Neural Computing and Applications (2020) 32:5951–5973
https://doi.org/10.1007/s00521-019-04082-3

ORIGINAL ARTICLE

Ensemble feature selection for high-dimensional data: a stability analysis across multiple domains

Barbara Pes[1]

Received: 26 March 2018 / Accepted: 12 February 2019 / Published online: 25 February 2019
© The Author(s) 2019

Abstract
Selecting a subset of relevant features is crucial to the analysis of high-dimensional datasets coming from a number of application domains, such as biomedical data, document and image analysis. Since no single selection algorithm seems to be capable of ensuring optimal results in terms of both predictive performance and stability (i.e. robustness to changes in the input data), researchers have increasingly explored the effectiveness of "ensemble" approaches involving the combination of different selectors. While interesting proposals have been reported in the literature, most of them have been so far evaluated in a limited number of settings (e.g. with data from a single domain and in conjunction with specific selection approaches), leaving unanswered important questions about the large-scale applicability and utility of ensemble feature selection. To give a contribution to the field, this work presents an empirical study which encompasses different kinds of selection algorithms (filters and embedded methods, univariate and multivariate techniques) and different application domains. Specifically, we consider 18 classification tasks with heterogeneous characteristics (in terms of number of classes and instances-to-features ratio) and experimentally evaluate, for feature subsets of different cardinalities, the extent to which an ensemble approach turns out to be more robust than a single selector, thus providing useful insight for both researchers and practitioners.

Keywords Feature selection · Stability of feature selection algorithms · Ensemble approaches · High-dimensional data analysis

1 Introduction

The dimensionality of datasets used in machine learning applications has increased exponentially in recent years. As extensively discussed in the literature [1, 2], the high dimensionality of data introduces a number of challenges for learning algorithms, due to the very large search space, and makes it difficult to extract valuable knowledge about a given domain of interest. In this scenario, feature selection has become almost indispensable since it can eliminate irrelevant and redundant information, thus reducing the dimensionality as well as the complexity of the original problem, with significant benefits in terms of computational efficiency, model interpretability and data understanding.

Many authors have investigated the strengths and weaknesses of the available feature selection techniques [3–6], but finding the "best" approach for a given task remains difficult. In particular, recent research in the field [7, 8] has highlighted that the existing algorithms are often deficient in terms of *stability*, i.e. robustness with respect to changes in the input data. This is a very important issue when the selected features are exploited for gaining knowledge on the underlying domain. Indeed, if the outcome of the selection process is too sensitive to variations in the set of training instances, with limited reproducibility of results, it can compromise the confidence of users and domain experts and the subsequent exploitation of the results in real-world applications. Moreover, as observed in [9], the robustness of feature selection may have practical implications for distributed applications where the

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s00521-019-04082-3) contains supplementary material, which is available to authorized users.

✉ Barbara Pes
pes@unica.it

[1] Dipartimento di Matematica e Informatica, Università degli Studi di Cagliari, Via Ospedale 72, 09124 Cagliari, Italy

algorithm should produce stable results across multiple data sources.

With the aim of achieving a better trade-off between predictive performance and stability, new and more sophisticated feature selection approaches are increasingly being explored [2, 10]. In particular, the *ensemble paradigm* has been investigated [11, 12] as a promising framework for improving the robustness of the selection process, especially in high-dimensional and low sample size settings, where the extraction of stable feature subsets is intrinsically more difficult [7]. The basic idea is to jointly exploit the strengths of different selectors, overcoming at the same time their weaknesses, similar to the logic of *ensemble classification*, that leverages multiple models [13] and has proved to be successful in a large variety of applications [14–17].

The ensemble selection techniques appeared in recent literature can be broadly categorized into two main groups [18, 19]: *functionally heterogeneous* approaches, which involve applying different selection algorithms to the same dataset, and *functionally homogeneous* approaches, where the same selection algorithm is applied to different perturbed versions of the original data (similar to *bagging* and *boosting* techniques [20] in the context of multi-classifier systems). In both cases, different outputs are produced which are subsequently combined to generate a single feature subset capable of better approximating, hopefully, the "optimal" solution for the problem at hand.

The focus of this work is on homogeneous ensembles which have turned out to be especially promising for handling the stability issue [11, 21] but, so far, have been experimented in a limited number of settings, particularly with biological and genomic data (mostly in the context of binary classification tasks) and in conjunction with specific selection approaches. To provide a more comprehensive evaluation of this ensemble technique, we present an empirical study that encompasses different kinds of selection algorithms (filters and embedded methods, univariate and multivariate approaches) and different application domains (from the classification of genomic and proteomic data to the analysis of texts, images, videos as well as specific kinds of signals, such as voice recordings and ECG recordings).

Specifically, we consider seven algorithms that are representatives of quite different heuristics and, for each of them, implement an "ensemble version" whose output is a feature subset resulting from the aggregation of the algorithm's outcomes on different perturbed versions of the available data. Considering different levels of data perturbation as well as feature subsets of different cardinalities, we systematically compare the robustness of this ensemble version to that of the original "simple" algorithm. This analysis has been conducted on 18 classification benchmarks with quite heterogeneous characteristics (in terms of numbers of features and instances, instances-to-features ratio, number of classes and class distribution, presence of noise), so as to obtain useful insight about the real potential and the concrete applicability of the ensemble selection paradigm across multiple, diversified, real-world scenarios.

The experimental results confirm that the considered approach can actually be useful to face the stability issue, though to an extent that may depend both on the specific selection algorithm and on the peculiar characteristics of the domain.

The remainder of this work is structured as follows. Section 2 provides background concepts and summarizes previous research in the field. Section 3 illustrates all the methods relevant to our study, i.e. the considered selection algorithms, the specific ensemble strategy here adopted, and the methodology used for evaluating both the stability and the predictive power of the selected feature subsets. Section 4 describes the 18 datasets used as benchmarks, the specific settings of the experiments, as well as the experimental results and their implications. Finally, Sect. 5 gives the concluding remarks.

2 Background and related work

As previously highlighted, feature selection plays a very important role in the analysis of high-dimensional data. This section provides an overview of best practices and research issues in the field, with special emphasis on classification tasks (which are the target of our study), though most of the concepts and ideas here discussed may also be applied in the context of regression problems [22, 23]. Specifically, we start from a general categorization of feature selection algorithms in Sect. 2.1. Next, Sect. 2.2 focuses on the rationale of ensemble feature selection and Sect. 2.3 discusses its applicability to real-world scenarios.

2.1 Foundations of feature selection

Feature selection, also known as *attribute selection* or *variable subset selection*, is the process of detecting the most relevant features for the problem at hand [1]. Many definitions have been proposed in the literature to formalize the concept of feature relevance and to quantify the degree of relevance [24], with no clear and well-established guidelines to choose the most suitable approach for a specific situation.

In the context of classification tasks, where feature selection essentially aims at extracting a subset of attributes highly correlated with the target class, the available

selection methods can be broadly categorized into three groups [4]:

- *Filters*, which carry out the selection process as a pre-processing step without interacting with the learning algorithm used at the model construction stage. The selection criteria rely on the general characteristics of the training data and may involve the *individual evaluation* of single features, whose correlation with the target is measured through proper statistic or entropic approaches, or the *evaluation of subsets* of features, where the mutual correlation among the features is also taken into account to minimize redundancy.
- *Wrappers*, which search for the feature subset that can optimize the performance of a given classifier. Then, the learning algorithm itself is used to evaluate the merit of each candidate subset of features, with the computational burden of inducing a model from each candidate subset, besides the intrinsic cost of the candidate construction process that depends on the search strategy used to explore the original feature space.
- *Embedded approaches*, where the selection process relies on the intrinsic capacity of certain classification algorithms to assign weights to the features, without a systematic search through different candidate subsets.

Due to their computational efficiency, filter methods are often preferred in the context of high-dimensional problems; indeed, the fine search involved by the wrappers, potentially capable of leading to better feature subsets, may be practically infeasible depending on the size of the search space. In turn, embedded approaches often provide a suitable trade-off between final predictive performance and computational cost.

With the aim of exploiting the strengths of different methods, overcoming at the same time their weaknesses, new and more sophisticated approaches are constantly been explored [2] that (*i*) rely on different algorithms at different stages of the selection process (e.g. initially reducing the problem dimensionality by a filter and then further refining the search by a wrapper) or (*ii*) combine, at a given stage of the selection process, the outcome of different selectors (e.g. different filters). This last approach, which has shown to be promising to improve the robustness of the selection process [7], leverages the ideas and the methodologies developed in the fruitful field of *ensemble learning*, as discussed in the next subsection.

2.2 Exploiting the ensemble learning paradigm for feature selection

Ensemble learning, also referred as *ensemble classification,* is a well-established paradigm [14] that relies on the decisions of multiple models to achieve a better predictive performance. As observed in [25], a crucial factor for the success of this approach is the degree of *diversity* among the models that are exploited in the ensemble. This diversity can be obtained in different ways: (*i*) by manipulating the input data to create a number of diversified training sets from which deriving, by a given induction algorithm, different models; (*ii*) by applying different learning algorithms (or differently parameterized versions on the same algorithm) to a given set of input records; (*iii*) by hybrid approaches where diversity is injected both at the data level and at the algorithm level. Once the different models (*ensemble components*) are induced, their predictions are properly combined in some way (e.g. by weighted or unweighted voting) to classify new records.

A similar ensemble logic has been recently experimented in the context of feature selection [18]. The rationale is to obtain a more reliable set of features by combining the outputs of a number of *base selectors*, which should be sufficiently diverse from each other to convey non-overlapping, and hence complementary, information on the considered domain. The base selectors can be functionally homogeneous, if they leverage the same algorithm, and in this case, their diversity is achieved by varying the training data (e.g. through resampling techniques). Alternatively, functionally heterogeneous selectors, i.e. different selection algorithms, can be exploited on the same data but hybrid approaches are also possible where both the data and the selection algorithms are varied.

In turn, the results produced by the different selectors can be combined in different ways. In particular, in the context of high-dimensional problems, the ensemble components usually exploit a *feature ranking* procedure (i.e. assign a score to each single feature), and the combination of the results is then formalized as a *rank aggregation* problem [26]. A number of aggregation functions have been proposed for this purpose, and it is not clear which of them should be chosen for a specific task, but simple approaches such as mean-based aggregation seem to be suitable in most cases [27–29]. Strategies that take feature interactions into account at the aggregation stage are investigated in [30]. More generally, the combination of multiple feature subsets into a single "ensemble subset" can be performed according to mixing strategies such those discussed in [31].

2.3 Application of ensemble feature selection to real-world problems

Theoretically, ensemble feature selection is potentially superior to standard feature selection techniques in many aspects. Indeed, it has been observed [11] that single selection algorithms tend to select locally optimal solutions in the search space of candidate subsets, while the ensemble approach has more chance to reach a better approximation to the best solution by "averaging" different hypotheses. Furthermore, it is possible that multiple subsets are capable of discriminating the data equally well, and different subsets may be selected under different settings, with limited stability and reproducibility of results. In these contexts, the ensemble approach may give a more stable outcome [10].

In practice, however, the real potential of ensemble feature selection is still object of debate and no clear guidelines are available for choosing the best ensemble strategy for a given problem. As regards the functionally heterogeneous approaches, a number of experimental studies have shown that, if properly built and tuned, they can outperform single selection methods in terms of final predictive performance. For example, micro-array data classification can benefit from the combination of different selection methods, as reported in [12, 32, 33]. In the context of text categorization, the study proposed in [34] explores the aggregation of three popular feature ranking techniques. A comprehensive evaluation of different ensembles, composed of two to 18 individual selectors, is conducted by Wang et al. [35] on real-world software measurement data. On the other hand, limited evidence exists on the potential of this approach in terms of selection stability [12, 36]. Further, in case of heterogeneous ensembles, the choice of how many and which methods to combine can be difficult, being highly dependent on the specific characteristics of the data at hand, and often requires an expensive phase of preliminary trials.

Actually, when the stability of the selection outcome is a primary concern (especially if the high dimensionality is coupled with a comparatively small number of instances), the use of homogeneous ensembles has been presented as an effective strategy [10]. In this case, all the base selectors (whose number can be very large compared to the size of heterogeneous ensembles) exploit the same algorithm but are trained on different sampled versions of the original data. Indeed, the outstanding studies of Saeys et al. [11] and Abeel et al. [21] have shown that this strategy can significantly improve the stability of standard selection techniques in the context of biomarker discovery from gene expression and mass spectrometry data. However, analysing different datasets from the same domain, other authors [37] have observed that the beneficial impact of the ensemble approach is limited to certain selection methods, at least for the considered subset size (100 genes). The effectiveness of this ensemble strategy on high-dimensional biomedical data is also evaluated by further studies [38, 39], which remark the need for wider experiments on different types of data. Besides the biological domain, Woznica et al. [30] have also investigated homogeneous ensembles in the context of text categorization tasks, but no work exists, to the best of our knowledge, that extensively evaluates the robustness of ensemble feature selection across multiple real-world scenarios and multiple learning settings (e.g. different numbers of classes and different instances-to-features ratios).

3 Methods

Encompassing different application domains, our study aims to evaluate the extent to which homogeneous ensembles can really improve the robustness of the selection process, ensuring a satisfactory trade-off between selection stability and predictive accuracy. Indeed, as previously observed, they have been so far evaluated in a limited number of classification tasks (and often only for feature subsets of a given size), leaving unanswered important questions about their large-scale applicability and utility. To give a contribution to the field, we conducted extensive experiments to compare the usual "simple" version of seven selection methods, chosen to represent different heuristics and selection approaches (as explained in Sect. 3.1), to their "ensemble" version, implemented according to the strategy detailed in Sect. 3.2. The comparison has been carried out in terms of both stability and predictive performance, according to the methodology presented in Sect. 3.3.

3.1 Simple ranking approach

In general, given a set of training instances, each described by N features, the output of feature selection can be given in the form of

(i) A *weighting* of the N features, i.e. a weight is assigned to each feature based on a suitable relevance criterion (e.g. the degree of correlation between the feature and the target class);
(ii) A *ranking* of the N features, i.e. the features are ordered based on their relevance, from the most important to the least important (of course, a feature weighting can be simply converted to a feature ranking by sorting the weights);

(iii) A *subset* of the N features, which can be chosen based on a subset evaluation strategy (as in the wrapper approaches) or grouping together features that have individually shown to be highly predictive (e.g. a weighting/ranking of the features can be converted to a feature subset by setting a suitable threshold).

According to common practice in high-dimensional data analysis, we adopt here a ranking-based selection framework which is flexible enough to encompass the use of filter methods, which weigh the features based on their correlation with the target class (such as Chi squared [26] and *information gain* [40]), and embedded methods, which leverage the features' weights induced by a proper classification algorithm (such as SVM-based selectors [41]). In both cases, features' weights can be used to build a *ranked list* where the features appear in decreasing order of relevance, from the most important (rank 1) to the least important (rank N). In turn, this list can be cut at a proper threshold point (th) to produce a subset of highly discriminative features, as schematized in Fig. 1. Though unable to automatically determine the optimal subset size (indeed, the choice of the "best" threshold value often requires a fine tuning, besides a good knowledge of the underlying domain), this approach is capable of removing irrelevant and noisy features in a simple and cost-effective way, especially when the size of the search space makes impractical the direct adoption of wrapper-based search strategies.

In what follows, we give a brief description of the seven ranking methods included in our experimental study:

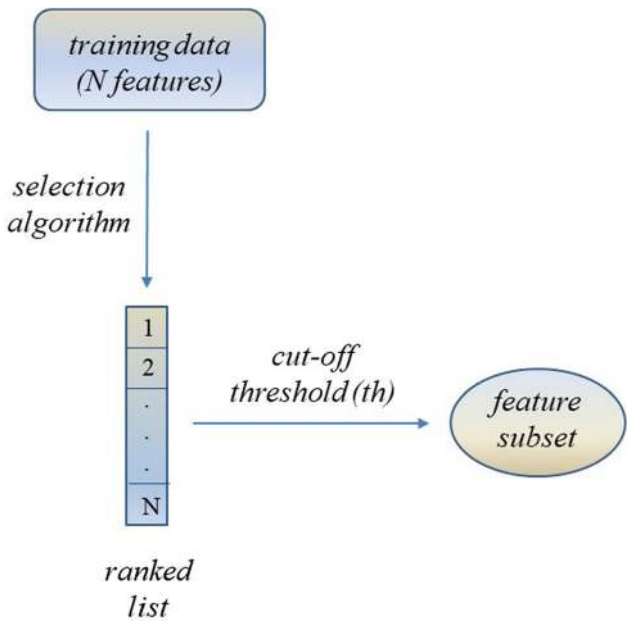

Fig. 1 Simple ranking approach

- Chi-squared (χ^2) quantifies the relevance of each single feature by measuring its Chi-squared statistic with respect to the class: the larger the Chi-squared, the higher the importance of the feature for the task at hand [26].
- *Information Gain (InfoG)* leverages the concept of entropy [40]. Specifically, a weigh for each feature is derived by evaluating the extent to which the class entropy decreases (and, hence, the uncertainty in the class prediction decreases) when the value of that feature is known.
- *Gain Ratio (GainR)*, in turn, relies on the concept of entropy. Essentially, it differs from *InfoG* for a normalization factor that tries to compensate for the information gain's bias towards features with more values [40].
- *OneR* ranks the features using the rule-based classification algorithm proposed by Holte [42]. Basically, for each feature, the algorithm finds a simple rule by determining the majority class for each feature's value. Then, the accuracy of each rule is evaluated, and the features are ordered according to the quality of the corresponding rules.
- *ReliefF* evaluates the features according to their ability to differentiate between data points that are near to each other in the attribute space [43]. Essentially, a sample instance is drawn from the dataset, and the values of its features are compared to those of the instance's nearest neighbours, one (or more) for each class. A relevance score is then assigned to each feature based on the assumption that a "good" feature should have the same value for instances from the same class and different values for instances of different classes. Iteratively, a suitable number of sample instances are considered, and the features' scores are updated accordingly. This method is often implemented by weighting the nearest neighbours by their distance: in our study we employed this weighted version of the algorithm.
- *SVM-AW* exploits the embedded capability of a linear SVM classifier to assign a weight to each feature, based on the contribution the feature gives to the decision function induced by the classifier [40]; the absolute value of this weight (AW) is used to build the ranked list of the features.
- *SVM-RFE*, in turn, leverages the weights assigned to the features by a linear SVM classifier. Once the features are weighted, a recursive feature elimination (RFE) strategy is adopted that consists in iteratively removing the features with the lowest weights and repeating the classifier construction (and the weighting process) on the remaining features [44]. In our experiments, we evaluated two versions of this method: *SVM-RFE10*, where the percentage of features removed at each

iteration is 10%, and *SVM-RFE50*, where this percentage is 50%.

The above selection algorithms are representatives of quite different (and popular) heuristics. In particular, as previously mentioned, some of them (χ^2, *InfoG*, *GainR*, *ReliefF*) can be categorized as filters since do not leverage any classifier, while the others (*OneR*, *SVM-AW*, *SVM-RFE*) exploit the embedded capability of certain classification algorithms to assign weights to the features. From another point of view, the adopted selection algorithms can be distinguished into *univariate* approaches (χ^2, *InfoG*, *GainR*, *OneR*), which evaluate each feature independently from the others, and *multivariate* approaches (*ReliefF*, *SVM-AW*, *SVM-RFE*), which take into account the interdependencies among the features. This twofold categorization of the adopted selection algorithms is summarized in Table 1.

3.2 Ensemble ranking approach

As mentioned in Sect. 2, the applicability of the ensemble learning paradigm to feature selection has been increasingly investigated in recent years, especially in the context of ranking-based frameworks, as the one here considered.

In particular, as summarized in Fig. 2, functionally homogeneous ensembles involve a number of base selectors which exploit the same core algorithm but are trained on different versions of the original data. A common strategy is the adoption of sampling techniques such as *bagging* or *bootstrap aggregating* [40], i.e. a number B of samples (*bootstraps*) are drawn with replacement from the input data according to a uniform probability distribution. Due to the diversity of the training data, the base selectors can then produce different outputs to be subsequently combined into a single final outcome.

Specifically, we focus here on constructing ensembles of *rankers*, i.e. each base selector provides as output a ranked list where the original features are ordered according to their relevance. The different ranked lists are then combined into a single *ensemble list* using a suitable aggregation function that assigns each feature an "overall score" based on the feature's position (rank) in the original lists.

Table 1 Categorization of the adopted selection algorithms: *filter* versus *embedded* methods and *univariate* versus *multivariate* approaches

	Univariate	Multivariate
Filter	χ^2, InfoG, GainR	ReliefF
Embedded	OneR	SVM-AW, SVM-RFE

More formally, let L_k be the ranked list resulting from the application of a given selection algorithm to the kth bootstrap sample ($k = 1,\ldots, B$). For each of the original features f_i ($i = 1,\ldots, N$), an overall score is then calculated as follows:

$$\text{score}_i = \text{score}(f_i) = \text{aggr}(r_{i1}, r_{i2}, \ldots, r_{iB}),$$

where r_{ik} is the rank of the ith feature in the kth ranked list and aggr is a suitable aggregation function. Based on their overall scores, the features are then ordered, from the most important to the least important, in the final ensemble list. In turn, this list can be cut at a proper threshold point (th) to obtain an *ensemble subset* of highly discriminative features.

In this study, the overall score of each feature is simply obtained by averaging its rank across all the B lists produced by the base selectors: the smaller the average rank, the most important the feature is. Though more sophisticated aggregation strategies could be adopted, they require higher execution times and often result in a limited (or null) improvement in terms of selection stability and/or predictive performance [27, 28]. In turn, our previous research on high-dimensional genomic data showed that different aggregation functions often produce comparable results [29].

3.3 Evaluating stability in conjunction with predictive performance

Feature selection stability has been a neglected issue until few years ago, and only recently, a number of methodologies and metrics have been proposed [9, 10, 45–47] for the experimental assessment of stability. In most of the studies, however, stability is not measured in conjunction with predictive performance but in independent experiments, thus making difficult to capture the extent to which a given selection approach impacts on the trade-off between stability and accuracy, which are both fundamental for real-world applications. (Indeed, stable but not accurate solutions would not be meaningful; on the other hand, accurate but not stable results could have limited utility for practitioners.)

The methodology here adopted involves the evaluation of both the aspects, i.e. the stability level of the selected features and the predictive performance of the classification models built from them. Specifically, given the input dataset, we repeatedly perform random sampling (without replacement) to create T different training sets, each containing a fraction X of the original records. For each training set, a test set is also formed using the remaining fraction (1-X) of the instances.

The feature selection process is then carried out in a twofold way:

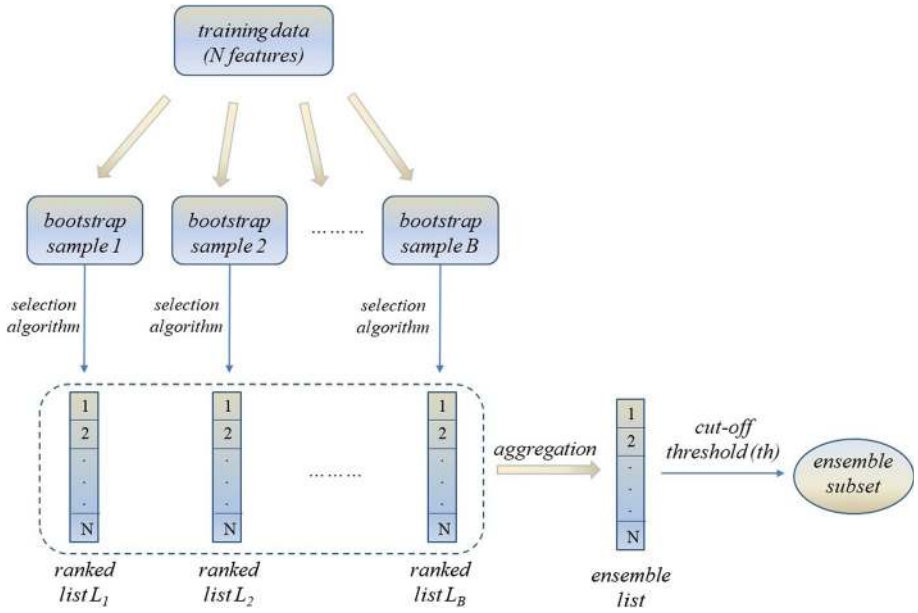

Fig. 2 Ensemble ranking approach

(i) *Simple ranking approach.* A given selection method is applied separately on each training set (as shown in Fig. 1) to obtain the corresponding ranked list and the resulting feature subset, with the first *th* features. This results in T different feature subsets (referred as simple subsets in what follows).

(ii) *Ensemble ranking approach*. An ensemble version of the same selection method is implemented according to the strategy described in Sect. 3.2, i.e. each training set is in turn sampled to construct B bootstraps (see Fig. 2) from which to derive the ensemble list (which is the aggregation of B ranked lists). An ensemble subset, with the first th features of the ensemble list, is hence derived for each training set, resulting in a total of T ensemble subsets.

In both simple and ensemble settings, a classification model is built on each training set using the selected feature subset, and the model performance is measured (through suitable metrics which can be chosen in dependence on the characteristics of the data at hand) on the corresponding test set. By averaging the classification performance of the resulting T models, we can obtain an estimate of the effectiveness of the applied selection approach (simple or ensemble) in identifying the most discriminative features.

At the same time, the stability of both simple and ensemble ranking is measured by performing a similarity analysis on the feature subsets derived from the T training sets. Indeed, the more similar they are, the more stable the selection approach. In more detail, for each pair of subsets S_i and S_j ($i, j, = 1,\ldots, T$), we use a proper consistency index [48] to quantify their degree of similarity:

$$\mathrm{sim}_{ij} = \frac{|S_i \cap S_j| - \mathrm{th}^2/N}{\mathrm{th} - \mathrm{th}^2/N}$$

where th is the size of the subsets (corresponding to the cut-off threshold) and N the overall number of features. The similarity sim_{ij} simply expresses the degree of overlapping between the subsets, i.e. the fraction of features which are common to them ($|S_i \cap S_j|$/th), with a correction term reflecting the probability that a feature is included in both subsets simply by chance. The need for this correction, which increases as the subset size approaches the total number of features, is experimentally demonstrated for example in [49].

The resulting similarity values are then averaged over all pair-wise comparisons, in order to evaluate the overall degree of similarity among the T subsets and hence the stability level of the selection process:

$$\text{stability} = \mathrm{sim}_{\mathrm{avg}} = \frac{2}{T(T-1)} \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \mathrm{sim}_{ij}$$

This analysis is performed separately for the simple and the ensemble subsets, and in both the settings, different values of the cut-off threshold (i.e. different subset sizes) are considered. Indeed, the "optimal" cut-off value may be dependent on the characteristics of the data, as well as on the ranking criteria adopted, so it is worth exploring the robustness of the selection process within a wide range of threshold values.

Note that we do not provide an automatic way to find that optimal cut-off, which is still an open problem [50],

but investigate the patterns of both stability and predictive performance for different subset sizes, in order to provide useful guidance for knowledge discovery applications, as discussed in what follows.

4 Experimental analysis and discussion

As explained in Sect. 3, we empirically evaluated the performance of different ranking methods (χ^2, *InfoG*, *GainR*, *OneR*, *ReliefF*, *SVM-AW*, *SVM-RFE10*, *SVM-RFE50*) in their simple and ensemble version. A detailed description of this study is here presented; in particular, the datasets used as benchmarks and the settings of the experiments are described in Sect. 4.1, while the experimental results are illustrated in Sects. 4.2 (stability analysis) and 4.3 (predictive performance analysis); a further discussion of the results is finally given in Sect. 4.4.

4.1 Datasets and settings of the experiments

Our study focuses on high-dimensional domains where the automatic extraction of meaningful and stable feature subsets is still an open issue. The intrinsic difficulty of a feature selection task is usually evaluated by the ratio between the number of instances (I) and the number of features (N), and the problems with a ratio $I/N \ll 1$ are recognized to be much more challenging. Indeed, most of the studies that investigate the robustness of ensemble feature selection have been so far conducted on high-dimensional genomic data [e.g. [21, 36, 37]], where the number of features exceeds the number of records by several order of magnitude; a few studies [30, 51], however, have also considered text categorization tasks where the number of features and the number of records are not so different. With the aim of evaluating the large-scale applicability and utility of ensemble feature selection, we conducted extensive experiments that encompass 18 classification tasks with very different I/N ratios [52–64], as given in Table 2, ranging from the setting $I/N \ll 1$ (as in the *Glioblastoma* dataset, where $I/N = 0.004$) to the setting $I/N > 1$ (as in the *MiceProtein* dataset, where $I/N = 14.03$).

Specifically, for each of the considered datasets, Table 2 reports a brief description of the task, the number of classes (both binary and multi-class problems are considered), the number of features, the number of instances and the instances-to-features ratio. As we can see, the 18 datasets have been chosen to be representatives of different kinds of applications, ranging from the classification of genomic and proteomic data to the analysis of texts, images, videos as well as specific kinds of signals (e.g. voice recordings and ECG recordings). An artificial dataset prepared for the NIPS 2003 feature selection challenge [53] has also been included.

Each of the above benchmarks has been managed according to the methodology presented in Sect. 3.3. In particular, for evaluating both stability and predictive performance, we built $T = 20$ training/test sets from the original set of records. (A number of preliminary experiments with higher values of T did not result in significantly different results.) As regards the other parameters of the methodology, i.e. the fraction X of the original records included in each training set, the number B of bootstraps used for the ensemble construction and the cut-off threshold th that determines the size of the selected subsets, we explored a number of potentially interesting values as detailed in what follows:

- We experimented with $X = 0.70$, $X = 0.80$ and $X = 0.90$, in order to evaluate the extent to which the amount of perturbation introduced in the training data affects the stability of the selection process; in particular, $X = 0.90$ corresponds to a "soft" perturbation setting, where the T training sets are not so different to each other, while lower values of X lead to training sets that overlap to a less extent. Since our experiments showed that the stability of the selection process decreases in a very dramatic way as the value of X decreases, we did not consider "hard" perturbation settings, with fractions $X < 0.70$.
- In the construction of the ensemble lists, one for each training set, we considered three different settings, i.e. $B = 20$, $B = 50$ and $B = 100$. Since B corresponds to the size of the ensemble, i.e. to the number of ranked lists that are aggregated to produce the ensemble list, a proper setting of this parameter is crucial to ensure an optimal trade-off between final performance and computational cost. In particular, we observed that the use of $B = 100$ resulted in drastically increased run times, with a limited (or null) improvement in terms of final performance (as discussed in Sect. 4.2), so higher values of B were not considered at all.
- Finally, regarding the cut-off threshold used to derive the simple and the ensemble subsets, we explored a very wide range of values, from 0.2 to 40% of the overall number of features. Of course, in dependence of the I/N ratio, some threshold values may be more suitable/interesting than others. In particular, when $I/N \ll 1$, the focus is usually on small feature subsets, but considering larger subset sizes can be still useful, even in the setting $I/N \ll 1$, in case the ranking process aims at achieving a preliminary dimensionality reduction, before the application of more sophisticated selection techniques.

Table 2 Datasets used in the empirical study

Dataset name	Problem description and reference	Number of classes	Number of features (N)	Number of instances (I)	I/N ratio
Glioblastoma	Gene expression-based classification of malignant gliomas [52]	4	12,625	50	0.004
Dexter	Text categorization benchmark as prepared for the NIPS 2003 feature selection challenge [53]	2	20,000	300	0.015
Ovarian	Proteomic patterns in serum that distinguish ovarian cancer from non-cancer [54]	2	15,154	253	0.017
Arcene	Proteomic benchmark built by merging different data sources, with the addition of spurious and noisy features [53]	2	10,000	200	0.020
Lymphoma	Gene expression-based classification of lymphoid malignancies, with unbalanced numbers of samples per class [55]	9	4026	96	0.024
Colonoscopy	Classification of gastrointestinal lesions from colonoscopy videos [56]	3	1396	76	0.054
LSVT voice rehabilitation	Analysis of vocal performance degradation in Parkinson's disease subjects [57]	2	310	126	0.406
Micromass	Identification of mixed bacterial species from mass spectrometry data [58]	20	1300	571	0.439
Secom	Defect recognition in semi-conductor manufacturing data [59]	2	590	520	0.881
Urban land cover	Urban land cover classification from high-resolution aerial image data [60]	9	147	168	1.143
CNAE-9	Categorization of free-text business descriptions [61]	9	856	1080	1.262
Arrhythmia	Arrhythmia disease classification, based on features extracted from ECG recordings [59]	13	279	452	1.620
Internet advertisements	Recognition of advertisements on Internet pages [62]	2	1558	3279	2.105
Sonar	Classification of sonar signals [59]	2	60	208	3.467
Madelon	Artificial dataset, which was part of the NIPS 2003 feature selection challenge [53]	2	500	2600	5.200
Semeion	Handwritten digits recognition [59]	10	256	1593	6.223
Scene	Outdoor scene classification from images [63]	2	294	2407	8.187
MiceProtein	Classification of Down syndrome mice, based on the expression levels of proteins measured in the cerebral cortex [64]	8	77	1080	14.03

The experimental study was conducted leveraging the WEKA machine learning workbench [65], which provides several functions for data manipulation, including the bootstrap aggregating approach, as well as several algorithms for feature ranking and classification. Considering the high number of datasets and the multiple settings of the experiments, the amount of results is large and, for the sake of clarity and readability, only a summary of them will be presented in the next subsections. For a more comprehensive view of the results, the reader can refer to the attached supplementary material.

4.2 Stability analysis

As a first point, we focus on the stability results, which give a measure of the extent to which the selection process is sensitive to perturbations in the input data. In our setting, as explained in Sect. 3.3, these perturbations are obtained by creating different training sets, each containing a fraction X of the original records. In this regard, it is interesting to initially consider how the stability, measured as the average similarity among the feature subsets extracted from the different training sets, is affected by the X value, i.e. by the amount of training data perturbation. To this end, limited to the simple ranking approach, Fig. 3 shows the stability behaviour of the selection methods here considered (χ^2, *InfoG*, *GainR*, *OneR*, *ReliefF*, *SVM-AW*, *SVM-RFE10*, *SVM-RFE50*), for different values of X and different subset sizes, on the *Glioblastoma* dataset, which is the one with the lowest I/N ratio (see Table 2).

As we can see, even a small amount of perturbation ($X = 0.90$) affects the stability in a significant way; indeed, the average similarity among the feature subsets selected from the different training sets is far lower than the maximum value of 1. As the amount of perturbation increases, the stability level dramatically falls off, for all the selection methods, though some of them exhibit a more robust behaviour. Similar considerations can be made for the other

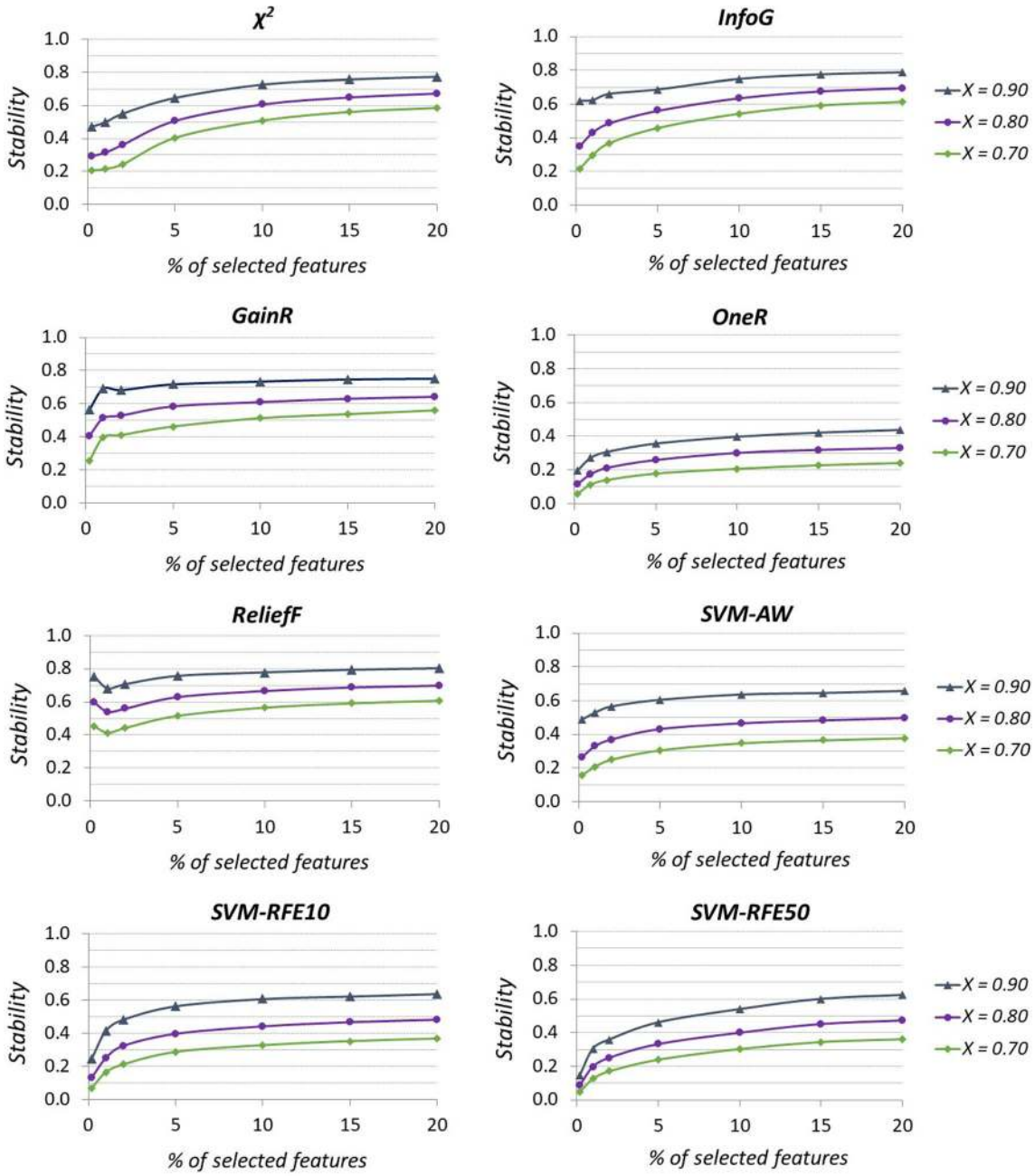

Fig. 3 *Glioblastoma* dataset ($I/N = 0.004$): stability patterns of the different ranking methods, in the simple ranking setting, for different levels of data perturbation ($X = 0.90$, $X = 0.80$, $X = 0.70$)

datasets in Table 2, even for higher I/N ratios, thus confirming that the instability of the selection outcome is a critical (though often neglected) concern. Quite surprisingly, even in the setting $I/N > 1$, some selection methods exhibit a problematic behaviour in terms of stability. The complete results, here omitted for the sake of space, are given in the attached supplementary material (section A).

The extent to which the adoption of an ensemble selection strategy improves the robustness of the selection process, in different domains and experimental settings, has been the main focus of our experimental study. In this regard, Figs. 4, 5, 6, 7 and 8 show, for five of the 18 datasets in Table 2, the stability of the different ranking methods, in both simple and ensemble versions, for different numbers of selected features, with a data perturbation level of $X = 0.80$. Specifically, for *Dexter* (Fig. 4), *Ovarian* (Fig. 5) and *Lymphoma* (Fig. 6) datasets, where the instances-to-features ratio is very low, the stability patterns are shown for subset sizes th $\leq 20\%$ of N, with a main focus on small thresholds, while for *LSVT Voice*

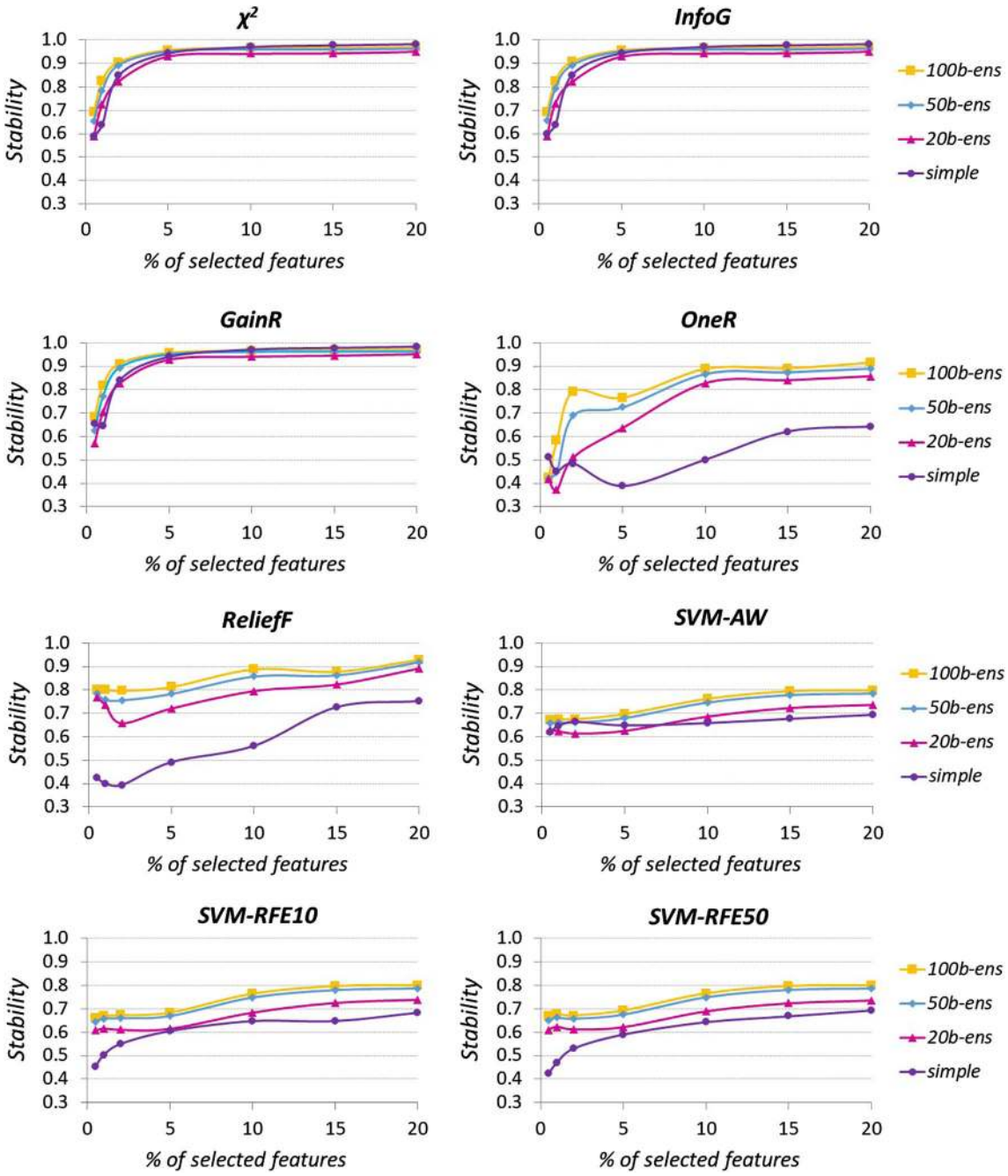

Fig. 4 *Dexter* dataset ($I/N = 0.015$): stability patterns of the different ranking methods, in the simple and in the ensemble version ($X = 0.80$)

Rehabilitation (Fig. 7) and *Urban Land Cover* (Fig. 8) datasets, with higher I/N ratios, larger feature subsets are also considered. Further, as explained in Sect. 4.1, three ensemble versions have been evaluated for each selection method, with different numbers of bootstraps ($B = 20$, $B = 50$, $B = 100$).

As we can see, the effectiveness of the ensemble approach is different for the different methods and varies in dependence on the subset size and the specific characteristics of the data at hand.

Specifically, the univariate filters χ^2 and *InfoG* exhibit a very similar behaviour and, when used in their simple form, turn out to be more robust than other selection methods. Globally, they seem to benefit to a limited extent from the ensemble approach which can be still useful, however, for some threshold values. In particular, in the *Lymphoma* dataset (Fig. 6), that presents a very low instances-to-features ratio and a high number of classes, the ensemble implementation of χ^2 and *InfoG* is appreciably

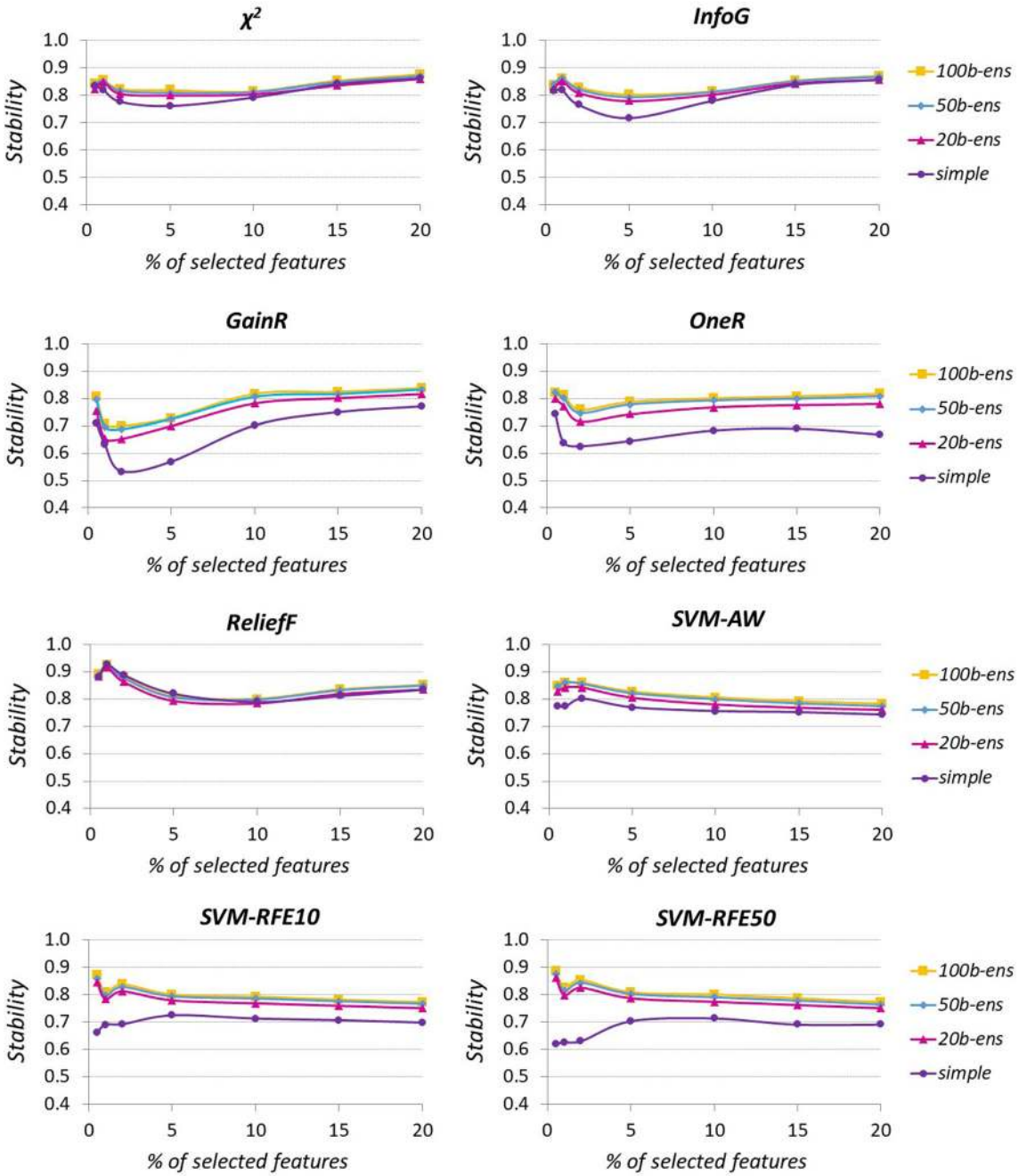

Fig. 5 *Ovarian* dataset ($I/N = 0.017$): stability patterns of the different ranking methods, in the simple and in the ensemble version ($X = 0.80$)

more robust for small feature subsets (where biomedical experts typically focus their attention).

The other univariate filter, i.e. *GainR*, has a behaviour quite similar to χ^2 and *InfoG* for some kinds of data, e.g. in the text categorization domain (Fig. 4), but turns out to be intrinsically less stable in other application contexts, e.g. on mass spectrometry data (Fig. 5), where the adoption of an ensemble selection strategy turns out to be quite beneficial. In turn, the remaining univariate method, *OneR*, which exploits a rule-based classifier to rank the features, has undoubtedly a problematic behaviour in terms of stability when used in its simple form. It is clear that this method, irrespective of the specific problem at hand, can take significant advantage of the ensemble implementation, with improvements in stability even in the order of 50% (and more) of the original value.

About the multivariate selection approaches, we can see that the stability behaviour of the *ReliefF* filter is highly dependent on the characteristics of the data at hand. Indeed, in Figs. 5, 6 and 8, it exhibits a good stability in its simple

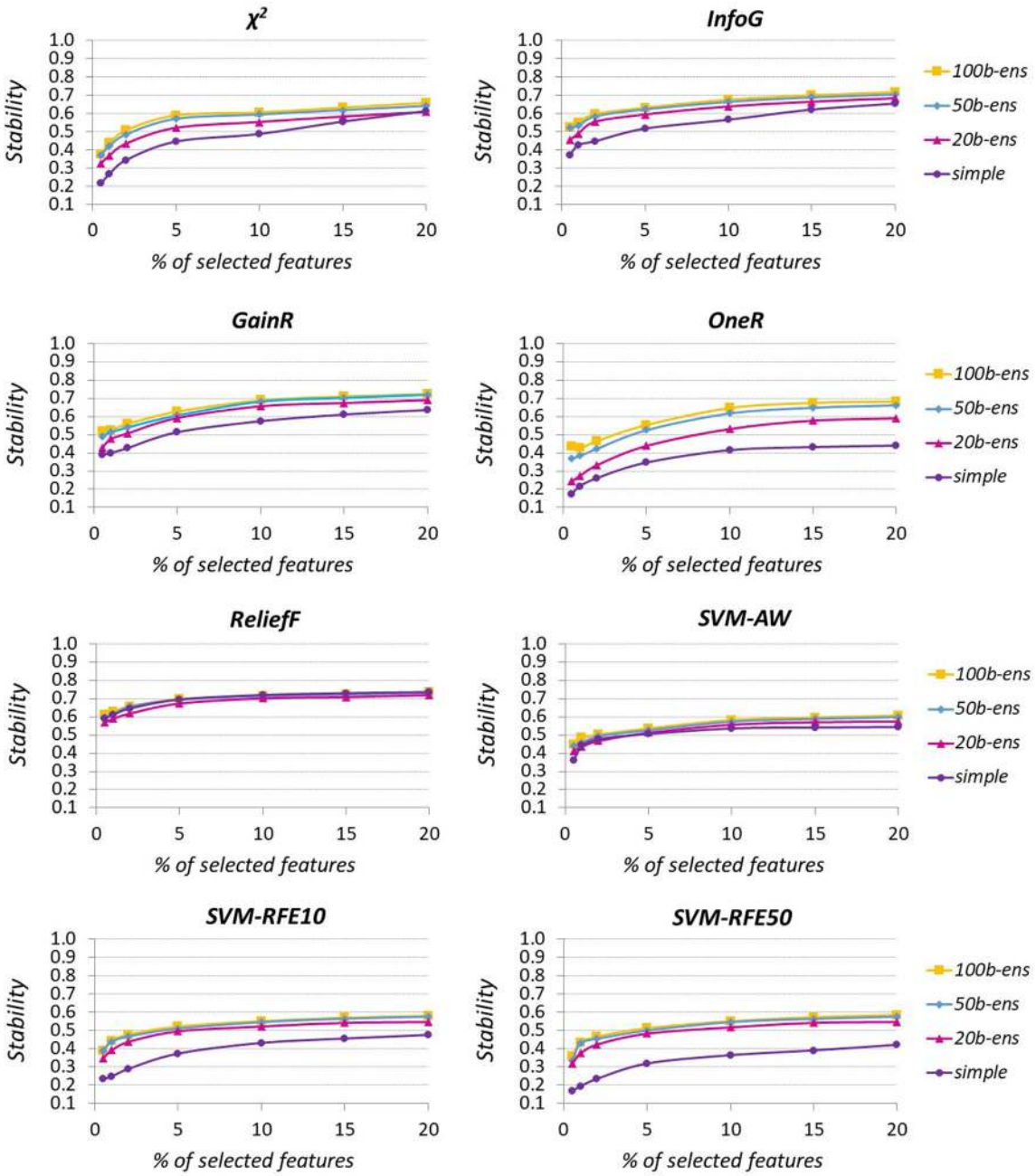

Fig. 6 *Lymphoma* dataset ($I/N = 0.024$): stability patterns of the different ranking methods, in the simple and in the ensemble version ($X = 0.80$)

form and does not benefit at all from the ensemble implementation. Nevertheless, the ensemble approach is quite beneficial in the text categorization domain (Fig. 4), where the simple version of *ReliefF* is very unstable, and in the *LSVT Voice Rehabilitation* dataset (Fig. 7), but in this case only for large threshold values.

Further, as regards the other multivariate techniques included in this study, i.e. the embedded SVM-based selectors (*SVM-AW*, *SVM-RFE10* and *SVM-RFE50*), we can see that *SVM-AW* has in turn a domain-dependent behaviour, sometimes taking advantage of the ensemble approach and sometimes not. In contrast, the two iterative selectors *SVM-RFE10* and *SVM-RFE50* exhibit a quite poor stability in their simple form, irrespective of the specific characteristics of the data at hand. In this case, the advantage of adopting an ensemble strategy is unquestionable, with improvements in stability that are somewhat dataset dependent but always significant.

Irrespective of the specific behaviour of each selection method, Figs. 4, 5, 6, 7 and 8 clearly show that using 50 bootstraps for the ensemble construction can be a suitable choice. Indeed, 20 bootstraps turn out to be sometimes

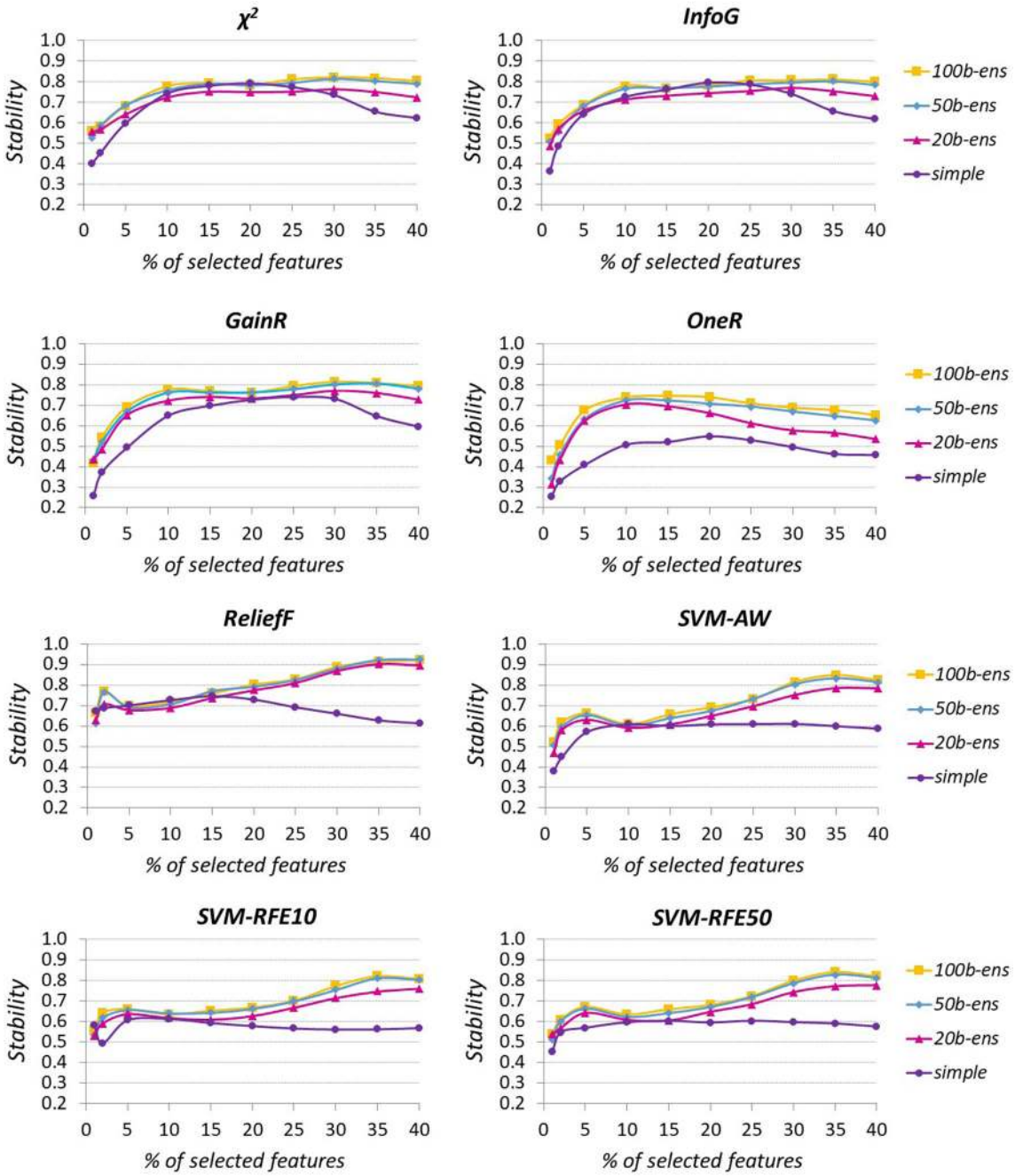

Fig. 7 *LSVT Voice Rehabilitation* dataset ($I/N = 0.406$): stability patterns of the different ranking methods, in the simple and in the ensemble version ($X = 0.80$)

insufficient, resulting in a still improvable outcome, while the use of 100 bootstraps is not beneficial since it increases in a significant way the computational cost without improving (with a few exceptions) the robustness of the selection process.

It is important to remark that the above considerations are not limited to the five benchmarks in Figs. 4, 5, 6, 7 and 8, but are essentially valid for the other datasets in Table 2 too. In particular, as previously highlighted, the domains with relatively high instances-to-features ratios are not exempt from the selection instability issue. Quite interestingly, the adoption of an ensemble selection strategy can be sometimes useful not only in the hard $I/N \ll 1$ setting, but even when the number of features is quite small compared to the number of instances, as in the *MiceProtein* dataset, which has the highest I/N ratio (Table 2). Indeed, as shown in Fig. 9, some methods (especially the filters χ^2, *InfoG* and *ReliefF*) turn out to be stable already in their simple form,

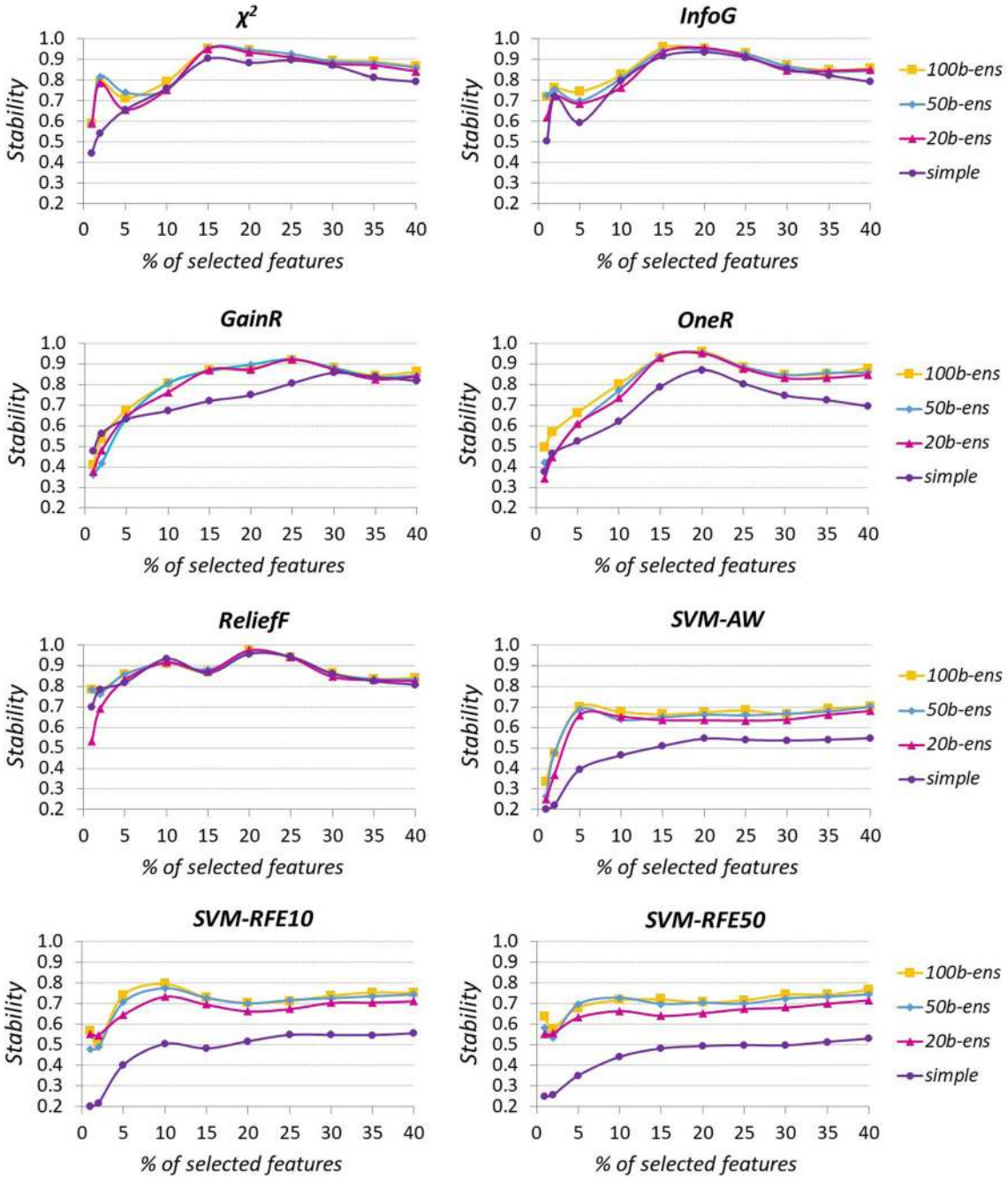

Fig. 8 *Urban Land Cover* dataset (I/N = 1.143): stability patterns of the different ranking methods, in the simple and in the ensemble version (X = 0.80)

while other selectors (especially *SVM-RFE10* and *SVM-RFE50*) achieve a good stability only in the ensemble version.

Globally, the analysis performed on the 18 benchmarks here considered shows that the beneficial impact of the ensemble approach strongly depends on the intrinsic stability of the simple selectors: the less stable they are, the higher the gain in stability achieved in the ensemble version. In particular, while the behaviour of the filter methods can be dependent on the characteristics of the data at hand, the embedded approaches exhibit in general a poorer robustness and take significant advantage of the ensemble implementation, as shown in Figs. 10 and 11. Specifically, Figs. 10 and 11 show the comparison between the stability of the simple and the ensemble ranking, for the univariate *OneR* and the multivariate *SVM-RFE50*, respectively, for the 12 datasets for which that comparison has not been

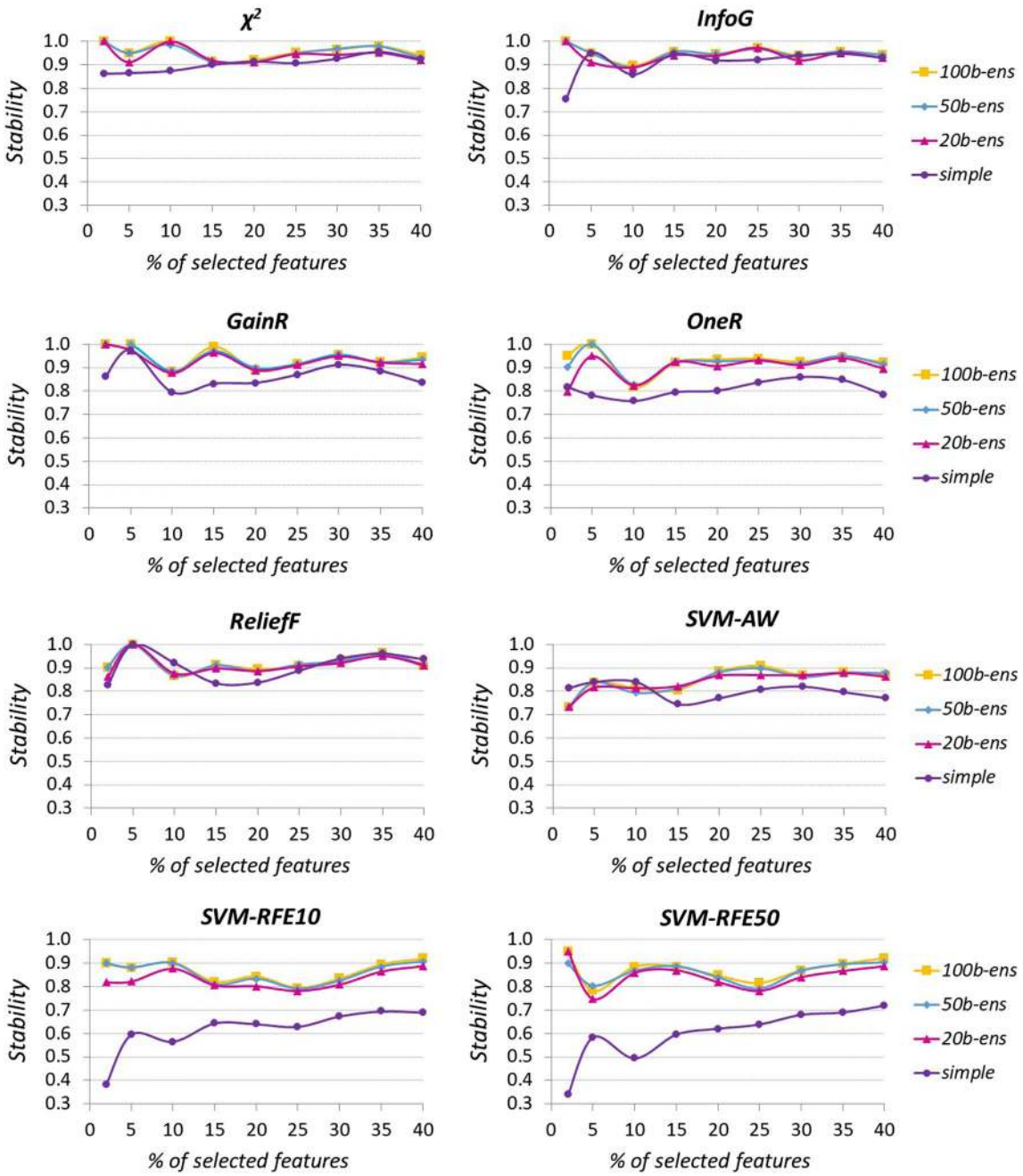

Fig. 9 *MiceProtein* dataset (I/N = 14.03): stability patterns of the different ranking methods, in the simple and in the ensemble version (X = 0.80)

shown above. As we can see, the ensemble ranking is significantly more robust, with a few exceptions.

4.3 Predictive performance analysis

The stability analysis has been complemented, according to the methodology presented in Sect. 3.3, with the analysis of the predictive performance. Specifically, the selected feature subsets have been used to train both *support vector machines* (SVM) and *random forest* (RF) classifiers, which have proved to be "best of class" learners in several domains [66–69]. In particular, for the SVM classifier we use a linear kernel, while the RF classifier is parameterized based on common practice in the literature [70, 71] ($\log_2$(th) + 1 random features and 100 trees). Note that our focus here is not to find which classifier performs best but to investigate, for a given classifier and a given selection algorithm, the impact of using an ensemble selection strategy.

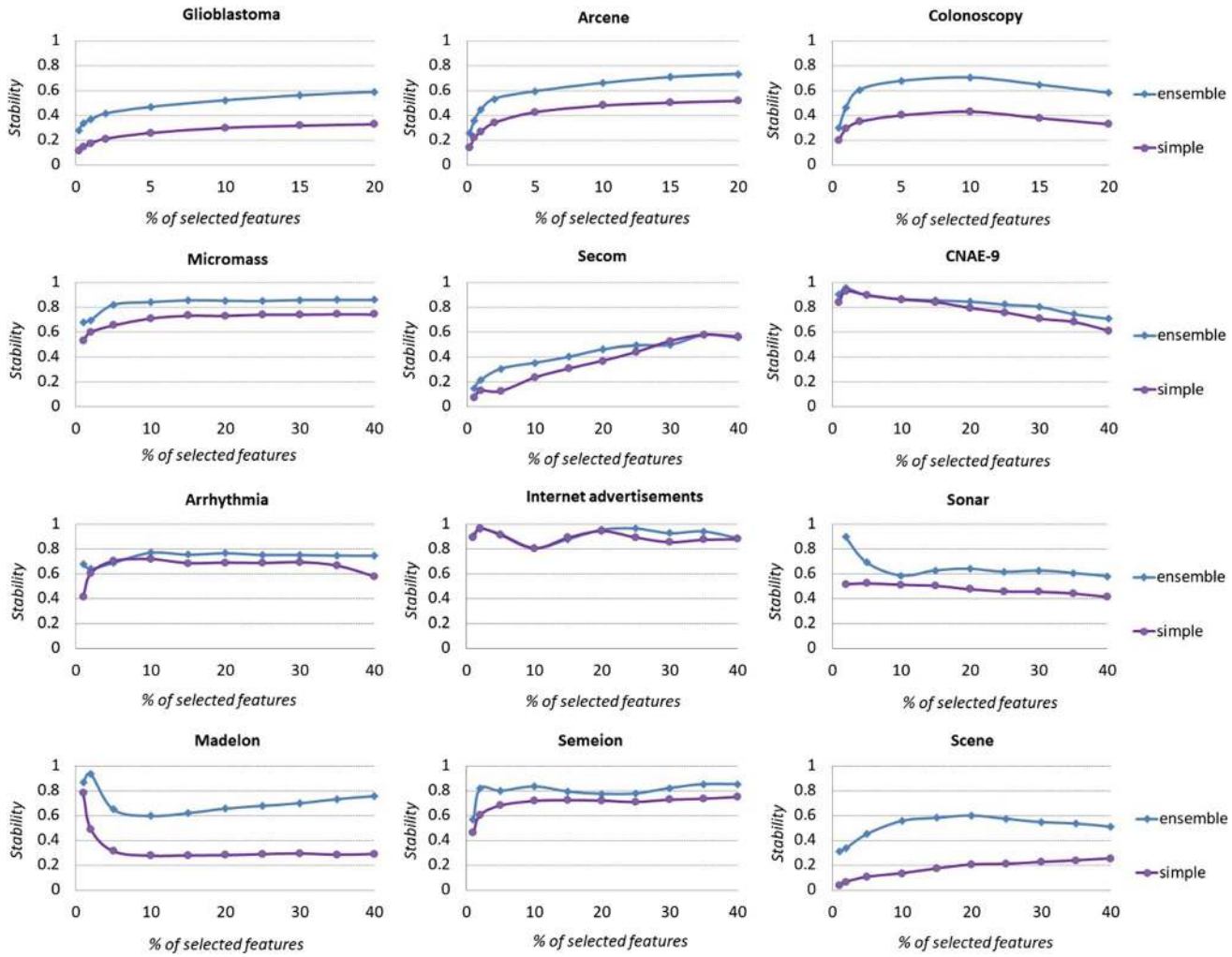

Fig. 10 Stability of the *OneR* method, in the simple and in the ensemble version ($X = 0.80$, $B = 50$), for different datasets

Though the most suitable metrics for performance evaluation should be chosen in dependence on the characteristics of the data at hand, our study deals with a high number of very heterogeneous datasets, as discussed in Sect. 4.1, so our comparative analysis is carried out using as a single synthesis measure the AUC, i.e. the *area under the receiver operating characteristic* (ROC) curve [40]. Indeed, it is widely employed in previous studies on ensemble feature selection [e.g. 21, 37] and provides a richer measure of classification performance than accuracy or error rate [72].

For the *Arcene* dataset, as an example, Fig. 12 shows the AUC performance achieved by both SVM and RF classifiers, when used in conjunction with the different ranking methods here considered, in their simple and ensemble version, for different percentages of selected features (with a special focus on small subset sizes, given the small I/N ratio). Specifically, the values here reported are obtained by averaging the AUC results over the $T = 20$ training/test sets derived from the original dataset. (The corresponding standard deviations are in the range 0.03–0.07, with the highest values registered for smaller feature subsets.)

As we can see, the RF classifier has here a better AUC performance but, for the purposes of our study, the most important observation is that, for both SVM and RF, the average AUC values registered in the simple and in the ensemble setting are almost identical. The only exception is the *SVM-AW* method that leads to inferior AUC results in the simple version and achieves a significant improvement in the ensemble version. Noteworthy, for a given classifier, there is no selection algorithm that turns out to be superior in terms of final predictive performance. (Indeed, the different methods produce very similar results in terms of AUC.) This confirms that, for a given problem, different feature subsets can discriminate the classes equally well.

The above observations are fully confirmed by the AUC analysis performed on the other datasets in Table 2. Indeed, irrespective of the I/N ratio and whatever the size

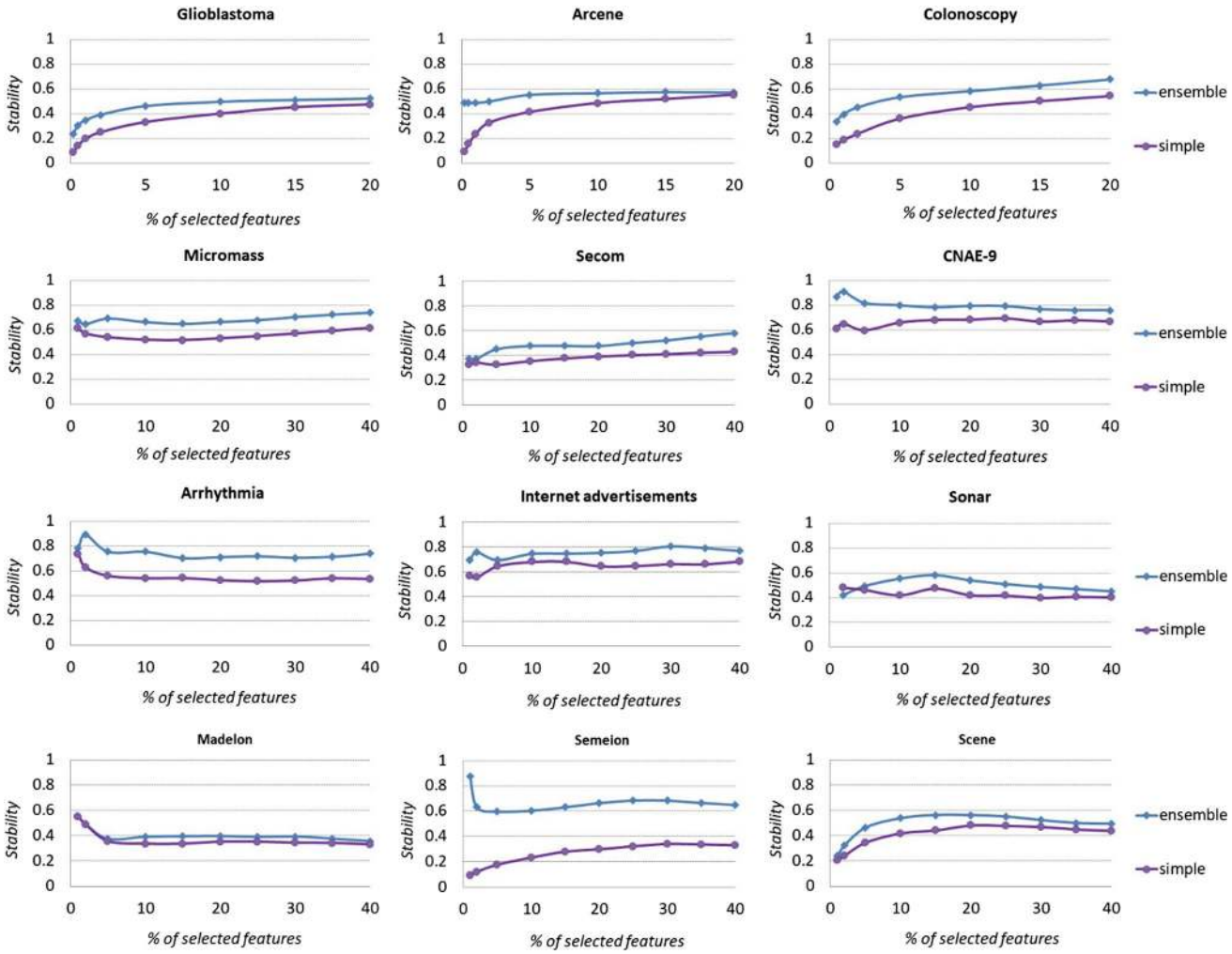

Fig. 11 Stability of the *SVM-RFE50* method, in the simple and in the ensemble version ($X = 0.80$, $B = 50$), for different datasets

of the selected subsets, we found that the adoption of an ensemble selection strategy, which can have a strong impact on the stability of selection outcome, does not influence in a significant way the final predictive performance. For the sake of space and readability, we summarize in Tables 3, 4 and 5 only the AUC results registered for the *OneR* and *SVM-RFE50* selection methods, whose stability behaviour has been discussed in detail previously.

Specifically, for the first six datasets in Table 2 (which have the lowest *I*/*N* ratio), Table 3 shows the AUC performance achieved by both SVM and RF in conjunction with *OneR* and *SVM-RFE50*, in their simple and ensemble version. We considered here a subset size th = 2% of *N*, which is enough to obtain a predictive performance comparable to that measured using all the original features (i.e. without reducing the dimensionality of the $T = 20$ training sets built from the original dataset). Note that both SVM and RF classifiers are known to scale well on high-dimensional spaces, but a dimensionality reduction is still crucial, in several domains, for data understanding/interpretability and knowledge discovery purposes. Further, the feature subsets selected through ranking techniques as the one here considered can often be refined through more sophisticated wrapper approaches that further improve the final predictive performance [2, 73]. The same AUC analysis is given in Tables 4 and 5, for datasets with higher *I*/*N* ratios; here larger values of the cut-off threshold are considered, respectively, th = 10% of *N* in Table 4 and th = 20% of *N* in Table 5. Again, subsets with a predictive performance comparable to that obtained with the original feature space are considered, to give an example of the impact of the adoption of an ensemble selection strategy; the extended AUC results for different subset sizes are provided in the attached supplementary material (section B).

The overall AUC analysis, even considering subset sizes different from the ones reported in Tables 3, 4 and 5, clearly shows that the simple and the ensemble ranking are

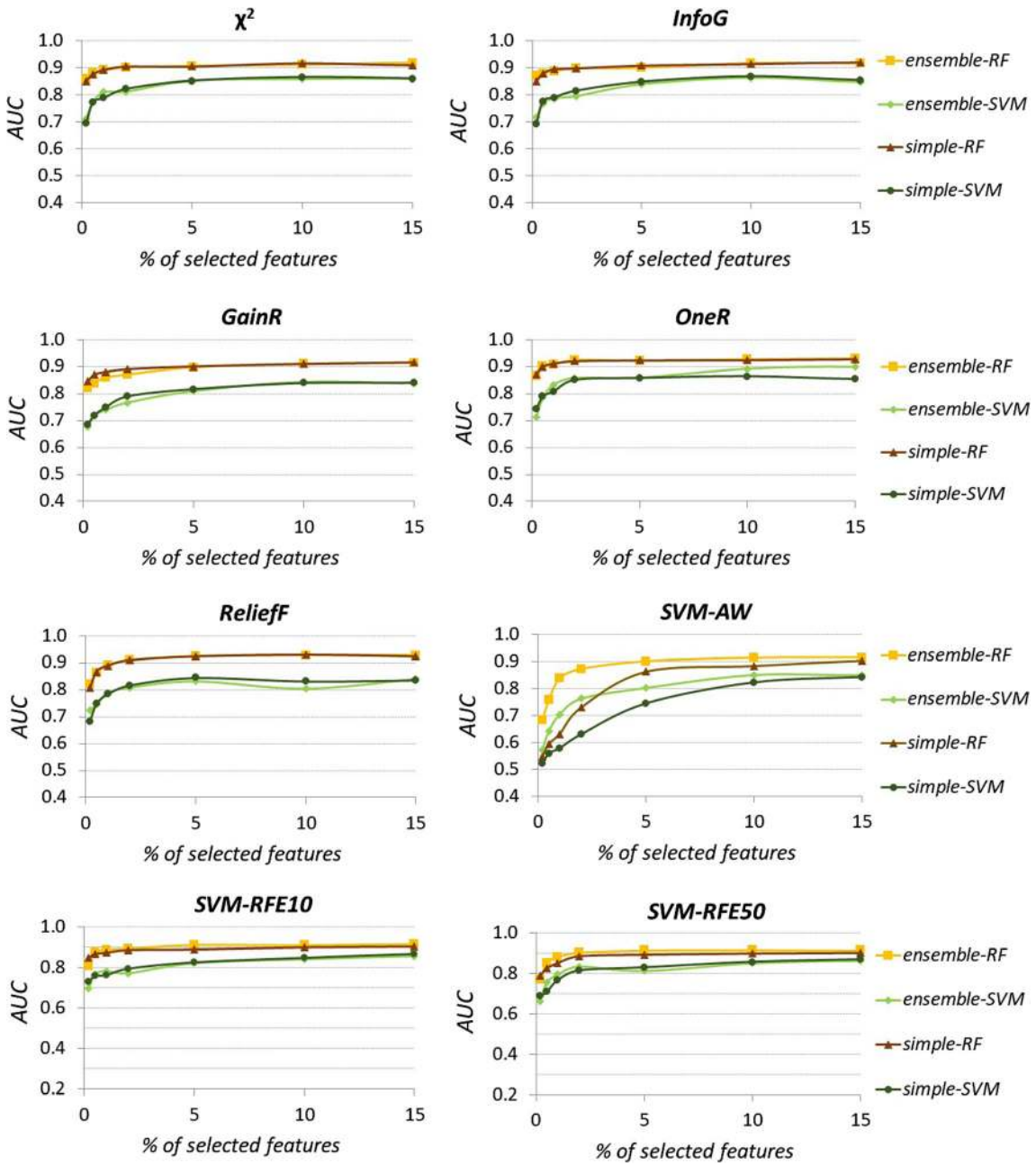

Fig. 12 *Arcene* dataset ($I/N = 0.020$): AUC performance of SVM and RF classifiers, in conjunction with the different ranking methods, in their simple and ensemble version ($X = 0.80$, $B = 50$)

almost equivalent in terms of final predictive performance, irrespective of the specific selection approach (filter or embedded, univariate or multivariate). When looking at the trade-off between the predictive performance and the stability of the selection process, we can then conclude that the adoption of an ensemble strategy can lead to more stable feature subsets without compromising at all the predictive power of these subsets.

4.4 Discussion

Globally, the results presented in Sects. 4.2 and 4.3 confirm that homogeneous ensembles are effective in making the feature selection process more robust. Indeed, even in those domains where the selection of stable subsets is intrinsically harder, due to a very low instances-to-features ratio, the ensemble approach can produce better, though sometimes not completely satisfactory, results. In turn, the applications that are less critical in terms of stability (e.g.

Table 3 AUC performance of SVM and RF classifiers, in conjunction with *OneR* and *SVM-RFE50* ranking methods, in their simple and ensemble version ($X = 0.80$, $B = 50$, th = 2% of N), for six datasets having $I/N \ll 1$

Dataset	SVM classifier					RF classifier				
	OneR		SVM-RFE50		Full dataset	OneR		SVM-RFE50		Full dataset
	Simple	Ensemble	Simple	Ensemble		Simple	Ensemble	Simple	Ensemble	
Glioblastoma	0.875	0.909	0.878	0.896	0.889	0.905	0.940	0.914	0.929	0.912
Dexter	0.882	0.887	0.900	0.898	0.901	0.959	0.945	0.960	0.962	0.943
Ovarian	1.000	0.999	1.000	1.000	1.000	1.000	0.999	1.000	1.000	0.996
Arcene	0.851	0.858	0.816	0.836	0.855	0.920	0.924	0.887	0.904	0.912
Lymphoma	0.941	0.943	0.962	0.964	0.964	0.963	0.956	0.972	0.971	0.972
Colonoscopy	0.739	0.733	0.761	0.762	0.771	0.795	0.787	0.751	0.767	0.785

Table 4 AUC performance of SVM and RF classifiers, in conjunction with *OneR* and *SVM-RFE50* ranking methods, in their simple and ensemble version ($X = 0.80$, $B = 50$, th = 10% of N), for six datasets having I/N in the range (0.4, 1.7)

Dataset	SVM classifier					RF classifier				
	OneR		SVM-RFE50		Full dataset	OneR		SVM-RFE50		Full dataset
	Simple	Ensemble	Simple	Ensemble		Simple	Ensemble	Simple	Ensemble	
LSVT voice rehabilitation	0.773	0.788	0.859	0.850	0.833	0.880	0.883	0.918	0.913	0.890
Micromass	0.953	0.955	0.956	0.962	0.971	0.990	0.990	0.992	0.994	0.992
Secom	0.521	0.573	0.581	0.583	0.575	0.659	0.733	0.730	0.732	0.736
Urban land cover	0.894	0.897	0.949	0.950	0.950	0.951	0.951	0.982	0.977	0.979
CNAE-9	0.965	0.966	0.966	0.976	0.982	0.979	0.979	0.983	0.986	0.995
Arrhythmia	0.685	0.684	0.690	0.691	0.708	0.781	0.786	0.783	0.788	0.781

Table 5 AUC performance of SVM and RF classifiers, in conjunction with *OneR* and *SVM-RFE50* ranking methods, in their simple and ensemble version ($X = 0.80$, $B = 50$, th = 20% of N), for six datasets having $I/N > 2$

Dataset	SVM classifier					RF classifier				
	OneR		SVM-RFE50		Full dataset	OneR		SVM-RFE50		Full dataset
	Simple	Ensemble	Simple	Ensemble		Simple	Ensemble	Simple	Ensemble	
Internet advertisements	0.899	0.897	0.922	0.920	0.923	0.975	0.975	0.978	0.980	0.982
Sonar	0.749	0.752	0.735	0.744	0.760	0.892	0.891	0.894	0.902	0.928
Madelon	0.591	0.593	0.559	0.558	0.566	0.800	0.867	0.705	0.702	0.713
Semeion	0.956	0.958	0.974	0.974	0.988	0.980	0.980	0.990	0.989	0.996
Scene	0.679	0.668	0.723	0.728	0.730	0.884	0.902	0.908	0.909	0.918
MiceProtein	0.982	0.982	0.989	0.988	0.996	0.999	0.999	0.999	0.999	1.0 00

in presence of relatively high values of I/N) can still benefit from the ensemble approach depending on the number of selected features and on the adopted selection algorithm. These findings complement the results of other studies in the literature which have investigated the robustness of ensemble feature selection but limited to a single application domain [36, 39], to a single selection method [21, 51] or to a given number of selected features [30, 37].

Encompassing different real-world problems and different feature subset cardinalities, our study is exhaustive in showing that the methods intrinsically less stable, such as the univariate *OneR* and the multivariate *SVM-RFE*, benefit to a greater extent from the ensemble approach.

Indeed, it significantly improves the robustness of the selection process with no degradation in the predictive performance of the selected subsets. Other methods, such as the univariate *InfoG* and the multivariate *ReliefF*, turn out to be intrinsically more robust but may still take advantage of the ensemble implementation for some kinds of data (and in dependence on the size of the selected subsets).

Interestingly, for each of the benchmarks here considered, the adoption of an ensemble strategy has the effect of reducing the gap between the least and the most stable methods, hence making the choice of the specific selection algorithm less decisive for the final outcome. This is noteworthy for practitioners and final users that could take advantage, in the ensemble setting, of functionally different, but almost equally robust, selection methods. We point out, though, that equally robust methods do not necessarily result in the same set of selected features.

With this regard, it would be interesting to deeply investigate the extent to which the adoption of an ensemble implementation affects the "intrinsic" similarity of different selection algorithms, i.e. the degree of overlapping among the subsets they produce. According to the methodology presented in [74], we performed a number of experiments to compare the composition of the subsets selected by the various ranking methods here considered, in both the simple and in the ensemble setting. Interestingly, it turns out that the similarity among the selected subsets often increases in the ensemble setting, though the ensemble subsets still overlap only to some extent. The preliminary results of this analysis, for a number of datasets chosen to be representative of different *I*/*N* ratios, are given in the attached supplementary material (section C). Further research will be carried out on this topic, with extensive experiments on more datasets and for a wide range of subset sizes.

To conclude this discussion, it can be useful to provide the reader with some details about the computational impact of the selection methods here adopted. Of course, the ensemble approach poses greater demands in terms of resource consumption, the execution time being dependent on the specific ensemble implementation. Specifically, in case of homogeneous ensembles as the ones here considered, the cost of building an ensemble list increases linearly with the number of bootstraps (i.e. with the number of ranked lists involved in the aggregation process). In turn, the cost of building a single ranked list depends on the number of instances/features of the dataset at hand as well as on the adopted selection algorithm. To give an idea, for the *Dexter* dataset, which is the benchmark with the highest number of features (20,000) among the ones in Table 2, the execution times[1] for the construction of a single ranked list are as follows: about 1 s for *ReliefF*, 2 s for χ^2, *InfoG*, *GainR* and *SVM-AW*, 3 s for *SVM-RFE50*, 6 s for *SVM-RFE10*, 30 s for *OneR*. But the most demanding benchmark, in terms of resource consumption, is the *Internet advertisements* dataset where the high number of features (1558) is coupled with a quite high number of instances (3279); in this case, the execution times are as follows: about 1 s for χ^2, *InfoG* and *GainR*, 7 s for *OneR* and *SVM-AW*, 21 s for *SVM-RFE50*, 96 s for *SVM-RFE10* and 110 s for *ReliefF*. For all the other datasets included in our study the computational cost turned out to be sensibly lower. Of course, since the time required for the ensemble construction increases proportionally to the number of the ensemble components, the overall computational cost may be quite high, depending on the characteristics of the dataset at hand. Distributing the ensemble components over several nodes could be a way to improve the overall efficiency, as discussed in [19].

5 Conclusions

Although the literature on feature selection stability is still quite limited, an increasing number of studies recognize the importance of devising feature selection protocols that ensure an acceptable trade-off between the final predictive performance and the stability of the selection process. This work contributes to demonstrate that the ensemble selection paradigm can be, in this regard, a suitable option.

Specifically, we experimentally explored the effectiveness of a *functionally homogenous* ensemble approach which involves the application of a given selection algorithm to a number of diversified datasets derived from the original set of records. As shown by extensive experiments conducted on high-dimensional benchmarks from different domains, this ensemble setting can lead to a significant gain in stability without any degradation of the predictive performance.

The extent to which the ensemble implementation outperforms the simple version of a given algorithm is strongly dependent on the "intrinsic" stability of the algorithm itself, with larger gains in robustness for the least stable methods. It is worth highlighting that even selection methods that are quite different to each other (from an algorithmic point of view) tend to exhibit a similar performance, in terms of both accuracy and stability, when used in their ensemble version (i.e. when diversity is injected in the training data). This does not mean, however, that such ensemble selectors will result in the same set of selected features: indeed, they may still produce subsets that contain different features, though exhibiting a similar

[1] Experiments were performed with a CPU of 3.3 GHz and 8 GB of RAM.

performance. These subsets, in turn, could be jointly exploited to achieve a better understanding of the underlying domain.

Indeed, as future line of research, it could be interesting to explore the full potential of hybrid ensemble approaches, where diversity is injected both at the data level and at the algorithmic level. This might open the way to the definition of more flexible selection strategies which leverage multiple heuristics while reducing the degree of dependence on the specific composition of the training data.

Acknowledgments This research was supported by Sardinia Regional Government, within the projects "DomuSafe" (L.R. 7/2007, annualità 2015, CRP 69) and "EmILIE" (L.R. 7/2007, annualità 2016, CUP F72F16003030002).

Compliance with ethical standards

Conflict of interest The author declares that there is no conflict of interest regarding the publication of this paper.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
2. Bolón-Canedo V, Sánchez-Maroño N, Alonso-Betanzos A (2015) Recent advances and emerging challenges of feature selection in the context of big data. Knowl-Based Syst 86:33–45
3. Bolón-Canedo V, Sánchez-Maroño N, Alonso-Betanzos A (2013) A review of feature selection methods on synthetic data. Knowl Inf Syst 34(3):483–519
4. Tang J, Alelyani S, Liu H (2014) Feature selection for classification: a review. In: Aggarwal CC (ed) Data classification: algorithms and applications. CRC Press, Boca Raton, pp 37–64
5. Khalid S, Khalil T, Nasreen S (2014) A survey of feature selection and feature extraction techniques in machine learning. In: Science and information conference (SAI), London, UK, pp 372–378
6. Vergara JR, Estévez PA (2014) A review of feature selection methods based on mutual information. Neural Comput Appl 24:175–186
7. Awada W, Khoshgoftaar TM, Dittman D, Wald R, Napolitano A (2012) A review of the stability of feature selection techniques for bioinformatics data. In: IEEE 13th international conference on information reuse and integration, IEEE, pp 356–363
8. Pes B (2017) Feature selection for high-dimensional data: the issue of stability. In: 26th IEEE international conference on enabling technologies: infrastructure for collaborative enterprises, WETICE 2017, Poznan, Poland, pp 170–175
9. Alelyani S, Zhao Z, Liu H (2011) A dilemma in assessing stability of feature selection algorithms. In: IEEE 13th international conference on high performance computing and communications, pp 701–707
10. Zengyou H, Weichuan Y (2010) Stable feature selection for biomarker discovery. Comput Biol Chem 34:215–225
11. Saeys Y, Abeel T, Van de Peer Y (2008) Robust feature selection using ensemble feature selection techniques. In: Machine learning and knowledge discovery in databases. Lecture notes in computer science, vol 5212. Springer, Berlin, pp 313–325
12. Yang F, Mao KZ (2011) Robust feature selection for microarray data based on multicriterion fusion. IEEE/ACM Trans Comput Biol Bioinform 8(4):1080–1092
13. Dietterich T (2000) Ensemble methods in machine learning. In: Multiple classifier systems. Lecture notes in computer science, vol 1857. Springer, Berlin, pp 1–15
14. Woźniak M, Graña M, Corchado E (2014) A survey of multiple classifier systems as hybrid systems. Inf Fusion 16:3–17
15. Rojas-Thomas JC, Mora M, Santos M (2017) Neural networks ensemble for automatic DNA microarray spot classification. Neural Comput Appl. https://doi.org/10.1007/s00521-017-3190-6
16. Mohapatra S, Patra D, Satpathy S (2014) An ensemble classifier system for early diagnosis of acute lymphoblastic leukemia in blood microscopic images. Neural Comput Appl 24:1887–1904
17. Ala'raj M, Abbod MF (2016) Classifiers consensus system approach for credit scoring. Knowl-Based Syst 104:89–105
18. Guan D, Yuan W, Lee YK, Najeebullah K, Rasel MK (2014) A review of ensemble learning based feature selection. IETE Tech Rev 31(3):190–198
19. Seijo-Pardo B, Porto-Díaz I, Bolón-Canedo V, Alonso-Betanzos A (2017) Ensemble feature selection: homogeneous and heterogeneous approaches. Knowl-Based Syst 118:124–139
20. Bühlmann P (2012) Bagging, boosting and ensemble methods. In: Gentle J, Härdle W, Mori Y (eds) Handbook of computational statistics. Springer handbooks of computational statistics. Springer, Berlin
21. Abeel T, Helleputte T, Van de Peer Y, Dupont P, Saeys Y (2010) Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. Bioinformatics 26(3):39–392
22. Golay J, Leuenberger M, Kanevski M (2017) Feature selection for regression problems based on the Morisita estimator of intrinsic dimension. Pattern Recognit 70:126–138
23. Neumann U, Heider D (2018) Ensemble feature selection for regression problems. In: European conference on data analysis (ECDA 2018), book of abstracts, p 19
24. Kumar V, Minz S (2014) Feature selection: a literature review. Smart Comput Rev 4(3):211–229
25. Kuncheva LI (2004) Combining pattern classifiers: methods and algorithms. Wiley-Interscience, Hoboken
26. Altidor W, Khoshgoftaar TM, Van Hulse J, Napolitano A (2011) Ensemble feature ranking methods for data intensive computing applications. In: Furth B, Escalante A (eds) Handbook of data intensive computing. Springer, New York, pp 349–376
27. Wald R, Khoshgoftaar TM, Dittman D, Awada W, Napolitano A (2012) An extensive comparison of feature ranking aggregation techniques in bioinformatics. In: IEEE 13th international conference on information reuse and integration, IEEE, pp 377–384
28. Wald R, Khoshgoftaar TM, Dittman D (2012) Mean aggregation versus robust rank aggregation for ensemble gene selection. In: 11th international conference on machine learning and applications, IEEE, pp 63–69
29. Dessì N, Pes B, Angioni M (2015) On stability of ensemble gene selection. In: Intelligent data engineering and automated learning—IDEAL 2015. Lecture notes in computer science, vol 9375. Springer, Berlin, pp 416–423
30. Woznica A, Nguyen P, Kalousis A (2012) Model mining for robust feature selection. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, ACM, pp 913–921

31. Rokach L, Chizi B, Maimon O (2007) A methodology for improving the performance of non-ranker feature selection filters. Int J Pattern Recognit Artif Intell 21(05):809–830
32. Bolón-Canedo V, Sánchez-Maroño N, Alonso-Betanzos A (2014) Data classification using an ensemble of filters. Neurocomputing 135:13–20
33. Latkowski T, Osowski S (2015) Data mining for feature selection in gene expression autism data. Expert Syst Appl 42:864–872
34. Olsson J, Oard DW (2006) Combining feature selectors for text classification. In: 15th ACM international conference on Information and knowledge management, ACM, pp 798–799
35. Wang H, Khoshgoftaar TM, Napolitano A (2012) Software measurement data reduction using ensemble techniques. Neurocomputing 92:124–132
36. Brahim AB, Limam M (2017) Ensemble feature selection for high dimensional data: a new method and a comparative study. Adv Data Anal Classif 1:2–8. https://doi.org/10.1007/s11634-017-0285-y
37. Haury AC, Gestraud P, Vert JP (2011) The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. PLoS ONE 6(12):e28210
38. Kuncheva LI, Smith CJ, Syed Y, Phillips CO, Lewis KE (2012) Evaluation of feature ranking ensembles for high-dimensional biomedical data: a case study. In: IEEE 12th international conference on data mining workshops, IEEE, pp 49–56
39. Pes B, Dessì N, Angioni M (2017) Exploiting the ensemble paradigm for stable feature selection: a case study on high-dimensional genomic data. Inf Fusion 35:132–147
40. Witten IH, Frank E, Hall MA, Pal CJ (2016) DATA MINING: practical machine learning tools and techniques. Morgan Kaufmann, Burlington
41. Rakotomamonjy A (2003) Variable selection using SVM based criteria. J Mach Learn Res 3:1357–1370
42. Holte RC (1993) Very simple classification rules perform well on most commonly used datasets. Mach Learn 11:63–91
43. Robnik-Sikonja M, Kononenko I (2003) Theoretical and empirical analysis of relief and rrelieff. Mach Learn 53:23–69
44. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46:389–422
45. Somol P, Novovicova J (2010) Evaluating stability and comparing output of feature selectors that optimize feature subset cardinality. IEEE Trans Pattern Anal Mach Intell 32(11):1921–1939
46. Wang H, Khoshgoftaar TM, Wald R, Napolitano A (2012) A novel dataset-similarity-aware approach for evaluating stability of software metric selection techniques. In: IEEE international conference on information reuse and integration, pp 1–8
47. Dernoncourt D, Hanczar B, Zucker JD (2014) Analysis of feature selection stability on high dimension and small sample data. Comput Stat Data Anal 71:681–693
48. Kuncheva LI (2007) A stability index for feature selection. In: 25th IASTED international multi-conference: artificial intelligence and applications, ACTA Press, pp 390–395
49. Cannas LM, Dessì N, Pes B (2013) Assessing similarity of feature selection techniques in high-dimensional domains. Pattern Recognit Lett 34(12):1446–1453
50. Seijo-Pardo B, Bolón-Canedo V, Alonso-Betanzos A (2019) On developing an automatic threshold applied to feature selection ensembles. Inf Fusion 45:227–245
51. Van Landeghem S, Abeel T, Saeys Y, Van de Peer Y (2010) Discriminative and informative features for biomolecular text mining with ensemble feature selection. Bioinformatics 26:i554–i560
52. Nutt CL, Mani DR, Betensky RA, Tamayo P et al (2003) Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. Cancer Res 63(7):1602–1607
53. Guyon I, Gunn SR, Ben-Hur A, Dror G (2004) Result analysis of the NIPS 2003 feature selection challenge. In: Advances in neural information processing systems, vol 17. MIT Press, pp 545—552
54. Petricoin EF, Ardekani AM, Hitt BA, Levine PJ et al (2002) Use of proteomic patterns in serum to identify ovarian cancer. The Lancet 359:572–577
55. Lymphoma/Leukemia Molecular Profiling Project, https://llmpp.nih.gov/lymphoma/
56. Mesejo P, Pizarro D, Abergel A, Rouquette O et al (2016) Computer-aided classification of gastrointestinal lesions in regular colonoscopy. IEEE Trans Med Imaging 35(9):2051–2063
57. Tsanas A, Little MA, Fox C, Ramig LO (2014) Objective automatic assessment of rehabilitative speech treatment in Parkinson's disease. IEEE Trans Neural Syst Rehabil Eng 22:181–190
58. Mahé P, Arsac M, Chatellier S, Monnin V et al (2014) Automatic identification of mixed bacterial species fingerprints in a MALDI-TOF mass-spectrum. Bioinformatics 30(9):6–1280
59. UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/index.php
60. Johnson B, Xie Z (2013) Classifying a high resolution image of an urban area using super-object information. ISPRS J Photogramm Remote Sens 83:40–49
61. Ciarelli PM, Oliveira E (2009) Agglomeration and elimination of terms for dimensionality reduction. In: Ninth international conference on intelligent systems design and applications, pp 547–552
62. Kushmerick N (1999) Learning to remove internet advertisements. In: Proceedings of the 3rd international conference on autonomous agents, ACM, pp 175–181
63. Boutell MR, Luo J, Shen X, Brown CM (2004) Learning multi-label scene classification. Pattern Recognit 37(9):1757–1771
64. Higuera C, Gardiner KJ, Cios KJ (2015) Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. PLoS ONE 10(6):e0129126
65. Weka 3: Data Mining Software in Java, https://www.cs.waikato.ac.nz/ml/weka/
66. Statnikov A, Wang L, Aliferis CF (2008) A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinform 9:319
67. Bhattacharyya S, Jha S, Tharakunnel K, Westland JC (2011) Data mining for credit card fraud: a comparative study. Decis Support Syst 50:602–613
68. Rao RS, Pais AR (2018) Detection of phishing websites using an efficient feature-based machine learning framework. Neural Comput Appl 1:2–9. https://doi.org/10.1007/s00521-017-3305-0
69. Zhu HJ, Jiang TH, Ma B, You ZH, Shi WL, Cheng L (2017) HEMD: a highly efficient random forest-based malware detection framework for Android. Neural Comput Appl 1:2–8. https://doi.org/10.1007/s00521-017-2914-y
70. Breiman L (2001) Random forests. Mach Learn 45:5–32
71. Rokach L (2016) Decision forest: twenty years of research. Inf Fusion 27:111–125
72. Fawcett T (2003) ROC graphs: notes and practical considerations for researchers, Technical Report, HPL-2003-4, HP Laboratories
73. Cannas LM, Dessì N, Pes B (2010) A filter-based evolutionary approach for selecting features in high-dimensional micro-array data. In: Proceedings of the 6th international conference on intelligent information processing (IIP 2010), Springer, Berlin, pp 297–307
74. Dessì N, Pes B (2015) Similarity of feature selection methods: an empirical study across data intensive classification tasks. Expert Syst Appl 42(10):4632–4642

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.