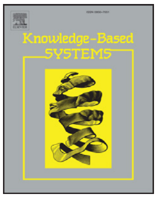
Knowledge-Based Systems 000 (2017) 1–16

Contents lists available at ScienceDirect

Knowledge-Based Systems

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



Analysis of training data using clustering to improve semi-supervised self-training

N. Piroonsup*, S. Sinthupinyo

Department of Computer Engineering, Chulalongkorn University, 254 Phayathai Road, Pathumwan, Bangkok 10330, Thailand

ARTICLE INFO

Article history:
Received 27 June 2017
Revised 2 December 2017
Accepted 7 December 2017
Available online xxx

Keywords:
Semi-supervised classification
Self-training
Cluster analysis
Semi-supervised clustering
Active learning

ABSTRACT

Applying unlabeled data in semi-supervised self-training can significantly improve the accuracy of a supervised classifier, but in some cases, it may dramatically decrease the classification accuracy. One reason for such degradation is a lack of labeled data for training an initial classifier in the self-training process. In this paper, we propose a method to determine the sufficiency of the labeled data and two methods to improve the labeled dataset in the insufficient portion. To determine the sufficiency of labeled data, we apply a semi-supervised cluster technique to estimate the labeled data distribution over the training set. The results show that the accuracy obtained from the final classifiers in clusters without labeled data is markedly lower than that obtained from clusters with labeled data. The two methods we propose for improving the labeled dataset are active labeling and co-labeling, for ensuring the sufficiency of labeled data. Comparison experiments on UCI and real-world datasets show that the proposed methods are an effective preprocessing step for determining and obtaining a sufficient quantity of labeled data, which is essential for attaining accuracy in a semi-supervised self-training classifier.

1. Introduction

Semi-supervised learning is a machine learning technique that applies unlabeled data along with labeled data to train a system. Semi-supervised learning is an approach that lies between supervised learning, which is learning with labeled data only, and unsupervised learning, which is learning with unlabeled data only [1]. A labeled dataset consists of attribute values and their corresponding class labels, whereas unlabeled data consist of only the attribute values. Labeled data are rather scarce because class labeling is a costly and time-consuming process. Although large quantities of unlabeled data are readily available, we cannot use them in standard supervised learning techniques. Hence, the lack of labeled data is a major difficulty in developing a competent learner, particularly in classification tasks.

Semi-supervised learning takes advantage of the vast quantity of unlabeled data available along with related labeled data information to train an efficient classifier. There have been many studies on semi-supervised learning approaches, which include self-training [2], co-training [3], generative models [4], graph-based methods [5], and semi-supervised support vector machines [6]. One of the most simple, efficient, and widely used approaches is a self-training approach.

The process of self-training is simple and straightforward. First, the labeled data are used to train an initial classifier. Then, the initial classifier assigns the predicted class labels to the unlabeled data having the highest confidence values. The newly labeled data are then included and used to update the initial classifier. The process repeats until a stopping criterion is met, such as that all unlabeled data have been assigned a class label. Finally, the originally labeled and the newly labeled data are combined and used to train the final classifier. The advantage of self-training is that it does not require a specific assumption, and as a result, it can be used in almost any situation. The self-training approach is widely used in many domains, including object detection [7], face recognition [8], sentence parsing [9], EEG classification [10,11], and time series problems [12].

Although the use of unlabeled data is beneficial, it can also degrade the classifier's performance. In recent years, questions regarding the usefulness of unlabeled data have been investigated in the area of semi-supervised learning [4,13–16]. However, the study of semi-supervised self-training has not received much attention. In self-training, unlabeled data can degrade the performance of a classifier if they are incorrectly labeled as an improper class by the initial classifier. Most studies have tried to avoid such mislabeling of data by post-processing, e.g., applying a noise-filtering method to remove the mislabeled data [17], self-training with editing [18],

* Corresponding author.
E-mail addresses: nareeporn.p@chula.ac.th, naree.p@gmail.com (N. Piroonsup), sukree.s@chula.ac.th (S. Sinthupinyo).

https://doi.org/10.1016/j.knosys.2017.12.006
0950-7051/© 2017 Elsevier B.V. All rights reserved.

or self-training with a nearest neighbor rule using cut edges [19]. However, the post-processing may edit the class of correctly labeled data into incorrectly labeled data or it may filter out correct data, thereby letting incorrect data be feed into the training process for the final classifier.

In this paper, we propose preprocessing methods for semi-supervised self-training. The first proposed method is an approach to analyze the sufficiency of labeled data for training the efficient initial classifier in self-training. We investigate the sufficiency of labeled data by considering the distribution of labeled data over the training dataset. The distribution of data is approximated by a semi-supervised clustering method that divides the data into two categories: a labeled cluster that contains a number of labeled data, and an unknown cluster that contains no labeled data. Our analysis showed that the accuracy of the semi-supervised classifier of data belonging to the unknown clusters is lower than that for those belonging to the labeled clusters and that the difference is statistically significant. The results indicate insufficiency of labeled data for data belonging to unknown clusters.

To improve the accuracy of a semi-supervised classification, we then propose two methods to increase the quantity of labeled data in the unknown cluster. The first method is an active labeling that applies an active learning technique. The active labeling method selects the most informative and representative data in the unknown clusters and then gives a class label provided by the user to the selected data. The results show that the labeled dataset improved with active labeling significantly improves the performance of semi-supervised classification. However, the active labeling requires manual action from the user. To overcome this limitation, we propose the second method, namely, co-labeling. The co-labeling method applies the efficient classifier to automatically label the selected data in the unknown cluster. The result shows that random forest is the best approach of the six approaches tried for assigning the class label. Moreover, the co-labeling with random forest was able to enhance the performance of a self-training classifier that was degraded by insufficient labeled data, to become significantly better than a supervised classifier on many datasets. To the best of our knowledge, this work is the first study that analyzes the sufficiency of the training data for semi-supervised classification.

The structure of this paper is as follows. A review and discussion of semi-supervised self-training are presented in Section 2. Related approaches to learning with unlabeled data and a cluster analysis are presented in Section 3. A description of the proposed method to analyze the training data with semi-supervised clustering and experimental results are presented in Section 4. The two proposed methods for improving the labeled dataset in semi-supervised classification are described in Section 5. Finally, the last section provides a conclusion and directions for future work.

2. Semi-supervised self-training

Semi-supervised learning is a machine learning approach that applies both labeled and unlabeled data to create a learner. Whereas labeled data consist of attribute values and class labels, unlabeled data consist only of attribute values. Labeled data are scarce because a process to give a class label to each data item is costly and time-consuming, but unlabeled data are typically available. Semi-supervised learning is halfway between supervised learning (e.g., the training of support vector machines to classify positive and negative examples) and unsupervised learning (e.g., data clustering with the K-means algorithm) [1]. The objective of semi-supervised learning is to create a learner L with a set of labeled data (X_l, y_l) and set of unlabeled data X_u, where X denotes a vector of attribute values and y denotes a class label. The number of labeled data is l and the number of unlabeled data is u, and $l < < u$.

There are many semi-supervised learning approaches, including self-training [2], co-training [3], generative models [4], graph-based methods [5,20], transductive support vector machines [6], and learning from positive and unlabeled examples [21]. Self-training is one of the most efficient, simple, and widely used methods in many domains, including object detection [7], face recognition [8], sentence parsing [9], EEG classification [10,11], and time series problems [12]. The self-training approach does not require any specific assumptions or prerequisites, as other approaches do. Moreover, self-training can be used along with other approaches, as in cost-sensitive learning [22], ensemble methods [23,24], and multiple classifier systems [25].

The process of self-training begins with the application of all available labeled data to train an initial classifier. Then, the initial classifier is used to find a class label for each unlabeled data item. The unlabeled data with the highest confidence values are combined with the originally labeled data to create a newly labeled dataset. The newly labeled dataset is then used to update the classifier for labeling unlabeled data in the next iteration. This iterative process continues until a stopping condition is met, such as that all unlabeled data have been given class labels. Finally, the originally labeled dataset and the recently labeled dataset are combined and used to train a final classifier. A self-training algorithm is given as Algorithm 1.

Algorithm 1 Self-training.

```
Initialize:
Given $(X_{train}, y_{train}) = (X_l, y_l)$
while stopping criteria not met do
    Train classifier $C_{int}$ from $(X_{train}, y_{train})$
    Use $C_{int}$ to predict class label $y_u$ of $X_u$
    Select confidence sample $(X_{conf}, y_{conf})$; $(X_{conf}, y_{conf}) \in (X_u, y_u)$
    Remove selected unlabeled data $X_u \leftarrow X_u - X_{conf}$
    Combine newly labeled data $(X_{train}, y_{train}) \leftarrow (X_l, y_l) \cup (X_{conf}, y_{conf})$
end while
```

Self-training overcomes the problem of insufficient labeled data by iteratively giving a class label to the unlabeled data having the highest confidence values. However, self-training can also introduce incorrectly labeled data into the original training set. This is because self-training uses a few labeled data, which may be insufficient, for training an initial classifier. To reduce the effect of the incorrectly labeled data in self-labeling, many techniques have been proposed, including self-training with editing [18], which adds a step to correct the mislabeled data; noise filtering for self-training [17], which applies a filtering method to refine the newly labeled data; and the self-training nearest neighbor rule using cut edges (SNNRCE) [19], which applies a neighbor graph for labeling in the initial labeling step. To evaluate the performance of each self-labeling method, Triguero and Salvador García [26] conducted experiments to compare 14 self-labeling methods on 55 standard datasets and using four classification algorithms: k-nearest neighbor, decision tree, naive Bayes, and support vector machines. Their results showed that there is no single best method for all datasets; the most efficient method for one dataset might perform inefficiently on other datasets. Interestingly, the performance of the standard self-training method was outstanding on many datasets.

In this work, we propose a new approach for improving standard semi-supervised self-training with preprocessing steps. Whereas most recent studies have focused on a post-processing technique to fix incorrectly labeled data, we propose a preprocess-