Given a sample of data* $\mathbf{X_L}$*, a learning algorithm* L*, a feature space* $\mathbf{F}$*, and a feature subset* $\mathbf{S}$ *(*$\mathbf{S} \subseteq \mathbf{F}$*), the feature* F_i *is incrementally useful to* L *with respect to* $\mathbf{S}$ *if the accuracy of the hypothesis that* L *produces using the group of features* $\{F_i\} \cup \mathbf{S}$ *is better than the accuracy achieved using just the subset of features* $\mathbf{S}$*.*

We consider this definition to be especially suited to obtain a predictive feature subset. In the next section, concepts can be applied to avoid a subset that contains attributes with the same information.

8.3.2 Redundancy

Notions of feature redundancy are normally in terms of feature correlation. It is widely accepted that two features are redundant to each other if their values are completely correlated. There are two widely used types of measures for the correlation between two variables: linear and non-linear. In the first case, the Pearson correlation coefficient is used, and in the second one, many measures are based on the concept of entropy, or the measure of the uncertainty of a random variable. Symmetrical uncertainty is frequently used, defined as

$$SU(X,Y) = 2\left[\frac{IG(X|Y)}{H(X)+H(Y)}\right]$$

where $H(X) = -\sum_i P(x_i)log_2(P(x_i))$ is the entropy of a variable X and $IG(X|Y) = H(X) - H(X|Y)$ is the information gain from X provided by Y.

The above-mentioned definitions are between pairs of variables. However, it may not be as straightforward in determining feature redundancy when one is correlated with a set of features. Reference [14] applies a technique based on cross-entropy, named Markov blanket filtering, to eliminate redundant features. This idea is formalized in the following definition.

DEFINITION 8.2 Markov blanket *Given a feature* $F_i \in \mathbf{S}$ *(a set of attributes) and the class* $\mathbf{Y}$*, the subset* $\mathbf{M} \subseteq \mathbf{S}$ $(F_i \notin \mathbf{M})$ *is a Markov blanket of* F_i *if, given* $\mathbf{M}$*,* F_i *is conditionally independent of* $\mathbf{S} - \mathbf{M} - \{F_i\}$ *and* $\mathbf{Y}$*.*

Two attributes (or sets of attributes) X, Y are said to be conditionally

independent given a third attribute Z (or set) if, the given Z makes X and Y independent, i.e., the distribution of X, knowing Y and Z, is equal to the distribution X knowing Z; therefore, Y does not have influence on X ($P(X|Y,Z) = P(X|Z)$).

Theoretically, it can be shown that once we find a Markov blanket $\mathbf{M}$ of feature F_i in a feature set $\mathbf{S}$, we can safely remove F_i from $\mathbf{S}$ without increasing the divergence from the original distribution. Furthermore, in a sequential filtering process, in which unnecessary features are removed one by one, a feature tagged as unnecessary based on the existence of a Markov blanket $\mathbf{M}$ remains unnecessary in later stages when more features have been removed. The Markov blanket condition requires that $\mathbf{M}$ assumes not only the information that F_i has about $\mathbf{Y}$, but also about all the other features. In [14] it is stated that the cardinality of set $\mathbf{M}$ must be small and fixed.

References [20] and [21] are among the most cited works at present following the above-mentioned framework (FR+FSS). Both are based on this concept of Markov blanket. In the first one, the number of attributes of $\mathbf{M}$ is not provided, but it is a fixed number among the highly correlated features. In the second one, a fast correlation-based filter is implemented (FCBF), where $\mathbf{M}$ is formed by only one attribute, and gradually eliminates redundant attributes with respect to $\mathbf{M}$ from the first to the final attributes of an ordered list. Other methods based on relevance and redundancy concepts can be found in [10, 6].

8.4 Incremental Performance over Ranking

In this section, we introduce first our ideas of relevance and redundancy taking into account the aim of applying a wrapper model to massive datasets; second, changes introduced by the filter model; and then our approach is described.

As previously indicated, the wrapper model makes use of the algorithm that will build the final classifier to select a feature subset. Thus, given a classifier L, and given a set of features $\mathbf{S}$, a wrapper method searches in the space of $\mathbf{S}$, using cross-validation to compare the performance of the trained classifier L on each tested subset. While the wrapper model is more computationally expensive than the filter model, it also tends to find feature sets better suited to the inductive biases of the learning algorithm and therefore provides superior performance.

In this work, we propose a fast search over a minimal part of the feature space. Beginning with the first feature from the list ordered by some evaluation criterion, features are added one by one to the subset of selected features only if such inclusion improves the classifier accuracy. Then, the learning al-

gorithm of the wrapper approach is always run N (number of features) times, usually with a few features. A feature ranking algorithm makes use of a scoring function computed from the values of each feature and the class label. By convention, we assume that a high score is indicative of a valuable feature and that we sort features in decreasing order of this score. We consider ranking criteria defined for individual features, independently of the context of others.

When a ranking of features is provided from a high dimensional data set, a large number of features with similar scores is generated, and a common criticism is that it leads to the selection of redundant subsets. However, according to [8], noise reduction and consequently better class separation may be obtained by adding variables that are presumably redundant. Moreover, a very high attribute correlation (in absolute value) does not mean the absence of attribute complementarity. Therefore, our idea of redundancy is not based only on correlation measures, but also on the learning algorithm target (wrapper or filter approach), in the sense that a feature is chosen if additional information is gained by adding it to the selected subset of features.

8.4.1 Incremental Ranked Usefulness

In feature subset selection, it is a fact that two types of features are generally perceived as being unnecessary: features that are irrelevant to the target concept, and features that are redundant given other features. Our approach is based on the concept of a Markov blanket, which is described in [14]. This idea was formalized using the notion of conditionally independent attributes, which can be defined by several approaches [20, 21]. We set this concept by a wrapper model, defining incremental ranked usefulness in order to devise an approach to explicitly identify relevant features and do not take into account redundant features.

Let $\mathbf{X_L}$ be a sample of labeled data, $\mathbf{S}$ be a subset of features of $\mathbf{X_L}$, and L be a learning algorithm; the *correct rate* (or accuracy) $\Gamma(\mathbf{X_L}/\mathbf{S}, L)$ is named to the ratio between the number of instances correctly classified by L and the total number of evaluated instances considering only the subset $\mathbf{S}$. In the training process, this accuracy will be an estimate of error by cross-validation.

Let $R = \{F_i\}$, $i = 1 \ldots N$ be a ranking of all the features in $\mathbf{X_L}$ sorted in descending order, and $\mathbf{S}$ be named the subset of the i first features of R.

DEFINITION 8.3 Incremental ranked usefulness *The feature F_{i+1} in R is incrementally useful to L if it is not conditionally independent of the class $\mathbf{Y}$ given $\mathbf{S}$; therefore, the correct rate of the hypothesis that L produces using the group of features $\{F_{i+1}\} \cup \mathbf{S}$ is significantly better (denoted by $\succcurlyeq$) than the correct rate achieved using just the subset of features $\mathbf{S}$.*

Therefore, if $\Gamma(\mathbf{X_L}/\mathbf{S} \cup \{F_{i+1}\}, L) \not\succcurlyeq \Gamma(\mathbf{X_L}/\mathbf{S}, L)$, then F_{i+1} is conditionally independent of class $\mathbf{Y}$ given the subset $\mathbf{S}$, and then we should be able to omit

```
Input: X_L training U-measure, L-subset evaluator
Output: BestSubset
1  list R = {}
2  for each feature F_i ∈ X_L
3      Score = compute(F_i, U, X_L)
4      append F_i to R according to Score
5  BestEvaluation = 0
6  BestSubset = ∅
7  for i = 1 to N
8      TempSubset = BestSubset ∪ {F_i} (F_i ∈ R)
9      TempEvaluation = WrapperOrFilter(TempSubset, L)
10     if (TempEvaluation ≽ BestEvaluation)
11         BestSubset = TempSubset
12         BestEvaluation = TempEvaluation
```

FIGURE 8.1: *BIRS* algorithm.

F_{i+1} without compromising the accuracy of class prediction.

A fundamental question in the previous definition is how the *significant* improvement is analyzed in this wrapper model. A five-fold cross-validation is used to estimate if the accuracy of the learning scheme for a set of features is significantly better ($\succcurlyeq$) than the accuracy obtained for another set. We conducted a 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 (five folds), our search method uses a t-test. We want to obtain a heuristic, not to do an accurate population study. However, on the one hand, it must be noticed that it is a heuristic based on an objective criterion, to determine the statistical significance degree of difference between the accuracies 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 allow us to add more features, because it would be difficult for the test to obtain significant differences between the accuracy of each subset. Obviously, if the confidence level is increased, more features can be selected, and vice versa.

Following a filter model in the subset evaluation, we need a different way to find out if the value of measurement of a set is significantly better ($\succcurlyeq$) than another one when adding an attribute. Simply, it is verified if the improvement surpasses a threshold (for example, 0.005), one resulted from the best previous subset and the other resulted from the joint candidate.

TABLE 8.1: Example of feature selection process by *BIRS*.

Rank	F_5	F_7	F_4	F_3	F_1	F_8	F_6	F_2	F_9

	Subset Eval.	Acc	P-Val	Acc	Best Sub
1	F_5	80		**80**	$\mathbf{F_5}$
2	$\mathbf{F_5}$,F_7	82			
3	$\mathbf{F_5}$,F_4	81			
4	$\mathbf{F_5}$,F_3	83			
5	$\mathbf{F_5}$,F_1	84	$<$ **0.1**	**84**	$\mathbf{F_5}, \mathbf{F_1}$
6	$\mathbf{F_5}, \mathbf{F_1}$,$F_8$	84			
7	$\mathbf{F_5}, \mathbf{F_1}$,$F_6$	86			
8	$\mathbf{F_5}, \mathbf{F_1}$,$F_2$	89	$<$ **0.1**	**89**	$\mathbf{F_5}, \mathbf{F_1}, \mathbf{F_2}$
9	$\mathbf{F_5}, \mathbf{F_1}, \mathbf{F_2}$,$F_9$	87			

8.4.2 Algorithm

There are two phases in the algorithm, named *BIRS* (Best Incremental Ranked Subset), shown in Figure 8.1: Firstly, the features are ranked according to some evaluation measure (lines 1–4). In the second phase, we deal with the list of features once, crossing the ranking from the beginning to the last ranked feature (lines 5-12). We obtain the classification accuracy with the first feature in the list (line 9) and it is marked as selected (lines 10-12). We obtain the classification rate again with the first and second features. The second will be marked as selected depending on whether the accuracy obtained is significantly better (line 10). We repeat the process until the last feature on the ranked list is reached. Finally, the algorithm returns the best subset found, and we can state that it will not contain irrelevant or redundant features.

The first part of the above algorithm is efficient since it requires only the computation of N scores and to sort them, while in the second part, time complexity depends on the learning algorithm chosen. It is worth noting that the learning algorithm is run N (number of features) times with a small number of features, only the selected ones. Therefore, the running time of the ranking procedure can be considered to be negligible regarding the global process of selection. In fact, the results obtained from a random order of features (without previous ranking) showed the following drawbacks: 1) The solution was not deterministic; 2) a greater number of features were selected; 3) the computational cost was higher because the classifier used in the evaluation contained more features since the first iterations.

Consider the situation depicted in Table 8.1: an example of the feature selection process done by *BIRS*. The first line shows the features ranked according to some evaluation measure. We obtain the classification accuracy with the first feature in the list (F_5:80%). In the second step, we run the

classifier with the first two features of the ranking (F_5,F_7:82%), and a paired t-test is performed to determine the statistical significance degree of the differences. Since it is greater than 0.1, F_7 is not selected. The same happens with the next two subsets (F_5,F_4:81%, F_5,F_3:83%). Later, the feature F_1 is added, because the accuracy obtained is significantly better than that with only F_5 (F_5,F_1:84%), and so on. In short, the classifier is run nine times to select, or not, the ranked features (F_5,F_1,F_2:89%): once with only one feature, four times with two features, three with three features, once with four, and once with four, features. Most of the time, the learning algorithm is run with few features. In short, this wrapper-based approach needs much less time than others with a broad search engine.

As we can see in the algorithm, the first feature is always selected. This does not mean a great shortcoming in high-dimensional databases, because usually several different sets of features share similar information. The main disadvantage of *sequential forward generation* is that it is not possible to consider certain basic interactions among features, i.e., features that are useless by themselves can be useful together. *Backward generation* remedies some problems, although there still will be many hidden interactions (in the sense of being unobtainable), but it demands more computational resources than the forward approach. The computer-load necessities of the forward search might become very inefficient in high-dimensional domains, as it starts with the original set of attributes and removes features increasingly.

8.5 Experimental 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 *BIRS* is in situations where there is a large number of features and instances.

The comparison was performed with two representative groups of datasets: Twelve datasets were selected from the UCI Repository (Table 8.2) and five from the NIPS 2003 feature selection benchmark [9]. In this group (Table 8.3), the datasets 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 dataset is artificial. The input variables are continuous or binary, sparse or dense. All problems are two-class classification problems. The full characteristics of all the datasets are summarized in Tables 8.2 and 8.3. We chose three different learning algorithms: C4.5, IB1, and Naïve Bayes, to evaluate the accuracy on selected features for each feature selection algorithm.

Figure 8.2 can be considered to illustrate both blocks that always com-

TABLE 8.2: UCI Repository of Machine Learning Databases. For each dataset we show the acronym used in this text, the number of features, the number of examples, and the number of possible classes.

Data	Acron.	#Feat.	#Inst.	#Classes
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 8.3: NIPS 2003 challenge data sets. For each dataset we show the acronym used in this text, the domain it was taken from, its type (dense, sparse, or sparse binary), the number of features, the number of examples, and the percentage of random features. All problems are two-class classification problems.

Data	Acron.	Domain	Type	#Feat.	#Inst.	%Ran.
Arcene	ARC	Mass Spectro.	Dense	10000	100	30
Dexter	DEX	Text classif.	Sparse	20000	300	50
Dorothea	DOR	Drug discove.	S. bin	100000	800	50
Gisette	GIS	Digit recogn.	Dense	5000	6000	30
Madelon	MAD	Artificial	Dense	500	2000	96

pose algorithm *BIRS* (originally introduced in [21]). Therefore, this feature selection algorithm needs measures to evaluate individual and subsets of attributes. Numerous versions of selection algorithms *BIRS* could be formed combining the criteria of each group of measures (individual and subset). In order to simplify, we will use the same evaluation measure in the two phases (individual and subset). In the experiments, we used two criteria: one belongs to the wrapper model, and one to the filter model. 1) In the wrapper approach (denoted by BI_{NB}, BI_{C4}, or BI_{IB}) we order features according to their individual predictive power, using as criterion the performance of the target classifier built with a single feature. The same classifier is used in the second phase to evaluate subsets. 2) In the filter approach, a ranking is provided using a non-linear correlation measure. We chose symmetrical uncertainty (denoted by BI_{CF}), based on entropy and information gain concepts [11] in

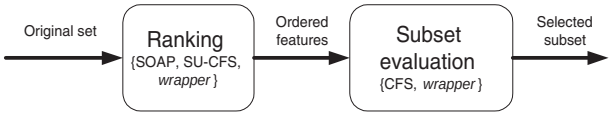

FIGURE 8.2: Type of feature evaluation in *BIRS*.

both phases. Note the similarity among the results obtained in previous works with several ranking measure approaches [18]. Accuracy differences are not statistically significant, although wrapper ranking is a little bit better.

Also in these experiments, to find out if the value of measurement of a set is significantly better ($\succcurlyeq$) than another one when adding an attribute, it is distinguished between filter and wrapper models in the subset evaluation. In the first case, it is simply verified if the improvement surpasses a threshold established in 0.005; nevertheless, in the second case, we conduct Student's paired two-tailed t-test in order to evaluate the statistical significance (at level 0,1) of the difference between two averaged accuracy values: one resulted from the joint candidate and the other resulted from the best previous subset.

Due to the high dimensionality of data, we limited our comparison to sequential forward (SF) techniques and a fast correlation-based filter (FCBF) algorithm [21] applied to the first group of datasets, and only FCBF with the NIPS datasets. We chose two representative subset evaluation measures in combination with the SF search engine. One, denoted by SF_{WR}, uses a target learning algorithm to estimate the worth of feature subsets; the other, denoted by SF_{CF}, is a subset search algorithm that exploits sequential forward search and uses the correlation measures (variation of the CFS correlation-based feature selection algorithm [11]) to guide the search.

The experiments were conducted using the WEKA's implementation of all these existing algorithms, and our algorithm is also implemented in the WEKA environment [19]. We must take into account that the proper way to conduct a cross-validation for feature selection is to avoid using a fixed set of features selected with the whole training dataset, because this induces a bias in the results. Instead, one should withhold a pattern, select features, and assess the performance of the classifier with the selected features using the leftout examples. The results reported in this section were obtained with a 5×2-fold cross-validation over each dataset, i.e., a feature subset was selected using the 50% of the instances; then, the accuracy of this subset was estimated over the unseen 50% of the data. In this way, estimated accuracies, selected attribute numbers, and time needed were the result of a mean over five executions of two cross-validation samples. We use two instead of ten cross-validations because of the time cost consuming with massive datasets. Standard methods have been used for the experimental section (sequential forward; Naïve Bayes, IB1, and C4.5 classifiers; and the t-Student statistical test). There exist other methods following the wrapper approach to extract

TABLE 8.4: Accuracy of NB on selected features for UCI data. The symbols $^+$ and $^-$ respectively identify statistically significant (at 0.05 level) wins or losses over BI_{NB}.

	Wrapper		Filter			Original
Data	BI_{NB}	SF_{NB}	BI_{CF}	SF_{CF}	FCBF	
ADS	95.42	95.83	95.38	95.81	95.64	96.38
ARR	66.99	67.70	66.50	68.05	63.98	60.13
HYP	95.10	95.32	94.15^-	94.15^-	94.90	95.32
ISO	83.30	82.28	77.61	80.79	74.62^-	80.42
KRV	94.27	94.32	90.43^-	90.43^-	92.50	87.50
LET	65.67	65.67	64.28^-	64.28^-	65.06	63.97
MUL	97.21	96.87	97.04	96.72	96.19	94.37
MUS	98.78	99.01	98.52	98.52	98.52	95.10
MUK	84.59	84.59	79.94	69.78^-	72.29	83.56
SIC	94.55	93.88	93.89	93.89	96.25	92.41
SPL	94.85	94.91	93.63^-	93.60^-	95.49	95.26
WAV	81.01	81.55	81.01	80.12	78.42^-	80.02
time(s)	6111	49620	49	133	68	

relevant features, which involve the selection process into the learning process (neural networks, Bayesian networks, support vector machines), although the source code of these methods is not freely available and therefore the experiments cannot be reproduced. In fact, some of them are designed for specific tasks, so the parameter settings are quite different for the learning algorithm.

Tables 8.4, 8.5, and 8.6 report accuracy by Naïve Bayes, IB1, and C4.5, respectively, by each feature selection algorithm and the original set. From the last row of each table, we can observe for each algorithm the running time. We conducted a Student's paired two-tailed t-test in order to evaluate the statistical significance of the difference between two averaged accuracy values: one resulted from the wrapper approach of $BIRS$ (BI_{NB}, BI_{C4} or BI_{IB}) and the other resulted from one of the wrapper version of SF (SF_{NB}, SF_{C4} or SF_{IB}), BI_{CF}, SF_{CF}, $FCBF$, and the original set. The symbols $+$ and $-$ respectively identify statistic significance, at 0.05 level, wins or losses over BI_{WR}.

We studied the behavior of BI_{WR} in three ways in Tables 8.4, 8.5, and 8.6: with respect to a whole set of features (last row, original); with respect to another wrapper approach (SF_{WR}); and with respect to three filter approaches (BI_{CF}, SF_{CF}, and $FCBF$).

As it is possible to be observed in the last column of Tables 8.4, 8.5, and 8.6, classification accuracies obtained with the wrapper approach of $BIRS$ (BI_{WR}) with respect to results obtained with the total set of attributes are statistically better in 4 and 3 occasions for classifiers NB and IB, respectively, and worse in 2 applying C4. Note that the number of selected attributes is drastically less than the original set, retaining on average 15% (NB, Ta-

TABLE 8.5: Accuracy of C4 on selected features for UCI data. The symbols $^+$ and $^-$ respectively identify statistically significant (at 0.05 level) wins or losses over BI_{C4}.

	Wrapper		Filter			Original
Data	BI_{C4}	SF_{C4}	BI_{CF}	SF_{CF}	FCBF	
ADS	96.55	96.85	96.43	96.39	95.85	96.46
ARR	68.01	67.39	66.42	67.04	64.87	64.29
HYP	99.07	99.30	96.56^-	96.56^-	98.03	99.36
ISO	69.43	N/D	72.68	71.94	66.63	73.38
KRV	95.11	94.26	90.43^-	90.43^-	94.07	99.07^+
LET	84.99	85.17	84.21^-	84.21^-	84.84	84.45
MUL	92.42	93.11	93.17	93.12	92.29	92.74
MUS	99.91	100.00^+	98.52^-	98.52^-	98.84^-	100.00^+
MUK	95.43	N/D	94.06	94.60	91.19^-	95.12
SIC	98.28	98.19	96.33^-	96.33^-	97.50	98.42
SPL	93.05	93.04	92.54	92.61	93.17	92.92
WAV	76.20	75.44	76.46	76.56	74.52	74.75
time(s)	17914	40098	49	133	68	

ble 8.4), 16.3% (C4, Table 8.5), and 13.1% (IB, Table 8.6) of the attributes. As we can see, BI_{WR} chooses less than 10% of the attributes in more than half of all the cases studied in these tables.

BI_{WR} versus SF_{WR}: No significant statistical differences are shown between the accuracy of our wrapper approach and the accuracy of the sequential forward wrapper procedure (SF_{WR}), except for the MUS dataset and C4 classifier (Table 8.5).

Notice that in two cases with C4 classifiers (ISO and MUK) and two with IBs (ADS and MUL), SF_{WR} did not report any results after three weeks running; therefore, there are no selected attributes or success rates. Without considering this lack of results with SF_{WR}, the chosen subset by $BIRS$ is considerably smaller with the IB classifiers, 13.1% versus 20%, and less difference with NB and C4, although it is supposed that the lack of results would favor $BIRS$, since SF has not finished because of the inclusion of many attributes.

On the other hand, the advantage of $BIRS$ with respect to the SF for NB, IB1, and C4.5 is clear having to take into account the running time needed. $BIRS$ takes 6,112, 5,384, and 21,863 seconds applying NB, C4, and IB, respectively, whereas SF takes 49,620, 40,098, and 210,642 seconds. We can observe that $BIRS$ is consistently faster than SF_W, because the wrapper subset evaluation is run less times. For example, for the ADS dataset and C4.5 classifier, $BIRS$ and SF retain 8.5 and 12.4 features, respectively, on average. To obtain these subsets, the first one evaluated 1,558 features individually (to generate the ranking) and 1,558 subsets, while the second one evaluated 18,630 subsets (1,558 features + 1557 pairs of features + ... + 1,547 sets of

TABLE 8.6: Accuracy of IB on selected features for UCI data. The symbols $^+$ and $^-$ respectively identify statistically significant (at 0.05 level) wins or losses over BI_{IB}.

	Wrapper		Filter			Original
Data	BI_{IB}	SF_{IB}	BI_{CF}	SF_{CF}	FCBF	
ADS	95.28	N/D	95.93	96.07	95.75	95.95
ARR	62.74	57.12	61.37	61.06	58.67	54.12
HYP	83.66	83.57	85.75	85.75	94.88	90.85
ISO	80.64	78.61	79.37	80.28	72.57^-	77.58
KRV	92.27	94.24	90.43	90.43	93.85	89.21
LET	95.52	95.58	93.62^-	93.62^-	94.81	94.23^-
MUL	96.72	N/D	97.54	97.70	97.53	97.52
MUS	98.36	99.99	98.52	98.52	98.88	100.00
MUK	93.34	94.72	92.59	93.17	89.04^-	95.14
SIC	96.55	97.05	94.73	94.73	95.82	95.58
SPL	86.35	85.62	86.40	86.34	79.21^-	73.74^-
WAV	76.39	77.18	78.89^+	78.72	71.76^-	73.42^-
time(s)	40253	210642	49	133	68	

TABLE 8.7: Number of features selected by each feature selection algorithm on UCI data. Last row shows number of features retained on average. N - number of features of the original set, N' - number of features selected.

	Wrapper						Filter		
Data	BI_{NB}	SF_{NB}	BI_{C4}	SF_{C4}	BI_{IB}	SF_{IB}	BI_{CF}	SF_{CF}	FCBF
ADS	10.5	16.4	8.5	12.4	5.2	N/A	6.7	9.2	83.1
ARR	5.8	8.4	6.7	8.6	14.1	12.7	11.4	17.2	8.0
HYP	4.6	8.5	4.2	5.9	1.0	1.0	1.0	1.0	5.3
ISO	68.5	29.0	22.5	N/A	35.5	29.4	68.8	95.2	22.9
KRV	5.0	5.2	6.2	4.9	6.5	10.0	3.0	3.0	6.5
LET	11.0	11.6	11.0	10.1	10.9	11.0	9.0	9.0	10.3
MUL	22.2	15.3	20.6	13.6	11.3	N/A	28.0	90.3	121.3
MUS	2.1	3.0	4.1	4.9	1.6	4.7	1.0	1.0	3.6
MUK	1.0	1.0	9.7	N/A	4.7	12.0	6.5	16.3	2.9
SIC	2.4	1.0	5.9	5.5	2.8	6.7	1.0	1.0	4.8
SPL	13.1	14.8	9.8	11.0	5.9	6.6	6.0	6.1	21.8
WAV	9.4	12.9	9.6	7.9	10.0	12.4	12.4	14.8	6.1
$\frac{N'}{N} * 100$	15.0	16.8	16.3	18.2	13.1	20.3	11.7	14.1	18.1

twelve features). The time savings of $BIRS$ became more obvious when the computer-load necessities of the mining algorithm increased. In many cases, the time savings were 10 times less, and we must take into account that SF did not report any results on several datasets.

These results verify the computational efficiency of incremental searches applied by $BIRS$ over greedy sequential searches used by SF, with a lower number of features selected and without significant statistical differences on accuracy.

***BIRS* wrappers versus filters:** We noticed that the computer-load necessities of filter procedures can be considered as negligible regarding wrapper models. Nevertheless, wrapper approaches of $BIRS$ (BI_{WR}) obtained better accuracies: They showed significant gains to the filter version of $BIRS$, ${}_{CF}BI_{CF}$, in 4, 5, and 1 cases for NB, C4, and IB respectively, and they only lost in one with IB; with respect to the sequential version SF_{CF}, $BIRS$ won in 5, 5, and 1 occasions for NB, C4, and IB, respectively; and with respect to $FCBF$, BI_{WR} was better in 2, 2, and 4 cases with each respective classifier.

Table 8.7 reports the number of features selected by each feature selection algorithm on UCI data, showing three different results for each wrapper approach, depending on the learning algorithm chosen. Obviously, there is one value for filter approaches because filters do not depend on the classifier used. From the last row, we can observe for each algorithm the number of features retained on average. The filter approach of $BIRS$ retains less attributes than the rest of the algorithms. BI_{CF} retains 11.7% of the attributes on average for the 12 databases, SF_{CF} retains 14.1% of the attributes on average for all datasets, whereas $FCBF$ retains 18.1%.

We used the WEKA implementation of the $FCBF$ algorithm with default values. However, if the threshold by which features can be discarded is modified, the results obtained might vary. Note that if this threshold is set to the upper value, the number of selected features diminishes considerably, together with a notable reduction of prediction.

Another comparison can be between the versions filters, that is to say, as the approach behaves filter of $BIRS$ (BI_{CF}) with respect to the sequential search SF_{CF} and to the $FCBF$ algorithm. About accuracies, results obtained with both ($BIRS$ and SF) first are similar and a little less than those obtained with $FCBF$. Nevertheless, the most reduced datasets are obtained with the filter model of $BIRS$. In addition, the time needed to reduce each dataset with BI_{CF} was faster than the others.

NIPS datasets: Table 8.8 shows the results obtained by the three classifiers, Naïve Bayes (NB), C4.5 (C4), and IB1 (IB), from the NIPS 2003-*Neural Information Processing Systems* (Table 8.3) feature selection benchmark data. The table gives the accuracy and number of features selected by each feature selection algorithm and the original set. We conducted a Student's paired

TABLE 8.8: *BIRS* accuracy of Naïve Bayes (NB), C4.5 (C4), and IB1 (IB) on selected features for NIPS data: Acc records 5×2CV classification rate (%) and #Att records the number of features selected by each algorithm. The symbols $^+$ and $^-$ respectively identify statistically significant (at 0.05 level) wins or losses over BI_{WR}.

	Data	BI_{WR}		BI_{CF}		FCBF		Original
		Acc	#Att	Acc	#Att	Acc	#Att	
NB	ARC	64.60	15.3	63.20	39.2	61.20	35.2	65.40
	DEX	81.33	30.2	82.47	11.3	85.07	25.1	86.47
	DOR	93.23	10.5	93.80	11.9	92.38	75.3	90.68^-
	GIS	92.66	35.3	90.83	11.6	87.58^-	31.2	91.88
	MAD	59.00	11.8	60.56	5.8	58.20	4.7	58.24
C4	ARC	65.80	7.9	59.00	39.2	58.80	35.2	57.00
	DEX	80.27	18.9	81.47	11.3	79.00	25.1	73.80
	DOR	92.13	7.2	91.63	11.9	90.33	75.3	88.73
	GIS	93.29	26.9	90.92	11.6	90.99^-	31.2	92.68
	MAD	73.02	17.0	69.77	5.8	61.11^-	4.7	57.73^-
IB	ARC	69.00	15.1	68.60	39.2	62.00	35.2	78.00
	DEX	81.00	34.1	81.73	11.3	79.20	25.1	56.67^-
	DOR	92.18	3.5	90.98	11.9	90.35	75.3	90.25
	GIS	82.25	2.3	90.07	11.6	90.06	31.2	95.21
	MAD	74.92	14.4	71.59	5.8	56.90	4.7	54.39

two-tailed t-test in order to evaluate the statistical significance of the difference between two averaged accuracy values: one resulted from BI_{WR} (BI_{NB}, BI_{C4}, or BI_{IB}) and the other resulted from one of BI_{CF}, $FCBF$, and the original set. The symbols + and − respectively identify statistic significance, at 0.05 level, wins or losses over BI_{WR}. Results obtained with *SF* algorithms are not shown. The wrapper approach is too expensive in time, and its filter approach selects so many attributes that the program ran out of memory after a long period of time due to its quadratic space complexity. On the other hand, the *CFS* algorithm has been modified to be able to obtain results with *BIRS* for the DEX and DOR databases. From Table 8.8 we can conclude the following:

- *BIRS* is a good method to select attributes, because with a very reduced set of attributes one can obtain similar results, even better, than with the whole set of features in a massive database. About accuracies obtained by the wrapper model of *BIRS*, it excels specially when the C4 classifier is applied, winning in four of the five datasets; with the NB classifier, *BIRS* obtains good results on the DEX dataset; and applying IB, it loses in ARC and GIS, but nevertheless wins by approximately 20 points in the DEX and MAD datasets. In all the cases, the reduction obtained with respect to the original data is drastic, emphasizing that obtained with the DOR dataset, where approximately 0.01% of the attributes (10

of 100,000) is always retained.

- The behavior of the filter approach of *BIRS* is excellent. It produces rates of successes similar to the wrapper approach, with the number of attributes equal or even lower. Note that the number of attributes in filter approaches does not depend on the classifier applied.
- If we study the comparison between *BIRS* approaches and the *FCBF* algorithm, it can be verified that, except for the DEX dataset with an NB classifier, the accuracies obtained applying *FCBF* are normally below those obtained applying *BIRS*, emphasizing the existing differences for MAD dataset with a C4 classifier, and for ARC and MAD datasets with IB. The subsets selected by *FCBF* are greater than those chosen by *BIRS* on average, however, the time cost is approximately six times less.

8.6 Conclusions

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. Traditional feature selection methods often select the top-ranked features according to their individual discriminative powers. However, the inclusion of irrelevant, redundant, and noisy features in the model building process phase can result in poor predictive performance and increased computation. The most popular search methods in machine learning cannot be applied to massive datasets, especially when a wrapper approach is used as an evaluation function. We use the incremental ranked usefulness definition to decide at the same time whether or not a feature is relevant and non-redundant. The technique extracts the best non-consecutive features from the ranking, trying to avoid the influence of unnecessary features in further classifications.

Our approach, named *BIRS*, uses a very fast search through the attribute space, and any subset evaluation measure, the classifier approach included, can be embedded into it as an evaluator. Massive datasets take a lot of computational resources when wrappers are chosen. *BIRS* reduces the search space complexity as it works directly on the ranking, transforming the combinatorial search of a sequential forward search into a quadratic search. However, the evaluation is much less expensive as only a few features are selected, and therefore the subset evaluation is computationally inexpensive in comparison to other approaches involving wrapper methodologies.

In short, our technique *BIRS* chooses a small subset of features from the original set with similar predictive performance to others. For massive

datasets, wrapper-based methods might be computationally unfeasible, so *BIRS* turns out to be a fast technique that provides good performance in predicting accuracy.

Acknowledgment

The research was supported by the Spanish Research Agency CICYT under grant TIN2004-00159 and TIN2004-06689-C03-03.

References

[1] D. Bell and H. Wang. A formalism for relevance and its application in feature subset selection. *Machine Learning*, 41(2):175–195, 2000.

[2] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

[3] R. A. Caruana and D. Freitag. How useful is relevance? In *Working Notes of the AAAI Fall Symp. on Relevance*, pages 25–29, 1994.

[4] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analisys*, 1(3):131–56, 1997.

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

[6] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. In *IEEE Computer Society Bioinformatics*, pages 523–529, IEEE PRess, Poscataway, NJ, 2003.

[7] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–37, 1999.

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

[9] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. Result analysis of the nips 2003 feature selection challenge. In *Advances in Neural Information Processing Systems*, pages 545–552. MIT Press, Cambridge, MA, 2005.

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

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

[12] M. A. Hall and G. Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Eng.*, 15(3), 2003.

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

[14] D. Koller and M. Sahami. Toward optimal feature selection. In *13th Int. Conf. on Machine Learning*, pages 284–292, 1996.

[15] P. Langley. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*, pages 140–144, 1994.

[16] H. Liu and H. Motoda. *Feature Selection for Knowlegde Discovery and Data Mining*. Kluwer Academic Publishers, London, UK, 1998.

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

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

[19] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, CA, 2005.

[20] E. P. Xing, M. I. Jordan, and R. M. Karp. Feature selection for high-dimensional genomic microarray data. In *Proc. 18th Int. Conf. on Machine Learning*, pages 601–608. Morgan Kaufmann, San Francisco, CA, 2001.

[21] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–24, 2004.