ARTICLE IN PRESS
JID: NEUCOM [m5G;July 10, 2018;5:7]

Neurocomputing 000 (2018) 1–13

Contents lists available at ScienceDirect

Neurocomputing

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

Combining multiple algorithms in classifier ensembles using generalized mixture functions

Valdigleis S. Costa [a], Antonio Diego S. Farias [a,b], Benjamin Bedregal [a], Regivan H.N. Santiago [a], Anne Magaly de P. Canuto [a,*]

[a] Department of Informatics and Applied Mathematics, Federal University of Rio Grande do Norte (UFRN), Natal - RN, 59.072–970, Brazil
[b] Department of Exacts and Naturals Sciences, Federal Rural University of Semi-Arid (UFERSA), Pau dos Ferros - RN, 59.900-000, Brazil

ARTICLE INFO

Article history:
Received 31 December 2017
Revised 3 April 2018
Accepted 15 June 2018
Available online xxx

Communicated by Haijun Zhang

Keywords:
Classifier ensembles
Aggregation functions
Pre-aggregation functions
Generalized mixture functions

ABSTRACT

Classifier ensembles are pattern recognition structures composed of a set of classification algorithms (members), organized in a parallel way, and a combination method with the aim of increasing the classification accuracy of a classification system. In this study, we investigate the application of a generalized mixture (GM) functions as a new approach for providing an efficient combination procedure for these systems through the use of dynamic weights in the combination process. Therefore, we present three GM functions to be applied as a combination method. The main advantage of these functions is that they can define dynamic weights at the member outputs, making the combination process more efficient. In order to evaluate the feasibility of the proposed approach, an empirical analysis is conducted, applying classifier ensembles to 25 different classification data sets. In this analysis, we compare the use of the proposed approaches to ensembles using traditional combination methods as well as the state-of-the-art ensemble methods. Our findings indicated gains in terms of performance when comparing the proposed approaches to the traditional ones as well as comparable results with the state-of-the-art methods.

1. Introduction

In machine learning, a classifier ensemble, also known as ensemble systems, ensemble of classifiers or simply ensembles, can be understood as a collaborative decision-making system composed of N members (individual classifiers), in which a strategy is applied to combine the predictions of ensemble members to generate a single prediction as output [17]. In other words, a classifier ensemble is a two-layer pattern recognition structure in which the first layer is composed of a set of N individual classifiers and the second layer is composed of a combination module [36]. Essentially, the combination module is responsible for combining the outputs of the individual classifiers and for transforming them into a single output, which is the final output of an ensemble. The use of classifier ensembles in machine learning is not recent and, as stated in [32], the first reference that uses classifier ensembles dates back to 1963, in [3]. Since then, classifier ensembles have been used in different classification problems, for example, recognition of faces [22], revocable biometrics [9], among other applications. In addition to classification, there are several other application domains that the combination of multiple input information has been efficiently applied in order to generate a single output, for example: data clustering [18], support decision-making [54,55,67] and images processing [6,25,26]. In this paper, we investigate the use of classifier ensembles in the pattern classification context.

When working with classifier ensembles, one important issue to be taken into consideration is related to the selection of an efficient combination method. Ideally, this method should be able to exploit the individual strengths of all individual classifiers and, at the same time, to minimize their drawbacks [42]. For many years, simple methods as majority vote [2], linear combination and fusion methods [12,35,36,62] were the most popular methods since they were simple and provided reasonable performance. However, with the increase of data complexity, classifier ensembles started to require flexible approaches that can adjust their combination methods to the properties of analyzed datasets. Therefore, the use of trained combiners has gained a significant attention of the machine learning community. Nevertheless, these combination

* Corresponding author.
E-mail addresses: valdigleis@ppgsc.ufrn.br (V.S. Costa), antonio.diego@ufersa.edu.br (A.D.S. Farias), bedregal@dimap.ufrn.br (B. Bedregal), regivan@dimap.ufrn.br (R.H.N. Santiago), anne@dimap.ufrn.br (A.M.d.P. Canuto).

https://doi.org/10.1016/j.neucom.2018.06.021
0925-2312/© 2018 Elsevier B.V. All rights reserved.

methods require additional training time and access to separate subset of examples and, once again, this requirement can become prohibitive in certain application areas.

One way to improve the efficiency of combination methods is through the use of weights that can be used to denote the confidence (influence) of the individual classifiers in classifying an input pattern to a particular class [42]. Different ways of calculating weights (confidence) of each class for each individual classifier can be used in determining the relative contribution of each classifier within a classifier ensemble and they can be classified as static [34,40,41,45,46,52,64] or dynamic weighting [4,40,46,52]. For offering more flexibility and efficiency, in this paper, we will be working with dynamic weight selection (dynamic weighting).

One possible solution to the selection of an efficient combination method is the use of aggregation functions. They are mathematical models that are capable of solving the task of aggregating several sources of information and generating a single output. Among the most common aggregation functions found in the literature, we can cite t-norms, t-corms [37], aggregation functions [24,49], among others. In a recent study, [25–29], Farias et al. have investigated a class of functions called generalized mixture (GM) functions. The GM functions are capable of generalize the notions of ordered weighted averaging (OWA) [66] and mixture function [5]. The main advantage of GM functions is that the weights of its inputs can be dynamically defined for each input. In other words, while the weights of the other weighted functions are assigned statically, without taking into account the testing patterns, the generalized mixture functions can assign weights as a function of the testing patterns [28]. In practical applications, the GM functions have presented good results in the tasks of noise treatment and image reduction [25,26,28]. This advantage can be very useful for classifier ensembles, leading to an efficient decision making process.

Therefore, in this paper, we propose a new combination approach for classifier ensembles using the generalized mixture (GM) functions proposed in [25]. In other words, we adapt the GM functions to be used as the combination method of an ensemble system. In this sense, we explore the main advantage of the GM functions, dynamic weighting, in the combination process of an ensembles. Then, the weights related to the decision of each individual classifier is defined dynamically, according to the classifier outputs and the relation among the outputs of all classifiers. In order to evaluate the feasibility of the proposed approach, we perform an empirical analysis of ensemble performance in 25 different classification data sets, comparing its performance with classifier ensembles using traditional methods (those presented in [35]) as well as the state-of-the-art ensemble methods. In addition, a statistical analysis is also performed to analyze the performance of classifier ensembles, from the statistical point of view.

This paper is divided into eight sections and it is organized as follows. In Section 2, we describe some recent studies in classifier ensembles. The fundamental notions of the generalized mixture functions are introduced in Section 3, while the basic concepts of classifier ensembles are presented in Section 4. The proposed approach is presented in Section 5. In Section 6, the experimental methodology is presented, while Section 7 presents an analysis of the obtained results of this work. Finally, Section 7 concludes this paper.

2. Recent studies in weighted combination methods for in classifier ensembles

As already mentioned, different ways of calculating the weights of each class for each individual classifier can be used in a classifier ensemble [10,34,40,41,45,46,52,64]. Although weighted combination methods appear to provide some flexibility, obtaining the optimal weights is not an easy task. Therefore, some optimization techniques have been applied to define the best set of weights, such as in [4,40,46,52]. In [52], for instance, a genetic algorithm (GA) was used to define an optimized set of weights that are used along with the output of the individual classifiers to define the final output of the classifier ensembles. However, all the aforementioned studies apply procedures to define static weights. In other words, these methods define a set of weights that are used throughout the testing phase. This static way to define weights can eventually become inefficient for a classifier ensemble, since the accuracy of an individual classifier can change in the testing search space and this change is not capture by static weights.

In a dynamic weighting process, the outputs of all individual classifiers are aggregated and the most competent ones receive the highest weight values. The competence of the classifier outputs are usually based on some local competence measure. There are some studies that apply dynamic weights in combination methods, such as in [10,33,39,47,48,53,60]. However, in most of these studies, the dynamic weighting process relies on a model that has to be built as a neural network [33], a histogram representation [47], quadratic programming [11], fuzzy classifier [58], among others. These models usually requires an extra processing to be built and become complex structures to be designed. In addition, in [39,65], adaptive mechanisms are applied. In [39], for instance, a dynamic weighted majority (DWM) method is proposed, in which it uses a weighted-majority vote of the classifier and dynamically creates and deletes classifiers in response to changes in performance. However, the dynamic weights are defined based on the performance of the classifiers based on previously seen instances and not based on the information of the current instance to be classified.

In addition, some studies proposed a dynamic weighting procedure for a specific domain, such as concept drift [39,47,56], textual and visual content-based anti-phishing [68], among others. In the dynamic weighting technique proposed in [47], for instance, each classifier is dynamically weighted based on the similarity between an input pattern and the histogram representation of each concept present in the ensemble. In the mentioned paper, the Hellinger distance between an input and the histogram representation of every previously-learned concept is computed, and the score of every classifier is weighted dynamically according to the resemblance to the underlying concept distribution. According to the authors, the empirical analysis with synthetic problems indicate that the proposed fusion technique is able to increase system performance when input data streams incorporate abrupt concept changes, yet maintains a level of performance that is comparable to the average fusion rule when the changes are more gradual.

There are also some studies that are limited to a data-dependent measures. For instance, in [48], the authors used ad-hoc data-dependent measures in the dynamic weight setting procedure and noisy data could compromise the overall performance of the ensemble system.

Unlike the aforementioned studies, in this paper, we present a family of aggregation functions (GM) that is adapted to be used in a dynamic weighting procedure for classifier ensembles. In other words, these functions define weights for the output of the individual classifiers, in a dynamic way, without having to build a model and using information of the current instance to be classified. These functions are inexpensive and straightforward in system design and setup, leading to accurate and robust classifier ensembles.

3. Mathematical framework

In this section, the mathematical background used in this paper is described, starting with the description of aggregation functions,