Speedup Two-class Supervised Outlier Detection

Yugen Yi, Wei Zhou, Yanjiao Shi, and Jiangyan Dai

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 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 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 neighbours distribution is more suitable for finding critical samples for 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 neighbours’ distribution information.

Index Terms—Supervised outlier detection, critical sample, nearest neighbours’ 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], [37], and industrial damage detection [5], [6].

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 [7]. 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) [8] for two-class supervised outlier detection [10]. 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 [24]–[26]. 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) [8], support vector data description (SVDD) [9], one-class Gaussian Processing [41]. Vilen 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 [10]. 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) [11]–[15], support vector regression (SVR) [16], [17], and OC-SVM [18]–[20]. The critical samples are located near the decision plane and the boundary of c-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. In [18], Li found boundary samples via extreme points. In [19], Zhu et al. found boundary samples via neighbours’ distribution information. In [20], 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 I.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X, Y )</td>
<td>the sets of training samples and labels</td>
</tr>
<tr>
<td>( l )</td>
<td>the number of training samples</td>
</tr>
<tr>
<td>( X_1, X_2 )</td>
<td>the sets of the samples belonging to class 1, class 2, respectively</td>
</tr>
<tr>
<td>( l_1, l_2 )</td>
<td>the size of the set ( X_1, X_2 ) respectively</td>
</tr>
<tr>
<td>( x_i, y_i )</td>
<td>a training sample and the associated label. ( x_i \in \mathbb{R}^{11} ). When ( x_i ) belongs to the class 1, ( y_i = 1 ); otherwise, ( y_i = -1 ).</td>
</tr>
<tr>
<td>( \Phi(x) )</td>
<td>the mapping of ( x ) in a high-dimensional space.</td>
</tr>
<tr>
<td>( K(x_i, x_j) )</td>
<td>the kernel function which is used to calculate the inner product of ( x_i ) and ( x_j ) in high-dimensional space. such as radial basis function (RBF) kernel function ( K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle = \exp(-|x_i - x_j|/(2\sigma^2)) ).</td>
</tr>
<tr>
<td>( x_i^j )</td>
<td>one of the ( k )-nearest neighbours of ( x_i ), ( j = 1, \ldots, k ).</td>
</tr>
<tr>
<td>( \bar{x}_i )</td>
<td>the mean of the ( k )-nearest neighbours</td>
</tr>
<tr>
<td>( d(x_i, x_i^j) )</td>
<td>the distance between ( x_i ) and its the ( k )-th nearest neighbour</td>
</tr>
<tr>
<td>( kNN(x_i) )</td>
<td>the set consists of the ( k )-nearest neighbours of ( x_i ).</td>
</tr>
</tbody>
</table>

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 decision plane are outliers. In Fig. 1 (b), the decision plane cannot
distinguish whether an unknown sample is outlier. If we want to distinguish 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 \( f_{c_1} = \langle w_1, \Phi(x) \rangle - \rho \) and \( f_{c_2} = \langle w_2, \Phi(x) \rangle - \rho \) represent the two-class supervised outlier detection can be written as follows:

\[
\min_{w_1, w_2, \rho_1, \rho_2} \frac{\gamma}{2} (\|w_1\|^2 + \|w_2\|^2) + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) - \rho_1 - \rho_2 \\
\text{s.t.} \quad y_i((\langle w_1, \Phi(x_i) \rangle - \rho_1) + \xi_i) \geq 0, \quad i \in [1, l] \\
\quad y_i((\langle w_2, \Phi(x_i) \rangle - \rho_2) + \xi_i^*) \leq 0, \quad i \in [1, l] \\
\quad \xi_i \geq 0, \xi_i^* \geq 0, \quad i \in [1, l].
\]

(1)

The decision function \( c(x) \) is defined as follows:

\[
c(x) = \begin{cases} 
\arg \max \limits_{c_i} f_{c_i}(x), & \text{if } \max \limits_{c_i} f_{c_i}(x) > 0 \\
\text{outlier, otherwise}
\end{cases}
\]

(2)

where, \( c_i \) is the index of the \( c_i \)-th hyperplane. If \( c_i = c_1 \), \( x \) belongs to class 1; if \( c_i = c_2 \), \( x \) 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:

\[
L(w_1, w_2, \rho_1, \rho_2, \xi^*, \alpha, \lambda, \beta, \beta^*) = \frac{\gamma}{2} (\|w_1\|^2 + \|w_2\|^2) + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) - \rho_1 - \rho_2 \\
- \sum_{i=1}^{l} \alpha_i(y_i(\langle w_1, \Phi(x_i) \rangle - \rho_1) + \xi_i) \\
+ \sum_{i=1}^{l} \lambda_i(y_i(\langle w_2, \Phi(x_i) \rangle - \rho_2) + \xi_i^*) \\
- \sum_{i=1}^{l} \beta_i \xi_i - \sum_{i=1}^{l} \beta_i^* \xi_i^*
\]

(3)

where \( \xi^*, \xi^* \), and \( \alpha, \lambda, \beta, \beta^* \) are the vectors form of slack variables and Lagrange multipliers, respectively. Setting the derivatives of Eq. (3) with respect to \( w_1, w_2, \xi, \xi^*, \alpha, \lambda, \beta, \beta^* \) to zeros, then:

\[
w_1 = \gamma \sum_{i=1}^{l} \alpha_i y_i \Phi(x_i) + \sum_{i=1}^{l} \lambda_i y_i \Phi(x_i) \\
w_2 = \gamma \sum_{i=1}^{l} \lambda_i y_i \Phi(x_i) + \sum_{i=1}^{l} \alpha_i y_i \Phi(x_i)
\]

(4)

\[
C - \beta_i - \alpha_i = 0 \\
C - \beta_i^* - \lambda_i = 0
\]

(6)

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

\[
\min_{w_1, w_2, \rho_1, \rho_2} \frac{\mu_1}{2} (\alpha^T G \alpha + \lambda^T G \lambda) + \mu_2 \alpha^T G \lambda \\
\text{s.t.} \quad 0 \leq \alpha_i \leq C, \quad i \in [1, l] \\
0 \leq \lambda_i \leq C, \quad i \in [1, l] \\
y^T \alpha = 1, y^T \lambda = 1.
\]

(10)

where \( \mu_1 = \frac{\gamma}{\sqrt{\gamma^2 - 1}}, \mu_2 = \frac{1}{\sqrt{\gamma^2 - 1}}, y \) is the vector form of labels, \( G \) is a \( l \times l \) matrix and \( K = \Phi(y y^T) \) where \( K \) is the kernel matrix and \( K(i, j) = K(x_i, x_j) \), \( \circ \) is component-wise multiplication. The \( f_{c_1}(x) \) and \( f_{c_2}(x) \) can be represented as follows:

\[
f_{c_1} = \gamma \sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + \sum_{i=1}^{l} \lambda_i y_i K(x_i, x) - \rho_1 \\
f_{c_2} = \gamma \sum_{i=1}^{l} \lambda_i y_i K(x_i, x) + \sum_{i=1}^{l} \alpha_i y_i K(x_i, x) - \rho_2.
\]

(11)

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

Obviously, only the sample with nonzero Lagrange multipliers are critical to the hyperplanes. \( f_{c_1}(x) \) and \( f_{c_2}(x) \) (\( \alpha_i \neq 0 \) 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 \( x_i \) be a sample in the training set. In the feature space, the distances between \( x_i \) and the hyperplanes, \( f_{c_1}(x) \) and \( f_{c_2}(x) \), are \( \frac{\|w_1 \Phi(x_i) - \rho_1\|}{\|w_1\|^2} \) and \( \frac{\|w_2 \Phi(x_i) - \rho_2\|}{\|w_2\|^2} \), respectively. The constraints \( y_i(\langle w_1, \Phi(x_i) \rangle - \rho_1) + \xi_i \) and \( y_i(\langle w_2, \Phi(x_i) \rangle - \rho_2) + \xi_i^* \) can be rewritten as

\[
\frac{y_i}{\|w_1\|^2} + \frac{\|w_1\|^2}{\|w_1\|^2} = \frac{\xi_i}{\|w_1\|^2} \quad \text{and} \quad \frac{y_i}{\|w_2\|^2} + \frac{\|w_2\|^2}{\|w_2\|^2} = \frac{\xi_i^*}{\|w_2\|^2}.
\]

The critical samples in two-class supervised outlier detection contain two cases: the sample with nonzero \( \alpha_i \) in class 1 and the sample with nonzero \( \lambda_i \) in class 2.

**Case 1:**

In class 1, \( y_i = 1 \), \( \frac{y_i}{\|w_1\|^2} + \frac{\|w_1\|^2}{\|w_1\|^2} = \frac{\xi_i}{\|w_1\|^2} \geq 0 \), and the corresponding KKT condition can be rewritten as

\[
\sum_{i=1}^{l} \alpha_i y_i = 1 \quad \text{(8)}
\]

\[
\sum_{i=1}^{l} \lambda_i y_i = -1 \quad \text{(9)}
\]
The boundary of the distribution of class 1.

\[ \alpha_i \left( \frac{y_i(x_i, \Phi(x_i)) - \rho_1}{\|w_1\|^2} \right) + \frac{\xi_i}{\|w_1\|^2} = 0 \]

If \( \alpha_i \neq 0 \), it must hold that

\[ \frac{y_i(x_i, \Phi(x_i)) - \rho_1}{\|w_1\|^2} + \frac{\xi_i}{\|w_1\|^2} = 0 \]

Since \( \xi_i \geq 0 \), the \( x_i \) must locate near the boundary of class 1. That is to say, the critical samples with nonzero \( \alpha_i \) in class 1 are located near the boundary of the distribution of class 1.

Case 2:

In class 2, \( y_i = -1 \),

\[ \frac{y_i(x_i, \Phi(x_i)) - \rho_2}{\|w_2\|^2} + \frac{\xi_i}{\|w_2\|^2} \geq 0 \]

and the corresponding to KTT condition can be rewritten as

\[ \lambda_i \left( \frac{y_i(x_i, \Phi(x_i)) - \rho_2}{\|w_2\|^2} + \frac{\xi_i}{\|w_2\|^2} \right) = 0 \]

Then, similar to the analysis of class 1, the critical samples with nonzero \( \lambda_i \) 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 neighbours' distribution [18], relative density degree [19], [39], and local geometry information [20] 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 near the boundary of class 2, respectively. In the last Step, the boundary 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 Step 2 and Step 5, the scores could be calculated by a boundary detection method, such as nearest neighbours' distribution [18], relative density degree [20], local geometry information [19]. In order to ensure the integrity of this paper, we recap nearest neighbours' distribution, relative density degree, and local geometry information as follows. Zhu et al. [18] pointed out that a sample's location \((x_i)\) in the dataset is related to the nearest neighbours' distribution \((kNN(x_i))\). The k-nearest neighbours is enclosed by a hypersphere with center which is itself and radius which is the distance between

Algorithm 1

**Input:**
- training set \( \{X, Y\} \);
- the number of nearest neighbours \( k \);

**Output:**
- a subset \( \{X', Y'\} \);

1: find the set \( X_1 \);
2: calculate the scores of all samples in \( X_1 \) 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 \( X'_1 \).
   The corresponding label, \( Y'_1 \), is \( e \) with length \( \tau * l_1 \);
4: find the set \( X_2 \);
5: calculate the scores of all samples in \( X_2 \) via a boundary detection method and sort the scores in descending (ascending) order;
6: retain the top \( \tau * l_2 \) samples to construct \( X'_2 \). The corresponding label, \( Y'_2 \), is \(-1 * e\) with length \( \tau * l_2 \);
7: return \( \{X', Y'\} \) where \( X' = X'_1 \cup X'_2 \) and \( Y' = Y'_1 \cup Y'_2 \);

to learn a two-class supervised outlier detection model on \( \{X', Y'\} \).

Fig. 2. An explanation of the two-class supervised outlier detection. Left: the samples in the original space; right: the samples in the feature space.
the $k$-th nearest neighbor and itself ($d(x_i, x_j^k)$). Then, the hypersphere is divided by a hyperplane which is perpendicular to the difference between this sample and the mean of $k$-nearest neighbours ($\bar{x}_i$). Then, the distribution of the nearest neighbours has the following properties

**Property 1:** The number of nearest neighbours in the part which $\bar{x}_i$ 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-neighbour angles majorly ranges in $[0,k]$.

Here the sample-neighbour angle is defined as follows.

**Definition 1. (Sample-neighbour angle) [18]:** Let $\theta_i^j$ be the sample-neighbour angle. The $\theta_i^j$ is the angle between $x_i - \bar{x}_i$ and $x_i - x_j^k$.

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-neighbour angles. The cosine sum can be represented as follows.

$$c_{sum}(x_i) = \sum_{j=1}^{k} \cos \theta_i^j = \sum_{j=1}^{k} \frac{\langle x_i - \bar{x}_i, x_i - x_j^k \rangle}{\| x_i - \bar{x}_i \| \| x_i - x_j^k \|}. \quad (13)$$

By introducing kernel trick, Eq. (13) can be rewritten as the following Eq. (14).

$$\rho_i = \exp \left\{ w' \times \frac{\text{Mean}^{k}_{kNN}}{d(x_i, x_i^k)} \right\} \quad (15)$$

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

Li [19] 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 neighbours are located on the opposite side of the tangent plane, as shown in Fig. 3 (a); when the surface is concave, most of nearest neighbours 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.

Let $v_{i,j}$ ($j = 1, \ldots, k$) represent the difference between $x_i$ and $x_{i,j}$ ($v_{i,j} = x_j^k - x_i$), $\bar{v}_{i,j}^n$ represent the normalization of $v_{i,j}$, and $v_{i,j}^n$ represent the sum of $v_{i,j}^n$ ($v_{i,j}^n = \sum_{j=1}^{k} v_{i,j}^n$ ). Then, if the nearest neighbour is located at the side which the normal vector points, the angle between $x_i$ and $v_{i,j}^n$ is in the range $[0,\pi/2]$; otherwise, the angle is the range $[0, \pi]$. The boundary sample could be found by the following equation.

$$L_i = \frac{1}{k} \sum_{j=1}^{k} (v_{i,j}^T v_{i}^n \geq 0). \quad (16)$$

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

In the 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 neighbours’ 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 [21]). 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 reserved 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 neighbours’ 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 [22]. Second, we evaluate the performance on 2 artificial synthetic datasets. The artificial synthetic
IEEE ACCESS

\[ c^{\text{sum}}(x_i) = \sum_{j=1}^{k} \frac{K(x_i, x_i) - K(x_i, x_i') - \frac{1}{k} \sum_{p=1}^{k} K(x_i, x_i') + \frac{1}{k} \sum_{p=1}^{k} K(x_i', x_i')}{\sqrt{K(x_i, x_i) - \frac{2}{k} \sum_{p=1}^{k} K(x_i, x_i') + \frac{1}{k^2} \sum_{p,q=1}^{k} K(x_i', x_i') - 2K(x_i, x_i') + K(x_i', x_i')}}. \]  

(14)

Fig. 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.

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 [22] to verify the proposed method. The detailed description of these datasets is listed in Table II. 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 svmguide2 and balance, 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 abnormal, 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 \( \tau \) 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 \( \{2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3, 2^4\} \). Both parameters \( \gamma \) 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 III. 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 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 IV and Table V, 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.

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 neighbours’ 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 neighbours’ 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

### TABLE III
THE TIME COMPARISON OF THE BENCHMARK DATASETS.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>NND</th>
<th>RDD</th>
<th>LGI</th>
<th>whole set (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>preprocess time (s)</td>
<td>training time (s)</td>
<td>preprocess time (s)</td>
<td>training time (s)</td>
</tr>
<tr>
<td>svmguide2</td>
<td>0.0218</td>
<td>0.0912</td>
<td>0.0218</td>
<td>0.0895</td>
</tr>
<tr>
<td>balance</td>
<td>0.0267</td>
<td>0.1502</td>
<td>0.0256</td>
<td>0.1446</td>
</tr>
<tr>
<td>segment</td>
<td>0.0307</td>
<td>0.4107</td>
<td>0.0284</td>
<td>0.4217</td>
</tr>
<tr>
<td>Abalone(12VS3)</td>
<td>0.1831</td>
<td>1.2787</td>
<td>0.1768</td>
<td>1.2761</td>
</tr>
<tr>
<td>Abalone(13VS2)</td>
<td>0.2019</td>
<td>0.6817</td>
<td>0.2014</td>
<td>0.6611</td>
</tr>
<tr>
<td>Abalone(23VS1)</td>
<td>0.2150</td>
<td>0.7887</td>
<td>0.2237</td>
<td>0.7572</td>
</tr>
<tr>
<td>Waveform(12VS3)</td>
<td>0.3931</td>
<td>0.8002</td>
<td>0.4079</td>
<td>0.7804</td>
</tr>
<tr>
<td>Waveform(13VS2)</td>
<td>0.4042</td>
<td>0.7972</td>
<td>0.3659</td>
<td>0.8239</td>
</tr>
<tr>
<td>Waveform(23VS1)</td>
<td>0.3994</td>
<td>0.9228</td>
<td>0.3950</td>
<td>0.9579</td>
</tr>
</tbody>
</table>

### TABLE IV
THE MISCLASSIFICATION ERROR COMPARISON OF THE BENCHMARK DATASETS

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

### TABLE V
THE OUTLIER DETECTION RATE COMPARISON OF THE BENCHMARK DATASETS

<table>
<thead>
<tr>
<th>Datasets</th>
<th>NND (%)</th>
<th>RDD (%)</th>
<th>LGI (%)</th>
<th>whole set (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>svmguide2</td>
<td>84.65</td>
<td>84.51</td>
<td>81.86</td>
<td>84.08</td>
</tr>
<tr>
<td>balance</td>
<td>94.7</td>
<td>93.97</td>
<td>92.23</td>
<td>95.57</td>
</tr>
<tr>
<td>segment</td>
<td>96.91</td>
<td>96.71</td>
<td>93.29</td>
<td>96.33</td>
</tr>
<tr>
<td>Abalone(12VS3)</td>
<td>68.34</td>
<td>68.34</td>
<td>67.85</td>
<td>69.72</td>
</tr>
<tr>
<td>Abalone(13VS2)</td>
<td>82.99</td>
<td>82.55</td>
<td>79.61</td>
<td>82.68</td>
</tr>
<tr>
<td>Abalone(23VS1)</td>
<td>64.84</td>
<td>64.52</td>
<td>63.82</td>
<td>64.71</td>
</tr>
<tr>
<td>Waveform(12VS3)</td>
<td>90.21</td>
<td>90.72</td>
<td>87.08</td>
<td>90.49</td>
</tr>
<tr>
<td>Waveform(13VS2)</td>
<td>91.97</td>
<td>90.62</td>
<td>90.22</td>
<td>91.38</td>
</tr>
<tr>
<td>Waveform(23VS1)</td>
<td>92.12</td>
<td>91.82</td>
<td>89.03</td>
<td>93.1</td>
</tr>
<tr>
<td>Avg.</td>
<td>85.2</td>
<td>84.86</td>
<td>82.78</td>
<td>85.34</td>
</tr>
</tbody>
</table>
TABLE VI
THE PERFORMANCE COMPARISON OF ARTIFICIAL SYNTHETIC DATASETS.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>retained subset</th>
<th>whole set (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NND (%)</td>
<td>RDD (%)</td>
</tr>
<tr>
<td>Banana</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misclassification error (%)</td>
<td>6.412±1.619</td>
<td>6.405±1.635</td>
</tr>
<tr>
<td>outlier detection rate (%)</td>
<td>92.541±3.547</td>
<td>91.100±3.549</td>
</tr>
<tr>
<td>Highleyman</td>
<td></td>
<td></td>
</tr>
<tr>
<td>outlier detection rate (%)</td>
<td>97.975±1.013</td>
<td>97.859±1.018</td>
</tr>
</tbody>
</table>

Fig. 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.


[34] Yan K, Ji Z, Chen W. Online fault detection methods for chillers combining extended kalman filter and recursive one-class SVM. Neurocomputing, 2017, 228: 205-212.


