Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 71 (2008) 2747–2753
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Game theoretical analysis of the simple one-vs.-all classifier

Yuichi Shiraishi

Department of Statistical Science, The Graduate University for Advanced Studies, The Institute of Statistical Mathematics, Minami-Azabu, Tokyo 106-8569, Japan

Received 13 April 2007; received in revised form 16 August 2007; accepted 7 October 2007
Communicated by L.C. Jain
Available online 23 October 2007

Abstract

One of the popular multi-class classification methods is to combine binary classifiers. As well as the simplest approach, a variety of methods to derive a conclusion from the results of binary classifiers can be created in diverse ways. In this paper, however, we show that the simplest approach by calculating linear combinations of binary classifiers with equal weights has a certain advantage.

After introducing the ECOC approach and its extensions, we analyze the problems from a game-theoretical point of view. We show that the simplest approach has the minimax property in the one-vs.-all case.

Keywords: Multi-class classification; Error correcting output code; Game theory; Minimaxity

1. Introduction

Multi-class classification problems, where the number of classes C is greater than 2, occur very often in science. The classification with $C = 2$ is called binary classification. For binary classification problems, many effective methods such as SVM [7] or ada-boost [13] have been developed. However, for multi-class classification problems, there is no conclusive method and a lot of methods are still being proposed at present.

There are two approaches to deal with multi-class problems. The one is to consider loss functions that can treat more than two classes and minimize them directly by some algorithms (see e.g. [27,6,8,21,30]). Since this approach is based on a "true" loss function, its properties like consistency to Bayes error rate are easier to analyze [29,26]. However, it is often computationally infeasible for a large number of classes and samples.

The other approach, which is the main focus of this paper, is to combine binary classifiers to derive a conclusion for multi-class classification. This can take advantage of strong binary classifiers such as SVM and ada-boost, in addition to its computational efficiency and simplicity. In this approach, the binary classifiers are trained first, and then a combiner aggregates the outputs of the binary classifiers to make a final decision. This can be regarded as a specific case of the aggregation methodology, which uses many classifiers to draw a conclusion [19,20,22].

For designing the combined binary classifiers, the two most popular learning schemes are the "one-vs.-one" and "one-vs.-all". In the former, each of the binary classifiers discerns a certain class from another class, while in the latter each discerns a certain class from all the remaining classes. For combining the binary classifiers, the majority vote, the directed acyclic graph model, Bradley–Terry model and the error correcting output code (ECOC) model are very popular.

Both the majority vote and the directed acyclic graph approach can be used only in the one-vs.-one case. In the majority vote [14,15], each classifier casts one vote for its predicted class for a new data, then the class with the most votes is returned as an output. In the directed acyclic graph approach [23], each node is associated with each binary classifier. On each node, two classes are compared and one of the classes is removed from further consideration. After this process is repeated $C-1$ times and traversing the directed graph, $C-1$ classes are removed and the remaining class is returned as an output.

E-mail address: yshira@ism.ac.jp

0925-2312/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2007.10.002

Bradley–Terry model [5,10] is used for estimating the merits of multiple players based on paired comparison results. In multi-class classification [17], we can regard the results of binary classifiers as those of battles between the two classes. The combiner estimates the merit of each class using the model, and returns the strongest class as an output. Although the original method can deal with only the one-vs.-one case, a number of extensions are proposed for general cases [28,18]. One of the problems of this approach is that the output of each binary classifier has to be transformed to a probabilistic value suitable to Bradley–Terry model. Not all binary classifiers return a probabilistic value. Furthermore, when transforming the outputs into probabilistic values, the ambiguity regarding how to do so remains.

The ECOC approach [11,1], which can be viewed as a generalization of the majority vote, is free from what sorts of outputs binary classifiers return and the designing scheme of binary classifiers. In the ECOC approach, each class has a code, which is a binary sequence of the length equal to the number of the binary classifiers. Each bit in the code of a class corresponds to the desired output of the binary classifier for the inputs within the class. Each binary classifier is trained so that it returns "+1" and "−1" for the data in the class of the bit "+1" and "−1", respectively. In practice, given the probabilistic nature of the training sample and imperfection of the training, the outputs of the binary classifiers are not exactly equal to the code. Based on a certain probabilistic model of the corruption of the code, the combiner returns the class with the highest posterior probability. Usually, we only need to consider linear combinations of the results of binary classifiers for each class to do the maximum a posteriori (MAP) estimation. Considering those linear combinations as the score of each class, the combiner chooses the class with the highest score. The weights of the linear combination are usually fixed as -1, 0 or $+1$. These weights are determined automatically when we decide the scheme for learning binary classifiers such as the one-vs.-all or one-vs.-one. We call this scheme as the *simple ECOC* approach.

One property of the simple ECOC is that the combiners are not trained with training data, and the decision rule are determined by the learning scheme. This does not take into account varying confidence of each classifier. For example, when each binary classifier uses a different number of training samples, some classifiers return accurate results while others do not. An extension for incorporating such confidence is to introduce weights over the binary classifiers based on the training samples. There have been several works along this line (see, for example, [9,25]). Furthermore, by assuming generative models of the results of binary classifiers and learn the parameters of the models from training data, we can construct an infinite number of combination methods. However, although these methods slightly outperform the simple ECOC in some cases, it is hard to say that they lead to significant improvements on the whole. Their approaches ignore the information on how we designed binary classifiers when learning the combiner, whereas the simple ECOC uses that information.

There are some problems in training the combiner, however, as discussed in [12]. Re-using training samples for both learning binary classifiers and the combiner is quite problematic, because it tends to cause overfitting. Even when trying to separate the training samples for the combiner from the samples for the binary classifiers, how we do so is a very difficult problem. In addition, we do not always have sufficient training samples for the combiner to learn. Thus, the question is: Which is better, training the combiner or not?

In this paper, we attempt to give a justification to the non-learning method by game theoretical analysis. We restate the classification problem as a game between the "decision maker" (DM) who performs classification and "Nature" who determines the joint distribution of binary classifiers and class labels. We show that the simple, non-learning ECOC method is a minimax strategy in the one-vs.-all case.

The outline of our paper is as follows. In Section 2, we introduce the simple ECOC approach and its extensions. In Section 3, we analyze the problem game-theoretically and show that the simple ECOC method is minimax in the one-vs.-all case. Concluding remarks are given in Section 4.

2. Multi-class classification by the ECOC approach

2.1. Binary classifiers for multi-class classification

Let $\mathcal{X}$ denote the input space. We consider each input data $x \in \mathcal{X}$ has an output $y \in \{1, 2, \ldots, C\}$. Training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$ are available. Let $f_j : \mathcal{X} \to \{-1, +1\}$, $(j = 1, 2, \ldots, J)$ denote binary classifiers, each of which has two disjoint and non-empty subsets $I_j^+, I_j^- \subset \{1, 2, \ldots, C\}$ and classifies I_j^+ from I_j^-. Note that C and J are the numbers of classes and classifiers, respectively. The examples of I_j^+ and I_j^- are as follows:

- one-vs.-one

$$I_j^+ = \{l\}, \quad I_j^- = \{k\}, \ l = 1, \ldots, C-1, \ k = l+1, \ldots, C;$$

- one-vs.-all

$$I_j^+ = \{j\}, \quad I_j^- = \{1, \ldots, j-1, j+1, \ldots, C\}, \ j = 1, \ldots, C.$$

Each binary classifier f_j uses only training samples with labels in $I_j^+ \cup I_j^-$ for its learning. When training f_j, we give the new label "+1" to the samples whose original labels are in I_j^+, and "−1" to those in I_j^-. Since the problem reduces to binary classification, we can learn f_j by any good binary classifiers such as SVM or ada-boost.

We obtain a vector $\boldsymbol{f}(x) = (f_1(x), f_2(x), \ldots, f_J(x))$ for a new input data $x \in \mathcal{X}$, after learning the binary classifiers. The next thing we have to do is to draw a conclusion from the vector about the multi-class classification