Fast feature selection aimed at high-dimensional data via hybrid-sequential-ranked searches

R. Ruiz, J.C. Riquelme, J.S. Aguilar-Ruiz, M. García-Torres

ABSTRACT

We address the feature subset selection problem for classification tasks. We examine the performance of two hybrid strategies that directly search on a ranked list of features and compare them with two widely used algorithms, the fast correlation based filter (FCBF) and sequential forward selection (SFS). The pro-posed hybrid approaches provide the possibility of efficiently applying any subset evaluator, with a wrap-per model included, to large and high-dimensional domains. The experiments performed show that our two strategies are competitive and can select a small subset of features without degrading the classifica-tion error or the advantages of the strategies under study.

1. Introduction

During the last decade, the motivation for applying feature selection (FS) techniques has shifted from being an optional subject to becoming a real prerequisite for model building. The main reason for this change is the high-dimensional nature of many modeling tasks in fields such as bioinformatics (Armañanzas et al., 2011; García-Torres et al., in press), materials (Pérez-Benítez & Padovese, 2011), text mining (Azam & Yao, 2011; Chen, Huang, Tian, & Qu, 2008), etc. The selection of features and the removal or reduction of redundant information unrelated to the classifica-tion task on hand will not only reduce the complexity of the prob-lem and improve the efficiency of the processing but will also simplify significantly the design of the classifier. The FS is one of the essential and frequently used techniques in machine learning (Arauzo-Azofra, Aznarte, & Benítez, 2010; Foithong, Pinngern, & Attachoo, 2011; García-López, García-Torres, Melián-Batista, Moreno-Périez, & Moreno-Vega, 2006; García-Torres, García-López, Melián-Batista, Moreno-Pérez, & Moreno-Vega, 2004; Kabir, Shahjahan, & Murase, 2011; Pacheco, Casado, & Núnez, 2007; Yang, Liao, Meng, & Lee, 2011). An FS method generates different candidates from the feature space and assesses them based on an evaluation criterion to find the best feature subset (Dash & Liu, 1997).

On the basis of the evaluation criterion, FS can be divided into filter methods and wrapper methods. Filters assess the relevance of features by looking only at the intrinsic properties of the data, such as distance, consistency, and correlation (Dash & Liu, 1997; Dash & Liu, 2003; Hall, 2000). These criteria are independent of any inductive learning algorithm. In contrast, the wrapper approach requires one predetermined mining algorithm and uses its performance to evaluate and determine which features are selected (Kohavi & John, 1997). Wrappers often select features that have a higher accuracy; however, they are criticized for their high computational cost and low generality. To take advantage of the above two approaches, a hybrid model was proposed to handle large data sets (Bermejo, Gámez, & Puerta, 2008; Das & Filters, 2001; Xing, Jordan, & Karp, 2001). Moreover, some methods, known as embedded, use internal information of the classification model to perform FS (Guyon & Elisseeff, 2003; Saeys, Abeel, & de Peer, 2008).

Based on the generation procedure, FS can be divided into individual feature ranking (FR) and feature subset selection (FSS) (Blum & Langley, 1997; Guyon & Elisseeff, 2003). FR measures the relevance of each feature to the class and then ranks features by their scores and selects the top-ranked features. These methods are widely used because of their simplicity, scalability, and good empirical success (Guyon & Elisseeff, 2003; Golub et al., 1999). However, FR is criticized because it can capture only the relevance of the features to the target concept, whereas the redundancy and basic interactions between features are not discovered. Additionally, the number of features retained is difficult to determine; as a result, a threshold is required. In contrast, FSS attempts to find a set of features that have good performance. This method integrates the metric for measuring the feature-class relevance and the feature–feature interactions. In (Liu & Yu, 2005), a large number of selection methods are categorized, in which different algorithms address these issues distinctively. We found different search strategies, namely exhaustive, heuristic and random searches, and combined them with several types of measures to form different algorithms. The time complexity is exponential in terms of the data dimensionality for an exhaustive search, and it is quadratic for a heuristic search. The complexity can be linear with the number of iterations in a random search, but experiments show that, to find the best feature subset, the number of iterations required is usually at least quadratic to the number of features (Dash, Liu, & Motoda, 2000). In this categorization, to handle large data sets, a hybrid model was also proposed to take advantage of the above two approaches (FR, FSS). These methods decouple relevance analysis and redundancy analysis, and they have been proven to be more effective than ranking methods and more efficient than subset evaluation methods on many traditional high-dimensional data sets. In this framework, (Yu & Liu, 2004) proposed a fast correlation-based filter algorithm (FCBF) that used a correlation measure to obtain relevant features and to remove redundancy. Recursive Feature Elimination (RFE) is a proposed FS algorithm described by Guyon, Weston, Barnhill, and Vapnik (2002) that works by choosing the *r* features which lead to the largest margin of class separation, using an SVM classifier. Ding and Peng (2003) uses mutual information for gene selection, finding maximum relevance with minimal redundancy by solving a simple two-objective optimization. In another method, (Hall & Holmes, 2003) proposes a rank search method to compare FS algorithms, but this method is not an efficient way to select a subset of features, especially in high-dimensional domains.

In this work, we present two FS methods that are based on a hybrid model, and we attempt to take advantage of all of the different approaches by exploiting their best performances in two steps: first, features are evaluated individually, providing a ranking based on a filter or wrapper criteria; second, a feature subset evaluator (filter or wrapper) is applied to a certain number of features in the previous ranking, following a search strategy. This approach provides the possibility of efficiently applying any subset evaluator, wrapper model included, in large and high-dimensional domains, obtaining a few features with high predictive power. The final subset is obviously not the optimum, but it is not feasible to search for every possible subset of features through the search space. Thus, in these types of domains, feature selection is more than necessary, it is indispensable. The remainder of this paper is structured as follows. Section 2 provides notions of feature relevance and redundancy and introduces our concept of incremental ranked usefulness. Subsequently, the algorithms are described. Experimental results are shown in Section 3, and the most interesting conclusions are summarized in Section 4.

2. Hybrid-generation feature selection

2.1. Introduction

In feature subset selection, two types of features are usually perceived as being unnecessary: features that are irrelevant to the target concept and features that are redundant given the other features.

In contrast, 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 with common agreement (Bell & Wang, 2000). The study in (Kohavi & John, 1997) includes three disjointed categories of feature relevance: strong relevance, weak relevance and irrelevance. The study in (Bell & Wang, 2000) makes use of Information Theory concepts to define the entropic or variable relevance of a feature with respect to the class, whereas (Blum & Langley, 1997) collects several relevance definitions.

Notions of feature redundancy are typically 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 linear case, the Pearson correlation coefficient is used, and in the non-linear case, many measures are based on the concept of entropy or on a measure of the uncertainty of a random variable. Symmetrical uncertainty (SU) (Press, Flannery, Teukolsky, & Vetterling, 1988) is frequently used, which is defined as follows:

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

where $H(X) = -\sum_i P(f_i) log_2(P(f_i))$ 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. Both of these measures are between pairs of variables. However, they may not be as straightforward when determining feature redundancy when one is correlated with a set of features. The study in (Koller & Sahami, 1996) applies a technique that is based on cross-entropy, named Markov blanket filtering, to eliminate redundant features. This concept was formalized with the notion of a conditionally independent attribute that can be defined by several approaches (Xing et al., 2001; Yu & Liu, 2004).

When databases with many features are ranked, there are usually many features with similar scores. The frequent selection of redundant features in the final subset is often criticized. However, according to (Guyon & Elisseeff (2003)), accounting for 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 could be a filter or a 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 it is rejected in the opposite case because the information provided is already contained (redundant) in the previous subset.

2.2. First hybrid subset generation

This first approach considers the relevance, and the redundancy concepts are included in the following "incremental usefulness" definition by Caruana and Freitag (1994): Given a sample of data, an evaluation measure 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 evaluation of the hypothesis that L produces using the group of features $\{F_i\} \cup \mathbf{S}$ is better than the evaluation achieved using just the subset of features $\mathbf{S}$. In other words, if the feature F_i is not incrementally useful to L with respect to $\mathbf{S}$, then the evaluation value given the subset $\mathbf{S}$ is equal to or better than the known subset evaluation result $\{F_i\} \cup \mathbf{S}$. This scenario suggests that if F_i gives no information beyond what is already in $\mathbf{S}$, then F_i can be safely removed. However, because the computational complexity for determining all of the possible interactions between the features is very high (mainly in high-dimensional domains), we consider using a guided search over an ordered list of attributes.

We present a heuristic to select features by means of a modification of the incremental usefulness concept. We address the incremental ranked usefulness to devise an approach to explicitly identify relevant features, and we do not take into account redundant features. The idea behind this technique is to choose features from a ranked list one by one in the following way:

1. First, the features are ranked according to an evaluation measure.
2. Second, we address the list of features once, crossing the ranking from the beginning to the last ranked feature.
   (a) First, we emphasize the possibility of using any subset evaluator, filter or wrapper. We obtain the evaluation results with the first feature in the list, and it is marked as selected.
   (b) We obtain a new result in the same way but using the first and second features. The second will be marked as selected depending on whether the evaluation obtained is significantly better.
   (c) Repeat the process with the remaining features until the last feature on the ranked list is reached.
   (d) Finally, the algorithm returns the best subset that was found, and we can state that it will not contain irrelevant or redundant features.

The first part of the above algorithm is efficient because it requires only the computation of N scores and to sort them; whereas in the second part, the time complexity depends on the learning algorithm that was chosen. It is worthwhile to note that the learning algorithm is run N (number of features) times with a small number of features: only the selected ones are used. Therefore, the running time of the ranking procedure can be considered to be negligible with respect to the global process of selection. In fact, the results obtained from a random order of features (without a 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 contains more features starting with the initial iterations.

A fundamental question is how a *significant* improvement is analyzed. In the wrapper model, a fivefold cross-validation is used to estimate whether the accuracy of the learning scheme for a set of features is significantly better than the accuracy obtained for another set. We conducted a paired, two-tailed Student?s t-test to evaluate the significance (at the 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 significantly increase the classification rate are chosen. Although the size of the sample is small (fivefold), our search method uses a t-test. We want to obtain a heuristic, not to perform an accurate population study. However, it must be noted that it is a heuristic based on an objective criterion for the purpose of determining the statistical significance level of the difference between the accuracies of each subset. However, the confidence level has been relaxed from 0.05 to 0.1 because of 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, then more features can be selected, and vice versa. Following a filter model in the subset evaluation, we need a different way to find out whether the value of the measurement of a set is significantly better than another set when adding an attribute. This criterion verified whether the improvement surpasses a threshold (for example, 0.005); one of the compared alternatives resulted from the best previous subset and the other resulted from the joint candidate. Consider an example of the feature selection process performed by our approach whereby a wrapper model is used:

1. First, the features are ranked according to the wrapper evaluation of each individual feature. We have the following ranking: $f_5, f_7, f_4, f_3, f_1, f_8, f_6, f_2, f_9$.
2. Subset selection:
   (a) Then, we take the classification accuracy with the first feature in the list (F_5:80%).
   (b) In the next 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 level of the differences. In this case, we suppose that if the p-value is greater than 0.1, then F_7 is not selected.
   (c) The same scenario occurs 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.
   (d) Finally, the algorithm returns the best subset found.

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 times with three features and once with four features. Most of the time, the learning algorithm is run with few features. In short, this wrapper-based approach requires much less time than other approaches that utilize a broad search engine.

As we can see in the algorithm, the first feature is always selected. This circumstance does not result in a large 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 many hidden interactions (in the sense of being unobtainable) will remain, but this approach demands more computational resources than does the forward approach. The computer-load necessities of the backward search could become very inefficient in high-dimensional domains because it starts with the original set of attributes and removes features increasingly.

2.3. Second hybrid subset generation

As in the previous subset generation, this method begins by generating a ranking, followed by the union of feature subsets by means of a down-top ranked-strategy, until subsequent feature subset combinations do not produce any better subsets. Once again, we emphasize the possibility of using any subset evaluator, filter or wrapper in both steps of this approach:

1. Step one generates a feature ranking that ranges from best to worst according to a specific evaluation measure.
2. Next, a list of solutions is generated in such a way that a solution for each individual feature is created, and the same ranking order is maintained. This second hybrid search consists of making a subset of relevant features by joining subsets with a lower number of features. With every iteration, a new list of solutions from the previous structure is generated. 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 is applied to it, gives back a higher measure value than the value obtained with the best (or first) subset from the previous list of solutions. To prevent the algorithm from becoming prohibitively time-consuming, new sets of features are generated by joining the first sets to the remaining previous list of solutions in the following way:
   (a) 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.
   (b) Next, the second set of the list is joined to the third set, the second set is joined to the fourth set, and so on until the last set on the list.

(c) This process of combining a set of features with the remaining sets on the list is conducted with the best k feature sets from the previous list of solutions.

(d) The process ends when combining the subsets no longer causes an improvement and returns the best-positioned feature subset of all of the subsets that were evaluated.

Consider an example of the feature selection process performed by this reduction process, where a filter model is used as an evaluator:

1. An initial feature ranking is generated. In this case, a filter measure is used as an individual evaluator (it could be a correlation measure), obtaining the following: $f_1, f_7, f_4, f_5, f_2, f_3, f_6, f_9, f_8$.
2. The evaluation of the first feature of the previous ranking (f_1) is used to set the limit. In this example, we use the same filter measure to evaluate a subset of attributes. The threshold is set at 0.167, which is obtained by applying the evaluator to the feature f_1.
3. Next, considering $k = 3$, subsets of features are generated with the first three features of the previous ranking (f_1,f_7,f_4) and the following features in the ranking, and they are evaluated with the filter. The sets with the evaluation in bold type have passed the threshold that was set beforehand with a feature (0.167):
   (a) With feature f_1 the following combinations are obtained: $(f_1,f_7-\mathbf{0.261})$, $(f_1,f_4-\mathbf{0.237})$, $(f_1,f_5-0.083)$, $(f_1,f_2-0.083)$, $(f_1,f_3-\mathbf{0.202})$, $(f_1,f_6-\mathbf{0.179})$, $(f_1,f_9-0.083)$, $(f_1,f_8-0.083)$
   (b) With feature f_7: $(f_7,f_4-\mathbf{0.289})$, $(f_7,f_5-0.123)$, $(f_7,f_2-0.123)$, $(f_7,f_3-\mathbf{0.234})$, $(f_7,f_6-\mathbf{0.230})$, $(f_7,f_9-0.123)$, $(f_7,f_8-0.123)$
   (c) And with feature f_4: $(f_4,f_5-0.101)$, $(f_4,f_2-0.101)$, $(f_4,f_3-\mathbf{0.237})$, $(f_4,f_6-\mathbf{0.198})$, $(f_4,f_9-0.101)$, $(f_4,f_8-0.101)$

Ranking the subsets that have improved compared with the previous best subset $(f_1-0.167)$ leaves the following: $(f_7,f_4-0.289)$ $(f_1,f_7-0.261)$, $(f_1,f_4-0.237)$, $(f_4,f_3-0.237)$, $(f_7,f_3-0.234)$, $(f_7,f_6-0.230)$, $(f_1,f_3-0.202)$, $(f_4,f_6-0.198)$, $(f_1,f_6-0.179)$

4. The evaluation of the first subset in the ranking produces the new threshold $(f_7,f_4-0.289)$. Once again, subsets are made with the three first sets of the last ranking generated ((f_7,f_4), (f_1,f_7), (f_1,f_4)) with the remaining pairs, and they are evaluated with a filter measure. As in the previous step, the subsets that pass a new limit (0.289) are in bold type:
   (a) The combinations given below are obtained with the set (f_7,f_4): $(f_7,f_4,f_1-\mathbf{0.296})$, $(f_7,f_4,f_1-0.296)$, $(f_7,f_4,f_3-0.289)$, $(f_7,\ f_4,f_3-0.289)$, $(f_7,f_4,f_6-0.273)$, $(f_7,f_4,f_1,f_3-\mathbf{0.301})$, $(f_7,f_4,f_6-0.273)$, $(f_7,f_4,f_1,f_6-0.287)$
   (b) With the set (f_1,f_7): $(f_1,f_7,f_4-0.296)$, $(f_1,f_7,f_4,f_3-0.301)$, $(f_1,f_7,f_3-0.261)$, $(f_1,f_7,f_6-0.255)$, $(f_1,f_7,f_3-0.261)$, $(f_1,f_7,f_4,f_6-0.287)$, $(f_1,f_7,f_6-0.255)$
   (c) And with (f_1,f_4): $(f_1,f_4,f_3-0.264)$, $(f_1,f_4,f_7,f_3-0.301)$, $(f_1,f_4,f_7,\ f_6-0.287)$, $(f_1,f_4,f_3-0.264)$, $(f_1,f_4,f_6-0.234)$, $(f_1,f_4,f_6-0.234)$

Ranking the subsets that pass the current threshold (0.289), we have the following: $(f_7,f_4,f_1,f_3-0{,}301)$, $(f_7,f_4,f_1-0{,}296)$

5. In the next step of this example, the limit is set at 0.301, which is not surpassed by any combination of subsets in the remaining ranking. Therefore, the process ends because the new list of solutions is empty. Therefore, the selected subset will be the one that occupies the first position of the last ranking of the feature subsets.

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.

3. Experiments and results

The aim of this section is to evaluate our approaches in terms of classification accuracy, degree of dimensionality and speed in selecting features, to see how good our two hybrid generation approaches ($H1$ and $H2$) are in situations where there is a large number of features and instances. We must consider that these proposals are search methods that are applied in a preprocessing phase, designed for the sake of the subsequent data analysis and classification; thus, the quality of these preprocessing methods must be investigated indirectly by their final performances in the data classification. However, it is well known that the performance of classification depends not only on the adopted preprocessing method but also on the properties of the data to be classified and the adopted classification method. To make the experiments objectively reflect the performances of the different search methods, we have carefully considered the representativeness of the selected data and the classification method. The data and methods used in the experiments, as well as in the experimental results, are described below.

Experiments were performed over three groups of data sets: Twelve data sets were selected from the University of California Irvine (UCI) Repository (Frank & Asuncion, 2010), five from the Neural Information Processing Systems (NIPS) 2003 feature selection benchmark (Guyon, Gunn, Ben-Hur, & Dror, 2005), and four data sets are microarrays related to cancer prediction. As can be seen in Table 1, these data sets are characterized by a large number of features and/or a large number of instances. First, from the well-known UCI Machine Learning collection of databases, we choose some of the biggest data sets from different domains (e.g., health, gene, Internet, mushrooms, waveform). Second, the NIPS 2003 workshops included a feature selection challenge, in which participants were provided with five data sets from different application domains (cancer prediction from mass spectrometry data, handwritten digit recognition, text classification, prediction of molecular activity, and one artificial data set). The input variables are continuous or binary, sparse or dense, and all of the data sets are two-class classification problems. Finally, four publicly available gene microarray datasets: (1) colon (Alon et al., 1999), with the expression levels of human genes from colon tissue samples; (2) a leukemia data set (Golub et al., 1999) that contains samples with

Table 1
Data sets Acron – acronym, Atts – number of attributes, Inst – number of instances.

Repository	Data	Acron.	#Atts.	#Inst.	#Class.
UCI	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
NIPS	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
BIO	Colon	COL	2000	62	2
	Leukemia	LEU	7129	72	2
	Lymphoma	LYM	4026	96	9
	GCM	GCM	16063	190	14

malignant neoplasms of hematopoietic stem cells; (3) lymphoma data (Alizadeh et al., 2000) comprising samples with nine different subtypes of lymphoma; and (4) Global Cancer Map (GCM) (Ramaswamy et al., 2000), containing samples divided into fourteen varieties of tumor.

In view of their maturity and properties, we selected the following widely used learning algorithms for our experiments to evaluate the accuracy of the selected features: C4.5 (*C*4) and Naïve Bayes (*NB*). These are two very representative methods in pattern recognition: *C*4 generates decision trees to classify instances, whereas *NB* classifies instances based on the Bayes' theorem.

As we stated previously, our hybrid-sequential-ranked algorithms always contain two blocks, which require a ranking and a feature subset evaluation measure. Several versions of the hybrid-generation selection algorithms could be made by combining the criteria for each group of measures (individual and subsets). To simplify, in the experiments made for the two approaches, the same evaluation measure was used to prepare the ranking and for the measure used in the second part of the algorithm for the feature subset search. To clarify the components that each approach uses in each case, a superscript is placed after *H*1 or *H*2 that indicates the evaluator used in the two phases. Two types of subset evaluation measures are used, one for each type of approach: (1) wrapper, the subsequence classification method, *NB* or *C*4.5; and (2) filter, CFS – correlation-based feature selection algorithm (Hall, 2000). For example, $H1^{CF}$ shows that *CFS* will be used as an individual measure in the first part and *CFS* will be used as a subset in the second part, and $H1^{NB}$ shows that the *NB* classifier will be used in both parts of the algorithm. As in the example of SubSection 2.3, the *H*2 parameter *k* was set to 3.

Because of the high dimensionality of the data, we limited our comparison to the sequential forward (SF) technique and the fast correlation-based filter (*FCBF*) algorithm (Yu & Liu, 2004). On the one hand, *SF*, also called the hill climbing or greedy search, looks for the best single attribute, then tries each of the remaining attributes in conjunction with the best to find the most suited pair, and continues in this way until no improvement is obtained when adding a new attribute. We chose two representative subset evaluation measures in combination with the *SF* search engine. One, denoted by SF^{NB} or SF^{C4}, uses a target learning algorithm to estimate the worth of the feature subsets; the other, denoted by SF^{CF}, is a subset search algorithm that exploits the sequential forward search and uses correlation measures (*CFS*) to guide the search. By contrast, *FCBF* is a filter approach that uses a correlation measure to obtain relevant features and to remove redundancy.

The experiments were conducted using the WEKA implementation of all of these existing algorithms, and our approaches are also implemented in the WEKA (Hall et al., 2009) environment. Table 2 shows the accuracy obtained with the *NB* and *C*4 classifiers, from column 3 to 10, and from column 11 to 18, respectively. For each group of results (*NB* or *C*4), the first three columns show the results that are obtained with the *H*1,*H*2 and *SF* algorithms using the classifier as a subset evaluator, and in the following three columns, we can see the results with the same algorithms using *CFS* instead of the classifier; the next two columns correspond to the *FCBF* algorithm and the results obtained with the complete dataset. By rows, we distinguish the three groups of data sets previously stated, with the last row of each group showing the accuracies averaged, while the average of the three groups can be seen in the last row of the table. In the first two groups, UCI and NIPS, the value of the success rate was obtained by calculating the mean of five executions of two cross-validations (5 × 2 CV), while one execution of ten cross-validations (1 × 10 CV) was conducted in the groups of BIO data sets. Then, two or ten reductions were made at each execution, one for each training set, to prevent the selection algorithm from becoming over-adjusted to the data used. Therefore, in all of the cases, each value shown in the Table is the average accuracy obtained from ten results. In the first two groups, we used two instead of ten cross-validations because of the time cost consumed with massive amounts of data.

Notice that *SF* did not report any results in several cases (*n*/*a* – not available); most of them were in the wrapper approaches but 2 and 5 were with *NB* and *C*4, respectively, because of the time cost consumption. Therefore, there are no success rates or selected attributes in these cases. For the data sets DOR and GCM, no results were provided because the program ran out of memory after a long period of time as a result of its quadratic space complexity.

For the accuracy results, we performed the following comparisons: (*a*) all of the results output by each classifier on each repository (six comparisons in all) and (*b*) all of the results output by each classifier (two comparisons in all). For dimensionality reduction, all of the result outputs were considered.

To support the conclusions obtained, statistical tests were applied. We applied the guidelines proposed by García and Herrera (2008) because we present the results of several strategies without a control method. They propose using a set or a family of hypotheses that are associated with a set of pairwise comparisons to compare the performance of a set of classifiers over multiple datasets. To adjust the value of the level of significance α, García and Herrera conclude that Bergmann–Hommel's procedure is the most suitable. They also propose an adjustment of the p-value of a pairwise comparison to account for the remaining comparisons that belong to the family.

From the data in Table 2, we evaluated whether the differences between the accuracy results were statistically significant at level $\alpha = 0.95$. On the one hand, the comparison with the *NB* classifier values yielded the following conclusions:

- With respect to the comparison with all of the results, the differences are not statistically significant between the wrapper approaches, and the same scenario occurs between the filter approaches. However, there are significant differences between the accuracies obtained with $H1^{NB}$, SF^{NB} and the accuracies obtained with the four filters and with the original set, where the wrapper versions win. Furthermore, $H2^{CF}$ and $H1^{CF}$ filters also win significantly for the complete data sets.
- If we analyze the results by each group of data sets, notice that, with the UCI data sets, we conclude exactly the statement above for all of the data sets, whereas no significant differences are shown in the NIPS and BIO groups.

On the other hand, with the *C*4 classifier, we have the following:

- With respect to the comparison with all of the data sets, the results are similar to those indicated by the classifier NB: there are not significant differences between the wrappers or between the filters. However, there are significant differences between the accuracies obtained with $H1^{C4}$, SF^{C4} and the accuracies obtained with the four filters. In addition, $H2^{CF}$ wins significantly on the complete data sets.
- If we analyze the results by each group of data sets, we emphasize that the result stated above for all of the data sets is valid for UCI and NIPS, adding that $H1^{CF}$ also won for the entire data set, whereas for the BIO group, there are no significant differences, except for the two approaches of the *H*1 with respect to the *H*2 wrapper approach.

In Table 3, we can see the reduction performed by each feature selection algorithm. In columns, in this case, we distinguish between the filter and wrapper results because the wrapper depends on the classifier that was subsequently applied, whereas the filters do not. In the first part, for the four filter approaches, the results

Table 2
Accuracy obtained with NB and C4 classifiers.

Rep.	ID	NB								C4.5							
		Wrapper			Filter				All	Wrapper			Filter				All
		H2	H1	SF	H2	H1	SF	FCBF		H2	H1	SF	H2	H1	SF	FCBF	
UCI	ADS	95.80	95.42	95.83	94.61	95.38	95.81	95.64	96.38	96.42	96.55	96.85	95.30	96.43	96.39	95.85	96.46
	ARR	68.94	68.01	67.70	67.30	66.50	68.05	63.98	60.13	67.92	68.01	67.39	66.46	66.42	67.04	64.87	64.29
	HYP	94.92	95.10	95.32	94.15	94.15	94.15	94.90	95.32	98.90	99.07	99.30	96.56	96.56	96.56	98.03	99.36
	ISO	77.41	83.30	82.28	66.95	77.61	80.79	74.62	80.42	68.15	69.43	*n/a*	67.29	72.68	71.94	66.63	73.38
	KRV	94.09	94.27	94.32	84.41	90.43	90.43	92.50	87.50	94.09	95.11	94.26	84.41	90.43	90.43	94.07	99.07
	LET	55.74	65.67	65.67	64.28	64.28	64.28	65.06	63.97	80.50	84.99	85.17	84.21	84.21	84.21	84.84	84.45
	MUL	96.80	97.21	96.87	96.55	97.04	96.72	96.19	94.37	93.74	92.42	93.11	92.77	93.17	93.12	92.29	92.74
	MUS	98.68	98.78	99.01	98.52	98.52	98.52	98.52	95.10	99.41	99.91	100.00	98.52	98.52	98.52	98.84	100.00
	MUK	84.60	84.59	84.59	74.54	79.94	69.78	72.29	83.56	95.71	95.43	*n/a*	94.44	94.06	94.60	91.19	95.12
	SIC	93.88	94.55	93.88	93.89	93.89	93.89	96.25	92.41	96.33	98.28	98.19	96.33	96.33	96.33	97.50	98.42
	SPL	94.65	94.85	94.91	93.63	93.63	93.60	95.49	95.26	92.73	93.05	93.04	92.54	92.54	92.61	93.17	92.92
	WAV	80.38	80.85	81.55	80.34	81.01	80.12	78.42	80.02	75.93	76.20	75.44	76.65	76.46	76.56	74.52	74.75
	Av.	86.32	90.36	89.19	85.08	86.24	87.70	86.26	85.44	88.32	87.03	88.07	85.04	84.78	85.87	86.31	85.94
NIPS	ARC	65.40	64.60	60.60	66.00	63.20	60.20	61.20	65.40	63.60	65.80	62.40	61.60	59.00	56.60	58.80	57.00
	DEX	79.13	81.33	75.33	80.67	82.47	87.73	85.07	86.47	78.30	80.27	90.47	80.40	81.47	80.13	79.00	73.80
	DOR	93.25	93.23	*n/a*	93.25	93.80	*n/a*	92.38	90.68	93.20	92.13	*n/a*	93.20	91.63	*n/a*	90.33	88.73
	GIS	91.17	92.66	93.55	87.26	90.83	92.64	87.58	91.88	93.00	93.29	*n/a*	89.60	90.92	93.07	90.99	92.68
	MAD	60.99	59.00	60.12	60.37	60.56	60.17	58.20	58.24	68.40	73.02	72.99	69.30	69.77	69.29	61.11	57.73
	Av.	77.99	78.16	72.40	77.51	78.17	75.19	76.89	78.53	79.29	80.90	75.29	78.82	78.56	74.77	76.05	73.99
BIO	COL	83.81	85.48	84.05	79.05	80.95	82.62	77.62	53.33	82.14	83.81	80.71	83.57	85.24	86.90	88.33	82.14
	LEU	91.43	93.04	87.32	93.04	94.46	91.43	95.89	98.57	83.21	88.57	87.32	87.32	85.89	84.82	83.21	82.14
	LYM	83.67	82.44	83.56	85.56	84.33	75.11	78.22	75.11	68.78	80.00	73.00	80.33	85.56	79.22	78.22	81.44
	GCM	62.11	67.33	*n/a*	62.11	70.53	*n/a*	68.95	65.79	50.00	46.84	*n/a*	53.68	52.63	*n/a*	52.63	60.00
	Av.	80.26	82.07	84.98	79.94	82.57	83.05	80.17	73.20	71.03	74.81	80.34	76.23	77.33	83.65	75.60	76.43
Av. total		81.52	83.53	82.19	80.84	82.33	81.98	81.10	79.06	79.55	80.91	81.23	80.03	80.22	81.43	79.32	78.79

Table 3
Reduction performed by each feature selection algorithm. UCI: percentage of features retained; NIPS–BIO: number of features.

Rep.	ID	Filter				Wrapper					
						NB			C4.5		
		H2	H1	SF	FCBF	H2	H1	SF	H2	H1	SF
UCI	ADS	0.3	0.4	0.6	5.3	0.5	0.7	1.1	0.5	0.5	0.8
	ARR	4.1	4.1	6.2	2.9	2.3	5.5	3.0	2.0	2.4	3.1
	HYP	3.4	3.4	3.4	18.3	10.3	15.9	29.3	10.7	14.5	20.3
	ISO	3.8	11.1	15.4	3.7	3.4	11.1	4.7	2.7	3.6	*n/a*
	KRV	7.2	8.3	8.3	18.1	11.4	13.9	14.4	11.1	17.2	13.6
	LET	56.3	56.3	56.3	64.4	39.4	68.8	72.5	45.0	68.8	63.1
	MUL	3.9	4.3	13.9	18.7	2.1	3.4	2.4	1.5	3.2	2.1
	MUS	4.5	4.5	4.5	16.4	7.3	9.5	13.6	9.1	18.6	22.3
	MUK	7.5	3.9	9.8	1.7	1.0	0.6	0.6	4.9	5.8	*n/a*
	SIC	3.4	3.4	3.4	16.6	3.4	8.3	3.4	7.2	20.3	19.0
	SPL	10.0	10.0	10.2	36.3	15.3	21.8	24.7	12.2	16.3	18.3
	WAV	35.3	31.0	37.0	15.3	21.3	30.5	32.3	17.5	24.0	19.8
	Av.	11.6	11.7	14.1	18.1	9.8	15.8	16.8	10.4	16.3	18.2
NIPS	ARC	22.5	39.2	42.6	35.2	4.6	15.3	3.8	2.7	7.9	3.7
	DEX	7.5	11.3	35.5	25.1	15.3	30.2	13.2	5.8	18.9	8.7
	DOR	2.1	11.9	*n/a*	75.3	2.3	10.5	*n/a*	2.6	7.2	*n/a*
	GIS	8.6	30.2	62.2	31.2	9.2	35.3	24.2	11.5	26.9	*n/a*
	MAD	6.3	5.8	9.9	4.7	4.9	11.8	5.8	4.3	17.0	12.4
BIO	COL	7.3	15.8	22.1	14.6	4.0	3.5	5.9	2.5	2.9	3.3
	LEU	5.6	6.7	40.3	45.8	3.0	2.5	3.2	1.9	1.2	1.6
	LYM	30.5	57.6	153.2	290.9	7.7	10.3	7.1	5.2	8.8	8.2
	GCM	32.1	56.1	*n/a*	60.9	19.3	44.0	*n/a*	9.1	9.8	*n/a*

were obtained with the *H*1, *H*2, and *SF* algorithms with *CFS* as an evaluator, and the *FCBF* algorithm results are shown from columns 3 to 6. In the second part, for the wrapper approaches, the first three columns (7–9) show the results obtained with the *H*1, *H*2 and *SF* algorithms using *NB* as a subset evaluator, and in the following three columns, we can see the results with *C*4. By rows, there are the same three groups, where the percentage of features retained can be seen in the first group, and in the following two groups (NIPS and BIO), the number of features is shown because the percentage of attributes retained is too low to be comparable.

As shown, for each approach (filter and wrapper), *H*2 is the strategy that selects, on average, the smallest subsets of features. These differences are statistically significant at level $\alpha = 0.95$ for *H*2 when using *C*4.5.

Differences between *H*2 (filter and wrapper using *NB*) and *FCBF* are significant at $\alpha = 0.99$ in favor of the first strategy. *H*2 reduction

Table 4
Running time in seconds for each feature selection algorithm.

Data	Filter				Wrapper					
					NB			C4.5		
	H2	H1	SF	FCBF	H2	H1	SF	H2	H1	SF
UCI	39.8	48.6	132.7	67.6	9610.1	6112.3	49620.2	6665.6	5384.3	40097.6
NIPS	423.1	799.9	8645.1	131.1	2493.0	11019.5	88279.7	1038.2	6114.0	16067.5
BIO	41.5	59.4	2083.6	11.1	574.0	442.5	1710.9	961.9	801.8	4887.4
Total	504.4	907.9	10861.4	209.8	12677.1	17574.3	139610.8	8665.7	12300. 1	61052.5

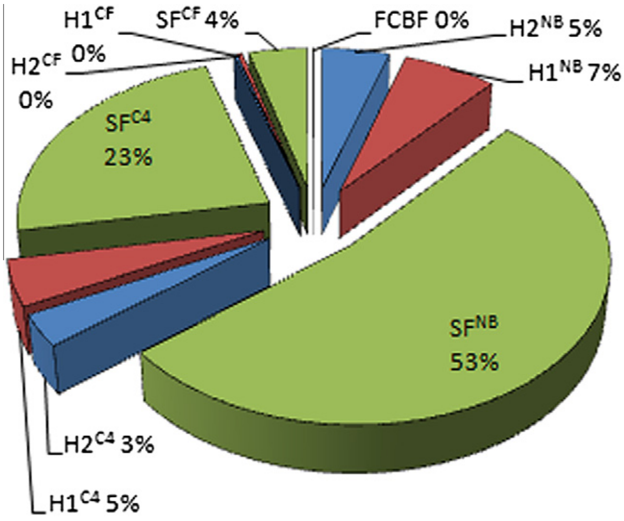

Fig. 1. Time percentage with respect to the total time for each algorithm.

is significant when compared with *SF* (filter and wrapper with NB) at $\alpha = 0.95$. *FCBF* is the strategy that finds the largest subsets. These results are significant at 0.95 when compared with $H1^{CF}$ and SF^{C4}.

Table 4 reports the running time for each feature selection algorithm on the UCI, NIPS and BIO data sets, showing three different results, two for the wrapper approaches (depending on the learning algorithm chosen) and one for the filter approach. These results do not include data for any algorithm when *SF* did not report any results, i.e., ISO, MUK, DOR, GIS and GCM with *C4*, DOR and GCM with *NB* and filter approaches with DOR and GCM. From the last row in Table 4, a global comparison can be viewed in Fig. 1, where the algorithm, the time needed for the algorithm and the percentage with respect to the total time are shown.

In the wrapper cases, on the one hand, we can observe in Table 4 that *H*1 requires slightly more time than *H*2 on average (columns 2–3, 6–7 and 9–10). The time savings of *H*2 with respect to *H*1 becomes more obvious when the volume of the data increases (the number of features and/or instances), as in the NIPS data. The time needed for both of the hybrid algorithms is similar with the BIO data sets; however, *H*1 obtains better times than *H*2 for the wrapper versions with the UCI data. Therefore, although the amount of data becomes massive, *H*2 becomes much more efficient. On the other hand, the advantage of *H*1 and *H*2 with respect to the *SF* (sequential forward search) is clear. Wrapper approaches of hybrid versions are between 5 and 10 times faster than *SF* because the wrapper subset evaluation is run fewer times. For example, for the lymphoma data set and the *C*4 classifier, *H*1 and *SF* retain 8.80 and 8.20 genes, respectively, on average. To obtain these subsets, the first run evaluated 4026 genes individually (to generate the ranking) and 4026 subsets, whereas the second run evaluated 32180 subsets (4026 genes + 4025 pairs of genes + $\cdots$ + 4019 sets of eight genes). The time savings of hybrid versions became more obvious when the computer load necessary for the mining algorithm increased. Finally, the filter approaches are compared (columns 2–5), and we can see that the values are not comparable with those obtained with wrappers. On average, the *FCBF* method is the fastest, closely followed by the *H*2 and *H*1 algorithms, and

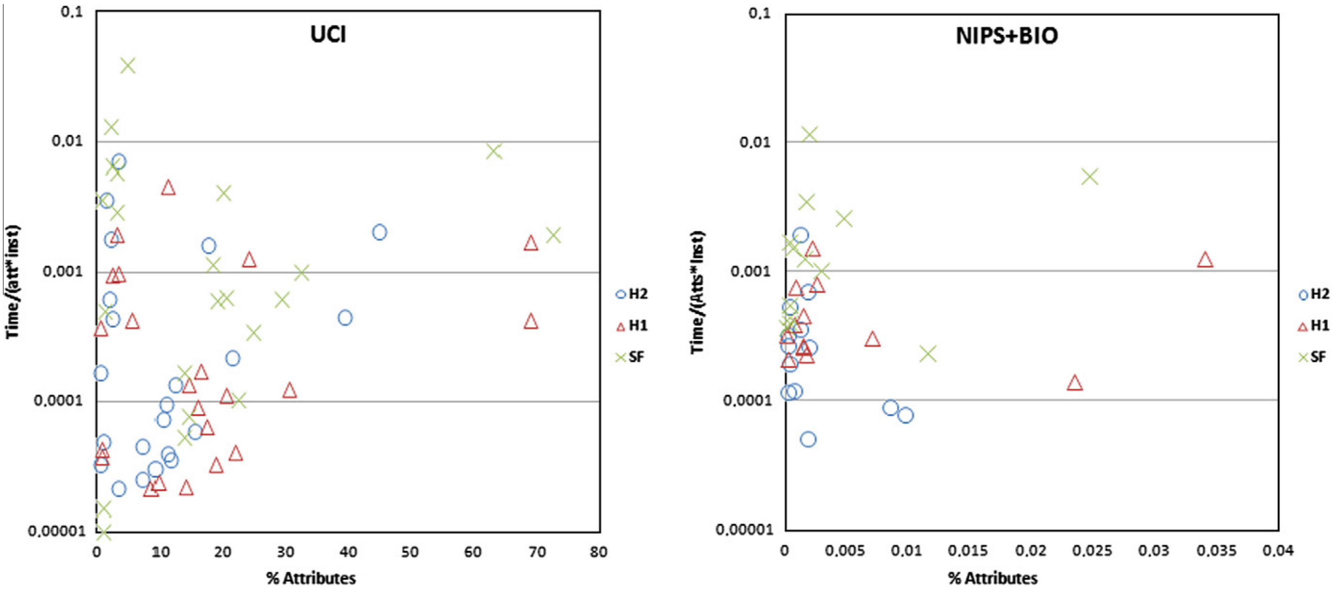

Fig. 2. Time versus reduction with wrapper algorithms.

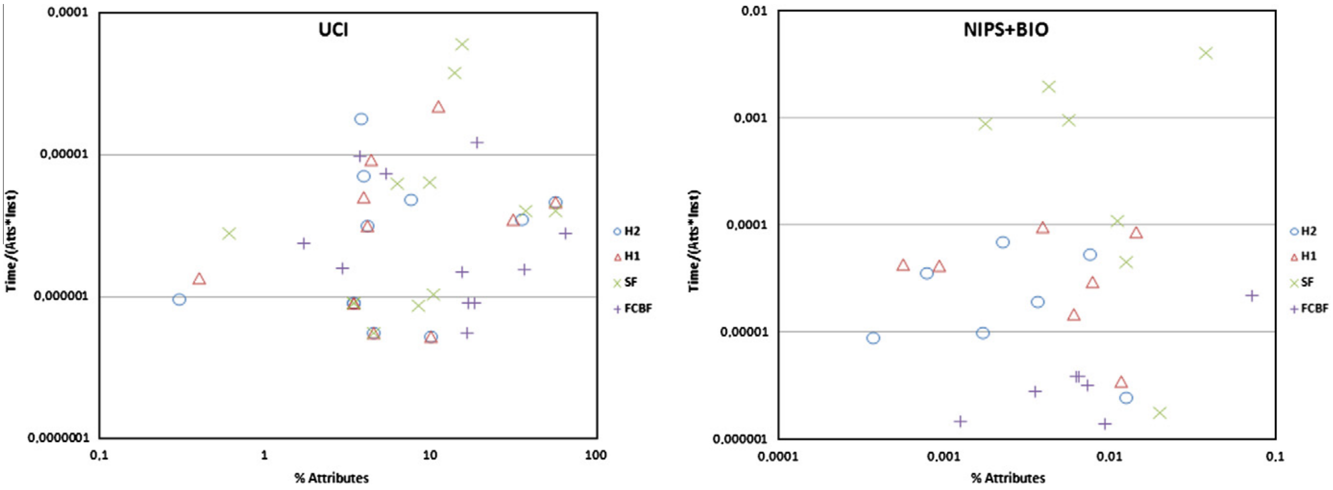

Fig. 3. Time versus reduction with filter algorithms.

in the last position, *SF* requires much more time. Comparing the results in Fig. 4, it can be observed that there are obvious differences.

Figs. 2 and 3 show the plot of time versus reduction with both the wrapper and filter approaches, respectively. In both cases, we can see the results from the UCI data sets on the left and from the NIPS + BIO on the right. The percentage of features retained is represented on the abscissa. To clarify, we used a logarithmic scale with the NIPS + BIO data instead of a decimal scale because of the low values with these very high–dimensional data sets. On the ordinate, we use $\frac{Time}{\#Atts \times \#Inst.}$ to capture the original size of each data set, and a logarithmic scale is used in all cases.

Smaller values in the subset size are on the left side of the plot and shorter times are in the lower position; thus, the bottom left positions refer to results with better time costs and a large reduction. As shown in Fig. 2 with the wrapper approaches, in both cases *H*2 is the strategy that has more points in this position; whereas in Fig. 3 with the filters, we can see the previous assertion that a lower percentage of attributes is retained by *H*2, and less time is needed by *FCBF* with the NIPS + BIO data sets.

4. Conclusions

In this work, we propose two hybrid models that are suitable for working with large datasets. We compare the performance of both algorithms with *FS* and *FCBF*. The experiments were conducted with *NB* and *C*45 as induction algorithms in the filter and wrapper approaches. We can state that the feature selection methods based on ranking achieve promising results. Moreover, the *SF* strategy is not suitable for large datasets because of its slow convergence. The *FCBF* is a fast algorithm; however, it selects large feature subsets.

In short, on the one hand, accuracy results achieved by the wrapper approaches of *H*1 are better than results obtained with filter approaches, whereas there are not significant differences between the accuracies obtained with respect to *H*2. On the other hand, from a reduction point of view, the results obtained with *H*2 are the best among the wrapper and filter comparisons.

References

Alizadeh, A., Eisen, M., Davis, R., Ma, C., Lossos, I., Rosenwald, A., et al. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature, 403*, 503–511.

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., et al. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences USA, 96*, 6745–6750.

Arauzo-Azofra, A., Aznarte, J. L., & Benítez, J. M. (2010). Empirical study of feature selection methods based on individual feature evaluation for classification problems. *Expert Systems with Applications, 38*(7), 8170–8177.

Armañanzas, R., Saeys, Y., Inza, I., García-Torres, M., Bielza, C., van de Peer, Y., et al. (2011). Peakbin selection in mass spectrometry data using a consensus approach with estimation of distribution algorithms. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 8*(3), 760–774.

Azam, N., & Yao, J. (2011). Comparison of term frequency and document frequency based feature selection metrics in text categorization. *Expert Systems with Applications, 39*(5), 4760–4768.

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

Bermejo, P., Gámez, J., & Puerta, J., (2008). On incremental wrapper-based attribute selection: Experimental analysis of the relevance criteria. In IPMU'08: proceedings of the 12th international conference on information processing and management of uncertainty in knowledge-based systems.

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

Caruana, R., & Freitag, D. (1994). How useful is relevance? In *Working notes of the AAAI fall symposium on relevance* (pp. 25–29).

Chen, J., Huang, H., Tian, S., & Qu, Y. (2008). Feature selection for text classification with Naïve Bayes. *Expert Systems with Applications, 36*(3), 5432–5435.

Das, S. (2001). Filters, wrappers and a boosting-based hybrid for feature selection. In *18th international conference on machine learning* (pp. 74–81). Morgan Kaufman Publishers Inc..

Dash, M., Liu, H., & Motoda, H. (2000). Consistency based feature selection. In *Pacific–Asia conference on knowledge discovery and data mining* (pp. 98–109).

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis, 1*(3), 131–156.

Dash, M., & Liu, H. (2003). Consistency-based search in feature selection. *Artificial Intelligence, 151*(1–2), 155–176. http://dx.doi.org/10.1016/S0004-370(03)00079-1.

Ding, C., & Peng, H. (2003). Minimum redundancy feature selection from microarray gene expression data. *IEEE Computer Society Bioinformatics*, 523–529.

Foithong, S., Pinngern, O., & Attachoo, B. (2011). Feature subset selection wrapper based on mutual information and rough sets. *Expert Systems with Applications, 39*(1), 574–584.

Frank, A., & Asuncion, A. (2010). UCI machine learning repository. URL <http://archive.ics.uci.edu/ml>.

García, S., & Herrera, F. (2008). An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *Journal of Machine Learning Research, 9*, 2677–2694.

García-López, F. C., García-Torres, M., Melián-Batista, B., Moreno-Pérez, J. A., & Moreno-Vega, J. M. (2006). Solving the feature selection problem by a parallel scatter search. *European Journal of Operations Research, 169*(2), 477–489.

García-Torres, V., García-López, F. C., Melián-Batista, B., Moreno-Pérez, J. A., & Moreno-Vega, J. M. (2004). Solving feature subset selection problem by a hybrid metaheuristic, In *First international workshop in hybrid metaheuristics at ECAI 2004 (HM 2004)* (pp. 59–69).

García-Torres, M., Armañanzas, R., Bielza, C., & Larrañaga, P., (in press). Comparison of metaheuristic strategies for peakbin selection in proteomic mass spectrometry data, <http://dx.doi.org/10.1016/j.ins.2010.12.013>.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., et al. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science, 286*, 531–537.

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

Guyon, I., Gunn, S., Ben-Hur, A., & Dror, G. (2005). Result analysis of the nips 2003 feature selection challenge. In *Advances in neural information processing systems* (pp. 545–552). MIT Press.

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

Hall, M. (2000). Correlation-based feature selection for discrete and numeric class machine learning. In *17th international conference on machine learning* (pp. 359–366). San Francisco, CA: Morgan Kaufmann.

Hall, M., & Holmes, G. (2003). Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering, 15*(3).

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explorations Newsletters, 11*, 10–18. http://doi.acm.org/10.1145/1656274.1656278.

Kabir, M. M., Shahjahan, M., & Murase, K. (2011). A new hybrid ant colony optimization algorithm for feature selection. *Expert Systems with Applications, 39*(3), 3747–3763.

Kohavi, R., & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence* (1–2), 273–324.

Koller, D., & Sahami, M. (1996). Toward optimal feature selection. In *13th International Conference on Machine Learning* (pp. 284–292).

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering, 17*(3), 1–12.

Pacheco, J., Casado, S., & Núnez, L. (2007). Use of VNS and TS in classification: Variable selection and determination of the linear discrimination function coefficients. *IMA Journal of Management Mathematics, 18*(2), 191–206.

Pérez-Benítez, J. A., & Padovese, L. R. (2011). Feature selection and neural network for analysis of microstructural changes in magnetic materials. *Expert Systems with Applications, 38*(8), 10547–10553.

Press, W., Flannery, B., Teukolsky, S., & Vetterling, W. (1988). *Numerical Recipes in C*. Cambridge: Press Syndicate of the University of Cambridge.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C., Angelo, M., et al. (2000). Multiclass cancer diagnosis using tumor gene expression signatures. *Journal of computational Biology, 7*(3–4), 559–584.

Saeys, Y., Abeel, T., & de Peer, Y. V. (2008). Robust feature selection using ensemble feature selection techniques. In *ECML/PKDD* (2) (pp. 313–325).

Xing, E., Jordan, M., & Karp, R. (2001). Feature selection for high-dimensional genomic microarray data. In *Proceedings 18th international conference on machine learning* (pp. 601–608). San Francisco, CA: Morgan Kaufman.

Yang, Y., Liao, Y., Meng, G., & Lee, J. (2011). A hybrid feature selection scheme for unsupervised learning and its application in bearing fault diagnosis. *Expert Systems with Applications, 38*(9), 11311–11320.

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