Neurocomputing 151 (2015) 434–448

Contents lists available at ScienceDirect

Neurocomputing

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

Enhancement of multi-class support vector machine construction from binary learners using generalization performance

Patoomsiri Songsiri [a], Thimaporn Phetkaew [b], Boonserm Kijsirikul [a,*]

[a] Department of Computer Engineering, Chulalongkorn University, Pathumwan, Bangkok 10330, Thailand
[b] School of Informatics, Walailak University, Thasala District, Nakhon Si Thammarat 80161, Thailand

ARTICLE INFO

Article history:
Received 11 September 2013
Received in revised form
24 July 2014
Accepted 19 September 2014
Communicated by H. Yu
Available online 28 September 2014

Keywords:
Support vector machine
Multi-class classification
Generalization performance

ABSTRACT

We propose several new methods to enhance multi-class support vector machines (SVMs) by applying the generalization performance of binary classifiers as the core idea. This concept is applied to the existing algorithms, i.e., the Decision Directed Acyclic Graph (DDAG), the Adaptive Directed Acyclic Graph (ADAG), and Max Wins. Although there have been many previous attempts to use information such as the margin size and number of support vectors as the performance estimators for binary SVMs, this type of information may not accurately reflect the actual performance of the binary SVMs. We demonstrate that the generalization ability that is evaluated using a cross-validation mechanism is more suitable for directly extracting the actual performance of binary SVMs than the previous methods. Our methods are built around this performance measure, and each of them is crafted to overcome the weakness of the previous algorithms. The proposed methods include the Modified Reordering Adaptive Directed Acyclic Graph (MRADAG), Strong Elimination of the classifiers (SE), Weak Elimination of the classifiers (WE), and Voting-based Candidate Filtering (VCF). The experimental results demonstrate that our methods are more accurate than traditional methods. In particular, WE provides superior results compared to Max Wins, which is recognized as one of the most powerful techniques, in terms of both accuracy and classification speed with two times faster in average.

1. Introduction

The support vector machine (SVM) [1,2] is a high performance learning algorithm that constructs a hyperplane to separate two-class data by maximizing the margin between them. There are two approaches for extending SVMs to multi-class problems, i.e., solving the problem by formulating all classes of data under a single optimization and combining several two-class subproblems. However, it is difficult and complex to solve the problem using the first method because of the increase in the number of classes and the size of the training data. Thus, the second method is more suitable for practical use. In this paper, we focus on the second approach.

To construct a multi-class classifier from binary classifiers, the *one-against-one* method trains each binary classifier on only two out of N classes and builds $N(N-1)/2$ possible classifiers. Several strategies have been proposed to combine the trained classifiers to make the final classification for unseen data. Friedman [3] suggested the combination strategy called *Max Wins*. In the classification process of Max Wins, every binary classifier provides one vote for its preferred class, and the class with the largest vote is set as the final output. Several previous works have been proposed to improve the aggregation of base binary classifiers using the Max Wins strategy [4–8]. However, the strategy may be affected by the problem of *incompetent classifiers* which may produce *nuisance* votes for unrelated classes and lead to misclassification [7–9].

One-against-the-rest has been proposed by some researchers [1,10]. This approach constructs a set of N binary classifiers, where each ith classifier is learned from examples in the ith class and examples in the remaining classes, which are labeled with positive and negative examples, respectively. The class that corresponds to the classifier with the highest output value is used to make the final output. Although one-against-the-rest may suffer from a training process, it is still an effective method for fast classification if sufficiently well-separated classifiers can be learned [11]. For some applications where testing time is a main concern especially for large scale classification problems, one-against-the-rest is applied and produces good results [12–14]. For large scale classification problems, methods using tree-based binary classification have also been proposed [15,16].

Error Correcting Output Code (ECOC) has been introduced as a multi-class classifier by Dietterich and Bakiri [17]. The code matrix

[*] Corresponding author. Tel.: +66 2 218 6956; fax: +66 2 218 6955.
E-mail addresses: patoomsiri.s@student.chula.ac.th (P. Songsiri), pthimapo@wu.ac.th (T. Phetkaew), boonserm.k@chula.ac.th (B. Kijsirikul).

http://dx.doi.org/10.1016/j.neucom.2014.09.021
0925-2312/© 2014 Elsevier B.V. All rights reserved.

with N rows and L columns is used in ECOC to represent the different combinations of positive and negative classes to construct L binary classifiers to distinguish among different N classes. To perform a classification, a test example is classified by all classifiers; then, the distance values from the binary classification results to the different classes are evaluated and the class with the closest distance is assigned to the final output class. Allwein et al. [18] extended the coding method to allow the binary model to be learned without considering some particular classes. However, the design of code matrices with different subsets of binary classifiers yields different abilities for separating classes, and the problem of selecting a suitable subset of binary classifiers is complicated with the large size of N. To obtain the suitable code matrix, some techniques using genetic algorithms have been proposed [19,20]. Pujol et al. [21] have proposed the code design using the discrimination of classes based on the mutual information between the feature data and the class label. In addition, Bagheri et al. [22,23] have introduced an added view on the third dimension of the code matrix that enables each binary classifier to use the different feature subsets, and proposed a genetic algorithm to search for the suitable three-dimensional code matrix.

Platt et al. [24] proposed the Decision Directed Acyclic Graph (DDAG) to reduce the evaluation time [25]. In each round, a binary model is randomly selected from all $N(N-1)/2$ classifiers. The binary classification result is used to eliminate the candidate output classes and to ignore all binary classifiers related to the defeated class. It guarantees that the number of classifications (applied classifiers) of the DDAG is always $N-1$. This recursive task is applied until there is only one remaining candidate class. However, the DDAG will produce misclassification if the selected *binary classifiers related to the target class* (hence forth *BCRT*) yield the incorrect answer. The likelihood that the DDAG produces misclassification increases as an increasing number of BCRTs are applied. To reduce this risk, Kijsirikul and Ussivakul [26] proposed the ADAG, which has a reversed triangular structure of the DDAG. It requires the target class to be examined against the other classes at most $\lceil \log_2 N \rceil$ times, whereas the DDAG may require up to $N-1$ times.

In addition, there have been many attempts to apply information such as the margin size [24], number of support vectors [27], and separability measures among the classes [28,29] to improve the performance of the multi-class classification. The margin size and the number of support vectors were applied to select the suitable binary classifiers in the DDAG [24,27]. The separability measure was used to automatically construct a binary tree of multi-class classification based on the minimum spanning tree [28]. Li et al. [29] used similar information to vote the preferred class for data in an unclassifiable region for both the one-against-one and one-against-the-rest techniques.

In this research, we investigate a framework to enhance three well-known methods: the DDAG, the ADAG, and Max Wins. Max Wins is currently recognized as one of the most powerful combining algorithms; Max Wins requires $N(N-1)/2$ classifications for an N-class problem, whereas the other two approaches reduce the number of classifications to $N-1$. We study the characteristics of these methods that lead to incorrect classification results. The first two techniques have identical hierarchical structures and the same limitation, i.e., they "*trust the individual opinion*" for making the decision to discard the candidate classes. Intuitively, if only one BCRT makes a mistake, the entire system will yield the incorrect output. The last technique, Max Wins, is based on the concept of "*trust the most popular opinion*" for making the decision to select the output class. If all $N-1$ BCRTs provide the correct answer, Max Wins will always provide the correct output class. However, if there is only one BCRT that provides the incorrect answer, it may lead to misclassification because of equal voting, or other non-target classes obtaining the largest vote, as demonstrated below. Examples that are incorrectly classified in this scenario can be recovered by our proposed strategies.

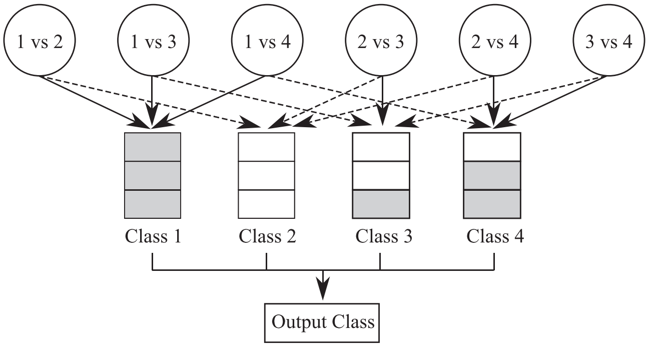

Fig. 1. An example of a four-class classification with Max Wins.

In this paper, we demonstrate that the above traditional methods can be improved, based similarly on the idea that if we access further important information of the *generalization performance* of all binary classifiers and properly estimate it, it can be used to enhance the performance of the methods. Based on this idea, we propose four new approaches: (1) the Modified Reordering Adaptive Directed Acyclic Graph (MRADAG), (2) Strong Elimination of the classifiers (SE), (3) Weak Elimination of the classifiers (WE), and (4) Voting-based Candidate Filtering (VCF). The first approach, the next two approaches, and the last approach are improved from the ADAG, the DDAG, and Max Wins, respectively. We also empirically evaluate our methods by comparing them with the traditional methods on sixteen datasets from the UCI Machine Learning Repository [30].

This paper is organized as follows. Section 2 reviews the traditional multi-class classification frameworks. Section 3 describes how to properly estimate the generalization performance of binary classifiers. Section 4 presents our proposed methodologies. Section 5 presents the experiments, explains the results, and provides a discussion. Section 6 concludes the research.

2. Multi-class support vector machines

2.1. Max wins

For an N-class problem, all possible pairs of two-class data are learned to construct $N(N-1)/2$ classifiers. All binary classifiers are applied to vote on the preferred class. A class with the maximum number of votes will be assigned as the final output class. This method is called Max Wins [3]. However, if there is more than one class that yields an identical maximum vote, the final output class can be obtained by random selection from the candidate classes with the equal number of votes. An example of the classification using this technique for a four-class problem is shown in Fig. 1. Each class will be voted (solid-line) or ignored (dash-line) by all related binary models. For example, class 1 has three related classifiers, i.e., 1 vs. 2, 1 vs. 3, and 1 vs. 4. The voting results of classes 1, 2, 3, and 4 are three, zero, one, and two, respectively. In this case, class 1 has the largest score; therefore, it is assigned as the final output class.[1]

[1] In case that the target class is class 1, *binary classifiers related to the target class* (*BCRT*) are classifiers 1 vs. 2, 1 vs. 3, and 1 vs. 4.