Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition 38 (2005) 1483 – 1494
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Switching class labels to generate classification ensembles

Gonzalo Martínez-Muñoz*, Alberto Suárez

Escuela Politécnica Superior, Universidad Autónoma de Madrid, C/ Francisco Tomás y Valiente, 11, Madrid E-28049, Spain

Received 4 August 2004; accepted 27 February 2005

Abstract

Ensembles that combine the decisions of classifiers generated by using perturbed versions of the training set where the classes of the training examples are randomly switched can produce a significant error reduction, provided that large numbers of units and high class switching rates are used. The classifiers generated by this procedure have statistically uncorrelated errors in the training set. Hence, the ensembles they form exhibit a similar dependence of the training error on ensemble size, independently of the classification problem. In particular, for binary classification problems, the classification performance of the ensemble on the training data can be analysed in terms of a Bernoulli process. Experiments on several UCI datasets demonstrate the improvements in classification accuracy that can be obtained using these class-switching ensembles.

Keywords: Classification; Ensemble methods; Bagging; Boosting; Decision tree

1. Introduction

Classification methods based on pooling the decisions of an ensemble of classifiers have demonstrated great potential for improvement in many regression and classification problems [1–15]. To produce a reduction of the error rate, the classifiers generated must perform well on the proposed task and yet be sufficiently diverse. In order to achieve this diversity, ensemble algorithms introduce some systematic variation into the learning task, either by perturbing the training data or by taking advantage of instabilities in the learning algorithm.

One of the common procedures to generate classifier ensembles is bagging [3] (Bootstrap sampling and aggregation). In bagging, diversity is obtained by constructing each classifier in the ensemble with a different set of labelled examples, which is obtained from the original training set by re-sampling with replacement. Bagging then combines the decisions of the classifiers using unweighted voting. Bagging is believed to improve the performance of single classifiers mainly by reducing the variance error [4]. Breiman categorises bagging decision trees as a particular instance of *random forest* classification techniques [12]. A random forest is a tree-based ensemble that uses some kind of independent randomisation in the construction of every individual classifier. Many variants of bagging and random forests with excellent classification performance have been developed: In Ref. [9] trees in the ensemble are grown by randomly selecting among the best partitions at every tree node. In double-bagging [13] two classifiers are grown in each iteration by making use of the *out-of-bag* examples [16]. Attribute-bagging [14] selects a random subset of attributes at every iteration. IPG-ensembles [15] use different random partitions of the training data as input for the iterative growing and pruning tree construction algorithm of Gelfand et al. [17].

Another common algorithm for generating ensembles is boosting [1]. In boosting, the committee members are sequentially generated using weighted training data. Initially

* Corresponding author. Tel.: +34 91 497 3364; fax: +34 91 497 2235.
E-mail addresses: gonzalo.martinez@uam.es (G. Martínez-Muñoz), alberto.suarez@uam.es (A. Suárez).

0031-3203/$30.00 © 2005 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2005.02.020

all example weights are equal to 1. At each iteration of the boosting process these weights are updated according to the classification given by the last committee member generated: weights of incorrectly classified examples are increased and weights of correctly classified ones are decreased. In this way the base learner focuses on the harder examples. This entails a reduction both in bias and variance [10]. A weighted vote is used to make the final class assignment. Boosting has demonstrated to be one of the most effective methods for constructing ensembles [2,7,9].

In this article we present a variant of the output flipping ensembles proposed by Breiman in Ref. [18], that belongs to the category of random forests. In Breiman's work, each classifier in the ensemble is generated using the original training set with randomised class labels: The class label of each example is switched according to a probability that depends on an overall switching rate (defined as the proportion of training examples that are switched on average) and on the proportions of the different class labels in the original training set. The switching probabilities are chosen to maintain the class distribution of the original training set. Error rates similar or better than bagging are reported by using ensembles with 100 classifiers.

In this article we show that still lower error rates can be achieved with ensembles generated by class switching provided that we use fairly large ensembles ($\approx$ 1000 classifiers) and relatively high class switching rates. In contrast to Ref. [18], we do not require that the original class distribution be maintained in the perturbed training data. This makes it possible to use larger values of the switching rate in unbalanced datasets. Larger values of the switching rate are sometimes needed to get better classification accuracy.

The paper is organised as follows: Section 2 introduces the algorithm for generating the ensemble by switching the class labels of the training examples. Section 3 describes a simple experiment that is used to analyse in detail the classification strategy of the proposed ensemble. The classification performance of the class switching ensemble algorithm is compared to that of Breiman's flipping ensemble algorithm, bagging and boosting in 15 datasets. Some of these problems are synthetic and some are real-world (taken from the UCI repository [19]). Finally, the conclusions of this research are summarised.

2. Switching outputs

In Ref. [18], Breiman proposes to generate diverse classifiers by randomly switching the class labels of the training dataset according to the transition matrix

$$P_{j\leftarrow i} = wP_j \quad \text{for } i \neq j,$$
$$P_{i\leftarrow i} = 1 - w(1 - P_i), \tag{1}$$

where $P_{j\leftarrow i}$ is the probability that an example with label i gets the label j, P_i is the proportion of elements of class i in the training set, and w is proportional to the switching rate (average fraction of switched examples), p

$$w = \frac{p}{1 - \sum_j P_j^2} = \frac{p}{2\sum_j \sum_{k>j} P_j P_k}. \tag{2}$$

This form of the transition matrix, Eq. (1), is chosen to maintain the class proportions approximately constant.

In order for this method to work, the value of the switching rate p should be small enough to ensure that the training error tends to zero as the size of the ensemble grows. In a binary classification problem, the condition is

$$p < P_{min}, \tag{3}$$

where P_{min} is the proportion of examples that belong to the minority class. Inequality (3) ensures that, on average, the fraction of switched examples in the minority class is smaller than $\frac{1}{2}$. Switching rate values over this limit would flip the class label of more than half of the minority class examples. Hence, the minority feature space regions would be flooded with examples labelled as the majority class and consequently these regions would be classified incorrectly by the ensemble.

In this work we propose to generate ensembles of classifiers using different perturbed versions of the training set. In each perturbed set, a fixed fraction p of examples of the original training data is selected at random. The class label of each of these examples is randomly switched to a different one. This defines the following transition probability matrix:

$$P_{j\leftarrow i} = p/(K - 1) \quad \text{for } i \neq j,$$
$$P_{i\leftarrow i} = 1 - p, \tag{4}$$

where K is the number of classes. This label switching procedure produces training sets whose class distribution is usually different from that of the original training data. In fact, the class distribution of the perturbed set tends to equalise with increasing p for the unbalanced sets.

In order to guarantee the convergence of the ensemble in the training set there should be, for any given class, a majority of correctly labelled examples (i.e. not switched). This condition is fulfilled on the training set (on average) if $P_{j\leftarrow i} < P_{i\leftarrow i}$ and according to Eq. (4) we have

$$p < (K - 1)/K \tag{5}$$

independently of the initial class distribution. Following this equation we define the maximum value of p:

$$p_{max} = (K - 1)/K. \tag{6}$$

It is also convenient to define the ratio of the class switching probability to its maximum value

$$\hat{p} = p/p_{max}. \tag{7}$$

Thus, for unbalanced datasets, the proposed method increases the range of allowed values of p, with respect to the