IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 3, 2018, accepted October 18, 2018, date of publication October 23, 2018,
date of current version November 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2877701

Speedup Two-Class Supervised Outlier Detection

YUGEN YI[1], WEI ZHOU[2], YANJIAO SHI[3], AND JIANGYAN DAI[4]
[1]School of Software, Jiangxi Normal University, Nanchang 330022, China
[2]College of Information Science and Engineering, Northeastern University, Shenyang 110819, China
[3]School of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai 201418, China
[4]School of Computer Engineering, Weifang University, Weifang 261061, China

Corresponding authors: Wei Zhou (zhouweineu@outlook.com) and Jiangyan Dai (daijyan@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61602221, Grant 61602222, Grant 61806126, Grant 41661083, and Grant 61762050, in part by the Natural Science Foundation of Jiangxi Province under Grant 20171BAB212009, and in part by the Science and Technology Research Project of Jiangxi Provincial Department of Education under Grant GJJ160333.

ABSTRACT Outlier detection is an important topic in the community of data mining and machine learning. In two-class supervised outlier detection, it needs to solve a large quadratic programming whose size is twice the number of samples in the training set. Thus, training two-class supervised outlier detection model is time consuming. In this paper, we show that the result of the two-class supervised outlier detection is determined by minor critical samples which are with nonzero Lagrange multipliers and the critical samples must be located near the boundary of each class. It is much faster to train the two-class supervised outlier detection on the subset which consists of critical samples. We compare three methods which could find boundary samples. The experimental results show that the nearest neighbors distribution is more suitable for finding critical samples for the two-class supervised outlier detection. The two-class supervised novelty detection could become much faster and the performance does not degrade when only critical samples are retained by nearest neighbors' distribution information.

INDEX TERMS Supervised outlier detection, critical sample, nearest neighbors' distribution.

I. INTRODUCTION

In many real applications, minor abnormal samples are more important than normal ones. The abnormal sample is called outlier and the process to find abnormal sample is called outlier detection. In outlier detection, we need to find minor outliers in massive normal samples. Outlier detection has been used in many fields, such as intrusion detection [1], [2], fraud detection [3], medical diagnosis [4], [36], and industrial damage detection [1], [5].

Generally, the outlier is the sample which is not consistent with the majority distribution. Outlier detection research contains two cases: supervised outlier detection and unsupervised outlier detection [6]. In supervised outlier detection, we need to collect many labelled samples. Different from classification problem, most of the labelled samples are normal since it is expensive to collect abnormal samples. When normal samples follow the same distribution, supervised outlier detection is a one-class classification problem which has been researched for several decades. However, it may not hold that all normal samples are consistent with the same distribution in some scenarios. For instance, we need to monitor more than one sensor in industrial fault detection. The signals from each sensor follow an independent distribution. Then, normal samples follow a mixture of two or more independent distributions. It is a two-class or multi-class supervised outlier detection problem. A simple scenario is that there are two normal classes. Each normal class follows an independent distribution. Vilen Jumutc and Suykens extended one-class support vector machine (OC-SVM) [7] for two-class supervised outlier detection [9]. The two-class supervised outlier detection can be converted as a quadratic programming whose size is the twice of the number of training samples. Thus, it costs much more time than OC-SVM.

It is urgent to speed up two-class supervised outlier detection. Fortunately, we find that the result of two-class supervised outlier detection is determined by minor critical samples which are with nonzero Lagrange multipliers. Merely retaining the critical samples, the performance of two-class supervised outlier detection does not degrade. The critical samples must be located near the boundary of each class. Then, we only need to retain a subset consisting of the ones which would be located near the boundary of each class. Therefore, it only needs to solve a smaller optimization programming which is much faster.

The rest of this paper is organized as follows: the related work is reviewed in Section II; a brief review of the two-class supervised outlier detection is summarized in Section III; the method to retain critical samples is introduced in Section IV; the experimental results are reported in Section V; the discussion and conclusions are provided in the last Section.

II. RELATED WORK

According to the existence of the label information, outlier detection can be categorized into two cases: unsupervised outlier detection and supervised outlier detection. In unsupervised outlier detection, each sample is assigned with a score to represent the probability that this sample is an outlier. Then all samples are sorted according to the scores. The outliers are the ones located at the top positions [23]–[25]. We do not have any label information in the unsupervised outlier detection. In supervised outlier detection, the outliers are determined by a model which is learnt from massive labelled samples. When the labelled samples follow the same distribution, it is a one-class classification problem, such as one-class support vector machine (OC-SVM) [7], support vector data description (SVDD) [8], one-class Gaussian Processing [40]. Jumutc and Suykens extended OC-SVM for the normal samples following a mixture of distribution, which means the normal samples could belong to two or more classes [9]. In their method, it needs to solve a big quadratic optimization (QP) which is time-consuming. For instance, when the normal samples belong to two classes, the number of variables in QP is twice of the number of training samples. It is urgent to speed up supervised outlier detection.

In support vector machine (SVM) related works, the result is determined by minor critical samples (called support vectors) which are with nonzero Lagrange multipliers. Training process could become much faster merely retaining the samples would become support vectors. The previous work mainly focuses on support vector classification (SVC) [10]–[14], support vector regression (SVR) [15], [16], and OC-SVM [17]–[19]. The critical samples are located near the decision plane and the boundary of ϵ-tube in SVC and SVR, respectively. It does not hold in supervised outlier detection. In OC-SVM, the critical samples are located near the boundary of the data distribution. Li [18] found boundary samples via extreme points. Zhu *et al.* [17] found boundary samples via neighbors' distribution information. In [19], the relative density degree is used to find useful samples for one-class support vector machine. However, all normal samples must follow the same distribution. In this paper, we try to find critical samples for supervised outlier detection. The research about two-class problem always is the basement of multi-class problem. In this paper, we only consider two-class situation. We trust that our work also can be used in multi-class situation in the future.

III. TWO-CLASS SUPERVISED OUTLIER DETECTION

The symbols used in the whole paper are listed in Table 1.

TABLE 1. Some notations used in our paper.

Notations	Description
$\boldsymbol{X}, \boldsymbol{Y}$	the sets of training samples and labels
l	the number of training samples
$\boldsymbol{X}_1, \boldsymbol{X}_2$	the sets of the samples belonging to class 1, class 2, respectively
l_1, l_2	the size of the set $\boldsymbol{X}_1, \boldsymbol{X}_2$ respectively
$\boldsymbol{x}_i, y_i$	a training sample and the associated label. $\boldsymbol{x}_i \in \mathbb{R}_n$. When $\boldsymbol{x}_i$ belongs to the class 1, $y_i = 1$; otherwise, $y_i = -1$.
$\Phi(\boldsymbol{x})$	the mapping of $\boldsymbol{x}$ in a high-dimensional space.
$K(\boldsymbol{x}_i, \boldsymbol{x}_j)$	the kernel function which is used to calculate the inner product of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in high-dimensional space, such as radial basis function (RBF) kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|/(2\sigma^2))$.
$\boldsymbol{x}_i^j$	one of the k-nearest neighbours of $\boldsymbol{x}_i$, $j = 1, \ldots, k$
$\bar{\boldsymbol{x}}_i$	the mean of the k-nearest neighbours
$d(\boldsymbol{x}_i, \boldsymbol{x}_i^k)$	the distance between $\boldsymbol{x}_i$ and its the k-th nearest neighbour
$kNN(\boldsymbol{x}_i)$	the set consists of the k-nearest neighbours of $\boldsymbol{x}_i$

A. PROBLEM DESCRIPTION

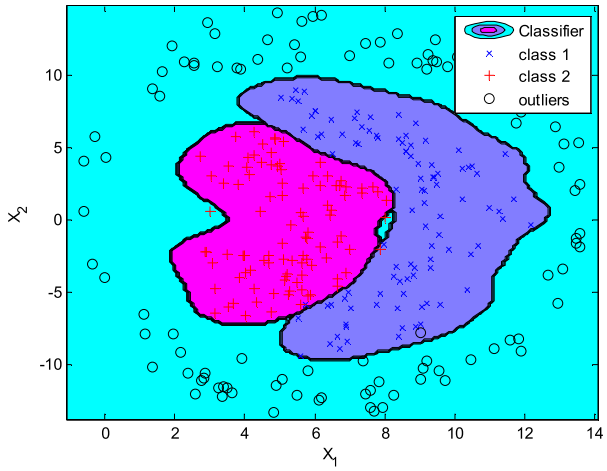
In some real applications, it needs to identify whether an unknown sample is abnormal according to many labelled samples. If all labelled samples follow the same distribution, it is a one-class classification problem. Sometimes, the labeled samples may follow a mixture of distributions. When it is a mixture of two distributions, the problem becomes a two-class supervised outlier detection problem. The aim of two-class supervised outlier detection is to build a data description that can describe all or most of the normal samples and tell us whether an unknown sample is an outlier or which normal class this unknown sample belongs to. In two-class classification, it can only return which class an unknown sample belongs to even it is an outlier. An illustration is shown in Fig. 1.

In Fig. 1 (a), the samples outside of the description are outliers. In Fig. 1 (b), the decision plane cannot distinguish whether an unknown sample is outlier. If we want to detect outliers via two-class classification, we need to learn two models at least. One is to distinguish whether an unknown sample is an outlier. The other is to distinguish which class an unknown sample belongs to if it is not an outlier.

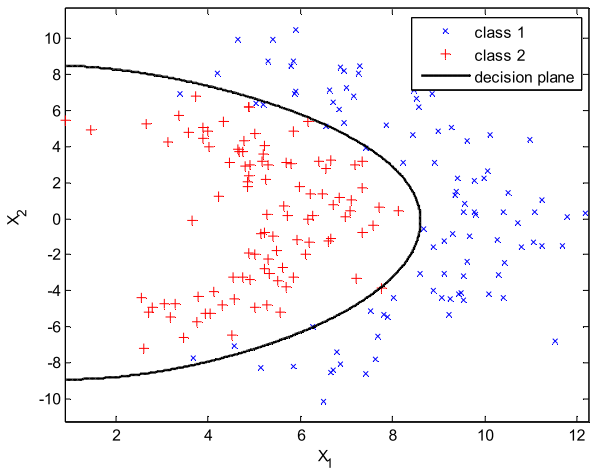
B. A BASIC REVIEW OF TWO-CLASS SUPERVISED OUTLIER DETECTION

The two-class supervised outlier detection needs to find two hyperplanes. Each hyperplane separates the samples in one class from their mappings in the feature space with maximum margin. The angle between two hyperplanes should be as large as possible. A graphical illustration is shown in Fig. 2.

Let $\boldsymbol{f}_{c1} = \langle \boldsymbol{w}_1, \Phi(\boldsymbol{x}) \rangle - \rho$ and $\boldsymbol{f}_{c2} = \langle \boldsymbol{w}_2, \Phi(\boldsymbol{x}) \rangle - \rho$ represent the two hyperplanes, then the two-class supervised



(a)



(b)

FIGURE 1. An illustration of the two-class supervised outlier detection and the two-class classification. The x-marks and pluses belong to class 1 and class 2, respectively. In (a), the circles are outliers and the solid line is the description of two-class supervised outlier detection. In (b), the solid line is the decision plane of the two class classification.

outlier detection can be written as follows:

$$\begin{aligned}\min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\rho_1\rho_2}\ & \frac{\gamma}{2}(\|\boldsymbol{w}_1\|^2+\|\boldsymbol{w}_2\|^2)+C\sum_{i=1}^{l}(\xi_i+\xi_i^*)-\rho_1-\rho_2\\ s.t.\ & y_i(\langle \boldsymbol{w}_1,\Phi(\boldsymbol{x}_i)\rangle-\rho_1)+\xi_i\geq 0, i\in[1,l]\\ & y_i(\langle \boldsymbol{w}_2,\Phi(\boldsymbol{x}_i)\rangle-\rho_2)+\xi_i^*\leq 0, i\in[1,l]\\ & \xi_i\geq 0,\xi_i^*\geq 0, i\in[1,l].\end{aligned}\tag{1}$$

The decision function $c(\boldsymbol{x})$ is defined as follows:

$$c(\boldsymbol{x})=\begin{cases}\arg\max\limits_{ci} f_{ci}(\boldsymbol{x}), & if\ \ \max f_{ci}(\boldsymbol{x})>0\\ outlier, & otherwise\end{cases}\tag{2}$$

where, ci is the index of the ci-th hyperplane. If $ci=c1$, $\boldsymbol{x}$ belongs to class 1; if $ci=c2$, $\boldsymbol{x}$ belongs to class 2.

Introducing $\alpha_i\geq 0$, $\lambda_i\geq 0$, $\beta_i\geq 0$, and $\beta_i^*\geq 0$ as the Lagrange multipliers for the constraints, the Lagrangian function of Eq. (1) can be written as follows:

$$\begin{aligned}&L(\boldsymbol{w}_1,\boldsymbol{w}_2,\rho_1,\rho_2,\boldsymbol{\xi},\boldsymbol{\xi}^*,\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\beta}^*)\\ &=\frac{\gamma}{2}(\|\boldsymbol{w}_1\|^2+\|\boldsymbol{w}_2\|^2)+C\sum_{i=1}^{l}(\xi_i+\xi_i^*)-\rho_1-\rho_2\\ &\quad-\sum_{i=1}^{l}\alpha_i(y_i(\langle \boldsymbol{w}_1,\Phi(\boldsymbol{x}_i)\rangle-\rho_1)+\xi_i)\\ &\quad+\sum_{i=1}^{l}\lambda_i(y_i(\langle \boldsymbol{w}_2,\Phi(\boldsymbol{x}_i)\rangle-\rho_2)+\xi_i^*)\\ &\quad-\sum_{i=1}^{l}\beta_i\xi_i-\sum_{i=1}^{l}\beta_i^*\xi_i^*\end{aligned}\tag{3}$$

where $\boldsymbol{\xi}$,$\boldsymbol{\xi}^*$ and $\boldsymbol{\alpha}$,$\boldsymbol{\lambda}$,$\boldsymbol{\beta}$,$\boldsymbol{\beta}^*$ are the vectors form of slack variables and Lagrange multipliers, respectively. Setting the derivatives of Eq. (3) with respect to $\boldsymbol{w}_1$,$\boldsymbol{w}_2$,$\boldsymbol{\xi}$,$\boldsymbol{\xi}^*$,ρ_1,ρ_2 to zeros, then

$$\boldsymbol{w}_1=\frac{\gamma\sum_{i=1}^{l}\alpha_i y_i\Phi(\boldsymbol{x}_i)+\sum_{i=1}^{l}\lambda_i y_i\Phi(\boldsymbol{x}_i)}{\gamma^2-1}\tag{4}$$

$$\boldsymbol{w}_2=\frac{\gamma\sum_{i=1}^{l}\lambda_i y_i\Phi(\boldsymbol{x}_i)+\sum_{i=1}^{l}\alpha_i y_i\Phi(\boldsymbol{x}_i)}{1-\gamma^2}\tag{5}$$

$$C-\beta_i-\alpha_i=0\tag{6}$$

$$C-\beta_i^*-\lambda_i=0\tag{7}$$

$$\sum_{i=1}^{l}\alpha_i y_i=1\tag{8}$$

$$\sum_{i=1}^{l}\lambda_i y_i=-1\tag{9}$$

Substituting Eqs. (4-9) into Eq. (3), the dual form of Eq. (1) can be written as follows:

$$\begin{aligned}\min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\rho_1\rho_2}\ & \frac{\mu_1}{2}(\boldsymbol{\alpha}^T\boldsymbol{G}\boldsymbol{\alpha}+\boldsymbol{\lambda}^T\boldsymbol{G}\boldsymbol{\lambda})+\mu_2\boldsymbol{\alpha}^T\boldsymbol{G}\boldsymbol{\lambda}\\ s.t.\ & 0\leq\alpha_i\leq C,\quad i\in[1,l]\\ & 0\leq\lambda_i\leq C,\quad i\in[1,l]\\ & \boldsymbol{y}^T\boldsymbol{\alpha}=1,\boldsymbol{y}^T\boldsymbol{\lambda}=1.\end{aligned}\tag{10}$$

where $\mu_1=\frac{\gamma}{1-\gamma^2}$, $\mu_2=\frac{1}{\gamma^2-1}$, $\boldsymbol{y}$ is the vector form of labels, $\boldsymbol{G}$ is a $l\times l$ matrix and $\boldsymbol{G}=\boldsymbol{K}\circ\boldsymbol{y}\boldsymbol{y}^T$ where $\boldsymbol{K}$ is the kernel matrix and $\boldsymbol{K}(i,j)=K(\boldsymbol{x}_i,\boldsymbol{x}_j)$, $\circ$ is component-wise multiplication. The $f_{c1}(\boldsymbol{x})$ and $f_{c2}(\boldsymbol{x})$ can be represented as follows:

$$f_{c1}=\frac{\gamma\sum_{i=1}^{l}\alpha_i y_i K(\boldsymbol{x}_i,\boldsymbol{x})+\sum_{i=1}^{l}\lambda_i y_i K(\boldsymbol{x}_i,\boldsymbol{x})}{\gamma^2-1}-\rho_1.\tag{11}$$

$$f_{c2}=\frac{\gamma\sum_{i=1}^{l}\lambda_i y_i K(\boldsymbol{x}_i,\boldsymbol{x})+\sum_{i=1}^{l}\alpha_i y_i K(\boldsymbol{x}_i,\boldsymbol{x})}{1-\gamma^2}-\rho_2.\tag{12}$$

IV. SELECTING CRITICAL SAMPLES FOR TWO-CLASS SUPERVISED OUTLIER DETECTION

Obviously, only the sample with nonzero Lagrange multipliers are critical to the hyperplanes, $f_{c1}(\boldsymbol{x})$ and

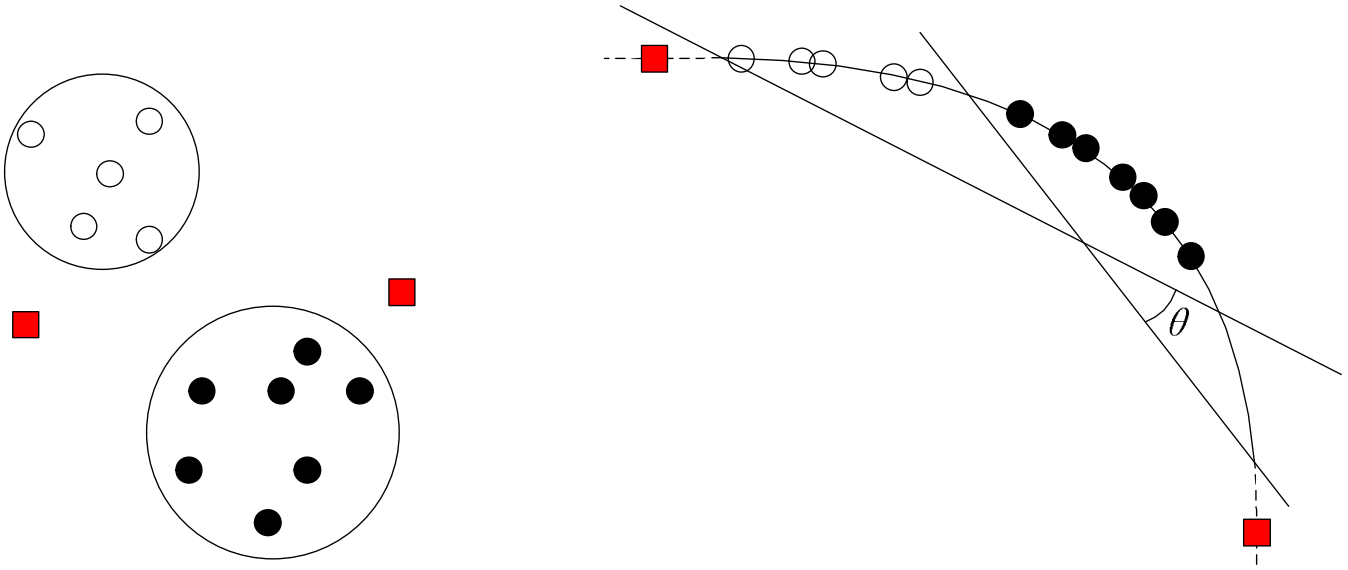


FIGURE 2. An explanation of the two-class supervised outlier detection. Left: the samples in the original space; right: the samples in the feature space.

$f_{c2}(\boldsymbol{x})$ ($\alpha_i \neq 0$ or $\lambda_i \neq 0$). The learning result of two-class supervised outlier detection would not change merely retaining the samples which would be with nonzero Lagrange multipliers. The scale of Eq. (10) would become much smaller if we can find those critical samples. Thus, it can train two-class supervised outlier detection on a small retained subset, which is much faster. Then, speeding up two-class supervised outlier detection is converted as finding critical samples before learning. The following proposition illustrates how to find critical samples.

Proposition 1: The critical samples in two-class supervised outlier detection must be located near the boundary of each class.

Proof: Let $\boldsymbol{x}_i$ be a sample in the training set. In the feature space, the distances between $\boldsymbol{x}_i$ and the hyperplanes, $f_{c1}(\boldsymbol{x})$ and $f_{c2}(\boldsymbol{x})$, are $\frac{|(\langle \boldsymbol{w}_1, \Phi(\boldsymbol{x}_i)\rangle - \rho_1|}{\|\boldsymbol{w}_1\|^2}$ and $\frac{|(\langle \boldsymbol{w}_2, \Phi(\boldsymbol{x}_i)\rangle - \rho_2|}{\|\boldsymbol{w}_2\|^2}$, respectively. The constraints $y_i(\langle \boldsymbol{w}_1, \Phi(\boldsymbol{x}_i)\rangle - \rho_1) + \xi_i$ and $y_i(\langle \boldsymbol{w}_2, \Phi(\boldsymbol{x}_i)\rangle - \rho_2) + \xi_i^*$ can be rewritten as

$$\frac{y_i(\langle \boldsymbol{w}_1, \Phi(\boldsymbol{x}_i)\rangle - \rho_1)}{\|\boldsymbol{w}_1\|^2} + \frac{\xi_i}{\|\boldsymbol{w}_1\|^2}$$
$$\frac{y_i(\langle \boldsymbol{w}_2, \Phi(\boldsymbol{x}_i)\rangle - \rho_2)}{\|\boldsymbol{w}_2\|^2} + \frac{\xi_i^*}{\|\boldsymbol{w}_2\|^2}$$

The critical samples in two-class supervised outlier detection contain two cases: the sample with nonzero α_i in class 1 and the sample with nonzero λ_i in class 2.

Case 1:

In class 1, $y_i = 1$, $\frac{y_i(\langle \boldsymbol{w}_1, \Phi(\boldsymbol{x}_i)\rangle - \rho_1)}{\|\boldsymbol{w}_1\|^2} + \frac{\xi_i}{\|\boldsymbol{w}_1\|^2} \geq 0$, and the corresponding KTT condition can be rewritten as $\alpha_i(\frac{y_i(\langle \boldsymbol{w}_1, \Phi(\boldsymbol{x}_i)\rangle - \rho_1)}{\|\boldsymbol{w}_1\|^2} + \frac{\xi_i}{\|\boldsymbol{w}_1\|^2}) = 0$. If $\alpha_i \neq 0$, it must hold that $\frac{y_i(\langle \boldsymbol{w}_1, \Phi(\boldsymbol{x}_i)\rangle - \rho_1)}{\|\boldsymbol{w}_1\|^2} + \frac{\xi_i}{\|\boldsymbol{w}_1\|^2} = 0$. Since $\xi_i \geq 0$, the $\frac{(\langle \boldsymbol{w}_1, \Phi(\boldsymbol{x}_i)\rangle - \rho_1)}{\|\boldsymbol{w}_1\|^2}$ must be as small as possible. When $\xi_i = 0$, the $\boldsymbol{x}_i$ is just on the hyperplane $f_{c1}(\boldsymbol{x})$; when $\xi_i > 0$, the $\boldsymbol{x}_i$ is in the opposite side of the hyperplane $f_{c1}(\boldsymbol{x})$. When $\frac{(\langle \boldsymbol{w}_1, \Phi(\boldsymbol{x}_i)\rangle - \rho_1)}{\|\boldsymbol{w}_1\|^2} \geq 0$, it is just the distance to hyperplane $f_{c1}(\boldsymbol{x})$; otherwise, its absolute value is the distance. Therefore, $\boldsymbol{x}_i$ must locate near the boundary of class 1. That is to say, the critical samples with nonzero α_i in class 1 are located near the boundary of the distribution of class 1.

Case 2:

In class 2, $y_i = -1$, $\frac{y_i(\langle \boldsymbol{w}_2, \Phi(\boldsymbol{x}_i)\rangle - \rho_2)}{\|\boldsymbol{w}_2\|^2} + \frac{\xi_i^*}{\|\boldsymbol{w}_2\|^2} \geq 0$, and the corresponding to KTT condition can be rewritten as $\lambda_i(\frac{y_i(\langle \boldsymbol{w}_2, \Phi(\boldsymbol{x}_i)\rangle - \rho_2)}{\|\boldsymbol{w}_2\|^2} + \frac{\xi_i^*}{\|\boldsymbol{w}_2\|^2}) = 0$. Then, similar to the analysis of class 1, the critical samples with nonzero λ_i in class 2 are located near the boundary of the distribution of class 2.

From case 1 and case 2, the Proposition 1 holds. □

Now, we need to find the samples which are located near the boundary of each class. We choose nearest neighbors' distribution [17], relative density degree [18], [38], and local geometry information [19] to find boundary samples for two-class supervised outlier detection. In these finding boundary samples methods, every sample is assigned a score. Then, all samples are sorted by the scores. The boundary samples are located at the top positions. We need to retain those samples at the top positions. The procedure for finding critical samples is described in the following algorithm.

In Steps 1-3 of the Algorithm 1, we find the samples near the boundary of class 1, and in Steps 4-6, we find the samples near the boundary of class 2, respectively. In the last Step, we output a subset of the original training set, which is much smaller than the original set. It would become much faster to learn a two-class supervised outlier detection model on $\{\boldsymbol{X}', \boldsymbol{Y}'\}$.

In Step 2 and Step 5, the scores could be calculated by a boundary detection method, such as nearest neighbors' distribution [17], relative density degree [19], local geometry information [18]. In order to ensure the integrity of this paper, we recap nearest neighbors' distribution, relative

Algorithm 1 Procedure for Finding Critical Samples

Input:

training set $\{\boldsymbol{X}, \boldsymbol{Y}\}$;

the number of nearest neighbours k;

Output:

a subset $\{\boldsymbol{X}', \boldsymbol{Y}'\}$;

1: find the set $\boldsymbol{X}_1$;

2: calculate the scores of all samples in $\boldsymbol{X}_1$ via a boundary detection method and sort the scores in descending (ascending) order;

3: retain the top $\tau * l_1$ $(0 < \tau < 1)$ samples to construct $\boldsymbol{X}'_1$. The corresponding label, $\boldsymbol{Y}'_1$, is $\boldsymbol{e}$ with length $\tau * l_1$;

4: find the set $\boldsymbol{X}_2$;

5: calculate the scores of all samples in $\boldsymbol{X}_2$ via a boundary detection method and sort the scores in descending (ascending) order;

6: retain the top $\tau * l_2$ samples to construct $\boldsymbol{X}'_2$. The corresponding label, $\boldsymbol{Y}'_2$, is $-1 * \boldsymbol{e}$ with length $\tau * l_2$;

7: **return** $\{\boldsymbol{X}', \boldsymbol{Y}'\}$ where $\boldsymbol{X}' = \boldsymbol{X}'_1 \cup \boldsymbol{X}'_2$ and $\boldsymbol{Y}' = \boldsymbol{Y}'_1 \cup \boldsymbol{Y}'_2$;

density degree, and local geometry information as follows. Zhu *et al.* [17] pointed out that a sample's location ($\boldsymbol{x}_i$) in the dataset is related to the nearest neighbors' distribution ($kNN(\boldsymbol{x}_i)$). The k-nearest neighbors is enclosed by a hypersphere with center which is itself and radius which is the distance between the k-th nearest neighbor and itself ($d(\boldsymbol{x}_i, \boldsymbol{x}_i^k)$). Then, the hypersphere is divided by a hyperplane which is perpendicular to the difference between this sample and the mean of k-nearest neighbors ($\bar{\boldsymbol{x}}_i$). Then, the distribution of the nearest neighbors has the following properties.

Property 1: The number of nearest neighbors in the part which $\bar{\boldsymbol{x}}_i$ is located must be more than that in the other one. The difference of the numbers is related to the sample's location. The closer to the boundary the sample is, the larger the difference is.

Property 2: The sum of the cosine of the sample-neighbor angles majorly ranges in [0,k].

Here the sample-neighbor angle is defined as follows.

Definition 1 (Sample-Neighbor Angle [17]): Let θ_i^j be the sample-neighbor angle. The θ_i^j is the angle between $\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i$ and $\boldsymbol{x}_i - \boldsymbol{x}_i^j$.

From Property 1 and Property 2, it is obtained that the location of a sample can be reflected by the cosine sum of the sample-neighbor angles. The cosine sum can be represented as follows.

$$c^{sum}(\boldsymbol{x}_i) = \sum_{j=1}^{k} \cos\theta_i^j = \sum_{j=1}^{k} \frac{\langle \boldsymbol{x}_i - \bar{\boldsymbol{x}}_i, \boldsymbol{x}_i - \boldsymbol{x}_i^j \rangle}{\|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k\| \|\boldsymbol{x}_i - \boldsymbol{x}_i^j\|}. \tag{13}$$

By introducing kernel trick, Eq. (13) can be rewritten as the following Eq. (14), as shown at the bottom of this page.

The larger the cosine sum is, the closer to the boundary the samples is. For Algorithm 1, we only need to retain the samples with large cosine sum.

The relative density degree is used to reflect how dense around a sample. Generally, the relative density degree of a sample near the boundary is smaller than that of a sample within the distribution. The relative density degree can be estimated by k-nearest neighbor [38] or Parzen window [39]. Let ρ_i represent the relative density degree of the sample $\boldsymbol{x}_i$. Then, ρ_i could be estimated via the following equation.

$$\rho_i = \exp\{w' \times \frac{Mean_{kNN}^k}{d(\boldsymbol{x}_i, \boldsymbol{x}_i^k)}\} \tag{15}$$

where w' is a weight factor ($0 \leq w' \leq 1$) and $Mean_{kNN}^k$ is the mean distances between the sample and its k-th nearest neighbor s ($Mean_{kNN}^k = \frac{1}{N}\sum_{i=1}^{N} d(\boldsymbol{x}_i, \boldsymbol{x}_i^k)$). For Algorithm 1, we only need to retain the samples with small relative density degree.

Li [18] pointed out that the boundary sample is related to its local geometrical statistical information. Let all samples be enclosed by a or some surface(s) and the tangent plane is drawn at a tangent to the surface. Then, the boundary sample should be crossed the surface and its tangent plane be located at the edge of the surface. When the surface is convex, all nearest neighbors are located on the opposite side of the tangent plane, as shown in Fig. 3 (a); when the surface is concave, most of nearest neighbors are located on one side of the tangent plane, as shown in Fig. 3 (b) and (c). The ratio is determined by the curvature of the surface.

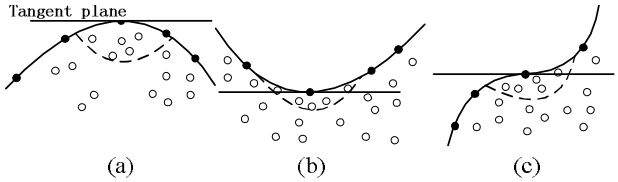


FIGURE 3. The illustration of edge sample. The solid circles are edge samples. The solid curve is the class surface. The straight line is the tangent plane. The region formed by nearest neighbors are circled by a dashed line.

Let $\boldsymbol{v}_{i,j}$ $(j = 1, \ldots k)$ represent the difference between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i,j}$ ($\boldsymbol{v}_{i,j} = \boldsymbol{x}_i^j - \boldsymbol{x}_i$), $\boldsymbol{v}_{i,j}^n$ represent the normalization of $\boldsymbol{v}_{i,j}$, and $\boldsymbol{v}_i^n$ represent the sum of $\boldsymbol{v}_{i,j}^n$ ($\boldsymbol{v}_i^n = \sum_{j=1}^{k} \boldsymbol{v}_{i,j}^n$). Then, if the nearest neighbor is located at the side which the normal vector points, the angle between $\boldsymbol{x}_{i,j}$ and $\boldsymbol{v}_i^n$ is in the range $[0, \pi/2]$; otherwise, the angle is the range $[0, \pi]$. The boundary sample

$$c^{sum}(\boldsymbol{x}_i) = \sum_{j=1}^{k} \frac{K(\boldsymbol{x}_i, \boldsymbol{x}_i) - K(\boldsymbol{x}_i, \boldsymbol{x}_i^j) - \frac{1}{k}\sum_{p=1}^{k} K(\boldsymbol{x}_i, \boldsymbol{x}_i^p) + \frac{1}{k}\sum_{p=1}^{k} K(\boldsymbol{x}_i^j, \boldsymbol{x}_i^p)}{\sqrt{K(\boldsymbol{x}_i, \boldsymbol{x}_i) - \frac{2}{k}\sum_{p=1}^{k} K(\boldsymbol{x}_i, \boldsymbol{x}_i^p) + \frac{1}{k^2}\sum_{p,q=1}^{k} K(\boldsymbol{x}_i^p, \boldsymbol{x}_i^q)}\sqrt{K(\boldsymbol{x}_i, \boldsymbol{x}_i) - 2K(\boldsymbol{x}_i, \boldsymbol{x}_i^j) + K(\boldsymbol{x}_i^j, \boldsymbol{x}_i^j)}}. \tag{14}$$

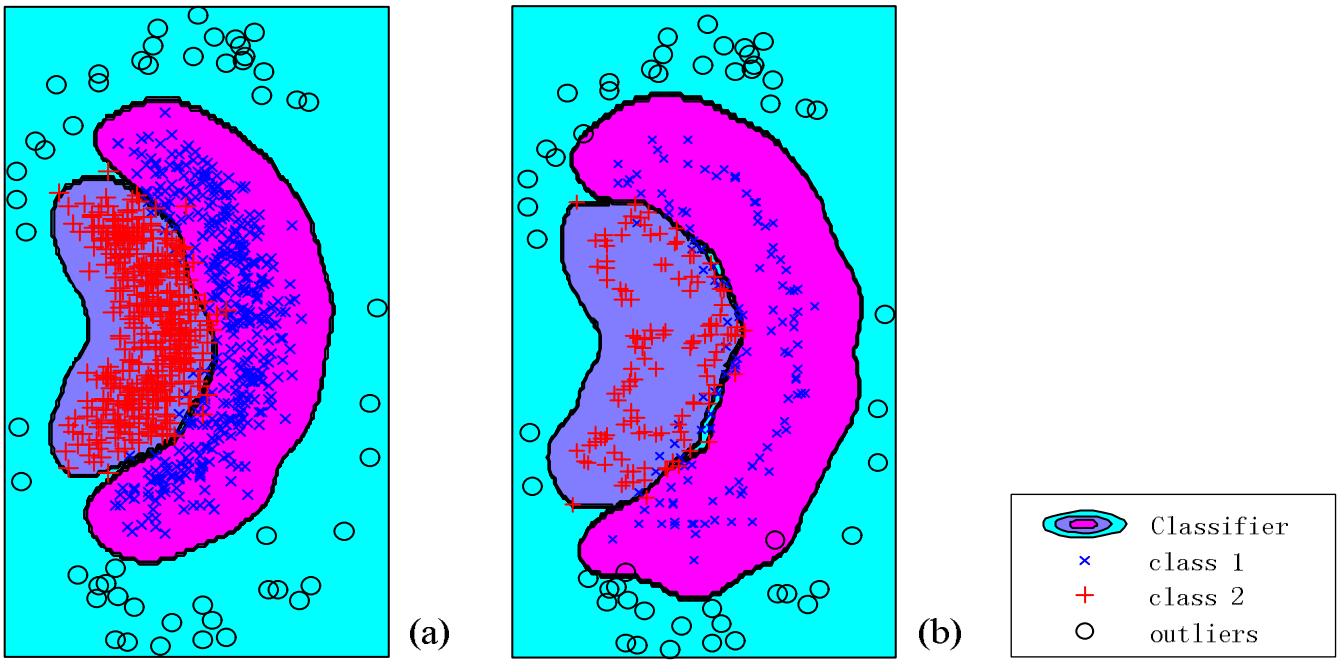


FIGURE 4. The pluses and x-marks belong to class 1 and class 2, respectively. The circles are outliers. The solid lines are the description to represent normal samples. (a) the description is learnt on whole set; (b) the description is learnt on selected subset.

could be found by the following equation.

$$L_i = \frac{1}{k}\sum_{j=1}^{k}(\boldsymbol{v}_{i,j}^T \boldsymbol{v}_i^n \geq 0). \tag{16}$$

For Fig. 3 (a), all nearest neighbors are with $\boldsymbol{v}_{i,j}^T \boldsymbol{v}_i^n \geq 0$, Eq. (16) is equal to 1; for Fig. 3 (b) and (c), most of nearest neighbors are with $\boldsymbol{v}_{i,j}^T \boldsymbol{v}_i^n \geq 0$, Eq. (16) is close to 1. Then, we only need to retain the samples with large values of Eq. (16) in Algorithm 1.

In Fig. 4 (a) and (b), the solid lines are the descriptions learnt on the original training set and reserved subset, respectively. The subset is selected by nearest neighbors' distribution.

From Fig. 4, it can be found that most of the retained samples locate near the boundary of the data distribution and the classifier learnt on selected subset is very close to that learnt on original set.

V. EXPERIMENTS AND SIMULATIONS

In this section, we verify the proposed method for two-class supervised outlier detection. We implement the proposed method via mex interface in matlab environment. The two-class supervised outlier detection is implemented via the Ipopt package (refer to [20]). All experiments are run on a laptop with Ubuntu 14.04 system, 8GB memory, and Intel® Core™ i5-6200U CPU. The radial basis function (RBF) is used as the kernel function. We compare the two-class supervised outlier detection learnt on the original training set and retained subsets in terms of running time, misclassification error, and outlier detection rate. The 'whole set' means that the two-class supervised outlier detection is learnt on the original training set, whilst the 'retained subset' means that the two-class supervised outlier detection is learnt on the subset retained by the proposed method. We compare three boundary detection method in our method, nearest neighbors' distribution (short for NND),relative density degree (short for NND), and local geometrical information (short for LGI).

First, we evaluate the performance on 5 benchmark datasets from the University of California at Irvine (UCI) machine learning repository [21]. Second, we evaluate the performance on 2 artificial synthetic datasets. The artificial synthetic datasets are generated by prtools [22]. Each one contains two dimensions, thus the description can be easily visualized.

A. EXPERIMENTS ON BENCHMARK DATASETS

In this subsection, we select 5 benchmark datasets which are from the University of California at Irvine (UCI) machine learning repository [21] to verify the proposed method. The detailed description of these datasets is listed in Table 2. The second column represents the number of dimensions. The third column represents the size of whole set. The numbers in parentheses are the sizes of each class. The dimensions are in the range 4-180 and the number of samples is in the range 391-5000. Since the two-class supervised outlier detection code is implemented by Ipopt which needs to store the whole kernel matrix, it is difficult to store whole kernel matrix on a personal computer when the training set contains more than 5000 samples.

The datasets are reorganized to suit for evaluating the two-class supervised outlier detection. In svmguide 2 and balance,

TABLE 2. The details of the benchmark datasets.

Datasets	#Dimensions	#Samples (classes)
svmguide2	20	391(221,117,53)
balance	4	625(288,288,49)
segment	19	2310(330,330,330,330,330,330,330)
Abalone	8	4177(2649, 1342,1307)
Waveform	21	5000(1696,1657,1647)

the samples in class 1 and class 2 are used as normal samples and the samples in class 3 are used as abnormal samples. In segment, the samples in class 1 and class 2 are used as normal samples and others are used as abnormal samples. The abalone and waveform are converted as three two-class supervised outlier detection. Two classes are regarded as normal ones and the rest one is used as anormal, denoted as Abalone (12VS3), Abalone (13VS2), and Abalone (23VS1) for Abalone, Waveform (12VS3), Waveform (13VS2), and Waveform (13VS2) for Waveform. The classes before 'VS' are used as the normal class.

The normal samples are equally divided into two parts. One part is used as training samples, the other part and the abnormal samples are used as test samples. The parameter k is set to 20 in NND, RDD, and LGI. The parameter τ is set to 0.2 which means 20% of the whole set is retained. The RBF is used as the kernel function and the width is chosen among $\{2^{-5},2^{-4},2^{-3},2^{-2},2^{-1},2^{0},2^{1},2^{2},2^{3},2^{4},2^{5}\}$. Both parameters γ and C in two-class supervised outlier detection are chosen among $\{2^{-1},2^{0},2^{1},2^{2},2^{3},\ 2^{4},2^{5},2^{6},2^{7},2^{8},2^{9},2^{10}\}$. The parameters are tuned to obtain the least misclassification error via grid search.

The running time comparison is listed in Table 3. For the proposed method, the running time contains two parts: the preprocessing time and training time. The preprocessing time is the one to retain critical samples. The training time is the one to learn two-class supervised novelty detection model on the retained subset. The boundary detection methods include NND, RDD and LGI. Even summing the preprocessing time and training time on the retained subset, it is still much faster than training two-class supervised outlier detection on the

TABLE 3. The time comparison of the benchmark datasets.

Datasets	retained subset						whole set (s)
	NND		RDD		LGI		
	preprocess time (s)	training time (s)	preprocess time (s)	training time (s)	preprocess time (s)	training time (s)	
svmguide2	0.0218	0.0912	0.0218	0.0895	0.0212	0.0956	0.4516
balance	0.0267	0.1502	0.0256	0.1446	0.0279	0.1501	1.3133
segment	0.0307	0.4107	0.0284	0.4217	0.0320	0.4202	2.7103
Abalone(12VS3)	0.1831	1.2787	0.1768	1.2761	0.1869	1.3299	11.8881
Abalone(13VS2)	0.2019	0.6817	0.2014	0.6611	0.1823	0.6607	10.9854
Abalone(23VS1)	0.2150	0.7887	0.2237	0.7572	0.2246	0.7611	17.0243
Waveform(12VS3)	0.3931	0.8002	0.4079	0.7804	0.4099	0.7774	14.1219
Waveform(13VS2)	0.4042	0.7972	0.3659	0.8239	0.4141	0.7848	14.0515
Waveform(23VS1)	0.3994	0.9228	0.3950	0.9579	0.3806	0.8881	14.6622

TABLE 4. The misclassification error comparison of the benchmark datasets.

Datasets	retained subset			whole set (%)
	NND (%)	RDD (%)	LGI (%)	
svmguide2	14.91	14.92	14.93	15.01
balance	5.13	5.09	5.28	5.11
segment	2.61	2.63	2.68	2.71
Abalone(12VS3)	31.03	30.83	31.54	30.95
Abalone(13VS2)	16.57	16.49	17.01	16.46
Abalone(23VS1)	36.14	36.32	36.41	36.21
Waveform(12VS3)	10.04	10.12	10.34	9.97
Waveform(13VS2)	8.17	8.1	8.17	8.31
Waveform(23VS1)	7.67	7.76	7.92	7.73
Avg.	14.7	14.69	14.92	14.72

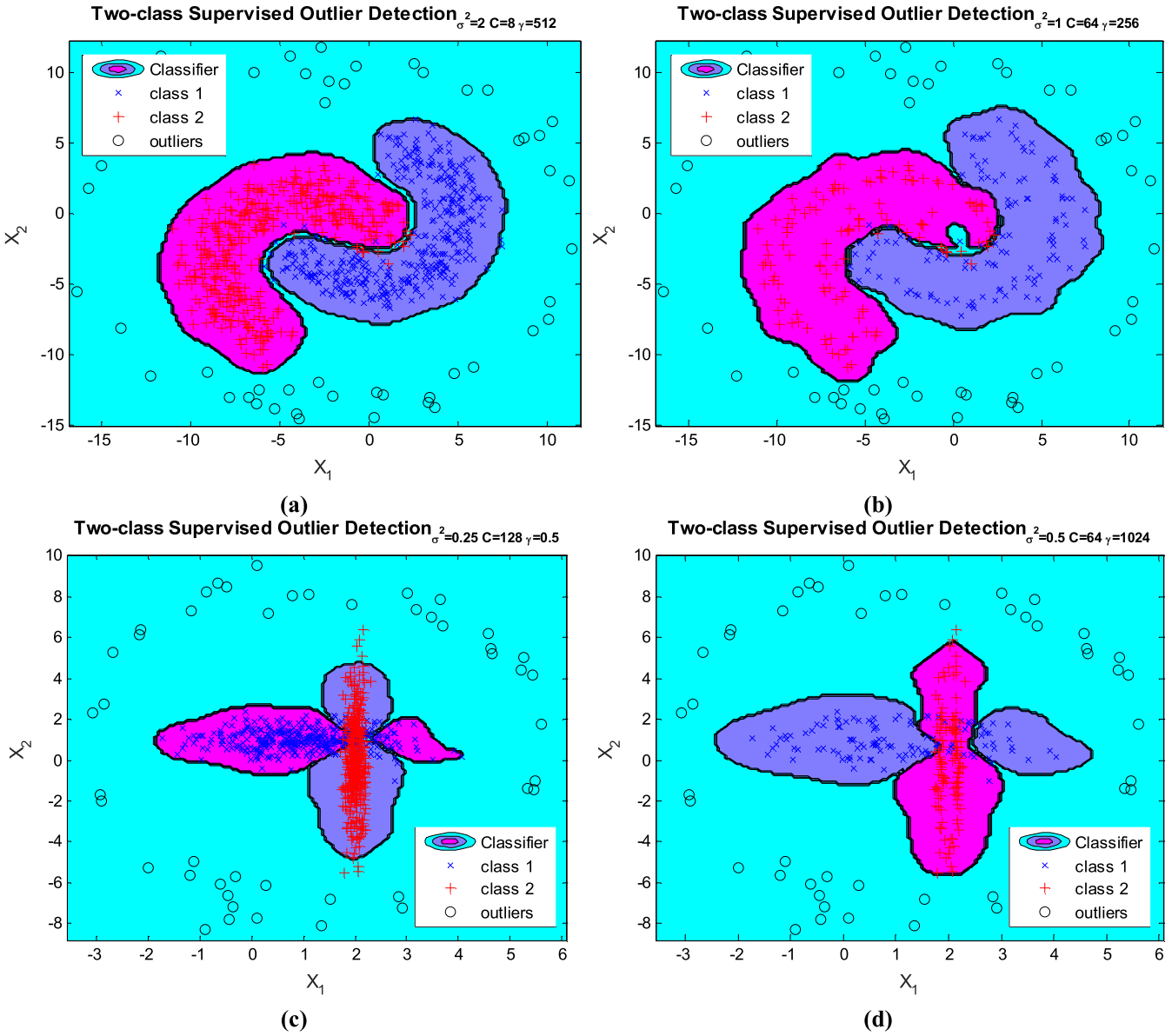


FIGURE 5. The description learnt on artificial synthetic datasets. (a) is learnt on whole set of banana shaped distribution; (b) is learnt on the selected subset of banana shaped distribution; (c) is learnt on whole set of Highleyman shaped distribution; (d) is learnt on the selected subset of Highleyman shaped distribution.

whole set directly. For instance, in svmguide 2, it costs 0.113, 0.112, and 0.117 seconds in all for subset retained by NND, RDD, and LGI respectively, and costs 0.4516 seconds for the whole set. It is nearly 4 times faster than training on whole set for svmguide 2. In waveform, it is nearly 11 times faster than whole set. The consumption time of NND, RDD, and LGI is very close.

It only makes sense to increase speed if the performance is not degraded. In Table 4 and Table 5, we list the misclassification error and outlier detection respectively. It can be found that when we use LGI to retain critical samples the difference to the whole set is the largest and NND is the closest one in three methods. For instance, the average outlier detection rate is 85.2%, 84.86%, and 82.78% when the retained subset is selected by NND, RDD, and LGI respectively. The outlier detection rate of the whole set is 85.34%. The difference is 0.14%, 0.48%, and 2.56% for NND, RDD and LGI. In LGI, the value of Eq. (16) ranges in $[0, 1]$ step by $\frac{1}{k}$. It may exist that many samples have the same value, therefore some boundary samples cannot be found. In RDD, $Mean^k_{kNN}$ is the global information which may be influenced by noise. Obviously, NND is more suitable than RDD and LGI in Algorithm 1.

B. EXPERIMENTS ON ARTIFICIAL SYNTHETIC DATASETS

In this subsection, we evaluate 2 artificial synthetic datasets which are generated by prtools [22]. The first one is the banana shaped distribution and the second one is Highleyman

TABLE 5. The outlier detection rate comparison of the benchmark datasets.

Datasets	retained subset			whole set (%)
	NND (%)	RDD (%)	LGI (%)	
svmguide2	84.65	84.51	81.86	84.08
balance	94.7	93.97	92.23	95.57
segment	96.91	96.71	93.29	96.33
Abalone(12VS3)	68.34	68.34	67.85	69.72
Abalone(13VS2)	82.99	82.55	79.61	82.68
Abalone(23VS1)	64.84	64.52	63.82	64.71
Waveform(12VS3)	90.21	90.72	87.08	90.49
Waveform(13VS2)	91.97	90.62	90.22	91.38
Waveform(23VS1)	92.12	91.82	89.03	93.1
Avg.	85.2	84.86	82.78	85.34

TABLE 6. The performance comparison OF artificial synthetic datasets.

Datasets		retained subset			whole set (%)
		NND (%)	RDD (%)	LGI (%)	
Banana	misclassification error (%)	6.412±1.619	6.405±1.635	6.583±1.587	6.257±1.314
	outlier detection rate (%)	92.541±3.547	91.100±3.549	91.734±3.517	91.673±3.250
Highleyman	misclassification error (%)	6.612±1.437	6.678±1.425	6.709±1.456	6.727±1.100
	outlier detection rate (%)	97.975±1.013	97.859±1.018	95.260±1.018	98.600±1.066

shaped distribution. In the training set, each dataset contains 2 classes and each class contains 400 samples. Both classes in the training set are used as normal ones. In the test set, we generate 1000 samples for each class and 1000 abnormal samples. In order to eliminate the randomness, the experiment is repeated 30 times. The parameter k and parameter τ are the same in the experiment 1. We use RBF as the kernel function and the width is chosen among $\{2^{-5},2^{-4},2^{-3},2^{-2},2^{-1},2^{0},2^{1},2^{2},2^{3},2^{4},2^{5}\}$. Both parameters γ and C in two-class supervised outlier detection are chosen among $\{2^{-1},2^{0},2^{1},2^{2},2^{3},2^{4},2^{5},2^{6},2^{7},2^{8},2^{9},2^{10}\}$. The parameters are tuned to obtain the least misclassification error via grid search.

The misclassification error and outlier detection rate are reported in Table 6. It can also be obtained that the performance of the retained subset is very close to that of the whole set and the performance of the retained subset selected by NND is better than others.

The visualization of one trail result is illustrated in Fig. 5. We only visualize the retained subset of NND. It can be found that the retained samples are located near the boundary of each class and the descriptions of retained subsets can still represent the normal samples even only 20% critical samples are retained in both toy datasets.

VI. DISCUSSION AND CONCLUSIONS

In this paper, we retain the critical samples to speed up the two-class supervised outlier detection which needs to solve a bigger quadratic program than one-class support vector machine. The critical samples are the ones with nonzero Lagrange multipliers. Since the sample whose Lagrange multiplier is equal to zero has no influence on the decision function, removing the samples with zero Lagrange multipliers would not change the learning result. Thus, we can only retain the samples which would be with nonzero Lagrange multipliers and dispose of others before training two-class supervised outlier detection. We prove that the samples with nonzero Lagrange multipliers must be located near the boundary of each class. We compare three boundary detection methods to retain critical samples for two-class supervised outlier detection, including nearest neighbors' distribution, relative density degree, and local geometrical information. The experimental results demonstrate the effectiveness of our strategy. In three boundary detection methods, we find that the nearest neighbors' distribution is more suitable than others. Although our strategy is used in two-class supervised outlier detection, it can be also migrated to multi-class supervised outlier detection in the future work.

REFERENCES

[1] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 266–282, 1st Quart., 2014.

[2] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Comput. Surv.*, vol. 46, no. 4, p. 55, 2014.

[3] N. S. Halvaiee and M. K. Akbari, "A novel model for credit card fraud detection using artificial immune systems," *Appl. Soft Comput.*, vol. 24, pp. 40–49, Nov. 2014.

[4] M. Hauskrecht, I. Batal, M. Valko, S. Visweswaran, G. F. Cooper, and G. Clermont, "Outlier detection for patient monitoring and alerting," *J. Biomed. Inform.*, vol. 46, pp. 47–55, Feb. 2013.
[5] L. Martí, N. Sanchez-Pi, J. M. Molina, and A. C. B. Garcia, "Anomaly detection based on sensor data in petroleum industry applications," *Sensors*, vol. 15, no. 2, pp. 2774–2797, 2015.
[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, p. 15, 2009.
[7] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.
[8] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, Jan. 2004.
[9] V. Jumutc and J. A. K. Suykens, "Supervised novelty detection," in *Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM)*, Apr. 2013, pp. 143–149.
[10] M. B. de Almeida, A. de Pádua Braga, and J. P. Braga, "SVM-KM: Speeding SVMs learning with *a priori* cluster selection and k-means," in *Proc. 6th Brazilian Symp. Neural Netw.*, Nov. 2000, pp. 162–167.
[11] F. Zhu, J. Yang, J. Gao, C. Xu, S. Xu, and C. Gao, "Finding the samples near the decision plane for support vector learning," *Inf. Sci.*, vol. 382, pp. 292–307, Mar. 2017.
[12] D. Wang and L. Shi, "Selecting valuable training samples for SVMs via data structure analysis," *Neurocomputing*, vol. 71, no. 13, pp. 2772–2781, 2008.
[13] F. Zhu, J. Yang, N. Ye, C. Gao, G. Li, and T. Yin, "Neighbors' distribution property and sample reduction for support vector machines," *Appl. Soft Comput.*, vol. 16, pp. 201–209, Mar. 2014.
[14] R. Koggalage and S. Halgamuge, "Reducing the number of training samples for fast support vector machine classification," *Neural Inf. Process.-Lett. Rev.*, vol. 2, no. 3, pp. 57–65, 2004.
[15] F. Zhu, J. Gao, C. Xu, J. Yang, and D. Tao, "On selecting effective patterns for fast support vector regression training," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3610–3622, Aug. 2018.
[16] G. Guo and J.-S. Zhang, "Reducing examples to accelerate support vector regression," *Pattern Recognit. Lett.*, vol. 28, no. 16, pp. 2173–2183, 2007.
[17] F. Zhu, N. Ye, W. Yu, S. Xu, and G. Li, "Boundary detection and sample reduction for one-class support vector machines," *Neurocomputing*, vol. 123, pp. 166–173, Jan. 2014.
[18] Y. Li, "Selecting training points for one-class support vector machines," *Pattern Recognit. Lett.*, vol. 32, no. 11, pp. 1517–1522, 2011.
[19] F. Zhu, J. Yang, S. Xu, C. Gao, N. Ye, and T. Yin, "Relative density degree induced boundary detection for one-class SVM," *Soft Comput.*, vol. 20, no. 11, pp. 4473–4485, 2016.
[20] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, May 2006.
[21] A. Frank and A. Asuncion. (2010). UCI machine learning repository. School of Information and Computer Science, University of California, Irvine, CA, USA, vol. 213, p. 2.2. [Online]. Available: http://archive.ics.uci.edu/ml
[22] R. P. W. Duin, P. Juszczak, D. De Ridder, P. Paclik, E. Pekalska, and D. M. J. Tax. (2004). *PRTools, A MATLAB Toolbox for Pattern Recognition*. [Online]. Available: http://www.prtools.org
[23] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, vol. 29, no. 2, 2000, pp. 93–104.
[24] H.-P. Kriegel and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 444–452.
[25] B. Liu, Y. Xiao, P. S. Yu, Z. Hao, and L. Cao, "An efficient approach for outlier detection with imperfect data labels," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1602–1616, Jul. 2014.
[26] Y. Xiao, B. Liu, Z. Hao, and L. Cao, "A k-farthest-neighbor-based approach for support vector data description," *Appl. Intell.*, vol. 41, no. 1, pp. 196–211, 2014.
[27] B. Krawczyk, I. Triguero, S. García, M. Wozniak, and F. Herrera, "Instance reduction for one-class classification," *Knowl. Inf. Syst.*, pp. 1–28, May 2018, doi: 10.1007/s10115-018-1220-z.
[28] M. Liu, C. Xu, C. Xu, and D. Tao, "Fast SVM trained by divide-and-conquer anchors," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 2322–2328.
[29] W. Zhang *et al.* (2016). "Scaling up sparse support vector machines by simultaneous feature and sample reduction." [Online]. Available: https://arxiv.org/abs/1607.06996
[30] X. Pang, C. Xu, and Y. Xu, "Scaling KNN multi-class twin support vector machine via safe instance reduction," *Knowl.-Based Syst.*, vol. 148, pp. 17–30, May 2018.
[31] C.-J. Hsieh, S. Si, and I. Dhillon, "A divide-and-conquer solver for kernel support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 566–574.
[32] X. Pan, X. Pang, H. Wang, and Y. Xu, "A safe screening based framework for support vector regression," *Neurocomputing*, vol. 287, pp. 163–172, Apr. 2018.
[33] T. Zhou, J. A. Bilmes, and C. Guestrin, "Divide-and-conquer learning by anchoring a conical hull," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1242–1250.
[34] N. Gillis and S. A. Vavasis, "Fast and robust recursive algorithmsfor separable nonnegative matrix factorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 698–714, Apr. 2014.
[35] K. Yan, Z. Ji, and W. Shen, "Online fault detection methods for chillers combining extended Kalman filter and recursive one-class SVM," *Neurocomputing*, vol. 228, pp. 205–212, Mar. 2017.
[36] M. Hu *et al.*, "Detecting anomalies in time series data via a meta-feature based approach," *IEEE Access*, vol. 6, pp. 27760–27776, 2018.
[37] V. Jumutc and J. A. K. Suykens, "Multi-class supervised novelty detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2510–2523, Dec. 2014.
[38] K. Lee, D.-W. Kim, K. H. Lee, and D. Lee, "Density-induced support vector data description," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 284–289, Jan. 2007.
[39] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2012.
[40] M. Kemmler, E. Rodner, E. S. Wacker, and J. Denzler, "One-class classification with Gaussian processes," *Pattern Recognit.*, vol. 46, no. 12, pp. 3507–3518, 2013.

YUGEN YI was born in Pingxiang, Jiangxi, China, in 1986. He received the B.S. degree from the College of Humanities and Sciences, Northeast Normal University, China, in 2009, and the M.S. degree from the College of Computer Science and Information Technology, Northeast Normal University, in 2012, and the Ph.D. degree from the School of Mathematics and Statistics, Northeast Normal University, in 2015. He is currently a Lecturer with the School of Software, Jiangxi Normal University. His research interests include dimensionality reduction and feature extraction.

WEI ZHOU received the M.S. degree in computer science and technology from the Northeast Normal University, Changchun, in 2015. She is currently pursuing the Ph.D. degree with Northeastern University, Shenyang, China. Her research interests include medical imaging processing, dimensionality reduction, and feature selection.

YANJIAO SHI received the B.S. degree from the School of Computer Science, Northeast Normal University, China, in 2009, and the Ph.D. degree from the School of Mathematics and Statistics, Northeast Normal University, in 2015. She is currently a Lecturer with the College of Computer and Information Engineering, Shanghai Institute of Technology. Her research interests include computer vision, image and video processing, and information security.

JIANGYAN DAI received the B.S. and M.S. degrees from the Computer School of Northeast Normal University, China, in 2008 and 2010, respectively, and the Ph.D. degree from the School of Mathematics and Statistics, Northeast Normal University, in 2014. She is currently a Lecturer with the School of Computer Engineering, Weifang University. Her main research interests are digital image processing, computer vision, biometrics, and information security.