Neurocomputing 119 (2013) 462–468

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

Semi-supervised multi-label image classification based on nearest neighbor editing

Zhihua Wei [a,b,c], Hanli Wang [a,b,*], Rui Zhao [a,d]

[a] Department of Computer Science and Technology, Tongji University, Shanghai 201804, China
[b] Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 200092, China
[c] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China
[d] The Third Research Institute of the Ministry of Public Security, Shanghai 201204, China

ARTICLE INFO

Article history:
Received 29 June 2012
Received in revised form
22 January 2013
Accepted 21 March 2013
Communicated by: Weifeng Liu
Available online 21 April 2013

Keywords:
Semi-supervised learning
Multi-label classification
Data editing
Image classification

ABSTRACT

Semi-supervised multi-label classification has been applied to many real-world applications such as image classification, document classification and so on. In semi-supervised learning, unlabeled samples are added to the training set for enhancing the classification performance, however, noises are introduced simultaneously. In order to reduce this negative effect, the nearest neighbor data editing technique is introduced to semi-supervised multi-label classification, and thus an algorithm named Multi-Label Self-Training with Editing (MLSTE) is proposed in this work. The proposed algorithm is able to solve the uncertainty problem in semi-supervised multi-label classification to some extent, by improving the performance of determining the label number and selecting confident samples during the course of semi-supervised learning. Extensive experimental results on several benchmark datasets have been carried out to verify the effectiveness of the proposed MLSTE algorithm.

1. Introduction

Multi-label learning is applied to the classification problem in which each example can be assigned to more than one of the multiple class labels simultaneously. It has attracted quite a number of applications, such as natural language processing, computer vision, bioinformatics, health care, physiology, etc. However, it is not trivial to obtain labeled samples for the learning purpose. To address this problem, semi-supervised learning has been developed to exploit a large number of unlabeled samples for augmenting the training set which often has only a limited number of manually labeled samples, and thus the classification performance can be improved. Recently, the semi-supervised learning has gained a significant interest in many practical applications in the field of pattern recognition and machine learning. Generally speaking, those semi-supervised learning methods could be categorized into five paradigms, including generative parametric models [1], self-training approaches [2], co-training approaches [3,4], graph-based models [5] and Semi-Supervised Support Vector Machines (S3SVM) [6]. A detailed literature survey could be found in [7].

One of the most important problems of semi-supervised learning resides in the noise brought by unlabeled data. That is, once false-labeled instances are accepted by the training set, they will serve as correct instances to impair the classification quality. In order to minimize the downside of noisy instances in semi-supervised learning, it is desired to remove the noisy instances, which can be achieved by the data editing technique [8]. As one of the most significant representative methods for noisy data removal, data editing eliminates erroneous instances from the training set mainly with the following two kinds of semi-supervised learning approaches, including self-training [2] and co-training [3].

Self-training is a commonly used technique for semi-supervised learning. In self-training, a classifier is firstly trained with a small number of labeled training samples, and then the most confident unlabeled samples together with their predicted labels are added to the training set. The classifier is re-trained and the process is repeated until a predefined stopping criterion is met. However, for a typical self-training approach, the learner converts the most confidently labeled testing samples of each class into training examples without error detection and label editing, which will degrade the training efficiency, since it is highly probable to misclassify a certain number of unlabeled samples. As a consequence, the augmented training set may cause an error reinforcement problem [9] and degrade the classification performance. To address this problem, the data editing methods, such as the SETRED (Self-Training with Editing) algorithm [10] has been applied to

[*] Corresponding author at: Department of Computer Science and Technology, Tongji University, No. 4800, Cao'an Road, Shanghai 201804, China.
Tel.: +86 21 69585020; fax: +86 21 69589359.
E-mail addresses: zhihua_wei@tongji.edu.cn, zhihua.wei@hotmail.com (Z. Wei), hanliwang@tongji.edu.cn (H. Wang), zhaorui@mail.trimps.ac.cn (R. Zhao).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.03.011

identify and remove mislabeled samples in each iteration of the self-training process, i.e., it estimates whether a newly labeled sample is reliable or not, and only the confident samples are used to augment the training set. In the SETRED algorithm, the statistical method CEW (Cut Edge Weight) [11] is employed, which is able to minimize the global cut edge weight while modifying a minimum number of labels, as it operates on examples that contribute the most to increase the cut edge weight. In order to further improve the CEW performance, the nearest neighbor rule is applied with CEW in [12], which performs well because it reduces the error reinforcement by using CEW on relative neighborhood graph for classification.

Co-training is another widely used semi-supervised learning technique which requires two views of the data [7]. It assumes that each example could be described with two different feature sets that provide different and complementary information about the instance. In co-training, two separate classifiers are trained with the labeled samples on the two sub-feature views, and the resultant information is shared between these two classifiers. Then, each of these two classifiers is retrained by using the newly added training samples given by the other classifier. Co-training makes strong assumptions on the independence between the two feature views, and it is controversial whether the assumption can be relaxed [1,13]. Some variants of the standard co-training method have been proposed to measure the labeling confidence on unlabeled examples implicitly, e.g., by simply using the classifier's posteriori probability outputs [14], by repeatedly performing cross-validation on the original labeled examples [3,15], or by additionally employing a third classifier [4]. Zhang and Zhou applied the data editing technique into co-training and proposed the COTRADE (Confident Co-Training with Data Editing) algorithm, which is a direct method to facilitate reliable labeling information exchange between different views [16].

Regarding the data editing techniques, traditional approaches such as [8,17–19] only employ the predicted labeled data for training set augmentation without considering the unlabeled samples which may be misclassified as negative training examples. For instance, the Edited Nearest Neighbor (ENN) [8] method proposed by Wilson removes all instances which may be misclassified by the kNN rule from the training set. The Repeated Edited Nearest Neighbor (RENN) and All k-NN (ANN) proposed in [17] further improve the performance by employing ENN in a repeated manner. Another kind of improvement on Wilson's ENN is performed by changing error estimate method such as Holdout editing [18], Multi-edit [18] and editing by estimating conditional class probabilities [19]. In addition, the sensitivity problem about the sample size for ENN is considered in [20]. To further improve the classification performance, both the labeled and unlabeled data are considered during the editing process in [21] to boost ENN. Although it is a great improvement by extending traditional data editing method to semi-supervised learning in [21], only the single-label classification problem is considered.

As compared to single-label classification problems, multi-label problems are fairly new which are attracting researchers' attentions in recent years [22–26]. The most different feature of multi-label classification is that one sample may be related to multiple class labels making the classification problem more complicated. As mentioned above, the data editing techniques for single-label classification problems search for noisy samples according to opposite instances. However, multi-label problems bring challenges to data editing because there is more than one kind of opposites and the strategy of removing noisy samples should be considered as a whole. To the best of our knowledge, there is little research work about semi-supervised multi-label classification with data editing technique. Therefore, in this work, a self-training semi-supervised algorithm based on improved editing nearest neighbor method is proposed for multi-label classification tasks. The proposed algorithm utilizes each of the training samples (including labeled and unlabeled) and reserves more than one label when the editing procedure is conducted, which can be realized by editing labels based on the distribution of class labels in a multi-label dataset. Extensive experimental results on UCI [27] and real image datasets [23] demonstrate the efficiency of the proposed algorithm.

The rest of the paper is organized as follows. Section 2 introduces the proposed Multi-Label Self-Training with Editing (MLSTE) algorithm. Experimental results and related discussions are given in Section 3. Section 4 concludes this paper and explores future research directions.

2. Proposed multi-label self-training with editing algorithm

2.1. Extension of classic ENN algorithm

The classic ENN algorithm proposed by Penrod and Wagner [9] only edits samples in the training set and the accuracy of the classifier may converge to the Bayes error as the number of instances approaches infinity. To address this issue, we propose to extend the searching scope of neighbors from the labeled samples to both labeled and unlabeled ones while trying to eliminate the erroneous labeled samples by ENN. Let L and U denote the labeled and unlabeled set drawn from a dataset $D(\boldsymbol{X}, \boldsymbol{Y})$, where $\boldsymbol{X}$ is the set of samples represented by a feature vector and $\boldsymbol{Y}$ is a label set. Generally, it is necessary to keep the same class distribution in L and U [10]. Whereas, for multi-label classification, it is hard to keep completely the same because there may be more than one class label related to a sample. In standard self-training processes, a learner selects a set of samples with highest confidence, denoted as L^+, to replenish the training set. The set $L \cup L^+$ are used as the new training set for further self-training processes. The process is shown in Algorithm 1 (Fig. 1).

In detail, for semi-supervised classification problems, Algorithm 1 initiates the self-training process by firstly learning a classifier from the training set. In the subsequent self-training iterations, the base classifier identifies the unlabeled samples which can achieve highly confident prediction and derive the labels for these samples according to the prediction results. Then, for each label y_i, the first n confident samples with their predicted labels are added to L^+.

When L^+ is formed, the nearest neighbor editing technique is performed on the augmented training set $L \cup L^+$. Firstly, a nearest neighbor list based on the Euclidean distance is constructed from the samples of the set $L \cup L^+$, which describes the proximity between samples in the feature space. In the next step, the proposed MLSTE algorithm evaluates the confidence of whether the label set Y_i associated with the sample x_i is accurate through exploring the label distribution of their k nearest neighbors. The basic assumption is that a correctly labeled sample should belong to the same class as most of its neighbors. It is notable that there may be more than one label selected from the neighbors for multi-label classification problem. For example, k is set to 10 and in the neighbor set of x_i which is selected by Algorithm 1, there are five samples associated with the label y_1, four samples associated with the label y_2 and one sample associated with the label y_3. According to the MLSTE algorithm, both y_1 and y_2 should be attached to the sample x_i, and x_i should possess the label set $Y_i = \{y_1, y_2\}$. The illustration is described in Fig. 2.

The detailed description of the proposed algorithm is shown in the next two subsections. Firstly, the label set for each unlabeled sample is predicted. According to the confident level of the prediction, a number of unlabeled samples are selected to augment the