Information Fusion 20 (2014) 49–59

Contents lists available at ScienceDirect

Information Fusion

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

Convex ensemble learning with sparsity and diversity

Xu-Cheng Yin [a,b,*], Kaizhu Huang [c], Chun Yang [a], Hong-Wei Hao [d]

[a] Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[b] Beijing Key Laboratory of Materials Science Knowledge Engineering, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[c] Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China
[d] Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 6 May 2013
Received in revised form 4 November 2013
Accepted 14 November 2013
Available online 28 December 2013

Keywords:
Classifier ensemble
Sparsity
Diversity
Convex quadratic programming

ABSTRACT

Classifier ensemble has been broadly studied in two prevalent directions, i.e., to diversely generate classifier components, and to sparsely combine multiple classifiers. While most current approaches are emphasized on either sparsity or diversity only, we investigate classifier ensemble focused on both in this paper. We formulate the classifier ensemble problem with the sparsity and diversity learning in a general mathematical framework, which proves beneficial for grouping classifiers. In particular, derived from the error-ambiguity decomposition, we design a convex ensemble diversity measure. Consequently, accuracy loss, sparseness regularization, and diversity measure can be balanced and combined in a convex quadratic programming problem. We prove that the final convex optimization leads to a closed-form solution, making it very appealing for real ensemble learning problems. We compare our proposed novel method with other conventional ensemble methods such as Bagging, least squares combination, sparsity learning, and AdaBoost, extensively on a variety of UCI benchmark data sets and the Pascal Large Scale Learning Challenge 2008 webspam data. Experimental results confirm that our approach has very promising performance.

1. Introduction

A variety of classifiers with different feature representations, construction architectures, learning algorithms, or training data sets usually exhibit different and complementary classification behaviors. Combination of their classification results can usually yield higher performance than the best individual classifier. Consequently, Classifier ensemble has been intensively studied for a long period [1–5]. In this field, many famous models have been proposed [6–15]. At the same time, classifier ensemble methods have been widely applied in many real-world applications [16,17]. Generally speaking, besides the accuracy of classifier components, there are two very important issues relevant to the performance of a classifier ensemble: (1) How to generate diverse classifiers; and (2) how to combine available multiple classifiers.

On the one hand, diversity learning for an ensemble is performed in two approaches, i.e. seeking implicit or explicit diversity [18]. The common way for the prior approach is to train individual classifiers on different training sets, for example Bagging [19], and Boosting [20,21]. For the latter approach, the general way is to train multiple classifiers by using different classifier architectures or different feature sets [1,22,23]. On the other hand, there are also numerous strategies for combining multiple classifiers. Some famous combination methods include averaging (e.g., Bagging [19]), weighting (e.g. Boosting [21]), and non-linear combining (e.g., Stacking [24]). Given a number of available component classifiers, many researchers argue that the sparse ensemble or pruning ensemble, which ensembles of parts of available component classifiers, may be better than ensemble as a whole [10,25–29].

The diversity learning and the sparsity learning[1] for classifier ensemble have different purposes and algorithmic treatments. Consequently, algorithms implementing these different learning strategies are initially separate and independent. Obviously, it is more rational to combine classifiers with both sparsity and diversity.

* Corresponding author at: Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China. Tel.: +86 10 82371191; fax: +86 10 62332873.

E-mail addresses: xuchengyin@ustb.edu.cn (X.-C. Yin), kaizhu.huang@xjtlu.edu.cn (K. Huang), ych.learning@gmail.com (C. Yang), hongwei.hao@ia.ac.cn (H.-W. Hao).

[1] In our context, the term of "sparsity learning" means learning and selecting a subset of a few classifiers for an ensemble, i.e., learning a sparse weight vector for linearly combining a list of available component classifiers. Specifically, we define "sparsity" as the percentage of selected component classifiers (which are with nonzero weights) in the ensemble. Please note that in Algebra, a "sparse matrix" is a matrix populated primarily with zeros as elements of the table, where the fraction of zero elements is called the "sparsity".

1566-2535/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.inffus.2013.11.003

However, there have been very few researchers who focus on such techniques for ensemble learning. Chen and Yao et al. analyzed diversity and regularization in neural network ensembles for balancing diversity, regularization and accuracy of multi-objectives [12,30]. Their methods were specifically designed for component classifier training and combination in neural network ensembles.

In this paper, for a general classifier ensemble with available numerous component classifiers, we formulate the sparsity and diversity learning problem in a general mathematical framework. In particular, derived from the error-ambiguity decomposition, we design a convex ensemble diversity measure. Consequently, accuracy loss, sparseness regularization, and diversity measure can be balanced and combined in a quadratic programming problem. We prove that the final convex optimization leads to a closed-form solution.

The main contributions of this work are summarized as follows. First, we present a general mathematical framework for learning both sparsity and diversity in classifier ensemble. Unlike conventional methods with implicit notes of sparsity or/and diversity, our approach explicitly combines and optimizes both in a unified learning model. Second, derived from the error-ambiguity decomposition, the sparsity and diversity learning can be formulated in a convex quadratic programming optimization problem. Distinct from those conventional methods with some heuristic or multi-stage algorithms [31,32], our approach leads to a closed-form solution, which is highly convenient for real ensemble learning problems.

The rest of this paper is organized as follows. Related work is presented in Section 2. Section 3 describes the problem statement and several learning models for classifier ensemble. Section 4 demonstrates our sparsity and diversity learning algorithm with convex quadratic programming. Comparison experiments with UCI data sets and the Pascal Competition 2008 spam data are conducted in Section 5. Final remarks are presented in Section 6.

2. Related work

Classifier ensemble can be divided into two categories. The first one aims at learning multiple classifiers at the feature level, where multiple classifiers are trained and combined in the learning process, e.g., Boosting [20] and Bagging [19]. The second tries to combine classifiers at the output level, where the results of multiple available classifiers are combined to solve the targeted problem, e.g., multiple classifier systems, or mixture of experts [14]. In this paper, we focus on the second one. Namely, given multiple classifiers (available or sequently learned, homogeneous or heterogeneous), classifier ensemble is learned by combining intelligently these component classifiers.

Classifiers with different features, architectures, learning algorithms, and training data usually bring different and complementary classification performances and the combination of their decisions can bring to higher performance than individual component classifiers. Generally speaking, besides the accuracy of classifier components, the performance of an ensemble mainly relies on the diversity of the classifier components, and the learning strategy for combination. Consequently, the research focuses include: How to generate diverse classifiers, and how to combine available multiple classifiers.

First, in many cases diversity can ensure that all the individual component classifiers craft uncorrelated errors. In classifier ensemble, diversity learning is performed in two approaches, i.e., seeking implicit or explicit diversity [18]. The common way for the prior approach is to train individual classifiers on different training sets, for example Bagging [19], and Boosting [20,21]. Bagging generates several different training sets with bootstrap sampling from the original training set and then trains a component classifier from each of those training sets. Boosting generates a series of component classifier whose training sets are different and determined by the performance of former classifiers. Training instances which are wrongly predicted by former classifiers will play more important role in the training of later classifiers. Similarly, when combining nearest-neighbor classifiers, Zhou et al. proposed an ensemble with local learners for utilizing multi-modal perturbation to help generate accurate but diverse component learners [33]. Yu et al. proposed the diversity regularized machine, which efficiently generates an ensemble of assorted support vector machines [34]. More recently, Hosseinzadeh and Reza proposed a new classifier combining strategy with virtual voting by random projection, which used the distortion to virtually generate different training sets from the total available samples [35]. Li et al. tried to present a theoretical study on the effect of diversity on the generalization performance of voting in the PAC-learning framework for classifier ensemble. Following this analysis, they also proposed the diversity regularized ensemble pruning method [32].

As in the latter approach for diversity learning, the general way is to train multiple classifiers by using different classifier architectures or different feature sets [1,22,23,36]. Liu investigated and combined classifiers with different structures using a variety of combination rules, such as sum-rule, product-rule, linear discriminants, and weighted combination [22]. Yin et al. proposed a variant of boosting and adaptively integrated classifiers built on different features [23]. Cevikalp and Polikar used local classifier accuracies to weight classifier outputs and proposed a dynamic approach with quadratic programming to combine classifiers built on different regions of the input space [37].

Regarding the Random Forests approach [38], it can exploit implicit and explicit diversities together. The method combines the "Bagging" idea for instance sampling and the random selection of variables for feature selection.

Second, in the combination strategy, multiple classifiers are grouped with proper combination rules or learning methods. Considering the level of classifier outputs with available classifiers, the combination methods can be categorized as abstract level (class level), rank level (rank order), and measurement level (class scores) [22,39]. Generally, the class scores can provide richer and more direct information than the class labels and the rank orders. Therefore, numerous methods with the measurement level combination such as an average, linear or nonlinear combination rules are employed [2,22,40]. For example, Bagging combines component classifiers by majority voting and the most-voted class is predicted [19]. AdaBoost combines components by weighted majority voting, where weights are adaptively learned from training rounds [21]. Stacking uses a meta-learner (in the second-level) to learn and combine individual classifiers (from the first-level) [24]. Hao et al. also proposed a stacking-like combination method for neural network classifiers in handwritten Chinese character recognition [41].

Given a number of available component classifiers, most conventional approaches employ all of these classifiers to constitute an ensemble. In the literature, many researchers suggested that ensemble of some parts of the available component classifiers may be better than ensemble as a whole. This leads to sparse ensemble, pruning ensemble or selective ensemble for the combination of multiple classifiers [10,25–29,42,43]. Specifically, Martinez-Munoz et al. investigated several pruning strategies with ordered aggregation to select and prune classifiers in the ensemble, and found that the generalization error reached a minimum at intermediate numbers of available classifiers [28]. Zhou et al. analyzed the relationship between the ensemble and its component neural networks, declared that it may be better to ensemble many instead of all of the available neural networks, and proposed an