Engineering Applications of Artificial Intelligence 53 (2016) 62–73

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

One-sided Dynamic Undersampling No-Propagation Neural Networks for imbalance problem

Qi Fan, Zhe Wang*, Daqi Gao

Department of Computer Science & Engineering, East China University of Science & Technology, Shanghai 200237, PR China

ARTICLE INFO

Article history:
Received 15 January 2015
Received in revised form
9 December 2015
Accepted 23 February 2016
Available online 13 April 2016

Keywords:
Imbalanced problem
Sample selection
Neural networks
Pattern recognition

ABSTRACT

Imbalanced problem occurs when the size of one class, i.e. the minority class, is much lower than that of the other classes, i.e. the majority classes. Conventional data level methods are employed as the preprocessing approaches to balance the datasets before the classifier learning. Since the balanced data remains unchanged during the learning process, one pre-deleted sample would never be used to train the classifier, which may result in information loss. To solve this problem, this work presents an One-sided Dynamic Undersampling (ODU) technique which adopts all samples in the training process, and dynamically determines whether a majority sample should be used for the classifier learning. Thus, ODU can dynamically undersample the majority class to balance the dataset. To validate the effectiveness of ODU, we integrate it into No-Propagation neural networks to propose an ODU No-Propagation Neural Networks (ODUNPNN). ODUNPNN takes all training samples into consideration, and dynamically undersamples majority class after each iteration, i.e. ODUNPNN integrates undersampling approach into the classifier learning process. Experimental results on both synthetic and real-world imbalance datasets demonstrate that ODUNPNN outperforms the NPNN-based algorithms, and results in comparative performance compared with LASVM-AL, EasyEnsemble, and DyS on real-world imbalance datasets. The contributions of this paper are: (1) ODUNPNN integrates undersampling approach into the classifier learning process. (2) ODUNPNN dynamically balances training data in each iteration. (3) ODU technique can be integrated into other classification learning machines.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

In recent years, imbalance learning problem has drawn a significant amount of interests in data mining (García-Pedrajas et al., 2013; Khoshgoftaar et al., 2011; Martino et al., 2013), and it is common in many real-world domains, such as e-mail filtering (Dai, 2015), fraud detection (Deng and Tian, 2013), detection of oil spills from satellite images (Guo and Zhang, 2014), and medical diagnosis (Ozcift and Gulten, 2011). This problem occurs when the number of samples of one class, i.e. the minority class, is much lower than that of the other classes, i.e. the majority classes. However, most standard algorithms are proposed with the assumption on the balanced class distributions or equal misclassification costs (Brown and Mues, 2012). When faced to complex imbalance problems, these algorithms fail to properly represent the distributive characteristics of the data and result in the unfavorable accuracies. Thus, the fundamental issue on the imbalance problem is the ability of imbalance data to significantly compromise the performance of the standard learning algorithms (He and Garcia, 2009).

Lots of approaches have been proposed on the data level to balance the data for the standard learning algorithms. It is obvious that the data level approaches are classifier-independent, i.e. one processed dataset can be adopted to train multiple different classifiers (He and Garcia, 2009). There are different balancing methods which can be classified into three groups: undersampling methods (Galar et al., 2013; Kubat and Matwin, 1997), oversampling methods (Barua et al., 2014; Chawla et al., 2011), and hybrid methods. One of the simplest undersampling method is the Random-Undersampling (RUS) (Batista et al., 2004) which aims to balance the class distribution by the random elimination of the majority samples. However, RUS might discard potentially useful data which could be important for the classifier learning. Kubat and Matwin (1997) propose an alternative undersampling approach named as One-Sided Selection (OSS). In OSS, all minority samples are preserved, and the majority samples are selected based on the nearest neighbor classification method and Tomek (1976) links which is a useful definition for cleaning data. The Random-Oversampling (ROS) (Batista et al., 2004) is known as a

* Corresponding author.
E-mail addresses: wangzhe@ecust.edu.cn (Z. Wang), gaodaqi@ecust.edu.cn (D. Gao).

http://dx.doi.org/10.1016/j.engappai.2016.02.011
0952-1976/© 2016 Elsevier Ltd. All rights reserved.

popular oversampling approach to balance the class distribution by the random replication of the minority samples, which might result in the risk of over-fitting and the aggravation of computational burden.

It should be declared that the existing data level approaches are all independently employed before the classifier learning. They are treated as the preprocessing to balance the class distribution. Since the balanced data remains unchanged during the learning process, one pre-removed sample would never be used to train the classifier, which may result in information loss (Lin et al., 2013). Thus, to overcome this problem, Ertekin et al. (2007a,b) propose an efficient way of selecting informative samples by selecting the samples from the decision border based on SVM-based active learning (LASVM-AL). Recently, Martino et al. (2013) present a Dynamic Sampling approach (DyS) to train neural networks for multi-class imbalance problems. In each epoch, DyS firstly estimates the probability of each sample selected for training the neural networks. Then, it dynamically selects informative samples to train the networks based on the probability. Alternatively, this paper proposes an One-sided Dynamic Undersampling (ODU) technique which adopts all training samples in training process, and dynamically undersample the majority class to balance the data with 1.00 Imbalance Ratio (IR) based on the contribution of the majority samples to the decision hyperplane. In this work, IR (Fernández et al., 2013), defined as the ratio of the number of majority and minority samples, is used to represent the imbalance level of a specific dataset.

In practice, we select one iterative-training classification algorithm named No-Propagation Neural Networks (NPNN) (Widrow et al., 2013) as the paradigm to integrate ODU with it, and result in an One-sided Dynamic Undersampling No-Propagation Neural Networks (ODUNPNN). In the training process of NPNN, the weights linking the input layer to the hidden layer are randomized and fixed during the training process. Only the weights between the hidden layer and the output layer are trained by the steepest descent to minimize mean squared error. Since NPNN employs the iterative optimization approach, ODUNPNN undersamples the majority class to balance the data to be 1.00 IR in each iteration, which results in dynamically undersampling the majority class. That is why the proposal is called as One-sided Dynamic Undersampling No-Propagation Neural Networks. It should be declared that although the proposed ODUNPNN seems similar to the recently proposed DyS (Martino et al., 2013) which also adopts a dynamic sampling approach to handle the class imbalance problems, there are several differences between them. (1) In DyS, the back-propagation approach is adopted to train the networks. While, in ODUNPNN, the weights linking the input layer to hidden layer are randomly initialized and fixed in the training process. The weights between hidden and output layers are trained. Thus, ODUNPNN does not employ back-propagation approach to train the networks, only the weights linking hidden layer to output layer are learned, which results in that ODUNPNN is easier to train the networks than DyS. (2) In each epoch, DyS needs to calculate the selection probability for each training sample. While ODUNPNN directly employs the network output to determine which samples should be selected for training. (3) DyS undersamples both majority and minority samples during the training process. While ODUNPNN only undersamples the majority samples and preserving all minority samples since the minority class is important for the learning task. (4) ODUNPNN can prevent the IR of the selected training samples to be 1.00 by selecting the same number of informative majority samples of the minority samples. While DyS does not have this property of balancing the selected training samples. Therefore, from the above differences, we can conclude that the proposed ODUNPNN is different from DyS. Moreover, ODUNPNN is simpler than DyS. The main contributions of this work are highlighted as follows:

- ODUNPNN integrates the undersampling approach into the classifier learning process to dynamically balance the data based on the contribution of the majority samples to the decision hyperplane.
- ODUNPNN results in the balanced data for each training iteration. In doing so, the classifier training process can pay more attention to the minority class to learn robust decision boundary.
- ODU technique can also be integrated into other classification learning machines which adopt the iterative optimization approaches.

The rest of this paper is organized as follows: Section 2 presents the detailed description on the proposed one-sided dynamic undersampling no-propagation neural networks. Then, the experimental results on both synthetic and real-world including binary-class and multi-class imbalance datasets are shown in Section 3. Following that, Section 4 presents the concluding remarks of this paper.

2. One-sided Dynamic Undersampling No-Propagation Neural Networks

In this section, the architecture of the proposed One-sided Dynamic Undersampling No-Propagation Neural Networks (ODUNPNN) is presented. We firstly introduce the proposed One-sided Dynamic Undersampling technique, then, present the architecture of ODUNPNN.

2.1. One-sided Dynamic Undersampling (ODU)

Conventional data level approaches balance the data distribution before the classifier learning which might result in the unfavorable balanced data since the pre-removed samples would never be used to train the classifier. Moreover, in experimental results, we find that not all the samples have the same contribution to the decision hyperplane. The samples located in the junction of the classes contain more contribution than the other ones, i.e. the samples near the decision boundary are more important than the ones far away from the boundary for the classifier training (Ertekin et al., 2007a,b). Thus, in this work, we adopt the samples in the majority class, which are near the decision boundary, as the negative samples, and all minority samples as the positive samples for training. To balance the class distribution, the number of the selected negative samples is equal to that of the positive, i.e. IR of the selected data is 1.00. By adopting the iterative-training algorithm, the selected data is dynamically determined after each training iteration. Fig. 1 provides a vivid description on ODU technique. The red line is the decision boundary after lth iteration. The blue stars are the majority points, which are treated to be near the decision boundary, selected as the negative samples for the next iteration. All minority samples are adopted as the positive samples, i.e. the red circles. Thus, the blue stars and the red circles are selected as the training samples for the $(l+1)$th iteration. It should be declared that all the samples are adopted as the training samples for the first iteration. After that, the training samples are dynamically determined. In doing so, the learner can pay more attention to the minority class to results in a more robust classifier.

2.2. One-sided Dynamic Undersampling No-Propagation Neural Networks (ODUNPNN)

Fig. 2 shows a fully connected tree layer feed-forward neural network with one output node. The network inputs are pattern vectors. The bias for each neuron is not drawn in Fig. 2 for simple

Download English Version:

https://daneshyari.com/en/article/380177

Download Persian Version:

https://daneshyari.com/article/380177

Daneshyari.com