

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/257482053>

A review of feature selection methods on synthetic data

Article in Knowledge and Information Systems · March 2012

DOI: 10.1007/s10115-012-0487-8

CITATIONS

403

READS

5,101

3 authors:



[Verónica Bolón-Canedo](#)

University of A Coruña

112 PUBLICATIONS 2,735 CITATIONS

[SEE PROFILE](#)



[Noelia Sánchez-Marroño](#)

University of A Coruña

84 PUBLICATIONS 2,209 CITATIONS

[SEE PROFILE](#)



[Amparo Alonso-Betanzos](#)

University of A Coruña

252 PUBLICATIONS 4,331 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



LOCAW (Low-Carbon at Work: Modelling agents and organisations to achieve transition to a low-carbon Europe [View project](#)



Scalable machine learning algorithms: Beyond classification and regression [View project](#)

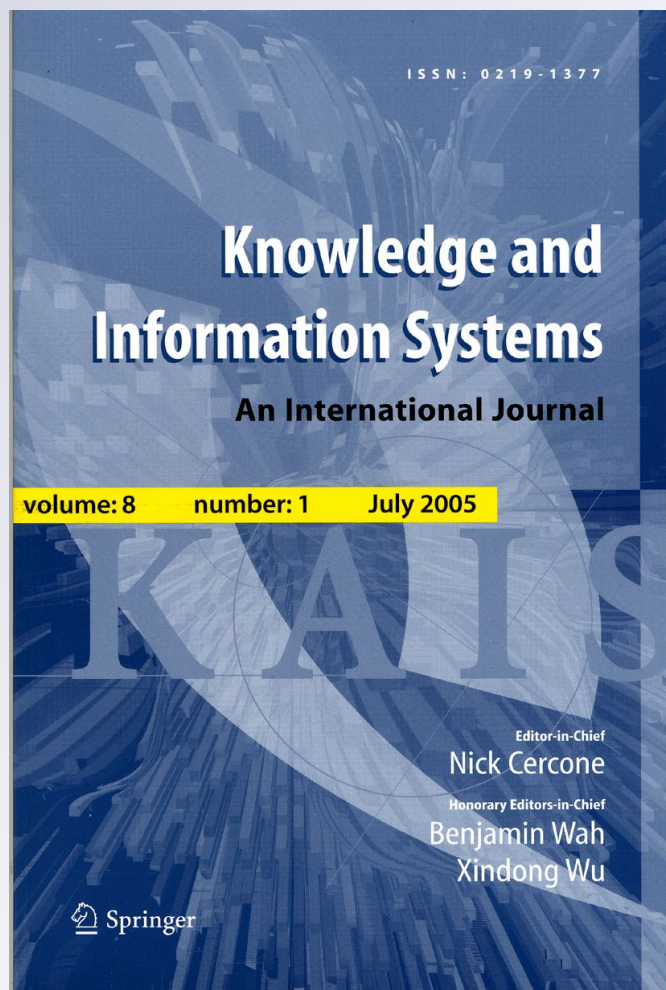
A review of feature selection methods on synthetic data

**Verónica Bolón-Canedo, Noelia
Sánchez-Marono & Amparo Alonso-
Betanzos**

Knowledge and Information Systems
An International Journal

ISSN 0219-1377

Knowl Inf Syst
DOI 10.1007/s10115-012-0487-8



Your article is protected by copyright and all rights are held exclusively by Springer-Verlag London Limited. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your work, please use the accepted author's version for posting to your own website or your institution's repository. You may further deposit the accepted author's version on a funder's repository at a funder's request, provided it is not made publicly available until 12 months after publication.

A review of feature selection methods on synthetic data

Verónica Bolón-Canedo · Noelia Sánchez-Marroño ·
Amparo Alonso-Betanzos

Received: 31 May 2011 / Revised: 2 November 2011 / Accepted: 5 March 2012
© Springer-Verlag London Limited 2012

Abstract With the advent of high dimensionality, adequate identification of relevant features of the data has become indispensable in real-world scenarios. In this context, the importance of feature selection is beyond doubt and different methods have been developed. However, with such a vast body of algorithms available, choosing the adequate feature selection method is not an easy-to-solve question and it is necessary to check their effectiveness on different situations. Nevertheless, the assessment of relevant features is difficult in real datasets and so an interesting option is to use artificial data. In this paper, several synthetic datasets are employed for this purpose, aiming at reviewing the performance of feature selection methods in the presence of a crescent number or irrelevant features, noise in the data, redundancy and interaction between attributes, as well as a small ratio between number of samples and number of features. Seven filters, two embedded methods, and two wrappers are applied over eleven synthetic datasets, tested by four classifiers, so as to be able to choose a robust method, paving the way for its application to real datasets.

Keywords Feature selection · Filters · Embedded methods · Wrappers · Synthetic datasets

1 Introduction

In the past few years, several datasets with high dimensionality have become publicly available on the Internet. This fact has brought an interesting challenge to the research community, since for the machine learning methods it is difficult to deal with a high number of input features. To confront the problem of the high number of input features, feature selection algorithms have become indispensable components of the learning process [1]. Feature selection is the process of detecting the relevant features and discarding the irrelevant ones. A correct selection of the features can lead to an improvement of the inductive learner, either in terms of learning speed, generalization capacity or simplicity of the induced model. Moreover, there

V. Bolón-Canedo (✉) · N. Sánchez-Marroño · A. Alonso-Betanzos
Department of Computer Science, University of A Coruña, A Coruña, Spain
e-mail: veronica.bolon@udc.es

are some other benefits associated with a smaller number of features: a reduced measurement cost and hopefully a better understanding of the domain.

There are several situations that can hinder the process of feature selection, such as the presence of irrelevant and redundant features, noise in the data or interaction between attributes. In the presence of hundreds or thousands of features, such as DNA microarray analysis, researchers notice [2,3] that is common that a large number of features is not informative because they are either irrelevant or redundant with respect to the class concept. Moreover, when the number of features is high but the number of samples is small, machine learning gets particularly difficult, since the search space will be sparsely populated and the model will not be able to distinguish correctly the relevant data and the noise [4].

There exist two major approaches in feature selection: *individual evaluation* and *subset evaluation*. Individual evaluation is also known as feature ranking [5] 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. Besides this classification, feature selection methods can also be divided into three models: *filters*, *wrappers* and *embedded* methods [6]. With such a vast body of feature selection methods, the need arises to find out some criteria that enable users to adequately decide which algorithm to use (or not) in certain situations.

This work reviews several feature selection methods in the literature and checks their performance in an artificial controlled experimental scenario, contrasting the ability of the algorithms to select the relevant features and to discard the irrelevant ones without permitting noise or redundancy to obstruct this process. A scoring measure will be introduced to compute the degree of matching between the output given by the algorithm and the known optimal solution, as well as the classification accuracy. Finally, real experiments are presented in order to check if the conclusions extracted from this theoretical study can be extrapolated to real scenarios.

2 State of the art

Feature selection (FS), since it is an important activity in data preprocessing, has been widely studied in the past years by the machine learning researchers. This technique has found success in many different real-world applications like DNA microarray analysis [7], intrusion detection [8,9], text categorization [10,11] or information retrieval [12], including image retrieval [13] or music information retrieval [14].

There exist numerous papers and books proving the benefits of the feature selection process [5,15,77]. However, most researchers agree that there is not a so-called “best method” and their efforts are focused on finding a good method for a specific problem setting. Therefore, new feature selection methods are constantly appearing using different strategies: a) combining several feature selection methods, which could be done by using algorithms from the same approach, such as two filters [16], or coordinating algorithms from two different approaches, usually filters and wrappers [17–19]; b) combining FS approaches with other techniques, such as feature extraction [20] or tree ensembles [21]; c) reinterpreting existing algorithms [22], sometimes to adapt them to specific problems [23]; d) creating new methods to deal with still unresolved situations [24,25]; and e) using an ensemble of feature selection techniques to ensure a better behavior [26,27].

Bearing in mind the large amount of FS methods available, it is easy to note that carrying out a comparative study is an arduous task. Another problem is to test the effectiveness of these FS methods when real datasets are employed, usually without knowing the relevant features.

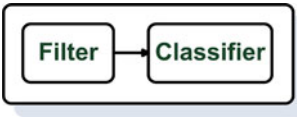
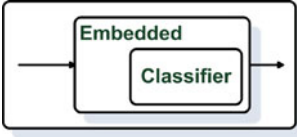
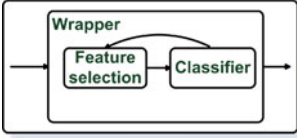
In these cases the performance of the FS methods clearly relies on the performance of the learning method used afterward and it can vary notably from one method to another. Moreover, performance can be measured using many different metrics such as computer resources (memory and time), accuracy, ratio of features selected, etc. Besides, datasets may include a great number of challenges: multiple class output, noisy data, huge number of irrelevant features, redundant or repeated features, ratio number of samples/number of features very close to zero and so on. It can be noticed that a comparative study tackling all these considerations could be unapproachable, and therefore, most of the interesting comparative studies are focused on the problem to be solved. So, for example, the work in [10] presents an empirical comparison of twelve feature selection methods evaluated on a benchmark of 229 text classification problem instances; the comparative study in [28] is used for the detection of breast cancers in mammograms; recent works analyze the application of FS in growing research areas such as Educational Data Mining (EDM) [29]. Other comparative studies are devoted to a specific approach such as in [30] where an experimental study of eight typical filter mutual information-based feature selection algorithms on thirty-three datasets is presented; or in [31] that evaluates the capability of the survival ReliefF algorithm (sReliefF) and of a tuned sReliefF approach to properly select the causative pair of attributes. Similarly, there are works examining different FS methods to obtain good performance results using a specific classifier (naive Bayes in [32], C4.5 in [33] or the theoretical review for SVM in [34]). Related to datasets challenges, there are several works trying to face the problem of high dimensionality, in both dimensions (samples and features) or in one of them, i.e., a high number of features versus low number of samples; most of these studies also tackle with the multiple class problems [24,35–38]. Also, the majority of current real datasets (microarray, text retrieval, etc.) also present noisy data; however, no specific FS comparative studies dealing with this complex problem were found in the literature, although some interesting works have been proposed, see for example [16,39,40]. Focusing on nonlinear methods is worth to mention the study of Guyon et al. [41]. Finally, from a theoretical perspective, in [42], a survey of feature selection methods was presented, providing some guidelines in selecting feature selection algorithms, paving the way to build an integrated system for intelligent feature selection.

More experimental work on feature selection algorithms for comparative purposes can be found in [43–47], some of which were performed over artificially generated data, like the widely used *Parity*, *LED* or *Monks* problems [48]. Several authors choose to use artificial data since the desired output is known, therefore a feature selection algorithm can be evaluated with independence of the classifier used. Although the final goal of a feature selection method is to test its effectiveness over a real dataset, the first step should be on synthetic data. The reason for this is twofold [49]:

1. Controlled experiments can be developed by systematically varying chosen experimental conditions, like adding more irrelevant features or noise in the input. This fact facilitates to draw more useful conclusions and to test the strengths and weaknesses of the existing algorithms.
2. The main advantage of artificial scenarios is the knowledge of the set of optimal features that must be selected; thus, the degree of closeness to any of these solutions can be assessed in a confident way.

In this work, several feature selection techniques will be tested over 11 synthetic datasets covering a large suite of problems (nonlinearity of the data, noise in the inputs and in the target, increasing number of irrelevant and redundant features, etc.). Although works studying some of these problems [50,51] can be found, up to the authors' knowledge a complete

Table 1 Feature selection techniques

Method	Advantages	Disadvantages	Examples
Filter 	Independence of the classifier Lower computational cost than wrappers Fast Good generalization ability	No interaction with the classifier	Consistency-based CFS INTERACT ReliefF \mathcal{M}_d Information Gain mRMR
Embedded 	Interaction with the classifier Lower computational cost than wrappers Captures feature dependencies	Classifier-dependent selection	FS-Perceptron SVM-RFE
Wrapper 	Interaction with the classifier Captures feature dependencies	Computationally expensive Risk of overfitting Classifier-dependent selection	Wrapper-C4.5 Wrapper SVM

study, such as the one described inhere, has not been carried out. Besides, a very interesting problem, since it is very probable in very datasets, such as the alteration of the input variables, has not been addressed elsewhere.

3 Feature selection techniques

With regard to the relationship between a feature selection algorithm and the inductive learning method used to infer a model, three major approaches can be distinguished:

- *Filters*, which rely on the general characteristics of training data and carry out the feature selection process as a pre-processing step with independence of the induction algorithm.
- *Wrappers*, which involve optimizing a predictor as a part of the selection process.
- *Embedded methods*, which perform feature selection in the process of training and are usually specific to given learning machines.

Table 1 provides a summary of the characteristics of the three feature selection methods, indicating the most prominent advantages and disadvantages, as well as some examples of each technique that will be further explained. Within filters, one can distinguish between *univariate* and *multivariate* methods. Univariate methods (such as InfoGain) are fast and scalable, but ignore feature dependencies. On the other hand, multivariate filters (such as CFS, INTERACT, etc.) model feature dependencies, but at the cost of being slower and less scalable than univariate techniques.

Besides this classification, feature selection methods can also be divided according to two approaches: *individual evaluation* and *subset evaluation* [3]. 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 redundant features 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 caused by searching through feature subsets required in the subset generation step, and thus, both approaches will be studied in this research.

The feature selection methods included in this work are subsequently described according to how they combine the feature selection search with the construction of the classification model: *filter* methods, *wrapper* methods and *embedded* methods (see Table 1). All of them are available in the Weka tool environment [52] or implemented in Matlab [53]. These feature selection methods belong to different families of techniques and conform an heterogeneous suite of methods to carry out a broad and complete study.

3.1 Filter methods

- **Correlation-based Feature Selection (CFS)** is a simple multivariate filter algorithm that ranks feature subsets according to a correlation-based heuristic evaluation function [54]. 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 because they will have low correlation with the class. Redundant features should be screened out as they will be highly correlated with one or more of the remaining features. The acceptance of a feature will depend on the extent to which it predicts classes in areas of the instance space not already predicted by other features
- The **Consistency-based Filter** [55] 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 **INTERACT** algorithm [56] is a subset filter based on symmetrical uncertainty (SU) and the consistency contribution, which is an indicator about how significantly the elimination of a 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 the consistency 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.
- **Information Gain** [57] is one of the most common attribute evaluation methods. This univariate filter provides an ordered ranking of all the features, and then a threshold is required. In this work the threshold will be set up selecting the features which obtain a positive information gain value.
- **ReliefF** [58] is an extension of the original Relief algorithm [59]. The original Relief works by randomly sampling an instance from the data and then locating its nearest neighbor from the same and opposite class. 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 an 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.

- The **mRMR** (minimum Redundancy Maximum Relevance) method [60] selects features that have the highest relevance with the target class and are also minimally redundant, i.e., selects features that are maximally dissimilar to each other. Both optimization criteria (Maximum Relevance and Minimum Redundancy) are based on mutual information.
- The \mathcal{M}_d filter [61] is an extension of mRMR which uses a measure of monotone dependence (instead of mutual information) to assess relevance and irrelevance. One of its contributions is the inclusion of a free parameter (λ) that controls the relative emphasis given on relevance and redundancy. In this work, two values of lambda will be tested: 0 and 1. When λ is equal to zero, the effect of the redundancy disappears and the measure is based only on maximizing the relevance. On the other hand, when λ is equal to one, it is more important to minimize the redundancy among variables. These two values of λ were chosen because we are interested in checking the performance of the method when the effect of the redundancy disappears. Also, the authors [61] state that $\lambda = 1$ performs better than other λ values.

3.2 Embedded methods

- **SVM-RFE** (Recursive Feature Elimination for Support Vector Machines) was introduced by Guyon in [62]. This embedded method performs feature selection by iteratively training a SVM classifier with the current set of features and removing the least important feature indicated by the SVM. Two versions of this methods will be tested: the original one, using a linear kernel and an extension using a nonlinear kernel in order to solve more complex problems [63].
- **FS-P** (Feature Selection—Perceptron) [64] is an embedded method based on a perceptron. A perceptron is a type of artificial neural network that can be seen as the simplest kind of feedforward neural network: a linear classifier. The basic idea of this method consists on training a perceptron in the context of supervised learning. The interconnection weights are used as indicators of which features could be the most relevant and provide a ranking.

3.3 Wrapper methods

- **WrapperSubsetEval** [52] evaluates attribute sets by using a learning scheme. Cross-validation is used to estimate the accuracy of the learning scheme for a set of attributes. The algorithm starts with the empty set of attributes and searches forward, adding attributes until performance does not improve further. In this work, two well-known learning schemes will be used: SVM and C4.5.

4 Artificial datasets

As was stated in Sect. 2, the first step to test the effectiveness of a feature selection method should be on synthetic data, since the knowledge of the optimal features and the chance to modify the experimental conditions allows to draw more useful conclusions.

The datasets chosen for this study try to cover different problems: increasing number of irrelevant features, redundancy, noise in the output, alteration of the inputs, nonlinearity of

Table 2 Summary of the synthetic datasets used. “Corr.” stands for “Correlation”.

Dataset	No. of features	No. of samples	Relevant features	Corr.	Noise	Non linear	No. feat > No. samples	Baseline accuracy
Corral	6	32	1–4	✓				56.25%
Corral-100	99	32	1–4	✓			✓	56.25%
XOR-100	99	50	1,2			✓	✓	52.00%
Parity3+3	12	64	1–3			✓		50.00%
Led-25	24	50	1–7		✓			16.00%
Led-100	99	50	1–7		✓		✓	16.00%
Monk3	6	122	2,4,5		✓			50.82%
SD1*	4020	75	G_1, G_2				✓	33.33%
SD2*	4040	75	$G_1 - G_4$				✓	33.33%
SD3*	4060	75	$G_1 - G_6$				✓	33.33%
Madelon	500	2400	1–5		✓	✓		50.13%

* G_i means that the feature selection method must select only one feature within the i -th group of features.

the data, etc. These factors complicate the task of the feature selection methods, which are very affected by them as it will be shown afterward. Besides, some of the datasets have a significantly higher number of features than samples, which implies an added difficulty for the correct selection of the relevant features.

The synthetic datasets employed are subsequently described, and Table 2 shows a summary of the different problems covered by them, as well as the number of features and samples and the relevant attributes which should be selected by the feature selection methods. Besides, the baseline accuracy is shown in last column, which indicates the minimum achievable accuracy when all samples are assigned to the majority class.

4.1 CorrAL

The CorrAL dataset [72] has six binary features (i.e., $f_1, f_2, f_3, f_4, f_5, f_6$), and its class value is $(f_1 \wedge f_2) \vee (f_3 \wedge f_4)$. Feature f_5 is irrelevant and f_6 is correlated to the class label by 75 %.

CorrAL-100 [73] was constructed by adding 93 irrelevant binary features to the previous CorrAL dataset. The data for the added features were generated randomly. Both datasets (CorrAL and CorrAL-100) have 32 samples that are formed by considering all possible values of the four relevant features and the correlated one (2^5). The correct behavior for a given feature selection method is to select the four relevant features and to discard the irrelevant and correlated ones. The correlated feature is redundant if the four relevant features are selected and, besides, it is correlated to the class label by 75 %, so if one applies a classifier after the feature selection process, a 25 % of error will be obtained.

4.2 XOR-100

XOR-100 [73] has 2 relevant binary features and 97 irrelevant binary features (randomly generated). The class attribute takes binary values and the dataset consists of 50 samples. Features f_1 and f_2 are correlated with the class value with XOR operation (i.e., class equals $f_1 \oplus f_2$). This is a hard dataset for the sake of feature selection because of the small ratio

between number of samples and number of features and due to its nonlinearity (unlike CorrAL dataset, which is a multi-variate dataset).

4.3 Parity3+3

The parity problem is a classic problem where the output is $f(x_1, \dots, x_n) = 1$ if the number of $x_i = 1$ is odd and $f(x_1, \dots, x_n) = 0$ otherwise. The Parity3+3 dataset is a modified version of the original parity dataset. The target concept is the parity of three bits. It contains 12 features among which 3 are relevant, another 3 are redundant (repeated) and other 6 are irrelevant (randomly generated).

4.4 The LED problem

The LED problem [74] is a simple classification task that consists of, given the active LEDs on a seven segments display, identifying the digit that the display is representing. Thus, the classification task to be solved is described by seven binary attributes (see Fig. 1) and ten possible classes available ($C = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$). A 1 in a attribute indicates that the LED is active, and a 0 indicates that it is not active.

Two versions of the LED problem will be used: the first one, Led25, adding 17 irrelevant attributes (with random binary values) and the second one, Led100, adding 92 irrelevant attributes. Both versions contain 50 samples. The small number of samples was chosen because we are interested in dealing with datasets with a high number of features and a small sample size. Besides, different levels of noise (altered inputs) have been added to the attributes of these two versions of the LED dataset: 2, 6, 10, 15 and 20%. In this manner, the tolerance to different levels of noise of the feature selection methods tested will be checked. Note that, as the attributes take binary values, adding noise means assigning to the relevant features an incorrect value.

4.5 Monk3

The MONK's problems [48] rely on an artificial robot domain, in which robots are described by six different attributes (x_1, \dots, x_6). The learning task is a binary classification task.

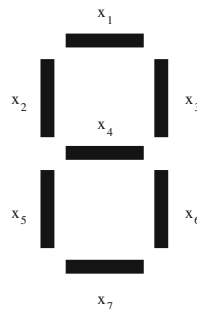


Fig. 1 LED scheme

The logical description of the class of the third problem (Monk3) is the following:

$$(x_5 = 3 \wedge x_4 = 1) \vee (x_5 \neq 4 \wedge x_2 \neq 3)$$

Among the 122 samples, 5 % are misclassifications, i.e., noise in the target.

4.6 SD1, SD2 and SD3

These three synthetic datasets (SD1, SD2 and SD3) [75] are challenging problems because of their high number of features (around 4,000) and the small number of samples (75), besides of a high number of irrelevant attributes. These characteristics reflect the problematic of micro-array data, and it is necessary to introduce some new definitions of multiclass relevancy features: full class relevant (FCR) and partial class relevant (PCR) features. Specifically, FCR denotes genes (features) that serve as candidate biomarkers for discriminating all cancer types. However, PCR are genes (features) that distinguish subsets of cancer types.

SD1, SD2 and SD3 are three-class datasets with 75 samples (each class containing 25 samples) generated based on the approach described in [76]. Each synthetic dataset consists of both relevant and irrelevant features. The relevant features in each dataset are generated from a multivariate normal distribution using mean and covariance matrixes [75]. Besides, 4,000 irrelevant features are added to each dataset, where 2,000 are drawn from a normal distribution of $N(0,1)$ and the other 2,000 are sampled with a uniform distribution $U[-1,1]$.

SD1 is designed to contain only 20 FCR and 4,000 irrelevant features. Two groups of relevant genes are generated from a multivariate normal distribution, with 10 genes in each group. Genes in the same group are redundant with each other and the optimal gene subset for distinguishing the three classes consists of any two relevant genes from different groups.

SD2 is designed to contain 10 FCR, 30 PCR, and 4,000 irrelevant features. Four groups of relevant, i.e., FCR and PCR, genes are generated from a multivariate normal distribution, with 10 genes in each group. Genes in each group are redundant to each other and in this dataset, only genes in the first group are FCR genes while genes in the three last groups are PCR genes. The optimal gene subset to distinguish all the three classes consists of four genes, one FCR gene from the first group and three PCR genes each from one of the three remaining groups.

SD3 has been designed to contain only 60 PCR and 4,000 irrelevant features. Six groups of relevant genes are generated from a multivariate normal distribution, with 10 genes in each group. Genes in the same group are designed to be redundant to each other and the optimal gene subset to distinguish all the three classes thus consists of six genes with one from each group.

It has to be noted that the easiest dataset in order to detect relevant features is SD1, since it contains only FCR features and the hardest one is SD3, due to the fact that it contains only PCR genes, which are more difficult to detect.

4.7 Madelon

The Madelon dataset [6] is a 2 class problem originally proposed in the NIPS'2003 feature selection challenge. The relevant features are situated on the vertices of a five-dimensional hypercube. Five redundant features were added, obtained by multiplying the useful features by a random matrix. Some of the previously defined features were repeated to create 10 more features. The other 480 features are drawn from a Gaussian distribution and labeled randomly. This dataset presents high dimensionality both in number of features and in number

of samples and the data were distorted by adding noise, flipping labels, shifting and rescaling. For all these reasons, it conforms a hard dataset for the sake of feature selection.

5 Experimental settings

Twelve different feature selection methods are tested and compared in this work in order to draw useful conclusions. As was mentioned in the Introduction, there exist two major approaches in feature selection: *individual evaluation* and *subset evaluation*. Individual evaluation provides an ordered ranking of the features while subset evaluation produces a candidate feature subset. When a ranking of the features is returned, it is necessary to establish a threshold in order to discard the features less relevant for the algorithm. Unfortunately, where to establish the threshold is not an easy-to-solve question. Belanche et al. [49] opted for discarding those weights associated to the ranking which were further than two variances from the mean. On the other hand, Mejía-Lavalle et al. [64] use a threshold defined by the largest gap between two consecutive ranked attributes, and other authors [51] just studied the whole ranking paying more attention to the first ranked features. However, in this work it is impossible to use a threshold related to the weights associated to the ranking, since some of the ranker methods (SVM-RFE, mRMR and M_d) eliminate chunks of features at a time and do not provide weights. To solve this problem and for the sake of fairness, in these experiments we heuristically set the following rules to decide the number of features that ranker methods should return, according to the number of total features (N):

- if $N < 10$, select 75 % of features
- if $10 < N < 75$, select 40 % of features
- if $75 < N < 100$, select 10 % of features
- if $N > 100$, select 3 % of features

At this point it has to be clarified that the datasets SD, due to their particularities, will be analyzed in a different manner which will be explained in Sect. 6.5. According to the rules showed above, the number of features that will be returned by ranker methods is 5 for the datasets Corral, Parity3+3 and Monk3, 10 for the datasets Corral-100, XOR-100 and both versions of LED, and 15 for Madelon.

A scoring measure was defined in order to fairly compare the effectiveness showed by the different feature selection methods. The measure presented is a index of success $Suc.$, see (1), which attempts to reward the selection of relevant features and to penalize the inclusion of irrelevant ones, penalizing two situations:

- The solution is *incomplete*: there are relevant features lacking.
- The solution is *incorrect*: there are some irrelevant features.

$$Suc. = \left[\frac{R_s}{R_t} - \alpha \frac{I_s}{I_t} \right] \times 100, \quad (1)$$

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. The term α was introduced to ponder that choosing an irrelevant feature is better than missing a relevant one (i.e., we prefer an incorrect solution rather than an incomplete one). The parameter α is defined as $\alpha = \min\{\frac{1}{2}, \frac{R_t}{I_t}\}$. Note that the higher the success, the better the method, and 100 is the maximum.

In the case of ranker methods and in order to be fair, if all the optimal features are selected before any irrelevant feature, the index of success will be 100, due to the fact that the number

of features that ranker methods are forced to select is always larger than the number of relevant features.

As was explained above, the evaluation of the feature selection methods is done by counting the number of correct/wrong features. However, it is also interesting and a common practice in the literature [65] to see the classification accuracy obtained in a 10-fold cross-validation, in order to check if the true model (that is, the one with an index of success of 100) is also unique (that is, if is the only one that can achieve the best percentage of classification success). For this purpose, four well-known classifiers, based on different models, were chosen: C4.5 [66], naive Bayes (NB) [67], IB1 [68] and SVM [69]. Experimental evidence has shown that decision trees, such as C4.5, exhibit a degradation in the performance when faced with many irrelevant features. Similarly, instance-based learners, such as IB1, 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 [70]. 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 deteriorates quickly by adding redundant features, even if they are relevant to the concept. Finally, SVM can indeed suffer in high-dimensional spaces where many features are irrelevant [71].

6 Experimental results

In this section, the results after applying 12 different feature selection methods over 11 datasets will be presented, grouped in different families which deal with situations such as presence of noise (both in the inputs and in the class), irrelevant features, redundancy, etc. The behavior of the feature selection methods will be tested according to the proposed index of success (see Eq. (1)) and the classification accuracy obtained by 4 different classifiers. It is necessary to note that all the feature selection methods tested in this work are deterministic, i.e., the set of selected features is unique; therefore, it is not necessary to repeat the experiments and average them.

In all tables of this section, the best index of success is highlighted in bold face, while best accuracy for each classifier is shaded and best accuracy for all 4 classifiers is also in bold face. Columns “Rel.”, “Irr.” and “Suc.” refer to the evaluation via counting the number of correct features selected, while the remaining columns show the classification accuracy obtained by four different classifiers. It has to be noted that for the calculation of the index of success, the redundant attributes selected have the same penalization than the irrelevant features. The results presented in this section will be analyzed and discussed in Sect. 8, after presenting some cases of study in Sect. 7.

6.1 Dealing with correlation and redundancy: CorrAL

Two versions of this well-known datasets were used: CorrAL (the classic dataset) and CorrAL-100, formed by adding 93 irrelevant binary features. The desired behavior of a feature selection method is to select the 4 relevant features and to discard the irrelevant ones, as well as detecting the correlated feature and not selecting it.

Tables 3 and 4 show the results obtained over the datasets Corral and Corral-100, respectively. Over Corral, FS-P was able to select the desired set of features, which led to 100% classification accuracy obtained by IB1 classifier. Regarding Corral-100, it is curious that the

Table 3 Results for CorrAL

Method	Rel.	C	Irr.	Suc.	Accuracy (%)			
					C4.5	NB	IB1	SVM
CFS	–	✓	0	–25	75.00	75.00	59.38	75.00
Consistency	–	✓	0	–25	75.00	75.00	59.38	75.00
INTERACT	–	✓	0	–25	75.00	75.00	59.38	75.00
InfoGain	–	✓	0	–25	75.00	75.00	59.38	75.00
ReliefF	1–4	✓	0	75	62.50	81.25	96.88	87.50
mRMR	1–4	✓	0	75	62.50	81.25	96.88	87.50
$M_d(\lambda = 0)$	1–4	✓	0	75	62.50	81.25	96.88	87.50
$M_d(\lambda = 1)$	1–4	✓	0	75	62.50	81.25	96.88	87.50
SVM-RFE	1–4	✓	0	75	62.50	81.25	96.88	87.50
SVM-RFE nonlinear	1–4	×	1	75	81.25	78.13	81.25	71.86
FS-P	1–4	×	0	100	81.25	78.13	100.00	81.25
Wrapper SVM	–	✓	0	–25	75.00	75.00	59.38	75.00
Wrapper C4.5	–	✓	0	–25	75.00	75.00	59.38	75.00

“Rel.” shows the relevant features selected, “C” indicates if the correlated feature is selected (✓) or not (×), “Irr.” means the number of irrelevant features selected and “Suc.” represents the index of success

Table 4 Results for CorrAL-100

Method	Rel.	C	Irr.	Suc.	Accuracy (%)			
					C4.5	NB	IB1	SVM
CFS	–	✓	0	–2	75.00	75.00	59.38	75.00
Consistency	–	✓	0	–2	75.00	75.00	59.38	75.00
INTERACT	–	✓	0	–2	75.00	75.00	59.38	75.00
InfoGain	–	✓	0	–2	75.00	75.00	59.38	75.00
ReliefF	1–3	✓	6	75	53.13	84.38	87.50	81.25
mRMR	1–4	✓	5	99	53.13	81.25	90.63	90.63
$M_d(\lambda = 0)$	1–4	✓	5	99	65.63	81.25	87.50	81.25
$M_d(\lambda = 1)$	1–4	✓	5	99	59.38	84.38	81.25	87.50
SVM-RFE	4	✓	8	25	62.50	87.50	68.75	96.88
SVM-RFE non-linear	–	✓	9	–44	68.75	68.75	62.50	75.00
FS-P	1,3,4	✓	6	75	53.13	87.50	84.38	87.50
Wrapper SVM	–	✓	0	–2	75.00	75.00	59.38	75.00
Wrapper C4.5	–	✓	2	–13	84.38	75.00	75.00	75.00

“Rel.” shows the relevant features selected, “C” indicates if the correlated feature is selected (✓) or not (×), “Irr.” means the number of irrelevant features selected and “Suc.” represents the index of success

best classification accuracy was obtained by SVM-RFE, which has a index of success of 25, but this fact can be explained because in this dataset there are some irrelevant features that are informative to the classifiers. This fact will be further analyzed in Sect. 8.

6.2 Dealing with nonlinearity: XOR and Parity

In this subsection, two nonlinear problems will be tested. XOR-100 contains 2 relevant features and 97 irrelevant features while Parity3+3 has 3 relevant, 3 redundant and 6 irrelevant

Table 5 Results for XOR-100

Method	Rel.	Irr.	Suc.	Accuracy (%)			
				C4.5	NB	IB1	SVM
ReliefF	1,2	0	100	100.00	64.00	100.00	70.00
mRMR	1	9	50	52.00	74.00	64.00	72.00
$M_d(\lambda = 0)$	1	9	50	54.00	74.00	58.00	70.00
$M_d(\lambda = 1)$	1	9	50	58.00	70.00	62.00	62.00
SVM-RFE	—	10	−21	48.00	68.00	56.00	78.00
SVM-RFE non-linear	1,2	0	100	100.00	64.00	100.00	70.00
FS-P	1	9	50	62.00	76.00	62.00	74.00
Wrapper SVM	—	1	−2	66.00	66.00	60.00	66.00
Wrapper C4.5	1,2	2	99	100.00	70.00	96.00	50.00

“Rel.” shows the relevant features selected, “Red” indicates the number of redundant features selected, “Irr.” means the number of irrelevant features selected and “Suc.” represents the index of success

Table 6 Results for Parity3+3

Method	Rel.	Red.	Irr.	Suc.	Accuracy (%)			
					C4.5	NB	IB1	SVM
ReliefF	1,2,3	2	0	93	90.63	29.69	100.00	37.50
mRMR	2,3	0	3	56	60.94	59.38	59.38	59.38
$M_d(\lambda = 0)$	—	0	5	−19	53.13	57.81	50.00	59.38
$M_d(\lambda = 1)$	—	0	5	−19	54.69	54.69	54.69	57.81
SVM-RFE	3	0	4	19	54.69	59.38	46.88	57.81
SVM-RFE non-linear	1,2,3	0	0	100	90.63	31.25	100.00	25.00
FS-P	—	0	5	−19	51.56	57.81	56.25	57.81
Wrapper SVM	—	0	1	−4	64.06	64.06	57.81	64.06
Wrapper C4.5	—	0	1	−4	64.06	64.06	57.81	64.06

“Rel.” shows the relevant features selected, “Red” indicates the number of redundant features selected, “Irr.” means the number of irrelevant features selected and “Suc.” represents the index of success

features. The ability of feature selection methods to deal with relevance, irrelevance and redundancy will be checked over two nonlinear scenarios, in the case of XOR-100 with the added handicap of a small ratio between number of samples and number of features. For the sake of completeness, SVM and naive Bayes will be applied over these two datasets. However, bearing in mind that those methods are linear classifiers (a linear kernel is being used for SVM) and no good results are expected, so they will not be the focus in the analysis in Sect. 8.

As can be seen in Tables 5 and 6, the methods CFS, Consistency, INTERACT and Info-Gain do not appear because they were not able to solve these nonlinear problems, so they returned an empty subset of features. In those cases, the classifiers were not applied either because with no features, the maximum achievable accuracy is known and it coincides with the baseline accuracy shown in Table 2.

On the other hand, the filter ReliefF and the embedded method SVM-RFE with a nonlinear kernel detected the relevant features both in XOR-100 and in Parity3+3, achieving the best indices of success and leading to high classification accuracies.

6.3 Dealing with noise in the inputs: LED

The LED dataset consists of correctly identifying seven LEDs that represent numbers between 0 and 9. Some irrelevant features were added forming the Led-25 dataset (17 irrelevant features) and the Led-100 dataset (92 irrelevant attributes). In order to make these datasets more complex, different levels of noise in the inputs (2, 6, 10, 15 and 20 %) were added. It has to be noted that, as the attributes take binary values, adding noise means assigning to the relevant features an incorrect value.

In Tables 7 and 8 one can see detailed results of these experiments. It is interesting to note that subset filters (CFS, Consistency and INTERACT) and the ranker filter Information Gain (which has a behavior similar to subset filters) do not select any of the irrelevant features in any case, at the expense of discarding some of the relevant ones, especially with high levels of noise. With regard to the classification accuracy, it decreases as the level of noise increases, as expected.

6.4 Dealing with noise in the target: Monk3

In this subsection, the Monk3 problem, which includes a 5 % of misclassifications, i.e., noise in the target, will be tested. The relevant features are x_2 , x_4 and x_5 . However, as it was stated in [77], for a feature selection algorithm it is easy to find the variables x_2 and x_5 , which together yield the second conjunction in the expression seen in Sect. 4.5. According to the experimental results presented in [77], selecting those features can lead to a better performance than selecting the three relevant ones. This additional information can help to explain the fact that in Table 9 several algorithms selected only two of the relevant features.

Studying the index of success in Table 9, one can see that only ReliefF achieved a value of 100. The worst behavior was showed by mRMR, since it selected the three irrelevant features. As was justified above, many methods selected only two of the relevant features and it can be considered a good comportment. For IB1 classifier, the best accuracy corresponds to ReliefF, which also obtained the best result in terms of index of success.

6.5 Dealing with a small ratio samples/features: SD1, SD2 and SD3

These synthetic datasets have a small ratio between number of samples and features, which makes difficult the task of feature selection. This is the problematic present in microarray data, a hard challenge for machine learning researchers. Besides these particularities of the data, there is a high number of irrelevant features for the task of gene classification and also the presence of redundant variables is a critical issue.

For these datasets, besides of using the index of success and classification accuracy, we will use the measures employed in [75], which are more specific for this problem. Hence, the performance of SD1, SD2 and SD3 will be also evaluated in terms of:

- (#): number of selected features.
- **OPT**(x): number of selected features within the optimal subset, where x indicates the optimal number of features.
- **Red**: number of redundant features.
- **Irr**: number of irrelevant features.

For the ranker methods ReliefF, mRMR, \mathcal{M}_d , SVM-RFE and FS-P, two different cardinalities were tested: the optimal number of features and 20, since the subset methods have a similar cardinality. It has to be noted that in this problem and for the calculation of the index of success, redundant features are treated the same as irrelevant features in Eq. (1). Notice

Table 7 Results for Led-25 dataset with different levels of noise (N) in inputs

N(%)	Method	Relevant	Irr. No.	Suc.	Accuracy (%)			
					C4.5	NB	IB1	SVM
0	CFS	1–5,7	0	86	92.00	100.00	100.00	96.00
	Consistency	1–5	0	71	92.00	100.00	100.00	96.00
	INTERACT	1–5,7	0	86	92.00	100.00	100.00	96.00
	InfoGain	1–7	0	100	92.00	100.00	100.00	96.00
	ReliefF	1–7	3	93	92.00	96.00	84.00	96.00
	mRMR	1–7	3	93	92.00	98.00	90.00	98.00
	$M_d(\lambda = 0)$	1–5,7	4	76	90.00	92.00	82.00	98.00
	$M_d(\lambda = 1)$	1–5,7	4	76	90.00	92.00	82.00	96.00
	SVM-RFE	1–7	3	93	92.00	98.00	80.00	96.00
	SVM-RFE non-linear	1–7	0	100	92.00	100.00	100.00	96.00
	FS-P	1–7	0	100	92.00	100.00	100.00	96.00
	Wrapper SVM	1–5	2	67	92.00	90.00	82.00	100.00
	Wrapper C4.5	1–5	0	71	92.00	90.00	82.00	96.00
2	CFS	1–5	0	71	90.00	98.00	96.00	94.00
	Consistency	1–5	0	71	90.00	98.00	96.00	94.00
	INTERACT	1–5	0	71	90.00	98.00	96.00	94.00
	InfoGain	1–7	0	100	90.00	96.00	94.00	94.00
	ReliefF	1–7	3	93	86.00	88.00	82.00	88.00
	mRMR	1–7	3	93	84.00	88.00	82.00	88.00
	$M_d(\lambda = 0)$	1–5,7	4	76	84.00	84.00	76.00	94.00
	$M_d(\lambda = 1)$	1–5,7	4	76	84.00	84.00	78.00	88.00
	SVM-RFE	1–7	3	93	86.00	88.00	86.00	94.00
	SVM-RFE non-linear	1–7	3	93	88.00	92.00	80.00	86.00
	FS-P	1–7	0	100	90.00	96.00	94.00	94.00
	Wrapper SVM	1–5	2	67	90.00	88.00	80.00	96.00
	Wrapper C4.5	1–5	0	71	90.00	98.00	96.00	94.00
6	CFS	1,2,4,5,7	0	71	70.00	76.00	66.00	68.00
	Consistency	1,2,4,5,7	0	71	70.00	76.00	66.00	68.00
	INTERACT	1,2,4,5,7	0	71	70.00	76.00	66.00	68.00
	InfoGain	1,2,4,5,7	0	71	70.00	76.00	66.00	68.00
	ReliefF	1–7	3	93	64.00	62.00	66.00	64.00
	mRMR	1–7	3	93	64.00	62.00	66.00	64.00
	$M_d(\lambda = 0)$	1–5,7	4	76	62.00	62.00	60.00	68.00
	$M_d(\lambda = 1)$	1–5,7	4	76	62.00	64.00	60.00	66.00
	SVM-RFE	1–4,7	5	59	62.00	62.00	58.00	68.00
	SVM-RFE non-linear	1–5	5	59	60.00	62.00	58.00	64.00
	FS-P	1–6	4	76	64.00	56.00	56.00	62.00
	Wrapper SVM	1,2,4,5	3	50	68.00	66.00	58.00	70.00
	Wrapper C4.5	1,2,4–6	1	69	72.00	68.00	58.00	66.00

Table 7 continued

N(%)	Method	Relevant	Irr. No.	Suc.	Accuracy (%)			
					C4.5	NB	IB1	SVM
10	CFS	1,2,4,7	0	57	60.00	50.00	58.00	46.00
	Consistency	1,2,4,7	0	57	60.00	50.00	58.00	46.00
	INTERACT	1,2,4,7	0	57	60.00	50.00	58.00	46.00
	InfoGain	1,2,4,7	0	57	60.00	50.00	58.00	46.00
	ReliefF	1–5,7	4	76	56.00	50.00	52.00	54.00
	mRMR	1–5,7	4	76	56.00	50.00	52.00	54.00
	$M_d(\lambda = 0)$	1–5,7	4	76	58.00	52.00	52.00	60.00
	$M_d(\lambda = 1)$	1–5,7	4	76	56.00	50.00	52.00	54.00
	SVM-RFE	2,4,7	7	26	44.00	40.00	42.00	50.00
	SVM-RFE non-linear	1–5	5	59	58.00	48.00	46.00	56.00
	FS-P	1–7	3	93	60.00	48.00	58.00	52.00
	Wrapper SVM	1,2,4–6	2	67	66.00	54.00	52.00	68.00
	Wrapper C4.5	1–4	0	57	68.00	64.00	58.00	62.00
15	CFS	1,7	0	29	28.00	28.00	32.00	36.00
	Consistency	1,7	0	29	28.00	28.00	32.00	36.00
	INTERACT	1,7	0	29	28.00	28.00	32.00	36.00
	InfoGain	1,7	0	29	28.00	28.00	32.00	36.00
	ReliefF	1–7	3	93	48.00	38.00	46.00	52.00
	mRMR	1–7	3	93	48.00	38.00	46.00	52.00
	$M_d(\lambda = 0)$	1–5,7	4	76	48.00	42.00	44.00	52.00
	$M_d(\lambda = 1)$	1–5,7	4	76	44.00	36.00	46.00	48.00
	SVM-RFE	2,7	8	9	26.00	34.00	28.00	36.00
	SVM-RFE non-linear	1,3,4,7	6	43	34.00	30.00	26.00	38.00
	FS-P	1–7	3	93	48.00	40.00	48.00	54.00
	Wrapper SVM	1,2	3	21	52.00	48.00	40.00	56.00
	Wrapper C4.5	1,2,5,7	0	57	58.00	44.00	40.00	54.00
20	CFS	1	0	14	28.00	20.00	28.00	28.00
	Consistency	1	0	14	28.00	20.00	28.00	28.00
	INTERACT	1	0	14	28.00	20.00	28.00	28.00
	InfoGain	1	0	14	28.00	20.00	28.00	28.00
	ReliefF	1–7	3	93	24.00	30.00	40.00	42.00
	mRMR	1–7	3	93	24.00	30.00	40.00	42.00
	$M_d(\lambda = 0)$	1–5,7	4	76	36.00	36.00	34.00	44.00
	$M_d(\lambda = 1)$	1–5,7	4	76	36.00	32.00	38.00	44.00
	SVM-RFE	5,6	8	9	16.00	24.00	18.00	20.00
	SVM-RFE non-linear	1,2,4,5	6	43	38.00	36.00	20.00	34.00
	FS-P	1–7	3	93	34.00	34.00	36.00	40.00
	Wrapper SVM	1,2,5,7	3	50	44.00	38.00	38.00	56.00
	Wrapper C4.5	1,2,5	3	36	48.00	40.00	34.00	42.00

that the index of success is 100 even with 25 irrelevant features selected, due to the high number of irrelevant features (4,000).

Studying the selected features, the subset filters and InfoGain (which exhibits a similar behavior) showed excellent results, in all SD1, SD2 and SD3. Also, SVM-RFE obtained

Table 8 Results for Led-100 dataset with different levels of noise (N) in inputs

N(%)	Method	Relevant	Irr. No.	Suc.	Accuracy (%)			
					C4.5	NB	IB1	SVM
0	CFS	1–5,7	0	86	92.00	100.00	100.00	96.00
	Consistency	1–5	0	71	92.00	100.00	100.00	96.00
	INTERACT	1–5,7	0	86	92.00	100.00	100.00	96.00
	InfoGain	1–7	0	100	92.00	100.00	100.00	96.00
	ReliefF	1–7	3	99	92.00	94.00	96.00	100.00
	mRMR	1–5,7	4	85	92.00	94.00	88.00	96.00
	$M_d(\lambda = 0)$	1–5,7	4	85	86.00	92.00	76.00	92.00
	$M_d(\lambda = 1)$	1–5,7	4	85	86.00	92.00	90.00	96.00
	SVM-RFE	3–7	5	71	46.00	54.00	48.00	48.00
	SVM-RFE non-linear	1–6	4	85	92.00	92.00	80.00	94.00
	FS-P	1–7	3	99	92.00	92.00	86.00	96.00
	Wrapper SVM	1–5	2	71	92.00	90.00	82.00	100.00
	Wrapper C4.5	1–5	0	71	92.00	100.00	100.00	96.00
2	CFS	1–5	0	71	90.00	98.00	96.00	94.00
	Consistency	1–5	0	71	90.00	98.00	96.00	94.00
	INTERACT	1–5	0	71	90.00	98.00	96.00	94.00
	InfoGain	1–7	0	100	90.00	96.00	94.00	94.00
	ReliefF	1–7	3	99	90.00	90.00	84.00	92.00
	mRMR	1–5,7	4	85	88.00	86.00	80.00	86.00
	$M_d(\lambda = 0)$	1–5,7	4	85	84.00	86.00	76.00	84.00
	$M_d(\lambda = 1)$	1–5,7	4	85	84.00	86.00	76.00	84.00
	SVM-RFE	3–7	5	71	68.00	70.00	54.00	70.00
	SVM-RFE non-linear	1–6	4	85	90.00	90.00	74.00	88.00
	FS-P	1–7	3	99	90.00	86.00	82.00	90.00
	Wrapper SVM	1–5	2	71	90.00	88.00	80.00	96.00
	Wrapper C4.5	1–5	0	71	90.00	98.00	96.00	94.00
6	CFS	1,2,4,5,7	0	71	72.00	78.00	72.00	70.00
	Consistency	1,2,4,5,7	0	71	72.00	78.00	72.00	70.00
	INTERACT	1,2,4,5,7	0	71	72.00	78.00	72.00	70.00
	InfoGain	1,2,4,5,7	0	71	72.00	78.00	72.00	70.00
	ReliefF	1–5,7	4	85	60.00	66.00	68.00	72.00
	mRMR	1–5,7	4	85	60.00	66.00	68.00	72.00
	$M_d(\lambda = 0)$	1,2,4,5,7	5	71	58.00	56.00	56.00	62.00
	$M_d(\lambda = 1)$	1–5,7	4	85	58.00	64.00	66.00	64.00
	SVM-RFE	2,3,5	7	42	52.00	50.00	34.00	52.00
	SVM-RFE non-linear	1–6	4	85	70.00	72.00	50.00	72.00
	FS-P	1–6	4	85	72.00	56.00	62.00	70.00
	Wrapper SVM	1–7	15	99	56.00	54.00	58.00	84.00
	Wrapper C4.5	1,2,4,5	2	57	76.00	72.00	66.00	72.00

Table 8 continued

N(%)	Method	Relevant	Irr. No.	Suc.	Accuracy (%)			
					C4.5	NB	IB1	SVM
10	CFS	1,2,4,7	0	57	60.00	50.00	58.00	46.00
	Consistency	1,2,4,7	0	57	60.00	50.00	58.00	46.00
	INTERACT	1,2,4,7	0	57	60.00	50.00	58.00	46.00
	InfoGain	1,2,4,7	0	57	60.00	50.00	58.00	46.00
	ReliefF	1,2,4,5,7	5	71	74.00	54.00	66.00	64.00
	mRMR	1,2,4,5,7	5	71	66.00	60.00	66.00	66.00
	$M_d(\lambda = 0)$	1,2,4,5,7	5	71	68.00	58.00	72.00	60.00
	$M_d(\lambda = 1)$	1,2,4,5,7	5	71	74.00	56.00	62.00	66.00
	SVM-RFE	2,3,5,7	6	57	44.00	36.00	38.00	42.00
	SVM-RFE non-linear	1,3,5	7	42	26.00	34.00	30.00	40.00
	FS-P	1-6	4	85	60.00	46.00	48.00	58.00
	Wrapper SVM	1,2,4	9	42	72.00	56.00	56.00	78.00
	Wrapper C4.5	1,2,4	3	43	76.00	58.00	56.00	66.00
15	CFS	1,7	0	29	28.00	28.00	32.00	36.00
	Consistency	1,7	0	29	28.00	28.00	32.00	36.00
	INTERACT	1,7	0	29	28.00	28.00	32.00	36.00
	InfoGain	1,7	0	29	28.00	28.00	32.00	36.00
	ReliefF	1,2,4,5,7	5	71	54.00	50.00	54.00	64.00
	mRMR	1,2,4,5,7	5	71	54.00	50.00	54.00	64.00
	$M_d(\lambda = 0)$	1,2,4,5,7	5	71	54.00	50.00	54.00	64.00
	$M_d(\lambda = 1)$	1,2,4,5,7	5	71	58.00	50.00	52.00	56.00
	SVM-RFE	3,5,7	7	42	30.00	20.00	16.00	26.00
	SVM-RFE non-linear	1,5	8	28	16.00	24.00	12.00	16.00
	FS-P	1,3,5,6,7	5	71	30.00	28.00	22.00	26.00
	Wrapper SVM	1,2,6	5	42	50.00	50.00	42.00	64.00
	Wrapper C4.5	1,2,5,7	2	57	58.00	50.00	46.00	52.00
20	CFS	1	0	14	28.00	20.00	28.00	28.00
	Consistency	1	0	14	28.00	20.00	28.00	28.00
	INTERACT	1	0	14	28.00	20.00	28.00	28.00
	InfoGain	1	0	14	28.00	20.00	28.00	28.00
	ReliefF	1,2,5,7	6	57	30.00	38.00	44.00	44.00
	mRMR	1,2,5,7	6	57	34.00	38.00	42.00	48.00
	$M_d(\lambda = 0)$	1,2,5,7	6	57	32.00	38.00	38.00	32.00
	$M_d(\lambda = 1)$	1,2,5,7	6	57	38.00	32.00	28.00	34.00
	SVM-RFE	—	10	—1	8.00	26.00	20.00	20.00
	SVM-RFE non-linear	1,2,3,5	6	57	32.00	32.00	14.00	26.00
	FS-P	1-3,5,6	5	71	18.00	24.00	24.00	20.00
	Wrapper SVM	1	3	14	36.00	38.00	28.00	44.00
	Wrapper C4.5	1,5	4	28	44.00	32.00	28.00	32.00

good results, although the version with a nonlinear kernel could not be applied on these datasets due to memory complexity. With respect to the classifiers, SVM achieves the highest accuracies.

Table 9 Results for Monk3

Method	Relevant	Irr. No.	Suc.	Accuracy (%)			
				C4.5	NB	IB1	SVM
CFS	2,5	0	67	93.44	88.52	89.34	79.51
Consistency	2,5	0	67	93.44	88.52	89.34	79.51
INTERACT	2,5	0	67	93.44	88.52	89.34	79.51
InfoGain	2,5	0	67	93.44	88.52	89.34	79.51
ReliefF	2,5,4	0	100	93.44	88.52	90.98	80.33
mRMR	2,5	3	17	92.62	88.52	80.33	78.69
$M_d(\lambda = 0)$	2,4,5	2	67	93.44	88.52	84.43	81.97
$M_d(\lambda = 1)$	2,4,5	2	67	93.44	88.52	84.43	81.97
SVM-RFE	2,4,5	2	67	93.44	88.52	84.43	84.43
SVM-RFE non-linear	2,4,5	2	67	93.44	88.52	84.43	84.43
FS-P	2,4,5	2	67	93.44	88.52	84.43	84.43
Wrapper SVM	2,4,5	1	83	93.44	89.34	82.79	79.51
Wrapper C4.5	2,5	0	67	93.44	88.52	89.34	79.51

Relevant features: 2,4,5

Table 10 Results for Madelon

Method	Relevant	Red. No.	Irr. No.	Suc.	Accuracy (%)			
					C4.5	NB	IB1	SVM
CFS	3	7	0	20	80.92	69.58	86.83	66.08
Consistency	3,4	10	0	40	83.54	69.67	90.83	66.83
INTERACT	3,4	10	0	40	83.54	69.67	90.83	66.83
InfoGain	3,4	10	0	40	83.54	69.67	90.83	66.83
ReliefF	1,3,4,5	11	0	80	84.21	69.83	89.88	66.46
mRMR	—	1	14	0	64.92	62.25	53.13	57.08
$M_d(\lambda = 0)$	3,4	10	2	40	83.23	70.21	85.29	66.42
$M_d(\lambda = 1)$	3,4	10	2	40	83.23	70.21	85.29	66.42
SVM-RFE	1,3,4,5	4	7	80	86.42	66.88	81.25	67.42
FS-P	3,4	3	10	40	70.50	66.17	62.54	66.96
Wrapper SVM	3	0	16	20	66.63	66.04	54.08	67.54
Wrapper C4.5	1-5	5	15	99	87.04	70.00	75.42	66.33

Relevant features: 1–5

6.6 Dealing with a complex dataset: Madelon

Madelon is a very complex artificial dataset which is distorted by adding noise, flipping labels, shifting and rescaling. It is also a nonlinear problem, so it conforms a challenge for feature selection researchers. The desired behavior for a feature selection method is to select the relevant features (1–5) and discard the redundant and irrelevant ones.

Table 10 shows the relevant features selected by the feature selection methods, as well and the number of redundant and irrelevant features selected by them and the classification accuracy. Notice that for the calculation of index of success, the redundant attributes selected stand for irrelevant features. Again the results for SVM and naive Bayes will not be analyzed, since they are linear classifiers. The best result in terms of index of success was obtained

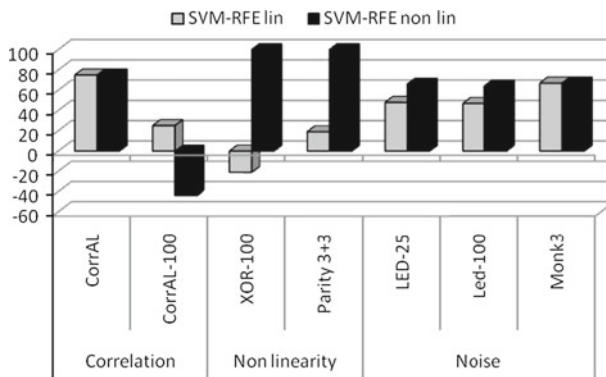


Fig. 2 SVM-RFE: linear vs. nonlinear kernel. The vertical axis represents the index of success

by the wrapper with C4.5, selecting all the 5 relevant features, which also led to the best classification accuracy for C4.5.

7 Cases of study

After presenting the experimental results, and before discussing and analyzing them in detail, we will describe several cases of study in order to decide among similar methods. These cases of study will be based on the index of success, to make this analysis classifier-independent.

7.1 Case of study I: linear vs. nonlinear kernel for SVM-RFE

As was stated in Sect. 3.2, two different kernels were applied on the embedded method SVM-RFE. A nonlinear kernel allows to solve nonlinear problems, but at the expense of being more computationally demanding. In fact, SVM-RFE with a nonlinear kernel could not be applied on the datasets SD and Madelon, due to the space complexity. In Fig. 2, one can see a comparison of these two versions of the method. Note that as for both Led-25 and Led-100 datasets there are results for 6 different levels of noise in the inputs, we have opted for computing the average of the index of success.

As expected, the linear kernel is not able to deal with nonlinear problems (XOR-100 and Parity3+3). On the other hand, the nonlinear kernel achieves a poor result over Corral-100 dataset (where the number of irrelevant features increases considerably). In the remaining datasets, the nonlinear kernel maintains or increases the performance of the linear kernel. In these cases, it is necessary to bear in mind that the nonlinear kernel raises the computational time requirements of the algorithms and it cannot be applied over high-dimensional datasets (such as Madelon and the SD family). For example, over XOR-100 dataset, it takes almost 4 times more time to use the nonlinear kernel than the linear one. The authors suggest to use the nonlinear kernel when there is some knowledge about the nonlinearity of the problem, and to use the linear kernel in the remaining cases, specially when dealing with large amounts of data.

7.2 Case of study II: mRMR vs. \mathcal{M}_d

The filter method \mathcal{M}_d is an extension of mRMR, which instead of mutual information, uses a measure of dependence to assess relevance and irrelevance. Besides, it included a free

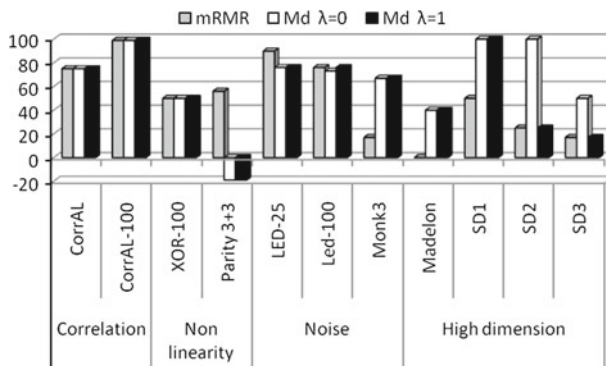


Fig. 3 mRMR vs. \mathcal{M}_d . The vertical axis represents the index of success

parameter (λ) that controls the relative emphasis given on relevance and irrelevance. In light of the above, the authors think that it is interesting to compare the behaviors showed by these two methods over the artificial datasets studied in this work. Two values of lambda were tested, 0 and 1, and it is also important to see the difference between them. When λ is equal to zero, the effect of the redundancy disappears and the measure is based only on maximizing the relevance. On the other hand, when λ is equal to one, it is more important to minimize the redundancy. For the sake of fairness, note that for the SD family of datasets, we considered the results achieved selecting 20 features.

With regard to the different values of λ , one can see in Fig. 3 that the index of success is the same for most of the datasets tested (8 out of 11). However, there is a important improvement in SD2 and SD3 when the value of λ is zero. Therefore, the authors suggest to use this value of λ , although the appropriate value of λ is not an easy-to-solve question that requires to be studied further and seems to be very dependent of the nature of data.

Comparing \mathcal{M}_d and mRMR, the latter performs better in two datasets (Parity3+3 and Led25) whereas \mathcal{M}_d is better in 5 datasets (Monk3, Madelon, and the SD family). In the remaining datasets, the index of success achieved by both methods is the same. In light of these results, the authors recommend the use of \mathcal{M}_d except in datasets with high nonlinearity.

7.3 Case of study III: subset filters

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. This approach can handle feature redundancy with feature relevance, besides of releasing the user from the task of choosing how many features to retain.

In Fig. 4 one can see a comparison among the three subset filters studied in this work (CFS, INTERACT and Consistency-based) with regard to the index of success. All the three methods show in general a very similar behavior, although some differences have been found. Consistency-based is slightly worse on datasets which present noise (Led25, Led100). This can be explained because for this filter, a pattern is considered inconsistent if there exist at least two instances such that they match all but their class labels, and therefore the given subset of features is inconsistent and the features are discarded. This case can happen when the data have been distorted with noise. On the other hand, CFS decays on Madelon. As each feature is treated individually, this algorithm cannot identify really strong interactions

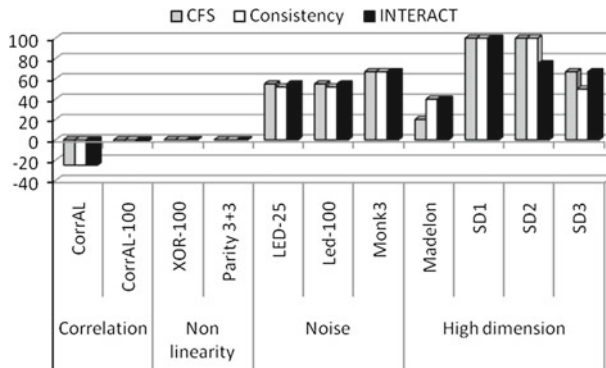


Fig. 4 Subset filters. The vertical axis represents the index of success

as the ones which may appear in parity problems (remind that Madelon is a generalization of a parity problem). In light of the above, the authors suggest to use INTERACT.

7.4 Case of study IV: different levels of noise in the input

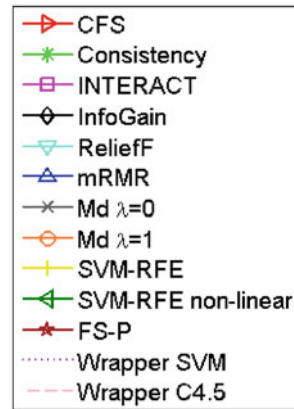
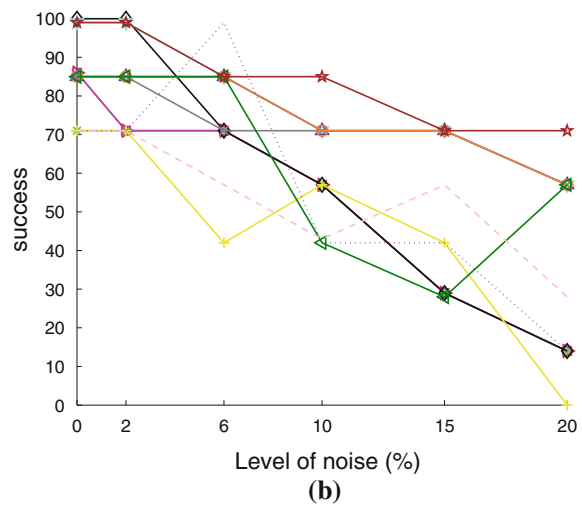
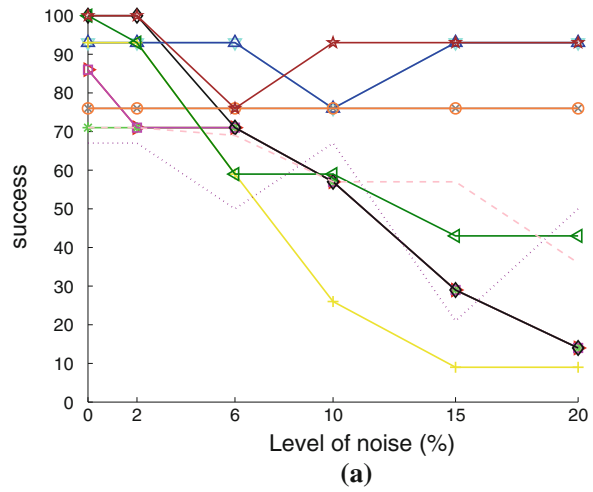
Figure 5 shows an overview of the behavior of feature selection methods with regard to different levels of noise, according to the index of success described in (1). As we would have expected, in general the index of success decreases when the level of noise increases, and worse performances were obtained over Led-100 due to the higher number of irrelevant features. It may seem strange that in some cases the index of success improves with higher levels of noise (for example, in Led-100 from 10 to 15 % of noise), but this fact can be explained by the random generation of the noise. Notice that the influence of each relevant feature is not the same in this problem, so adding noise to one or another may cause different results. In fact, the first five features (see Fig. 1) are enough to distinguish among the ten digits, therefore if these attributes are distorted, the result may be altered.

Several conclusions can be extracted from the graphs in Fig. 5. Regarding the wrapper model, both versions tested degrade their results with the presence of noise, both in Led-25 and Led-100. With respect to embedded methods, two opposite behaviors have been observed. On the one hand, FS-P achieved very promising results on both datasets, without showing degradation as the level of noise increases. In fact, the index of success oscillates between 76 and 100 on Led-25 (Fig. 5a) and between 71 and 99 on Led-100 (Fig. 5b). On the other hand, SVM-RFE (specially the version with the linear kernel) deteriorates considerably its behavior with high levels of noise. Note that SVM-RFE with linear kernel obtained 9 as index of success on Led-25 and -1 on Led-100, which is the worst results for all the feature selection methods tested.

Concerning the filter model, mRMR and ReliefF are the methods that achieve the best indices of success, being ReliefF slightly better in two cases (Led-100 with 0 and 2 % of noise). These two filters obtain very good results without being very affected by noise. On the contrary, the subset filters (CFS, Consistency and INTERACT) and Information Gain are affected by high levels of noise although they are robust to the addition of irrelevant features. Finally, with respect to \mathcal{M}_d , it attains constant results, particularly on Led-25, and no significative differences have been found between the two values of λ tested.

It is curious the opposite behaviors of Information Gain and mRMR, bearing in mind that both come from the Information Theory field. However, this fact can be explained because

Fig. 5 Results for Led-25 and Led-100. **a** Led-25. **b** Led-100. **c** Legend



(c)

Table 11 Average of success for every feature selection method tested

Method	Correlation		Non-linear		Noise			High dimension		Av.
	Corr.	Corr100	XOR100	Par3+3	Led25	Led100	Monk3	SD	Mad.	
CFS	−25	−2	0	0	55	55	67	89	20	29
Consistency	−25	−2	0	0	52	52	67	83	40	30
INTERACT	−25	−2	0	0	55	55	67	81	40	30
InfoGain	−25	−2	0	0	62	62	67	89	40	33
ReliefF	75	75	100	93	90	80	100	47	80	82
mRMR	75	99	50	56	90	76	17	31	0	55
M_d^a	75	99	50	−19	76	73	67	83	40	60
M_d^b	75	99	50	−19	76	76	67	47	40	57
SVM-RFE	75	25	−21	19	48	47	67	89	80	48
SVM-RFE ^c	75	−44	100	100	66	64	67	N/A	N/A	N/A
FS-P	100	75	−19	−19	93	85	67	22	40	49
Wrap. SVM	−25	−2	−4	−4	54	57	83	31	20	23
Wrap. C4.5	−25	−13	−4	−4	60	55	67	22	99	29

“N/A” stands for “Not Applicable”

^a $\lambda = 0$

^b $\lambda = 1$

^c Nonlinear kernel

Information Gain is a univariate measure that considers the entropy between a given feature and the class level. On the other hand, mRMR takes into account the mutual information among features. The latter is a multivariate measure and therefore a better behavior is expected when noise is present in data, because although some features may be affected by noise in a sample, not all of them are supposed to suffer it. This is why Information Gain obtains excellent results with low levels of noise but as it increases, its performance decays until reaching an index of success with value 14.

To sum up, the filters mRMR and ReliefF and the embedded method FS-P are the methods most tolerant to noise in the inputs and the subsets filters (CFS, Consistency and INTERACT) and Information Gain are the most affected by noise.

8 Analysis and discussion

In this section, an analysis and discussion of the results presented in Sect. 6 will be carried out, trying to check which method is the best and to explain some behaviors showed in the experimental results. In Sect. 8.1 we start analyzing the index of success while in Sect. 8.2 we will discuss the relation between index of success and classification accuracy focusing on the specific problems studied in this work.

8.1 Analysis of index of success

Table 11 shows the average of success for each feature selection method over each scenario and also an overall average for each method (last column). For Led25 and Led100 only one result is presented, respectively, corresponding to the average of the results for the distinct levels of noise tested. Analogously, for the SD family of datasets, only the average result of the 3 datasets is shown.

We are interested in an analysis of the index of success (regardless of the classification accuracy) in order to check the behavior of the feature selection methods in a classifier-independent manner. In light of the results shown in Table 11, the best method according to the index of success is the filter ReliefF, followed by the filters mRMR and both versions of \mathcal{M}_d . However, the subset filters and Information Gain (which has a similar behavior than those) showed poor results. Regarding the embedded model, FS-P is slightly better than SVM-RFE, and both of them are in the middle of the ranking. Finally, wrapper methods turned out to be the worst option in this study, since they achieved the poorest averages of success.

In light of the results presented in Table 11, the authors suggest some guidelines:

- In complete ignorance of the particulars of data, the authors suggest to use the filter ReliefF. It detects relevance in a satisfactory manner, even in complex datasets such as XOR-100, and it is tolerant to noise (both in the inputs and in the output). Moreover, due to the fact that it is a filter, it has the implicit advantage of its low computational cost.
- When dealing with high nonlinearity of data (such as XOR-100 and Parity3+3), SVM-RFE with a nonlinear kernel is an excellent choice, since it is able to solve these complex problems. However, at the expense of being computationally more expensive than the remaining approaches seen in this work.
- In the presence of altered inputs, the best option is to use the embedded method FS-P, since it has proved to be very robust to noise. A less expensive alternative is the use of the filters ReliefF or mRMR, which also showed good behaviors over this scenario. With low levels of noise (up to 6 %), the authors also suggest the use of the filter Information Gain.
- When the goal is to select the smallest number of irrelevant features (even at the expense of selecting fewer relevant features), we suggest to employ one of the subset filters (CFS, Consistency-based or INTERACT). This kind of methods have the advantage of releasing the user from the task of deciding how many features to choose.
- When dealing with datasets with a small ratio between number of samples and features and a high number of irrelevant attributes, which is part of the problematics of microarray data, the subset filters and Information Gain presented a promising behavior. SVM-RFE performs also adequately, but because of being an embedded method is computationally expensive, especially in high-dimensional datasets like these.
- In general, the authors suggest the use of filters (specifically ReliefF), since they carry out the feature selection process with independence of the induction algorithm and are faster than embedded and wrapper methods. However, in case of using another approach, we suggest to use the embedded method FS-P.

As was stated in Table 1, filters are the methods with the lower computational cost while wrappers are computationally the most expensive. To illustrate this fact, in Fig. 6 one can see the execution time of the different feature selection methods over two datasets: XOR-100 and Led-100 (with no noise). In order to be fair, mRMR was not included in this study because it was executed in a different machine; however, one can expect a computational time similar to the one required by \mathcal{M}_d .

As expected, the filter model achieves the lowest execution times, always below 1 second. The embedded methods require more computational time, specially SVM-RFE with a nonlinear kernel. In fact, the time required by this method over Led-100 is especially high because this is a multiclass dataset, a fact that also increases the computational time. Wrapper SVM contrary to Wrapper C4.5 is a very demanding method, particularly with Led-100, which needed almost 1 h.

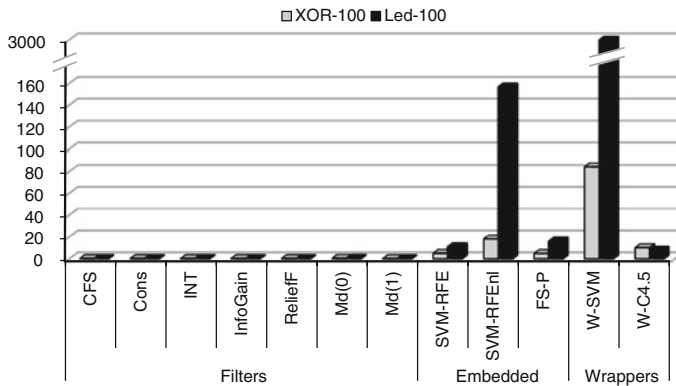


Fig. 6 Time in seconds for the datasets XOR-100 and Led-100

8.2 Analysis of classification accuracy

Although the previous analysis is interesting because it is not classifier-dependent, one may want to see the classification accuracy in order to check if the desired subset of features is unique and if it is the best option. For the sake of brevity, only the two extreme values of noise for Led-25 and Led-100 datasets were included in this study. Table 12 shows, for each classifier and dataset, the best accuracy obtained, as well as the corresponding index of success and feature selection method employed. In this manner, it is easy to see at a glance if the best accuracy matches with the best index of success. In fact, this happens for all datasets except Led-100 with 20% of noise, where the inputs are clearly disturbed. This may be explained because the irrelevant features (randomly generated) are adding some information useful to the classifier, while the disturbed relevant features are not so informative.

IB1 classifier, based on nearest neighbors, seems to be the best match for the proposed index of success, since it obtains the best result for classification when the index obtains also its best result, specifically in 5 out of 13 datasets tested. Instance-based learners are very susceptible to irrelevant features; therefore, when a feature selection method only selects the relevant features, its index of success is high and also the classification accuracy obtained by this classifier. It has to be also noted that IB1 is a nonlinear classifier; therefore, it is capable to correctly classify problems such as XOR-100 or Parity3+3, achieving 100% of classification accuracy when other methods like SVM obtained poor results.

SVM obtained the highest classification accuracy in 7 out of 13 datasets showed in Table 12; however, it only coincides with the highest index of success in SD2 dataset. This predictor takes advantage of the embedded method SVM-RFE and Wrapper SVM, both methods using this classifier performance to select the features. In fact, the highest accuracies were obtained after applying one of those methods for all datasets except for Led-25 with 20% of noise.

Although the behavior of the classifiers is interesting, one may want to focus on the problems studied in this work. For dealing with correlation and redundancy, two datasets were evaluated in this paper: Corral and Corral-100 (see Tables 3 and 4). Focusing on Corral, the subset filters, InfoGain and the wrappers selected only the correlated feature, which leads to an accuracy of 75% for all the classifiers except IB1, which apparently is not able to take advantage of the relation between this feature and the class. When the four relevant features plus the correlated one are selected (rows 6–10 at previous tables), one can see that the

Table 12 Summary of results grouped by classifier

Dataset	C4.5			NB			IB1			SVM		
	Best Acc.	Suc.	Method	Best Acc.	Suc.	Method	Best Acc.	Suc.	Method	Best Acc.	Suc.	Method
CorrAL	81.25	100	FS-P	81.25	75	Rf,mRMR, M_d ,SR	100.00	100	FS-P	87.50	75	Rf,mRMR, M_d ,SR
		75	SRn									
CorrAL-100	84.38	-13	W-C4.5	87.50	75	FS-P	90.63	99	mRMR	96.88	25	SR
					25	SR						
XOR-100	100.00	100	Rf,SRn	76.00	50	FS-P	100.00	100	Rf,SRn	78.00	-21	SR
		99	W-C4.5									
Parity3+3	90.63	100	SRn	64.06	-4	W-SVM,W-C4.5	100.00	100	SRn	64.06	-4	W-SVM,W-C4.5
		93	Rf					93	Rf			
Led-25 (0%)	92.00	100	IG,SRn,FS-P	100.00	100	IG,SRn,FS-P	100.00	100	IG,SRn,FS-P	100.00	67	W-SVM
		93	Rf,mRMR,SR		86	CFS,INT		86	CFS,INT			
		86	CFS,INT		71	Cons		71	Cons			
		71	Cons,W-C4.5									
		67	W-SVM									
Led-25 (20%)	48.00	36	W-C4.5	40.00	36	W-C4.5	40.00	93	Rf,mRMR	56.00	50	W-SVM
		100	IG	100.00	100	IG	100.00	100	IG	100.00	99	Rf
Led-100 (0%)	92.00	99	Rf,FS-P		86	CFS,INT		86	CFS,INT		71	W-SVM
		86	CFS,INT		71	Cons,W-C4.5		71	Cons,W-C4.5			
		85	mRMR,SRn									
		71	Cons,W-SVM,W-C4.5									
Led-100 (20%)	44.00	28	W-C4.5	38.00	57 [*]	Rf,mRMR, M_d (0)	44.00	57 [*]	Rf	48.00	57 [*]	mRMR
					14	W-SVM						
Monk3	93.44	100	Rf	89.34	83	W-SVM	90.98	100	Rf	84.43	67	SR
		83	W-SVM									SRn
SD1	77.33	0	Rest except mRMR									FS-P
SD2	72.00	25	W-C4.5	88.00	100	SRn	76.00	100	SRn	94.67	50	W-SVM
SD3	68.00	17	W-C4.5	84.00	100	CFS	74.67	75	INT	84.00	100	SRn
Madelon	87.04	99	W-C4.5	85.33	67	SRn	73.33	67	CFS	82.67	67	SRn
		99	W-C4.5	70.21	40	M_d	90.83	40	Cons,INT,IG	67.54	20	W-SVM

^{*} 'Rf' stands for 'ReliefF', 'SR' for 'SVM-RFE', 'SRn' for 'SVM-RFE nonlinear' and 'IG' for 'Information Gain', respectively

^{*} This is not the highest index of success achieved

correlated feature (since it is correlated for 75 % of data) is hindering the process of classification, preventing the predictors to correctly classify all samples. FS-P was the only method that selected the four relevant features and discarded the irrelevant and correlated ones; IB1 was able to achieve a 100 % of classification accuracy, while the other methods were not. This fact is explained because of the complexity of the problem that may cause that a given classifier may not solve the problem satisfactorily, even with the proper features. Regarding Corral-100, the highest accuracy (96.88 %) was obtained by SVM having only one of the relevant features, the correlated one, and 8 irrelevant ones. This fact can seem surprising but it can be explained because the irrelevant features (randomly generated) are informative in this problem. Classifying only with the relevant feature and the correlated one, SVM achieves 65.62 % of classification accuracy; therefore, it is clear that the irrelevant features are adding some useful information to the learner. In fact, by randomly generating 94 binary features and having only 32 samples, the probability that some of these irrelevant features could be correlated with the class is very high. This situation happens again with Wrapper C4.5 and C4.5 classifier, while the remaining methods exhibit a similar behavior than on Corral.

Nonlinearity is a difficult problem to deal with. In fact, two of the classifiers employed in this work (SVM with linear kernel and naive Bayes) and several feature selection methods do not turn very good results. This problematic is present in XOR-100 and Parity3+3 datasets, in Tables 5 and 6. As we have said, naive Bayes and SVM cannot deal with nonlinearity therefore they will not be the focus in this section. On the other hand, IB1 (and C4.5 only over XOR-100) achieve 100 % of classification accuracy when the desired features are selected. Over XOR-100, C4.5 obtains also 100 % of prediction accuracy after applying its own wrapper, even when it selected two extra irrelevant features. It has to be noted that this classifier performs an embedded selection of the features; therefore, it may be using a subset of features smaller than the given by the feature selection method. Finally, it needs to be remarked the improvement in SVM-RFE when using a nonlinear kernel for this kind of datasets. SVM-RFE over XOR-100 did not select any of the relevant features, which led to a classification accuracy below the baseline accuracy. However, the computational time when using a nonlinear kernel is almost four times the one needed using the classical SVM-RFE (see Table 6) so when choosing one or the other it is a fact to bear in mind.

Different levels of noise in the input features were tested over Led-25 (Table 7) and Led-100 (Table 8) datasets. As expected, the classification accuracy decreases when the level of noise increases. It has to be noted that, for both datasets, selecting 5 out of the 7 relevant features is enough to achieve 100 % classification accuracy. Because segments 1–5 (see Fig. 1) are enough to distinguish the 10 digits (actually, 5 binary features allow to represent 32 different states). In fact, when the level of noise is 6 %, the first four methods miss the third feature (which allows to distinguish between digit 5 and 6) and the performance decays in 24 %, which cannot be ascribed to the level of noise. This is a case of classification showing that the true model is not unique. On the other hand, it is curious that in some cases such as ReliefF over Led-25 with 20 % of noise, where it achieves an index of success of 93 (selecting the 7 relevant features), the maximum classification accuracy obtained with these features was 40 % (SVM) which is not the result expected. This fact can be explained because of the high level of noise, which corrupts the relevant features and makes the classification task very complex. In those cases with high levels of noise, wrappers appear to be a good alternative, since they are classifier-dependent and try to search for the best features to the given classifier. To sum up, the filters mRMR and ReliefF and the embedded method FS-P are the methods most tolerant to noise in the inputs and the subsets filters (CFS, Consistency and INTERACT) and Information Gain are the most affected by noise, although wrappers are also a good choice if one is interested in maximizing the classification accuracy.

Table 13 Features selected by each algorithm on synthetic dataset SD1

	(#)	OPT(2)	Red	Irr	Suc	Accuracy (%)			
						C4.5	NB	IB1	SVM
CFS	28	2	1	25	100	57.33	82.67	69.33	77.33
Cons	8	2	0	6	100	54.67	76.00	60.00	66.67
INT	23	2	0	21	100	60.00	81.33	66.67	80.00
IG	42	2	15	25	100	58.67	72.00	70.67	78.67
ReliefF ^a	2	1	1	0	50	40.00	45.33	44.00	46.67
ReliefF ^b	20	2	13	5	100	60.00	61.33	70.67	73.33
mRMR ^a	2	1	0	1	50	41.33	49.33	34.67	50.67
mRMR ^b	20	1	0	19	50	54.67	82.67	68.00	78.67
SVM-RFE ^a	2	2	0	0	100	56.00	60.00	52.00	57.33
SVM-RFE ^b	20	2	3	15	100	46.67	88.00	76.00	92.00
FS-P ^a	2	0	0	2	0	37.33	49.33	41.33	50.67
FS-P ^b	20	1	2	17	50	53.33	76.00	65.33	73.33
$\mathcal{M}_d(\lambda = 0)^a$	2	1	1	0	50	41.33	48.00	32.00	44.00
$\mathcal{M}_d(\lambda = 0)^b$	20	2	17	1	100	56.00	62.67	46.67	66.67
$\mathcal{M}_d(\lambda = 1)^a$	2	1	0	1	50	48.00	61.33	58.67	57.33
$\mathcal{M}_d(\lambda = 1)^b$	20	2	17	1	100	54.67	62.67	46.67	66.67
W-SVM	19	1	0	18	50	44.00	74.67	58.67	94.67
W-C4.5	10	0	0	10	0	77.33	38.67	40.00	38.67

Ranker methods are tested selecting the optimal number and 20 features as cardinality

^a Selecting the optimal number of features

^b Selecting 20 features

Monk3 dataset (see Table 9) is studied to deal with noise in the target. As was explained in Sect. 6.4, there are evidences that features x_2 and x_5 are enough for certain classifier, which in fact happens in the experiments presented in this work. This is an example of the optimal feature subset being different than the subset of relevant features. On the other hand, again one can see the implicit capacity of C4.5 to select features, since it achieves the same result in cases where different subsets of features were selected, although for mRMR some of the irrelevant features caused the incorrect classification of one extra feature. This is not the case for IB1 classifier, which achieves the highest accuracy only when the “known” set of relevant features is selected. Naive Bayes and SVM seem to be more affected by misclassifications, since they obtain the worst results and do not take advantage of the best indices of success.

SD1, SD2 and SD3 (Tables 13, 14 and 15) introduce the problematic of microarray data: a small ratio between number of samples and features and a high number of redundant and irrelevant features. In general, the classification results are poor, because this kind of problems are very difficult to solve since the classifiers tend to overfit. Moreover, the accuracy decreases when the complexity of the dataset increases (SD3). The embedded method SVM-RFE achieves very good results, specially with the SVM classifier. CFS and INTERACT filters also work satisfactorily together with naive Bayes and IB1 classifiers. The small ratio between number of samples and features prevents the use of wrappers, which have the risk of overfitting due to the small sample size. In fact, one can see in Tables 13, 14 and 15 that the wrappers obtain high accuracies in conjunction with their corresponding classifiers, but the performance decreases when using other classifiers. Regarding the classifiers, SVM achieves good results, specially over SD1. SVMs have many mathematical features that make them

Table 14 Features selected by each algorithm on synthetic dataset SD2

	(#)	OPT(4)	Red	Irr	Suc	Accuracy (%)			
						C4.5	NB	IB1	SVM
CFS	21	4	0	17	100	64.00	84.00	72.00	81.33
Cons	9	4	0	5	100	54.67	70.67	60.00	72.00
INT	20	3	0	17	75	70.67	80.00	74.67	81.33
IG	40	4	19	17	100	61.33	69.33	61.33	76.00
ReliefF ^a	4	0	0	4	0	48.00	64.00	50.67	52.00
ReliefF ^b	20	1	9	10	25	54.67	60.00	61.33	70.67
mRMR ^a	4	1	0	3	25	54.67	64.00	60.00	57.33
mRMR ^b	20	1	0	19	25	60.00	70.67	44.00	68.00
SVM-RFE ^a	4	3	1	0	75	46.67	62.67	54.67	65.33
SVM-RFE ^b	20	4	4	12	100	57.33	82.67	69.33	84.00
FS-P ^a	4	0	0	20	0	42.67	54.67	40.00	57.33
FS-P ^b	20	0	0	20	0	52.00	68.00	42.67	61.33
$\mathcal{M}_d(\lambda = 0)^a$	4	2	2	0	50	56.00	56.00	26.67	50.67
$\mathcal{M}_d(\lambda = 0)^b$	20	4	16	0	100	54.67	64.00	49.33	68.00
$\mathcal{M}_d(\lambda = 1)^a$	4	1	0	3	25	46.67	69.33	56.00	69.33
$\mathcal{M}_d(\lambda = 1)^b$	20	1	9	10	25	52.00	62.67	60.00	74.67
W-SVM	13	1	0	12	25	44.00	60.00	45.33	77.33
W-C4.5	6	1	0	5	25	72.00	46.67	34.67	42.67

Ranker methods are tested selecting the optimal number and 20 features as cardinality

^a Selecting the optimal number of features

^b Selecting 20 features

attractive for gene expression analysis, including their flexibility in choosing a similarity function, sparseness of solution when dealing with large datasets, the ability to handle large feature spaces, and the ability to identify outliers [78]. Naive Bayes obtained also high accuracies, specially over SD2 and SD3. This learner is robust with respect to irrelevant features, although it deteriorates quickly by adding redundant features. In fact, it obtains the best accuracies when a small number of redundant features are present.

Madelon (Table 10) is a complex dataset which includes noise, flipping labels and non-linearity. Due to the latter, naive Bayes and SVM cannot obtain satisfactory results so they will not be analyzed. C4.5 obtained its highest accuracy after applying its own wrapper, as expected. It is more surprising the behavior of IB1, which obtained the highest prediction accuracy after applying methods that achieve poor indices of success. However, this fact can be explained because these methods selected a high number of redundant features. These redundant features were built by multiplying the useful features by a random matrix; therefore, they are also informative.

9 Real datasets

In order to check if the behaviors showed by the different feature selection methods can be extrapolated to the real world, two real datasets were chosen. The first one, Colon Cancer,¹ is

¹ Colon Cancer dataset is available on <http://datam.i2r.a-star.edu.sg/datasets/krbd>.

Table 15 Features selected by each algorithm on synthetic dataset SD3

	(#)	OPT(6)	Red	Irr	Suc	Accuracy (%)			
						C4.5	NB	IB1	SVM
CFS	23	4	2	17	67	64.00	80.00	73.33	70.67
Cons	9	3	0	6	50	58.67	76.00	62.67	76.00
INT	19	4	1	14	67	61.33	82.67	70.67	66.67
IG	49	4	31	14	67	62.67	65.33	65.33	73.33
ReliefF ^a	6	1	5	0	17	50.67	57.33	45.33	53.33
ReliefF ^b	20	1	9	10	17	56.00	69.33	61.33	68.00
mRMR ^a	6	1	0	5	17	62.67	62.67	66.67	65.33
mRMR ^b	20	1	0	19	17	50.67	77.33	52.00	66.67
SVM-RFE ^a	6	3	0	3	50	56.00	70.67	61.33	65.33
SVM-RFE ^b	20	4	2	14	67	49.33	85.33	70.67	82.67
FS-P ^a	6	0	0	6	0	36.00	54.67	34.67	46.67
FS-P ^b	20	1	0	19	17	38.67	61.33	45.33	56.00
$\mathcal{M}_d(\lambda = 0)^a$	6	1	5	0	17	52.00	58.67	40.00	54.67
$\mathcal{M}_d(\lambda = 0)^b$	20	3	15	2	50	45.33	57.33	50.67	54.67
$\mathcal{M}_d(\lambda = 1)^a$	6	1	5	0	17	52.00	58.67	42.67	53.33
$\mathcal{M}_d(\lambda = 1)^b$	20	1	9	10	17	54.67	66.67	60.00	70.67
W-SVM	10	1	0	9	17	48.00	61.33	61.33	81.33
W-C4.5	5	1	0	4	17	68.00	50.67	37.33	48.00

Ranker methods are tested selecting the optimal number and 20 features as cardinality

^a Selecting the optimal number of features

^b Selecting 20 features

a microarray binary dataset with 2,000 features and 62 samples, very similar to the SD family of datasets introduced in Sect. 4, which consists of detecting if a patient has colon cancer or not. The second dataset, called Optical Recognition of Handwritten Digits,² has 64 features and 5,620 samples. It consists of identifying digits from 0 to 9, so it is a multiclass dataset as well as LED dataset. Results obtained over these datasets can be seen in Tables 16 and 17. When it comes to real datasets, the only way to evaluate the feature selection performance is to compute the classification accuracy; therefore, the same four classifiers as above were employed. It also became necessary to compare the results after applying feature selection with the result when no feature reduction was performed, showed in the first row and labeled as 'Original'. It must be clarified that the high dimensionality of these datasets (either in number of samples or in number of features) prevents the use of SVM-RFE with a nonlinear kernel; therefore, it does not appear in these real experiments.

The first thing one can note when studying Table 16 is that feature selection improves classification accuracy, in some cases remarkably (see naive Bayes original and with any of the feature selection methods). This is due to the fact that most genes measured in a DNA microarray experiment are not relevant for an accurate distinction among different classes of the problem, and therefore feature selection plays a crucial role in DNA microarray analysis. It is also noticeable that SVM-RFE leads to the highest classification accuracies, and it also happened with SD datasets. This fact is not surprising since this method was introducing by the authors in the context of gene selection for cancer classification [62].

² OptDigits dataset is available on <http://archive.ics.uci.edu/ml/>.

Table 16 Results for Colon Cancer

Method	No. features	Accuracy (%)			
		C4.5	NB	IB1	SVM
Original	2000	82.26	53.23	77.42	85.48
CFS	26	87.10	85.48	83.87	85.48
Consistency	5	85.48	85.48	88.71	82.26
INTERACT	16	90.32	87.10	79.03	87.10
InfoGain	60	85.48	82.26	80.65	87.10
ReliefF	60	82.26	85.48	77.42	87.10
mRMR	60	67.74	50.00	69.35	80.64
$M_d(\lambda = 0)$	60	87.10	80.65	80.65	87.10
$M_d(\lambda = 1)$	60	87.10	80.65	80.65	87.10
SVM-RFE	60	88.71	90.32	91.94	100.00
FS-P	60	83.87	87.10	82.26	82.26
Wrapper SVM	7	79.03	87.10	83.87	91.94
Wrapper C4.5	6	95.16	74.19	85.48	80.65

Table 17 Results for Optical Digits

Method	No. features	Accuracy (%)			
		C4.5	NB	IB1	SVM
Original	64	90.71	91.33	98.61	98.38
CFS	38	90.53	91.51	98.68	98.02
Consistency	9	81.87	80.69	84.77	86.41
INTERACT	23	90.59	90.98	97.83	96.26
InfoGain	26	90.87	90.80	97.94	96.76
ReliefF	26	90.81	91.28	98.45	97.06
mRMR	26	91.41	90.82	98.10	96.98
$M_d(\lambda = 0)$	26	91.57	90.73	97.95	96.90
$M_d(\lambda = 1)$	26	91.55	90.73	97.95	96.90
SVM-RFE	26	90.62	90.55	97.76	96.78
FS-P	26	90.62	89.52	97.85	96.98
Wrapper SVM	37	90.50	91.69	98.54	98.19
Wrapper C4.5	25	91.55	90.93	98.08	96.89

With regard to Table 17, feature selection maintains or improves the classification performance reducing the number of features needed. In this case, the number of features is not so high, and we do not know a priori if there are some irrelevant features; therefore, a drastic improvement like the one over Colon Cancer dataset was not expected. Even then, for three classifiers the best result was obtained after applying feature selection and for SVM, although the best result was achieved with no feature selection, after using its own wrapper the result is very similar but reducing drastically the number of features. IB1 obtained the highest accuracy after performing feature selection and it has to be reminded that this classifier is the one with the best relation performance/index of success (see Table 12). As it happened with LED dataset with no noise (see Tables 7 and 8), CFS leads to good results (the highest accuracy obtained).

Table 18 Summary

Method	Correlation and redundancy	Non Linearity	Noise Inputs	Noise Target	No. feat >> No. samples
CFS	•	•	•	•••	••••
Consistency	•	•	•	•••	••
INTERACT	•	•	•	•••	•••
InfoGain	•	•	•	•••	•••
ReliefF	••••	•••••	•••••	•••••	••
mRMR	••••	•••	•••••	••	•
$M_d(\lambda = 0)$	••••	••	•••	•••	•••
$M_d(\lambda = 1)$	••••	••	•••	•••	•••
SVM-RFE	••••	•	•	••••	•••••
SVM-RFEnl	••••	•••••	•••	••••	—
FS-P	•••••	••	••••	••••	•
Wrapper SVM	•	•	•••	••••	••
Wrapper C4.5	••	•••	•••	•••	•••

10 Conclusions

Feature selection has been an active and fruitful field of research in machine learning. The importance of it is beyond doubt and it has proven effective in increasing predictive accuracy and reducing complexity of machine learning models. However, choosing the appropriate feature selection method for a given scenario is not an easy-to-solve question. In this paper, a review of 11 feature selection methods applied over 11 synthetic datasets and 2 real datasets was presented aimed at studying their performance with respect to several situations that can hinder the process of feature selection. The suite of synthetic datasets chosen covers phenomena such as presence of irrelevant and redundant features, noise in the data or interaction between attributes. A scenario with a small ratio between number of samples and features where most of the features are irrelevant was also tested. It reflects the problematic of datasets such as microarray data, a well-known and hard challenge in the machine learning field where feature selection becomes indispensable.

Within the feature selection field, three major approaches were evaluated: filters, wrappers and embedded methods. To test the effectiveness of the studied methods, an evaluation measure was introduced trying to reward the selection of the relevant features and to penalize the inclusion of the irrelevant ones. Besides, four classifiers were selected to measure the effectiveness of the selected features and to check if the true model was also unique.

Table 18 shows the behavior of the different feature selection methods over the different problems studied, where the larger the number of dots, the better the behavior. To decide which methods were the most suitable under a given situation, it was computed a trade-off between index of success and classification accuracy. In light of these results, ReliefF turned out to be the best option independently of the particulars of the data, with the added benefit that it is a filter, which is the model with the lowest computational cost. However, SVM-RFE with a nonlinear kernel showed outstanding results, although its computational time is in some cases prohibitive (in fact, it could not be applied over some datasets). Wrappers have proven to be an interesting choice in some domains, nevertheless they must be applied together with their own classifiers and it has to be reminded that this is the model

with the highest computational cost. In addition to this, Table 18 provides some guidelines for specific problems.

Besides an detailed analysis of the findings of this work, some cases of study were also presented in order to decide among methods that showed similar behaviors and helping to find the adequacy of them on different situations. Finally, the feature selection methods were also tested over two real datasets, demonstrating the conclusions extracted from this theoretical study over real scenarios, and proving the effectiveness of feature selection.

In light of the results presented in this work, the authors suggest the use of filters (particularly, ReliefF), since they are independent of the induction algorithm and are faster than embedded and wrapper methods, as well as having a good generalization ability. As future work, we plan to extend this study to other scenarios such as regression problems or application to image analysis.

Acknowledgments This work was supported by Spanish Ministerio de Ciencia e Innovación under project TIN 2009-02402, partially supported by the European Union ERDF. Verónica Bolón-Canedo acknowledges the support of Xunta de Galicia under *Plan I2C* Grant Program.

References

1. Kalousis A, Prados J, Hilario M (2007) Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowl Inf Syst* 12(1):95–116
2. Yang Y, Pederson JO (2003) A comparative study on feature selection in text categorization. In: *Proceedings of the 20th international conference on machine learning*, pp 856–863
3. Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. *J Mach Learn Res* 5:1205–1224
4. Provost F (2000) Distributed data mining: scaling up and beyond. In: Kargupta H, Chan P (eds) *Advances in distributed data mining*. Morgan Kaufmann, San Francisco
5. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. *J Mach Learn Res* 3:1157–1182
6. Guyon I, Gunn S, Nikravesh M, Zadeh L (2006) *Feature extraction, foundations and applications*. Springer, Heidelberg
7. Yu L, Liu H (2004) Redundancy based feature selection for microarray data. In: *Proceedings of the 10th ACM SIGKDD conference on knowledge discovery and data mining*, pp 737–742
8. Bolón-Canedo V, Sánchez-Marño N, Alonso-Betanzos A (2011) Feature selection and classification in multiple class datasets: an application to KDD Cup 99 dataset. *J Expert Syst Appl* 38(5):5947–5957
9. Lee W, Stolfo SJ, Mok KW (2000) Adaptive intrusion detection: a data mining approach. *Artif Intell Rev* 14(6):533–567
10. Forman G (2003) An extensive empirical study of feature selection metrics for text classification. *J Mach Learn Res* 3:1289–1305
11. Gomez JC, Boiy E, Moens MF (2011) Highly discriminative statistical features for email classification. *Knowl Inf Syst*. doi:10.1007/s10115-011-0403-7
12. Egozi O, Gabrilovich E, Markovitch S (2008) Concept-based feature generation and selection for information retrieval. In: *Proceedings of the twenty-third AAAI conference on artificial intelligence*, pp 1132–1137
13. Dy JG, Brodley CE, Kak AC, Broderick LS, Aisen AM (2003) Unsupervised feature selection applied to content-based retrieval of lung images. *IEEE Trans Pattern Anal Mach Intell* 25(3):373–378
14. Saari P, Eerola T, Lartillot O (2011) Generalizability and simplicity as criteria in feature selection: application to mood classification in music. *IEEE Trans Audio Speech Lang* 19(6):1802–1812
15. Dash M, Liu H (1997) Feature selection for classification. *Intell Data Anal* 1:131–156
16. Zhang Y, Ding C, Li T (2008) Gene selection algorithm by combining relief and mrmr. *BMC Genomics* 9(Suppl 2):S27. doi:10.1186/1471-2164-9-S2-S27
17. Abraham R Dimensionality reduction through bagged feature selector for medical data mining
18. Peng Y, Wu Z, Jiang J (2010) A novel feature selection approach for biomedical data classification. *J Biomed Inf* 43(1):15–23

19. El Akadi A, Amine A, El Ouardighi A, Aboutajdine D (2011) A two-stage gene selection scheme utilizing MRMR filter and GA wrapper. *Knowl Inf Syst* 26(3):487–500
20. Vainer I, Kraus S, Kaminka GA, Slovin H (2010) Obtaining scalable and accurate classification in large-scale spatio-temporal domains. *Knowl Inf Syst*. doi:[10.1007/s10115-010-0348-2](https://doi.org/10.1007/s10115-010-0348-2)
21. Tuv E, Borisov A, Runger G (2009) Feature selection with ensembles, artificial variables, and redundancy elimination. *J Mach Learn Res* 10:1341–1366
22. Sun Y, Li J (2006) Iterative RELIEF for feature weighting. In: *Proceedings of the 21st international conference on machine learning*, pp 913–920
23. Sun Y, Todorovic S, Goodison S (2008) A feature selection algorithm capable of handling extremely large data dimensionality. In: *Proceedings of the 8th SIAM international conference on data mining*, pp 530–540
24. Chidlovskii B, Lecerf L (2008) Scalable feature selection for multi-class problems. *Mach Learn Knowl Discov Databases* 5211:227–240
25. Loscalzo S, Yu L, Ding C (2009) Consensus group based stable feature selection. In: *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining*, pp 567–576
26. Saeys Y, Abeel T, Peer Y (2008) Robust feature selection using ensemble feature selection techniques. In: *Proceedings of the European conference on machine learning and knowledge discovery in databases—part II*, pp 313–325
27. Bolon-Canedo V, Sanchez-Marono N, Alonso-Betanzos A (2012) An ensemble of filters and classifiers for microarray data classification. *J Pattern Recognit* 45:531–539
28. Sun Y, Babbs CF, Delp EJ (2005) A comparison of feature selection methods for the detection of breast cancers in mammograms: adaptive sequential floating search vs. genetic algorithm. In: *Proceedings of the IEEE conference on engineering in medicine and biology society*, pp 6532–6535
29. Ramaswami M, Bhaskaran R (2009) A study on feature selection techniques in educational data mining. *Int J Adv Comput Sci Appl* 2(1):7–11
30. Liu H, Liu L, Zhang H (2008) Feature selection using mutual information: an experimental study. In: *Proceedings of the 10th Pacific rim international conference on artificial intelligence: trends in artificial intelligence*, pp 235–246
31. Beretta L, Santaniello A (2011) Implementing ReliefF filters to extract meaningful features from genetic lifetime datasets. *J Biomed Inf* 44(2):361–369
32. Zhang ML, Peña JM, Robles V (2009) Feature selection for multi-label naive Bayes classification. *J Inf Sci* 179(19):3218–3229
33. Perner P, Apte C (2000) Empirical evaluation of feature subset selection on a real-world data set. In: *Proceedings of conference on principles of data mining and knowledge discovery*, pp 575–580
34. Victo Sudha G, Cyril Raj V (2011) Review on feature selection techniques and the impact of SVM for cancer classification using gene expression profile. *Int J Comput Sci Eng Survey*. doi:[10.5121/ijces.2011.2302](https://doi.org/10.5121/ijces.2011.2302)
35. Li T, Zhang C, Ogiwara M (2004) A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *J Bioinf* 20(15):2429–2437
36. Hua J, Tembe W, Dougherty E (2009) Performance of feature-selection methods in the classification of high-dimension data. *J Pattern Recognit* 42(3):409–424
37. Bontempi G, Meyer PE (2010) Causal filter selection in microarray data. In: *Proceedings of the 27th international conference on machine learning*, pp 95–102
38. Aliferis CF, Statnikov A, Tsamardinos I, Mani S, Koutsoukos XD (2010) Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: algorithms and empirical evaluation. *J Mach Learn Res* 11:171–234
39. Byeon B, Rasheed K (2008) Simultaneously removing noise and selecting relevant features for high dimensional noisy data. In: *Proceedings of the 2008 seventh international conference on machine learning and applications*, pp 147–152
40. Yang SH, Hu BG (2008) Efficient feature selection in the presence of outliers and noises. In: *Proceedings of the 4th Asia information retrieval conference on information retrieval technology*, pp 184–191
41. Guyon I, Bitter HM, Ahmed Z, Brown M, Heller J (2005) Multivariate non-linear feature selection with kernel methods. *Stud Fuzz Soft Comput* 164:313–326
42. Liu H, Yu L (2005) Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans Knowl Data Eng* 17(4):491–502
43. Molina LC, Belanche L, Nebot A (2002) Feature selection algorithms: a survey and experimental evaluation. In: *Proceedings of the 2002 IEEE international conference on data mining*, pp 306–313
44. Doak J (1992) An evaluation of feature selection methods and their application to computer security. Technical report CSE-92-18, University of California, Department of Computer Science

45. Jain AK, Zongker D (2002) Feature selection evaluation, application, and small sample performance. *IEEE Trans Pattern Anal Mach Intell* 19(2):153–158
46. Kudo M, Sklansky J (1997) A comparative evaluation of medium and large-scale feature selectors for pattern classifiers. In: *Proceedings of the 1st international workshop on statistical techniques in pattern recognition*, pp 91–96
47. Liu H, Setiono R (1998) Scalable feature selection for large sized databases. In: *Proceedings of the 4th world conference on machine learning*, pp 101–106
48. Thrun S, et al (1991) The MONK's problems: a performance comparison of different learning algorithms. Technical report CS-91-197, CMU
49. Belanche LA, González FF, Review and evaluation of feature selection algorithms in synthetic problems. <http://arxiv.org/abs/1101.2320> (Last access: Nov 2011)
50. Liu H, Setiono R (2002) Chi2: feature selection and discretization of numeric attributes. In: *Proceedings of the 7th international conference on tools with artificial intelligence*, pp 388–391
51. Sánchez-Marofío N, Alonso-Betanzos A, Tombilla-Sanromán M (2007) Filter methods for feature selection: a comparative study. In: *Proceedings of the 8th international conference on intelligent data engineering and automated learning*, pp 178–187
52. Witten IH, Frank E (2005) *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco. <http://www.cs.waikato.ac.nz/ml/weka/> (Last access: Nov 2011)
53. The Mathworks, Matlab Tutorial (1998). http://www.mathworks.com/academia/student_center/tutorials/ (Last access: Nov 2011)
54. Hall MA (1999) Correlation-based feature selection for machine learning. PhD thesis, University of Waikato, Hamilton
55. Dash M, Liu H (2003) Consistency-based search in feature selection. *J Artif Intell* 151(1–2):155–176
56. Zhao Z, Liu H (1991) Searching for interacting features. In: *Proceedings of the international joint conference on artificial intelligence*, pp 1156–1167
57. Hall MA, Smith LA (1998) Practical feature subset selection for machine learning. *J Comput Sci* 98:4–6
58. Kononenko I (1994) Estimating attributes: analysis and extensions of RELIEF. In: *Proceedings of the European conference on machine learning*, pp 171–182
59. Kira K, Rendell L (1992) A practical approach to feature selection. In: *Proceedings of the 9th international workshop on machine learning*, pp 249–256
60. Peng H, Long F, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans Pattern Anal Mach Intell* 27(8):1226–1238
61. Seth S, Principe JC (2010) Variable selection: a statistical dependence perspective. In: *Proceedings of the international conference of machine learning and applications*, pp 931–936
62. Guyon I, Weston J, Barnhill SMD, Vapnik V (2002) Gene selection for cancer classification using support vector machines. *J Mach Learn* 46(1–3):389–422
63. Rakotomamonjy A (2003) Variable selection using SVM-based criteria. *J Mach Learn Res* 3:1357–1370
64. Mejía-Lavalle M, Sucar E, Arroyo G (2006) Feature selection with a perceptron neural net. In: *Proceedings of the international workshop on feature selection for data mining*, pp 131–135
65. Mamitsuka H (2006) Query-learning-based iterative feature-subset selection for learning from high-dimensional data sets. *Knowl Inf Syst* 9(1):91–108
66. Quinlan JR (1993) *C4.5: programs for machine learning*. Morgan Kaufmann, San Francisco
67. Rish I (2001) An empirical study of the naive Bayes classifier. In: *Proceedings of IJCAI-01 workshop on empirical methods in artificial intelligence*, pp 41–46
68. Aha DW, Kibler D, Albert MK (1991) Instance-based learning algorithms. *J Mach Learn* 6(1):37–66
69. Shawe-Taylor J, Cristianini N (2000) *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, Cambridge
70. Langley P, Iba W (1993) Average-case analysis of a nearest neighbor algorithm. In: *Proceedings of the 11th international conference on artificial intelligence*, vol 13, pp 889–894
71. Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T, Vapnik V (2001) Feature selection for SVMs. *J Adv Neural Inf Process Syst* 13:668–674
72. John GH, Kohavi R, Pfleger K (1994) Irrelevant features and the subset selection problem. In: *Proceedings of the 11th international conference on machine learning*, pp 121–129
73. Kim G, Kim Y, Lim H, Kim H (2010) An MLP-based feature subset selection for HIV-1 protease cleavage site analysis. *J Artif Intell Med* 48:83–89
74. Breiman L, Friedman J, Olshen R, Stone C (1984) *Classification and regression trees*. Wadsworth International Group, Belmont
75. Zhu Z, Ong YS, Zurada JM (2010) Identification of full and partial class relevant genes. *IEEE Trans Comput Biol Bioinf* 7(2):263–277

76. Díaz-Uriarte R, Andrés A (2006) Gene selection and classification of microarray data using random forest. *J Bioinf* 7(1):1–13
77. de Kohavi R, John GH (1997) Wrappers for feature subset selection. *J Artif Intell* 97(1–2):273–324
78. Brown MPS, Grundy WN et al (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc Natl Acad Sci USA* 97(1):262–267

Author Biographies



Verónica Bolón-Canedo received her B.S. degree in Computer Science from University of A Coruña, Spain, in 2008. She received her M.S. degree in 2010 and is currently a Ph.D. student in the Department of Computer Science at the same university. Her research interests include machine learning and feature selection.



Noelia Sánchez-Marroño received a Ph.D. degree for her 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 current research areas include agent-based modeling, machine learning and feature selection.



Amparo Alonso-Betanzos received the Ph.D. degree for her 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, machine learning and feature selection.