Information Fusion 45 (2019) 227–245

Contents lists available at ScienceDirect

Information Fusion

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

# On developing an automatic threshold applied to feature selection ensembles

B. Seijo-Pardo, V. Bolón-Canedo, A. Alonso-Betanzos*

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

ARTICLE INFO

*Keywords:*
Ensemble learning
Feature selection
Automatic thresholding

ABSTRACT

Feature selection ensemble methods are a recent approach aiming at adding diversity in sets of selected features, improving performance and obtaining more robust and stable results. However, using an ensemble introduces the need for an aggregation step to combine all the output methods that confirm the ensemble. Besides, when trying to improve computational efficiency, ranking methods that order all initial features are preferred, and so an additional thresholding step is also mandatory. In this work two different ensemble designs based on ranking methods are described. The main difference between them is the order in which the combination and thresholding steps are performed. In addition, a new automatic threshold based on the combination of three data complexity measures is proposed and compared with traditional thresholding approaches based on retaining a fixed percentage of features. The behavior of these methods was tested, according to the *SVM* classification accuracy, with satisfactory results, for three different scenarios: synthetic datasets and two types of real datasets (where sample size is much higher than feature size, and where feature size is much higher than sample size).

## 1. Introduction

In recent years the size of the datasets used for machine learning has increased considerably, with the result that feature selection (FS) has become an essential preprocessing step for many data mining applications. Since FS reduces storage needs and removes irrelevant and redundant information, it improves the computational time needed for the machine learning algorithms. Several studies have demonstrated that FS can greatly improve the performance of subsequent classification [1–3]. Many approaches and algorithms [1,4,5] have been employed for this task, in the quest for more robust, compact and high-quality feature subsets.

To evaluate the features of a dataset, two different general approaches may be used: (i) individual evaluation and (ii) subset evaluation [6]. Individual evaluation methods, also known as rankers, assign a level of relevance to each feature and return an ordered ranking of all the features. Although this approach is not capable of eliminating redundant features, it notably improves computational performance over the subset evaluation approach. Subset evaluation generates successive subsets of features that are iteratively evaluated, using an optimality criterion, until the final subset of selected features is obtained. Although this approach has the advantage of detecting feature redundancy, it is computationally less efficient.

Although machine learning methods traditionally have used a single learning model to solve a particular problem, recently it has been shown that combining multiple different models can improve results. This approach, called ensemble learning, is based on the supposition that combining the output of multiple experts is better than using the output of any single expert [7–9]. Analogously, while FS is more frequently based on using a single algorithm, lately a few works have adopted the idea of ensemble learning for this task [10–12]. An ensemble for FS works by combining the outputs of several FS methods, aggregating partial results to obtain more robust and stable features for subsequent learning tasks. Two general strategies can be used to introduce the key concept of diversity in the ensembles. In the heterogeneous approach several different FS algorithms are used, whereas the traditional homogeneous approach uses different partitions of the training dataset fed to the same algorithm and producing different results that are also combined. This second strategy is the one exploited by the well-known bagging and boosting algorithms [13,14]. Diversity and robustness are thus achieved through the use of multiple feature evaluation criteria [15]. Although both approaches—in which diversity is the key concept—are of interest, the heterogeneous strategy is of most interest when the user does not have the technical knowledge necessary to select the most suitable algorithm for their problem. Ensembles of filters have previously been used for different scenarios and also for different classifiers, with outputs combined by means of common simple voting [16,17]. Ensembles of feature rankers have also

* Corresponding author.
*E-mail address:* ciamparo@udc.es (A. Alonso-Betanzos).

https://doi.org/10.1016/j.inffus.2018.02.007
Received 12 June 2017; Received in revised form 14 February 2018; Accepted 18 February 2018
Available online 19 February 2018
1566-2535/ 

been used for different applications [18,19], with the single ranked features combined in a global ranking using different approaches. Other works propose a feature ranking scheme for an ensemble of multilayer perceptrons (MLPs) [20], applied with a stopping criterion based on the Out-of-Bootstrap (OOB) estimate [21].

In this study, the ensemble learning idea was applied to the FS process and different ensemble configurations and designs were executed and compared. An heterogeneous ensemble approach was implemented, aimed at reducing the variability induced by using individual FS methods and taking advantage of the strengths and overcoming the weaknesses of the individual methods. In addition, ranker methods were used to configure the FS ensemble, since rankers can reduce the size of data without compromising the time and memory requirements of machine learning algorithms.

Since we were working with rankings, at some point we needed to establish a threshold to retain only the relevant features and to combine the rankings obtained by the different methods configuring the ensemble. In this respect, the main novelty of our proposal herein is the use of two different models, depending on whether thresholding was performed before or after combination (*Design TC* and *Design CT*). The performance of each model is analyzed and compared to the other according to the *SVM* classification accuracy. Since establishing an adequate threshold is not trivial, we also propose a methodology for establishing automatic thresholds based on measurements of data complexity [22] for feature rankings, both in *Design TC* and *Design CT*.

To sum up, the main contributions of our proposal are: (i) to free the user from having to select a specific FS method that works well with their dataset, given that most methods produce variable results depending on application characteristics; and (ii) to free the user from having to select a specific threshold and having to experiment with different percentages of retained features. The outcome is completely automatic FS methods that are independent of the nature of the dataset in that they obtain a generic threshold that runs smoothly in different scenarios and extracts the best subset of features from each dataset without having to pre-set threshold in feature percentages.

We experimented with a large and assorted suite of datasets, including artificial datasets, classical real datasets and microarray datasets. Based on our results, we state conclusions and propose guidelines of possible interest for future applications of ensembles for FS purposes.

The remainder of this paper is organized as follows. Section 2 describes the rationale under the design of the two ensemble approaches proposed; Section 3 is an introduction to the proposed method and its different components: ranker methods, combination (also called aggregation) methods, threshold values and classifier method used; Section 4 describes the proposed scenarios, experimental design and experimental results; and finally, Section 5 summarizes our conclusions and recommendations and proposes new lines of future work.

## 2. Information fusion design

In this study an ensemble of FS methods was used with the aim of obtaining more consistent, efficient and robust solutions than those yielded by individual methods. Using an ensemble means that the performance variance of obtaining a single result is reduced; in addition, the combination of multiple subsets might help to remove less relevant features [10–12]. The approach also has the advantage of not requiring the user to understand the technical details of individual algorithms and their suitability for certain datasets. We tested different ensemble methods and different numbers of ranking techniques to configure an ensemble (described in [11,23]), formed of six different FS methods—the combination that produced the best results.

There are several ways to design an ensemble [24] and the first decision is to select the FS methods. In our proposal, rankers were used since computational efficiency was our priority. The different FS rankers were individually applied to a particular dataset and the single final subset was obtained by combining the obtained outputs, for which reason a combination method was chosen. The use of rankers made it mandatory to apply a threshold to limit the number of selected features and so ensure efficiency in the subsequent learning methods. Different designs were obtained depending on the order of the combination and thresholding operations. Finally, of other possibilities for the ensemble [11,24], we opted for an ensemble of $n$ different ranker methods applied to the same training data, with two different designs: (i) rankings combined before thresholding; and (ii) a threshold cutoff applied before combining rankings.

### 2.1. Design CT: combination followed by thresholding

The generic design of an ensemble of feature rankers is based on obtaining the result of each ranker method—an ordered ranking—using an aggregator to fuse the rankings into a single final ranking and subsequently applying a threshold cutoff to obtain a final practical subset of features [7]. The pseudo-code for this approach is given in Algorithm 1.

**Data**: $N$ — number of ranker methods
**Data**: $T$ — number of features to be selected

**Result**: $P$ — classification prediction

**for** *each n from 1 to N* **do**
  Obtain ranking $R_n$ using ranker method $r_n$
**end**
$R$ = Obtain the final ranking by joining all $R_n$ rankings using the *Min* combination method.
$T$ = Select a threshold value cutoff $t$ from those available and apply.
$S$ = Select the $T$ top attributes from $R$.
Build the classifier with the selected attributes $S$.
Obtain prediction $P$.

**Algorithm 1.** Pseudo-code for Design CT: combination followed by thresholding.

### 2.2. Design TC: thresholding followed by combination

We redesigned the generic ensemble (i.e. *Design TC*) by reversing the order of the combination and thresholding steps. Therefore, the result of each ranker method was obtained as a first step, as in the generic design. A threshold cutoff was selected and applied to each single output to obtain individual partial subsets of features. Finally, these subsets were joined to achieve a single final subset of features. The pseudo-code for this approach is given in Algorithm 2.