Applied Soft Computing 37 (2015) 296–302

Contents lists available at ScienceDirect

Applied Soft Computing

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

On learning dual classifiers for better data classification

Wei-Chao Lin[a], Chih-Fong Tsai[b,*], Shih-Wen Ke[c], Mon-Loon You[a]

[a] Department of Computer Science and Information Engineering, Hwa Hsia University of Technology, Taiwan
[b] Department of Information Management, National Central University, Taiwan
[c] Department of Information and Computer Engineering, Chung Yuan Christian University, Taiwan

ARTICLE INFO

Article history:
Received 30 October 2014
Received in revised form 6 August 2015
Accepted 19 August 2015
Available online 2 September 2015

Keywords:
Data mining
Machine learning
Instance selection
Training effectiveness

ABSTRACT

Instance selection aims at filtering out noisy data (or outliers) from a given training set, which not only reduces the need for storage space, but can also ensure that the classifier trained by the reduced set provides similar or better performance than the baseline classifier trained by the original set. However, since there are numerous instance selection algorithms, there is no concrete winner that is the best for various problem domain datasets. In other words, the instance selection performance is algorithm and dataset dependent. One main reason for this is because it is very hard to define what the outliers are over different datasets. It should be noted that, using a specific instance selection algorithm, over-selection may occur by filtering out too many 'good' data samples, which leads to the classifier providing worse performance than the baseline. In this paper, we introduce a dual classification (DuC) approach, which aims to deal with the potential drawback of over-selection. Specifically, performing instance selection over a given training set, two classifiers are trained using both a 'good' and 'noisy' sets respectively identified by the instance selection algorithm. Then, a test sample is used to compare the similarities between the data in the good and noisy sets. This comparison guides the input of the test sample to one of the two classifiers. The experiments are conducted using 50 small scale and 4 large scale datasets and the results demonstrate the superior performance of the proposed DuC approach over the baseline instance selection approach.

1. Introduction

Developing effective classification (or prediction) models is usually the key to success to most data mining or pattern recognition problems. Effective classification models are those which provide high classification accuracy or other related measures, such as the ROC curve [20] and *F*-score [1]. Generally, in order to examine the performance of classification models, a given testing set with the ground truth answer is used. The models that produce more correct answers corresponding to the testing data are regarded as performing better than the ones that produce fewer correct answers. Reducing the classification error rate is one of the key issues in most data mining and pattern recognition research strategies.

However, despite the fact that well-known supervised learning (or classification) techniques are used, including support vector machines, decision trees, or *k*-nearest neighbors [2], the developed classifiers will inevitably produce a certain proportion of incorrect answers in many problem domains. In other words, the development of a classifier with 100% classification accuracy over various testing sets is unlikely.

In order to improve the classification accuracy (or reduce the classification error), one of the most popular solutions is to design a more sophisticated classification model based on novel hybrid and ensemble method [3–6]. In addition, these kinds of models have been shown to outperform single learning based models.

Since the data samples in the feature space are usually nonlinear and very complex, the training effectiveness of the learning models over a given training set is limited. In other words, a given training set generally contains a certain amount of noisy data, which is likely to degrade the performance of the trained models. Instance selection, a data pre-processing step during knowledge discovery in the databases (KDD) process, aims at reducing the dataset size by filtering out noisy data from a given dataset [7,8].

In this paper, we propose a dual classification (DuC) approach, which aims at improving the training effectiveness of the learning models through instance selection. In particular, performing instance selection for a given dataset can result in the selected subset containing 'good' training data and a filtered out subset of 'noisy' training data. Two different classifiers are then respectively trained using these two subsets. For the testing step, the *k*-nearest

* Corresponding author. Tel.: +886 3 422 7151; fax: +886 3 4254604.
E-mail address: cftsai@mgt.ncu.edu.tw (C.-F. Tsai).

http://dx.doi.org/10.1016/j.asoc.2015.08.038
1568-4946/

neighbor similarity measure is used to respectively compare the testing data sample with the data in the two subsets. The resultant training sample is the nearest to the testing sample that can be identified. Thus, depending on which subset the training sample belongs to the testing sample is then input into its corresponding classifier. Finally, the classification results are obtained from the outputs of the two classifiers.

This proposed DuC approach is different from conventional instance selection in that it omits the filtered out data samples during the model training step. This is because it is hard to define the noise level of the data samples (or outliers), meaning that over- or under-selection result may be produced which is likely to degrade the model performance (c.f. Section 2.2).

The rest of this paper is organized as follows. Section 2 briefly describes the basic concepts of pattern classification and instance selection. Section 3 introduces the proposed approach. Section 4 presents the experimental results and conclusions are provided in Section 5.

2. Literature review

2.1. Pattern classification

The goal of pattern classification is to allocate an object represented by a number of measurements (i.e. feature vectors) into one of a finite set of classes. Supervised learning can be thought of as learning by example or learning with a teacher. The teacher has knowledge of the environment which is represented by a set of input–output examples. In order to classify unknown patterns, a certain number of training samples are available for each class, and they are used to train the classifier [9].

The learning task is to compute a classifier or model $\hat{f}$ that approximates the mapping between the input–output examples and correctly labels the training set with some level of accuracy. This can be called the *training* or *model generation* stage. After the model $\hat{f}$ is generated or trained, it is able to classify an unknown instance, into one of the learned class labels in the training set. More specifically, the classifier calculates the similarity of all trained classes and assigns the unlabelled instance to the class with the highest similarity measures.

2.2. Instance selection

Instance selection can be defined as follows. Given a dataset D composed of training set T and testing set U, let X_i be the ith instance in D, where $X_i = (X_1, X_2, ..., X_m)$ which contains m different features. Let $S \subset T$ be the subset of selected instances resulting from the execution of an instance selection algorithm. Then, U is used to test a classification technique trained by S [10,11].

It should be noted that, in practice, it is very difficult to optimize the instance selection result, which produces the largest reduction rate and at the same time makes the classifiers perform better than the baseline classifiers without performing instance selection. In other words, given a training set over- or under-selection could occur, where the former filters out too many instances and the later very few instances. As a result, the classifier trained by the over- or under-selected training set is unlikely to provide comparable results with the baseline classifier. This is a major challenge for instance selection, but it is not the major focus of this paper, which is to propose a better instance selection algorithm.

2.2.1. ENN

Edited Nearest Neighbor (ENN) [12] is one representative noise-filtering algorithm, in which S starts out the same as T, and then each instance in S is removed if it does not agree with the majority of its k nearest neighbor (with $k = 3$, typically). This edits out noisy instances, as well as close border cases, leaving smoother decision boundaries. The ENN algorithm can be expressed as follows:

1. Let $S = T$.
2. For each x in T do:
   - discard x from S if it is misclassified using the k-NN rule with prototypes in $T-\{x\}$.

2.2.2. IB3

IB3 was introduced by Aha et al. [13], which is based on an acceptable instance concept to carry out the selection. That is, instance x from the training set is added to a new set S if the nearest acceptable instance in S (if there is no acceptable instance a random one is used) has different class than x. Acceptability is defined by a confidence interval

$$\frac{p + (z^2/2n) \pm z\sqrt{(p(p-1))/n + (z^2/2n^2)}}{1 + (z^2/n)} \quad (1)$$

where z is a confidence factor (in IB3 0.9 is used to accept, 0.7 to reject), p is the classification accuracy of a given instance (while added to S), and n is equal to a number of classification-trials for the given instance (while added to S).

In Aha et al. [13], IB1, IB2, and IB3 were compared with the C4 decision tree algorithm over six datasets. In particular, the sizes of the chosen datasets range from 303 to 800 data samples and the numbers of attributes of each data sample range from 7 to 21. The results show shat IB3 can significantly reduce IB1's storage requirements and does not display IB2's sensitivity to noise. On average, IB3 can allow k-NN to provide the highest rage of classification accuracy than C4 does. However, they found that IB3's learning performance is highly sensitive to the number of irrelevant attributes used to describe instances.

2.2.3. DROP3

The Decremental Reduction Optimization Procedure 1 (DROP1) was proposed by Wilson and Martinez [7]. DROP1 uses the following basic rule to decide if it is safe to remove an instance from the instance set S (where $S = T$ originally):

*R*emove P if at least as many of its associates in

S would be classified correctly without$P$. (2)

The DROP1 algorithm begins by building a list of nearest neighbors for each instance, as well as a list of associates. Then each instance in S is removed if its removal does not hurt the classification of the instances remaining in S. When an instance P is removed, all of its associates must remove P from their list of nearest neighbors, and then must find a new nearest neighbor so that they still have $k + 1$ neighbors in their list. When they find a new neighbor N, they also add themselves to N's list of associates so that at all times every instance has a current list of neighbors and associates.

However, in DROP1 a noisy instance will typically have associates of a different class, and will thus cover a somewhat small portion of the input space. If its associates are removed according to the above rule, the noisy instance may cover more and more of the input space. Eventually it is hoped that the noisy instance itself will be removed. On the other hand, if many of its neighbors are removed first, its associates may eventually include instances of the same class from the other side of the original decision boundary.

Therefore, DROP2, the second version of DROP, was proposed to overcome this problem by the process from sorting instances in a chosen training set according to their distances from the nearest opposite class instance (or nearest enemy). Then, instances are removed if they are furthest from their nearest enemies.