Knowledge-Based Systems 102 (2016) 87–102

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A robust multi-class AdaBoost algorithm for mislabeled noisy data

Bo Sun*, Songcan Chen, Jiandong Wang, Haiyan Chen

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, 29 Yudao Street, Nanjing 210016, China

ARTICLE INFO

Article history:
Received 12 June 2015
Revised 23 March 2016
Accepted 25 March 2016
Available online 30 March 2016

Keywords:
Ensemble learning
AdaBoost
Robustness
Multi-class classification
Mislabeled noise

ABSTRACT

AdaBoost has been theoretically and empirically proved to be a very successful ensemble learning algorithm, which iteratively generates a set of diverse weak learners and combines their outputs using the weighted majority voting rule as the final decision. However, in some cases, AdaBoost leads to overfitting especially for mislabeled noisy training examples, resulting in both its degraded generalization performance and non-robustness. Recently, a representative approach named noise-detection based AdaBoost (ND_AdaBoost) has been proposed to improve the robustness of AdaBoost in the two-class classification scenario, however, in the multi-class scenario, this approach can hardly achieve satisfactory performance due to the following three reasons. (1) If we decompose a multi-class classification problem using such strategies as one-versus-all or one-versus-one, the obtained two-class problems usually have imbalanced training sets, which negatively influences the performance of ND_AdaBoost. (2) If we directly apply ND_AdaBoost to the multi-class classification scenario, its two-class loss function is no longer applicable and its accuracy requirement for the (weak) base classifiers, i.e., greater than 0.5, is too strong to be almost satisfied. (3) ND_AdaBoost still has the tendency of overfitting as it increases the weights of correctly classified noisy examples, which could make it focus on learning these noisy examples in the subsequent iterations. To solve the dilemma, in this paper, we propose a robust multi-class AdaBoost algorithm (Rob_MulAda) whose key ingredients consist in a noise-detection based multi-class loss function and a new weight updating scheme. Experimental study indicates that our newly-proposed weight updating scheme is indeed more robust to mislabeled noises than that of ND_AdaBoost in both two-class and multi-class scenarios. In addition, through the comparison experiments, we also verify the effectiveness of Rob_MulAda and provide a suggestion in choosing the most appropriate noise-alleviating approach according to the concrete noise level in practical applications.

1. Introduction

How to effectively boost the generalization performance of a learner has been an important research topic in machine learning community. Ensemble learning [1–5] is a powerful paradigm for achieving such a goal and has been widely studied and successfully applied in many areas since its invention, e.g., it has been applied in face recognition[6], class imbalance learning [7], concept drift handling [8,9], etc. Many successful ensemble learning algorithms have constantly been proposed in the literature [4,10–17], where Bagging [5,10,12] and AdaBoost [13–15] are most famous. Bagging is a parallel ensemble learning algorithm, which first samples the original training example set $D_{tr} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ ($x_i \in \mathbb{R}^d$, $y_i \in Y$, $d > 1$, Y is the class label set) with replacement to construct multiple diverse bootstrap training subsets $D_{tr\_1}, D_{tr\_2}, \ldots, D_{tr\_T}$, then trains one classifier C_t on each sampled subset D_{tr_t} ($t = 1, 2, \ldots, T$), finally combines these generated classifiers using the majority voting rule to get the final decision. Different from Bagging, AdaBoost is a sequential ensemble learning algorithm and constructs a final ensemble in the following manner: initially, it trains the first classifier C_1 on the original training set D_{tr} whose components have a uniform weight distribution, i.e., $w_i = 1/N$ ($i = 1, 2, \ldots, N$), then the training examples misclassified by classifier C_1 are assigned larger weights, leading to a greater possibility of being selected as components in the training set of the next classifier. That is, the training set of the $(t+1)$th ($1 \le t \le T-1$, T is the ensemble size) classifier g_{t+1} is determined by the classification results of the tth classifier g_t. In this way, AdaBoost focuses more on learning the 'hard' (misclassified) examples and trains classifiers in an iterative way, finally combines the sequentially-generated classifiers using the weighted majority voting rule to obtain the final decision.

In many practical applications, however, the collected training set D_{tr} often contains some 'noisy' examples, e.g., examples

* Corresponding authors. Tel.: +86 158 505 582 15; fax: +86 025 84892956.
E-mail addresses: sunbo87@126.com (B. Sun), s.chen@nuaa.edu.cn (S. Chen), aics@nuaa.edu.cn (J. Wang), chenhaiyan@nuaa.edu.cn (H. Chen).

http://dx.doi.org/10.1016/j.knosys.2016.03.024

with class label mislabeled are called *label noises*, or examples with incorrect feature values are called *feature noises*. It has been shown that label noises are generally more detrimental than feature noises to the generalization performance of learned classifiers [18,19]. Thus, in this paper, we mainly investigate the label noises and the term 'noise' that appears in the remaining of this paper refers to the mislabeled noisy training examples. Existing studies have demonstrated that Bagging is insensitive and robust to mislabeled noisy examples while AdaBoost is very sensitive and non-robust to such noises [5,12]. The reason of the poor performance of AdaBoost in the noisy training scenario is probably due to that, compared with the 'clean' (correctly labeled) training examples, the mislabeled noisy ones are more easily misclassified by the current classifier and thus improperly assigned larger weights by the weight updating scheme of AdaBoost, which would make AdaBoost focus more on learning these mislabeled noises and thus overfit them in the subsequent iterations. Therefore, in the mislabeled noisy training scenario, how to effectively improve the robustness of AdaBoost is an important research issue.

Since AdaBoost is originally designed as a two-class classification algorithm, naturally most existing works focus on improving its robustness in the two-class classification problems. A recent work [26] proposed a representative approach named noise-detection based AdaBoost (ND_AdaBoost) to improve the robustness of AdaBoost. However, ND_AdaBoost is hard to be applied in the multi-class classification scenario due to the following three reasons. (1) If we decompose a multi-class classification problem into multiple two-class problems using such strategies as one-versus-all [19] or one-versus-one [20] and then directly apply ND_AdaBoost, the obtained two-class problems usually have imbalanced training sets, which would negatively influence the generalization performance of ND_AdaBoost. (2) If we directly apply ND_AdaBoost to the multi-class classification problems, its two-class loss function is no longer applicable and its accuracy requirement for the base classifiers, i.e., greater than 0.5, is usually too strong to be met. (3) ND_AdaBoost still has the tendency of overfitting as it increases the weights of correctly classified noisy examples in each iteration, which could make it focus on learning these noisy examples in the subsequent iterations.

To solve the above dilemma, we propose a robust multi-class AdaBoost algorithm (Rob_MulAda) based on the popular multi-class AdaBoost algorithm SAMME [21]. In Rob_MulAda, (1) we formally design a noise-detection based multi-class loss function and solve its minimization problem by proving a proposition such that the optimal weight α_T of base classifier g_T ($T \geq 1$) is obtained; (2) we present a new weight updating scheme to alleviate the harmful effect of noisy examples. In the experimental study, we first directly compare ND_AdaBoost with its modified version ND_AdaBoost* that is specifically designed by just substituting the weight updating scheme of ND_AdaBoost with ours to verify the robustness of our weight updating scheme in the two-class classification scenario; Then, we verify the effectiveness of the proposed Rob_MulAda by empirically comparing it with several closely-related algorithms in the multi-class scenario. Experimental results demonstrate that our weight updating scheme is indeed more robust to mislabeled noises than the one used in ND_AdaBoost in both two-class and multi-class classification scenarios, and besides, the proposed algorithm Rob_MulAda is most effective when the mislabeled noise level is low and moderate (no more than 20%). Based on the experimental results, we also provide a suggestion in choosing the most appropriate noise-alleviating approach according to the concrete noise level in practical applications.

The rest of this paper is organized as follows. In Section 2, we briefly review the approaches that have been proposed to enhance AdaBoost's robustness in the two-class classification scenario. In Section 3, we introduce the noise-detection based AdaBoost algorithm ND_AdaBoost for two-class classification problems and Zhu et al.'s multi-class AdaBoost algorithm SAMME to facilitate the later descriptions. In Section 4, we illustrate our robust multi-class AdaBoost algorithm Rob_MulAda in detail. In Section 5 we first empirically verify the robustness of our weight updating scheme in two-class classification problems, and then verify the effectiveness of the proposed Rob_MulAda in multi-class problems. Conclusion and future work are given in Section 6.

2. Related work

Many works have been conducted in the literature to improve the robustness of the classical two-class AdaBoost algorithm in the mislabeled noisy training scenario.

(1) Identifying and removing the noisy training examples is the simplest category of approaches, which first detects the noisy examples in the original training set D_{tr} using a certain method, and then directly removes the detected noisy examples to obtain a refined training set D'_{tr}, after that, AdaBoost algorithm is directly applied on the refined training set. Different methods of detecting noisy examples differentiate concrete approaches in this category.

  Muhlenbach et al., [18] proposed an approach that first detects noisy examples based on the class label comparison between each example (x_i, y_i) $(i = 1, 2, \ldots, N)$ and its k (e.g., $k = 5$) nearest neighbors $\{(x_{ij}, y_{ij})\}_{j=1}^{k}$ in the training set, and then removes all the detected noisy examples. Concretely, example (x_i, y_i) is considered to be a noise if its class label y_i is inconsistent with the most prevalent label y among its k nearest neighbors. Their experiments demonstrated that, when introduced 0 to 20% of artificial mislabeled noisy examples into the training set D_{tr}, the approach can improve the generalization performance of learned classifiers to some extent and yields better performance than the approach of relabeling the detected noises. However, this approach shrinks the size of training set D_{tr} and in some cases degrades the generalization performance of the learned classifiers especially when the original size of D_{tr} is small. Angelova et al., [22] proposed an approach which first trains multiple diverse classifiers $\{C_t\}_{t=1}^{T}$ on the original training set D_{tr} using a technique like bootstrap sampling, then detects noisy examples according to the decisions of all the generated classifiers. Concretely, if example (x_i, y_i) $(1 \leq i \leq N)$ causes a large disagreement among the decisions of these classifiers: $C_1(x_i), \ldots, C_T(x_i)$, e.g., nearly a half of these classifiers classify (x_i, y_i) as a positive example while the remaining classifiers classify it as a negative one, it is considered to be a noise and removed from the training set. In this way, a refined training set D'_{tr} is obtained and the quality of the original training set is accordingly improved. When the original training set contains a relative high proportion of noises, the generated classifiers $\{C_t\}_{t=1}^{T}$ may be not accurate enough, in this case, some misdetected noisy examples (in fact, correctly labeled examples) will be removed, leading to a degraded generalization performance of the classifiers learned on the refined training set.

(2) Another category of approaches tries to improve the robustness of AdaBoost by modifying its loss function or weight updating scheme.

To enhance the noise tolerance of AdaBoost, Domingo et al., [23] proposed a modification of AdaBoost in which the weights of training examples are bounded by their initial values. Servedio [24] designed a boosting algorithm that generates smooth weight distributions on the training set and prevents from assigning too large weight to any single training example. Although the two approaches both try to reduce the harmful effect of noisy examples