Knowledge-Based Systems 106 (2016) 85–95

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A multi-level approach using genetic algorithms in an ensemble of Least Squares Support Vector Machines

Carlos Alberto de Araújo Padilha[a,*], Dante Augusto Couto Barone[a], Adrião Duarte Dória Neto[b]

[a] Federal University of Rio Grande do Sul, Institute of Informatics, Porto Alegre, RS, Brazil
[b] Federal University of Rio Grande do Norte, Department of Computer Engineering and Automation, Natal, RN, Brazil

ARTICLE INFO

Article history:
Received 12 October 2015
Revised 28 March 2016
Accepted 16 May 2016
Available online 17 May 2016

Keywords:
Ensemble systems
Diversity
Genetic algorithms
Least Squares Support Vector Machines
Feature selection

ABSTRACT

Despite the ensemble systems have been shown to be an efficient method to increase the accuracy and stability of learning algorithms in recent decades, its construction has a question to be elucidated: diversity. The disagreement among the models that compose the ensemble can be generated when they are built under different circumstances, such as training dataset, parameter setting and selection of learning algorithms. The ensemble may be viewed as a structure with three levels: input space, the base components and the combining block of the components responses. We propose a multi-level approach using genetic algorithms to build the ensemble of Least Squares Support Vector Machines (LS-SVM), performing a feature selection in the input space, the parameterization and the choice of which models will compose the ensemble at the component level and finding a weight vector which best represents the importance of each classifier in the final response of the ensemble. The combination of feature selection and parameterization should help create even more diversity. In order to evaluate the performance of the proposed approach, we use some benchmarks to compare with other classification algorithms, including some change in the fitness function of our approach.

1. Introduction

In the last decades, researches have been proposing to improve the accuracy and stability of the predictive systems generated by machine learning algorithms. One method that stands out is the combination of multiple learning algorithms, known as ensembles or committees. The main idea of using ensembles is that the combination of different individual classifiers (components) can offer complementary information about unknown instances, improving the quality of the overall classification in terms of generalization and accuracy [51].

An effective ensemble system should balance the individual accuracy against diversity among its components, i.e, the ensemble should consist of a set of classifiers that are not only highly accurate, but whose errors are uncorrelated [29–31,37,38]. Thus, when combining them, individual failures will be minimized. The diversity can be reached when the base components are built under different circumstances, such as: datasets, parameter setting and learning algorithm type.

In this way, a ensemble system can be seen as a structure in three levels: input space, components and the combination block. In [40,41], it has been considered an ensemble of LS-SVM (Least Squares Support Vector Machine) with RBF (Radial Basis Function) kernel and linear combination as combination method. In [40], the focus of the work was just the combination block level, using a genetic algorithm (GA) to analyze the importance of each LS-SVM in the set, by means of a weight vector, while its parameters were arbitrarily selected. In [41], the use of the GA has also been extended to components level. Thus, the GA had to find effective values for the parameters σ (Gaussian width) and C (regularization term) of each LS-SVM and the weight vector measuring their importance in the set. On both works, any kind of optimization was used in the input space of the ensemble. In [40], Bagging strategy was used to select the training dataset for each classifier and in [41], Random Subspace method was employed to make each LS-SVM responsible for the classification of a subproblem with a lower dimension than the original problem.

What we propose here is a step forward in the GA acting area, extending it also for the input space level of the ensemble. Now the GA will also realize a feature selection, so that different subproblems will be generated, and a model selection, including or not a classifier in the ensemble. Thus, the GA has a multilevel

* Corresponding author.
E-mail addresses: caapadilha@inf.ufrgs.br, carlos.engcomp@gmail.com (C.A.d.A. Padilha), barone@inf.ufrgs.br (D.A.C. Barone), adriao@dca.ufrn.br (A.D.D. Neto).

http://dx.doi.org/10.1016/j.knosys.2016.05.033
0950-7051/

task: feature selection on the input space level, model selection and parameter optimization on the components level and finding a weight vector to measure the importance of each LS-SVM in the ensemble on the last level. We believe that the inclusion of feature selection will help to create even more diversity among the components. Then the population of solutions is evaluated by a fitness function defined as the quadratic error norm of the ensemble, the same used in [40,41].

In order to evaluate the performance of multilevel approach, we compare the results with other algorithms, including some modifications on the fitness function of this approach.

This paper is organized as follows: Section 2 presents some related works in the state of the art. Section 3 introduces the theoretical background related to this work, that is, a brief explanation of LS-SVM, ensembles systems and genetic algorithm in subsections. Section 4 describes our proposed method, while Section 5 has the experimental results and its analysis. Section 6 presents the conclusions and future works.

2. Related works

In this section, several approaches involving ensemble designing and evolutionary algorithms will be presented. These approaches used evolutionary algorithms aiming the enhancement of 1 or 2 levels in the ensemble, most of them using genetic algorithms or genetic programming.

2.1. Feature selection

In the context of ensembles, the purpose of applying feature selection methods is to reduce the number of attributes presented to the base classifiers, in addition to dealing with the problems of dimensionality and diversity among the members of such systems.

There are many works in the literature involving feature selection methods and ensembles such as in [1,5,7,13,19,21,25,30,38,52]. In [21,52], the authors demonstrated that even a simple random sampling in the features space may be considered a satisfactory method for increasing the accuracy of ensemble systems. In [7], the approach is divided in two stages. First, an appropriate attribute subset size M is found by testing the accuracy of variously sized random subsets of attributes. In the second stage, the classification accuracy of randomly selected M-attribute subsets is evaluated. In [57], the amount of selected features for use is composed by the first eigenvectors of the data covariance matrix and dimensions randomly selected. In [5], the random forests algorithm ranks the attributes in terms of their contribution to the classification accuracy and it can be used to select the most useful features.

An early work to consider the use of GA to perform feature selection is [45]. In [54], GA is applied to select attributes for rule based classifiers. In the context of ensemble feature selection, [18,39] used a GA to search over the entire feature space, but considering only one ensemble.

Guerra-Salcedo and Whitley [18] tested four types of ensemble methods, including Bagging and Adaboost (using the complete set of features), Ho's method [21] and their method. The results presented that ensembles constructed using feature selected by the GA showed best performance, followed by RSM (Random Subspace Method) [21].

In [38], Genetic Ensemble Feature Selection (GEFS) used variable feature subset size to promote diversity among the classifiers and allowing features to be selected more than once. The ensemble components are evaluated in terms of both accuracy and diversity. The diversity was calculated as the average difference between the prediction of component classifiers and the ensemble. GEFS reached better results at around 2/3 of the 21 datasets tested.

Kim et al. [25] proposed the Meta-Evolutionary Ensembles (MEE) that consider multiple ensembles simultaneously and allows each component classifier to move into the best-fit ensemble. Genetic operators change the size of ensembles and membership of individual classifier over time. The population is initialized with randomly selected features and a random ensemble assignment. The classifiers compete with each other only if they belong to the ensemble. They are evaluated and rewarded based on two criteria, accuracy and diversity. The ensemble with highest accuracy is defined as their final classification model. Compared to the traditional ensembles (Bagging and Boosting) and GEFS, the resulting ensemble shows comparable performance while maintaining a smaller structure.

In [1], the authors used both filter and wrapper approaches to select salient features for classification ensembles. In the first phase, the statistical paired t-test is exploited to eliminate redundant features. In the second phase, the genetic algorithm is employed to determine the feature sets for each ensemble member. As in [30], each ensemble is coded in a chromosome.

Recently, Emmanuella et al. [13] applied GAs and others two well-known optimization techniques (Particle Swarm Optimization and Ant-Colony Optimization), in both mono and bi-objective versions, to choose features subsets for each individual ensemble member. The feature selection procedure used filter-based methods that simulated the idea of individual (mono-objective) and group (bi-objective) diversities, so the optimization techniques try to maximize these measures.

2.2. Ensemble selection

Most works involve choosing the base classifiers to include in the final ensemble [2,3,10,20,62]. The approach proposed by Zhou et al. [62] called GASEN which trains several individual neural networks at first. Then it assigns random weights to those networks and employs a GA to evolve the weights so that they can represent to some extent the fitness of the neural networks in constituting an ensemble. Finally, it selects the best subset of classifiers to constitute an ensemble based on minimizing the generalization ensemble error. GASEN is started with twenty neural networks, but the ensemble generated has far less than twenty. In [20], Hernandez-Lobato et al. propose a genetic pruning ensemble approach and compare with other heuristics (Reduce-error, Kappa and Early Stopping pruning). A probabilistic ensemble pruning algorithm is introduced in [10] to approximate the component weights using the Expectation Propagation algorithm.

In [41], the authors used a GA to optimize the internal parameters of the LS-SVMs (Gaussian width and regularization parameter) that constitute the ensemble. The pruning process was implicitly realized by the weight vector when the outputs were combined. A classifier with a very low weight value was discarded.

Soares et al. [48] presented a comparison of GA and Simulated Annealing (SA) based approaches for the automatic development of Neural Network (NN) ensembles. The ensemble construction is performed by two main steps: (1) Generation of candidate NN models; (2) Selection of the best subset of models and the optimal combination strategy taking into account the following factors: diversity, training ensemble members and combination strategy.

In [44], the construction of stacking ensembles using Artificial Bee Colony (ABC) [24] algorithm is proposed. The first implementation consists of using ABC to select the base classifiers and meta-classifier is a fixed learning algorithm. In the second implementation, the optimal subset of classifiers and an optimal meta-classifier are selected simultaneously.

Yao and co-works have an extensive number of works involving ensemble selection [2,3,9]. In [2], a new approach was proposed using multiobjective genetic programming (MOGP) optimization to