Applied Soft Computing 47 (2016) 262–270

Contents lists available at ScienceDirect

Applied Soft Computing

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

Self-correcting ensemble using a latent consensus model

Namhyoung Kim [a], Youngdoo Son [b], Youngjo Lee [c], Jaewook Lee [b,*]

[a] Department of Applied Statistics, Gachon University, 1342 Seongnamdaero, Sujeong-gu, Seongnam-si, Gyeonggi 461-701, South Korea
[b] Department of Industrial Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 151-744, South Korea
[c] Department of Statistics, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 151-744, South Korea

ARTICLE INFO

Article history:
Received 13 March 2015
Received in revised form 28 August 2015
Accepted 26 April 2016
Available online 7 June 2016

Keywords:
Ensemble
Latent consensus model
Self-correction
Decision tree
Artificial neural network

ABSTRACT

Ensemble is a widely used technique to improve the predictive performance of a learning method by using several competing expert systems. In this study, we propose a new ensemble combination scheme using a latent consensus function that relates each predictor to the other. The proposed method is designed to adapt and self-correct weights even when a number of expert systems malfunction and become corrupted. To compare the performance of the proposed method with existing methods, experiments are performed on simulated data with corrupted outputs as well as on real-world data sets. Results show that the proposed method is effective and it improves the predictive performance even when a number of individual classifiers are malfunctioning.

1. Introduction

In the real world, people obtain professional advice from several experts before finally deciding on significant matters such as investing financially, seeking treatment for a disease, and buying products. Combining the opinions of several different specialists is natural. In artificial intelligence and data mining, ensemble systems are techniques that combine multiple experts opinions (e.g., classifier) to obtain better predictive performance than using a single opinion. Ensemble methods are also known as multiple classifier systems, committee of classifiers, or mixture of experts.

Using the ensemble technique has several advantages. First, ensemble learning improves accuracy and robustness better than a single model does. Each classifier in an ensemble may capture the big picture of the problem and the ensemble technique may obtain more sensitive results by making up for each weak learner. Combining diverse, independent multiple predictors reduces variance and bias because of less dependence on the outliers of the training sets, which increases functional flexibility. This combination may also reduce the total error when each error occurs in different directions. The ensemble technique reduces the risk of selecting a poor classifier by averaging the outputs.

Second, the ensemble technique is suitable for cases when the size of the data set to be analyzed is extremely large or extremely small. With the rapid development of hardware and software technologies, the size of data increases at a fast rate. Applying existing data mining techniques to large data is difficult [20]. An extremely small size of an available data set is problematic. Ensemble techniques can be useful when the size of the data set is extremely small because these techniques reproduce the training data set by using a resampling technique.

Third, ensemble systems provide the means to solve difficult problems. The complex decision boundary that divides data sets cannot be learned by using a simple linear model. However, an ensemble can learn the complex boundary or function by appropriately combining simple classifiers.

Lastly, ensemble systems can be applied to data sets with different data types, such as medical image data from different sources such as MRI, FDG, or PIB, as well as numerical data with text information. These data sets are more informative but difficult to analyze using a single model. In such cases, we can train different models for each data type and then collect them to create a final model.

Most ensemble learning systems consist of two phases. The first phase is the building model process wherein each classifier is diversified, and the second phase combines the outputs in specific ways. A number of different algorithms for the first phase of ensemble learning have been suggested. When each classifier that forms an ensemble system is highly diverse, the effect of ensemble systems can be maximized. A number of algorithms can achieve diversity by using resampling techniques or different training parameters for different models.

* Corresponding author.
E-mail addresses: nhkim@gachon.ac.kr (N. Kim), hand02@snu.ac.kr (Y. Son), youngjo@snu.ac.kr (Y. Lee), jaewook@snu.ac.kr (J. Lee).

http://dx.doi.org/10.1016/j.asoc.2016.04.037
1568-4946/

Popular ensemble methods are bagging, random forest, boosting, and adaBoost. Several approaches for the second phase of ensemble combination are also available. Existing methods use majority voting, simple averaging of outputs, or the weighted sum of the votes of the weak learners. The weights are different for each component classifier of the ensemble system. The strategies used to calculate the weights are grouped as trainable and non-trainable methods. When weights depend on the performance or results of predictors, we do not have to train the weights. However, trainable methods require another training algorithm for computing the weights.

In this study, we propose a new ensemble combination scheme that uses a latent consensus function with related predictors. The proposed method aims to reveal better predictive power and functional reliability compared with the existing methods. We intend to show through simulation that the proposed method works effectively in situations when the component predictors in the ensemble fail or does not work well in predicting an output for a system malfunction.

The paper is organized as follows. In Section 2, we summarize related literature and algorithms. In Section 3, we describe the proposed method. Section 4 shows simulation results for predictor malfunction. We present the results of the empirical analysis and comparison of the proposed method with other ensemble methods in Section 5. We conclude the paper in Section 6.

2. Literature review

The concept of ensemble systems was proposed in [6], which discussed the division of feature space. Since then, several algorithms have been proposed [20,12,26,17,18,24] and applied to diverse fields [4,5]. Most of the contemporary ensemble methods develop an ensemble system based on the following equation:

$$E = \sum_{i=1}^{m} w_i c_i, \tag{1}$$

where E is an ensemble, c_i is an individual base predictor, and w_i is the weight for ith predictor. Fig. 1 shows that the predictions of the individual models c_i, $i = 1, 2, \ldots, m$ are combined using w_i. These weights can vary depending on combination scheme. As mentioned, ensemble systems obtain diversity by using a resampling technique, different training parameters, and different features. According to a combination scheme, ensembles are divided into two main classes, namely, ensembles combined by learning and ensembles combined by consensus.

Bagging is the simplest algorithm used to construct an ensemble [1], and is also known as bootstrap aggregating. Bagging creates individual predictors by training randomly selected training set. Each training set is generated by randomly with replacement, n examples. In bagging, the ensemble is formed by majority voting. Bagging does not consider the performance of each predictor. It

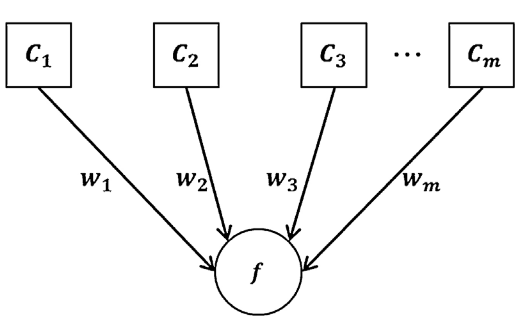

Fig. 1. Ensemble of classifiers.

averages the predictions of individual models in an unsupervised scheme. In bagging, the weights are uniform, i.e., the ensemble prediction is given by

$$E = \frac{1}{m}\sum_{i=1}^{m} c_i. \tag{2}$$

In classification, majority voting is used to predict the class of data. Thus, bagging is a simple but powerful method. A random forest is an ensemble technique for decision trees algorithm that combines the concept of bagging and the random selection of features [2].

Boosting combines multiple base predictors by learning. It uses the information of the performance of previous predictors to select the most informative data as a training data set. This kind of ensemble is also known as supervised scheme. Boosting algorithms learn iteratively weak classifiers using a training set selected according to the previous classified results, and then combine these classifiers with different weights to create an ensemble [10]. The weights are determined by the performance of the weak learners. Boosting results sometimes show poor performance because of overfitting the training set. Boosting method for updating the probabilities may be overemphasizing noisy data. Therefore, if noise exists in the data, boosting performance may perform poorly. Other ensemble systems proposed include stacked generalization and mixture of experts model. These systems are similar in such a way that both have another learning phase for computing the weights. In stacked generalization, the outputs of each classifier are used as inputs to learn the relationship between the ensemble outputs and actual classes [27]. Similarly, the mixture of experts model also uses a second level classifier, but the inputs are the training data instances rather than the outputs of classifiers [17].

In this study, we assume that the classifier outputs are already given. Thus, the diversity of the classifiers is not our focus. We propose a new trainable ensemble combination method for computing weights. In the following section, we describe our proposed method.

3. Proposed method

In this paper, we propose an ensemble combination scheme. In contrast to the conventional ensemble combination scheme that follows (1) (or Fig. 1), the proposed method extracts a latent consensus from opinions of experts. To be specific, we build a latent consensus model where each expert predictor c_i with $\mathbb{E}[c_i(\mathbf{x})] = \mu_i$ reflects the so-called latent consensus function f as follows:

$$c_i(\mathbf{x}) - \mu_i = \lambda_i(f(\mathbf{x}) - \mu_f) + \theta_i\delta_i(\mathbf{x}), \quad i = 1, 2, \ldots, m \tag{3}$$

where λ_i and θ_i are constants, and δ_i is a specific noise term with zero mean and unit variance. δ_i is assumed to be uncorrelated with f and δ_j's for $j \neq i$, and $f(\cdot)$ is assumed to have mean μ_f and variance σ^2. The conceptual model which extends and generalizes a latent one-factor model is described in Fig. 2. Given a data set $\mathcal{D}$, it will be shown in this section that our final ensemble combination model of experts c_i can share the behavior of (1) but approximately as follows.

$$E = \hat{f}(\mathbf{x}) \simeq \sum_{i=1}^{m} w_i^{\mathcal{D}} c_i(\mathbf{x}), \quad i = 1, 2, \ldots, m \tag{4}$$

where $w_i^{\mathcal{D}}$ is a weight to be learned from data $\mathcal{D}$.

We next describe how to calculate $w_i^{\mathcal{D}}$ as well as λ_i measuring a relative consensus of each individual experts. Eq. (3) can be equivalently written in the vector form

$$\mathbf{c}(\mathbf{x}) - \boldsymbol{\mu} = (f(\mathbf{x}) - \mu_f)\boldsymbol{\lambda} + \mathbf{D}_{\boldsymbol{\theta}}\boldsymbol{\delta}(\mathbf{x}), \quad i = 1, 2, \ldots, m. \tag{5}$$