Pattern Recognition 45 (2012) 531–539

Contents lists available at ScienceDirect

Pattern Recognition

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

An ensemble of filters and classifiers for 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 28 May 2010
Received in revised form
21 June 2011
Accepted 30 June 2011
Available online 14 July 2011

Keywords:
Feature selection
Ensemble methods for classification
Microarray data sets

ABSTRACT

In this paper a new framework for feature selection consisting of an ensemble of filters and classifiers is described. Five filters, based on different metrics, were employed. Each filter selects a different subset of features which is used to train and to test a specific classifier. The outputs of these five classifiers are combined by simple voting. In this study three well-known classifiers were employed for the classification task: C4.5, naive-Bayes and IB1. The rationale of the ensemble is to reduce the variability of the features selected by filters in different classification domains. Its adequacy was demonstrated by employing 10 microarray data sets.

1. Introduction

Nowadays, machine learning algorithms have to deal with large amounts of data. Usually, data contain thousands to tens of thousands of instances and each instance is represented by hundreds to many thousands of features. Specifically, in gene expression microarray data, 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 (often less than 100 patients) for training and testing, 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".

Therefore, microarray data classification is a difficult challenge for machine learning researchers due to its high number of features and the small sample sizes. Theoretically, having more genes should give more discriminating power. However, experimental evidence has shown that this is not always the case. Decision trees, such as C4.5 [1], exhibit a degradation in the performance when faced with many irrelevant features. Similarly, instance-based learners, such as IB1 [2], are also very susceptible to irrelevant features. It has been shown that the number of training instances needed to produce a predetermined level of performance for instance-based learning increases exponentially with the number of irrelevant features present [3]. On the other hand, algorithms such as naive Bayes are robust with respect to irrelevant features, degrading their performance very slowly when more irrelevant features are added. However, the performance of such algorithms deteriorate quickly by adding redundant features, even if they are relevant to the concept. Furthermore, the presence of many features not only affects the predictive performance, but also the runtime of the learning algorithms [4].

Several studies have shown that most genes measured in a DNA microarray experiment are not relevant for an accurate classification among different classes of the problem [5]. To avoid the problem of the "curse of dimensionality" [6], feature (gene) selection plays a crucial role in DNA microarray analysis. Another important reason to reduce dimensionality is to help biologists to identify the underlying mechanism that relates gene expression to diseases.

Feature selection is defined as the process of identifying and removing irrelevant and redundant features from the training data, so that the learning algorithm focuses only on those aspects of the training data useful for analysis and future prediction. This reduction in the input dimensionality implies, most of the time, an improvement in the performance [7]. There are three main models that deal with feature selection: filter methods, wrapper methods and embedded methods [7]. 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 with independence of any predictor. The embedded methods generally use machine learning models for classification, and then an optimal subset of features is built by the classifier algorithm. Although wrapper model tends to obtain better performances, it is very time consuming and has the risk of overfitting due to the reduced number of instances of microarray data and the small ratio between number of samples and number

* Corresponding author.
E-mail addresses: vbolon@udc.es (V. Bolón-Canedo), nsanchez@udc.es (N. Sánchez-Maroño), ciamparo@udc.es (A. Alonso-Betanzos).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.06.006

of features. Embedded methods may also suffer from overfitting. So, in those cases in which the number of features is very large, filter methods usually are the best option to obtain a reduced set of features. Thus, in this paper, and after a study involving filters and embedded methods, the former were chosen because they allow for reducing the dimensionality of the data without compromising the time and memory requirements of machine learning algorithms.

There exists a vast body of filters in the literature, based on distinct metrics. The proliferation of feature selection algorithms, however, has not brought about a general methodology that allows for intelligent selection from existing algorithms. In order to make a correct choice, a user not only needs to know the domain well, but also is expected to understand technical details of available algorithms [8]. Therefore, the more the algorithms available, the more challenging it is to choose a suitable one for a given application.

Each filter uses a different metric (entropy, probability distributions, information theory, etc.). Then, for a specific data set, employing one or another filter varies the selected subset of features and, consequently, the performance result obtained by a machine learning algorithm. Besides, the filter that achieves the best results in that specific data set may perform poorly with another one. So, there exists a high *variability* in the performance results and the choice of which filter should be used becomes a complicated issue. The aim of this work is to reduce this variability associated to feature selection methods and to obtain a method that could be applied over any data set regardless of its characteristics. An ensemble of filters is proposed in order to obtain good performance independently on the data set. The idea of this ensemble is to apply several filters based on different metrics and then joining the result obtained after training a classifier with the selected subset. In this manner, the user is released from the task of choosing an adequate filter for each scenario. Recently, high dimensionality data has become a trendy issue in machine learning research, so the proposed approach will be tested over 10 different microarray data sets, which represent a difficult challenge for the sake of classification. The obtained results are promising and the adequacy of the ensemble was demonstrated.

2. A new ensemble of filters and classifiers

2.1. The background

In recent years, ensemble approaches to classification have been the focus of much attention [9]. The idea builds on the assumption that combining the output of multiple experts is better than the output of any single expert. In a typical ensemble of classifiers, several classifiers are generated from a single classifier, the so-called base classifier, by changing the training set of the input features or the parameters of the classifier. Ensemble learning is illustrated in Fig. 1.

Bagging [10] and boosting [11] are the two most popular examples of ensemble learning. Bagging creates an ensemble by training individual classifiers on bootstrap samples of the training set. Each bootstrap sample is generated by randomly selecting, with replacement, n instances from the training set where n is the size of the training set. As a result of the sampling with replacement procedure, each classifier is trained on the average of 63.2% of the training instances. The prediction of each classifier is combined using simple voting. On the other hand, boosting takes a different resampling approach than bagging: sampling is proportional to an instance's weight. Bagging and boosting are two of the most well-known ensemble learning methods due to their theoretical performance guarantees and strong experimental results. However, new ensemble learning techniques on the feature subspace have been proposed by recent research as an attempt to improve the classification results. The *Random Subspace* [12] method is a simple random selection of feature subsets derived from the theory of stochastic discrimination. Optiz [13] describes an ensemble feature selection technique for neural networks called *Genetic Ensemble Feature Selection*. Another ensemble method for decision trees is called *Stochastic Attribute Selection Committees* [14] and finally, *Multiple Feature Subsets* [15] is a combining algorithm for nearest neighbor classifiers.

2.2. The rationale of the approach

In some of the examples mentioned above, the disturbances in the training set due to resampling cause diverse base classifiers to be built. In other cases, the technique employed was to use different features for each of the base classifiers. Usually, the ensembles found in the literature involving feature selection are based on the idea of applying several feature selection methods in order to distribute the whole set of features into the instances of the classifier [4]. It has to be noted that this method implies that all the features in the training set are exhaustively used.

Nevertheless, the purpose of the proposed ensemble is different. As stated before, one of the problems of choosing a filter is its variability of results over different data sets. That is, a filter can obtain excellent classification results in a given data set while performing poorly in another data set, even in the same domain, depending on the specific properties of the different data sets. Our goal is to achieve a method that reduces the variability of the features selected by the filters in the different classification domains. Therefore, this work will be based on the idea of combining several filters, employing different metrics and performing a feature reduction. Each filter selects a subset of features and this subset is used for training the classifier. There will be as many outputs as filters employed and the result of the filters and classifier will be combined using simple voting. For a particular instance, each classifier votes for a class and the class with the greatest number of votes is considered the output class. Notice that with the proposed approach, not all the features have to be

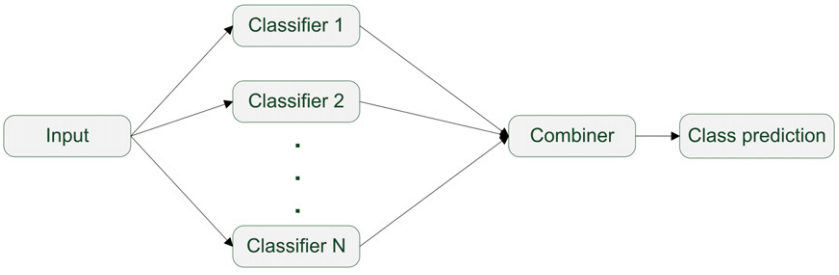

Fig. 1. An illustration of an ensemble of classifiers.

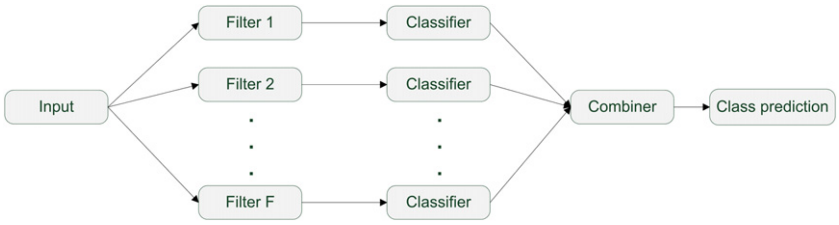

Fig. 2. An illustration of an ensemble of filters and classifiers. In each execution, F filters but only one classifier are used.

necessarily employed. Each filter selects a subset of features and the union of these subsets is not usually the whole set of features.

So, our proposal is an ensemble that obtains a classification prediction for every filter and then combines these predictions by simple voting, as can be seen in Fig. 2. Different induction algorithms may be chosen for the classification task, and the pseudo-code is shown in Algorithm 1.

Algorithm 1. The pseudo-code for the ensemble of filters and classifiers

1. $F :=$ number of filters
   (a) for each f from 1 to F
      (i) select attributes A using filter f
      (ii) build classifier C_f with the selected attributes A
      (iii) obtain prediction P_f from classifier
   (b) apply simple voting over predictions $P_1 \ldots P_f$
   (c) obtain prediction P

Besides the approach proposed in this ensemble, another different type of ensemble was tested. The alternative approach consisted of combining the subsets selected by each filter obtaining only one subset of features (result of the union of subsets) and then apply the classifier only once. This method had the advantage that a combiner method was not required for obtaining the class prediction. After trying this approach over 5 of the 10 data sets involved in this work, we observed that the accuracy results in a degradation. This fact can be twofold: the redundancy that is usually introduced by joining subsets of features selected by different filters and the employment of a single classifier instead of several ones. Therefore, this approach was discarded.

2.3. The process of selecting the methods for the ensemble

As was stated in Section 1, feature selection algorithms designed with different evaluation criteria broadly fall into three categories: the filter model, the embedded model and the wrapper model [7]. In DNA microarray data, the ratio between the number of samples and the number of features is very small, and this fact prevents the use of a wrapper model because it could not generalize adequately. Therefore, in a first stage, filters and embedded methods were chosen to perform a previous study, paving the way for its application to the ensemble.

As the goal is to choose methods based on different metrics, five filters and two embedded methods – all of them available in the Weka tool environment [16] – were tested over five synthetic data sets under different situations: increasing number of irrelevant features and the insertion of noise in the inputs, as well as the inclusion of correlated features. Both filter and embedded methods are subsequently described and it has to be noted that the first three provide a subset of features, whereas the last four (two filters and two embedded methods) provide features ordered according to their relevance (a ranking of features).

- *Correlation-based Feature Selection, CFS*: This is a simple filter algorithm that ranks feature subsets according to a correlation-based heuristic evaluation function [17]. The bias of the evaluation function is toward subsets that contain features that are highly correlated with the class and uncorrelated with each other. Irrelevant features should be ignored and redundant features should be screened out.
- *Consistency-based Filter*: The consistency-based filter [18] 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.
- *INTERACT*: The INTERACT algorithm [19] is based on symmetrical uncertainty (SU) [20]. Besides SU, INTERACT also includes the consistency contribution (c-contribution). The algorithm consists of ranking the features in descending order based on their SU values and then evaluating one by one starting from the end of the ranked feature list. Then, a feature is selected or not depending on a threshold, either established by the user or employing the default value provided. Theoretically, INTERACT can handle feature interaction.
- *Information Gain*: One of the most common attribute evaluation methods is called Information Gain [21]. This method provides an ordered ranking of all the features and then a threshold is required.
- *ReliefF*: ReliefF [22] is an extension of the original Relief algorithm [23] that adds the ability of dealing with multiclass problems and it is more robust and capable of dealing with incomplete and noisy data. The Relief family of methods are specially attractive because they may be applied in all situations, have low bias, include interaction among features and may capture local dependencies that other methods miss.
- *SVM-RFE*: SVM-RFE (Recursive Feature Elimination for Support Vector Machines) was introduced by Guyon in [24]. This embedded method performs feature selection by iteratively training an SVM classifier with the current set of features and removing the least important feature indicated by the SVM.
- *FS-P*: FS-P (Feature Selection – Perceptron) [25] is an embedded method that uses a simple linear perceptron as the classification model and ranks the importance of the features based on the weights returned by the perceptron.

In order to determine the effectiveness of each one of the feature selection methods described above at different situations, several widely used synthetic data sets were employed: the LED data set [26], the CorrAL data set [27] and the XOR-100 data set [28]. Table 1 summarizes the different problems that are covered by them. It is easy to see that it is difficult to deal with these data sets because they are complex: they include correlation with the

Table 1
Different problems tested over the synthetic data sets.

Problem	CorrAL	CorrAL-100	XOR-100	Led-25	Led-100
Correlation	✓	✓			
Noise				✓	✓
No. features ≫ No. samples		✓	✓		✓

Table 2
Average of success for every feature selection method tested.

Method	CorrAL	CorrAL-100	XOR-100	Led-25	Led-100	Average
CFS	50.00	94.00	46.00	71.50	71.33	66.57
Consistency	50.00	94.00	46.00	68.00	64.00	64.40
INTERACT	25.00	92.00	47.00	66.67	73.50	60.83
InfoGain	0.00	88.00	−1.00	66.33	70.00	44.67
ReliefF	50.00	88.00	95.00	78.17	82.50	78.73
SVM-RFE	50.00	59.00	−15.00	22.83	25.33	27.93
FS-P	0.00	43.00	−9.00	72.00	70.67	35.33

class, different levels of noise in the inputs and several irrelevant features. Besides, we employed data sets where the number of features is significantly higher than the number of samples, which involves an added difficulty for the correct selection of the relevant features and represents the same problem as when using microarray data.

Table 2 shows the averages of success for each feature selection method over each scenario and also an overall average for each method (last column). The percentage of success (% success) is defined as:

$$\%\ success = \left[\frac{R_s}{R_t} - \frac{I_s}{I_t}\right] \times 100,$$

where R_s is the number of relevant features selected, R_t is the total number of relevant features, I_s is the number of irrelevant features selected and I_t is the total number of irrelevant features.

As can be seen in Table 2, the two embedded methods (SVM-RFE and FS-P) achieve the poorest averages of success. As SVM-RFE achieved the worst result, we decided not to use it in the proposed ensemble. Focusing on the filters, although ReliefF obtained the best average, CFS, Consistency and INTERACT also showed a good performance. Information Gain obtained the poorest results of the filters methods, and similar to those obtained by FS-P. However, since Information Gain performs better than FS-P and bearing in mind the higher computational cost of the embedded methods, FS-P is discarded. Thus, all the five filters were selected to conform the proposed ensemble.

3. Experimental settings

In this work, an ensemble of filters and classifiers is designed. The suite of filters employed for this purpose contains CFS, Consistency-based filter, INTERACT, ReliefF and Information Gain, all of them described in the previous section. Because of using filters, the ensemble is independent of the classifier employed. However, for the sake of completeness, three different well-known classifiers were tested: C4.5 [1], naive Bayes [29] and IB1 [2].

The proposed ensemble was designed with the aim of dealing with the variability of the filters and to release the user from the task of choosing an adequate filter for each scenario. When handling extremely high-dimensional data, such as DNA microarray data, filtering has become indispensable. In the following subsections, the description of the validation procedure and an analysis of the stability of the different filters selected and, therefore the stability of the proposed ensemble, are presented.

3.1. The data sets and the validation procedure

The performance of the proposed ensemble will be tested over 10 well-known microarray data sets. These types of data sets are a difficult challenge for machine learning algorithms, as stated in Section 1, due to their high number of features and small sample sizes. Typical values are around 10 000 gene expressions and a hundred or fewer tissue samples. Furthermore, several studies [5] have shown that most genes measured in DNA microarray data are not relevant for an accurate classification and the large number of input features makes it difficult to achieve a high level of stability. So all these characteristics of DNA microarray data make it the perfect candidate to check the performance of the proposed ensemble.

The 10 microarray data sets employed are listed in Table 3. All data sets, except GCM and Lymphoma, are related with binary classification problems and are publicly available at Kent Ridge Biomedical Data Set Repository [30]. On the other hand, GCM and Lymphoma (two last rows) have 14 and 9 classes, respectively, and both of them can be freely downloaded from the Broad Institute Cancer Program Data Sets Repository [31]. In Table 3 one can see that the number of samples ranges from 60 to 253 and the number of features oscillates from 2000 to 24 481, which conforms an interesting suite of data sets to check the adequacy of the ensemble.

For obtaining the ensemble performance results, a 10-fold cross validation was executed over each data set following the recommendations in [32]. Therefore, although some of the data sets used are originally divided into training and tests sets, both were joined in a unique set in order to achieve a fair comparison.

In this ensemble, five different filters were involved. While three of them return a feature subset, the other two (ReliefF and Information Gain) are ranker methods, so it is necessary to establish a threshold in order to obtain a subset of features. Our initial experiments showed that for most of the data sets, the subset filters selected a number of features between 25 and 50. For the sake of fairness, the rankers were forced to select a number of features similar to the cardinality obtained by the other type of filters. Several experiments were carried out with 25 and 50 features. As performance did not improve using 50 features with respect to 25, we have decided to force these ranker methods to obtain subsets with 25 features.

3.2. The stability of the selected filters

After obtaining the results shown in Table 2, we have decided that the ensemble will be focused on filter methods. With the advent of high-dimensionality data sets in classification problems, a variety of feature selection methods have been

Table 3
Data set description. All of them are free to download in [30,31].

Data set	Features	Samples	Classes
Breast [33]	24481	97	2
CNS [34]	7129	60	2
Colon [35]	2000	62	2
DLBCL [36]	4026	47	2
Leukemia [5]	7129	72	2
Lung [37]	12533	181	2
Ovarian [38]	15154	253	2
Prostate [39]	12600	136	2
GCM [40]	16063	190	14
Lymphoma [36]	4026	96	9

developed to tackle them. The major challenge in feature selection methods is to extract a set of features, as small as possible, that accurately classifies the learning examples [41]. But a relatively neglected issue in the work on high-dimensionality problems is the stability of the feature selection methods used, which is defined as the sensitivity of a method to variations in the training set.

In this paper, we checked the stability of the different filters used in the proposed ensemble. For this purpose, we measured similarity between two subsets of features $\{s,s'\}$, using an adaptation of the Tanimoto distance between two sets proposed in [41]:

$$S(s,s') = 1 - \frac{|s|+|s'|-2|s \cap s'|}{|s|+|s'|-|s \cap s'|}.$$

Table 4 shows the stability of the 10 microarray data sets tested in this work in terms of average. For each data set and algorithm, the stability was measured comparing the subset selected in each fold of a 10-fold cross-validation. Then, an average of each one of these results was performed and is shown in Table 4. It has to be noted that according to Tanimoto measure, 0 means the minimum stability and 1 the maximum. The best result for each data set is emphasized in bold font.

As one can see in Table 4, the best stabilities are achieved by the ReliefF filter, which also obtained the best performance in the previous study (see Table 2). Therefore, this filter is expected to obtain good classification results. On the other hand, Information Gain obtained very good stability results, although this filter performed poorly in the previous study over synthetic data (see Table 2), but this result suggests that this can be a useful filter since its stability is very high. It has to be noted that both ReliefF and Information Gain are ranker methods. Interact, CFS and Consistency are the least stable, however they obtained very good results in average of success. Thus, this study reaffirms the choice of these five filters to comprise the proposed ensemble.

Considering the five filters selected, the stability of the ensemble formed by them is shown in the last row of Table 4. As was expected, the stability of the ensemble is close to the mean of the stabilities of the five filters, since the features selected by the ensemble are the union of the features selected by each filter. It has to be noted that the term stability is used for this study prior to the evaluation of the ensemble, aiming at choosing the best methods to form part of it, and therefore stability will not be taken into consideration when assessing the performance of the ensemble because a high stability does not imply a high accuracy in classification.

4. Experimental results

In this section we present the results obtained with the proposed ensemble after performing a 10-fold cross-validation. Following this methodology, the original sample set is randomly partitioned into 10 subsamples. One of these subsamples is retained as the validation data for testing the model, and the remaining nine subsamples are used as training data, selecting from there the features for the ensemble method.

For the sake of comparison, the results obtained by the ensemble are compared with the results achieved by the given classifier without feature selection (second row of Tables 5–7) and with the results obtained by each of the filters involved in the

Table 4
Stability of the filters and the ensemble over the 10 data sets tested.

Algorithm	Leuk.	Lymp.	GCM	CNS	DLBCL	Breast	Lung	Ovar.	Prost.	Colon	AVG
CFS	0.213	0.308	0.261	0.208	0.274	0.194	0.247	0.386	0.340	0.319	0.275
Cons	0.170	0.042	0.069	0.151	0.470	0.061	0.126	0.486	0.077	0.138	0.179
INT	0.246	0.281	0.160	0.182	0.232	0.178	0.221	0.262	0.207	0.264	0.223
InfoGain	0.654	0.349	0.342	0.252	0.488	0.212	**0.721**	**0.875**	0.322	0.529	0.474
ReliefF	**0.684**	**0.600**	**0.788**	**0.307**	**0.621**	**0.301**	0.605	0.688	**0.395**	**0.675**	**0.566**
Ensemble	0.396	0.351	0.324	0.266	0.380	0.229	0.287	0.476	0.363	0.511	0.358

Table 5
Tenfold cross-validation error (in %) obtained for C4.5 classifier.

Algorithm	Colon	Leuk.	Lymp.	GCM	CNS	DLBCL	Prostate	Lung	Ovarian	Breast	AVG
Ensemble	13.10	11.96	**18.78**	**40.00**	**36.67**	20.50	11.81	2.75	**1.20**	36.22	**19.30**
C4.5	23.81	23.04	23.11	44.21	45.00	22.50	19.78	6.08	4.34	39.00	25.09
CFS	19.29	16.25	29.11	48.42	40.00	27.00	22.97	6.08	2.80	38.33	25.03
Cons	16.43	14.82	40.89	60.53	41.67	**14.50**	16.92	7.22	1.57	39.78	25.43
INT	**10.95**	16.25	21.00	50.00	38.33	23.50	18.46	6.67	2.77	40.11	22.80
InfoGain	14.29	12.50	31.22	58.42	**36.67**	24.50	**9.40**	**1.11**	1.98	**33.33**	22.34
ReliefF	22.62	**11.25**	35.00	57.89	41.67	24.50	10.27	1.67	2.35	41.11	24.84

Table 6
Tenfold cross-validation error (in %) obtained for naive Bayes classifier.

Algorithm	Colon	Leuk.	Lymph.	GCM	CNS	DLBCL	Prostate	Lung	Ovarian	Breast	AVG
Ensemble	16.19	4.11	**19.78**	32.11	**30.00**	6.50	41.87	**0.00**	0.80	46.56	**19.79**
Naive Bayes	47.86	**1.25**	25.00	40.53	36.67	8.50	44.78	2.22	6.37	48.11	26.13
CFS	18.10	2.86	23.89	32.63	38.33	**4.00**	36.70	1.11	**0.40**	48.33	20.64
Cons	20.00	10.54	32.56	46.84	46.67	14.50	39.01	8.83	0.78	43.22	26.30
INT	19.29	4.11	22.00	**31.58**	36.67	6.00	**30.93**	1.11	**0.40**	51.56	20.37
InfoGain	22.86	4.11	22.78	54.21	41.67	8.50	42.80	**0.00**	2.35	43.33	24.26
ReliefF	**14.52**	4.29	24.67	57.89	33.33	5.00	40.27	3.33	3.95	**28.67**	21.59

Table 7
Tenfold cross-validation error (in %) obtained for IB1 classifier.

Algorithm	Colon	Leuk.	Lymph.	GCM	CNS	DLBCL	Prostate	Lung	Ovarian	Breast	AVG
Ensemble	**19.05**	5.54	5.33	**34.74**	**36.67**	**4.00**	**12.53**	1.11	**0.00**	**28.11**	**14.71**
IB1	26.62	9.46	7.22	44.21	51.67	24.00	20.11	5.00	5.54	39.78	23.36
CFS	19.29	5.54	8.44	40.53	50.00	6.50	21.04	**0.56**	**0.00**	36.22	18.81
Cons	26.43	9.46	29.56	58.95	41.67	18.50	20.11	7.22	0.40	38.33	25.06
INT	22.86	**4.11**	**4.33**	41.05	50.00	8.50	20.05	**0.56**	0.80	32.22	18.45
InfoGain	20.95	6.61	18.56	54.21	46.67	6.50	16.87	1.11	2.74	**28.11**	20.23
ReliefF	25.71	6.96	16.44	55.26	41.67	6.50	12.58	2.19	1.18	31.00	19.95

ensemble separately (rows 3–7). Columns correspond with the 10 different data sets and the last column is the average of the errors obtained by each method.

The behavior of the ensemble was tested with three different classifiers: C4.5 (Table 5), naive Bayes (Table 6) and IB1 (Table 7). The best result for each classifier and data set is emphasized in bold font.

Regarding Table 5 (where AVG stands for average), one can see that for C4.5 classifier, the ensemble achieved the best result for 4 out of the 10 data sets. When not, it obtains the second (see data sets Colon, Leukemia, DLBCL and Breast) or the third (see data sets Prostate and Lung) best result with an average difference with the best of 2.63. It has to be noted that there is no filter that always outperforms the results obtained by the ensemble, since the performances of the several filters are very different. The Information Gain filter seems to work adequately for 4 out of the 10 data sets, but it performs very poorly when dealing with multiclass data sets, i.e. Lymphoma and GCM. In terms of average, the ensemble clearly obtains the best result.

In Table 6 one can analyze the results obtained by naive Bayes classifier. In this case, the ensemble achieved the best performance in three out of the 10 data sets involved, but again none of the filters has demonstrated to be the best individually. It has to be noted that the Information Gain filter, that previously obtained good results (see Table 5) behaves poorer when using the naive Bayes classifier, except in the case of the Lung data set. For this classifier, INTERACT is the filter that exhibits better results, being the best method in 3 out of the 10 data sets. However, it does not surpass the average results of the ensemble method, that is again the one with the best results, although with a higher average difference with the best in those cases in which it is not the winner (5.26).

The results obtained with the IB1 classifier can be seen in Table 7. The individual filter that obtains the best results is again INTERACT, with the lower error rate in 3 out of the 10 data sets. However, the ensemble achieved the lower percentage of error in 7 out of the 10 data sets analyzed and, for the three remaining data sets, it gets the second best score with an average difference of approximately 1. In terms of average, the ensemble obtained the lowest percentage of error (outperforming in almost 4 points to the second best).

An important issue when applying feature selection is to check the reduction accomplished in the number of features. In this work, an ensemble of filters is proposed consisting of joining five filters based on different metrics. Then, each filter selects its own subset of features. Moreover, a 10-fold cross-validation was performed to evaluate the ensemble, obtaining 50 different subsets of features (5 filters $\times$ 10 folds). In order to calculate the number of features the ensemble uses, all these subsets have to be joined, and the cardinality of this union computed. Ovarian was the data set with the lowest number of features for classification (1.22%), whereas Lymphoma was the data set that uses the highest number (28.88%). Apart from Lymphoma, there is no data set that requires more than 7.5% of the initial set of features. It has to be noted that the ensemble uses more features than a single filter, although it still obtains a significant reduction in the number of features needed.

4.1. Study of other alternatives

In the previous section, the adequacy of the proposed ensemble formed by five filters was demonstrated, since their results outperformed the ones obtained by the five filters independently. However, an arising question could be if a better ensemble could be constructed by employing a smaller number of filters. In order to best answer this question one can look back on the experiments done in previous sections of this work aiming at selecting the appropriate methods for the ensemble. In Section 2.3, several widely used synthetic data sets were employed to test the effectiveness of different feature selection methods. According to these experiments, the two methods with the best performance were CFS and Consistency-based. On the other hand, in Section 3.2, the stability of filters to variations in the training set was checked. As a result of this investigation, ReliefF and Information Gain turned out to be the most stable filters. Therefore, and this is the point, two new ensembles are studied trying to find if the proposed ensemble consisting of five filters is the best option. The best one, called *Ensemble1*, is formed by CFS and Consistency-based (the two with clearly best performance), while the second one, called *Ensemble2* is formed by ReliefF and Information Gain (the two with the highest stability).

Tables 8–10 compare the results obtained by the proposed ensemble of five filters (first row) with *Ensemble1* and *Ensemble2*, for the classifiers C4.5, naive Bayes and IB1, respectively. In light of the results reported by these tables, the ensemble proposed in this research, formed by five filters, is the one which achieves the lowest errors on average for all three classifiers. Therefore, the adequacy of the filters which arrange the ensemble remains demonstrated.

4.2. Comparison with other authors

For the sake of completeness, the performances over the data sets Leukemia, Colon, Lymphoma and GCM are contrasted with those provided by Ruiz et al. [42]. The authors of the latter study performed a 10-fold cross-validation over those data sets in order to test the efficacy of their wrapper method, called BIRS (Best Incremental Ranked Subset). In addition, they also checked the performance of several filters. Due to the high dimensionality of the data, they limited their comparison to sequential forward techniques, such as SF (Sequential Forward), CFS and FOCUS besides the FCBF (Fast Correlation-Based Filter) algorithm. The classifiers used were C4.5, naive Bayes and IB1 and the comparison of the results obtained by the proposed ensemble and their

Table 8
Tenfold cross-validation error (in %) obtained for C4.5 classifier.

Algorithm	Colon	Leuk.	Lymp.	GCM	CNS	DLBCL	Prostate	Lung	Ovarian	Breast	AVG
Ensemble	13.10	**11.96**	**18.78**	**40.00**	**36.67**	20.50	11.81	**2.75**	**1.20**	**36.22**	**19.30**
Ensemble1	15.95	15.00	31.89	51.05	41.67	**12.00**	**9.51**	3.86	1.98	43.33	22.62
Ensemble2	**12.86**	16.25	25.11	57.37	51.67	38.50	13.30	5.56	3.57	37.44	26.16

Table 9
Tenfold cross-validation error (in %) for naive Bayes classifier.

Algorithm	Colon	Leuk.	Lymph.	GCM	CNS	DLBCL	Prostate	Lung	Ovarian	Breast	AVG
Ensemble	**16.19**	4.11	**19.78**	**32.11**	**30.00**	6.50	41.87	**0.00**	**0.80**	46.56	**19.79**
Ensemble1	17.86	**3.93**	25.89	47.89	48.33	**6.00**	**35.82**	1.64	1.98	**31.00**	22.03
Ensemble2	18.81	5.18	26.22	53.68	40.00	8.50	39.95	2.78	2.37	33.22	23.07

Table 10
Tenfold cross-validation error (in %) obtained for IB1 classifier.

Algorithm	Colon	Leuk.	Lymph.	GCM	CNS	DLBCL	Prostate	Lung	Ovarian	Breast	AVG
Ensemble	**19.05**	5.54	**5.33**	**34.74**	**36.67**	**4.00**	12.53	1.11	**0.00**	**28.11**	**14.71**
Ensemble1	25.95	**5.36**	12.33	46.84	41.67	12.00	16.26	1.64	0.80	31.11	19.40
Ensemble2	23.57	8.04	11.56	54.74	46.67	6.50	**11.10**	**0.56**	1.60	35.22	19.96

Table 11
Comparison with Ruiz et al. [42]. Tenfold cross-validation error (in %) obtained for C4.5 classifier.

Algorithm	Colon	Leukemia	Lymphoma	GCM	AVG
Ensemble	13.10	11.96	**18.78**	**40.00**	**20.96**
BIRS	16.19	11.43	20.00	46.84	23.62
SF	19.29	12.68	27.00	N/A	N/A
CFS	13.10	15.18	20.78	N/A	N/A
FOCUS	20.95	**11.07**	37.56	50.53	30.03
FCBF	**11.67**	16.79	21.78	47.37	24.40

Table 12
Comparison with Ruiz et al. [42]. Tenfold cross-validation error (in %) obtained for naive Bayes classifier.

Algorithm	Colon	Leukemia	Lymphoma	GCM	AVG
Ensemble	16.19	**4.11**	19.78	32.11	18.05
BIRS	**14.52**	6.96	17.86	32.63	**17.99**
SF	15.95	12.68	**16.44**	N/A	N/A
CFS	17.38	8.57	24.89	N/A	N/A
FOCUS	22.86	15.18	29.93	43.16	27.78
FCBF	22.38	**4.11**	21.78	**31.05**	19.83

Table 13
Comparison with Ruiz et al. [42]. Tenfold cross-validation error (in %) obtained for IB1 classifier.

Algorithm	Colon	Leukemia	Lymphoma	GCM	AVG
Ensemble	**19.05**	**5.54**	**5.33**	**34.74**	**16.17**
BIRS	20.24	6.96	14.44	41.05	20.67
SF	33.33	11.07	19.89	N/A	N/A
CFS	19.29	9.82	7.22	N/A	N/A
FOCUS	30.71	18.04	38.78	53.16	35.17
FCBF	19.29	**5.54**	8.11	38.95	17.97

methods are shown in Tables 11–13. It has to be noted that INTERACT and FCBF are algorithms consisting of two steps, the first one is the same for both methods, although they differ in the second one. For that reason, they will exhibit a similar behavior.

Regarding the results for the three classifiers evaluated, one can see that our proposed ensemble achieved the best average error for two of the classifiers (C4.5 and IB1). For the naive Bayes classifier, the result achieved by the ensemble in terms of average is very close to the one obtained by BIRS.

Therefore, the proposed ensemble achieves better or at least equal performance results as the BIRS wrapper method.

5. Conclusions

In this paper, a new ensemble of filters and classifiers is proposed. The goal of this ensemble is to reduce the variability of the results obtained by feature selection methods over different data sets. The method consists of combining filters and classifiers obtaining a classification prediction for each of them and deciding a final result by simple voting. Different induction algorithms may be used for the classification task, and in this work three well-known classifiers were chosen: C4.5, naive Bayes and IB1.

To choose the filters that are part of the ensemble, two previous studies were carried out. The first one, which involved synthetic data sets, helped us to know if the feature selection methods tested were able to select the – already known – relevant features and discard the irrelevant ones in complex scenarios. On the other hand, the second study assessed the

stability of the filters tested, which is defined as the sensitivity of a method to variations in the training set. In light of the results obtained from these studies, we decided to use five filters based on different metrics: CFS, Consistency-based, INTERACT, Information Gain and ReliefF.

In order to test the adequacy of the proposed ensemble, a challenging scenario was chosen: DNA microarray data. These are extremely high-dimensional data sets, because of their high number of input features and small sample size. Therefore, filtering becomes indispensable. In this study 10 different microarray data sets were chosen. The results obtained by the ensemble over the 10 data sets achieved the lowest average of error for every one of the classifiers tested, showing the adequacy of the ensemble. This fact proves that, although in some specific cases there is a filter that performs better than the ensemble, there is not a better filter in general, and the ensemble seems to be the most reliable alternative when a feature selection process has to be carried out. Most remarkable are the results obtained by the IB1 classifier, since it achieved the best error results for 7 out of the 10 data sets studied.

The results achieved by our proposed method were also compared with those obtained by other authors, over four DNA microarray data sets. The ensemble achieved the best average error for two of the classifiers (C4.5 and IB1). For the naive Bayes classifier, the result achieved by the ensemble in terms of average is very close to the one obtained by BIRS (with a difference of 0.06), which is a wrapper method with the disadvantage of its higher computational cost.

This ensemble was proposed as a framework to feature selection in order to be able to reduce the dimensionality of the data and achieve good classification performance in diverse scenarios, without limiting its use to microarray data sets. So, as future work, a broader study including other high dimensionality domains, in order to check the generalization of the ensemble will be carried out.

Acknowledgments

This work was supported in part by the Spanish *Ministerio de Ciencia e Innovación*, grant code TIN2009-10748, and by the Xunta de Galicia, grant codes 08TIC012105PR and 2007/134, all of them partially supported by FEDER funds.

References

[1] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
[2] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Journal of Machine Learning 6 (1) (1991) 37–66.
[3] P. Langley, W. Iba, Average-case analysis of a nearest neighbor algorithm, Proceedings of International Joint Conference on Artificial Intelligence, vol. 13, 1993, pp. 889–894.
[4] N. Pradhananga, Effective linear-time feature selection, Master Thesis, University of Waikato, 2007.
[5] T.R. Golub, D. K Stomin, P. Tamayo, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Journal of Science 286 (5439) (1999) 531–537.
[6] A. Jain, D. Zongker, Feature selection: evaluation, application, and small sample performance, Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (2) (1997) 153–158.
[7] I. Guyon, S. Gunn, M. Nikravesh, L. Zadeh, Feature Extraction. Foundations and Applications, Springer, 2006.
[8] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, Journal of IEEE Transactions on Knowledge and Data Engineering (2005) 491–502.
[9] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[10] L. Breiman, Bagging predictors, Journal of Machine Learning 24 (2) (1996) 123–140.
[11] R.E. Schapire, The strength of weak learnability, Journal of Machine Learning 5 (2) (1990) 197–227.
[12] T.K. Ho, The random subspace method for constructing decision forests, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (8) (1998) 832–844.
[13] D.W. Optiz, Feature selection for ensembles, in: Proceedings of the National Conference on Artificial Intelligence, 1999, pp. 379–384.
[14] Z. Zheng, G.I. Webb, Stochastic Attribute Selection Committees, in: Lecture Notes in Computer Science, 1998, pp. 321–332.
[15] S.D. Bay, Combining nearest neighbor classifiers through multiple feature subsets, in: Proceedings of the Fifteenth International Conference on Machine Learning, 1998, pp. 37–45.
[16] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second ed., Morgan Kaufmann, San Francisco, 2005. 〈http://www.cs.waikato.ac.nz/ml/weka/〉 (Last access: June 2011).
[17] M.A. Hall, Correlation-based feature selection for machine learning, Ph.D. Thesis, University of Waikato, Hamilton, New Zealand, 1999.
[18] M. Dash, H. Liu, Consistency-based search in feature selection, Artificial Intelligence Journal 151 (1–2) (2003) 155–176.
[19] Z. Zhao, H. Liu, Searching for interacting features, in: Proceedings of International Joint Conference on Artificial Intelligence, IJCAI, 1991, pp. 1156–1167.
[20] W.H. Press, B. P Flannery, S. A Teukolsky, W.T. Vetterling, Numerical Recipes in C, Cambridge University Press, Cambridge, 1988.
[21] M.A. Hall, L.A. Smith, Practical feature subset selection for machine learning, Journal of Computer Science 98 (1998) 4–6.
[22] I. Kononenko, Estimating attributes: analysis and extensions of RELIEF, in: European Conference on Machine Learning, 1994, pp. 171–182.
[23] K. Kira, L. Rendell, A practical approach to feature selection, in: Proceedings of the Ninth International Workshop on Machine learning, 1992, pp. 249–256.
[24] I. Guyon, J. Weston, S.M.D. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Journal of Machine Learning 46 (1–3) (2002) 389–422.
[25] M. Mejía-Lavalle, E. Sucar, G. Arroyo, Feature selection with a perceptron neural net, in: Proceedings of the International Workshop on Feature Selection for Data Mining, 2006, pp. 131–135.
[26] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Wadsworth International Group, 1984.
[27] G.H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: Proceedings of the 11th International Conference on Machine Learning, 1994, pp. 121–129.
[28] G. Kim, Y. Kim, H. Lim, H. Kim, An MLP-based feature subset selection for HIV-1 protease cleavage site analysis, Journal of Artificial Intelligence in Medicine 48 (2010) 83–89.
[29] I. Rish, An empirical study of the naive Bayes classifier, in: Proceedings of IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence, 2001, pp. 41–46.
[30] K. Ridge, Kent Ridge Bio-Medical Dataset 〈http://datam.i2r.a-star.edu.sg/datasets/krbd〉, 2009 (accessed: June 2011).
[31] Broad Institute. Cancer Program Data Sets 〈http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi〉 (accessed: June 2011).
[32] C. Ambroise, G.J. McLachlan, Selection bias in gene extraction on the basis of microarray gene-expression data, Proceedings of the National Academy of Sciences 99 (10) (2002) 6562–6566.
[33] L.J. Van't Veer, H. Dai, M.J. Van de Vijver, et al., Gene expression profiling predicts clinical outcome of breast cancer, Journal of Nature 415 (6871) (2002) 530–536.
[34] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, et al., Prediction of central nervous system embryonal tumour outcome based on gene expression, Journal of Nature 415 (6870) (2002) 436–442.
[35] U. Alon, N. Barkai, D.A. Notterman, K. Gish, et al., 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 96 (12) (1999) 6745–6750.
[36] A.A. Alizadeh, M.B. Elisen, R.E. Davis, et al., Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, Journal of Nature 403 (6769) (2000) 503–511.
[37] G.J. Gordon, R.V. Jenson, L.L. Hsiao, et al., Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma, Journal of Cancer Research 62 (17) (2002) 4963–4967.
[38] E.F. Petricoin, A.M. Ardekani, B.A. Hitt, et al., Use of proteomic patterns in serum to identify ovarian cancer, Journal of the Lancet 359 (9306) (2002) 572–577.
[39] D. Singh, P.G. Febbo, K. Ross, D.G. Jackson, et al., Gene expression correlates of clinical prostate cancer behavior, Journal of Cancer Cell 1 (2) (2002) 203–209.
[40] S. Ramaswamy, P. Tamayo, R. Rifkin, et al., Multiclass cancer diagnosis using tumor gene expression signatures, Proceedings of the National Academy of Sciences 98 (26) (2001) 15149–15154.
[41] A. Kalousis, J. Prados, M. Hilario, Stability of feature selection algorithms: a study on high-dimensional spaces, Journal of Knowledge and Information Systems 12 (1) (2007) 95–116.
[42] R. Ruiz, J.C. Riquelme, J.S. Aguilar-Ruiz, Incremental wrapper-based gene selection from microarray data for cancer classification, Journal of Pattern Recognition 39 (12) (2006) 2383–2392.

Verónica Bolón-Canedo received the B.S. degree in Computer Science in 2008 from the University of A Coruña, Spain. She is currently a Ph.D. student in the Department of Computer Science of the same university. Her research interests include machine learning and feature selection, which are expected to be her thesis research area.

Noelia Sánchez-Maroño received a Ph.D. degree for work in the area of functional and neuronal networks in 2005 at the University of A Coruña. She is currently teaching at the Department of Computer Science in the same university. Her research interests include intelligent multi-agent systems, functional and artificial neural networks and feature selection.

Amparo Alonso-Betanzos received the Ph.D. degree for work in the area of medical expert systems in 1988 at the University of Santiago de Compostela. Later, she was a postdoctoral fellow in the Medical College of Georgia, Augusta. She is currently a Full Professor in the Department of Computer Science, University of A Coruña. Her main current areas are intelligent systems, intelligent multi-agent systems, optimization methods and neural and functional networks.