Pattern Recognition 45 (2012) 4451–4465

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

A noise-detection based AdaBoost algorithm for mislabeled data

Jingjing Cao, Sam Kwong*, Ran Wang

Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 14 September 2011
Received in revised form
16 March 2012
Accepted 5 May 2012
Available online 16 May 2012

Keywords:
Pattern recognition
Ensemble learning
AdaBoost
k-NN
EM

ABSTRACT

Noise sensitivity is known as a key related issue of AdaBoost algorithm. Previous works exhibit that AdaBoost is prone to be overfitting in dealing with the noisy data sets due to its consistent high weights assignment on hard-to-learn instances (mislabeled instances or outliers). In this paper, a new boosting approach, named noise-detection based AdaBoost (ND-AdaBoost), is exploited to combine classifiers by emphasizing on training misclassified noisy instances and correctly classified non-noisy instances. Specifically, the algorithm is designed by integrating a noise-detection based loss function into AdaBoost to adjust the weight distribution at each iteration. A *k*-nearest-neighbor (*k*-NN) and an expectation maximization (EM) based evaluation criteria are both constructed to detect noisy instances. Further, a regeneration condition is presented and analyzed to control the ensemble training error bound of the proposed algorithm which provides theoretical support. Finally, we conduct some experiments on selected binary UCI benchmark data sets and demonstrate that the proposed algorithm is more robust than standard and other types of AdaBoost for noisy data sets.

1. Introduction

AdaBoost [1–3] is one of the most popular techniques for generating ensembles due to its adaptability and simplicity. In the past few decades, AdaBoost has been successfully extended to many fields such as cost-sensitive classification [4,5], semi-supervised learning [6], tracking [7] and network intrusion detection [8]. The main idea of AdaBoost is to construct a succession of weak learners by using different training sets that are derived from resampling the original data. Through a weighted vote, these learners are combined to predict the class label of a new test instance. Normally, the performance of a weak learner is slightly better than random guessing [9]. The weak learner that used in the ensemble is named as base classifier or component classifier.

However, AdaBoost tends to be overfitting when the number of combined classifiers increases. Some researchers attributed this failure of AdaBoost to the high proportion of noisy instances [10,11]. In [10], Rätsch et al. defined three conditions to identify noisy data: (1) overlapping class probability distributions, (2) outliers and (3) mislabeled instances. It should be noted that our work only discusses noisy instances with mislabeled property. Mislabeled instances typically refer to those instances inconsistent with most of their surrounding neighbors' class labels. Dietterich [11] designed an experimental test to prove the poor generalization of AdaBoost with C4.5 by adding artificial noise. He explained that the mislabeled instances would possibly be assigned to higher weights, which gave rise to unsatisfactory performance of AdaBoost.

By analyzing the inner impelling force of AdaBoost, one may notice that essentially it aims to minimize an exponential loss function [12] sequentially. In detail, it puts emphasis on penalizing misclassified instances by giving incremental weights whereas assigning lessened weights to correctly classified instances for the next iteration. In this way, AdaBoost will only focus on punishing the misclassified instances whereas ignore their mislabeled property, which leads to the noise sensitivity of AdaBoost.

Therefore, in this paper, a noise-detection based AdaBoost algorithm (ND-AdaBoost), associated with the mislabeled properties of instances, is proposed to address the noise sensitivity and overfitting problem. The main contributions of this paper are as follows.

(1) Four types of instances with respect to noise and class label decisions, which are different from conventional concern on taxonomy of misclassified and correctly classified instances, are introduced. More specifically, they are correctly classified noisy instances, misclassified noisy instances, misclassified non-noisy instances and correctly classified non-noisy instances. This division is in line with the assumption that the probability of mislabeled instances being misclassified should be as high as possible, while the correctly labeled instances are expected to be classified correctly.

* Corresponding author. Tel.: +852 34427704; fax: +852 34420503.
E-mail address: cssamk@cityu.edu.hk (S. Kwong).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.05.002

(2) A revised exponential loss function is proposed by considering these types of instances. At each iteration, a noise label determined by a noise detection function is assigned to each instance. With the new loss function, we aim to minimize the optimization objective by assigning less weights to misclassified noisy instances and correctly classified non-noisy instances. To identify noisy data, both EM and k-NN based functions are employed to test noise labeling effects under different detection methods.

(3) In order to guarantee the generalization ability of the proposed method, a new regeneration condition based on the analysis of empirical margin error bound of ND-AdaBoost is developed, so as to control the bound of the proposed algorithm within a reasonable range.

The performance of noise-detection based AdaBoost algorithm is examined through experiments on 13 binary data sets from UCI repository [13]. Experimental results show that the proposed algorithm outperforms other boosting methods under noisy environment.

2. Related work

For decades, researchers have made different modifications on AdaBoost technique to handle its noisy detrimental effect through two directions: (1) revising the optimization objective (loss function) and rebuilding the weight updating mechanism according to the corresponding loss function; (2) limiting the incremental weight update of the noisy instance or discarding them directly. Through these methods, the disturbance originated from mistrust instances could be minimized, and the noise tolerance of the ensemble model could be improved.

Regarding the first direction, many techniques have been employed. Depending on the statistical interpretation of AdaBoost, LogitBoost [14] utilized the additive logistic regression model function to replace the original loss function. However, it often suffers numerical problems caused by computing the regression variable. A generalized version of traditional AdaBoost is called Real AdaBoost [14,15]. It calculates the class probability to construct better real-valued output of weak learners. In essence, Real AdaBoost employs a log-odds ratio to replace the exponential loss function. With this modification, Real AdaBoost can converge quicker than AdaBoost but is also sensitive to outliers and mislabeled data [16]. Hastie and Tibshirani [14] presented an improved version of Real AdaBoost by utilizing adaptive Newton steps, which was similar to LogitBoost algorithm, to minimize the loss function. The empirical evidence implies that Gentle AdaBoost outperforms Real AdaBoost in terms of noisy data but has similar performance on regular data. Since weak learner may bias on training the data that have been correctly classified with high margin, Modest AdaBoost [17] was proposed to revise the loss function by focusing on decreasing this impact of base classifier [18]. This modification improves the generalization capability and relieves the overfitting problem of AdaBoost to some extent. MadaBoost was proposed by [19] with the aim to modify the reweighting scheme of AdaBoost. In the literature [19], Carlos et al. proved that one version of MadaBoost kept an adaptive boosting property. However, in the framework of MadaBoost, the corresponding advantages γ_i of the weak hypotheses ($\gamma_i \stackrel{\text{def}}{=} 1/2-\epsilon_i$, ϵ_i denote the errors at iteration i) are monotonically decreasing and its boosting speed is slower than AdaBoost [16]. Rätsch et al. [10] claimed that AdaBoost intended to overfit on data set with high noise level. Inspired by the margin theory of SVM, a revised version of AdaBoost was introduced by embedding soft-margin into the algorithm AdaBoost$_{Reg}$. In this case, the noise effect could be mitigated by controlling the influence of an instance on ensemble classifiers. In summary, by utilizing various loss functions, these methods can perform well on noisy data.

However, Gao et al. [20] pointed out that the modified loss functions were fixed and were independent of the noisy or noisy-free property of input instances. Thus, they took into account the filter procedure to filter the noisy instances. Likewise, as for the second trend, Nikunj [21] proposed a revised Aveboost2 by averaging the current and the previous distributions to generate new base classifiers' distribution. Nicolas [22,23] employed the distribution generated by boosting to build a supervised projection of the original data to train the next classifier. However, Nikunj [21] and Nicolas [22,23] only emphasized on reducing the weights of the misclassified patterns while without considering the noise issue. Gao et al. [24] also proposed a weighted k-NN algorithm to identify and removed some suspect instances. But editing suspect instances will shrink the size of the training set. Additionally, it is probable that the mislabeled instances are maintained while the correctly labeled instances are removed.

Based on the above-mentioned discussions, we propose a noise-detection based method by modifying the loss function. It integrates the advantages of these two research directions while restricts the weight of instance simultaneously.

Furthermore, when dealing with noisy data, another major problem is how to detect noisy instances. It is found that most of the previous works are related to data pruning, but we are in favor of utilizing instance weighting approach instead of editing. For example, Rätsch et al. [10] described that mistrust instances were those which are highly influential to the decision. They used the average weight of an instance based upon the assumption that a difficultly classified instance probably has high average weight. Rebbapragada et al. [25] proposed a method named pairwise expectation maximization (PWEM) to produce instance weight. They conducted some experiments to show that instance weighting performed better than instance editing.

Different from selecting subset of instances to generate new weight distribution, another groups of Boosting algorithms emphasize on choosing subset of features to construct ensemble of classifiers. Random subspace method (RSM) [26] is a standard method to randomly choose a subspace from the original feature space, which could be composed from any base classifier. In [27], Satoshi et al. empirically showed that combining RSM and AdaBoost algorithm had less generalization error than using RSM and AdaBoost, respectively. Additionally, Nicolas et al. [28] improved the performance of the RSM based AdaBoost by concerning the discriminate information among subspaces. Recently, Nanni et al. [29] designed a Reduced Reward-punishment editing strategy to construct different subspaces used in feature transform based ensemble approaches. In our experimental studies, we also compare our approach with the RSM+AdaBoost method since it is more robust than AdaBoost in the presence of noise [28].

The rest of this paper is arranged as follows: the related knowledge about AdaBoost is introduced in Section 3; the motivation and the procedure of the proposed algorithm are provided in Section 4 which are associated with some related analysis; the experimental comparisons on 13 real world binary data sets are presented and analyzed in Section 5; discussions and conclusions of the paper are finally given in Section 6.

3. Framework of AdaBoost algorithm

Since our work is an extension of AdaBoost approach, preliminary knowledge of AdaBoost is firstly introduced in this section. Suppose we have a two class supervised learning task. Let the n-th training instance denoted as $z_n=(x_n,y_n), n=1,\ldots,N$,