Bolón-Canedo, V. and Alonso-Betanzos, A. (2019) 'Ensembles for Feature Selection: A Review and Future Trends', *Information Fusion*, 52, pp. 1–12. doi:10.1016/j.inffus.2018.11.008.

Link to published version: https://doi.org/10.1016/j.inffus.2018.11.008

Ensembles for feature selection: A review and future trends

Verónica Bolón-Canedo and Amparo Alonso-Betanzos

Department of Computer Science - University of A Coruña
Campus de Elviña s/n 15071 - A Coruña, Spain

Abstract

Ensemble learning is a prolific field in Machine Learning since it is based on the assumption that combining the output of multiple models is better than using a single model, and it usually provides good results. Normally, it has been commonly employed for classification, but it can be used to improve other disciplines such as feature selection. Feature selection consists of selecting the relevant features for a problem and discard those irrelevant or redundant, with the main goal of improving classification accuracy. In this work, we provide the reader with the basic concepts necessary to build an ensemble for feature selection, as well as reviewing the up-to-date advances and commenting on the future trends that are still to be faced.

Keywords: Ensemble Learning, Feature Selection

1. Introduction

Ensemble learning is based on combining multiple models instead of a single model to solve a particular problem, and it is founded on the old proverb "two heads are better than one". The rationale is based upon the idea of building a set of hypothesis using different methods, and then they are combined trying to obtain better results than learning only one hypothesis with a single method [1]. It is the diversity of the approaches and the control of the variance which make this approach successful, also called as "committees".

In the field of machine learning, the typical approach consists in using a single learning model to solve a problem. However, the use of ensemble learning (i.e. using multiple prediction models for solving the same problem), has proven its effectiveness over the last years. In particular, ensemble learning has been very popular for classification; in fact there is a series of workshops on Multiple Classifier Systems (MCS) run since 2000 by Fabio Roli and Josef Kittler.

The most popular approaches for ensemble learning are bagging and boosting, both of them based on introducing diversity by modifying the training set, in such a way that the learning algorithm is executed multiple times over different training sets. The main difference between these models is that bagging

does a random sampling of the data with replacement, whilst boosting performs a random sampling with replacement on weighted data, in which these weights are iteratively updated trying to give more importance to the samples that have been previously misclassified. A very popular ensemble is also Random Forest, which is a special type of bagging combined with tree models, adding the particularity that the trees are built from different random subsets of features.

However, the idea of ensemble learning is not only applicable to classification, but it can can be used to improve other machine learning disciplines such as *feature selection*. It is common to have to deal with datasets containing a large number of features, which is an interesting challenge because classical machine learning methods cannot deal efficiently with high dimensionality. Therefore, it is typical to apply a preprocessing step to remove irrelevant features and reduce the dimensionality of the problem. 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 are some other benefits associated with a smaller number of features: a reduced measurement cost and hopefully a better understanding of the domain [2].

There are two typical ways of categorizing feature selection methods. On the one hand, it depends on the outcome of the feature selector: whether it returns a subset of relevant features or an ordered ranking of *all* the features , according to their relevance (known as feature ranking). In this latter case, it is necessary to establish a threshold in order to reduce the dimensionality of the problem, which is not an easy-to-solve question, as we will see in Section 3.3. On the other hand, feature selection methods are typically divided into three major approaches according to the relationship between a feature selection algorithm and the inductive learning method used to infer a model [3]: *filters*, which rely on general characteristics of the data and are independent of the induction algorithm; *wrappers*, which use the prediction provided by a classifier to evaluate subsets of features; and *embedded methods*, which perform FS in the process of training and are specific to given learning machines.

There exists a vast body of feature selection methods in the literature, including filters based on distinct metrics (e.g. entropy, probability distributions or information theory) and embedded and wrapper methods using different induction algorithms [4]. The proliferation of feature selection algorithms, however, has not brought about a general methodology that allows for intelligent selection from existing algorithms. In order to make a correct choice, a user not only needs to know the domain well, but also is expected to understand technical details of available algorithms. Ensemble feature selection can be a solution for the aforementioned problem since, by combining the output of several feature selectors, the performance can be usually improved and the user is released from having to chose a single method. The goal of this paper is to offer a comprehensive review of ensemble learning in the field of feature selection.

Ensembles for feature selection can be classified into homogeneous (the same base feature selector) and heterogeneous (different feature selectors). Both approaches have successful examples in the literature, and this paper will review the most recent ones. In addition to this, it is important to pay attention to the

combination step, in which the joining of the individual outputs produced by each feature selector should be carried out. These outputs can be in the form of subsets of features or ranking of features, and specific combination strategies are needed accordingly. Other aspects that can be also interesting for the reader are covered in this work, such as how to evaluate the performance of the ensembles in terms of diversity and stability, or a guide with software tools including implementations of feature selection ensembles.

The remainder of this paper is structured as follows. Section 2 states the foundations of ensembles for feature selection, commenting on the different types available. Then, Section 3 delves into the combination step, a crucial part when having multiple models. After the ensemble is built, it is necessary to evaluate its performance, which is addressed in Section 4. Section 5 surveys the recent works using ensembles for feature selection and Section 6 provides the reader with a review of some popular software tools that include useful implementations for ensembles for feature selection. Finally, Section 7 closes this work with the new challenges that researchers need to face in this field.

2. Types of ensembles for feature selection

The motivation under the use of the ensemble approach for learning has been recently extended to other machine learning fields, such as feature selection. Thus, the idea is that combining the outputs of several single feature selection models will obtain better results than using a single feature selection approach. But this improvement does not come only from having several models, as it is also the case with classification ensembles, but also from the diversity of the feature subsets obtained.

The scientific literature considers relations between the ensemble paradigm and feature selection in two different schemes: (1) using a feature selection preprocessing in order to produce the diversity that will be needed for subsequent ensemble methods, as in [5, 6]; or (2) using ensembles of feature selectors aiming to improve the stability of the process, as in [7, 8, 9, 10, 11, 12, 13]. This latter aspect is specially relevant in knowledge discovery, and even more in those cases in which data dimensionality is very high, but the number of samples is not such, as they are more sensible to generalization problems. Thus, several feature selection processes are carried out (either using different training sets, different FS methods, or both), and their results are aggregated to obtain a final subset of features that hopefully will add stability and thus be more transparent in the process of knowledge discovery. The idea is that a more appropriate (stable) feature subset is obtained by combining the multiple feature subsets of the ensemble, as the aggregated result tends to obtain more accurate and stable results, reducing the risk of choosing an unstable subset.

If several FS methods are used, the individual selectors in an ensemble are named, by analogy with the base learners, *base selectors*. Figure 1, shows the different levels than can be used to construct different types of ensembles for feature selection, that is, using different combination methods, using different

base learners, using different feature subsets or using different subsets of the original dataset.

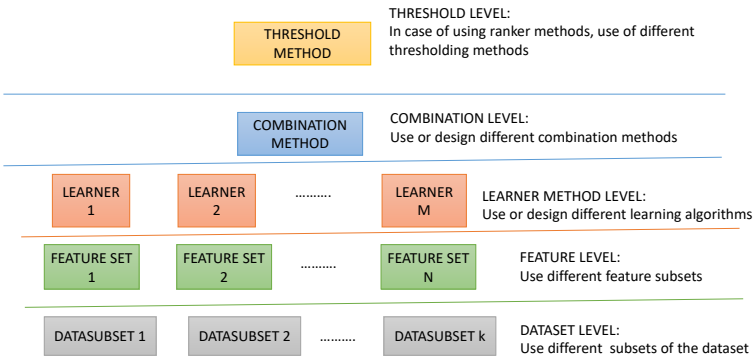

Figure 1: The different levels than can be varied in an ensemble for feature selection

In general, when aiming at designing a feature selection ensemble several main decisions are to be taken:

- The individual FS methods to be used. Three types of methods are available: wrappers, filters and embedded [2, 14, 1, 3]. As using more than one FS method has inevitably a computational cost, filters and embedded methods are preferred over wrappers for being included in an ensemble. Each individual methods has its pro and cons, and the methods employed should guarantee diversity while increasing the regularity of the FS process, so as to take advantage of them to boost performance. In [15, 16], some metrics for stability are discussed, and in [9] others for diversity are employed over rankers.

- The number of different FS methods to use. As stated above, there is a need to balance complexity, diversity and stability of the process. In the case of ensembles for classification, studies addressing the need of a priori determination of ensemble size are scarce [17, 18, 19], with results suggesting that using as many individual methods as class labels is the best option. However, in the case of feature selection ensembles those studies have not been addressed as yet, and thus at present statistical tests are mainly used for determining the best number of components.

- The number and size of the different training sets to use. Regarding these two parameters, there are some studies in the literature aiming at determining the optimal size of the training dataset again for ensembles for classification and prediction purposes [20, 21, 22, 23]. Again, in the

case of feature selection there are no reported studies on the size of the optimal training sets for ensembles, although some authors have studied the consequences of distributing the training set regarding the number of features and using ranker methods [24].

- The aggregation (also named combination) method to use. Different methods are available (see Section 3), and the scientific literature have explored those combination methods [25, 9, 26, 27], and also different strategies implying linear and non linear weighting of the base classifiers [28, 29], using genetic algorithms [30], their relation with the base classifiers chosen [31], etc. Most of the previous works have dealed with ensembles for classification, and only [27, 26, 9] have investigated the behavior of different aggregation methods for ensembles for feature selection.

- The threshold method to use if the FS methods are rankers, that is, if the methods return an ordered list of all features involved in the problem. For most studies, the thresholds chosen are based on a fixed percentage of retained features, for example 25%, 50%, or a fixed number of top features [14, 32, 33, 26]. Other authors have tried to derive a threshold based on different metrics. In [12], in which tree ensembles are used, a feature importance measure, that is derived as the average information gain achieved during tree construction, is employed. Other authors have used thresholds based on data complexity measures [26, 27].

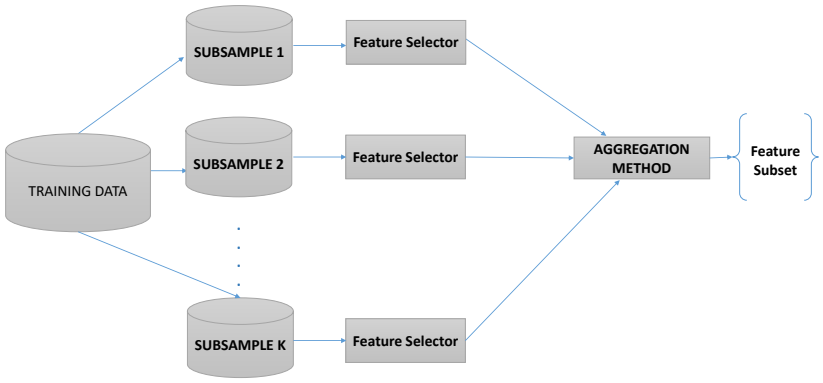

Figure 2: A scheme of the Homogeneous approach

Ensembles for feature selection might be classified following diverse criteria regarding any or several of the aspects above, but the most simple division is regarding the type of base selectors used. If the base selectors are all of the

same kind, the ensemble is known as *homogeneous*; otherwise the ensemble is *heterogeneous*.

In the homogeneous approach, the same feature selection method is used, but with different training data subsets. These data subsets may be distributed over several nodes (or several partitions), and thus in this case a reduction of temporal requirements is also achieved (see a scheme in Figure 2). In this scheme, the size of the partitions is also a design parameter. These methods are also named as data variation ensembles. Some examples of homogeneous approaches, mainly with the aim of being able to manage large scale scenarios, can be found in [26, 34], and in [35], this latter with the added goal of being able to deal with imbalanced data sets.

For the heterogeneous approach, a number of different feature selection methods, but over the same training data, are applied, as can be seen in Figure 3. In this scheme, the number of different feature selectors to be used is also a design parameter, as stated above. These methods are also called function variation methods.

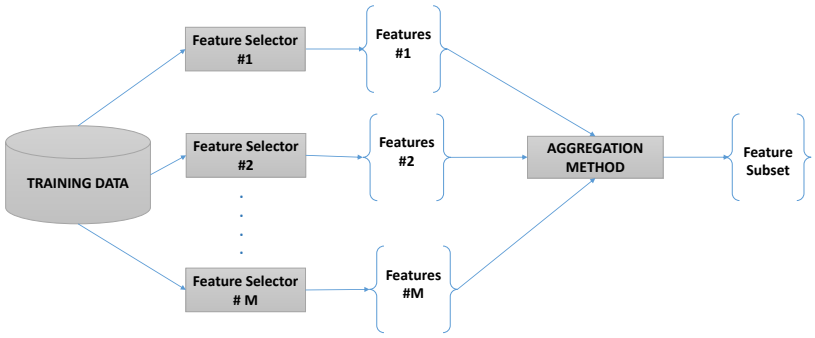

Figure 3: A scheme of the Heterogeneous approach

Heterogeneous feature selection ensembles are more common than homogeneous, and several examples can be found in [36, 32, 26, 9, 27, 37, 30]. In some of these studies both, homogeneous and heterogeneous approaches are compared, as in [37, 26, 38].

In both schemes, depending on the type of feature selectors employed, one can obtain as a result a feature subset or a feature ranking, and in this last case, an additional threshold step is needed. Finally, and as in all ensembles, the results of the base selectors are to be combined to obtain a final result, and thus several aggregation methods (see section 3 for a description of the different combiners available) can be used. Some recent works [27] have also explored different designs exchanging the order of the combination and thresholding steps

when rankers are used as base feature selectors.

3. Combination of outputs

A crucial point in any ensemble scheme is the combination of partial results to obtain a final output. In the case of feature selection ensembles, the typical situation is to combine the different features selected by the different selectors (see Figures 2 and 3, for example). Another possibility is to apply a classifier after each feature selector, so it is necessary to combine the label predictions of the classifiers. Both approaches will be discussed in this section.

3.1. Combination of label predictions

Combining the outputs of the individual classifiers has been broadly studied since it is necessary when designing an ensemble of classifiers [39]. The output produced by a classifier can be just a class label (without information about the certainty of the guessed labels) or a degree of certainty of the prediction (e.g. probability of belonging to a given class).

Depending on the type of classifier outputs, different methods for combining the outputs can be used. When having classifiers that only return the class labels, the most popular technique is *majority vote*, which consists of establishing the final output as the option that has been predicted by the majority of the classifiers. Although widely used, it has some limitations, as for example how to deal with ties, which are usually resolved arbitrarily.

If the classifiers used return also a degree of certainty, there are more sophisticated decision rules that can be applied [40]. Let us suppose that we have a classification problem in which instance x is to be assigned to one of the C different classes of the problem $c_1, c_2, ..., c_C$. Consider that we have N classifiers which will lead to N outputs $y_i, i = 1, ..., N$ to make the decision. When the classifiers provide a degree of certainty, the posterior probability can be estimated as $P(c_j|x) = y_i$, where y_i is computed as the response of a classifier i. Now, let us denote $y_{ij}(x)$ as the output of the classifier i in the class j for the instance x and assuming that the outputs y_i are normalized. Some popular decision rules can be defined as follows:

- Product rule, $x \rightarrow c_j$ if

$$\prod_{i=1}^{N} y_{ij}(x) = \max_{k=1}^{C} \prod_{i=1}^{N} y_{ik}(x)$$

- Sum rule, $x \rightarrow c_j$ if

$$\sum_{i=1}^{N} y_{ij}(x) = \max_{k=1}^{C} \sum_{i=1}^{N} y_{ik}(x)$$

- Max rule, $x \rightarrow c_j$ if

$$\max_{i=1}^{N} y_{ij}(x) = \max_{k=1}^{C} \max_{i=1}^{N} y_{ik}(x)$$

  This rule approximates the sum rule assuming that the output classes are a priori equiprobable. The sum will be dominated by the prediction which lends the maximum support for a particular hypothesis.

- Min rule, $x \rightarrow c_j$ if

$$\min_{i=1}^{N} y_{ij}(x) = \max_{k=1}^{C} \min_{i=1}^{N} y_{ik}(x)$$

  This rule approximates the product rule assuming that the output classes are a priori equiprobable. The product will be dominated by the prediction which have the minimum support for a particular hypothesis.

- Median rule, $x \rightarrow c_j$ if

$$\frac{1}{N}\sum_{i=1}^{N} y_{ij}(x) = \max_{k=1}^{C} \frac{1}{N}\sum_{i=1}^{N} y_{ik}(x)$$

3.2. Combination of subsets of features

As mentioned in the Introduction, feature selection methods can be classified based on if their output is a subset of features or an ordered ranking of all the features. This subsection will be focused on the case of having different feature selectors that return subsets of features and we need to combine them before classification (given that classification is the final goal of our system).

The most typical way to combine subsets of selected features is to compute the intersection and the union of them. The intersection consists in selecting only those features which are selected by *all* the feature selectors. Although this approach might seem very logical (if a feature is selected by all selectors, it must be highly relevant), it can lead to very restrictive sets of features (the empty set, in the worst case scenario) and in practice it does not produce good results [41].

The union consists in combining all the features which have been selected by at least one of the feature selectors. Contrary to the intersection, it can lead to select even the whole set of features. This approach tends to produce better results than the intersection [41], but at the expense of a lower reduction in the number of the features.

A more sophisticated technique is to use the classification accuracy to combine the subsets of features returned by the different selectors. A simple approach is to include a subset of features into the final selection only if it contributes to improve classification performance [42]. The first subset of features S_1 is arbitrarily taken to calculate the classification accuracy, which will be the *baseline*, and the features in S_1 will always become part of the final selection

S. For the remaining selections, the features in $S_i, i = 2...n$ will become part of the final selection S if they improve the baseline accuracy. The authors expect that combining the features in this manner can help reduce redundancy, since a redundant feature will not improve the accuracy and hence will not be added to the final selection.

The main problem of using classification performance to combine subsets of features is that it requires a high computational cost, which in some cases can be even higher than the time necessary for the feature selection process. Trying to solve this issue, Morán-Fernández et al. [43] proposed to combine the subsets of features using data complexity measures instead of classification performance. The reason for this decision was that they assume that good candidate features would contribute to decrease the theoretical complexity of the data and must be maintained.

3.3. Combination of rankings of features

In the previous subsection, we have seen how to combine the results obtained by the weak selectors when their output is a subset of features. But, as seen in the Introduction, there are feature selection methods that return an ordered ranking of all the features, according to their relevance. In this case, it is necessary to find methods that can receive as an input several ranking obtained by the different feature selectors and combine them into a single final ranking, trying not to incur in an important loss of information. Depending on the ensemble approach, it is possible that all the feature selectors rank all the features, or only a subset of them.

The easiest way to combine rankings of features is to apply simple operations through them, such as the median or the mean. Some popular methods are defined in the following:

- **min**: assigning to each element to be ranked the minimum (best) position that it has achieved among all rankings.
- **median**: assigning to each element to be ranked the median of all the positions that it has achieved among all rankings.
- **arith.mean**: assigning to each element to be ranked the mean of all the positions that it has achieved among all rankings.
- **geom.mean**: assigning to each element to be ranked the geometric mean of all the positions that it has achieved among all rankings.

More sophisticated methods can be found in the literature. For example, Stuart et al. [44] introduced the first attempt to use order statistics in the combination of rankings, although the computational scheme for their method was further optimized by Aerts et al. [45]. This method works by comparing the actual rankings with the expected behavior of uncorrelated rankings, and then re-ranks the features and assigns significance scores. Despite being robust to noise, this method requires simulations to define significance thresholds and

does not support partial rankings (i.e. rankings which do not contain all the features).

Robust Rank Aggregation [46] was then proposed to improve the limitations of Stuart and other classical methods. In this case, the combination is based on the comparison of the actual ranking with a null model that assumes random order of the different obtained rankings. A P-value assigned to each feature in the aggregated ranking described how much better it was ranked than expected. This provides basis for reordering and identifies significant features. As the P-value calculation procedure takes into account only the best ranks for each feature, the method is said to be very robust. Finally, we can also find SVM-Rank [47], which is a SVM-based method that can be trained to learn ranking functions.

4. Evaluation of ensembles

Performance is the universal measure to evaluate a learning system, which in the case of classification is usually measured as accuracy in the prediction. However, in the case of ensembles, there are other factors that have relevance in this process, such as diversity and stability. On the one hand, we need to use in the ensemble single methods that produce *diverse* results. But, on the other hand, we need ensembles that are *robust*. So far, and although measures for diversity and stability in classifier ensembles have been devised, the subjects are still rare for the case of feature selection ensembles.

4.1. Diversity

Diversity is one of the main reasons to use an ensemble method, as in the case of classification the examples that are misclassified by some members of the ensemble have the chance to be correctly classified by other, so that the final accuracy is improved. This is the reason why diversity among the members of the ensemble is a key issue–it makes no sense to build an ensemble in which all the single methods offer the same result.

But, how can we be sure that we are using *diverse* methods? There are several statistics that can be used as a measure of diversity. Kuncheva and Whitaker [48] recommended the pair-wise Q statistics [49], as it is simple to understand and to implement. Although there are several works regarding diversity in ensembles for classification [50, 48, 51], there is a necessity for the establishment of novel diversity measures for ensembles for other machine learning algorithms, as feature selection [8]. And not only diversity is important, but also the function that combines the results of the different components of the ensemble (see Section 3). Brodley et al. [52] showed that diversity in the feature subsets created alone is not enough for increasing the accuracy of the machine learning process, as the combination method should also make proper use of the diversity obtained in order to maintain the benefit.

4.2. Stability

As mentioned above, it is desirable that the methods conforming the ensemble are *diverse*, i.e. they provide different enough outputs on the same sample of data. However, when the sample of data changes, it is also desirable that these methods return similar outputs, a property which is known as *stability*.

As pointed out by Nogueira & Brown [15], in ensemble-based feature selection, the goal must be to use diverse feature selection methods within the ensemble, as well as obtaining robustness of the final feature selection made by the ensemble (corresponding to high stability). Therefore, the stability of ensembles for feature selection has been gaining attention in recent years [53, 54, 55, 7].

There are plenty of measures in the literature to compute stability. If our goal is to measure stability among methods that return a subset of features, probably the two most famous metrics are Jaccard index [56] (also referred as Tanimoto distance) or the relative Hamming distance [57]:

$$Jac(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| + |A \cap B|}. \tag{1}$$

$$Ham(A,B) = 1 - \frac{|A \setminus B| + |B \setminus A|}{n}. \tag{2}$$

However, both these measures are subset-size-biased, which means that they provide different results depending on the number of features selected so they cannot be considered consistent. For this reason, Kuncheva [16] proposed a consistency index to measure stability that solves this problem:

$$Kun(A,B) = \frac{f - \frac{k^2}{n}}{k - \frac{k^2}{n}} = \frac{rn - k^2}{k(n-k)}, \tag{3}$$

such that $|A| = |B| = k$ and where $0 < k < |X| = n$.

The problem with this stability measure is that it requires that subset sizes are the same, which in practice does not always happen, and new variants of Kuncheva's similarity measure for feature sets of varying cardinalities have been appearing in the last few years. For more information about stability measures, please refer to the work by Nogueira & Brown [15].

Among the most popular measures to compute the similarity between rankings of features we can find the Kendall Tau [58], the Canberra Distance [59] and the Spearman's ρ [56]. Let R_1 and R_2 be two rankings and f the number of features in the dataset, these measures can be defined as follows:

$$Spear(R_1,R_2) = 1 - \frac{6 \sum d^2}{f(f^2-1)}, \tag{4}$$

where d is the distance between the same feature in both rankings.

$$Cam(R_1,R_2) = \sum_{i=1}^{f} \frac{|R_{1_i} - R_{2_i}|}{|R_{1_i}| + |R_{2_i}|} \tag{5}$$

$$Kend(R_1, R_2) = \sum_{\{i,j\} \in P} \bar{K}_{i,j}(R_1, R_2) \tag{6}$$

where

P is the set of unordered pairs of distinct elements in R_1 and R_2
$\bar{K}_{i,j}(R_1, R_2) = 0$ if i and j are in the same order in R_1 and R_2
$\bar{K}_{i,j}(R_1, R_2) = 1$ if i and j are in the opposite order in R_1 and R_2

4.3. Performance

After making sure that we have an ensemble of feature selectors that are diverse among them and stable to variations in the data, we need to check if the final selection of features is relevant. In an ideal situation, a feature selection system should be evaluated based only on the quality of the features selected, without involving any classifier. But, in practice, the set of relevant features are not known a priori unless we are using artificial data. In fact, several authors choose to use artificial data stating that 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.

If we use artificial data and then we know the relevant features, there are several measures we can use to evaluate the performance of the ensemble, depending on if the ensemble returns a subset of features or a ranking of features.

In the case of subsets of features, we proposed several measures in a previous work, provided that we know a priori the relevant ones [60]. For the description of the methods, note that $feat\_sel$ stands for the subset of selected features, $feats$ is the total set of features, $feat\_rel$ is the subset of relevant features, and $feat\_irr$ represents the subset of irrelevant features (the last two known *a priori*).

- The *Hamming_loss* (H) measure evaluates how many times a feature is misclassified (selected when is irrelevant or not selected when is relevant)

$$H = \frac{\#(feat\_sel \cap feat\_irr) + \#(feat\_not\_sel \cap feat\_rel)}{\#(feat\_rel \cup feat\_irr)}$$

- The *F1-score* is defined as the harmonic mean between precision and recall. *Precision* is computed as the number of relevant features selected divided by the number of features selected; and *recall* is the number of relevant features selected divided by the total number of relevant features. Therefore, the F1-score can be interpreted as a weighted average of the precision and recall. Considered $1 -$ F1-score, it reaches its best value at 0 and worst score at 1.

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

In the case of ensembles that return a ranking of all the features, the measures described above are not useful because all the features are present in the ranking. A possible solution is to establish a threshold and transform the ranking in a subset of features. But there are also methods specifically defined to evaluate rankings, which in essence check if the relevant features are ranked above the irrelevant ones. Below we describe some popular ones [60]:

- The *ranking_loss* (R) evaluates the number of irrelevant features that are better ranked than the relevant ones. The fewer irrelevant features are on the top of the ranking, the best classified are the relevant ones. Notice that *pos* stands for the position of the last relevant feature in the ranking.

$$R = \frac{pos - \#feat\_rel}{\#feats - \#feat\_rel}$$

- The *average_error* (E) evaluates the mean of E_i, in which $i \in feats\_sel$ and E_i is the average fraction of relevant features ranked above a particular feature i.

$$E_i = \frac{\sum_j feat\_sel(j) \in feat\_rel \cap j<i - \frac{\#feat\_rel \times (\#feat\_rel - 1)}{2}}{\#feat\_irr \times \#feat\_rel}$$

5. Recent advances on ensembles for feature selection

The crescent digital transformation of our society has originated a explosion of data that increases in both size and dimension. Machine learning is one of the techniques that is being used for obtaining information and knowledge from Big Data. A large number of features (dimension) usually implies that a certain amount of them are redundant or irrelevant, and their presence increases the error of the learning algorithms. Thus, feature selection is almost a mandatory preprocessing step in order to reduce the data dimensionality.

There are different feature selection algorithms available [3], but as they rely on different metrics and approaches, the feature subsets obtained are also different, configuring different local optima in the space of feature subsets. The rationale for the use of ensembles in the feature selection process is thus clear [7], as by generating many predictors, the solution space can be massively explored and by later combining all individual results, the ensemble is able to reflect this exploration, so as to obtain a more robust final feature subset regarding not only performance but also stability. There are two schemes that can be find in the literature aiming at using the idea of ensembles in feature selection: some authors have used a previous feature selection step in order to obtain the diversity needed for using posterior ensemble classification methods, such as in [5, 6]; the other, and the one in which this review is centered, is using ensembles of feature selectors for improving accuracy, diversity and stability of the feature selection process [7, 8, 9, 10, 11, 12, 26]. This last scheme is of special interest in knowledge discovery scenarios, and mainly in high dimensional cases (a much larger number of characteristics than samples), due to overfitting of machine

learning algorithms. In [8], five different pairwise measures of diversity were compared over 21 datasetss regarding their use for ensemble feature selection for ensembles of fixed sizes. The study employed as search strategies forward and backward sequential selection, genetic search and the classical hill-climbing. The main idea was to design a fitness function that could reflect both accuracy and diversity. The results obtained showed that there is a close relation between the functions employed and the number of ensemble members needed to achieve the highest accuracy. Finally, a detailed analysis of the optimal ensemble size for the different diversities and search strategies in ensemble feature selection was proposed for future research. Other works, as [32] proposed an ensemble formed by a fixed number of filters for being employed in high dimensional scenarios, such as microarrays. Two studies were carried out in order to select the specific base selectors that were employed, as well as their number. First, synthetic data were used to check whether the individual feature selection methods were able to select adequately the relevant features and discarding the irrelevant ones in complex scenarios. Once a set of base selectors was pre-defined, the second study assessed their stability, defined as the sensitivity of a method to variations in the training set. Two different basic schemes are proposed in [36], both of the heterogeneous type. The first one uses 5 filters that fed five classifiers followed by the aggregation step, while the second proposal uses the same filters followed by the aggregation step, that is previous to classification.

Other works address specifically the use of ensembles to improve not only accuracy, but also stability of the results obtained. In [61], the authors develop a new algorithm named *Multicriterion Fusion-based Recursive Feature Elimination* aimed at increasing robustness of feature selection algorithms by using multiple feature selection evaluation criteria. The idea was to be able to work in high-dimensional scenarios, but with low number of samples, as it happens in the case of microarray datasets. These same dataset types were confronted in [36]. Another study restricted the study to a type of ensembles, those using as base selectors Multi-layer perceptrons [62]. In this case, the proposal employed a feature ranking scheme, with a stopping criterion based on the *Out-of-Bootstrap (OOB)* estimate [63].

As stated before, ensembles can be composed by different base selectors. As diversity is one of the important characteristics to emphasize, some ensembles use feature selectors of several types [2](rankers and subset methods; filters, wrappers and embedded methods; and among them also univariate and multivariate methods), as in [36, 9]. However, another set of works achieved as well diversity but employing only one of the types of feature selection methods, in this case rankers. Three filter rankers with simple combining methods (lowest, highest, and average rank), were used in [64]. In [65, 66] several ensembles of filter rankers, employing a variety of thresholds to select the final subset of features, were applied to the area of software quality. Other studies describe different methods for combining individually generated rankings, with the aim of obtaining an adequate final ensemble. The combination of individual rankings covers from simple methods, based on computing the mean, median, minimum, etc., to more complex methods like *Complete Linear Aggregation* [67, 9] *(CLA)*,

Robust Ensemble Feature Selection (Rob-EFS) [68], *SVM-Rank* [9, 26], or the use of data complexity measures [27].

The works above focus on the increase of stability, but although ensembles have shown to be able to address this problem, they usually achieved it with the common drawback of also increasing the running times of the procedure, and thus limiting their application due to scalability issues, mainly in the sample size. In [69], two methods which enhance correlation-based feature selection such that the stability of feature selection comes with little or even no extra runtime were devised. Another idea exploits the heterogeneous type of ensemble with a parallel application of the multiple feature selection methods.

Although there are available several parallel and distributed implementation of individual feature selection methods [70, 71, 72, 73], only a few research works developed ensembles making use of distributed or parallel schemes. An heterogeneous approach is proposed in [9], with the idea of distributing the dataset in several nodes, and then apply the same feature selection method in each of them, aggregating later the results. Similar ideas, making use of distributing the datasets, are proposed in [42, 43], analyzing different partitioning strategies (vertical–by features–, and horizontal–by samples). The combination of partial outputs is also analyzed to achieve a final recommendation in terms of selected features, accuracy and running times. While the more common combination strategies on these type of ensembles are based on classifier accuracy, as in [42], or in combinations of classification performance and reliability assessment as in [38], also some new proposals based on data complexity are explored in [43, 27], achieving high accuracies while reducing considerably the computational time.

Usually, feature selection is performed in a supervised manner (i.e. all the training samples are labeled), and so are the ensembles revised in this sec-tion. However, there are cases in which the samples could not be labeled, a case known as *unsupervised learning*. Some typical algorithms that deal with unsupervised learning are clustering and anomaly detection methods, among others. Although not very common, there are also feature selection methods that can work with unsupervised data and also some ensembles. Related to clustering, we can find a work [74] in which the authors show that the way that internal estimates are used to measure the variable importance in Ran-dom Forests are also applicable to feature selection in unsupervised learning, and they proposed a new method called Random Cluster Ensemble that esti-mates the out-of-bag feature importance from an ensemble of partitions. Hong et al. [75] also presented a novel feature selection algorithm for unsupervised clustering, which combines the clustering ensembles method and the population based incremental learning algorithm. The same authors also addressed the challenging task of feature ranking for unsupervised clustering [76] for guiding the computations of the relevances of features. They proposed a novel consen-sus unsupervised feature ranking approach which obtains multiple rankings of all features from different views of the same data and then aggregates all the obtained feature rankings into a single consensus one. A different approach is followed in the work by Morita et al. [77], in which they proposed an ensemble of classifiers based on unsupervised feature selection. It takes into account a

hierarchical multi-objective genetic algorithm that generates a set of classifiers by performing feature selection and then combines them to provide a set of powerful ensembles.

Semi-supervised learning falls between unsupervised learning and supervised learning, since a small amount of data is labeled but the training set contains a large amount of unlabeled data. A few ensemble methods try to deal with this situation. Grabner et al. [78] presented a novel online boosting method which formulated the updated process in a semi-supervised fashion as combined decision of a given prior and an online classifier. Later on, Bellal et al. [79] proposed a new method called semi-supervised ensemble learning guided fea-ture ranking method (SEFR) that combined a bagged ensemble of standard semi-supervised approaches with a permutation-based out-of-bag feature im-portance measure taking into account both labeled and unlabeled data. A new wrapper-type semi-supervised feature selection framework that can select the relevant features using confident unlabeled data has been proposed by Han et al. [80]. They employ an ensemble classifier that supports the estimation of the confidence of the unlabeled data. Finally, in [13], the authors propose a new semi-supervised feature evaluation method named OFFS (Optimized co-Forest for Feature Selection) combining ideas from co-forest and from the embedded principle of selecting in Random Forest based on the permutation of out-of-bag set.

6. Software tools

When building an ensemble for feature selection, it is necessary to implement the feature selectors and also the distribution and combination of the data. This can be done from scratch or use already implemented methods. There are plenty of feature selection algorithms available in popular frameworks, which usually also offer facilities to distribute and combine the data in an ensemble scheme. Although not so common, there are some platforms that provide implementations for ensembles for feature selection.

Matlab[1] provides some methods for feature selection in its *Statistics and Machine Learning* toolbox, such as ReliefF or sequential feature selection. Moreover, in the same toolbox, there is a framework for ensemble learning. It provides a method for classification, `fitcensemble`, and for regression, `fitrensemble`. It allows the user to control parameters such as the aggregation method, the number of ensemble learning cycles and the weak learners. There is also the option to use the function `predictorImportance` which, used together with an ensemble, computes estimates of predictor importance by summing these estimates over all weak learners in the ensemble, where a higher value means a more important feature.

In Weka (Waikato Environment for Knowledge Analysis) [81] there is a wide suite of feature selection algorithms available, including Correlation-Based Fea-

[1] `https://www.mathworks.com/products/matlab.html`

ture Selection, Consistency-based, Information Gain, ReliefF, or SVM-RFE, just to name a few. Moreover, it provides several methods for ensemble learning, such as AdaBoost, Bagging, RandomForest, etc.

R is a free programming language and software environment for statistical computing and graphics. There are several R-packages for feature selection, but probably the most famous ones are Caret[2] and Boruta[3]. There are also several packages available for ensemble learning, such as `adabag`[4], `randomForest`[5], or `gbm`[6]. Furthermore, one can find some works providing R packages for ensemble feature selection, such as that by Neumann et al [82]. They propose a software called EFS (Ensemble Feature Selection) available as R-package[7] and as a web application[8]. It makes use of eight feature selection methods and combines their normalized outputs to a quantitative ensemble importance. Another example, is mRMRe[9], an R package for parallelized mRMR ensemble feature selection. The two crucial aspects of the implementation they propose are the parallelization of the key steps of the algorithm and the use of a lazy procedure to compute only the part of the mutual information minimization (MIM) that is required during the search for the best set of features (instead of estimating the full MIM).

KEEL (Knowledge Extraction based on Evolutionary Learning) [83] is an open source Java software tool that provides the implementation of a large number of feature selection methods, as for example ReliefF, mutual information, or those based on genetic algorithms. It also includes several ensemble methods, as well as specific methods for ensembles for imbalanced data.

RapidMiner [84] is a data science software platform that provides several feature selection tools, including information gain, Gini index, chi-square, and others. It also features tools for ensemble learning, including popular methods such as baggins, boosting, Adaboost, etc. What it is more interesting is the possibility of obtaining a plugin, called *Feature Selection Extension*[10], which offers the Ensemble-FS operator for ensembles for feature selection. It loops several times over subsamples of the input sample. The inner feature selection operator chosen is performed each time, and the resulting attribute weights are averaged (or somewhat combined). Then, the robustness of the feature selection can be estimated by calculating the Jaccard-Index for the different subsets of selected features.

Scikit-learn [85] is a free software machine learning library for the Python programming language. It is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy, and includes several feature selection

[2] `https://CRAN.R-project.org/package=caret`
[3] `https://CRAN.R-project.org/package=Boruta`
[4] `https://CRAN.R-project.org/package=adabag`
[5] `https://CRAN.R-project.org/package=randomForest`
[6] `https://CRAN.R-project.org/package=gbm`
[7] `https://CRAN.R-project.org/package=EFS`
[8] `http://efs.heiderlab.de`
[9] `https://CRAN.R-project.org/package=mRMRe`
[10] `https://sourceforge.net/projects/rm-featselext/`

algorithms such as the popular mutual information, chi-square, L1-based feature selection or Tree-based feature selection. Apart from these algorithms already included in scikit-learn, there are other feature selection frameworks built upon it. It is particularly interesting *scikit-feature*[11], which is an open-source feature selection repository in Python developed at Arizona State University. It contains around 40 popular feature selection algorithms, including traditional feature selection algorithms and some structural and streaming feature selection algorithms. As for ensemble learning, it also offers several options (e.g. bagging, Random Forest, Adaboost, etc.).

Last but not least, several paradigms for performing parallel learning have emerged in the last years, such as MapReduce [86], Hadoop[12], or Apache Spark[13]. Developed within the Apache Spark paradigm was MLlib[14], created as a scalable machine learning library containing algorithms. It is more focused on learning algorithms, such as SVM and naive Bayes classification, k-means clustering, etc., but it also includes a few, very simple, feature selection algorithms, such as chi-square and ensemble methods such as Random Forest and Gradient-boosted trees. Moreover it is possible to find works in the literature that accelerate more sophisticated feature selection algorithms using these platforms. For example, in a previous work we have developed a distributed implementation of a generic feature selection framework using Apache Spark [72] (available on GitHub[15]). This framework includes well-known information theory-based methods such as mRMR, conditional mutual information maximization, or joint mutual information (JMI), that have been designed to be able to be integrated in the Spark MLlib library. Also, we have also proposed a Spark implementation of other popular feature selection methods such as ReliefF, SVM-RFE or CFS[16].

Apache Flink[17] is also an open-source stream processing framework for distributed, high-performing, always-available, and accurate data streaming applications with a library for machine learning, called FlinkML. However, as for now it does not include any feature selection or ensemble learning algorithms. As happens with Spark, it is possible to find works devoted to feature selection to work in Flink[18]. Another solution to make existing algorithms more scalable is the use of graphics processing units (GPUs) to distribute and thus accelerate calculations made in feature selection algorithms. In a previous work, we have redesigned the popular mRMR method to take advantage of GPU capabilities [73], showing outstanding results (available on GitHub[19]).

[11] `http://featureselection.asu.edu/index.php`
[12] `http://hadoop.apache.org/`
[13] `https://spark.apache.org`
[14] `https://spark.apache.org/mllib`
[15] `https://github.com/sramirez/spark-infotheoretic-feature-selection`
[16] `http://www.lidiagroup.org/index.php/en/materials-en.html`
[17] `https://flink.apache.org/`
[18] `https://github.com/sramirez/flink-infotheoretic-feature-selection`
[19] `https://github.com/sramirez/fast-mRMR`

7. Future trends

Ensembles for feature selection are relatively recent, appearing for the need of more accurate, robust and stable feature selection, a step that, if before Big data was already relevant, nowadays has converted in essential for Machine Learning pipelines. Feature selection has been applied successfully in different scenarios in which high dimensional datasets are present, such as DNA microarray analysis, image classification, face recognition, text classification, the so-called "-omics" sciences, etc. Ensembles of feature selection have been tried with the aim of achieving more accurate, robust and stable results in some of these areas, showing better results than individual FS methods [26, 32]. However, ensembles are also a hot line of research in other fields of Machine Learning:

7.1. Ensembles in other areas of Machine Learning

Ensembles in Machine Learning were first applied for classification and regression at the end of the 70s [87, 88]. Since then, ensemble learning has been a prolific field for researchers, that have investigated in many alternatives that have been proposed in classification, regression, preprocessing and other fields[89, 39, 90, 91, 92, 93], although there is not a clear and definite winner method, as in many other areas of Machine Learning. During the last years, ensembles have been extended beyond "classic" classification, regression and clustering to problems related to quantification [94, 95] or anomaly detection [96, 97, 98]. At the same time, in those initial classical fields, they have been applied to the new problems that arise, mostly, derived from the Big data phenomenon, such as streaming processing , supporting incremental learning [99, 100, 101, 102, 103], the problem of imbalanced data [104, 105], missing data [10, 106] or the need for distributed and parallel learning [107, 108, 109]. As it can be seen, all these publications are recent, paving the way for new research lines in the field of ensemble learning for the following years.

7.2. Fields of application

Regarding the areas of application of ensembles for feature selection mentioned above , the most recent trends are the following:

- Microarray datasets: In [110] an exhaustive review of the most recent feature selection algorithms that have been developed in the area of microarrays is presented. Due to the high demand of computational resources of the wrapper methods, those are the least employed, while filters based on information theory have been the preferred. The present tendency is towards algorithm combination in ensemble or hybrid schemes. Examples of this trend can be found in [111, 112, 113]. In [113], first a filter that is employed to reduce the number of genes, followed by a wrapper that works over an already reduced search space are employed. The features selected are evaluated using ROC curves and finally the most effective and smallest one is the one remaining. In [112], the authors propose an Ensemble Gene Selection by Grouping (EGSG), that employs Information

Theory and approximate Markov blankets, instead of a random selection, obtaining thus not only better accuracies, but also improving stability. An ensemble of four filter rankers is proposed in [111], and their results are aggregated with different combination methods. An interesting point of this proposal is the use of an automatic threshold based on dataset complexity measures.

- Image classification, in which feature selection has become a popular preprocessing step, as there is a call for efficient methods [114]. During the last years, classification ensembles have been increasingly used after a previous feature selection [115, 116, 117, 118, 119]. More recently, feature extraction ensembles, that aim at reducing dimensionality but not using a reduced set of the original features, have been applied by means of Deep Networks, as in the works described in [120, 121]. However, as feature selection aims at explanation and transparency by selecting those features relevant from the initial set of features, it constitutes an interesting line of open research.

- Face recognition, which specifically has attracted a lot of attention from the research community, due to its important commercial and legal applications. The task of selecting the features that are relevant for recognition purposes is far from trivial, as facial images datasets are scarce in samples, abundant in features, and redundancy is commonplace. As in the case of microarrays, filter feature selection methods have been the most popular, followed by classification ensembles, as in the works carried out in [122, 123, 124]. Feature selection ensembles have also been proposed in this area, as in [125, 126]. The "International Conference for Machine Learning" launched in 2012 a competition called "Ensemble Feature Selection in Face Recognition" , in which the winner [127] applied an ensemble that employed only around 1% of the features of the images, obtaining impressive accurate results. As in the general case of image classification, Deep Neural networks have improved performances, but explainability and transparency is lost, and thus ensemble feature selection for face recognition might be also an interesting open line of research.

- Text classification, another high dimensional problem area, as it aims to categorize documents into a fixed number of predefined categories, usually considering each word as a feature, and thus using more than an order of magnitude of features more than samples. For this reason, even a preprocessing that eliminates rare words and merges some word forms (verbs conjugations, plurals, etc) needs to be applied before feature selection [128, 129]. Recently, ensembles for feature selection have been also applied, as in [130, 131], obtaining better results in performance than those of the individual filter methods employed in the experiments.

As can be seen, several fields might benefit from the use of feature selection ensembles for preprocessing purposes, since they usually improve accuracy, while

boosting stability and reducing the computational costs of pattern recognition. The areas mentioned above have covered some of the more popular applications for feature selection, but the literature describes many more application areas, as diverse as intrusion detection [132, 133, 134, 135, 136], machinery fault diagnosis, [137, 138, 139, 140, 141, 142, 143], or automatic evaluation of open response assignments [144].

7.3. Open topics for ensemble design

Beyond the areas of application in which ensembles could make a statement, there are various general aspects related with the subject that are in need for further research:

- In-depth analysis of the optimal number of components in ensembles for FS. The large dimension, the need for better accuracies and the restrictions for computer time and memory call for approaches that can determine appropriate ensemble sizes. Such studies are relatively recent for classification ensembles [17, 145, 146], while in feature selection ensembles the size of the ensemble has been approached theoretically only in specific high dimensional domains as in [24], and most times empirically, as in [9, 26]. The relation between diversity and number of components might be also an interesting line of exploration.

- Stability, that for a FS algorithm defines the robustness of its feature preferences, with respect to data sampling and to its stochastic nature[15, 147]; that is, it quantifies the difference in the feature preferences obtained with different training sets derived from the same generating distribution [56]. Thus, if small changes in data produce large changes in the resulting features, the method is deemed as unstable. The development of stability measures has become a bountiful area of research, with several proposals along the years [57, 56, 16, 148, 149, 150, 151, 152, 153], but without any work that permits a comparison among them. The proposed measures comply with a certain number of properties, in some cases defined only for certain measurement categories, increasing the diversity on cross-comparisons and thus the difficulty on reaching stability conclusions. In [147] five properties that are applicable to any stability measure are proposed for the case of algorithms which output are feature subsets, allowing the analysis and comparison of all existing measures in terms of properties. However, the study of similar approaches for feature rankers still remains an open issue.

- Scalability measures. As said above, data is becoming larger increasingly, in both samples and feature dimensions, a fact that at the same time that makes feature selection desirable, poses a severe challenge to feature selection algorithms, as most can not confront scalability issues and thus new methods should be devised[154, 2, 155]. Several more scalable feature selection algorithms have been developed during the last years, following online [156, 100], or parallel and distributed strategies [157, 73, 72, 71].

But unlike stability, scalability studies are not common still in the scientific literature, despite the fact that evaluation of performance should probably take into account not only accuracy, diversity and stability but also scalability issues. In [60] some new evaluation measures of the kind are proposed and tested in several datasets, but future work could advance on the design of a more theoretical framework that aims to achieve similar results as for the stability issue.

- Threshold methods for rankers. In those cases in which feature selection rankers are employed, classically the ensemble approaches have retained a fixed percentage of the top ranked features [2, 14, 9]. However, this approach has the problem that the adequate percentage depends on the specific dataset used. Some authors have evaluated other types of thresholding that could take into account combinations between precision and recall (F-measure), Area under ROC curve, etc, most of which rely on the posterior classification of the datasets [158]. Nevertheless, this approach implies a significant computational burden, undesirable for large scale datasets, if not impossible, while in any case the threshold remains highly dependent on the classification algorithm used. Finally and more recently, some studies have tried to devise methods of thresholding that do not rely on the posterior classification stage, as in [159, 26, 27].

- Feature aggregation. As detailed in section 3 there are several methods that can be employed for combining the features obtained by the individual methods. However, there are some problems that are still under research, and that constitute interesting lines for the future, as for example the possibility of using more informed methods that aid to solve the ties that result from some combiners, or develop new methods that could assist in eliminating the redundancy that might be introduced when aggregating the partial feature subsets derived from the individual feature selectors.

- Explainability. During the last years the trade-off between accuracy and explainability in Machine Learning has been clearly imbalanced towards the accuracy side. In fact, this is one of the main reasons for the success of Deep learning, that has set forth one record after another on most benchmark datasets since 2006 [160]. In fact, in many competitions the only algorithm deep learning is up against is itself. However nowadays a new tendency stands up towards transparency and explainability, as new laws and regulations concerning Artificial Intelligence (AI) usage have put them into stage (notably, the new General data Protection Regulation– GDPR– that will go into effect in May, 2018 in all EU); but also social interest on AI in general as traceability, governance, compliance, etc need human-like justification. For that reason, ensemble models of the past have been revisited [161, 162] due to their explainability properties, and at the same time Special Sessions, like the "Interpretable Learning Classifiers" that will be chaired by P.P. Angelov and J.C. Principe in the 2018 IEEE World Congress on Computational Intelligence, in which Ensem-

bles of Deep Learning Classifiers was an specific subtopic, or the Special Session "Interpretable ML Symposium" at NIPS 2017, aim at addressing the bottleneck issue for achieving more interpretable results. In a society that envisions a future in which algorithms will deal with vast quantities of data and features in all kinds of disciplines, there is an urgent need for solutions to the indispensable issue of feature selection, some of which perhaps could be confronted using an ensemble approach.

Acknowledgments

This research has been financially supported in part by the Spanish Ministerio de Economa y Competitividad (research project TIN 2015-65069-C2-1-R), by the the Xunta de Galicia (research projects GRC2014/035 and the Centro Singular de Investigación de Galicia, accreditation 2016-2019) and by the European Union (FEDER/ERDF).

References

[1] G. Brown, Ensemble learning, in: Encyclopedia of Machine Learning, Springer, 2011, pp. 312–320.

[2] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, Feature selection for high-dimensional data, Springer, 2015.

[3] I. Guyon, A. Elisseeff, An Introduction to Variable and Feature Selection, Journal of Machine Learning Research 3 (2003) 1157–1182.

[4] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, Feature selection for high-dimensional data, Progress in Artificial Intelligence 5 (2) (2016) 65–75.

[5] P. Cunningham, J. Carney, Diversity versus quality in classification ensembles based on feature selection, in: Proc. European Conference on Machine Learning (ECML), R. Lpez de Mntaras and E. Plaza (Eds), LNAI 1810, 2000, pp. 109–116.

[6] D. Opitz, Feature selection for ensembles, in: Proc. 16th Nat. Conf. on Artificial Intelligence, AAAI Press, 1999, pp. 379–384.

[7] Y. Saeys, T. Abeel, Y. Van der Peer, Robust feature selection using ensemble feature selection techniques, in: Proc. European Conference on Machine Learning (ECML PKDD), In W. Daelemans et al. (Eds), LNAI 5212, 2008, pp. 313–325.

[8] A. Tsymbal, M. Pechenizkiy, P. Cunningham, Diversity in search strategies for ensemble feature selection, Information fusion 6 (1) (2005) 83–98.

[9] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo, A. Alonso-Betanzos, Ensemble feature selection: Homogeneous and heterogeneous approaches, Knowledge-Based Systems 114 (2017) 124–139.

[10] A. Das, A. Das, S.and Ghosh, Ensemble feature selection using bi-objective genetic algorithm, Knowledge-Based Systems 123 (2017) 116–127.

[11] E. Tuv, A. Borisov, K. Runger, G.and Torkkola, Feature selection with ensembles, artificial variables and redundancy elimination, Journal of Machine Learning Research 10 (2009) 1241–1366.

[12] J. Rogers, S. Gunn, Ensemble algorithms for feature selection, in: Deterministic and Statistical Methods in Machine Learning. Lecture Notes in Computer Science, In: Winkler J., Niranjan M., Lawrence N. (eds), Vol. 3635, 2005, pp. 180–198.

[13] N. Settouti, M. Chikh, V. Barra, A new feature selection approach based on ensemble methods in semi-supervised classification, Pattern Analysis and Applications 20 (3) (2017) 673–686. doi:10.1007/s10044-015-0524-9. URL https://doi.org/10.1007/s10044-015-0524-9

[14] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, A review of feature selection methods on synthetic data, Knowledge and Information systems 34 (2013) 483–519.

[15] S. Nogueira, G. Brown, Measuring the stability of feature selection with applications to ensemble methods, in: International Workshop on Multiple Classifier Systems, Springer, 2015, pp. 135–146.

[16] L. I. Kuncheva, A stability index for feature selection., in: Artificial intelligence and applications, 2007, pp. 421–427.

[17] F. Bonab, H.R.and Can, A theoretical framework on the ideal number of classifiers for online ensembles in data streams, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM, 2016, pp. 2053–2056.

[18] A. Kumar, C. Mingyue, Inherent predictability, requirements on the ensemble size, and complementarity, Monthly Weather Review 143 (2015) 3192–3202.

[19] H. Bonab, F. Can, Less is more: A comprehensive framework for the number of components of ensemble classifiers, IEEE Transactions on Neural Networks and Learning Systems 14 (2017) 1–7.

[20] Z. Zhou, D. Wei, G. Li, H. Dai, On the size of training set and the benefit from ensemble, in: In: Dai H., Srikant R., Zhang C. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2004. Lecture Notes in Computer Science, vol 3056., 2004.

[21] G. Martínez-Muñoz, A. Suárez, Out-of-bag estimation of the optimal sample size in bagging, Pattern Recognition 43 (2010) 143–152.

[22] C. Ferro, T. Jupp, F. Lambert, C. Hungtingford, P. Cox, Model complexity versus ensemble size: allocating resources for climate prediction, Philosophical Transactions of the Royal Society 370 (2012) 1087–1099.

[23] M. Ponti, I. Rossi, Ensembles of optimum-path forest classifiers using input data manipulation and undersampling, in: In Zhou ZH., Roli F., Kittler J. (eds) Multiple Classifier Systems. MCS 2013. Lecture Notes in Computer Science, vol 7872, 2013, pp. 236–246.

[24] V. Bolón-Canedo, K. Sechidis, N. Sánchez-Maroño, A. Alonso-Betanzos, G. Brown, Exploring the consequences of distributed feature selection in dna microarray data., in: In Proceedings 2017 International Joint Conference on Neural Networks (IJCNN), CFP17-US-DVD, 2017.

[25] J. Torres-Sospedra, M. Fernandez-Redondo, C. Hernandez-Espinosa, A research on combination methods for ensembles of multilayer feedforward, in: Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on, Vol. 2, IEEE, 2005, pp. 1125–1130.

[26] B. Seijo-Pardo, V. Bolón-Canedo, A. Alonso-Betanzos, Testing different ensemble configurations for feature selection, Neural Processing Letters 46 (2017) 857–880.

[27] B. Seijo-Pardo, V. Bolón-Canedo, A. Alonso-Betanzos, On developing an automatic threshold applied to feature selection ensembles, Information Fusion (2019) 1227–245.

[28] P. Granitto, P. Verdes, H. Ceccatto, Neural network ensembles: evaluation of aggregation algorithms, Artificial Intelligence 163 (2) (2005) 139 – 162.

[29] S. E. Lacy, M. A. Lones, S. L. Smith, A comparison of evolved linear and non-linear ensemble vote aggregators, in: 2015 IEEE Congress on Evolutionary Computation (CEC), 2015, pp. 758–763. doi:10.1109/CEC.2015.7256967.

[30] M. Haque, N. Noman, R. Berretta, P. Moscato, Heterogeneous ensemble combination search using genetic algorithm for class imbalanced data classification, PLoS One 11 (2016) e0146116.

[31] A. Canuto, M. Abreu, L. de Melo Oliveira, J. J. Xavier, A. Santos, Investigating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensembles, Pattern Recognition Letters 28 (2007) 472–486.

[32] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, An ensemble of filters and classifiers for microarray data classification, Pattern recognition 45 (2012) 531–539.

[33] H. Wang, T. M. Khoshgoftaar, A. Napolitano, A comparative study of ensemble feature selection techniques for software defect prediction, in: 2010 Ninth International Conference on Machine Learning and Applications, 2010, pp. 135–140. doi:10.1109/ICMLA.2010.27.

[34] B. Pes, N. Dess, M. Angioni, Exploiting the ensemble paradigm for stable feature selection: A case study on high-dimensional genomic data, Information Fusion 35 (2017) 132 – 147. doi:https://doi.org/10.1016/j.inffus.2016.10.001.

[35] V. Nikulin, On the homogeneous ensembling via balanced subsets combined with wilcoxon-based feature selection., in: In: Yao J. et al. (eds) Rough Sets and Current Trends in Computing. RSCTC 2012. Lecture Notes in Computer Science, vol 7413., Springer, Berlin, Heidelberg, 2012.

[36] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, Data classification using an ensemble of filters, Neurocomputing 135 (2014) 13–20.

[37] D. Guru, M. Suhil, S. Pavithra, G. Priya, Ensemble of feature selection methods for text classification: An analytical study., in: In: Abraham A., Muhuri P., Muda A., Gandhi N. (eds) Intelligent Systems Design and Applications. ISDA 2017. Advances in Intelligent Systems and Computing, vol 736., Springer, Cham, 2018.

[38] A. Ben Brahim, M. Limam, Ensemble feature selection for high dimensional data: a new method and a comparative study, Advances in Data Analysis and Classification 1–16doi:10.1007/s11634-017-0285-y.

[39] L. I. Kuncheva, Combining pattern classifiers: Methods and algorithms, John Wiley & Sons, 2014.

[40] D. Peteiro-Barral, B. Guijarro-Berdiñas, A survey of methods for distributed machine learning, Progress in Artificial Intelligence 2 (1) (2013) 1–11.

[41] D. Álvarez-Estévez, N. Sánchez-Maroño, A. Alonso-Betanzos, V. Moret-Bonillo, Reducing dimensionality in a database of sleep eeg arousals, Expert Systems with Applications 38 (6) (2011) 7746–7754.

[42] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, Distributed feature selection: An application to microarray data classification, Applied soft computing 30 (2015) 136–150.

[43] L. Morán-Fernández, V. Bolón-Canedo, A. Alonso-Betanzos, Centralized vs. distributed feature selection methods based on data complexity measures, Knowledge-Based Systems 117 (2017) 27–45.

[44] J. M. Stuart, E. Segal, D. Koller, S. K. Kim, A gene-coexpression network for global discovery of conserved genetic modules, science 302 (5643) (2003) 249–255.

[45] S. Aerts, D. Lambrechts, S. Maity, P. Van Loo, B. Coessens, F. De Smet, L.-C. Tranchevent, B. De Moor, P. Marynen, B. Hassan, et al., Gene prioritization through genomic data fusion, Nature biotechnology 24 (5) (2006) 537.

[46] R. Kolde, S. Laur, P. Adler, J. Vilo, Robust rank aggregation for gene list integration and meta-analysis, Bioinformatics 28 (4) (2012) 573–580.

[47] T. Joachims, Optimizing search engines using clickthrough data, in: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2002, pp. 133–142.

[48] L. I. Kuncheva, C. J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Machine learning 51 (2) (2003) 181–207.

[49] L. I. Kuncheva, M. Skurichina, R. P. Duin, An experimental study on diversity for bagging and boosting with linear classifiers, Information fusion 3 (4) (2002) 245–258.

[50] R. Lysiak, M. Kurzynski, T. Woloszynski, Optimal selection of ensemble classifiers using measures of competence and diversity of base classifiers, Neurocomputing 126 (2014) 29–35.

[51] G. D. Cavalcanti, L. S. Oliveira, T. J. Moura, G. V. Carvalho, Combining diversity measures for ensemble pruning, Pattern Recognition Letters 74 (2016) 38–45.

[52] C. Brodley, T. Lane, Creating and exploiting coverage and diversity, in: Proc. AAAI-96 Workshop on Integrating Multiple Learned Models, Portland, OR, 1996, pp. 8–14.

[53] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, Y. Saeys, Robust biomarker identification for cancer diagnosis with ensemble feature selection methods, Bioinformatics 26 (3) (2009) 392–398.

[54] G. Ditzler, R. Polikar, G. Rosen, A bootstrap based neyman-pearson test for identifying variable importance, IEEE transactions on neural networks and learning systems 26 (4) (2015) 880–886.

[55] Z. He, W. Yu, Stable feature selection for biomarker discovery, Computational biology and chemistry 34 (4) (2010) 215–225.

[56] A. Kalousis, J. Prados, M. Hilario, Stability of feature selection algorithms: a study on high-dimensional spaces, Knowledge and information systems 12 (1) (2007) 95–116.

[57] K. Dunne, P. Cunningham, F. Azuaje, Solutions to instability problems with sequential wrapper-based approaches to feature selection, Journal of Machine Learning Research (2002) 1–22.

[58] E. M. Voorhees, Evaluation by highly relevant documents, in: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2001, pp. 74–82.

[59] G. Jurman, S. Riccadonna, R. Visintainer, C. Furlanello, Canberra distance on ranked lists, in: Proceedings of Advances in Ranking NIPS 09 Workshop, Citeseer, 2009, pp. 22–27.

[60] V. Bolón-Canedo, D. Rego-Fernández, D. Peteiro-Barral, A. Alonso-Betanzos, B. Guijarro-Berdiñas, N. Sánchez-Maroño, On the scalability of feature selection methods on high-dimensional data, Knowledge and Information Systems (2017) 1–48.

[61] F. Yang, K. Mao, Robust feature selection for microarray data based on multicriterion fusion, IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 8 (2011) 1080–1092.

[62] T. Windeatt, R. Duangsoithong, R. Smith, Embedded feature ranking for ensemble mlp classifiers, IEEE Transactions on Neural Networks 22 (2011) 988–994.

[63] T. Windeatt, M. Prior, Stopping criteria for ensemble-based feature selection, in: Multiple Classifier Systems,In: Haindl M., Kittler J., Roli F. (eds) MCS 2007. Lecture Notes in Computer Science, vol 4472, Springer, Berlin, Heidelberg, 2007, pp. 271–281.

[64] J. Olsson, D. W. Oard, Combining feature selectors for text classification, in: Proceedings of the 15th ACM international conference on Information and knowledge management, 2006, pp. 798–799.

[65] H. Wang, T. M. Khoshgoftaar, K. Gao, Ensemble feature selection technique for software quality classification., in: International Conference on Software Engineering, SEKE 2010, 2010, pp. 215–220.

[66] H. Wang, T. M. Khoshgoftaar, A. Napolitano, A comparative study of ensemble feature selection techniques for software defect prediction, in: Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on, 2010, pp. 135–140.

[67] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, Y. Saeys, Robust biomarker identification for cancer diagnosis with ensemble feature selection methods, Bioinformatics 26 (3) (2010) 392–398.

[68] A. Ben Brahim, M. Limam, Robust ensemble feature selection for high dimensional data sets, in: High Performance Computing and Simulation (HPCS), 2013 International Conference on, 2013, pp. 151–157.

[69] B. Schowe, K. Morik, Fast-ensembles of minimum redundancy feature selection, in: In: Okun O., Valentini G., Re M. (eds) Ensembles in Machine Learning Applications. Studies in Computational Intelligence, vol 373., Springer, Berlin, Heidelberg, 2011, pp. 75–95.

[70] L. Mitchell, T. Sloan, M. Mewissen, P. Ghazal, T. Forster, M. Piotrowski, A. Trew, Parallel classification and feature selection in microarray data using sprint, Concurrency and Computation. Practice and Experience 26 (4) (2014) 854–865.

[71] C. Eiras-Franco, V. Bolón-Canedo, S. Ramos, J. Gónzález-Domínguez, A. Alonso-Betanzos, J. Touriño, Multithreaded and spark parallelization of feature selection filters, Journal of Computational Sciences 17 (2016) 609–619.

[72] S. Ramírez-Gallego, H. Mouriño-Talín, D. Martínez-Rego, V. Bolón-Canedo, J. Benítez, A. Alonso-Betanzos, F. Herrera, An information theory-based feature selection framework for big data under apache spark, IEEE Transactions on Systems, Man, and Cybernetics: Systems-doi:10.1109/TSMC.2017.2670926.

[73] S. Ramírez-Gallego, I. Lastra, D. Martínez-Rego, V. Bolón-Canedo, J. Benítez, F. Herrera, A. Alonso-Betanzos, Fast-mrmr: Fast minimum redundancy maximum relevance algorithm for high-dimensional big data, International Journal of Intelligent Systems 32 (2) (2017) 134–152.

[74] H. Elghazel, A. Aussem, Unsupervised feature selection with ensemble learning, Machine Learning 98 (1-2) (2015) 157–180.

[75] Y. Hong, S. Kwong, Y. Chang, Q. Ren, Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm, Pattern Recognition 41 (9) (2008) 2742–2756.

[76] Y. Hong, S. Kwong, Y. Chang, Q. Ren, Consensus unsupervised feature ranking from multiple views, Pattern Recognition Letters 29 (5) (2008) 595–602.

[77] M. Morita, L. S. Oliveira, R. Sabourin, Unsupervised feature selection for ensemble of classifiers, in: Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on, IEEE, 2004, pp. 81–86.

[78] H. Grabner, C. Leistner, H. Bischof, Semi-supervised on-line boosting for robust tracking, in: European conference on computer vision, Springer, 2008, pp. 234–247.

[79] F. Bellal, H. Elghazel, A. Aussem, A semi-supervised feature ranking method with ensemble learning, Pattern Recognition Letters 33 (10) (2012) 1426–1433.

[80] Y. Han, K. Park, Y.-K. Lee, Confident wrapper-type semi-supervised feature selection using an ensemble classifier, in: Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on, IEEE, 2011, pp. 4581–4586.

[81] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann, 2016.

[82] U. Neumann, N. Genze, D. Heider, Efs: an ensemble feature selection tool implemented as r-package and web-application, BioData mining 10 (1) (2017) 21.

[83] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework., Journal of Multiple-Valued Logic & Soft Computing 17.

[84] M. Hofmann, R. Klinkenberg, RapidMiner: Data mining use cases and business analytics applications, CRC Press, 2013.

[85] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, Journal of machine learning research 12 (Oct) (2011) 2825–2830.

[86] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Communications of the ACM 51 (1) (2008) 107–113.

[87] J. Tukey, Exploratory data analysis, Addison-Wesley, 1977.

[88] B. Dasarathy, B. Sheela, A composite classifier system: Concepts and methodology, in: Proceedings of the IEEE, Vol. 67, 1979, pp. 708–713.

[89] V. Bolón-Canedo, A. Alonso-Betanzos, Recent Advances in Ensembles for Feature Selection, Springer International Publishing, 2018.

[90] Z. Zhou, Ensemble Methods: Foundations and Algorithms, Chapman and Hall, 2012.

[91] R. Schapire, Y. Freund, Boosting: Foundations and Algorithms, MIT Press, 2012.

[92] L. Rokach, Pattern Classification using ensemble methods, World Scientific, 2010.

[93] G. Seni, J. Elder, Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions, Morgan and Claypool Publishers, 2010.

[94] V. Mallet, I. Herlin, Quantification of uncertainties from ensembles of simulations, in: In International Meeting Foreknowledge Assessment Series, 2016.
URL `http://www.foreknowledge2016.com/`

[95] P. Pérez-Gallego, Quevedo-Pérez, J. J.R., Coz-Velasco, Using ensembles for problems with characterizable changes in data distribution: A case study on quantification, Information Fusion 34 (2017) 87–100. doi:10.1016/j.inffus.2016.07.001.

[96] D. Fernández-Francos, O. Fontenla-Romero, A. Alonso-Betanzos, One-class convex hull-based algorithm for classification in distributed environments, IEEE Transactions on Systems, Man and Cybernetics: Systems-doi:10.1109/TSMC.2017.2771341.

[97] C. Silva, T. Bouwmans, C. Frelicot, Superpixel-based online wagging one-class ensemble for feature selection in foreground/background separation, Pattern Recognition Letters 100 (2017) 144–151.

[98] B. Krawczyk, B. Cyganek, Selecting locally specialised classifiers for one-class classification ensembles, Pattern Analysis and Applications 20 (2) (2017) 427–439.

[99] J. Gama, Knowledge discovery from data streams, Chapman & Hall/CRC, 2010.

[100] X. Hu, P. Zhou, P. Li, J. Wang, X. Wu, A survey on online feature selection with streaming features, Frontiers of Computer Sciencedoi:10.1007/s11704-016-5489-3.

[101] F. Duan, L. Dai, Recognizing the gradual changes in semg characteristics based on incremental learning of wavelet neural network ensemble, IEEE Transactions on Industrial Electronics 64 (5) (2017) 4276–4286.

[102] I. Khan, J. Z. Huang, K. Ivanov, Incremental density-based ensemble clustering over evolving data streams, Neurocomputing 191 (2016) 34–43.

[103] Z. Yu, P. Luo, J. You, H. Wong, H.S.and Leung, S. Wu, J. Zhang, G. Han, Incremental semi-supervised clustering ensemble for high dimensional data clustering, IEEE Transactions on Knowledge and Data Engineering 28 (3) (2016) 701–714.

[104] W. Lu, Z. Li, J. Chu, Adaptive ensemble undersampling-boost: A novel learning framework for imbalanced data, Journal of Systems and Software 132 (2017) 272–282.

[105] W. Lin, C. Tsai, Y. Hu, J. Jhang, Clustering-based undersampling in class-imbalanced data, Information Sciences 409 (2017) 17–26.

[106] H. Gao, S. Jian, Y. Peng, X. Liu, A subspace ensemble framework for classification with high dimensional missing data, Multidimensional systems and Signal processing 28 (4) (2017) 1309–1324.

[107] S. Huang, B. Wang, J. Qiu, J. Yao, G. Wang, G. Yu, Parallel ensemble of online sequential extreme learning machine based on mapreduce, Neurocomputing 174 (2016) 352–367.

[108] R. Bekkerman, M. Bilenko, J. Langford, Scaling up machine Learning: parallel and distributed approaches, Cambridge University Press, 2012.

[109] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. Franklin, R. Zadeh, M. Zaharia, A. Talwalkar, Mllib: Machine learning in apache spark, Journal of MAchine Learning Research 17 (2016) 1–7.

[110] V. Bolón-Canedo, N. Sanchez-Maroño, A. Alonso-Betanzos, J. M. Benítez, F. Herrera, A review of microarray datasets and applied feature selection methods, Information Sciences 282 (2014) 111–135.

[111] B. Seijo-Pardo, V. Bolón-Canedo, A. Alonso-Betanzos, Using a feature selection ensemble on dna microarray datasets, in: Proceedings European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN, 2016, pp. 277–282.

[112] L. Liu, H.and Liu, H. Zhang, Ensemble gene selection by grouping for microarray data classification, Journal of Biomedical Informatics 43 (1) (2010) 81–87.

[113] F. Sharbaf, S. Mosafer, M. Moattar, A hybrid gene selection approach for microarray data classification using cellular learning automata and ant colony optimization, Genomics 107 (6) (2016) 231–238.

[114] B. Remeseiro, V. Bolón-Canedo, D. Peteiro-Barral, A. Alonso-Betanzos, B. Guijarro-Berdiñas, A. Mosquera, M. Penedo, N. Sánchez-Maroño, A methodology for improving tear film lipid layer classification, IEEE Journal of Biomedical and Health Informatics 18 (4) (2014) 1485–1493.

[115] P. Chowriappa, S. Dua, U. Acharya, M. Krishnan, Ensemble selection for feature-based classification of diabetic maculopathy images, Computers in Biology and Medicine 43 (12) (2013) 2156–2162.

[116] S. Huda, J. Yearwood, H. F. Jelinek, M. M. Hassan, G. Fortino, M. Buckland, A hybrid feature selection with ensemble classification for imbalanced healthcare data: A case study for brain tumor diagnosis, IEEE Access 4 (2016) 9145–9154.

[117] T. Sivapriya, A. Kamal, P. Thangaiah, Ensemble merit merge feature selection for enhanced multinomial classification in alzheimers dementia, Computational and Mathematical Methods in Medicine (2015) 676129doi:10.1155/2015/676129.

[118] E. Varol, B. Gaonkar, G. Erus, R. Schultz, C. Davatzikos, Feature ranking based nested support vector machine ensemble for medical image classification., in: Proceedings IEEE International Symposium on Biomedical Imaging: from nano to macro IEEE International Symposium on Biomedical Imaging, 2012, pp. 146–149. doi:10.1109/ISBI.2012.6235505.

[119] J. Goh, V. Thing, A hybrid evolutionary algorithm for feature and ensemble selection in image tampering detection, International Journal of Electronic Security and Digital Forensics 7 (1) (2015) 76–104.

[120] H. Reeve, G. Brown, Modular autoencoders for ensemble feature extraction, Journal of Machine Learning Research, NIPS 2015. 44 (2015) 242–259.

[121] S. Tang, T. Pan, Feature extraction via recurrent random deep ensembles and its application in group-level happiness estimation, arXiv:1707.09871v1 [cs.CV] 24 Jul 2017.

[122] A. Polyakova, L. L Lipinskiy, A study of fuzzy logic ensemble system performance on face recognition problem, in: IOP Conference Series: Materials Science and Engineering, Vol. 173, 2017, p. 012013.

[123] J. Yang, D. Zhang, X. Yong, J. Yang, Two-dimensional discriminant transform for face recognition, Pattern recognition 38 (7) (2005) 125–129.

[124] R. Mallipeddi, M. Lee, Ensemble based face recognition using discriminant pca features, in: Proceedings IEEE Congress on Evolutionary Computation, 2012, pp. 1–7.

[125] Y. Su, S. Shan, X. Chen, W. Gao, Hierarchical ensemble of global and local classifiers for face recognition, IEEE Transactions on Image Processing 18 (8) (2009) 1885–1896.

[126] A. Lumini, L. Nanni, S. Brahnam, Ensemble of texture descriptors and classifiers for face recognition, Applied Computing and Informatics 13 (1) (2017) 79–91.

[127] S. Alelyani, H. Liu, Ensemble feature selection in face recognition: Icmla 2012 challenge, in: Proceedings 11th International Conference on Machine Learning and Applications, 2012, pp. 588–591. doi:10.1109/ICMLA.2012.182.

[128] C. Shang, M. Li, S. Feng, Q. Jiang, J. Fan, Feature selection via maximizing global information gain for text classification, Knowledge-Based Systems 54 (2013) 298–309.

[129] S. Baccianella, A. Esuli, F. Sebastiani, Feature selection for ordinal text classification, Neural computation 26 (3) (2014) 557–591.

[130] B. Shravankumar, V. Ravi, Text classification using ensemble features selection and data mining techniques, in: Proc. Swarm, Evolutionary, and Memetic Computing. SEMCCO 2014. Lecture Notes in Computer Science, Vol. 8947, 2015.

[131] S. Van Landeghem, T. Abeel, Y. Saeys, Y. Van de Peer, Discriminative and informative features for biomolecular text mining with ensemble feature selection, Bioinformatics 26 (18) (2010) i554–60.

[132] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, Feature selection and classification in multiple class datasets: An application to kdd cup 99 dataset, Expert systems with Applications 38 (5) (2011) 5947–5957.

[133] A. Alazab, M. Hobbs, A. J., M. Alazab, Using feature selection for intrusion detection system, in: Proc. International Symposium on Communications and Information Technologies (ISCIT), 2012, pp. 296–301.

[134] V. Balasaraswathi, M. Sugumaran, Y. Hamid, Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms, Journal of Communications and Information Networks 2 (4) (2017) 107–119.

[135] M. Hasan, M. Nasser, S. Ahmad, K. Molla, Feature selection for intrusion detection using random forest, Journal of Information Security 7 (2016) 129–140.

[136] R. Zuech, K. T.M., A survey on feature selection for intrusion detection, in: Proc. 21st ISSAT International Conference on Reliability and Quality in Design, 2015, pp. 150–155.

[137] B. Li, P. Zhang, H. Tian, S. Mi, D. Liu, G. Ren, A new feature extraction and selection scheme for hybrid fault diagnosis of gearbox, Expert Systems with Applications 38 (8) (2011) 10000–10009.

[138] M. Islam, M. M. M. Islam, J. Kim, Feature selection techniques for increasing reliability of fault diagnosis of bearings, in: Proc. 9th International Conference on Electrical and Computer Engineering (ICECE), 2016, pp. 396–399.

[139] M. Luo, C. Li, X. Zhang, R. Li, X. An, Compound feature selection and parameter optimization of elm for fault diagnosis of rolling element bearings, ISA Transactions 65 (2016) 556–566.

[140] K. Hui, C. Ooi, M. Lim, M. Leong, S. Al-Obaidi, An improved wrapper-based feature selection method for machinery fault diagnosis, Plos One 12 (12) (2017) e0189143.

[141] B. Chebel-Morello, S. Malinowski, H. Senoussi, Feature selection for fault detection systems: application to the tennessee eastman process, Applied Intelligence 44 (1) (2016) 111–122.

[142] C. Rajeswari, B. Sathiyabhama, S. Devendiran, K. Manivannan, Bearing fault diagnosis using multiclass support vector machine with efficient feature selection methods, International Journal of Mechanical & Mechatronics Engineering 15 (1) (2016) 1–12.

[143] H. Li, J. Zhao, X. Zhang, X. Ni, Fault diagnosis for machinery based on feature selection and probabilistic neural network, International Journal of Performability Engineering 13 (7) (2017) 1165–1170.

[144] V. Bolón-Canedo, J. Díez, O. Luaces, A. Bahamonde, A. Alonso-Betanzos, Paving the way for providing teaching feedback in automatic evaluation of open response assignments, in: Proceedings International Joint Conference on Neural Networks (IJCNN),CFP17-US-DVD, 2017.

[145] L. Pietruczuk, L. Rutkowski, M. Jaworski, P. Duda, How to adjust an ensemble size in stream data mining?, Information Sciences 381 (2017) 46–54.

[146] Hernández-Lobato, G. G. Martínez-Muñoz, A. Suárez, How large should ensembles of classifiers be?, Pattern Recognition 46 (5) (2013) 1323–1336.

[147] S. Nogueira, K. Sechidis, G. Brown, On the stability of feature selection algorithms., Journal of Machine Learning Research 18 (2018) 1–54.

[148] L. Yu, C. Ding, S. Loscalzo, Stable feature selection via dense feature groups, in: KDD '08 Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp. 803–811.

[149] M. Zhang, L. Zhang, J. Zou, C. Yao, H. Xiao, Q. Liu, J. Wang, D. Wang, C. Wang, Z. Guo, Evaluating reproducibility of differential expression discoveries in microarray studies by considering correlated molecular changes., Bioinformatics 25 (13) (2009) 1662–8.

[150] P. Somol, J. Novovicova, Evaluating stability and comparing output of feature selectors that optimize feature subset cardinality, IEEE Trans. Pattern Anal. Mach. Intell. 32 (11) (2010) 1921–1939.

[151] R. Guzmán-Martínez, R. Alaiz-Rodríguez, Feature selection stability assessment based on the jensen-shannon divergence, in: In: Gunopulos D., Hofmann T., Malerba D. and Vazirgiannis M. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2011. Lecture Notes in Computer Science, vol 6911, 2011.

[152] T. M. Khoshgoftaar, A. Fazelpour, H. Wang, R. Wald, A survey of stability analysis of feature subset selection techniques, in: 2013 IEEE 14th International Conference on Information Reuse Integration (IRI), 2013, pp. 424–431.

[153] W. W. B. Goh, L. Wong, Evaluating feature-selection stability in next-generation proteomics, Journal of Bioinformatics and Computational Biology 14 (05) (2016) 1650029. doi:10.1142/S0219720016500293.

[154] L. Yu, H. Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, in: Proceedings of the Twentieth International Conference on Machine Learning (ICML), Vol. 2, 2003, pp. 856–863.

[155] M. Moshki, P. Kabiri, A. Mohebalhojeh, Scalable feature selection in high-dimensional data based on grasp, Appl. Artif. Intell. 29 (3) (2015) 283–296. doi:10.1080/08839514.2015.1004616.

[156] K. Yu, X. Wu, W. Ding, J. Pei, Scalable and accurate online feature selection for big data, ACM Trans. Knowl. Discov. Data 11 (2) (2016) 16:1–16:39. doi:10.1145/2976744.

[157] S. Zadeh, M. Ghadiri, V. S. Mirrokni, M. Zadimoghaddam, Scalable feature selection via distributed diversity maximization, 2017, pp. 2876–2883. URL http://www.cs.toronto.edu/~sepehr/papers/AAAI17_DDisMI.pdf

[158] A. Shanab, T. Khoshgoftaar, R. Wald, Robustness of threshold-based feature rankers with data sampling on noisy and imbalanced data, in: Proceedings of the 25th International Florida Artificial Intelligence Research Society Conference, 2012.

[159] C. Sarkar, S. Cooley, J. Srivastava, Robust feature selection technique using rank aggregation, Applied Artificial Intelligence 28 (3) (2014) 243257.

[160] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press, 2016.

[161] J. Friedman, B. Popescu, Predictive learning via rule ensembles, The Annals of Applied Statistics 2 (3) (2008) 916–954.

[162] D. Petkovic, R. Atman, M. Wong, A. Vigil, Improving the explainability of random forest classifier user centered approach, in: Pacific Symposium on Biocomputing, Vol. 23, 2018, pp. 204–215.