Applied Soft Computing 30 (2015) 136–150

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Distributed feature selection: An application to microarray data classification

V. Bolón-Canedo*, N. Sánchez-Maroño, A. Alonso-Betanzos

Laboratory for Research and Development in Artificial Intelligence (LIDIA), Computer Science Department, University of A Coruña, 15071 A Coruña, Spain

ARTICLE INFO

Article history:
Received 14 March 2013
Received in revised form 15 January 2015
Accepted 16 January 2015
Available online 7 February 2015

Keywords:
Feature selection
Distributed learning
Microarray data

ABSTRACT

Feature selection is often required as a preliminary step for many pattern recognition problems. However, most of the existing algorithms only work in a centralized fashion, i.e. using the whole dataset at once. In this research a new method for distributing the feature selection process is proposed. It distributes the data by features, i.e. according to a vertical distribution, and then performs a merging procedure which updates the feature subset according to improvements in the classification accuracy. The effectiveness of our proposal is tested on microarray data, which has brought a difficult challenge for researchers due to the high number of gene expression contained and the small samples size. The results on eight microarray datasets show that the execution time is considerably shortened whereas the performance is maintained or even improved compared to the standard algorithms applied to the non-partitioned datasets.

1. Introduction

Over the last decade, feature selection, which consists of detecting the relevant features and discarding the irrelevant ones [1,2], has been an active research area due to the high dimensionality of the datasets. Feature selection methods usually come divided into three types: filters, wrappers and embedded methods. While wrapper models involve optimizing a predictor as part of the selection process, filter models rely on the general characteristics of the training data to select features independently of any predictor. The embedded methods generally use machine learning models for classification, and then an optimal subset of features is built by the classification algorithm. In the last few years, ensemble methods represented a new type of methods for feature selection. They aim to cope with the instability issues observed in many techniques for feature selection when the characteristics of the data change [3–5]. In previous works, several feature selection methods are applied and the obtained features are merged into a more stable subset of features prior to classification or the different predictions obtained with the different subsets of features are somewhat combined [6,7]. However, as stated in [8], even when the subset of features is not optimal, filters are preferable due to their computational and statistical scalability, so they will be the focus of this research.

The filter approach is commonly divided into two different subclasses: individual evaluation and subset evaluation [9]. Individual evaluation is also known as feature ranking and assesses individual features by assigning them weights according to their degrees of relevance. On the other hand, subset evaluation produces candidate feature subsets based on a certain search strategy. Each candidate subset is evaluated by a certain evaluation measure and compared with the previous best one with respect to this measure. While the individual evaluation is incapable of removing redundant features because these are likely to have similar rankings, the subset evaluation approach can handle feature redundancy with feature relevance. However, methods in this framework can suffer from an inevitable problem which is caused by searching though the possible feature subsets. This stage, required in the subset generation step, usually increases the computational time. Having said that, a revision of the existing literature showed that the subset evaluation approach outperformed the ranking methods [9–12].

Feature selection is usually applied in a centralized manner, i.e. a single learning model is used to solve a given problem. However, if the data is distributed, feature selection may take advantage of processing multiple subsets in sequence or concurrently. The need to use distributed feature selection can be two-fold. On the one hand, with the advent of network technologies data is sometimes distributed in multiple locations and may consequently be biased. On the other hand, most of the existing feature selection algorithms do not scale well and their efficiency significantly deteriorates or even becomes inapplicable when dealing with large-scale data. In order to increase efficiency, learning can be parallelized by

* Corresponding authors. Tel.: +34 981167000; fax: +34 981167160.
E-mail addresses: vbolon@udc.es (V. Bolón-Canedo), ciamparo@udc.es (A. Alonso-Betanzos).

http://dx.doi.org/10.1016/j.asoc.2015.01.035
1568-4946/

distributing the subsets of data to multiple processors, learning in parallel and then combining them. There are two main techniques for partitioning and distributing data: vertically, i.e. by features, and horizontally, i.e. by samples. Distributed learning has been used to scale up datasets that are too large for batch learning in terms of samples [13–15]. While not common, there are some other developments that distribute the data by features [16,17]. In this study, the data are distributed vertically in order to have the feature selection process distributed. After having the data distributed in small feature subsets and selecting the relevant features from each subset, a merging procedure is performed which updates the feature subset according to improvements in the classification accuracy.

Although this approach can be applied to any feature-abundant classification problem, it is especially suitable for application to microarray data. This type of data has become very popular in the past decade since it poses a difficult challenge for machine learning researchers due to its high number of features and its small sample size. In this domain, features represent gene expression coefficients corresponding to the abundance of mRNA – messenger ribonucleic acid – in a sample (e.g. tissue biopsy), for a number of patients. Although there are usually very few samples, the number of features in the raw data ranges from 6000 to 60,000. A typical classification task is to separate healthy patients from cancer patients based on their gene expression "profile". By applying the proposed distributed methodology to this domain, we will be able to deal with subsets with a more balanced feature/sample ratio and avoid overfitting problems. The experimental results from eight different databases demonstrate that our proposal can improve the performance of original feature selection methods and show important savings in running times.

The remainder of the paper is organized as follows: Section 2 describes the state of the art on distributed feature selection and feature selection methods on microarray data, Section 3 introduces our distributed approach, Section 4 reveals the experimental setup and Section 5 visualizes the experimental results. Finally, Sections 6 and 7 provide the discussion and conclusions, respectively.

2. State of the art

The literature about feature selection for microarray data is abundant. Different feature selection strategies have been proposed over the last years for feature/gene selection. Ferreira and Figueiredo [18] proposed combining unsupervised feature discretization and feature selection techniques which have improved previous related techniques over several microarray datasets. In [19] a new framework for feature selection based on dependence maximization between the selected features and the labels of an estimation problem is presented and tested over microarray data, showing promising results. Wang et al. [20] also proposed a novel filter framework to select optimal feature subsets based on a maximum weight and minimum redundancy criterion. Hybrid methods have been recently tested on this type of data [21,22] obtaining high classification accuracies. Embedded methods have also been proposed, such as in [23], where the authors introduced an algorithm that simultaneously selects relevant features during classifier construction by penalizing each feature's use in the dual formulation of support vector machines. On the other hand, trying to overcome the problem that a weakly ranked gene could be relevant within an appropriate subset of genes, Sharma et al. [24], introduced an algorithm that first distributes genes into relative small subsets, then selects informative smaller subsets of genes from a subset and merges the chosen genes with another gene subset to update the final gene subset. Their method showed promising classification accuracy for all the test datasets.

As we have said, many filter approaches were applied to successfully classify microarray data [8,12,25], however according to the authors' knowledge, there has been no attempt in the literature to tackle this problem with distributed feature selection, apart from the aforementioned proposal of Sharma et al. [24]. In fact, feature selection in distributed environments is a poorly explored field and very few references were found in the literature, most of which undertaken over the few last years. It is worth mentioning the research of Das et al. [26], where an algorithm is presented which, based on a horizontal partition (by samples), performs feature selection in an asynchronous fashion with a low communication overhead by which each peer can specify its own privacy constraints. A vertical partition of the data (by features) to generate the diverse components of an ensemble [27] is also present in the literature. However in these cases, feature selection is not applied to the different partitions of the data and therefore the model may be constructed based on irrelevant features. More recently, Banerjee and Chakravarty [28] proposed a distributed feature selection method evolved from a method called virtual dimension reduction, where the partition of the data can be done both vertically or horizontally. Zhao et al. [29] presented a distributed parallel feature selection algorithm based on maximum variance preservation. The algorithm can read data in a distributed form and perform parallel feature selection in both symmetric multiprocessing mode via multithreading and massively parallel processing.

Still, DNA microarray data prevents the use of horizontal partitioning because of the small sample size. The distributed methods mentioned above based on vertical partitioning have not been designed specifically for dealing with microarray data, so they do not tackle the particularities of this type of data, such as the high redundancy present among the features. The method proposed in [24] does address these issues, but it has the disadvantage of being computationally expensive by interacting with a classifier to select the genes in each subset. In this work we will propose a distributed filter method, suitable for application to microarray data and with a low computational cost.

3. The proposed method

Distributed feature selection has not been deeply explored yet. So, in this paper we present a distributed filter approach trying to improve upon previous accuracy results over microarray data as well as reducing the running time. Our proposal consists of performing several fast filters over several partitions of the data, combined afterwards into a single subset of features. Thus, we divide each dataset D into several small disjoint subsets D_i. The filter is applied to each of them, generating a corresponding selection S_i. After all the small datasets D_i have been used (which could be done in parallel, as all of them are independent of each other), the combination method builds the final selection S as the result of the filtering process. To sum up, there are three main steps in this methodology:

1. Partition of the datasets.
2. Application of filtering to the subsets.
3. Combination of the results.

The partition of the dataset consists of dividing the original dataset into several disjoint subsets of approximately the same size that cover the full dataset (see Fig. 1). As mentioned in Section 1, in this research the partition is made vertically. Two different methods are used for partitioning the data: (a) performing a random partition and (b) ranking the original features before generating the subsets. The second option was introduced to try to improve the performance obtained by the first one. By having an ordered ranking, features with similar relevance to the class will be in the same subset, which will facilitate the task of the subset filter which

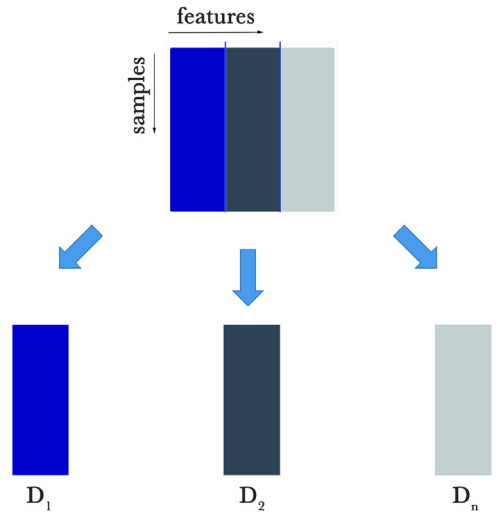

Fig. 1. Vertical partition of the data.

will be applied later. These two techniques for partitioning the data will generate two different approaches for the distributed method: distributed filter (DF) with the random partition and distributed ranking filter (DRF) associated to the ranking partition.

After this step, the data is split by assigning groups of k features to each subset, where the number of features k in each subset is half the number of samples, to avoid overfitting. When opting for the random partition (DF), the groups of k features are constructed randomly, taking into account that the subsets have to be disjointed. In the case of the ranking partition (DRF), the groups of k features are generated sequentially over the ranking, so features with a similar ranking will be in the same group. Notice that the random partition is equivalent to obtaining a random ranking of the features (by shuffling the features) followed by the same steps as with the ordered ranking. Fig. 2 shows a flow chart which reflects the two algorithms proposed, DF and DRF.

Algorithm 1. Pseudo-code for the distributed filter algorithm

$D_{(m\times s)}$: =training dataset with m samples and s features
n : =number of subsets of k features

1. Select the partition method:
   - DF → Shuffle features from dataset D to obtain a random ranking R or
   - DRF → Apply a ranker method over D to obtain an ordered ranking R
2. for $i = 1$ to n do
   (a) R_i = first k features in R
   (b) $R = R \setminus R_i$
   (c) $D_i = D_{(m\times|R_i|)}$
3. for $i = 1$ to n do
   (a) S_i = subset of features obtained after applying the chosen filter over D_i
4. $S = S_1$
5. *baseline* = accuracy obtained by classifying subset $D_{(m\times|S_1|)}$ with classifier C
6. for $i = 2$ to n do
   (a) $S_{aux} = S \cup S_i$
   (b) *accuracy* = accuracy obtained by classifying subset $D_{(m\times|S_{aux}|)}$ with classifier C
   (c) if *accuracy* > *baseline*
   i. $S = S_{aux}$
   ii. *baseline* = *accuracy*
7. Build classifier C with $D_{(m\times|S|)}$
8. Obtain prediction P

After having several small disjoint datasets D_i, the filter method will be applied to each of them, returning a selection S_i for each subset of data. Finally, to combine the results, a merging procedure is necessary. In this work we have opted for a merging procedure involving a classifier, since simpler methods, such as the union,

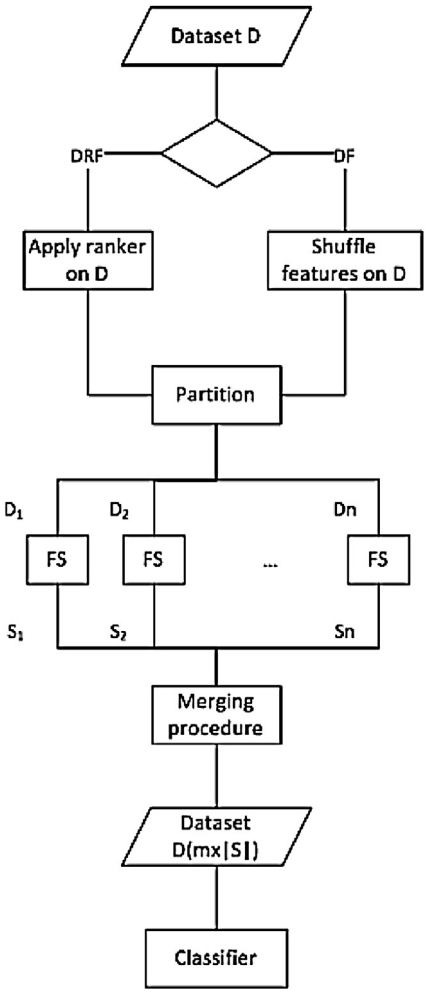

Fig. 2. Flow chart of proposed algorithm.

result in a huge number of features (probably containing redundancy, and perhaps overfitting), as will be seen in Section 5.4. In the method we propose, the first selection S_1 is taken to calculate the classification accuracy, which will be the *baseline*, and the features in S_1 will always become part of the final selection S. For the remaining selections, the features in S_i, $i = 2 \ldots n$ will become part of the final selection S if they improve the baseline accuracy, as can be seen in more detail in Algorithm 1. Combining the features in this manner is expected to remove redundancy, since a redundant feature will not improve the accuracy and hence will not be added to the final selection. At the end, this final selection S is applied to the training and test sets in order to obtain the ultimate classification accuracies. It must be noted that this algorithm can be used with any feature subset filter.

4. Experimental setup

This section will present the microarray datasets chosen for testing the distributed approaches, as well as the ranker method and filters which will carry out the feature selection process. For testing the adequacy of DF and DRF, four well-known supervised classifiers, of different conceptual origin, were selected to evaluate the proposal. All the classifiers and filters are executed using the Weka tool [30], with default values for their parameters. Experimentation is performed on an Intel(R) Xeon(R) CPU W3550 @ 3.07 QUAD-CORE with 12 GB RAM.

Table 1
Dataset description for binary datasets.***

Dataset	Attributes	Samples		Train distribution (%)	Test distribution (%)
		Train	Test		
Colon	2000	42	20	67–33	60–40
DLBCL	4026	32	15	50–50	53–47
CNS	7129	40	20	65–35	65–35
Leukemia	7129	38	34	71–29	59–41
Prostate	12,600	102	34	49–51	26–74
Lung	12,533	32	149	50–50	90–10
Ovarian	15,154	169	84	35–65	38–62
Breast	24,481	78	19	56–44	37–63

4.1. Datasets

The performance of the distributed filter will be tested over DNA microarray data. These types of datasets pose a huge challenge for feature selection researchers due to the high number of gene expression data contained and the small samples size. Eight well-known binary microarray datasets[1] are considered, listed in Table 1. Those datasets originally divided into training and test sets were maintained, whereas, for the sake of comparison, datasets which come originally with only a training set were randomly divided using the common rule 2/3 for training and 1/3 for testing. This division introduces an interesting scenario, since in some datasets, the distribution of the classes in the training set differs from the one in the test set. Table 1 depicts the number of attributes and samples and also the distribution of the binary classes, i.e. the percentage of binary labels in the datasets, showing if the data are unbalanced.

As can be seen in Table 1, Leukemia dataset presents the so-called imbalance problem. A dataset is considered unbalanced when the classification categories are not approximately equally represented in the training set. This situation is very common in microarray datasets, when most instances correspond to "normal" patterns while the main goal consists of identifying the "abnormal" patterns.

Some techniques are available to overcome the imbalance problem, however some of them are not appropriate for microarray data [31]. For this sake, and following the recommendations in [31], we will use random oversampling to deal with the Leukemia dataset, which consists of replicating, at random, elements of the undersized class until it matches the size of the other classes. Notice that, when dealing with a real-life dataset (especially if it involves people's health), it is necessary to take into account the specialists' opinion before taking any action to modify the data.

4.2. Selection of the ranker method

There are two different types of ranker feature selection: univariate and multivariate. Univariate methods are fast and scalable, but ignore feature dependencies. On the other hand, multivariate filters model feature dependencies, but at the cost of being slower and less scalable than univariate techniques [32].

As representatives of univariate ranker filters we have chosen the well known Information Gain [33] and ReliefF [34] methods. As for multivariate ranker filter, minimum-Redundancy-maximum-Relevance (mRMR) was chosen, which has proven to be very suitable for application to microarray data [12].

Because of their nature, the rankings returned by Information Gain and ReliefF place the more relevant features on the top of the ranking (even when they are redundant) whilst mRMR is able to detect the redundant features and move them to lower positions in the ranking. Feature/gene redundancy is an important problem in microarray data classification. In the presence of thousands of features, researchers noticed that it is common that a large number of features are not informative because they are redundant. Empirical evidence showed that along with irrelevant features, redundant features also affect the speed and accuracy of mining algorithms [35].

In light of the above, the behavior of mRMR might seem to be an advantage in removing redundancy, but it is not the case in this research. Preliminary experiments with these three ranker filters (not included for the sake of brevity) revealed that it was better to have the redundant features in the same packet of features, so the filter applied afterwards could remove the redundant features at once. On the contrary, if the redundant features are split in different packets, there are more chances to keep them and degrade the performance. Moreover, multivariate ranker filters are slower than univariate ones, and to deal with the entire set of features (in the order of thousands) it is important to reduce this processing time. Between Information Gain and ReliefF, the former obtained more promising results in those preliminary experiments, so it was chosen for this research. This univariate filter provides an ordered ranking of all the features where the worth of an attribute is evaluated by measuring the Information Gain with respect to the class. Theoretically, the most relevant features will be placed in the first partition and so on. In this manner, it will be easier to remove redundancy, since features with similar relevance to the class would be together in the same partition.

One of the particularities of the Information Gain filter is that, if a feature is not relevant to the prediction task, its Information Gain with respect to the class is zero. For this reason, we will also try an approach which consists of eliminating the features with Information Gain zero from the initial ranking.

4.3. Feature selection filters

The distributed methods proposed herein can be used with any feature selection filter. In this work, three well-known subset filters and two popular ranking filters were chosen for testing our proposal. The ranking methods return an ordered ranking of the features, so it is necessary to establish a threshold in order to obtain a subset of features. In this case we have opted for retaining 10% and 25% of the top features. Notice that although most of the filters work only over nominal features, the discretization step is done by default by Weka, working as a black box for the user.

- **Correlation-based Feature Selection** (CFS) is a simple multivariate filter algorithm that ranks feature subsets according to a correlation based heuristic evaluation function [36]. The bias of the evaluation function is toward subsets that contain features that are highly correlated with the class and uncorrelated with each other, so irrelevant and redundant features should be screened out.

[1] Datasets available on http://datam.i2r.a-star.edu.sg/datasets/krbd/.

- The **Consistency-based Filter** [37] evaluates the worth of a subset of features by the level of consistency in the class values when the training instances are projected onto the subset of attributes. The algorithm generates a random subset S from the number of features in every round. If the number of features of S is less than the current best, the data with the features prescribed in S is checked against the inconsistency criterion. If its inconsistency rate is below a pre-specified one, S becomes the new current best. The inconsistency criterion, which is the key to the success of this algorithm, specifies to what extent the dimensionally reduced data can be accepted. If the inconsistency rate of the data described by the selected features is smaller than a pre-specified rate, it means the dimensionally reduced data is acceptable.
- The **INTERACT** algorithm [38] is based on symmetrical uncertainty (SU) [39], which is defined as the ratio between the Information Gain (IG) and the entropy (H) of two features, x and y: $SU(x, y) = 2IG(x|y)/[H(x) + H(y)]$, where the Information Gain is defined as $IG(x|y) = H(y) + H(x) - H(x, y)$, being $H(x)$ and $H(x, y)$ the entropy and joint entropy, respectively. Besides SU, INTERACT also includes the consistency contribution (c-contribution). C-contribution of a feature is an indicator about how significantly the elimination of that feature will affect consistency. The algorithm consists of two major parts. In the first part, the features are ranked in descending order based on their SU values. In the second part, features are evaluated one by one starting from the end of the ranked feature list. If c-contribution of a feature is less than an established threshold, the feature is removed, otherwise it is selected. The authors stated that this method can handle feature interaction, and efficiently selects relevant features.
- The **Information Gain** filter [33] is one of the most common univariate methods of evaluating attributes. This filter evaluates the features according to their Information Gain and considers a single feature at a time. It provides an orderly classification of all the features, and then a threshold is required to select a certain number of them according to the order obtained.
- The filter **ReliefF** [34] is an extension of the original Relief algorithm [40]. The original Relief works by randomly sampling an instance from the data and then locating its nearest neighbor from the same and opposite classes. The values of the attributes of the nearest neighbors are compared to the sampled instance and used to update relevance scores for each attribute. The rationale is that a useful attribute should differentiate between instances from different classes and have the same value for instances from the same class. ReliefF adds the ability of dealing with multiclass problems and is also more robust and capable of dealing with incomplete and noisy data. This method may be applied in all situations, has low bias, includes interaction among features and may capture local dependencies which other methods miss.

4.4. Classifiers

Since we are dealing with real datasets, the relevant features are not known a priori. Therefore, it is necessary to use a classification algorithm to evaluate the performance of the feature selection, focusing on the classification accuracy. Unfortunately, the class prediction depends also on the classification algorithm used, so when testing a feature selection method, a common practice is to use several classifiers to obtain results as classifier-independent as possible. In this research the following classification algorithms has been used:

- **C4.5** is a classifier developed by Quinlan [41], as an extension of the ID3 algorithm (Iterative Dicotomiser 3). Both algorithms are based on decision trees. A decision tree classifies a pattern by filtering it in a descending way until it finds a leaf, that points to the corresponding classification. One of the improvements of C4.5 with respect to ID3 is that C4.5 can deal with both numerical and symbolic data. In order to handle continuous attributes, C4.5 creates a threshold and depending on the value that the attribute takes, the set of instances is divided.
- A **naive Bayes** classifier [42] is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. This classifier assumes that the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, given the class variable. A naive Bayes classifier considers each of the features to contribute independently of the probability that a sample belongs to a given class, regardless of the presence or absence of the other features. Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. In fact, naive Bayes classifiers are simple, efficient and robust to noise and irrelevant attributes.
- **k-Nearest neighbor** (k-NN) [43] is a classification strategy that is an example of a "lazy learner". An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (where k is some user specified constant). If $k = 1$ (as is the case in this paper), then the object is simply assigned to the class of that single nearest neighbor. This method is more adequate for numerical data, although it can also deal with discrete values.
- A **Support Vector Machine** (SVM) [44] is a learning algorithm typically used for classification problems (text categorization, handwritten character recognition, image classification, etc.). More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. In its basic implementation, it can only work with numerical data and binary classes.

5. Experimental results

In this section we present and discuss the experimental results over eight microarray datasets in terms of (a) the number of selected features; (b) the classification accuracy; and (c) the feature selection runtime. Last but not least, we compare our best results with a wrapper approach and with other methods in the literature.

Four different approaches will be compared in the tables of this section: the centralized filter approach (CF), the distributed filter approach (DF), the distributed ranking filter approach (DRF) and the distributed ranking filter approach removing the features with Information Gain zero from the ranking (DRF0). The name of the specific filter used will be added at the end. For example, the distributed filter approach employing CFS will be represented as DF-CFS. Notice that with the DF method, the subsets of features are randomly generated so they are different in each iteration. For this reason, the experiments are run 5 times and their average results are shown in the tables.

5.1. Number of selected features

Table 2 displays the number of features selected by the centralized approach, which is independent of the classifier, as well as the original number of features for each dataset. Notice that the number of features selected by the ranker methods (Information Gain and ReliefF) are notably higher than those of the subset filters, because it is necessary to establish a percentage of features to

Table 2
Number of features selected by the centralized approach.

	Colon	Leukemia	CNS	DLBCL	Prostate	Lung	Ovarian	Breast
Full set	2000	7129	7129	4026	12,600	12,533	15,154	24,481
CFS	19	36	60	47	89	40	37	130
Cons	3	1	3	2	4	1	3	5
INT	16	36	47	36	73	40	27	102
IG10%	200	713	713	403	1260	1254	1516	2449
IG25%	500	1783	1783	1007	3150	3134	3789	6121
ReliefF10%	200	713	713	403	1260	1254	1516	2449
ReliefF25%	500	1783	1783	1007	3150	3134	3789	6121

retain. Tables 3 and 4 show the number of features selected by the distributed approaches on the eight datasets. Note that the features selected by the distributed approaches depend on the classifier because of the merging procedure.

From these tables it can be observed that the number of features selected by any method is considerably smaller than that of the full feature set. In Table 2, in the worst case 25% of the features are retained (due to the restriction we applied on ranker methods). As for Tables 3 and 4, the number of selected features is below 2% of the total number of features. Therefore, all the feature selection algorithms tested herein are able to reduce significantly the number of features as well as the storage requirements and the classification runtime.

In general, the feature selection method that selects the lowest number of features for any dataset is the consistency-based filter, especially in its centralized approach. The subset filters (CFS, consistency-based and INTERACT) tend to select a larger number of features when they are applied within a distributed approach. On the contrary, the ranker methods (Information Gain and ReliefF) reduce the number of features in the final subset when the process is distributed. This can be explained because with the centralized approach, they have no option but to select the established percentage of features, and this fixed number is smaller when it is applied to a subset of the data.

It is worth noticing that the centralized version of consistency-based, as well as some distributed approaches, only returned

Table 3
Number of features selected by the distributed approaches with C4.5 and naive Bayes classifiers.

		Colon	Leukemia	CNS	DLBCL	Prostate	Lung	Ovarian	Breast
C4.5	DF-CFS	8	11	8	6	50	11	95	20
	DRF-CFS	25	13	28	11	121	17	16	31
	DRF0-CFS	22	13	28	11	129	17	16	27
	DF-Cons	5	8	7	4	39	9	62	16
	DRF-Cons	5	1	12	2	21	1	22	24
	DRF0-Cons	10	1	11	2	13	1	22	6
	DF-INT	8	10	8	6	53	11	75	19
	DRF-INT	20	13	16	9	87	17	33	30
	DRF0-INT	18	13	26	9	88	17	33	27
	DF-IG10%	14	9	14	9	53	9	85	48
	DRF-IG10%	9	2	6	2	30	2	36	44
	DRF0-IG10%	9	2	4	2	30	2	27	16
	DF-IG25%	34	22	28	19	125	17	194	118
	DRF-IG25%	6	5	15	4	91	4	44	100
	DRF0-IG25%	6	5	5	4	65	4	44	50
	DF-ReliefF10%	17	11	14	11	49	9	70	41
	DRF-ReliefF10%	9	4	8	2	30	2	18	20
	DRF0-ReliefF10%	9	4	6	2	24	2	18	8
	DF-ReliefF25%	34	22	34	21	99	18	176	142
	DRF-ReliefF25%	12	5	20	4	52	4	66	80
	DRF0-ReliefF25%	12	5	10	4	26	4	66	10
NB	DF-CFS	11	12	12	10	58	9	97	24
	DRF-CFS	25	13	50	26	20	17	38	152
	DRF0-CFS	10	13	50	26	20	17	34	150
	DF-Cons	5	10	10	6	56	8	66	16
	DRF-Cons	12	4	28	5	5	3	3	33
	DRF0-Cons	5	4	17	5	5	3	3	33
	DF-INT	11	11	12	10	60	9	87	24
	DRF-INT	18	13	41	9	22	17	22	126
	DRF0-INT	15	13	34	9	21	17	22	123
	DF-IG10%	16	11	22	14	48	9	108	32
	DRF-IG10%	6	6	14	4	12	2	45	28
	DRF0-IG10%	3	6	8	4	12	2	36	20
	DF-IG25%	37	22	47	33	109	20	172	76
	DRF-IG25%	12	5	15	8	26	4	66	100
	DRF0-IG25%	6	5	5	8	39	4	22	30
	DF-ReliefF10%	19	11	16	14	50	10	99	44
	DRF-ReliefF10%	6	4	14	4	24	4	27	40
	DRF0-ReliefF10%	6	4	4	4	24	4	18	24
	DF-ReliefF25%	35	26	38	30	114	17	202	116
	DRF-ReliefF25%	6	10	25	4	13	8	88	40
	DRF0-ReliefF25%	12	10	10	4	13	8	66	20

Table 4
Number of features selected by the distributed approaches with k-NN and SVM classifiers.

		Colon	Leukemia	CNS	DLBCL	Prostate	Lung	Ovarian	Breast
k-NN	DF-CFS	12	16	9	8	70	11	77	22
	DRF-CFS	27	13	37	11	48	17	33	34
	DRF0-CFS	23	13	32	11	35	17	33	27
	DF-Cons	7	13	8	7	54	10	56	15
	DRF-Cons	12	4	19	6	48	3	22	44
	DRF0-Cons	9	4	18	2	31	3	23	54
	DF-INT	12	18	9	8	66	11	77	24
	DRF-INT	20	13	31	9	58	17	11	35
	DRF0-INT	7	13	27	9	51	17	11	27
	DF-IG10%	15	12	16	14	55	9	61	60
	DRF-IG10%	6	4	12	4	24	2	18	52
	DRF0-IG10%	6	4	6	4	30	2	18	32
	DF-IG25%	41	25	43	33	135	20	158	136
	DRF-IG25%	18	10	10	8	65	4	66	40
	DRF0-IG25%	6	10	5	8	65	4	44	10
	DF-ReliefF10%	16	12	16	11	46	10	68	42
	DRF-ReliefF10%	9	6	8	6	30	2	18	40
	DRF0-ReliefF10%	6	6	6	6	24	2	18	20
	DF-ReliefF25%	35	27	39	24	101	18	172	100
	DRF-ReliefF25%	12	15	30	12	65	4	22	50
	DRF0-ReliefF25%	6	15	15	8	52	4	22	40
SVM	DF-CFS	8	17	10	9	59	14	51	21
	DRF-CFS	12	13	52	11	62	17	16	82
	DRF0-CFS	10	13	64	11	113	17	16	82
	DF-Cons	7	15	5	7	54	10	41	21
	DRF-Cons	5	10	18	5	59	5	8	42
	DRF0-Cons	5	10	17	5	51	5	8	51
	DF-INT	8	18	10	9	68	13	47	18
	DRF-INT	7	13	78	9	65	17	11	116
	DRF0-INT	7	13	27	9	61	17	11	97
	DF-IG10%	14	14	13	16	52	9	49	56
	DRF-IG10%	3	8	2	4	30	2	45	44
	DRF0-IG10%	3	8	2	2	36	2	45	24
	DF-IG25%	35	34	57	30	117	18	84	130
	DRF-IG25%	12	15	30	12	26	4	66	100
	DRF0-IG25%	6	15	15	12	39	4	44	40
	DF-ReliefF10%	18	14	16	13	54	10	47	47
	DRF-ReliefF10%	6	6	12	6	30	2	18	44
	DRF0-ReliefF10%	6	6	8	6	18	2	18	24
	DF-ReliefF25%	32	26	43	22	99	18	84	102
	DRF-ReliefF25%	12	10	40	8	52	4	22	100
	DRF0-ReliefF25%	18	10	20	4	52	4	22	60

one feature for datasets Leukemia and Lung. These two datasets come originally divided in training and test datasets, with the data extracted under different conditions, which may hinder the process of feature selection and classification. In fact, the single feature returned by consistency-based filter on Leukemia was feature #4847. This feature in the training set can be used to clearly distinguish the target concept values, as shown in Fig. 3a. However, the same feature is not that informative in the test set and the class is not linearly separable, as displayed in Fig. 3b. For this reason, although 100% classification accuracy was obtained in these two datasets on the training set with a single feature (not included in the paper for the sake of brevity), the precision decreases on the test set (see Tables 5–7). A similar situation happens with Lung dataset, as can be seen in Fig. 3c and d.

This situation reflects the enormous complexity of microarray data, when in some cases training and test samples are recorded under completely different situations. For example, in the Leukemia dataset, the training samples are extracted from adult patients, whereas the test samples are obtained mainly from children.

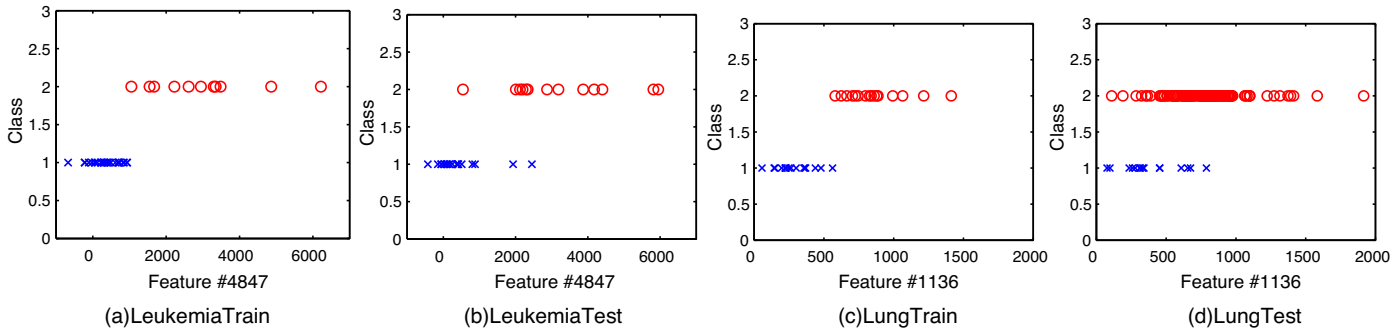

Fig. 3. Feature #4847 in Leukemia and feature #1136 in Lung.

Table 5
Test classification accuracy of C4.5.

		Colon	Leukemia	CNS	DLBCL	Prostate	Lung	Ovarian	Breast	Average
CFS	CF	85.00	**91.18**	50.00	**86.67**	26.47	81.88	**100.00**	68.42	73.70
	DF	76.00	74.71	69.00	80.00	35.88	91.28	95.24	58.95	72.63
	DRF	85.00	**91.18**	45.00	**86.67**	29.41	89.26	**100.00**	57.89	73.05
	DRF0	85.00	**91.18**	45.00	**86.67**	29.41	89.26	**100.00**	52.63	72.39
Cons	CF	85.00	**91.18**	50.00	**86.67**	23.53	81.88	**100.00**	68.42	73.33
	DF	80.00	70.00	62.00	84.00	36.47	90.60	95.95	65.26	73.04
	DRF	70.00	**91.18**	**80.00**	**86.67**	58.82	89.26	96.43	68.42	80.10
	DRF0	85.00	**91.18**	**80.00**	**86.67**	58.82	89.26	96.43	73.68	**82.63**
INT	CF	85.00	**91.18**	55.00	**86.67**	26.47	81.88	98.81	**78.95**	75.49
	DF	76.00	78.24	69.00	78.67	37.06	91.28	94.76	54.74	72.47
	DRF	85.00	**91.18**	55.00	**86.67**	29.41	89.26	98.81	52.63	73.49
	DRF0	85.00	**91.18**	60.00	**86.67**	29.41	89.26	98.81	52.63	74.12
IG 10%	CF	85.00	**91.18**	45.00	**86.67**	26.47	89.26	98.81	73.68	74.51
	DF	80.00	74.71	54.00	82.67	35.88	91.41	93.81	68.42	72.61
	DRF	**90.00**	**91.18**	55.00	**86.67**	23.53	89.26	**100.00**	63.16	74.85
	DRF0	85.00	91.18	40.00	**86.67**	38.24	89.26	**100.00**	73.68	75.50
IG 25%	CF	85.00	**91.18**	60.00	**86.67**	26.47	89.26	98.81	73.68	76.38
	DF	80.00	82.35	61.00	81.33	29.41	91.28	96.67	62.11	73.02
	DRF	85.00	**91.18**	65.00	**86.67**	23.53	89.26	98.81	47.37	73.35
	DRF0	80.00	**91.18**	65.00	**86.67**	26.47	89.26	98.81	52.63	73.75
Rel 10%	CF	85.00	**91.18**	45.00	**86.67**	29.41	**97.32**	98.81	63.16	74.57
	DF	75.00	78.24	60.00	**86.67**	25.88	91.28	96.90	65.26	72.40
	DRF	85.00	67.65	65.00	**86.67**	26.47	89.93	98.81	52.63	71.52
	DRF0	85.00	67.65	65.00	**86.67**	26.47	89.93	98.81	57.89	72.18
Rel 25%	CF	85.00	**91.18**	45.00	**86.67**	29.41	**97.32**	98.81	73.68	75.88
	DF	80.00	74.71	54.00	84.00	40.59	90.60	95.95	64.21	73.01
	DRF	**90.00**	**91.18**	65.00	**86.67**	**61.76**	89.93	98.81	47.37	78.84
	DRF0	85.00	**91.18**	65.00	**86.67**	41.18	89.93	98.81	42.11	74.98

5.2. Classification accuracy results

In this section we discuss the test classification accuracy obtained by C4.5, naive Bayes, k-NN and SVM classifiers with the centralized and distributed approaches. Notice that in these tables, the best results are marked in bold.

Table 5 reports the classification accuracy obtained by C4.5 for all datasets and approaches tested. The centralized approach

Table 6
Test classification accuracy of naive Bayes.

		Colon	Leukemia	CNS	DLBCL	Prostate	Lung	Ovarian	Breast	Average
CFS	CF	**90.00**	94.12	**70.00**	93.33	26.47	**100.00**	97.62	36.84	76.05
	DF	84.00	72.94	57.00	92.00	25.29	87.79	95.95	49.47	70.56
	DRF	85.00	88.24	60.00	93.33	20.59	98.66	**100.00**	36.84	72.83
	DRF0	**90.00**	88.24	60.00	93.33	20.59	98.66	**100.00**	36.84	73.46
Cons	CF	85.00	91.18	55.00	86.67	**32.35**	85.91	**100.00**	36.84	71.62
	DF	85.00	74.12	53.00	90.67	26.47	86.71	95.00	62.11	71.63
	DRF	**90.00**	94.12	55.00	73.33	26.47	94.63	97.62	36.84	71.00
	DRF0	85.00	94.12	65.00	73.33	26.47	94.63	97.62	36.84	71.63
INT	CF	85.00	94.12	65.00	93.33	26.47	**100.00**	**100.00**	36.84	75.10
	DF	84.00	79.41	57.00	90.67	26.47	88.05	93.33	52.63	71.45
	DRF	90.00	88.24	65.00	93.33	23.53	98.66	98.81	36.84	74.30
	DRF0	85.00	88.24	65.00	93.33	23.53	98.66	98.81	36.84	73.68
IG 10%	CF	85.00	**97.06**	60.00	93.33	26.47	98.66	92.86	36.84	73.78
	DF	81.00	73.53	59.00	**94.67**	24.71	90.47	92.14	42.11	69.70
	DRF	85.00	94.12	60.00	86.67	26.47	90.60	100.00	42.11	73.12
	DRF0	85.00	94.12	55.00	86.67	26.47	90.60	97.62	31.58	70.88
IG 25%	CF	80.00	**97.06**	50.00	93.33	26.47	98.66	92.86	36.84	71.90
	DF	83.00	72.35	61.00	89.33	26.47	92.62	91.19	38.95	69.36
	DRF	85.00	94.12	35.00	86.67	26.47	96.64	98.81	36.84	69.94
	DRF0	85.00	94.12	55.00	86.67	26.47	96.64	97.62	36.84	72.30
Rel 10%	CF	85.00	94.12	65.00	93.33	26.47	99.33	92.86	68.42	78.07
	DF	84.00	73.53	60.00	85.33	25.29	92.21	94.05	73.68	73.51
	DRF	85.00	82.35	**70.00**	80.00	23.53	98.66	**100.00**	**78.95**	77.31
	DRF0	85.00	82.35	60.00	80.00	23.53	98.66	**100.00**	**78.95**	76.06
Rel 25%	CF	85.00	94.12	55.00	93.33	26.47	99.33	92.86	63.16	76.16
	DF	79.00	77.65	65.00	84.00	26.47	88.72	91.67	65.26	72.22
	DRF	85.00	91.18	65.00	86.67	23.53	98.66	98.81	**78.95**	**78.47**
	DRF0	85.00	91.18	55.00	86.67	23.53	98.66	**100.00**	73.68	76.71

Table 7
Test classification accuracy of k-NN.

		Colon	Leukemia	CNS	DLBCL	Prostate	Lung	Ovarian	Breast	Average
CFS	CF	80.00	85.29	65.00	86.67	32.35	**100.00**	**100.00**	63.16	76.56
	DF	79.00	78.24	61.00	88.00	31.18	91.68	99.05	57.89	73.25
	DRF	80.00	88.24	55.00	**93.33**	47.06	97.32	98.81	57.89	77.21
	DRF0	80.00	88.24	55.00	**93.33**	**61.76**	97.32	98.81	73.68	**81.02**
Cons	CF	85.00	91.18	65.00	73.33	26.47	81.88	**100.00**	47.37	71.28
	DF	79.00	76.47	59.00	82.67	32.94	88.72	97.14	58.95	71.86
	DRF	80.00	**94.12**	60.00	80.00	26.47	93.96	98.81	63.16	74.56
	DRF0	75.00	**94.12**	65.00	80.00	26.47	93.96	98.81	68.42	75.22
INT	CF	80.00	85.29	60.00	86.67	32.35	**100.00**	**100.00**	52.63	74.62
	DF	77.00	78.24	61.00	88.00	38.82	91.68	97.86	56.84	73.68
	DRF	70.00	88.24	60.00	**93.33**	32.35	97.32	**100.00**	73.68	76.87
	DRF0	85.00	88.24	60.00	**93.33**	41.18	97.32	**100.00**	68.42	79.19
IG 10%	CF	90.00	76.47	50.00	86.67	26.47	98.66	97.62	73.68	74.95
	DF	80.00	75.29	59.00	85.33	34.71	88.99	99.05	62.11	73.06
	DRF	70.00	88.24	70.00	86.67	38.24	90.60	**100.00**	**78.95**	77.84
	DRF0	80.00	88.24	65.00	86.67	52.94	90.60	**100.00**	68.42	78.98
IG 25%	CF	**95.00**	79.41	60.00	80.00	38.24	99.33	96.43	73.68	77.76
	DF	82.00	73.53	54.00	81.33	34.71	92.48	97.14	65.26	72.56
	DRF	80.00	91.18	45.00	86.67	26.47	97.32	97.62	63.16	73.43
	DRF0	70.00	91.18	50.00	86.67	26.47	97.32	98.81	47.37	70.98
Rel 10%	CF	90.00	76.47	50.00	86.67	26.47	98.66	97.62	73.68	74.95
	DF	83.00	80.00	56.00	90.67	37.06	94.23	98.10	54.74	74.22
	DRF	70.00	82.35	65.00	86.67	26.47	96.64	**100.00**	73.68	75.10
	DRF0	80.00	85.29	60.00	86.67	50.00	96.64	**100.00**	73.68	79.04
Rel 25%	CF	85.00	73.53	55.00	86.67	29.41	97.99	96.43	**78.95**	75.37
	DF	79.00	75.29	54.00	84.00	34.71	93.56	95.95	63.16	72.46
	DRF	80.00	91.18	**75.00**	**93.33**	23.53	97.99	**100.00**	63.16	78.02
	DRF0	55.00	91.18	**75.00**	**93.33**	29.41	97.99	**100.00**	52.63	74.32

achieves the best result for two datasets (Lung and Breast), whilst for the remaining datasets, the distributed approach DRF outperforms the other methods with the Colon, CNS and Prostate datasets. It is worth mentioning the case of the CNS dataset, in which DRF and DRF0 using the consistency-based filter surpassed the best centralized results by 20%. As for the Prostate dataset, the results obtained with DRF combined with ReliefF retaining 25% of the features outperformed the best centralized results by 32.35%. In terms of average accuracy for all datasets, the best option is DRF0 using the consistency-based filter. It seems that, in general, the features with no information gain with respect to the class are not relevant to the prediction task.

From Table 6 the classification accuracy reported by naive Bayes for the eight datasets at hand can be observed. For some datasets the best choice is the centralized approach (Leukemia, Prostate and Lung), whereas for others it is better to apply a distributed method (DLBCL and Breast). The differences between the centralized and the distributed approach are not so prominent as with the C4.5 classifier. Even still, the case of Breast dataset can be emphasized, in which DRF combined with ReliefF outperforms the best centralized result in more than 10%. In fact, the best result on average for all datasets was achieved by DRF together with ReliefF when retaining the top 25% of features in each partition.

Table 7 shows the results obtained by k-NN. On the one hand, the highest accuracy for the datasets Colon and Lung was reported by a centralized approach. On the other hand, the distributed approach performed better with Leukemia, CNS, DLBCL and Prostate datasets. In general, focusing on the distributed approach, the methods which include a ranking as a first step (DRF and DRF0) obtain better results than DF, which divides the data randomly. As a matter of fact, the method with the best average accuracy for all datasets was DRF0 combined with CFS.

Finally, from Table 8 we can infer that the SVM classifier is very suitable for application to microarray data, since it achieves high classification accuracies on average for all datasets. Notice that this classifier can successfully handle these kinds of datasets with a much higher number of features than samples. Particularly, the best result in terms of average accuracy was obtained by DRF0 combined with Information Gain when retaining the top 25% of features in each partition. It is worth noting the excellent result for the Prostate dataset (obtained by CF with INTERACT and DRF with Information Gain 10%), which outperforms in more than 35% for the highest accuracy reported by the other classifiers.

Since for the CF, DRF and DRF0 approaches the experiments can be run only once, it is not possible to conduct statistical tests to check if the differences among the methods are statistically significant. Nevertheless, one can see easily from Tables 5–8 that the average results are very similar. In any case, the best average accuracy was obtained by a distributed approach for all the classifiers tested. Therefore we can affirm that, in terms of classification accuracy, our proposed distributed approaches at least maintain the performance compared with that of the standard centralized approach.

5.3. Runtime

Table 9 reports the runtime of the feature selection algorithms studied applied to the eight microarray datasets. In the distributed approaches (DF, DRF and DRF0), all the subsets can be processed at the same time, so the time displayed in the table is the maximum of the times required by the filter for all the subsets generated at the partitioning stage.

As expected, when applying a distributed approach the time is reduced for all datasets. It is worth pointing out the case of the Brain dataset combined with the CFS filter, in which the centralized approach employed almost 4 h whilst the time required by the distributed approaches was under 1 s. Having said this, it is necessary to remember that the distributed approaches also have a stage for combining the results from the different partitions. However, in this case, the time required for merging the features is in the order

Table 8
Test classification accuracy of SVM.

		Colon	Leukemia	CNS	DLBCL	Prostate	Lung	Ovarian	Breast	Average
CFS	CF	**85.00**	82.35	65.00	**93.33**	91.18	98.66	**100.00**	73.68	86.15
	DF	76.00	82.35	67.00	89.33	66.47	91.01	99.05	67.37	79.82
	DRF	80.00	88.24	50.00	86.67	76.47	95.30	98.81	73.68	81.15
	DRF0	80.00	88.24	70.00	86.67	85.29	95.30	98.81	73.68	84.75
Cons	CF	70.00	73.53	65.00	86.67	26.47	52.35	98.81	52.63	65.68
	DF	78.00	73.53	64.00	89.33	68.24	91.81	98.81	61.05	78.10
	DRF	75.00	67.65	**80.00**	80.00	35.29	91.28	98.81	68.42	74.56
	DRF0	75.00	85.29	70.00	80.00	26.47	91.28	98.81	63.16	73.75
INT	CF	80.00	82.35	55.00	86.67	**97.06**	98.66	**100.00**	73.68	84.18
	DF	76.00	78.82	67.00	90.67	78.82	91.14	98.57	67.37	81.05
	DRF	75.00	88.24	70.00	**93.33**	91.18	95.30	98.81	73.68	85.69
	DRF0	75.00	88.24	60.00	**93.33**	82.35	95.30	98.81	**84.21**	84.66
IG 10%	CF	80.00	**94.12**	60.00	**93.33**	82.35	**99.33**	98.81	73.68	85.20
	DF	77.00	78.82	59.00	88.00	57.65	86.04	98.33	68.42	76.66
	DRF	75.00	82.35	65.00	86.67	**97.06**	74.50	**100.00**	68.42	81.12
	DRF0	75.00	88.24	65.00	86.67	85.29	74.50	**100.00**	63.16	79.73
IG 25%	CF	80.00	**94.12**	65.00	**93.33**	73.53	**99.33**	98.81	57.89	82.75
	DF	74.00	78.24	65.00	84.00	60.00	94.23	99.05	60.00	76.81
	DRF	**85.00**	91.18	60.00	**93.33**	79.41	97.99	**100.00**	63.16	83.76
	DRF0	75.00	91.18	**80.00**	**93.33**	79.41	97.99	98.81	73.68	**86.18**
Rel 10%	CF	70.00	91.18	65.00	**93.33**	79.41	**99.33**	98.81	57.89	81.87
	DF	77.00	78.82	57.00	90.67	68.24	91.54	98.10	67.37	78.59
	DRF	80.00	79.41	60.00	80.00	91.18	95.97	**100.00**	52.63	79.90
	DRF0	80.00	82.35	65.00	80.00	70.59	95.97	**100.00**	63.16	79.63
Rel 25%	CF	75.00	88.24	65.00	**93.33**	76.47	**99.33**	98.81	63.16	82.42
	DF	75.00	78.24	63.00	90.67	69.41	93.69	98.57	70.53	79.89
	DRF	75.00	85.29	75.00	86.67	73.53	96.64	**100.00**	57.89	81.25
	DRF0	80.00	85.29	70.00	86.67	64.71	96.64	**100.00**	68.42	81.47

of minutes, therefore our proposed method is able to shorten the execution time impressively compared to the standard version of the filter algorithm. Notice also that the time required by the first step of the DRF and DRF0 approaches (ordering the features according to their Information Gain) is the same as the time showed by the centralized IG (row 9 in Table 9) and it is under 4 s for all the datasets tested.

Regarding the time complexity of the distributed approaches proposed herein, it depends on the filter used. The time complexities of the three filters employed can be uniformly described as $O(m \cdot f(s))$, where m is the number of samples, s is the number of features, and $f(s)$ is a function of s associated to each filter. And the order of $f(s)$ is higher than linear [33,34,36–38]. Since these algorithms are more affected by the number of features than by the number of samples, it is not surprising that the distributed approach, which divides the data vertically and hence works with smaller subsets of features than the standard one, could be able to shorten the runtime so significantly.

In light of the above, we can conclude that our distributed proposals performed successfully, since the running time was

Table 9
Runtime (in seconds) for the feature selection methods tested.

		Colon	Leukemia	CNS	DLBCL	Prostate	Lung	Ovarian	Breast
CFS	CF	4.92	7.31	198.14	47.52	1225.87	13.73	202.62	13,323.50
	DF	0.18	0.19	0.17	0.17	0.27	0.16	0.38	0.22
	DRF	0.20	0.25	0.35	0.18	0.32	0.28	0.42	0.28
	DRF0	0.19	0.19	0.18	0.18	0.31	0.18	0.42	0.28
Cons	CF	1.00	2.73	3.06	1.39	8.83	4.89	14.40	25.26
	DF	0.20	0.21	0.20	0.19	0.43	0.19	0.47	0.28
	DRF	0.22	0.20	0.20	0.19	0.40	0.20	0.49	0.29
	DRF0	0.23	0.20	0.21	0.19	0.34	0.19	0.50	0.30
INT	CF	1.64	15.47	12.75	3.69	123.10	38.23	224.72	285.21
	DF	0.24	0.24	0.24	0.23	0.31	0.23	0.48	0.28
	DRF	0.24	0.27	0.23	0.22	0.33	0.33	0.48	0.29
	DRF0	0.24	0.24	0.23	0.22	0.34	0.23	0.53	0.29
IG	CF	0.66	1.08	1.11	0.86	1.79	1.49	3.51	3.32
	DF	0.16	0.17	0.17	0.16	0.24	0.16	0.31	0.21
	DRF	0.16	0.17	0.16	0.16	0.23	0.28	0.31	0.25
	DRF0	0.16	0.16	0.16	0.15	0.24	0.15	0.32	0.22
ReliefF	CF	0.61	1.14	1.19	0.78	4.50	1.56	12.92	8.14
	DF	0.19	0.20	0.20	0.28	0.27	0.18	0.34	0.24
	DRF	0.20	0.19	0.19	0.18	0.28	0.24	0.34	0.24
	DRF0	0.18	0.19	0.18	0.18	0.28	0.18	0.34	0.24

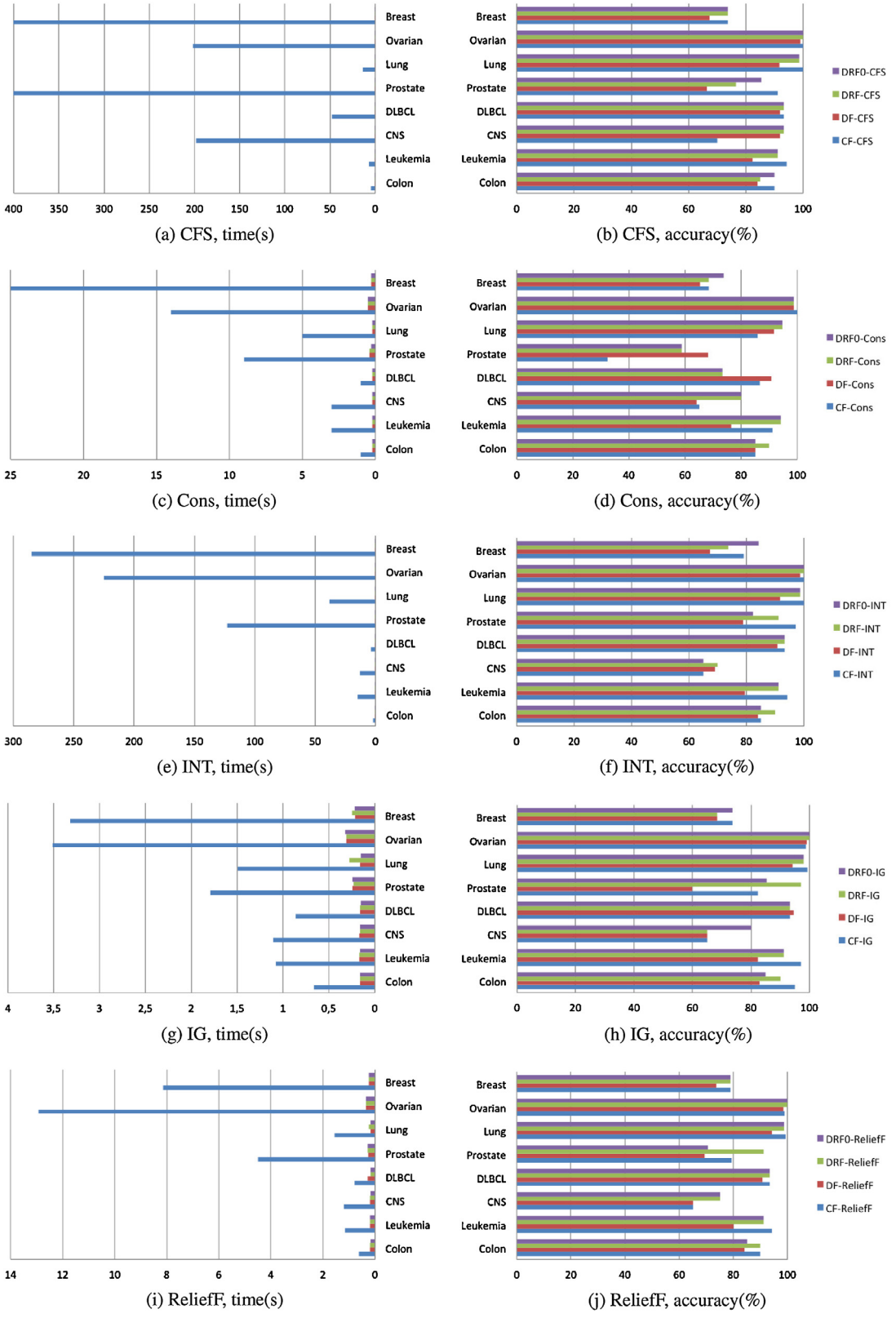

Fig. 4. Comparison of accuracy and time.

considerably reduced and the accuracy did not drop to inadmissible values. On the contrary, our approaches are able to match and in some cases even improve upon the standard algorithms applied to non-partitioned datasets. This situation is reflected in Fig. 4, where the best result among all the classifiers is displayed for any dataset and feature selection method. It is easy to see at a glance that the accuracies fall into similar values whilst the differences in time are outstanding. Note that in Fig. 4a the runtimes for Breast (13,323 s) and Prostate (1226 s) had to be cut for the sake of visualization.

Table 10
Test classification accuracy for the different merging procedures.

Method	Colon	Leukemia	CNS	DLBCL	Prostate	Lung	Ovarian	Breast
Proposed merging	90.00	94.12	80.00	94.67	97.06	98.66	100.00	84.21
Basic union	95.00	97.06	80.00	94.67	94.12	100.00	100.00	84.21

5.4. Study of other alternatives

The adequacy of the proposed approaches for distributed feature selection has been demonstrated, since the running time was considerably reduced while maintaining the classification accuracy. However, an arising question could be if it is worth applying the proposed merging procedure (which involves a classifier) instead of a basic merging of the features (such as the union or the intersection). In order to answer this question, further experiments were performed using the union as a merging procedure. Notice that it is not possible to apply the intersection since the whole set of features is split without replacement over the different nodes, i.e. the subsets of features are disjointed.

Regarding the classification accuracy, Table 10 shows the best result for classifier and method for each dataset, comparing the proposed merging procedure with the basic union. As can be seen, in some cases the highest accuracy is obtained by the basic union (Colon, Leukemia and Lung), in one of the cases the highest accuracy is obtained by the proposed merging procedure (Prostate) and, in most of the cases, there are no differences in classification accuracy. It is worth highlighting the good behavior or the proposed merging method on Prostate dataset, which poses a big challenge for machine learning methods since the test dataset was extracted from a different experiment and has a nearly 10-fold difference in overall microarray intensity from the training data. In fact, as can be seen in Table 1, the test distribution differs significantly from the train distribution and with an inappropriate feature selection, some classifiers just assign all the samples to one of the classes (see Table 5).

The results achieved by the basic union are competitive in terms of classification accuracy. However, they are obtained at the expense of retaining a much higher number of features than our proposed merging procedure. In fact, as can be seen in Tables 3 and 4, in general our method selects in the order of tens of features. However, by applying the basic union to merge the features, this number increases up to the order of thousands. As an example, we will analyze the cases in which the union achieved the highest accuracy (Colon, Leukemia and Lung) as can be seen in Table 10. For the Colon dataset, DRF-IG with the basic union obtained 95% of classification accuracy, selecting 285 features. The highest accuracy for this dataset with the proposed merging method was obtained by DRF-Cons, which selected only 12 features. As for the Leukemia dataset, DRF0-CFS with the basic union reported 97.06% classification accuracy at the expense of retaining 689 features, whilst DRF0-Cons with the proposed merge degraded the accuracy by 3% but selected 4 features. In the case of the Lung dataset, DRF0-ReliefF25 obtained 100% of accuracy when using the basic union and selected 540 features, whilst DRF0-CFS with the proposed merging achieved an accuracy of 98.66% and retained 17 features. A more extreme case occurred for the Lung dataset and the CFS filter, in which our merging procedure selected between 9 and 17 features (depending on the distributed approach chosen) whilst, when using the basic union, the number of selected features varied between 1849 and 2518. Having such a high number of features might lead the classifiers to learn on redundant features. Besides, in microarray data classification, it is crucial to reduce the dimensionality in order to help biologists identify the underlying mechanism that relates gene expression to diseases.

In light of the results reported in this subsection, the authors suggest to use the proposed merging procedure. Although in some cases there is an improvement in classification accuracy when using the basic union, it implies selecting a much higher number of features, which can hamper the biologists' task.

5.5. Comparison with the wrapper approach

As explained in Section 1, wrappers are a type of feature selection method. The wrapper model involves a learning algorithm as a black box and consists of using its prediction performance to assess the relative usefulness of subsets of variables. In other words, the feature selection algorithm uses the learning algorithm as a subroutine with the computational cost that comes from calling the learning algorithm to evaluate each subset of features. However, this interaction with the classifier tends to give better performance results than filters and embedded methods [45]. This is probably due to the fact that the relevant feature subset could not reflect the classifier's specific characteristics.

Bearing in mind that this model tends to obtain better performance than filters, one may think of using wrappers when dealing with microarray data. However, the extremely large number of features makes their application impossible on most standard computers. A possible solution could be to distribute the wrapper execution into smaller subsets of features. For this sake, some experiments were performed executing a distributed wrapper with the ranking partitioning (DRW). The four classifiers used in this research were also used as the learning algorithm of the wrapper.

In order to determine if the distributed wrapper makes the difference on microarray data, a comparison with the distributed filter is visualized in Table 11. Only the best result among all the four classifiers is displayed for each method, for the sake of brevity.

In terms of average accuracy, the distributed wrapper (DRW) is the best option, but only outperforms DRF0-IG by 0.55%. Moreover, only in one out of the eight datasets (Prostate) the wrapper obtains the highest accuracy. As commented previously, the Prostate dataset is a complex problem in which some classifiers just assign all the samples to one of the classes. In this case, the wrapper approach takes advantage of the interaction with the classifier to obtain the highest accuracy and this difference on Prostate dataset is the reason why it obtains the best accuracy on average. As a matter of fact, in the remaining datasets, there is always a distributed filter which matches or even improves the result obtained by the wrapper. If we recalculate the average accuracy disregarding the Prostate dataset, DRF0-IG appears to be the best method (89.16%), followed by DRF0-ReliefF (88.87%), DRF0-Cons (88.27%), DRF0-INT (88.20%) and DRF0-CFS (88.12%). DRW, however, reports the worst average accuracy (88.10%).

Since one of the main disadvantages of the wrapper model is its high computational cost, it is worth focusing on that issue. By distributing the features into small subsets, the runtime for the wrapper in each subset has dropped to very acceptable values. In fact, in some cases it is very similar to the one required by the filters. However, the execution time of the wrapper is significantly affected by the number of features in each subset. In datasets such as Prostate, Ovarian and Breast with SVM, the distributed filter takes 13%, 10% and 17% of the time required by the wrapper, respectively.

Table 11
Test classification accuracy of a distributed filter and a distributed wrapper. Best results highlighted in bold font.

Method	Colon	Leukemia	CNS	DLBCL	Prostate	Lung	Ovarian	Breast	Average
DRF0-CFS	**90.00**	91.18	70.00	**93.33**	85.29	**98.66**	**100.00**	73.68	87.77
DRF0-Cons	**90.00**	**94.12**	**80.00**	86.67	58.82	94.63	98.81	73.68	84.59
DRF0-INT	85.00	91.18	65.00	**93.33**	82.35	**98.66**	**100.00**	**84.21**	87.47
DRF0-IG	85.00	**94.12**	**80.00**	**93.33**	85.29	97.99	**100.00**	73.68	88.68
DRF0-Rel	85.00	91.18	75.00	**93.33**	70.59	**98.66**	**100.00**	78.95	86.59
DRW	85.00	91.18	75.00	**93.33**	**97.06**	93.29	**100.00**	78.95	**89.23**

Table 12
Comparative study in terms of test classification accuracy. Best results highlighted in bold font.

	Method	Colon	Leukemia	CNS	DLBCL	Prostate	Lung	Ovarian	Average
Proposed	DF-CFS	84.00	82.35	92.00	92.00	66.47	91.68	99.05	86.79
	DRF-CFS	85.00	91.18	**93.33**	93.33	76.47	98.66	**100.00**	91.14
	DRF0-CFS	90.00	91.18	**93.33**	93.33	85.29	98.66	**100.00**	**93.11**
	DF-Cons	85.00	76.47	64.00	90.67	68.24	91.81	98.81	82.14
	DRF-Cons	90.00	**94.12**	80.00	73.33	58.82	94.63	98.81	84.24
	DRF0-Cons	85.00	**94.12**	80.00	73.33	58.82	94.63	98.81	83.53
	DF-INT	84.00	79.41	69.00	90.67	78.82	91.68	98.57	84.59
	DRF-INT	90.00	91.18	70.00	93.33	91.18	98.66	**100.00**	90.62
	DRF0-INT	85.00	91.18	65.00	93.33	82.35	98.66	**100.00**	87.93
	DF-IG	83.00	82.35	65.00	**94.67**	60.00	94.23	99.05	82.61
	DRF-IG	90.00	91.18	65.00	93.33	**97.06**	97.99	**100.00**	90.65
	DRF0-IG	85.00	91.18	80.00	93.33	85.29	97.99	**100.00**	90.40
	DF-ReliefF	84.00	80.00	65.00	90.67	69.41	94.23	98.57	83.13
	DRF-ReliefF	90.00	91.18	75.00	93.33	91.18	98.66	**100.00**	91.34
	DRF0-ReliefF	85.00	91.18	75.00	93.33	70.59	98.66	**100.00**	87.68
SotA	MWMR	82.00	–	–	92.80	–	–	–	–
	Ensemble	85.00	91.18	70.00	93.33	**97.06**	**100.00**	**100.00**	90.94
Filters	CFS	90.00	**94.12**	70.00	93.33	**97.06**	**100.00**	**100.00**	92.07
	Cons	85.00	91.18	65.00	86.67	32.35	85.91	**100.00**	78.02
	INT	85.00	**94.12**	65.00	93.33	70.59	**100.00**	100.00	86.86
	IG	85.00	**94.12**	70.00	93.33	**97.06**	99.33	**100.00**	91.26
	ReliefF	85.00	91.18	70.00	93.33	94.12	**100.00**	**100.00**	90.52
Classifiers	C4.5	90.00	91.18	60.00	86.67	26.47	81.88	98.81	76.43
	NB	70.00	88.24	60.00	93.33	26.47	95.30	88.10	74.49
	k-NN	**95.00**	70.59	55.00	73.33	52.94	97.99	92.86	76.82
	SVM	75.00	85.29	70.00	86.67	52.94	99.33	**100.00**	81.32

5.6. Comparative study

Finally, we compare our best results with those obtained by other methods in the literature. Carrying out a fair comparison between our methods and those described in Section 1 is not an easy task. Unfortunately, there are many microarray datasets in the literature, some of them with the same name but different characteristics, and with different training/test partitions or validations. In this situation, we found results for two of the datasets employed in this work in [20] (Colon and DLBCL). They proposed a filter based on a maximum weight and minimum redundancy (MWMR) criterion. With this method, it is possible to select the feature subset in which the features are more beneficial to the subsequent tasks (such as clustering or classification) while the redundancy among them is minimal. As in this research, 2/3 were randomly selected from each database for training and the rest of the samples were used for testing.

To widen the scope of this comparison, the results obtained in [7] are included, in which we performed an extensive study over microarray data using the same partition of the datasets. Table 12 displays the best results for our proposed distributed approaches (rows 1–15), two state of the art (SofA) methods; the MWMR method proposed in [20] and an ensemble of filters proposed in [6] (rows 16 and 17), five filters widely applied to microarray data in a centralized fashion (rows 18–22); CFS, consistency-based, INTERACT, Information Gain and ReliefF, respectively) and the four classifiers considered in this study when no feature selection is applied (rows 23–26). Note that the last column reports the average for all the microarray datasets and that Breast dataset could not be compared because it was not included in the experiments carried out in [7].

From Table 12 one can notice that the best approach in terms of average is our proposed DRF0-CFS. It is interesting to observe that this distributed filter outperforms the ensemble approach. Ensemble learning has recently been the focus of much attention, its strength is its capability to reduce the variability associated to feature selection methods and obtaining a method that could be applied over any dataset regardless of its characteristics. Additionally, the distributed method presented herein is more computationally efficient, since the set of features is split into different subsets, whilst the ensemble method applies each filter over the whole dataset.

Regarding the results achieved by the classifiers alone, we demonstrated the adequacy of addressing feature selection on microarray data, considering that the classifiers obtained the worst performance on average. It is also worth commenting that the consistency-based filter significantly improves its effectiveness by using it with distributed data, which suggests that its performance degrades with large amounts of features.

6. Discussion

In the previous section the experimental results of our distributed approaches have been shown in terms of number of selected features, runtime and classification accuracy. Moreover,

the most accurate results were compared with those of the wrapper approach and other methods in the literature.

In light of the above, the most important advantage of our distributed method is the large reduction in execution time whilst maintaining accuracy at reasonable levels, and sometimes even improving it. This outcome might be based on the idea of *divide-and-conquer* since, in some cases, the result obtained by the learner can be more accurate if it is focused on a local region of the data.

Moreover, the proposed method has the additional advantage of allowing an easy parallel implementation. The application of the filter algorithm to each subset of features is independent of all the remaining subsets, so all the subsets can be processed at the same time. Even for the random partitioning, different rounds can be run at once. It is worth mentioning that there is little communication among the nodes of the parallel execution, since it occurs only in the combination step to compute the final subset of selected features and in the ranking step in the case of ranking partitioning. Regarding the complexity of the method, it is determined by the filter method chosen and the number of features in each subset, so it is not higher than that of the standard algorithm used.

Two main different versions of the distributed algorithm have been proposed: the distributed ranking filter (DRF) and the distributed filter (DF), which performs a random partition of the data. DRF and its variant DRF0, using Information Gain in the first ranking stage, obtained the best results in terms of accuracy. However, as mentioned in Section 1, the reason for which a user would like to apply distributed feature selection is two-fold: (i) data are sometimes distributed in multiple locations and often with multiple partitions; and (ii) most existing feature selection algorithms do not scale well and their efficiency deteriorates significantly or even becomes inapplicable when dealing with large-scale data. In the first scenario, an ordered ranking of the features as a first step is not possible, since features are distributed in origin, therefore the distributed algorithm cannot take advantage of the ranking provided by Information Gain. On the other hand, there might be cases in which the extremely huge amount of features prevent the use of Information Gain in order to obtain a previous ordered ranking of the features, so the random partition (DF) is the only option left. To sum up, we consider it interesting to count on these two different options for distributed feature selection, although we recommend using DRF or DRF0 if possible.

7. Conclusions

In this research we have proposed a new method for distributing the feature selection process applied to a complex scenario such as microarray data classification. These data are characterized by having a much larger number of features than of samples, so the idea was to distribute the features into disjoint subsets and then combine the results of the light feature-selectors into a final set of features.

The main goal was to design a method that would be able to successfully distribute the feature selection process. The experiments on eight microarray datasets showed that our proposal was able to reduce the running time significantly with respect to the standard (centralized) filtering algorithms as well as to the number of input features. In terms of execution time, the behavior was excellent, this fact being the most important advantage of the proposed method. Furthermore, with regard to classification accuracy, the distributed approach was able to match, and even in some cases improve, the standard algorithms applied to the non-partitioned datasets.

The extensive experiments performed herein included a comparison with well-known and highly recommended feature selection algorithms, involving also the wrapper approach. Results demonstrated that our method achieved a similar performance than this suite of standard algorithms, while shortening the running time impressively.

Finally, it is worth mentioning that the proposed method can be used with any feature selection algorithm without any modifications, so it could be seen as a general framework for distributed feature selection.

Acknowledgments

This research has been economically supported in part by the Ministerio de Economía y Competitividad of the Spanish Government through the research project TIN 2012-37954, partially funded by FEDER Funds of the European Union; and by the Consellería de Industria of the Xunta de Galicia through the research project GRC2014/035.

References

[1] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[2] I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh, Feature Extraction: Foundations and Applications, vol. 207, Springer, 2006.
[3] D.W. Opitz, Feature selection for ensembles, in: AAAI/IAAI, 1999, pp. 379–384.
[4] Z. Zheng, G.I. Webb, Stochastic Attribute Selection Committees, Springer, 1998.
[5] S.D. Bay, Combining nearest neighbor classifiers through multiple feature subsets, in: ICML, vol. 98, 1998, pp. 37–45, Citeseer, 1998.
[6] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, An ensemble of filters and classifiers for microarray data classification, Pattern Recognit. 45 (1) (2012) 531–539.
[7] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, Data classification using an ensemble of filters, Neurocomputing 135 (2014) 13–20.
[8] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.
[9] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (2004) 1205–1224.
[10] D. Koller, M. Sahami, Toward optimal feature selection, in: 13th International Conference on Machine Learning, vol. 28, 1995, pp. 4–292.
[11] L. Yu, H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, in: The 20th International Conference on Machine Learning, vol. 85, 2003, pp. 6–863.
[12] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, J. Bioinform. Comput. Biol. 3 (02) (2005) 185–205.
[13] P.K. Chan, S.J. Stolfo, et al., Toward parallel and distributed learning by meta-learning, in: AAAI workshop in Knowledge Discovery in Databases, 22, 1993, pp. 7–240.
[14] V.S. Ananthanarayana, D.K. Subramanian, M.N. Murty, Scalable, distributed and dynamic mining of association rules, in: High Performance Computing HiPC 2000, 2000, pp. 9–566.
[15] G. Tsoumakas, I. Vlahavas, Distributed data mining of large classifier ensembles, in: Proceedings Companion Volume of the Second Hellenic Conference on Artificial Intelligence, 2002, pp. 249–256.
[16] S. McConnell, D.B. Skillicorn, Building predictors from vertically distributed data, in: Proceedings of the 2004 Conference of the Centre for Advanced Studies on Collaborative research, IBM Press, 2004, pp. 150–162.
[17] D.B. Skillicorn, S.M. McConnell, Distributed prediction from vertically partitioned data, J. Parallel Distrib. Comput. 68 (1) (2008) 16–36.
[18] A.J. Ferreira, M.A.T. Figueiredo, An unsupervised approach to feature discretization and selection, Pattern Recognit. 45 (9) (2012) 3048–3060.
[19] A. Smola, A. Gretton, J. Bedo, K. Borgwardt, Feature selection via dependence maximization, J. Mach. Learn. Res. 13 (2012) 1393–1434.
[20] J. Wang, L. Wu, J. Kong, Y. Li, B. Zhang, Maximum weight and minimum redundancy: a novel framework for feature subset selection, Pattern Recognit. 46 (6) (2013) 1616–1627.
[21] L.-Y. Chuang, C.-H. Yang, K.-C. Wu, C.-H. Yang, A hybrid feature selection method for DNA microarray data, Comput. Biol. Med. 41 (4) (2011) 228–237.
[22] C.-P. Lee, Y. Leu, A novel hybrid feature selection method for microarray data analysis, Appl. Soft Comput. 11 (1) (2011) 208–213.
[23] S. Maldonado, R. Weber, J. Basak, Simultaneous feature selection and classification using kernel-penalized support vector machines, Inf. Sci. 181 (1) (2011) 115–128.
[24] A. Sharma, S. Imoto, S. Miyano, A top-R feature selection algorithm for microarray gene expression data, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (3) (2012) 754–764.
[25] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, A. Nowé, A survey on filter techniques for feature selection in gene expression microarray analysis, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (4) (2012) 1106–1119.
[26] K. Das, K. Bhaduri, H. Kargupta, A local asynchronous distributed privacy preserving feature selection algorithm for large peer-to-peer networks, Knowl. Inform. Syst. 24 (3) (2010) 341–367.

[27] L. Rokach, Taxonomy for characterizing ensemble methods in classification tasks: a review and annotated bibliography, Comput. Stat. Data Anal. 53 (12) (2009) 4046–4072.
[28] M. Banerjee, S. Chakravarty, Privacy preserving feature selection for distributed data using virtual dimension, in: Proceedings of the 20th ACM international conference on Information and Knowledge Management, ACM, 2011, pp. 2281–2284.
[29] Z. Zhao, J. Cox, D. Duling, W. Sarle, Massively parallel feature selection: an approach based on variance preservation, Mach. Learn. Knowl. Discov. Databases 23 (2012) 7–252.
[30] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The weka data mining software: an update, ACM SIGKDD Explor. Newsl. 11 (1) (2009) 10–18.
[31] R. Blagus, L. Lusa, Evaluation of smote for high-dimensional class-imbalanced microarray data, in: 2012 11th International Conference on Machine Learning and Applications (ICMLA), vol. 2, IEEE, 2012, pp. 89–94.
[32] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, A review of feature selection methods on synthetic data, Knowl. Inform. Syst. 34 (3) (2013) 483–519.
[33] J. Ross Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.
[34] I. Kononenko, Estimating attributes: analysis and extensions of relief, in: Machine Learning: ECML-94, Springer, 1994, pp. 171–182.
[35] L. Yu, H. Liu, Redundancy based feature selection for microarray data, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2004, pp. 737–742.
[36] M.A. Hall, Correlation-based Feature Selection for Machine Learning (PhD thesis), The University of Waikato, 1999.
[37] M. Dash, H. Liu, Consistency-based search in feature selection, Artif. Intell. 151 (1) (2003) 155–176.
[38] Z. Zhao, H. Liu, Searching for interacting features, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers Inc., 2007, pp. 1156–1161.
[39] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical recipes in C: the art of scientific programming, Section 10 (1992) 408–412.
[40] K. Kira, L.A. Rendell, A practical approach to feature selection, in: Proceedings of the Ninth International Workshop on Machine Learning, Morgan Kaufmann Publishers Inc., 1992, pp. 249–256.
[41] J.R. Quinlan, C4.5: Programs for Machine Learning, vol. 1, Morgan kaufmann, 1993.
[42] I. Rish, An empirical study of the naive Bayes classifier, in: IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, 2001, pp. 41–46.
[43] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1) (1991) 37–66.
[44] V.N. Vapnik, Statistical Learning Theory, Wiley, 1998.
[45] M.A. Hall, G. Holmes, Benchmarking attribute selection techniques for discrete class data mining, IEEE Trans. Knowl. Data Eng. 15 (6) (2003) 1437–1447.