Neurocomputing 167 (2015) 459–466

Contents lists available at ScienceDirect

Neurocomputing

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

Optimal construction of one-against-one classifier based on meta-learning

Seokho Kang, Sungzoon Cho *

Department of Industrial Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 151-742, Republic of Korea

ARTICLE INFO

Article history:
Received 28 August 2014
Received in revised form
11 February 2015
Accepted 20 April 2015
Communicated by Dorothy Ndedi Monekosso
Available online 29 April 2015

Keywords:
Multi-class classification
Multiple classifier system
Meta-learning
One-against-one
Diversified one-against-one
Optimally diversified one-against-one

ABSTRACT

A commonly used strategy for solving a multi-class classification problem is to decompose the original problem into several binary subproblems. The recently proposed method, diversified one-against-one (*DOAO*), constructs a one-against-one classifier by selecting the best classifier for each class pair from the set of heterogeneous base classifiers. It was found to yield better classification accuracy than other one-against-one classifiers that are based on individual classification algorithms. This paper presents a novel method, called optimally diversified one-against-one (*ODOAO*) which is an improvement of *DOAO*. *ODOAO* is based on meta-learning, and seeks to construct a multiple classifier system where a meta-classifier effectively combines the outputs from all the heterogeneous base classifiers that are trained using various classification algorithms for every class pair. Experimental results show that *ODOAO* outperforms *DOAO* and other one-against-one based methods with statistical significance.

1. Introduction

Classification is a type of supervised learning task that involves predicting output variables consisting of a finite number of categories called classes. In a classification task, a classification algorithm $\mathcal{A}$ defines its hypothesis space $\mathcal{H}_\mathcal{A}$. Training a classifier is to find the hypothesis $h \in \mathcal{H}_\mathcal{A}$ that approximates the true function f given a set of instances called a training dataset. Thus, a classifier corresponds to its hypothesis in the hypothesis space. Finding the hypothesis that is closest to the true function f is crucial for obtaining high classification accuracy. Multiple classifier system (MCS), which combines the outputs from a diverse of classifiers, has received considerable attention and has been studied by a wide range of researchers [1–5].

In general, an MCS offers better classification accuracy and robustness than any individual classifier. Dietterich [6] explained three fundamental reasons for why an MCS successfully performs well. The first is a statistical reason. Given a finite number of training instances, many hypotheses are equally good. Therefore, averaging these hypotheses may result in a more stable approximation of f. The second is a computational reason. Because the hypothesis space is so large, a heuristic search is conducted to find the best hypothesis. However, the search may get stuck at a local optimum. Repeating the search with several random starts provides a better chance of finding the global optimum. The third is a representational reason. The true function f may not be represented by any of the hypotheses in the hypothesis space $\mathcal{H}_\mathcal{A}$, but may be better approximated by aggregating several hypotheses.

The concept of an MCS has also been successfully applied to multi-class classification problems. This is typically accomplished by decomposing the original problem into several binary subproblems. The base classifiers for the subproblems constitute an MCS. Regarding the decomposition strategy, the two commonly used approaches are one-against-one and one-against-rest [4,7]. Several experimental studies argued that the one-against-one approach outperforms the one-against-rest approach [8,9], and that such decomposition strategy is also effective for classification algorithms that are capable of dealing with multi-class classification problems directly [8,10–12].

Recently, we proposed diversified one-against-one (*DOAO*) [13] which seeks to find the best classification algorithm for each class pair when applying the one-against-one approach to multi-class classification problems. With *DOAO*, the best classification algorithm for each class pair is selected as having the minimum validation error. The experimental results confirmed that *DOAO* outperforms other one-against-one classifiers that are based on individual classification algorithms or voting of them. This is because, according to the so-called no-free-lunch theorem [14],

* Corresponding author. Tel.: +82 2 880 6275.
E-mail addresses: prokids@snu.ac.kr (S. Kang), zoon@snu.ac.kr (S. Cho).

http://dx.doi.org/10.1016/j.neucom.2015.04.048
0925-2312/

there is no single algorithm that always outperform the others for every classification problem [15–17]. Employing a variety of classification algorithms takes the advantages of different inductive biases of the algorithms, thereby yielding better classification accuracy. Such effectiveness can be also explained as the extension of the hypothesis space. An MCS with different classification algorithms is more likely to obtain a better hypothesis by searching the union of hypothesis spaces defined by different algorithms.

However, there are two major limitations to *DOAO*. First, the minimum validation error does not always indicate the minimum test error, especially when comparing heterogeneous classifiers [18]. Therefore, selecting the classifier based on the validation error does not guarantee the optimal. Second, several classifiers that are properly fused can outperform the single best classifier [19,20]. To address such limitations, we consider employing a meta-classifier to find the optimal combination of base classifiers. When a meta-classifier is employed, the new instance is first classified by the base classifiers, and the results are used as inputs for the meta-classifier to determine the final classification result.

In this paper, we propose optimally diversified one-against-one (*ODOAO*) that improves *DOAO* in order to achieve better classification accuracy. *ODOAO* seeks to find the optimal combination of base classifiers that are built for every class pair and candidate classification algorithm according to the concept of *DOAO*. To do this, a meta-classifier is trained based on stacking [21], where the input variables are the predicted labels from the base classifiers on the validation dataset, and the output variable is the target label. *ODOAO* is further enhanced by applying a classification algorithm that can effectively deal with high dimensionality and non-linear relationship between the predictions of the base classifiers when training the meta-classifier. We investigate the effectiveness of the proposed method through experiments on multi-class benchmark datasets.

The rest of this paper is organized as follows. In Section 2, we briefly review the related work. In Section 3, we describe our proposed method. We report the experimental results in Section 4, and offer conclusions and future work in Section 5.

2. Related work

2.1. Multiple classifier systems for multi-class classification

When the number of classes in a classification problem is more than two, the problem is called a multi-class classification problem. To solve a multi-class classification problem, three strategies can be considered. The first is simply to use the classification algorithms that solve the problems directly, such as decision trees (DT), *k*-nearest-neighbors (*k*NN), and artificial neural networks (ANN).

The second strategy is to decompose the original problem into several binary subproblems. This strategy permits the use of classification algorithms that were originally designed for binary classification problems, such as support vector machines (SVM), logistic regression (LR), and linear discriminant analysis (LDA). Two common approaches to this strategy are one-against-one and one-against-rest [4,7]. Supposing that a c-class classification problem is given, the one-against-one approach builds $c(c-1)/2$ different binary classifiers for all possible class pairs. Given the same problem, on the other hand, the one-against-rest approach builds c different binary classifiers, where each separates a single class from all the remaining classes.

The third strategy is to decompose the original problem into several one-class subproblems [22–27]. This strategy utilizes one-class classification algorithms, such as parzen window, support vector data description, kernel principal component analysis, for multi-class classification. For a c-class classification problem, c one-class classifiers are built, each of which is trained on a single respective class. Each classifier evaluates the degree of belonging to a class independently.

The latter two relate to the concept of an MCS. They can treat multi-class classification problems effectively. Since the decision boundary for multi-class classification problems tends to more complex than it is for one-class or binary classification problems, solving several smaller subproblems is more preferable [8]. Among them, the second strategy has received more attention and has been more often used. Regarding this strategy, Galar et al. [8] compared these two approaches for various classification algorithms, and demonstrated that the one-against-one approach outperforms the one-against-rest in most cases. Our previous work [13] proposed *DOAO* that selects the best classification algorithm for each class pair when applying the one-against-one approach. This work is also in accordance with the one-against-one approach and utilizes various classification algorithms.

2.2. Classifier combinations in multiple classifier systems

An MCS is composed of a diverse of base classifiers by offering diversity to the classifiers. The classification results are different depending on the combination strategy, despite having the same set of base classifiers. Therefore, choosing the appropriate combination strategy is an important issue. There have been various combination methods proposed [4,7], and the two basic ideas are known as classifier selection and classifier fusion [20,28].

Classifier selection finds the best classifier from among a set of base classifiers. It works well when one classifier is superior to the others, particularly with heterogeneous base classifiers that have come from different classification algorithms. Classifier selection is effective when a few classifiers are superior to others. Our previous method *DOAO* is on the basis of classifier selection [13], thereby the classification accuracy is improved.

Classifier fusion, by contrast, utilizes the group consensus of the whole base classifiers, and therefore it depends on the comparable success of the base classifiers. Classifier fusion is generally used for combining homogeneous base classifiers. Majority voting is a simple but the most popular method, which finds the largest selected class from the base classifiers. However, this method may fail when the majority of base classifiers provide incorrect classification results [29]. Instead, weighted voting puts more weights for more superior base classifiers [30]. The voting-based methods use linear combination of the outputs from the base classifiers to make final decision, while meta-learning can be taken into account to find the non-linear combination of base classifiers.

The proposed method in this paper seeks to find the optimal combination of the outputs from the base classifiers, with considering non-linear structure. Therefore, we utilized meta-learning, specifically, stacking [21]. We provide a detailed description of stacking in Section 2.3.

2.3. Stacking

Stacking [21] is a meta-learning [31] technique that attempts to induce which classifiers are reliable and which are not, and is usually employed to combine classifiers from different classification algorithms. The basic idea of stacking is to build a meta-classifier that predicts target labels by combining the predictions of base classifiers. Suppose that a set of base classifiers $\mathcal{C}_1,\dots,\mathcal{C}_L$ and a set of instances $\mathcal{D}=\{\boldsymbol{x}_t, y_t\}_{t=1}^{N}$ is given, the predictions for the N instances of each base classifiers are $\hat{y}_t^i=\mathcal{C}_i(\boldsymbol{x}_t)$, $t=1,\dots,N$, $i=1,\dots,L$, and they constitute a meta-dataset $\mathcal{M}=\{(\hat{y}_t^1,\hat{y}_t^2,\dots,\hat{y}_t^L), y_t\}_{t=1}^{N}$ [32]. This meta-dataset is used to train the meta-classifier. During the test phase, a test instance is first classified