Neurocomputing 74 (2011) 680–688

Contents lists available at ScienceDirect

Neurocomputing

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

# Statistical approaches to combining binary classifiers for multi-class classification

Yuichi Shiraishi [a,*], Kenji Fukumizu [b]

[a] Human Genome Center, Institute of Medical Science, The University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan
[b] The Institute of Statistical Mathematics, 10-3 Midori-cho, Tachikawa, Tokyo 190-8562, Japan

ARTICLE INFO

*Article history:*
Received 28 September 2009
Received in revised form
2 July 2010
Accepted 9 September 2010
Communicated by M.T. Manry
Available online 16 October 2010

*Keywords:*
Multi-class classification
Combining binary classifiers
Meta-learning
Stacking
Group lasso

ABSTRACT

One of the popular methods for multi-class classification is to combine binary classifiers. In this paper, we propose a new approach for combining binary classifiers. Our method trains a combining method of binary classifiers using statistical techniques such as penalized logistic regression, stacking, and a sparsity promoting penalty. Our approach has several advantages. Firstly, our method outperforms existing methods even if the base classifiers are well-tuned. Secondly, an estimate of conditional probability for each class can be naturally obtained. Furthermore, we propose selecting relevant binary classifiers by adding the group lasso type penalty in training the combining method.

## 1. Introduction

### 1.1. Background

The support vector machine (SVM) [8] has been one of the most popular classifiers. The SVM performs linear classification in the reproducing kernel Hilbert spaces (RKHS). We can freely choose a kernel (or RKHS) which captures important features of the data. Furthermore, the solution of the SVM can be obtained fast by novel methods such as the sequential minimal optimization (SMO) algorithm [28].

Although the SVM is very powerful for binary classification, it cannot be directly used for multi-class classification, where the number of class labels is greater than two. There are two approaches to deal with multi-class problems that maintain the advantage of the SVM (or the kernel). The first one is to consider loss functions treating more than two classes and minimize them directly with certain algorithms [5,37,9,25,43]. Since this approach is based on a multi-class loss function, its properties such as consistency to Bayes error rate are easier to analyze [42,34]. However, this 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 several SVMs for binary classification to derive a conclusion for the multi-class problem. Several empirical studies [20,32] show that this approach is computationally more feasible and never inferior to the former one. 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. The basic concept of this approach is shown in Fig. 1. Popular combiners for multi-class classification are the "one-vs-all method", the majority vote [17], the directed acyclic graph model [30], the Bradley–Terry model [19] and the error correcting output code (ECOC) model [14,1].

What is a good way to combine binary classifiers? Firstly, the whole classification system obtained via the combining method should have a low error rate. Many researchers have tried to propose combiners to achieve this aim. On the other hand, Rifkin and Klautau [32], using SVMs with Gaussian kernels for binary classifiers in their experiments, made the following assertions:

1. The most important step in multi-class classification is to use accurate binary classifiers.
2. Sophisticated methods of combination make little difference if the base classifiers are as strong as well-tuned SVMs.

Therefore, they recommended the "one-vs-all method" or the majority vote, which are very simple combining methods.

Furthermore, it is desirable that the combiner should generate probabilistic outputs. Probabilistic outputs of class labels are very important in many practical situations to incorporate uncertainty of prediction. Several studies have focused on providing a multi-class

[*] Corresponding author.
*E-mail addresses:* yshira@hgc.jp (Y. Shiraishi), fukumizu@ism.ac.jp (K. Fukumizu).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.09.004

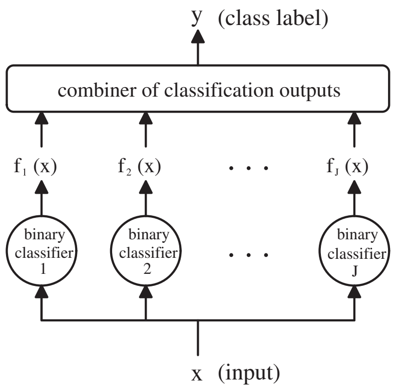

**Fig. 1.** The conceptual diagram of multi-class classifier via combining binary classifiers.

probability estimate from the outputs of binary classifiers [19,40,21]. However, most methods require that binary classifiers output probabilistic values, which is not the case for SVMs.

With the preceding discussion as background, we propose a new method for combining binary classifiers, adopting some statistical tools of logistic regression, stacking [38,6,24] and the group lasso [41,26]. The proposed method has several advantages described in the following subsection.

### 1.2. Advantages of the new method

#### 1.2.1. Accuracy

Contrary to [32], we show that our method outperforms simple combiners such as the “one-vs-all method” and the majority vote, especially in the one-vs-one case, even when the underlying binary classifiers are well-tuned. The key ingredient of this improvement is that, regarding the combiner of binary classifiers as a meta-level classifier, we train the combiner as well as the binary classifiers using the training samples, whereas most existing approaches such as the “one-vs-all method” use fixed, non-trained combiners. A problem in training the meta-level learner is that we have to use the same training data for both the base learners and the meta-level learner. To avoid this problem, we use stacking [38,6].

#### 1.2.2. Probabilistic outputs

A probabilistic estimate for each class can be obtained with our approach. Furthermore, concerning this aspect, our method has three advantages.

Firstly, according to our experiments, probabilistic estimates via our method are more reliable than those of previous methods. Estimated log-likelihoods of our method are almost consistently higher than those of the previous method, and much higher for some data sets.

Secondly, our approach does not require that each binary classifier return probabilistic values, unlike many existing methods [19,40,21]. Although some classifiers such as logistic regression return an estimated conditional probability of each class, many classifiers such as the SVM and the Adaboost do not. Transforming mere outputs of binary classifiers to probabilistic values is accompanied by ambiguity. Since our method can treat any type of output from the binary classifier, it will no longer be necessary to consider how to obtain probabilistic values.

Furthermore, our method is independent of frameworks for training binary classifiers such as “one-vs-one” or “one-vs-all”. The original method using the Bradley–Terry model [19] can treat only the one-vs-one case, although several researchers have extended the original method so that it can handle general cases such as the one-vs-all case [21].

#### 1.2.3. Sparsity

Our method can select classifiers relevant to the classification system as a whole by adding in the group lasso type penalty in training the combiner. Several researchers have tried to reduce the number of support vectors after training the SVM to save the computational cost [7,22]. Instead, we propose removing several dispensable binary classifiers. For this purpose, we adopt the group lasso type penalty.

The group lasso [41,26] is an extension of the lasso [35]. The lasso is a shrinkage method for variable selection. It selects relevant variables for regression by setting the coefficients of irrelevant variables to zero via a special kind of optimization. The group lasso is devised for group-wise variable selection. In this paper, we propose to apply the group lasso to prune unimportant binary classifiers. The experiments show that this successfully reduces the number of relevant binary classifiers.

#### 1.2.4. Remark

The binary classifier for multi-class classification does not need to be the SVM. We can use any good binary classifier such as the Adaboost or the neural networks. The methods proposed in this paper do not depend on the choice of binary classifiers. However, considering a number of studies of multi-class classification in machine learning is motivated to use the SVM for multi-class classification, and in order to facilitate a comparison with the results of Rifkin and Klautan [32], we focus on the SVM in the experiment.

### 1.3. Outline

The outline of our paper is as follows. In Section 2, we review multi-class classification by combining binary classifiers. In Section 3, we give detailed explanations for the generation of training data for the combiner via stacking. In Section 4, we propose training the combiner via penalized logistic regressions. In Section 5, we present some experiments. Concluding remarks are given in Section 6.

## 2. Multi-class classification by combining binary classifiers

### 2.1. Binary classifiers for multi-class classification

Let $\mathcal{X}$ denote the input space, and suppose each input data $x \in \mathcal{X}$ has an output $y \in \{1,2,\ldots,G\}$. Assume we have training data $T=\{(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)\}$. Let $f_j : \mathcal{X} \to \mathbb{R}$ $(j=1,2,\ldots,J)$ denote binary classifiers, each of which has two disjoint and nonempty subsets $I_j^+, I_j^- \subset \{1,2,\ldots,G\}$ and classifies $I_j^+$ from $I_j^-$. Note that $G$ and $J$ are the numbers of classes and classifiers, respectively. Examples of $I_j^+$ and $I_j^-$ are as follows:

- One-vs-one: $I_j^+ = \{l\}$, $I_j^- = \{k\}$ $(l=1,\ldots,G-1, k=l+1,\ldots,G)$.
- One-vs-all: $I_j^+ = \{j\}$, $I_j^- = \{1,\ldots,j-1,j+1,\ldots,G\}$ $(j=1,\ldots,G)$.

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 assign the new label “+1” to the samples whose original labels are in $I_j^+$, and “$-1$” to those in $I_j^-$. Since the problem is now reduced to binary classification, we can train $f_j$ using any good binary classifiers such as SVM.

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 training the binary classifiers. The next task is to draw a conclusion from the vector about the multi-class classification problem. When binary classifiers are trained using the one-vs-one scheme, the popular method is to use the majority vote, where the combiner returns the class who got the most *wins*. When the binary classifiers are trained by the one-vs-all scheme, the