Han C, Tan YK, Zhu JH *et al.* Online feature selection of class imbalance via PA algorithm. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 31(4): 673–682 July 2016. DOI 10.1007/s11390-016-1656-0

Online Feature Selection of Class Imbalance via PA Algorithm

Chao Han, Yun-Kun Tan, Jin-Hui Zhu, *Member, IEEE*, Yong Guo, Jian Chen*, *Member, CCF, ACM, IEEE*, and Qing-Yao Wu*, *Member, CCF, IEEE*

School of Software Engineering, South China University of Technology, Guangzhou 510000, China

E-mail: {hanchaos, yunkuntan}@163.com; csjhzhu@scut.edu.cn; guoyong2219@gmail.com
{ellachen, qyw}@scut.edu.cn

Received March 6, 2016; revised May 18, 2016.

Abstract Imbalance classification techniques have been frequently applied in many machine learning application domains where the number of the majority (or positive) class of a dataset is much larger than that of the minority (or negative) class. Meanwhile, feature selection (FS) is one of the key techniques for the high-dimensional classification task in a manner which greatly improves the classification performance and the computational efficiency. However, most studies of feature selection and imbalance classification are restricted to off-line batch learning, which is not well adapted to some practical scenarios. In this paper, we aim to solve high-dimensional imbalanced classification problem accurately and efficiently with only a small number of active features in an online fashion, and we propose two novel online learning algorithms for this purpose. In our approach, a classifier which involves only a small and fixed number of features is constructed to classify a sequence of imbalanced data received in an online manner. We formulate the construction of such online learner into an optimization problem and use an iterative approach to solve the problem based on the passive-aggressive (PA) algorithm as well as a truncated gradient (TG) method. We evaluate the performance of the proposed algorithms based on several real-world datasets, and our experimental results have demonstrated the effectiveness of the proposed algorithms in comparison with the baselines.

Keywords online learning, feature selection, class imbalance, passive-aggressive (PA) algorithm

1 Introduction

Imbalance classification is an important research topic which has been extensively studied for the last decade in data mining and machine learning area. The main objective of imbalance classification is to build effective and efficient classification models to learn from the imbalanced dataset in which the majority (or positive) class has many more examples than the minority (or negative) class[1]. One example is the disease diagnostic problem where the disease cases are usually quite rare in comparison with normal cases. In this case, a traditional classification model might have high accuracy overall but it provides a much lower identification rate w.r.t. the minority class (i.e., the disease category) because most of the classification algorithms aim to optimize the performance on the whole training dataset. Therefore, the traditional classification approaches cannot be applied to solve the imbalanced classification problem directly.

In order to improve the scalability, efficiency, and accuracy of high-dimensional datasets, feature selection (FS) has been applied to many real-world machine learning and data mining applications[2-5]. The goal of feature selection is to select a subset of active features to build effective and efficient classification models. Due to the imbalanced data, the features related to the negative class are usually not comparable with those related to the positive class. Furthermore, feature selection and imbalance classification methods only focus on off-line batch learning, that is, the feature selection and the classification task run in an off-line fashion. This as-

Regular Paper

Special Section on Data Management and Data Mining 2016

This research was supported by the Guangzhou Key Laboratory of Robotics and Intelligent Software under Grant No. 15180007, the Fundamental Research Funds for the Central Universities of China under Grant Nos. D215048w and 2015ZZ029, and the National Natural Science Foundation of China under Grant Nos. 61005061 and 61502177.

*Corresponding Author

sumption, however, may not hold in many real-world scenarios where the training data actually are formed in sequence.

Recently, online feature selection (OFS) is considered in the literature[6-7], and the task is to build an online learner which only uses a small and fixed number of features for accurately classifying examples arriving in a sequential manner. OFS is particularly important and necessary when a real-world application has to deal with sequential and high-dimensional training data or it is expensive to collect the full information of the training data.

Motivated by recent progresses in online feature selection and imbalance classification[8-10], in this paper, we propose two novel online learning algorithms to conduct feature selection and imbalance classification at the same time. Our goal is to alleviate the effect of the curse of dimensionality, to speed up the learning process, and to improve the model interpretability by constructing an accurate and efficient online imbalanced classifier. To achieve this, we extend the passive-aggressive (PA) algorithm to build the online learner in following perspectives: 1) we set different margins of loss function for the majority class and the minority class; 2) we employ an oversampling method to increase the number of examples of the minority class; 3) we use a truncated gradient (TG)[11] method to select a small and fixed number of features to simplify the weight vector. We then design an optimization scheme to infer the learner for deciding the separating hyperplane by using an iterative approach. The major contributions of this paper are as follows.

• We propose two new algorithms to address the online imbalanced classification problem by extending the PA algorithm.

• We consider the case by using a feature selection method to improve the classification performance based on high-dimensional imbalanced datasets in an online learning setting.

• We validate the effectiveness and the efficiency of the proposed algorithms by conducting highly valuable experiments based on real-world datasets.

The rest of the paper is organized as follows. In Section 2, we give a brief review of related work in class imbalance, feature selection, and online learning. In Section 3, we describe our proposed algorithms. Extensive experimental results are presented in Section 4. Finally, we provide concluding remarks and a discussion of future directions in Section 5.

2 Related Work

Our work is closely related to imbalanced classification, feature selection, and online learning. Therefore, in this section, we briefly review the important works w.r.t. these three areas.

In the past decade, a number of imbalanced classification approaches have been developed. These methods can be mainly divided into three categories: sampling techniques, new algorithms, and feature selection methods[1]. Sampling techniques, such as random oversampling and random undersampling methods, are exploited to solve the class imbalance problem in a dataset. Random oversampling duplicates randomly selected examples of the minority class to balance the class distribution. For example, Chawla *et al.*[12] proposed synthetic minority over-sampling technique (SMOTE) to synthesize minority examples by artificially interpolating the preexisting minority instances. On the other hand, the random undersampling method discards examples from the majority class, which may lead to a loss of important information[13].

Some researchers have exploited new approaches to enhance the learning performance of imbalanced classification, such as the one-class learning and the cost-sensitive learning. These methods deal with the class imbalance problem in different ways to optimize the performance of learning algorithm based on unseen data[5]. For example, when positive examples greatly outnumber the negative ones, most classifiers tend to overfit[10]. And one-class learning methods aim to combat the overfitting problem by approaching it from an unsupervised learning perspective, in which a one-class learner is built to recognize the samples about whether they belong to a specific target class. These methods attempt to measure the similarity between a query example and the target class, where classification is accomplished by imposing a threshold on the similarity value, such as one-class SVM trainer[14]. The main idea of cost-sensitive learning is to minimize the overall cost of training dataset with respect to a specific loss function, and the cost can be considered as a penalty representation when an example is classified into the wrong class. And these methods usually suppose that the misclassification for the minority class will be assigned higher cost compared with that for the majority class[15]. In recent years, some researchers have extended the idea of cost-sensitive learning to address the issue of online class imbalance[16-17].

Feature selection has been applied to various applications of high dimensionality in order to improve

the scalability, efficiency, and accuracy[5]. A number of feature selection metrics have been explored, such as information gain, chi-square, correlation coefficient, and odds ratio[4,18-19]. Imbalanced datasets are commonly encountered in high-dimensional and large-scale classification problems. Consequently, some researchers have considered feature selection to relieve the effect of skewed data[5,12]. For example, Maldonado *et al.*[13] proposed a backward elimination approach based on successive holdout steps, whose contribution measure is based on a balanced loss function obtained on an independent subset. Wu *et al.*[20] proposed the ForesTexter (RF) algorithm to solve the imbalanced text categorization problem based on an ensemble method, where each decision tree was built by using a simple random sampling. And Wu *et al.*[21] used a stratified sampling method to select feature subspace when generating decision trees of a random forest for genome-wide association (GWA) data.

Various researchers have also considered different online learning approaches. For example, Perceptron algorithm is a well-known online learning model, where the parameters are iteratively updated when the misclassification happens in the training phase until it finds a support vector that can correctly classify all the training data[22-23]. The PA algorithm[24] follows the criterion of maximum margin learning principle to update a classifier that is near to the previous function while suffering less loss based on the current instance.

Recently, online feature selection (OFS) has received considerable attention[6-7] where an online learner is only allowed to maintain a classifier with only a small and fixed number of features. OFS aims to use feature selection techniques in an online fashion, which is more appropriate for real-world applications. The key challenge of OFS is to build an accurate model in the online learning process by using a small and fixed number of features which must be relevant and efficiently identified. In [6-7], an effective algorithm is developed to resolve this problem by studying sparsity regularization and truncation techniques.

3 Methodology

In this paper, we propose two online feature selection methods for the imbalanced classification problem, namely the PA algorithm which is used to learn an online classifier, and the truncated gradient method which is used to select a subset of features to infer the classifier.

Unlike batch learning, online learning sequentially builds a prediction model based on the feedback from a sequence of data by processing each data instance upon its arrival. Specially, on each round the online classfier observes one instance, makes a prediction, and receives the ground-truth label. Then the classifier is subsequently updated based on the predicted result. We focus on binary classification where the instance's label is either $+1$ or -1. Formally, we denote the instance by $\boldsymbol{x}$, which is a vector in $\mathbb{R}^n$. We assume that $\boldsymbol{x}$ is associated with a unique label $y \in \{+1, -1\}$ and refer to each instance label pair as $(\boldsymbol{x}, y)$.

In this section, we introduce our proposed algorithms to select features for online imbalanced data. Firstly, we describe the idea and concreteness of the PA algorithm. Secondly, we present two strategies to extend the PA algorithm to exploit imbalanced datasets. Finally, a truncated gradient method is employed to select a fixed number of active features while inferring the online learning models.

3.1 Passive-Aggressive Algorithm

The PA algorithm[24] uses a classification function based on a vector of weights $\boldsymbol{w} \in \mathbb{R}^n$. The prediction model of PA is given as $sign(\boldsymbol{w} \cdot \boldsymbol{x})$ and the model is updated from round to round for the arrived instances. The magnitude $|\boldsymbol{w} \cdot \boldsymbol{x}|$ is the degree of confidence in the prediction. The algorithm sequentially learns the weight vector $\boldsymbol{w}$ where $\boldsymbol{w}_t$ denotes the weight vector on round t. The term $y_t(\boldsymbol{w}_t, \boldsymbol{x}_t)$ where $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ denote the instance and its label on round t respectively, is the signed margin attained on round t. The margin is a positive number when $sign(\boldsymbol{w}_t \cdot \boldsymbol{x}_t) = y_t$, i.e., the algorithm makes a correct prediction. To maintain high confidence on prediction, the PA algorithm achieves a margin of at least 1 as often as possible. However, the choice of 1 as the margin threshold is rather arbitrary. On rounds where the algorithm attains a margin less than 1, it suffers an instantaneous loss. Such loss is defined by the following hinge loss function,

$$\ell(\boldsymbol{w}; (\boldsymbol{x}, y)) = \begin{cases} 0, & \text{if } y(\boldsymbol{w} \cdot \boldsymbol{x}) \geqslant 1, \\ 1 - y(\boldsymbol{w} \cdot \boldsymbol{x}), & \text{otherwise.} \end{cases} \tag{1}$$

Whenever the margin exceeds 1, the loss equals zero. Otherwise, it is equal to the difference between the margin and 1.

Concretely, the weight vector $\boldsymbol{w}_1$ is initialized to $(0,\cdots,0)$. There are three variants of update rules used to modify the weight vector at the end of each round. The simplest one is, on round t, to set the new weight

vector $\boldsymbol{w}_{t+1}$ to be the solution to the following constrained optimization problem.

$$\begin{aligned} &\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w}\in\mathbb{R}^n} \frac{1}{2}\|\boldsymbol{w}-\boldsymbol{w}_t\|^2 \\ &\text{s.t.} \quad \ell(\boldsymbol{w};(\boldsymbol{x}_t,y_t)) = 0. \end{aligned} \tag{2}$$

Geometrically, $\boldsymbol{w}_{t+1}$ is set to be the projection of $\boldsymbol{w}_t$ onto the half-space of vectors which attain a hinge-loss of zero on the current example. Whenever the hinge-loss is zero, that is $\ell_t = 0$, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t$. In contrast, on those rounds where the loss is positive, the algorithm forces $\boldsymbol{w}_{t+1}$ to satisfy the constraint $\ell(\boldsymbol{w}_{t+1};(\boldsymbol{x}_t,y_t)) = 0$ regardless of the step-size required. The update, on one hand, requires $\boldsymbol{w}_{t+1}$ to correctly classify the current example with a sufficiently high margin, and thus, progress is made. On the other hand, $\boldsymbol{w}_{t+1}$ must stay as close as possible to $\boldsymbol{w}_t$, as a result of retaining the information learned at previous rounds.

The solution to the optimization problem in (2) has a simple closed form. The detailed induction is given in [24],

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \tau_t y_t \boldsymbol{x}_t, \quad \text{where} \quad \tau_t = \frac{\ell_t}{\|\boldsymbol{x}_t\|^2}. \tag{3}$$

Considering the update where the objective function scales linearly with ξ, namely,

$$\begin{aligned} &\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w}\in\mathbb{R}^n} \frac{1}{2}\|\boldsymbol{w}-\boldsymbol{w}_t\|^2 + C\xi \\ &\text{s.t.} \quad \ell(\boldsymbol{w};(\boldsymbol{x}_t,y_t)) \leqslant \xi \ \& \ \xi \geqslant 0. \end{aligned} \tag{4}$$

Here C in (4) is a positive parameter which controls the influence of the slack term on the objective function. Alternatively, the objective function scales quadratically with ξ, resulting in the following constrained optimization problem,

$$\begin{aligned} &\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w}\in\mathbb{R}^n} \frac{1}{2}\|\boldsymbol{w}-\boldsymbol{w}_t\|^2 + C\xi^2 \\ &\text{s.t.} \quad \ell(\boldsymbol{w};(\boldsymbol{x}_t,y_t)) \leqslant \xi. \end{aligned} \tag{5}$$

Note that the constraint $\xi \geqslant 0$ in (5) is no longer necessary since ξ^2 is always non-negative. The above two updates are called PA-I and PA-II. The updates of PA-I and PA-II also share the simple closed form $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \tau_t y_t \boldsymbol{x}_t$, where

$$\tau_t = \min\{C, \frac{\ell_t}{\|\boldsymbol{x}_t\|^2}\}, \tag{6}$$

or

$$\tau_t = \frac{\ell_t}{\|\boldsymbol{x}_t\|^2 + \frac{1}{2C}}.$$

3.2 Two Extensions of PA Algorithm

In terms of the class imbalance problem, the number of positive examples is much more than that of negative examples. The positive examples contribute much more than the negative examples when updating the online classification models. The PA algorithm trains weighted vector $\boldsymbol{w}$ mainly based on the instantaneous loss when examples are misclassified, and it would incline to attaining a sufficiently large margin for positive examples in imbalanced data. Moreover, compared with the real classification hyperplane, the separating hyperplane trained by the PA algorithm is far away from the positive examples and closed to the negative ones, and thus, many negative examples might be misclassified as positive. Since the classifier based on the PA algorithm would select features with a heavy weighted value of positive examples regardless of negative ones, the informative feature with respect to the negative class might not be included.

3.2.1 Margin-Based PA Algorithm

To deal with the issue of online learning of imbalance dataset, we propose a new algorithm, called Margin-Based PA (MBPA), which sets different margins for positive and negative examples when training the online learner using the PA algorithm. When examples of the minority class are assigned to the wrong class, the penalty is higher compared with the case of misclassifying the majority examples.

In MBPA, when the classifier encounters a positive example and makes a wrong prediction, the algorithm achieves a negative margin, and the loss equals the absolute value of margin. For negative examples, to predict negative examples with a high confidence, the classifier achieves a margin of at least 1. That is, when a negative example is misclassified, it suffers a high loss, which equals the difference between the ratio of numbers of the two classes and the margin value. Formally, instead of using (1) as the loss function, MBPA uses a new loss function as follows:

$$\begin{aligned} &\ell_{\mathrm{MB}}(\boldsymbol{w};(\boldsymbol{x},y)) \\ &= \begin{cases} -y(\boldsymbol{w}\cdot\boldsymbol{x}), & \text{if } y=+1 \ \& \ y(\boldsymbol{w}\cdot\boldsymbol{x}) \leqslant 0, \\ \rho - y(\boldsymbol{w}\cdot\boldsymbol{x}), & \text{if } y=-1 \ \& \ y(\boldsymbol{w}\cdot\boldsymbol{x}) \leqslant 1, \\ 0, & \text{otherwise}, \end{cases} \end{aligned} \tag{7}$$

where $\rho = \frac{p}{n}$, p and n are the number of positive examples and negative examples that have been learned, respectively. The value of ρ dynamically changes with the number of examples incrementally learned.

According to the resolution of PA-I algorithm ((6)), the MBPA algorithm updates the weighted vector $\boldsymbol{w}_{t+1}$ using the following rules:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \tau_t y_t \boldsymbol{x}_t, \text{ where } \tau_t = \min\{C, \frac{\ell_{\mathrm{MB}}}{\|\boldsymbol{x}_t\|^2}\}. \quad (8)$$

3.2.2 Oversampling-Negative PA Algorithm

The PA algorithm aims to train a hyperplane by iteratively updating a weighted vector $\boldsymbol{w}$ in every round. The loss of misclassification takes great effects on the value of $\boldsymbol{w}$. Since the positive examples are much more than the negative ones, the loss inclines to increase the weight of features distinguishing the majority examples and the informative features of minority class will be missed. We propose a sampling approach, called Oversampling-Negative PA (ONPA) algorithm, which increases the minority examples by artificially synthesizing negative examples in learning process to alleviate the imbalance of the majority and the minority classes.

We assume that the hypothesis for classification is linear, and thus, all examples between the two randomly selected examples on one side of the hyperplane are on the same side, which means that all the examples belong to one same class. The PA algorithm is linear and we can artificially synthesize new negative examples based on the above assumption. Concretely, when a negative example $(\boldsymbol{x}_t, y_t)$ reaches in one round, we randomly select N negative examples from the dataset $D = \{(\boldsymbol{x}_i, y_i)|0 < i < t \ \& \ y_i = -1\}$, each denoted by $(\boldsymbol{x}_p, y_p)$. Every intermediate point (example) on connection lines decided by $\boldsymbol{x}_p$ and $\boldsymbol{x}_t$ is viewed as the new negative example, denoted by $(\boldsymbol{x}_q, y_q)$, and $\boldsymbol{x}_q = \boldsymbol{x}_t + \alpha(\boldsymbol{x}_t - \boldsymbol{x}_p)$ $(0 < \alpha < 1)$, $y_q = y_t$. After that, we use the newly synthesized negative examples to update the classifier.

In practice, we cannot know the precise number of training examples in advance. Therefore, N is a variable which changes with the number of examples that have arrived, $N = \beta\frac{p}{n}(\beta > 0)$, where p and n are the numbers of positive and negative examples that have arrived, respectively. β is a scale parameter defined beforehand.

3.3 Truncated Gradient Method

Feature selection can help to improve computational time and reduce occupied memory for high dimensional data classification problems. In this paper, we use the truncated gradient (TG) method to select a fixed number (M) of available features in building online learners. Our goal is to induce sparsity in learning the weight vector while building online classifiers. It is motivated from the stochastic gradient descent and the idea is to combine the coefficient rounding and the sub-gradient algorithm for L_1-regularization. Stochastic gradient descent is a method to calculate the minimum value of a function whose first order derivation is continuous. And the subgradient algorithm for L_1-regularization is an online method to calculate the weighted vector. The detailed induction of TG is presented in [11].

TG shrinks the small coefficients (no larger than a threshold $\theta > 0$) to zero by decreasing a smaller amount each time. The amount of shrinkage is measured by a gravity parameter $g \geqslant 0$:

$$TG(\boldsymbol{w}) = T(\boldsymbol{w} - \eta\nabla_1 L(\boldsymbol{w}; (\boldsymbol{x}, y)), \eta g, \theta), \quad (9)$$

where $\nabla_1 L(\boldsymbol{a}; (\boldsymbol{b}, c))$ is a sub-gradient of $L(\boldsymbol{a}; (\boldsymbol{b}, c))$ with respect to the first variable $\boldsymbol{a}$. The parameter $\eta > 0$ is often referred to as the learning rate.

For a vector $\boldsymbol{v} = (v_1, \cdots, v_d) \in \mathbb{R}^d$, and a scalar $g \geqslant 0$, $T(v, \alpha, \theta) = (T(v_1, \alpha, \theta), \cdots, T(v_d, \alpha, \theta))$, where T is defined as follows,

$$T(v_j, \alpha, \theta) = \begin{cases} \max(0, v_j - \alpha), & \text{if } v_j \in (0, \theta), \\ \min(0, v_j + \alpha), & \text{if } v_j \in (-\theta, 0), \\ v_j, & \text{otherwise.} \end{cases} \quad (10)$$

The truncation is conducted in all K rounds in online learning process. In general, we should not take $K = 1$, especially when η is small since each round modifies $\boldsymbol{w}$ by only a small amount. If t/k is not an integer, we set $g_t = 0$; if t/k is an integer, we set $g_t = Kg$ for a gravity parameter $g > 0$. In general, the larger the parameters g and θ are, the more the incurred sparsity is.

The pseudo-codes of the proposed algorithms, MBPA, and ONPA, are shown in Algorithm 1 and Algorithm 2, respectively.

4 Experiments

In this section, we conduct extensive experiments with six public machine learning datasets, including covtype, mnist, ups, w7a, w8a and farm, to investigate the effectiveness of the proposed algorithms. And the results show that our proposed algorithms are able to achieve remarkable improvement against the compared baselines. These datasets are downloaded from http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/.

Algorithm 1. MBPA Algorithm

Input: $\{\boldsymbol{x}_t, y_t | t = 1, \cdots, T\}$, C, M, η, g, θ, K
Output: M selected features
Initialization: $\boldsymbol{w}_1 = (0, \cdots, 0)$, $p = 1$, $n = 1$, $\rho = 1$;
foreach $t = 1, 2, \cdots, T$ **do**
  Attain an example $(\boldsymbol{x}_t, y_t)$;
  Predict its label $\widehat{y}_t = sign(\boldsymbol{w}_t \cdot \boldsymbol{x}_t)$;
  if $y_t = +1$ **then**
    $p = p + 1$;
  else
    $n = n + 1$
  Update the value of $\rho = \frac{p}{n}$;
  Update weighted vector $\boldsymbol{w}$ by using (8) and (7);
  if $t\%K = 0$ **then**
    Truncate weighted vector $\boldsymbol{w}$ by using (9) and (10);
return the first M largest features;

Algorithm 2. ONPA Algorithm

Input: $\{\boldsymbol{x}_t, y_t | t = 1, \cdots, T\}$, C, M, α, β, η, g, θ, K
Output: M selected features
Initialization: $\boldsymbol{w}_1 = (0, \cdots, 0)$, $p = 1$, $n = 1$, $N = 0$;
foreach $t = 1, 2, \cdots, T$ **do**
  Attain an example $(\boldsymbol{x}_t, y_t)$;
  Predict its label $\widehat{y}_t = sign(\boldsymbol{w}_t \cdot \boldsymbol{x}_t)$;
  if $y_t = +1$ **then**
    $p = p + 1$;
  else
    $n = n + 1$
  Update the value of $N = \frac{p}{n}$;
  Update weighted vector $\boldsymbol{w}$ by using (3) and (1);
  if $y_t = -1$ **then**
    for $k = 1, 2, \cdots, N$ **do**
      Randomly select a negative example $(\boldsymbol{x}_p, y_p)$ in D;
      Randomly select a value between 0 and 1 for α;
      $\boldsymbol{x}_q = \boldsymbol{x}_t + \alpha(\boldsymbol{x}_t - x_p)$ $(0 < \alpha < 1)$, $y_q = y_t$;
      Use $(\boldsymbol{x}_q, y_q)$ to update weighted vector $\boldsymbol{w}$ by using (3) and (1);
  if $t\%K = 0$ **then**
    Truncate weighted vector $\boldsymbol{w}$ by using (9) and (10);
return the first M largest features;

In these experiments, the majority class is considered as the positive class, and the others are considered as the negative class. The negative examples of covtype dataset are sampled from the aboriginal negative dataset. For the mnist dataset, the class numbered 0 and the classes numbered 1∼9 are redefined as the negative class and the positive class, respectively. The preparation of ups dataset goes the same as that of the mnist dataset. And we randomly sample 100 negative examples from the farm dataset as its negative examples. Table 1 shows the description of the six datasets.

Table 1. Datasets

Dataset	Positive	Negative	Feature
covtype	283 581	60 000	54
mnist	24 720	7 841	123
ups	6 097	1 194	256
w7a	21 985	661	300
w8a	44 226	1 230	300
farm	1 933	100	20 000

Note: Positive, negative, and feature represent the number of the positive examples, the negative examples and the features of each dataset, respectively.

4.1 Evaluation Criteria

For class imbalance learning, it is unfair to evaluate the performance of algorithms by using the average accuracy (the percentage of testing examples correctly recognized by the classifier) since the number of positive examples is much larger than the number of negative examples. For instance, there are 990 positive examples and 10 negative examples. Even if all examples are classified as positive, the classifier will achieve 99% accuracy, and this assessment is not acceptable for class imbalance learning. Therefore, we choose the specificity and the geometric mean (abbreviated as g-mean) as performance evaluation criteria for our proposed and compared algorithms. We define the specificity as the percentage of negative examples which are correctly classified as the negative class by a classifier, and it is calculated by using the confusion matrix (in Table 2) whose fields characterize the classification behavior of a given classifier. For instance, c is the number of misclassified negative examples and d is the number of correctly classified negative examples. The equation of calculating the specificity is given as following:

$$specificity = \frac{d}{c + d}.$$

Informally, besides that the negative examples are correctly recognized, the classifier is expected to perform well on both negative and positive examples, rather than only on the minority class at the cost of the majority class. The g-mean is firstly used by Kubat and Matwin in 1997[25], which indicates the geometric mean of the accuracies separately on each of two opposite classes. This measure maximizes the accuracy on each class while keeping these two accuracies balanced. Following is the specific calculation of g-mean.

$$g\text{-}mean = \sqrt{a^+ \times a^-},$$

where a^+ is the accuracy on positive examples, $a^+ = \frac{a}{a+b}$, and a^- is the accuracy on negative examples, $a^- = \frac{d}{c+d}$.

Table 2. Confusion Matrix

		Predicted	
		Positive	Negative
True	Positive	a	b
	Negative	c	d

4.2 Experimental Results

We compare our proposed two algorithms with the cost-sensitive (CS) algorithm and the PA algorithm (which is optimized by using truncated gradient method). Because the sequence that examples arrive slightly influences the experimental results, we show the average value of 20 times testing in random sequence, and four algorithms observe the same sequence every time. We assign the same value of parameters (i.e., η, g, θ, and K) to these four algorithms, as well as the same number of selected features.

In Fig.1, we show the g-mean of six datasets conducted on four algorithms, where each g-mean changes with the number of selected features. And in Fig.2, the measure, specificity, changes with the same situation. We can see from the figure that MBPA and ONPA algorithms always perform better than the other two compared algorithms on the measurement of g-mean and specificity. That is, the two proposed algorithms not only take effect on the classification of both positive and negative examples (indicated by g-mean) but also highly recognize the negative examples of the minor class (indicated by specificity). It demonstrates that our proposed two extended algorithms by modify-

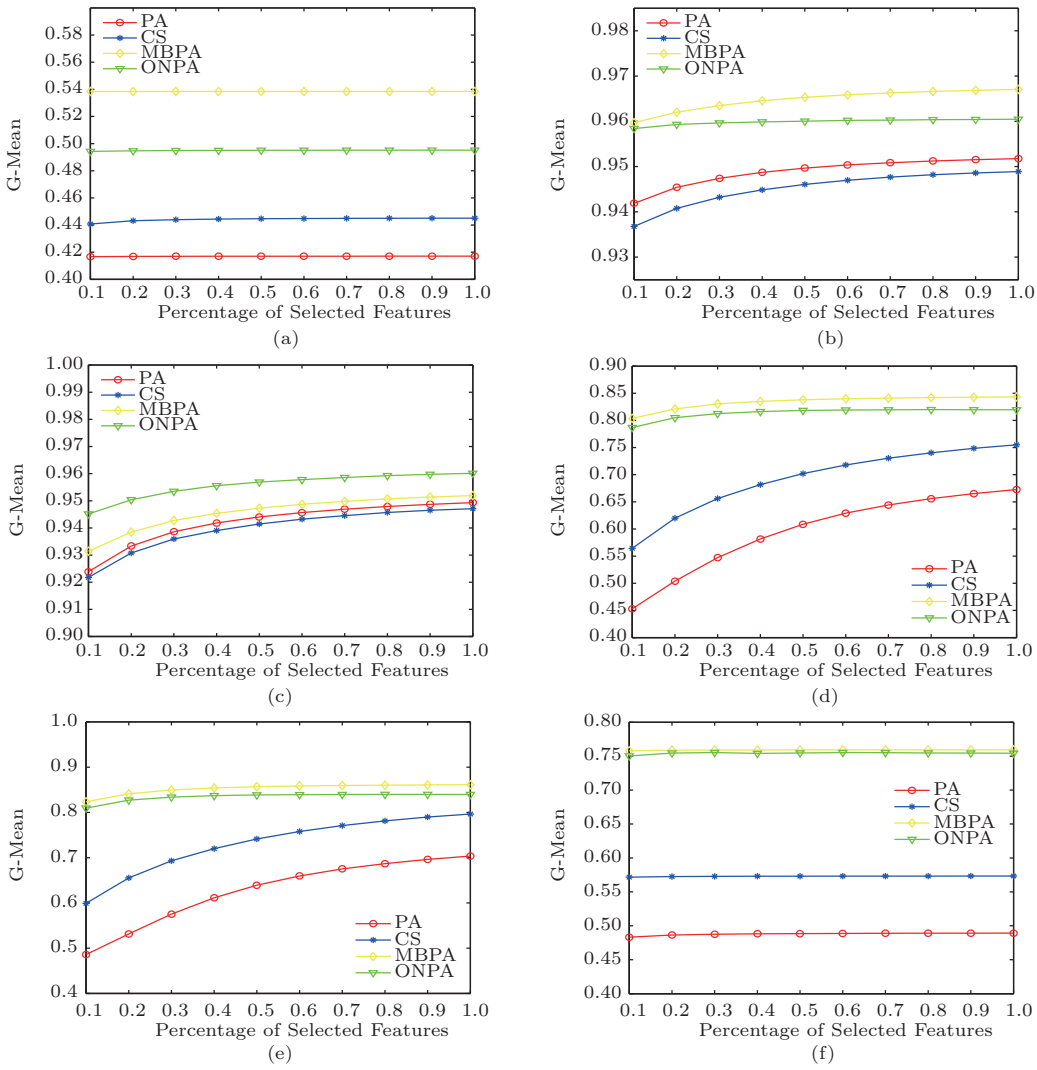


Fig.1. G-mean of four algorithms with respect to the percentage of selected features on six datasets. (a) covtype. (b) mnist. (c) ups. (d) w7a. (e) w8a. (f) farm.

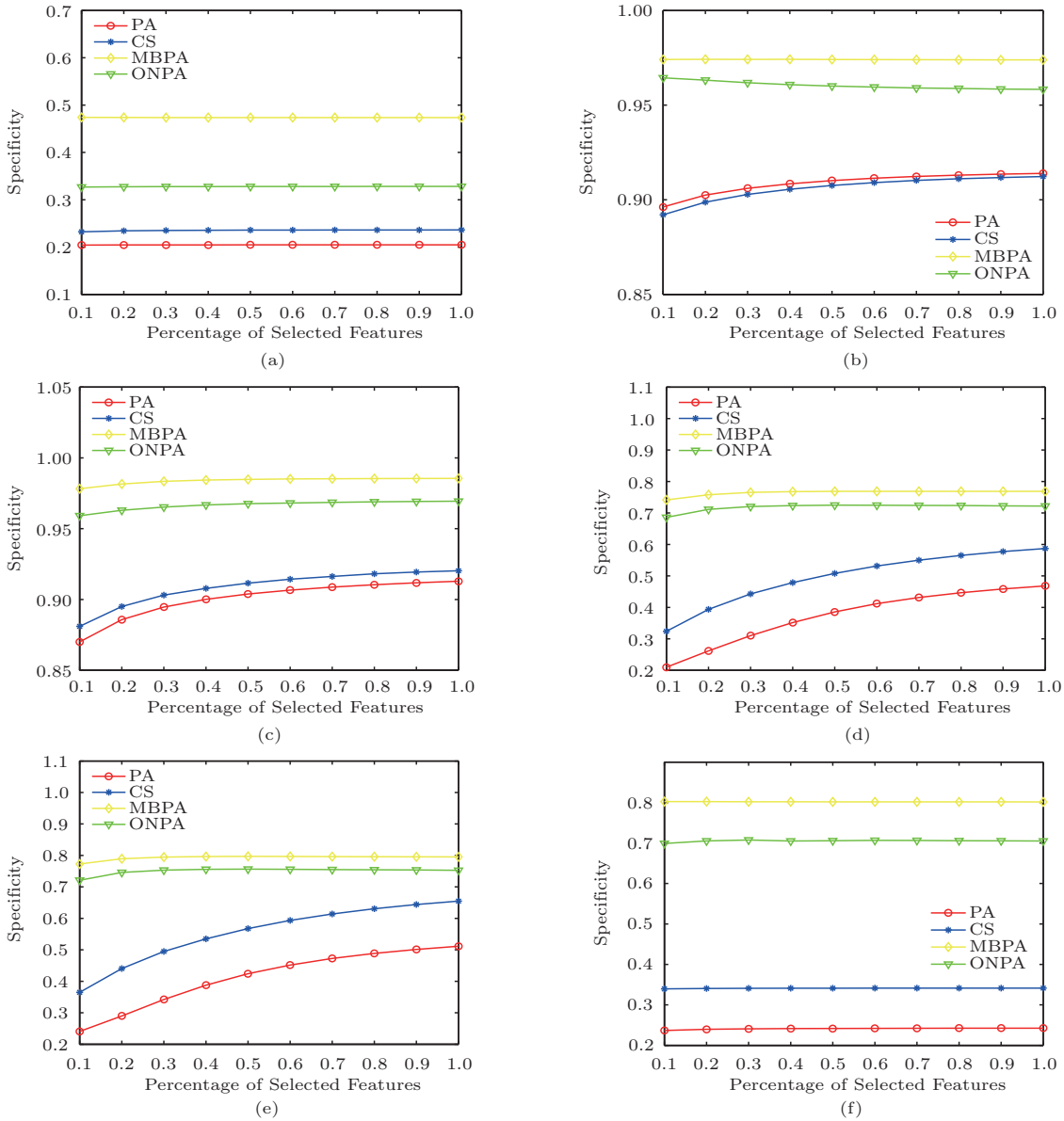


Fig.2. Specificity of four algorithms with respect to the percentage of selected features on six datasets. (a) covtype. (b) mnist. (c) ups. (d) w7a. (e) w8a. (f) farm.

ing the PA algorithm show remarkable improvement on the online feature selection of class imbalance as compared with two baselines. Besides, CS algorithm is better than the optimized PA algorithm for most of the datasets. In general, MBPA algorithm performs better than ONPA algorithm.

As can be seen from the presented two figures, the tendency of two measurements changes almost consistently on all six datasets. On the covtype dataset and the farm dataset, the g-mean and the specificity of four algorithms remain stable as the number of selected features increases. However, on the other four datasets, the g-mean of MBPA and ONPA algorithms grows when the percentage of selected features increases, and then it levels off. The g-mean of CS and PA algorithms consistently goes up along with the number of selected features increasing. For the specificity measurement on the six datasets, our proposed MBPA and ONPA algorithms level out with slight fluctuation, and compared baselines show a steady upward trend when the number of selected features changes. Hence, the MBPA and the ONPA algorithms are not so sensitive to the number of selected features and they can be effective and efficient when the dataset is high-dimensional. It can be seen

from Fig.1 and Fig.2, the g-mean and the specificity of MBPA and ONPA algorithms on the mnist and the ups datasets are over 0.95 even if the number of selected features is small.

5 Conclusions

In this paper, we proposed two algorithms to tackle the problem of online feature selection on imbalanced classification problem based on the PA algorithm. We modified the PA algorithm by using different margin thresholds. We also artificially synthesized negative examples to reduce the imbalance between the majority and the minority classes. Furthermore, we employed the truncated gradient method to simplify the weight vector of the separating hyperplane. This leads to an effective and efficient online learning model that is built from only a small and fixed number of active features for imbalanced classification. Experimental results on real-world datasets showed that the proposed algorithms provide a better performance against the compared baselines.

Some studies remain to be investigated in our future work. The datasets used in the experiments are not large-scale, and this suggests one way to extend our approach. Besides, this paper only focuses on binary classification, and it is interesting to study the online feature selection of class imbalance with multiple classes.

References

[1] Longadge R, Dongre S S, Malik L. Class imbalance problem in data mining: Review. *International Journal of Computer Science and Network*, 2013, 2(1): 1305-1707.

[2] Dash M, Liu H. Feature selection for classification. *Intelligent Data Analysis*, 1997, 1(1/2/3/4): 131–156.

[3] Mladenic D, Grobelnik M. Feature selection for unbalanced class distribution and Naive Bayes. In *Proc. the 16th Int. Conf. Machine Learning*, June 1999, pp.258-267.

[4] Guyon I, Elisseeff A. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 2003, 3: 1157-1182.

[5] Wasikowski M, Chen X. Combating the small sample class imbalance problem using feature selection. *IEEE Trans. Knowl. Data Eng.*, 2010, 22(10): 1388-1400.

[6] Hoi S C H, Wang J, Zhao P, Jin R. Online feature selection for mining big data. In *Proc. the 1st Int. Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, August 2012, pp.93-100.

[7] Wang J, Zhao P, Hoi S C H, Jin R. Online feature selection and its application. *IEEE Trans. Knowl. Data Eng.*, 2014, 26(3): 698-710.

[8] Forman G. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 2003, 3: 1289-1305.

[9] Zheng Z, Wu X, Srihari R K. Feature selection for text categorization on imbalanced data. *ACM SIGKDD Explorations Newsletter*, 2004, 6(1): 80-89.

[10] Chawla N V, Japkowicz N, Kotcz A. Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 2004, 6(1): 1-6.

[11] Langford J, Li L, Zhang T. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 2009, 10: 777-801.

[12] Chawla N V, Bowyer K W, Hall L O, Philip Kegelmeyer W. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 2011, 16(1): 321-357.

[13] Maldonado S, Weber R, Famili F. Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines. *Information Sciences: an International Journal*, 2014, 286: 228-246.

[14] Tax D M J, Duin R P W. Support vector data description. *Machine Learning*, 2004, 54(1): 45–66.

[15] Zhou Z, Liu X. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Trans. Knowl. Data Eng.*, 2006, 18(1): 63–77.

[16] Zhao P, Hoi S C H. Cost-sensitive online active learning with application to malicious URL detection. In *Proc. the 19th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Aug. 2013, pp.919-927.

[17] Wang J, Zhao P, Hoi S C H. Cost-sensitive online classification. *IEEE Trans. Knowl. Data Eng.*, 2014, 26(10): 2425-2438.

[18] Yu L, Liu H. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 2004, 5: 1205-1224.

[19] Chen X, Jeong J C. Minimum reference set based feature selection for small sample classification. In *Proc. the 24th Int. Conf. Machine Learning*, June 2007, pp.153-160.

[20] Wu Q, Ye Y, Zhang H, Ng M K, Ho S S. ForesTexter: An efficient random forest algorithm for imbalanced text categorization. *Knowledge-Based Systems*, 2014, 67: 105-116.

[21] Wu Q, Ye Y, Liu Y, Ng M K. SNP selection and classification of genome-wide SNP data using stratified sampling random forests. *IEEE Trans. Nanobioscience*, 2012, 11(3): 216-227.

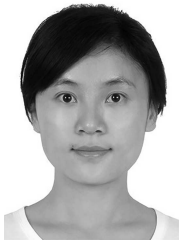
[22] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 1988, 65(6): 386-408.

[23] Freund Y, Schapire R E. Large margin classification using the perceptron algorithm. *Machine Learning*, 1999, 37(3): 277-296.

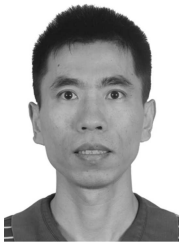
[24] Crammer K, Dekel O, Keshet J, Shalev-Shwartz S, Singer Y. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 2006, 7: 551-585.

[25] Kubat M, Matwin S. Addressing the curse of imbalanced training sets: One-sided selection. In *Proc. 14th Int. Conf. Machine Learning*, April 1997, pp.179-186.

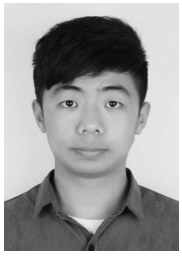
Chao Han received her B.S. degree in computer science and technology from Yunnan University, Kunming, in 2013, and enrolled in the graduate division of the School of Software Engineering at the South China University of Technology, Guangzhou, in 2013 to become a Ph.D. student. Her research interests mainly include data mining, machine learning, social network, and big data processing.



Yun-Kun Tan received her B.S. degree in automatic control, and her M.S. degree in software engineering, in 2011 and 2016, respectively, both from South China University of Technology, Guangzhou. Her research interest lies in data mining and distributed computing.



Jin-Hui Zhu received his B.S. degree in automatic control, his M.S. degree in mechanical design and theory, and his Ph.D. degree in computer science from the South China University of Technology, Guangzhou, in 1999, 2002 and 2010, respectively. Since 2002, he has been with the School of Computer Science and Engineering, the South China University of Technology, where he became an associate professor in 2011. In 2012, he joined the School of Software Engineering, the South China University of Technology. His current research interests include robot software architecture, embedded operating system, machine learning and their applications.



Yong Guo received his B.S. degree from the School of Software Engineering at the South China University of Technology, Guangzhou, in 2016. He became a Ph.D. student in the School of Software Engineering at the South China University of Technology at the same year. His research interests focus on machine learning, computer vision, and data mining.

Jian Chen is currently a professor of the School of Software Engineering (SSE) at South China University of Technology, Guangzhou. She joined SSE at South China University of Technology as a faculty member in 2005. She received her B.S. and Ph.D. degrees, both in computer science, from Sun Yat-sen University, Guangzhou, in 2000 and 2005 respectively. Her research interests can be summarized as developing effective and efficient data analysis techniques for complex data and the related applications. Particularly, she is currently interested in various techniques of data mining, Web search, information retrieval, and recommendation techniques as well as their applications. Her research has been supported in part by the National Natural Science Foundation of China, the Natural Science Foundation of Guangdong Province of China, the Natural Science Key Program of Higher Education Institutions of Guangdong Province, the Fundamental Research Funds for the Central Universities (SCUT), Hewlett-Packard Company (HP), Samsung, etc.

Qing-Yao Wu is currently an associate professor with the School of Software Engineering, South China University of Technology, Guangzhou. He received his B.S. degree in software engineering from the South China University of Technology, and his M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, in 2007, 2009, and 2013, respectively. He was a post-doctoral research fellow with the School of Computer Engineering, Nanyang Technological University, Singapore, from 2014 to 2015. His current research interests include machine learning, data mining, big data research, image processing, and bioinformatics.