2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Fast Support Vector Data Description Training Using Edge Detection on Large Datasets

Chenlong HU, Bo ZHOU and Jinglu HU

***Abstract*— Support Vector Data Description (SVDD) inherits properties of Support Vector Machines (SVM) and has become a prominent One Class Classifier (OCC). Same to standard SVM, its $O(n^3)$ time and $O(n^2)$ space complexities, where n is the number of training samples, have become major limitations in cases of large training datasets. As a simple and effective method, reducing the size of training dataset through reserving only samples mostly relevant to learned classifier, can be adopted to overcome the limitations. A trained SVDD enclosed decision boundary always locates on edge area of data distribution and is decided by a small subset of Support Vectors(SVs). Therefore, in this paper, we present a method based on edge detection such that edge samples mostly relevant to decision boundary can be preserved. And clustering techniques are also be applied to keep centroids representing the global distribution properties so as to avoid over-outside of decision boundary. To restrict the influences of noises, each training pattern is assigned with a weight. Experiments on real and artificial data sets prove that the classifier trained on reconstruction training set consisting of edge points and centroids can preserve performance with much faster training speed.**

I. Introduction

ONE Class Classification (OCC) is a special type of classification which estimates a description of a single class of samples and to reject samples referred to *outliers* or *negative* that don't resemble this class. OCC assumes that only information of *one* class is available(we call it *target* class) [1]. The most intuitive method used for OCC may be to train ordinary two class algorithms by generating some artificial negative data. However, since one has to create enough artificial data to make negative samples around target class in all feature directions, the methods can be available only in low dimensional problems and some limited classifiers [2]. In another kind of straightforward methods, one class problem can be solved by probability density estimation of target patterns and then decide whether a test object follow the same distribution $\mathcal{P}$ or not. For instance, Gaussian and mixture of Gaussian model are often adopted [3]. However, the estimation methods depend critically on the parametric form of the density function which is hard to be well-defined in high-dimensional data. It may fail when this assumption is incorrect [4]. Another drawback is that it often focuses on modeling the high density areas resulting to reject low density regions [1].

Self-organizing-mapping (SOM) [5] and kernel principal component analysis (kernel PCA) [6] belong to reconstruction methods wherein a model is built and fitted to target data by using prior knowledge about the target class and making assumptions about the generating process. Then outliers can be rejected when they do not satisfy the assumptions about the target distribution. However, the methods are relatively sensitive to the scaling of the features and lead to poor descriptions.

Finally, boundary methods follow the principle of Vapnik that never to solve a general problem rather than the one we actually need to solve [7]. Boundary methods attempt to only train the description boundary of target data without a probability density estimation step which might be too demanding. Support Vector Data Description [8] and ν-Support Vector Classifier (ν-SVC) [9] are two commonly used boundary methods for solving OCC problems. Above both boundary methods also follow the Structural Risk Minimization (SRM) principle and inherit the excellent generalization ability of Support Vector Machines (SVM). ν-SVC estimates a hyperplane in kernel feature space with maximum margin from the origin. SVDD finds a minimum-volume hypersphere that contains all or most of the training data. [1], [10] have proven that these two approaches will give identical solution when the data is preprocessed to unit norm.

In this paper, we will just focus on the implement and improvement of SVDD. SVDD has successful applications in various fields [11], [12], [13]. Same to SVM, SVDD classifer is obtained through a quadratic programming (QP) formula. Therefore, it has $O(n^3)$ time and $O(n^2)$ space complexities which make it unapplicable on large training data. To solve this problem, Sequential Minimal Optimisation(SMO) [14] algorithm improves the complexity through breaking up large QP into the smallest optimization steps. However it still needs large training time with the implement of an improved version of SMO algorithm, as shown in [10], [15].

Scaling down the training set as a pre-process step before training is a simple but effective method to overcome the limitation of time complexity. This means that some samples mostly irrelevant to trained classifier should be discarded while relevant samples must be reserved. Li et al. [16] and Shin et al. [17] train two-class SVM classifier on a reduced subset of original training set and achieve much faster training speed. Though above methods can obtain successful results on large two class problems, they cannot be directly applied to one class case due to the lack of information from both class labels. [18] proposes a method selecting training points for one class ν-SVC, but this approach may suffer from bad descriptions because they didn't concern the structure of entire training set. Another method mentioned

Chenlong HU, Bo ZHOU and Jinglu HU are with the Graduate School of Information, Production and Systems, Waseda University, JAPAN (email: huchenlong@ruri.waseda.jp, zhoubo_bird@toki.waseda.jp, jinglu@waseda.jp).

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

in [19] using simple neighbour number to obtain reduced subset cannot perform well in case of different densities existing and high dimensional data space.

Therefore, it is necessary to come up with an approach which can efficiently and completely choose relevant subset representing both local and global properties of training data in high dimensional and different densities existing cases. As a result, SVDD classifier trained on this subset must has faster training process and comparative performance with over result learned from the whole set. The closed decision boundary of SVDD is decided by Support Vectors(SVs), a small subset that locate close to edge areas of training data [20]. Therefore, samples locating on edge areas seem to be mostly relevant to decision boundary.

In this paper, we propose a fast training method based on edge detection technique to reduce the training size. A reduced reconstruction subset is obtained by merging edge samples and clustering centers, which contains mostly important relevant samples. Edge samples preserve local properties around the enclosed SVDD boundary. In the proposed Density-Angle edge detector, relative density of each sample is taken into consideration to solve the bias problem when different densities exists[21]. On the other hand, angle variances between difference vectors from training object to its neighbor are also taken into account because angles are more robust in high dimensional data.

For efficiency and completeness of edge detector, we combine both density and angle measurements to generate two subsets using density based detector called Candidate and Complement sets. Candidate set can be used to improve the time efficiency of angle detection. And Complement set fills up missing edge points can not be detected by pure angle based method . The volume of edge set can be effectively and smoothly controlled via two parameters relative to density and angle respectively. In order to restrict over-outside of the SVDD boundary result from the extremely small edge subset sometimes, the global distribution properties of the entire data set also should be obtained. The centers computed by clustering techniques represent global distribution properties and then are combined with edge samples to reconstruct a more complete reduced training subset. However, the influences of noise data may be larger because of the reduction of normal samples. Therefore, to enhance the ability of SVDD in defending the sensitivity to noises, local data uncertainty of each data is incorporated into the construction of the classifier by assigning each sample a weight. With incorporations of weights, samples that are more likely to be noises possess relative smaller importance in learning process. Finally, we train weighted SVDD on the reconstruction subset with much faster speed while preserving classification performances on a variety of data sets.

The rest of this paper is organized as follows: Section II introduces Support Vector Data Description. Section III shows the details of proposed method. Some simulation results are presented in Section IV to show its performance. Finally, Section V concludes the paper.

II. Support Vector Data Description

SVDD is one of the best-known support vector learning methods for one-class classification. It is a data domain description method which aims to obtain a enclosed boundary around the target data set. Let $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathcal{X}$, be a set of training samples, where $N \in \mathbb{N}$ is the number of observations and $\mathcal{X}$ is a compact subset of $\mathbb{R}^d$. Consider a ball with center $\boldsymbol{a}$ and radius R, the main idea of SVDD is to find a ball that can achieve the trade-off between the volume of ball should be as small as possible and contain as many training data as possible simultaneously. To solve a quadratic program problem:

$$\min \quad F(R, \boldsymbol{a}, \xi_i) = R^2 + \frac{1}{\nu N} \sum_i \xi_i \tag{1}$$

$$\text{subject to } \|\boldsymbol{x_i} - \boldsymbol{a}\|^2 \leq R^2 + \xi_i \quad \xi_i \geq 0 \tag{2}$$

where ξ_i is the slack variable representing a penalty associated with the deviation of ith training sample outside the ball and ν is the parameter controls tradeoff between volume of ball and cost of misclassification. By introducing Lagrange multipliers, the above optimization problem is transformed into the dual formulation :

$$\max_{\alpha_i} \sum_i \alpha_i \langle \boldsymbol{x_i}, \boldsymbol{x_i} \rangle - \sum_{i,j} \alpha_i \alpha_j \langle \boldsymbol{x_i}, \boldsymbol{x_j} \rangle \tag{3}$$

$$\text{subject to } \sum_i \alpha_i = 1, 0 \leq \alpha_i \leq \frac{1}{\nu N} \tag{4}$$

By incorporating feature map Φ: $\mathcal{X} \longrightarrow \mathcal{F}$, mapping feature vector in $\mathcal{X}$ to a inner product space $\mathcal{F}$ such that the dot products in high-dimensional feature spaces can be obtained efficiently via a suitable pre-specified kernel function k: $k(\boldsymbol{x}, \boldsymbol{x'}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x'}) \rangle$ (Gaussian kernel: $k(\boldsymbol{x}, \boldsymbol{x'}) = \exp(-\|\boldsymbol{x} - \boldsymbol{x'}\|^2 / 2\sigma^2)$), we can express more flexible decision boundary in $\mathcal{X}$ as shown in Fig.1. From Karush-Kuhn-Tucker (KKT) conditions, the center $\boldsymbol{a}$ can be expressed as:

$$\boldsymbol{a} = \sum_i \alpha_i \Phi(\boldsymbol{x}) \tag{5}$$

Equation (5) shows that the center of the sphere is a linear combination of the objects. It is necessary to be noticed that samples with $\alpha_i > 0$ usually decide the decision boundary of SVDD. Therefore, we call the small set the support vectors (SV's) of the description. Finally, the decision function can be obtained as :

$$\begin{aligned} y(\boldsymbol{x}) &= \text{sign}(\|\boldsymbol{x} - \boldsymbol{a}\|^2 - R^2) \\ &= \text{sign}(\|\boldsymbol{x} - \sum_i \alpha_i \Phi(x_i)\|^2 - R^2) \end{aligned} \tag{6}$$

From Equation (6), it is more clear that the construction of SVDD classifier only depends on SVs. A new test object $\boldsymbol{x}$ will be accepted as target class if $y(\boldsymbol{x})$ is non-positive, i.e. the distance to the center of the sphere is not greater than than the radius R.

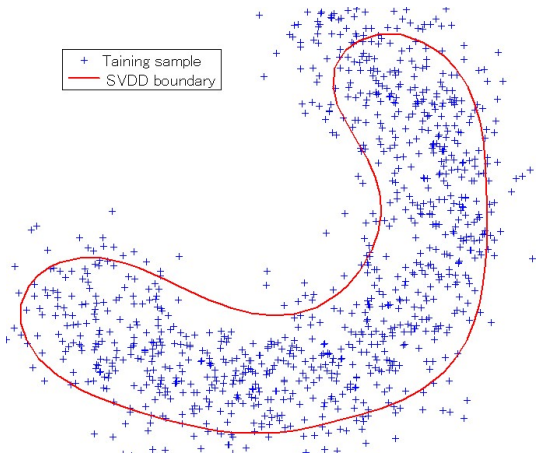


Fig. 1. SVDD Decision Boundary

III. Reducing Training Set Using Edge Detection

SVDD classifiers are found by solving a quadratic programming(QP) problem which can get unique global optimization. However, the QP problem here has $O(n^3)$ time and $O(n^2)$ space complexities, where n is the number of training samples, and makes SVDD become unapplicable on large data sets. As a pre-processing method, reducing the training size with discarding irrelevant samples can be taken into consideration. The idea of this method is to construct a complete and small reduced subset composing of samples mostly decisive to SVDD boundary. Then SVDD classifier learned from reduced training set should obtain comparative classification performance as that on whole training set. The reduced training set should capture the whole properties which decide the performance of classifier. As mentioned in Section II, the description boundary of SVDD is decided on the basis of SVs. Therefore, samples that are likely to be SVs are important samples to test accuracy. So firstly, we want to precisely and completely detect edge samples which seem to play most important roles in the training case.

A. Input and Kernel Space

With the incorporation of kernel function, an implicit mapping of the data samples from input space into another feature space is defined. For instance, the feature space induced by the Gaussian kernel is infinite-dimensional. And all mapped feature vectors $\Phi(\boldsymbol{x})$ with equal norm 1 lie on a unit ball [22]. Kernel functions are always adapted into SVDD training to learn a more flexible description instead of a rigid hypersphere in input space. However, we want to perform the selection of samples close to decision boundary in input data space. If we want to use neighbour relationships among training samples to do this, the invariance features of neighborhood relations under input space to feature space mapping must be guaranteed. The validity of proposed method cannot be preserved in case of neighborhood relation in the input space is not constant. Shin et al. [17] has proven and tested the invariant property of the neighborhood relation from input to feature space mapping in SVMs with different kernels. This result can also be used in SVDD. In the case of Gaussian kernel, the relationship can be preserved very well. This means that if Gaussian kernel is accepted, we can perform edge detection in input space and get a comparative size reduction in feature space.

B. Density-Angle Based edge detection

To retrieve edge subset efficiently and completely, we adapt considerations from both density and angle based differences between training samples. In our density-angle based edge detection method, we firstly incorporate density step to obtain two subsets of input data, Candidate set and Complement set. Candidate set can be used to improve the time efficiency of purely angle based detector and Complement set fills the missing edge patterns which angle method could not acquire in some cases.

As shown in Fig.1, edge samples aim to be extracted are locating at the outside margin of densely distributed data. From the perspective of density, we can judge edge patterns if they are located in lower density areas. In another aspect, consider the angle motivation in Fig.2, P is an edge pattern aim to detect and Q is an inner sample burying inside the data cluster. The neighbors of P scatter in similar directions, and angles between difference vectors from P to its neighbors vary in smaller range compared with Q. One step further, we can assert a training sample x_i need to be labelled as edge set when the variance of the angles between difference vectors of x_i to its k neighbors is smaller than a threshold value. The variance $v(x_i)$ related to each sample x_i and its k neighbors $x_j{}'s$ can be computed by formula:

$$v(x_i) = \text{Variance}\left(\theta(\overrightarrow{x_i x_j}, \overrightarrow{x_i x_{j'}})\right), x_j, x_{j'} \in knn(x_i), j \neq j' \tag{7}$$

where θ denotes the angle between two vectors $\overrightarrow{x_i x_j}$ and $\overrightarrow{x_i x_{j'}}$, then we can judge x_i as edge set if the edge label $e(x_i)$ equals to 1.

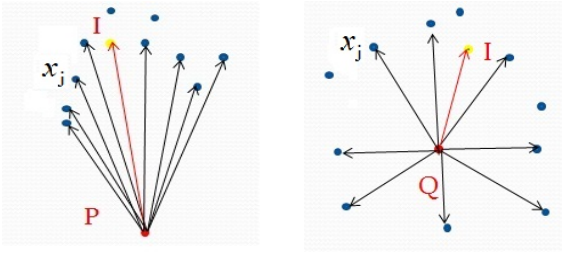
$$e(x_i) = \begin{cases} 1 & v(x_i) < \tau \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

here the threshold τ is defined as $\lfloor \varepsilon * N \rfloor$-th largest $v(x_i)$ value and determine the volume of detected edge set, ε is the parameter of angle method.

Since density based methods implicitly utilize assessments of differences in distances between objects, and the performance of purely distance based differences may suffer to deteriorate in high dimensional data space. While angle based method can alleviate the effects of "curse of dimensionality" and thus becomes more robust than purely density method. However, when the size of input data is very large, angle based method has the problem of time complexity due to :

- All input objects must be considered ;
- All neighbor difference vectors for each object must be considered.

Another problem comes from the observation of experiments, neither angle method nor density method can acquire complete edge points in some cases. Thus, to utilize the merits of



(a) Edge sample P (b) Inner sample Q

Fig. 2. Angles variances between inner and edge sample)

angle method with small time consuming and completeness of edge set, density based method is introduced. The proposed *Density-Angle based edge detection* method can be summarized as following 4 steps :

Step 1 Choose Candidate set **C1** and Complement set **C2** for angle detector using *Algorithm 1* ;

Step 2 Determine Support Neighbor of x_i to refine the computation of $v(x_i)$;

Step 3 Detect edge set **A** based on Equation(8)

Step 4 Combine **A** and **C2** to obtain final edge set **E**

Candidate set **C1** is a subset of entire input data chose to reduce the judge turns of edge based detector. *Algorithm 1* illustrates the implementation of detecting **C1**. The determination of **C1** can be seen as a rough pre-selection process by using density based edge detector, and the parameter δ representing the volume of **C1** usually is assigned to a large value. As discussed above, samples possessing small density are more likely to be edge patterns. So samples with small density degree are chose as Candidate set. Inspired from local outlier factor (LOF) [21], we estimate the relative density to avoid the shortcoming in case of different densities exits where edge samples located at high density areas are easy to discarded incorrectly and a biased classifier boundary is estimated. Here, each object x_i is given a score $o(i)$ to measure how isolated the object is with respect to its surrounding neighbours. For most inner points, their $o(i)$ are approximately close to 1, while edge points output larger values of $o(i)$.

In step (2), with the introduction of Support Neighbor **I** which has the largest density value, as shown in Fig.2, only variances of the angles between difference vectors of x_i to its neighbors and the vector of x_i to support neighbor **I**, instead of all pairs of feature vectors. The computation of $v(x_i)$ turns into :

$$v(x_i) = \text{Variance}\left(\theta(\overrightarrow{x_i\mathbf{I}}, \overrightarrow{x_i x_j})\right), x_j \in knn(x_i) \quad (9)$$

In step (3), the parameter of angle edge detector is the same with ε appeared on the selection of Complement set **C2**. And once the threshold τ known, calculations and ranking of $v(x_i)s$ complete immediately. Then we can get edge set **A**. From *Algorithm 1*, we also know that Compliment set **C2** can be chose in the same way with **C1**. The parameter ε means the size of **C2**. Note that in the step (4), we combine edge set detected by angle based method and Complement **C2** to acquire a complete edge set $\mathbf{E} = \mathbf{A} \cup \mathbf{C2}$. This based on the observation that some edge samples cannot be determined with purely angle method and Compliment set detected by density based method can become a complement of angle method. Two parameters ε and δ can precisely and effectively control time cost of edge detection and the volumes of edge set, i.e. $\varepsilon \leq \frac{1}{N}|\mathbf{E}| \leq \delta$.

Algorithm 1 Select Candidate and Complement sets

Input:

Training set $X = \{x_1, \ldots, x_N\}$;

The number of neighbors , k ;

Parameters $0 < \delta \leq 1$ and $0 < \varepsilon \leq 1$ and $\varepsilon \leq \delta$

Output:

Candidate **C1** and Complement set **C2**;

1: Construct the k-distance neighbor matrix $\mathcal{D}_k$;

2: Compute reachabability distance matrix :

$$\text{reach-dist}_k(x_i, x_j) = \max\{\text{k-dist}(x_j), \text{dist}(x_i, x_j)\}$$

3: Compute local reachability distance (lrd) :

$$\text{lrd}_k(x_i) = \frac{1}{|knn(x_i)|} \sum_{x_j \in knn(x_i)} \text{reach-dist}_k(x_i, x_j)$$

4: Compute local outlier factor(lof) :

$$\text{lof}(x_i) = \frac{1}{|knn(x_i)|} \sum_{x_j \in knn(x_i)} \frac{\text{lrd}(x_j)}{\text{lrd}(x_i)}$$

5: Let $o(i) = lof(x_i)$

6: Samples are sorted in ascending order of $o(i)$

7: Choose largest $\lfloor \delta * N \rfloor$ samples as **C1**

8: Choose largest $\lfloor \varepsilon * N \rfloor$ samples as **C2**

9: **return** **C1** and **C2** ;

C. Training set reconstruction

Reduced training set using edge detection can scale down the time complexity problem of SVDD. The edge points detected possess local properties of data distribution and we can get an approximate description boundary. However, in some simulations, SVDD may suffer from over-outside problem which means that the boundary is much too loose resulting in enclose excess outliers. These results don't follow the destination of SVDD that to obtain a minimum volume boundary while reject all other negative objects. Hence, it is better to add some data representing the structure information of target data set. Therefore, we use clustering technique to reserve centroids of data clusters. These centroids amount to global properties relative to local properties represented by edge data. This idea is something like reconstruction methods in one class problems which use a model to represent the distribution of target data. K-means clustering group data into k number of clusters based on the measure distance similarity. The optimization function of K-means is a non-convex function, and so it is not guaranteed to converge to the global minimum [23]. But the destination of K-means we use here is different from what in reconstruction methods.

So precision of clustering here is not quite important as reconstruction methods do. After the detection of centroids, reconstruction set can be obtained by merging edge set and clustering centroids.

D. Noises to be considered

It must be considered that noises are easy to detected as edge points and added to training process. And their influences become larger because of the reduction of normal patterns even though SVDD can reject part of these noises and tolerate some misclassifications. Therefore, we assign a weight to each point so as to restrict the influence to the classifier. Here, based on the knowledge from *Algorithm* 1 of edge detection, we make $m(i) = 1/o(i)$ to be the weight of ith edge pattern in reconstruct training set. This designs mean that each noises data get less importance in the training process. This consideration can be seen as an incorporation of local uncertainty into the construction of global classifier [24]. It must be mentioned that weights of centers obtained by clustering techniques equal to 1, since centers represent global distribution of target class data.

IV. Experiments

In order to evaluate the proposed method described above, we conduct experiments on a variety of data sets. All reported results are implemented by matlab code based on DDtools, an OCC tool developed by David Tax [25]. Our experiments are run on a dell desktop with Intel Core 2 Duo CPU, 2.66 GHZ, and the OS is Linux 2.6.18.

For comparisons, SVDD classifiers learned from original training data are utilized here as baselines. Experiment results on artificial and real word data sets are shown on Section III-A and B respectively. Among the three comparison metrics, the first is the size of reduced training set to check whether our proposed method can significantly scale down the training size. And we can prove that weighted SVDD trained on reduced training set take much less CPU time over SVDD on whole sets. AUC accuracy, the area under the curve of receiver operating characteristic (ROC), used to evaluate the performance of proposed method. AUC value is always between 0 and 1, and the larger AUC indicates the better performance of a method. In all tables here, we use WSVDD+Re to represent the proposed method, which means weighted SVDD trained on Reconstruction training set(we use SVDD+Re in figures). Gaussian kernels: $k(x, x') = \exp(-\|x - x'\|^2/2\sigma^2)$ are chose, and all the parameters of SVDD and Gaussian kernels are set to default values, i.e. σ = 5, ν = 0.05 in all experiments mentioned.

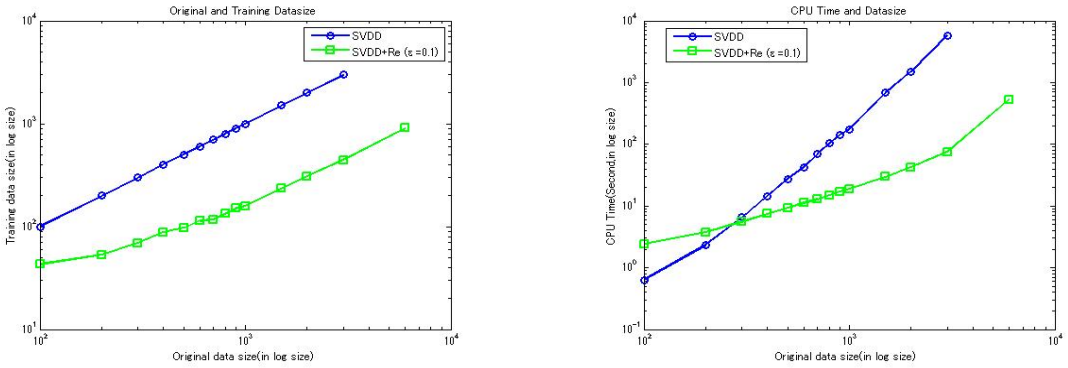
A. Artificial Data

Firstly, we investigate 10 independent experiment results on two artificial data sets: circle shaped and banana shaped data sets. We use training data sets with different sizes from 100 to 6,000, and the amounts of testing data here are same with corresponding training data. Based on the simulations, the neighbor number k will be set to 35. Two parameters of edge detection are set to $\delta = 0.3$ and $\varepsilon = 0.1$, which mean that Candidate set and Complement set account for 30% and 10% of training set respectively. And at least 10% of samples will be preserved. And the number of clustering centers detected is 30 which is nonsensitive to the training sizes. Results of circle data are shown in Table I and Fig.3. The third column of Table I presents comparison results of AUC accuracy, we can see that the proposed method preserves AUC accuracy on acceptable levels and shows better sometimes. Training data size and CPU time in terms of percentages are shown in last two columns. When original data sizes are smaller than 300, SVDD runs faster than the proposed method since edge detection and clustering have to run. However, we can obtain much smaller training set and CPU time on large data sets. To understand the effects better, the variant tendencies of training size and time are also described in Fig.3.

TABLE I
COMPARISON RESULTS ON 10 TIMES EXPERIMENTS(CIRCLE DATA,WITH STANDARD DEVIATIONS)

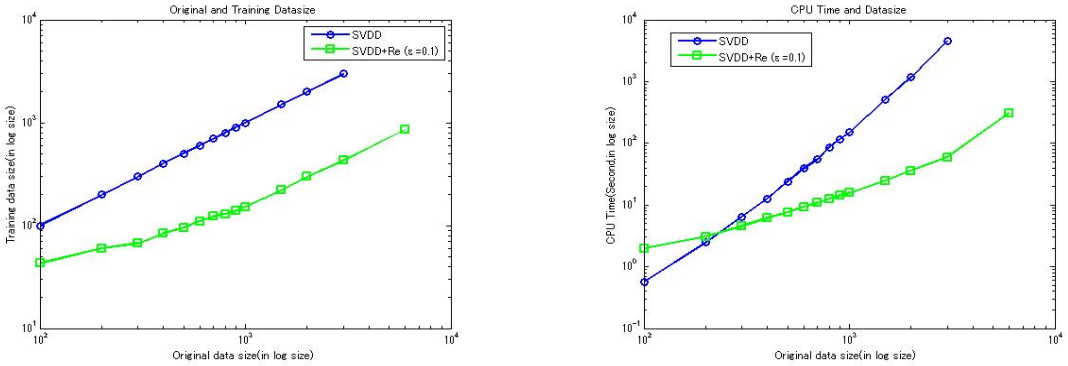
Target	Method	AUC	Size(%)	Time(%)
Cricle100	SVDD	0.9667(0.0004)	100	100
	WSVDD+Re	0.9631(0.0004)	45.00	381.06
Cricle200	SVDD	0.8891(0.0022)	100	100
	WSVDD+Re	0.8945(0.0110)	26.50	160.74
Cricle300	SVDD	0.86840(0.0034)	100	100
	WSVDD+Re	0.8653(0.0035)	23.00	84.34
Cricle400	SVDD	0.9046(0.0062)	100	100
	WSVDD+Re	0.8901(0.0045)	22.20	51.34
Cricle500	SVDD	0.8997(0.0018)	100	100
	WSVDD+Re	0.9003(0.0006)	19.4	33.67
Cricle600	SVDD	0.8890(0.0026)	100	100
	WSVDD+Re	0.8917(0.0098)	19.00	26.45
Cricle700	SVDD	0.8886(0.0020)	100	100
	WSVDD+Re	0.8907(0.0023)	16.71	18.59
Cricle800	SVDD	0.8951(0.0006)	100	100
	WSVDD+Re	0.8934(0.0005)	16.625	14.52
Cricle900	SVDD	0.9043(0.0019)	100	100
	WSVDD+Re	0.9052(0.0012)	16.77	11.87
Cricle1000	SVDD	0.8974(0.0013)	100	100
	WSVDD+Re	0.8988(0.0001)	15.80	10.86
Cricle1500	SVDD	0.8808(0.0006)	100	100
	WSVDD+Re	0.8729(0.0049)	15.64	4.38
Cricle2000	SVDD	0.8861(0.0011)	100	100
	WSVDD+Re	0.8817(0.0038)	15.45	2.84
Cricle3000	SVDD	0.8891 (0.0064)	100	100
	WSVDD+Re	0.8886(0.0034)	14.43	1.75
Cricle6000	SVDD	*	*	*
	WSVDD+Re	0.8931(0.0010)	15.18	*

Similar results are obtained on banana shaped data sets. As shown in Table II and Fig.4, variations of training size and CPU time follow the same tendencies with circle data sets. About AUC accuracy, our method takes better values with most volumes of data. Therefore, the proposed method can successfully obtain reconstruction set with much smaller



(a) Training size (b) CPU time

Fig. 3. Comparision between SVDD and proposed method(circle data sets)



(a) Training size (b) CPU time

Fig. 4. Comparision between SVDD and proposed method(banana data sets)

size and the decision classifier trained on it can achieve comparable accuracy. It must be added that SVDD cannot be operated on our machines when training sizes surpass 6,000. The symbol "*" appeared in Table I and II indicates that there's no result in that block .

B. Real World Data

We then conduct simulations on two real world data sets, MNIST handwrite data set [26] and Abalone class data from UCI machine learning repository [27]. Abalone data set contains $4,177$ instances and is used to the prediction of abalone ages from physical measurements. A common preprocess step is to treat Abalone problem as a 3-category classification problem by grouping ring classes 1-8, 9 and 10, 11-29. 70% percent of target class are used as original training set. MNIST is a high dimensional data set which has a handwritten digit from 0 to 9 with 784 pixel in total as the features. The task is to classify a digit as one of the 10 categories. Digits 1, 2, 3 are chose as targets, all other digits as negative class. MNIST has a training set of $60,000$ examples, and testing set consists of $10,000$ patterns. Since SVDD can not get results on three digits with more than $4,000$ training size, we just choose $4,000$ target digits for experiments. The kernel we used here is still Gaussian kernel. Clustering number and the neighbor number k will be set to 30 and 35 respectively.

The comparison results between the proposed method and weighted SVDD are presented in Table III. Parameters of edge detector presented in the first column with different data sets. First three rows presents three class problems about Abalone data. The proposed approach successfully decreased the training size and preserved the accuracy with much faster speed. In the last three rows of Table III, similar results prove that our fast training method using edge detection can obtain desirable performance on large MNIST data set.

V. Conclusions

In this paper, we proposed a fast SVDD training method using edge detection to improve the time complexity problem of SVDD on large data sets. For better edge detection, an Density-Angle based edge detector was presented. Edge points and clustering centers were extracted and merged to the reconstruction training set with small size. And weighted SVDD was also adapted to defend the influences of noises. Experiments on a number of artificial and real-world data sets have shown its performances in terms of training size, training time and AUC accuracy. As shown in the results, our method trained on reduced training set could spend much less CPU time and get comparative AUC accuracy or even better than SVDD in some cases.

TABLE II
COMPARISON RESULTS ON 10 TIMES EXPERIMENTS(BANANA DATA,WITH STANDARD DEVIATIONS)

Target	Method	AUC	Size(%)	Time(%)
Banana100	SVDD	0.9992(0.0211)	100	100
	WSVDD+Re	0.9996(0.0032)	43.00	436.96
Banana200	SVDD	0.9321(0.0569)	100	100
	WSVDD+Re	0.9847(0.0124)	30.10	116.85
Banana300	SVDD	0.8959(0.0196)	100	100
	WSVDD+Re	0.9915(0.0135)	22.33	59.95
Banana400	SVDD	0.9088(0.0372)	100	100
	WSVDD+Re	0.9941(0.0033)	21.00	48.82
Banana500	SVDD	0.8839(0.0363)	100	100
	WSVDD+Re	0.9932(0.0029)	19.20	32.92
Banana600	SVDD	0.8916(0.0191)	100	100
	WSVDD+Re	0.9960(0.0145)	18.33	18.37
Banana700	SVDD	0.9199(0.0061)	100	100
	WSVDD+Re	0.9916(0.0208)	17.57	20.35
Banana800	SVDD	0.9063(0.0228)	100	100
	WSVDD+Re	0.9948(0.002)	16.23	15.88
Banana900	SVDD	0.9052(0.0124)	100	100
	WSVDD+Re	0.9893(0.001)	15.44	12.67
Banana1000	SVDD	0.9046(0.0117)	100	100
	WSVDD+Re	0.9797(0.0168)	15.20	10.43
Banana1500	SVDD	0.9038(0.0092)	100	100
	WSVDD+Re	0.8974(0.0027)	14.66	4.38
Banana2000	SVDD	0.9078(0.0035)	100	100
	WSVDD+Re	0.8966(0.0275)	15.00	5.45
Banana3000	SVDD	0.8963(0.0023)	100	100
	WSVDD+Re	0.8894(0.0046)	14.35	1.65
Banana6000	SVDD	*	*	*
	WSVDD+Re	0.8931(0.0198)	14.32	*

TABLE III
COMPARISON RESULTS ON 10 TIMES EXPERIMENTS(REAL DATA,WITH STANDARD DEVIATIONS)

Target	Method	AUC	Size(%)	Time(%)
Abalone1-8 $\varepsilon = 0.3, \delta = 0.6$	SVDD	0.8515(0.0493)	100	100
	WSVDD+Re	0.8509(0.0063)	56.15	28.44
Abalone9-10 $\varepsilon = 0.1, \delta = 0.4$	SVDD	0.6375(0.0054)	100	100
	WSVDD+Re	0.6551(0.0124)	18.85	2
Abalone11-29 $\varepsilon = 0.4, \delta = 0.5$	SVDD	0.7560(0.0171)	100	100
	WSVDD+Re	0.7563(0.0135)	49.24	17.48
MNIST0 $\varepsilon = 0.1, \delta = 0.3$	SVDD	0.9843(0.0015)	100	100
	WSVDD+Re	0.9842(0.0020)	26.62	3.51
MNIST1 $\varepsilon = 0.1, \delta = 0.3$	SVDD	0.9941(0.0023)	100	100
	WSVDD+Re	0.9949(0.0147)	26.04	3.58
MNIST2 $\varepsilon = 0.1, \delta = 0.3$	SVDD	0.8197(0.0031)	100	100
	WSVDD+Re	0.8199(0.0027)	26.42	3.97

REFERENCES

[1] D. M. Tax, "One-class classification," 2001.

[2] D. M. Tax and R. P. Duin, "Uniform object generation for optimizing one-class classifiers," *The Journal of Machine Learning Research*, vol. 2, pp. 155–173, 2002.

[3] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[5] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[6] H. Hoffmann, "Kernel pca for novelty detection," *Pattern Recognition*, vol. 40, no. 3, pp. 863–874, 2007.

[7] V. N. Vapnik, "Statistical learning theory," 1998.

[8] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[9] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection." in *NIPS*, vol. 12, 1999, pp. 582–588.

[10] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[11] W. Sakla, A. Chan, J. Ji, and A. Sakla, "An svdd-based algorithm for target detection in hyperspectral imagery," *Geoscience and Remote Sensing Letters, IEEE*, vol. 8, no. 2, pp. 384–388, 2011.

[12] I. Kang, M. K. Jeong, and D. Kong, "A differentiated one-class classification method with applications to intrusion detection," *Expert Systems with Applications*, vol. 39, no. 4, pp. 3899–3905, 2012.

[13] Y. Chen, X. S. Zhou, and T. S. Huang, "One-class svm for learning in image retrieval," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 1. IEEE, 2001, pp. 34–37.

[14] J. Platt *et al.*, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.

[15] C. S. Chu, I. W. Tsang, and J. T. Kwok, "Scaling up support vector data description by using core-sets," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 1. IEEE, 2004.

[16] B. Li, Q. Wang, and J. Hu, "A fast svm training method for very large datasets," in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. IEEE, 2009, pp. 1784–1789.

[17] H. Shin and S. Cho, "Invariance of neighborhood relation under input space to feature space mapping," *Pattern recognition letters*, vol. 26, no. 6, pp. 707–718, 2005.

[18] Y. Li, "Selecting training points for one-class support vector machines," *Pattern recognition letters*, vol. 32, no. 11, pp. 1517–1522, 2011.

[19] L. Sanyang, L. Jinjin, W. De, and D. Wei, "Confidence support vector domain description," *Systems Engineering and Electronics, Journal of*, vol. 20, no. 4, pp. 852–857, 2009.

[20] Y.-H. Liu, Y.-C. Liu, and Y.-J. Chen, "Fast support vector data descriptions for novelty detection," *Neural Networks, IEEE Transactions on*, vol. 21, no. 8, pp. 1296–1313, 2010.

[21] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.

[22] B. Schölkopf and A. J. Smola, *Learning with kernels*. The MIT Press, 2002.

[23] K. Teknomo, "K-means clustering tutorial," *Medicine*, vol. 100, no. 4, p. 3, 2006.

[24] L. Bo, Y. Jie, X. Y. Shan, C. Longbing, and Y. Philip, "Exploiting local data uncertainty to boost global outlier detection," 2010.

[25] D. Tax, "Ddtools, the data description toolbox for matlab (2012)," *Software available at http://prlab. tudelft. nl/david-tax/dd_tools. html*.

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[27] C. Blake and C. J. Merz, "{UCI} repository of machine learning databases," 1998.