2009 IEEE International Conference on Data Mining Workshops

Multi-sphere Support Vector Data Description for Outliers Detection on Multi-Distribution Data

Yanshan Xiao, Bo Liu*, Longbing Cao
University of Technology, Sydney

Xindong Wu
University of Vermont

Chengqi Zhang
University of Technology, Sydney

Zhifeng Hao
GuangDong University of Technology

Fengzhao Yang, Jie Cao
E-Business Department
Nanjing University of Finance and Economics

Abstract

SVDD has been proved a powerful tool for outlier detection. However, in detecting outliers on multi-distribution data, namely there are distinctive distributions in the data, it is very challenging for SVDD to generate a hyper-sphere for distinguishing outliers from normal data. Even if such a hyper-sphere can be identified, its performance is usually not good enough. This paper proposes an multi-sphere SVDD approach, named MS-SVDD, for outlier detection on multi-distribution data. First, an adaptive sphere detection method is proposed to detect data distributions in the dataset. The data is partitioned in terms of the identified data distributions, and the corresponding SVDD classifiers are constructed separately. Substantial experiments on both artificial and real-world datasets have demonstrated that the proposed approach outperforms original SVDD.[1]

1 Introduction

Support Vector Data Description (SVDD) [1] is one of the best known support vector machine learning methods [2] for one-class classification problems. In SVDD, the training data is mapped from the input space into a higher dimensional feature space via a kernel function [2, 3]. The classifier is learnt to obtain an optimal hypersphere boundary by enclosing the target data. The hypersphere is then considered as a descriptive classifier to classify data into the target or non-target class. To date, SVDD has been successfully applied in many practical problems, such as time-series novelty detection [4], windows registry accesses [6], audio signal segmentation [5] and image retrieval [7].

In the formulation of SVDD, each data point is assumed to come from a single uniform distribution function $F(\mathbf{x})$ [1]. In this case, kernel function [3] is strong enough to make the data compact and spherical in the feature space guaranteeing high performance of the hypersphere classifier. However, this hypothesis cannot always hold true, because in real-life applications, the data always comes from distinctive distribution functions [7][2]. For multi-distribution data, each distribution of data samples occupies its individual area in feature space after kernel function is utilized. Therefore, SVDD will reduce its performance if single hypersphere is constructed to enclose the data from multi-distribution functions, because some insider outliers will be mistaken as target class, shown in Figure 1.

In order to improve the learning ability of SVDD, this paper proposes a novel method, called multi-sphere SVDD (MS-SVDD), for outlier detection on multi-distribution data. Our framework is divided into two key steps, i.e., (1) identify each data distribution in the dataset; (2) establish SVDD for each data distribution and construct multi-SVDD classifier for future prediction. Following our framework, a novel and effective method, called "Adaptive Sphere Detection" (ASD), is proposed to detect distinctive data distributions in the dataset. ASD can not only automatically detect the number of data distributions, but also determines the data structure of dataset by classifying each data point into its belonging distribution. Substantial experiments on both artificial and real-world datasets have shown that the proposed approach outperforms SVDD which encloses distinctively distributed data into a single hypersphere.

[1] Acknowledge the financial support of the Capital Markets CRC
* corresponding author: csbliu@it.uts.edu.au

[2] we denote these distribution functions as $F_1(\mathbf{x}), F_2(\mathbf{x}), \ldots, F_M(\mathbf{x})$, where M is the number of distribution functions.

978-0-7695-3902-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ICDMW.2009.87

The rest of the paper is organized as follows: Section 2 discusses the limitation of SVDD. Section 3 put forwards our approach MS-SVDD. Experiments on both artificial and real-world datasets are presented in Section 4. Section 5 concludes this paper and gives future work.

2 Limitation of SVDD on Multi- Distribution Data

In SVDD, the normal data is assumed to be from the same distribution function $F(\mathbf{x})$. In this case, kernel function can be strong enough to make the normal data compact in the feature space so that the hypersphere can be constructed as a classifier with high performance. However, this hypothesis can not always hold true, since the normal data may come from distinctive distribution functions, i.e., $F_1(\mathbf{x})$, $F_2(\mathbf{x}), \ldots, F_M(\mathbf{x})$, in practical applications.

Take an example of fraud detection in insurance for further understanding. Here, we discover that people at different stages of age have different behaviors in both *normal* and *fraud*, and the age stages could be [20, 30], [31, 45], [46, 65]. The *behavior pattern* of policy holders at the same age stage comes from one distribution function in a statistical view. Then all the *behavior patterns* are from three distribution functions. Taking these multi-distribution data into account, the hypothesis is transformed into: let the training set be $S = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l$, where $\mathbf{x}_i \in R^n$, and $\mathbf{x}_i$ comes from one of the distribution functions, i.e, $F_1(\mathbf{x}), F_2(\mathbf{x}), \ldots, F_M(\mathbf{x})$, where M represents the number of the distribution functions.

SVDD determines a tight hypersphere without considering the possibility of multi-distribution data, and presents a superior performance when the data really comes from the same distribution function. For multi-distribution data, we illustrate that the performance of SVDD will be depressed by taking Figure 1 for understanding.

Assume the data denoted by "$\times$", "$\circ$" and "Δ" comes from three distribution functions $F_1(\mathbf{x})$, $F_2(\mathbf{x})$, and $F_3(\mathbf{x})$ respectively. Meanwhile, each of the three sub-spheres cover the normal data from the same distribution function, and the other data outside them is denoted as the *outliers*. From the purpose of outlier detection, all the *outliers* from three distribution functions must be determined. However, if we just perform SVDD to confirm the classification boundary, only *outside outliers* can be identified, and all the *inside outliers* will be misclassified as normal data.

3 Multi-sphere SVDD on Multi-distribution Data

In order to enhance the detection capability of SVDD on *inside outliers*, this paper proposes an improved SVDD solution. The main idea of our approach is: (1) identify the domains of each distribution data in the training set. (2) establish corresponding one-class classifier for data from the same distribution, and combine multiple one-class classifiers for future prediction.

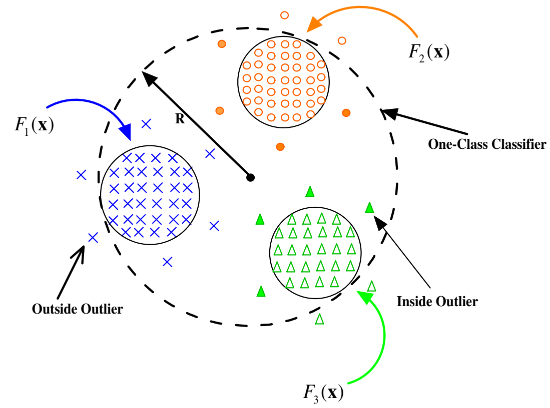

Figure 1. Illustration of SVDD for multi-distribution data in feature space

3.1 Determination of Data Distribution

For multi-distribution data, SVDD cannot identify the "inside outliers". Therefore, each data distribution must be determined in the dataset before performing SVDD. To do this, a novel ASD method is proposed as follows.

3.1.1 Initialization

At first, the *solid sphere* is defined as follows.

Definition 1: (**Solid sphere**) $B(C, R, N)$ is a "*solid sphere*" on the condition that data point size N inside the sphere with the radius of R and the center of C is more than a threshold N^0.

At the first step, we intend to detect the relatively dense areas in S by determining the solid spheres, and C, R and N^0 are initialized as follows. For the R and N^0, R should be a small value, and $MinDis_S \leq R \leq MaxDis_S$ must satisfy, and N^0 had better be a large value. This setting can help us to find some dense areas in S. However, if there exist no this kind of spheres, the solid spheres can be determined by enlarging R or reducing N^0.

For initialization of C, which is the center of sphere at the seeding step, we restrict every data point in the dataset to be C. The sphere is then denoted as $B(\phi(x_i), R, N), i = 1, 2, \ldots, l$.

Let $\phi(\mathbf{x}_i), i = 1, 2, \ldots, l$ be the center of the sphere, and scan the row $d_{i,j}, j = 1, 2, \ldots, l$ or the column $d_{j,i}, j =$

Table 1. Notion Definition.

Symbol	Definition
1. N^0	Threshold for sphere seeding
2. $d(\mathbf{x}_i, \mathbf{x}_j)$	Distance between $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$. $d(\mathbf{x}_i, \mathbf{x}_j)$= $\sqrt{K(\mathbf{x}_i, \mathbf{x}_i,) + K(\mathbf{x}_j, \mathbf{x}_j) - 2K(\mathbf{x}_i, \mathbf{x}_j)}$
3. S_k	kth sub-set
4. $\|S_k\|$	Sample size of S_k S_k={ $\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_{\|S_k\|}^k$ }
5. $\mathbf{C}_k$	Centroid of S_k $\mathbf{C}_k = \frac{1}{\|S_k\|} \sum_{i=1}^{S_k} \phi(\mathbf{x}_i^k)$
6. $COV(S_k)$	Variance of S_k $COV(S_k) = \frac{1}{\|S_k\|} \sum_{i=1}^{\|S_k\|} (\phi(\mathbf{x}_i^k) - \mathbf{C}_k)^2$
7. $D(X_i, \mathbf{C}_k)$	Distance between $\phi(\mathbf{x}_i)$ and $\mathbf{C}_k$ $D(X_i, C_k) = \\| \phi(\mathbf{x}_i) - \mathbf{C}_k \\|$
8. $MinDis_{S_k}$	Minimum distance of two data points in S_k $MinDis_{S_k}$ = $Min\\| \phi(\mathbf{x}_i^k) - \phi(\mathbf{x}_j^k) \\|, i, j = 1, \ldots, \|S_k\|,$
9. $MaxDis_{S_k}$	Maximum distance of two data points in S_k $MaxDis_{S_k}$= $Max\\| \phi(\mathbf{x}_i^k) - \phi(\mathbf{x}_j^k) \\|, i, j = 1, \ldots, \|S_k\|$
10. $B(C, R, N)$	Sphere with the centroid C and radius R, sample size in sphere is N.
11. $B_k^t(C_k(t), R_k(t), N_k(t))$	kth sphere after growing for $t-1$ times, $C_k(t)$, $R_k(t), N_k(t))$ represent centroid, radius and data points size enclosed in the sphere. $C_k(t)$ is calculated according to notion 5.
12. $B_k^t(., ., .)$	Abbreviation of $B_k^t(C_k(t), R_k(t), N_k(t))$
13. $\triangle R_k(t)$	Increment of radius $\triangle R_k(t) = R_k(t+1) - R_k(t)$
14. $MinDis_{B_i^t(.,.,.)}$	Minimum distances of two data points in $B_i^t(., ., .)$, obtained according to notion 8.
15. $COV(B_k^t(., .))$	Variance of $B_k^t(., ., .)$, can be calculated according to notion 6.
16. η	Parameter
17. $N_{B_i(.,.,.) \cap B_k(.,.,.)}$	Sample size of data points belonging to $B_i(., ., .)$ and $B_k(., ., .)$ together
18. λ_0	Threshold for spheres merging
19. $B_k^{end}(., ., .,)$	Abbreviation of $B_k^{end}(C_k(end), R_k(end), N_k(end)$
20. $D(\mathbf{x}, C_k(end))$	Distance between $\phi(\mathbf{x}_i)$ and $C_k(end)$, which can be calculated according to notion 7.
21. $D(\mathbf{x}, B_k^{end}(., ., .))$	Distance between $\phi(\mathbf{x}_i)$ $andB_k^{end}(., ., .)$ $D(\mathbf{x}, B_k^{end}(., ., .))$= $D(\mathbf{x}, C_k(end)) - R_k(end)$
22. R_j	Radius of jth one-class classifier.
23. $\mathbf{o}_j$	Centroid of jth one-class classifier.
24. C_j	Parameter of jth one-class classifier.

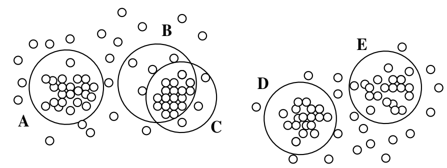

Figure 2. Initialization

$1, 2, \ldots, l$ to confirm which data points are covered in the sphere, i.e., which data point satisfies $d_{j,i} \leq R$ or $d_{i,j} \leq R$.

After this, assume g_0 solid spheres have been determined in S as follows.

$$B_1(\phi(\mathbf{x}_{k_1}), R_1, N_1), \ldots, B_i(\phi(\mathbf{x}_{k_i}), R_i, N_i), \ldots, B_{g_0}(\phi(\mathbf{x}_{k_{g_0}}), R_{g_0}, N_{g_0}), \quad (1)$$

where

$$\mathbf{x}_{k_i} \in S, \quad R_1 = R_2 = \cdots = R_{g_0}, \quad N_i \geq N^0 \quad i = 1, \ldots, g_0. \quad (2)$$

Let us consider Figure 2 for example. Five solid spheres, represented by A, B, C, D and E, have been determined in the data set at the initialization.

3.1.2 Sphere Growing

Although the solid spheres represent some dense areas in S, it is not enough. Because we want not only to find the high-density areas, but also to identify the multi-distribution data, we need to enlarge the spheres radius to catch more data points from the same distribution function.

We first present the following formula for sphere growing strategy of $B_k^t(., ., .)$.

$$R_k(t+1) = R_k(t) + \triangle R_k(t) \quad (3)$$

$$\triangle R_k(t) = f(R_k(t), COV_k(t), N_k(t)) \quad (4)$$

$f(.)$ is a function related to $R_k(t), COV_k(t), N_k(t)$ of $B_k^t(., ., .)$. Assume $B_k^t(., ., .)$ and $B_p^t(., ., .)$ are two spheres to grow, we have the following three rules.

Rule 1: If $R_k(t) = R_p(t)$, $COV_k(t) = COV_p(t)$ yet $N_k(t) > N_p(t)$ holds true, then we let $\triangle R_k(t) > \triangle R_p(t)$.

Rule 2: If $N_k(t) = N_p(t)$, $COV_k(t) = COV_p(t)$ and $R_k(t) < R_p(t)$ holds true, and then we let $\triangle R_k(t) > \triangle R_p(t)$.

Rule 3: If $N_k(t) = N_p(t)$, $R_k(t) = R_p(t)$ but $COV_k(t) < COV_p(t)$, then we let $\triangle R_k(t) < \triangle R_p(t)$.

Let

$$MR(t) = Min\{R_i(t)\}, i = 1, 2, \ldots, g_e \quad (5)$$

$$MN(t) = Max\{N_i(t)\}, i = 1, 2, \ldots, g_e \quad (6)$$

$$MC(t) = Max\{COV_i(t)\}, i = 1, 2, \ldots, g_e \quad (7)$$

g_e is the number of spheres need to increase their radius at the time of t. We then present the relationship between the elements in f function as follows.

$$\triangle R_k(t) = \frac{MR(t)}{R_k(t)} \frac{N_k(t)}{MN(t)} \frac{COV_k(t)}{MC(t)} \cdot \triangle \tag{8}$$

$$\triangle = \frac{1}{g_e} \sum_{i=1}^{g_e} MinDis_{B_i^t(.,.,.)} \tag{9}$$

From (9), we can see that each solid sphere $B_k^t(.,.,.)$ $k = 1, 2, \ldots, g_e$ grows individually. We let the centroid $C_k(t)$ and $R_k(t+1)$ to be the new sphere center and radius, then the new sphere $B_k^{t+1}(C_k(t+1), R_k(t+1), N_k(t+1))$ is obtained.

3.1.3 Sphere Stopping

In this method, there exist two cases to stop sphere growing. After $B_k^t(.,.,.)$ increases the radius by $\triangle R_k(t)$, some new data points will be included by the enlarged sphere $B_k^{t+1}(.,.,.)$. We denote them as

$$\mathbf{x}_{N_1}^k, \mathbf{x}_{N_2}^k, \ldots, \mathbf{x}_{N_h}^k, \quad h = N_k(t+1) - N_k(t). \tag{10}$$

If all of the data points $\mathbf{x}_{N_i}^k$, $i = 1, 2 \ldots, h$ have already been covered by other sphere, i.e.,

$$\forall i, \ \exists \ j \ \ and \ \ tt \longrightarrow \mathbf{x}_{N_i}^k \in B_j^{tt}(.,.,.)$$
$$1 \leq j \leq g, \ \ 1 \leq tt \leq t, \ \ i = 1, \ldots, h, \tag{11}$$

g is the number of spheres at time of t. Then $B_k^t(.,.,.)$ stops.

Another case is if

$$\frac{R_k(t+1) - R_k(t)}{N_k(t+1) - N_k(t)} < \eta \frac{R_k(t) - R_k(t-1)}{N_k(t) - N_k(t-1)} \tag{12}$$

holds true, then $B_k^t(.,.,.)$ stops. This setting can forbid the sphere to cover the data points from other distribution function.

3.1.4 Sphere Merger

Assume at time of $t+1$, all the spheres stop growing and g_m spheres are acquired, i.e.,

$$B_1^{t_1}(.,.,.), \ldots, B_i^{t_i}(.,.,.), \ldots, B_{g_m}^{t_{gm}}(.,.,.),$$
$$1 \leq i \leq g_m \ t_i \in \{1, 2, \ldots, t+1\}. \tag{13}$$

Because more than one solid spheres may be generated within one data distribution, we then automatically merge the spheres belonging to the same distribution as follows. Consider sphere $B_i^{t_i}(.,.,.)$ and $B_k^{t_k}(.,.,.)$, and let

$$N_{ik} = Min\{N_i(t_i), N_k(t_k)\}, \ t_i, t_k \in \{1, 2, \ldots, t+1\}, \tag{14}$$

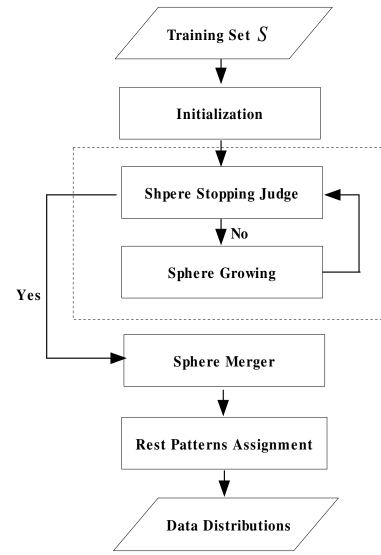

Figure 3. The work flow of ASD method

then, we obtain

$$\lambda_{ik} = \frac{N_{B_i^{t_i}(.,.,.) \cap B_k^{t_k}(.,.,.)}}{N_{ik}}. \tag{15}$$

If λ_{ik} is larger than a threshold λ_0, merge $B_i^{t_i}(.,.,.)$ and $B_k^{t_k}(.,.,.)$. In this way, all the spheres covering the same data distribution merge, and distinctive data distributions are detected automatically.

3.1.5 Rest Data Assignment

After the procedures of sphere growing, stopping and merger, g_f spheres have been determined in the dataset S, i. e.,

$$B_1^{end}(C_1(end), R_1(end), N_1(end)), \ldots,$$
$$B_{g_f}^{end}(C_{g_f}(end), R_{g_f}(end), N_{g_f}(end)). \tag{16}$$

If some data points are not included in one of these g_f spheres, they will be assigned into one sphere as follows. For data point $\mathbf{x}$, it is assigned into

$$\underset{i=1,\ldots,g_f}{arg \ min}(D(\mathbf{x}, B_i^{end}(.,.,.)). \tag{17}$$

The work flow of "ASD" method is illustrated in Figure 3.

3.1.6 Discussion Of "ASD" Method

ASD method can not only detect the relatively dense areas in the dataset, but also identify the multi-distribution data through enlarging and merging these spheres. ASD can automatically determine the number of data distributions according to the data structure.

It is noted that some clustering algorithms, such as K-mean and density clusterings, can also be used to determine multi-distribution data, but they are subject to some constraints. In K-Mean , K should be known beforehand. Density clustering always detects the highest dense areas [8].

3.2 Establish SVDD Classifier

After g_f spheres have been determined, they are put into sub-sets $S_1, S_2, \ldots, S_{g_f}$, respectively. The following task is how to utilize the data in each sub-set to construct the classifiers for outlier detection. We propose a multi-sphere approach to solve this problem, as follows.

The purpose of kernel function $K(.,.)$ in SVDD is to render the data much more compact in feature space, and then the hypersphere can be constructed as the classifier. When the data comes from different distributions, if all the data is enclosed by one hypersphere, the classification accuracy will be greatly decreased because some internal outliers are neglected inside the hypersphere. Therefore, our approach is proposed to utilize multi-spheres covering the sub-datasets. By doing this, the whole QP function can be transferred into a series of sub-QP problems, as follows:

$$\begin{aligned}
&min \quad R_j^2 + C_j \sum_{i=1}^{|S_j|} \xi_i^j \\
&s.t. \| \phi_j(\mathbf{x}_i^j) - \mathbf{o}_j \|^2 \le R_j^2 + \xi_i^j, \quad i = 1, 2, \ldots, |S_j|, \\
&\xi_i^j \ge 0, \ \ \mathbf{x}_i^j \in S_j \\
&i = 1, 2, \ldots, |S_j|, \ \ j = 1, 2, \ldots, g_f. \qquad (18)
\end{aligned}$$

After solving these sub-QP problems, we can obtain the hypersphere center O_j and radius R_j of each sub-classifier. For an unknown data, it is regarded as a normal data when $R_j^2 = \| \phi_j(\mathbf{x}_k^j) - \mathbf{o}_j \|^2$. By contraries, **x** is classified as an **outlier**.

4 Experiments

In this section, we conduct the experiments on both artificial and real-world datasets to investigate the performance of our proposed approach. In addition, K-mean and density clustering methods are also considered to detect the different distributions of data.

All the programs are written in Matlab 7.0, and the SVDD is implemented by using the SVM-KM toolbox [3]. The RBF kernel is used in our experiments, because RBF kernel function always offers a better performance than other kernel functions [2]. The RBF kernel induces an infinite-dimensional kernel space, where the kernel width parameter controls the scaling of the mapping, i.e.,

$$K(\mathbf{x}, \mathbf{x}_i) = exp(-\| \mathbf{x} - \mathbf{x}_i \|_2^2 / 2\sigma^2). \qquad (19)$$

For RBF kernel function, the value of parameter σ is chosen in $\{0.1, 0.2, 0.3, 0.4\}$, and C is in the range of $\{0.0001, 0.001, 0.01, 0.1\}$. Sixty percents of data points are randomly selected for training, and the rest are regarded as testing set. Grid search method [9] is used to determine the kernel parameters.

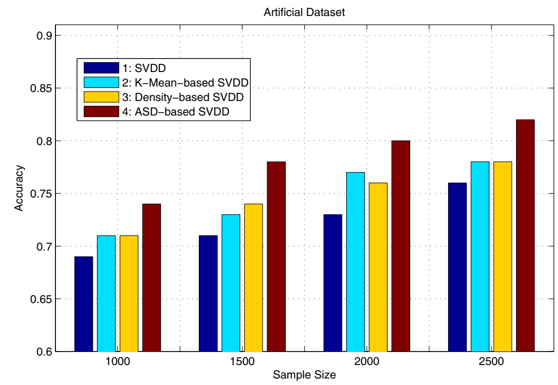

Figure 4. Artificial Dataset

4.1 Artificial Dataset

The artificial datasets are generated from 3 normal distribution functions with 5 attributes. The sample sizes are 1000, 1500, 2000 and 2500, respectively, and outliers from each class are generated at the same time. Because the samples of datasets come from three distribution functions, we set K to 3 for K-mean method.

The results of SVDD, K-mean-based SVDD and density clustering-based SVDD are illustrated in Figure 4. It is clear that, the three clustering approaches can work better than original SVDD. That is to say, after the distributions of data are determined using no matter K-mean, density method or ASD, the accuracies have been greatly improved compared to enclosing the whole dataset in a single hypersphere. Additionally, compared with K-mean and density clustering method, our ASD method always achieves superior performance.

4.2 Real-world dataset

We utilize a real-world underground gas pipelines (UGS) dataset to compare the performance of our approach and SVDD. The UGS dataset consists of 306 data points for underground gas pipelines and each data point is described

[3] http://asi.insa-rouen.fr/enseignants/arakotom/toolbox/index.html

using nine numerical attribute: Pipe wall thickness (PWT), Coating Type (CT), Buried Year (BY), The Number of Years of Operation (TNYO), Coating Resistance (CR), Leakage Point Line Density (LPLD), Soil Corrosion (SC), Anode or Not (AN), Potential. In the dataset, all the gas pipelines are classified into five categories, i.e., from level 1 to level 5. If a gas pipeline belongs to level 4 or 5, that means it must be repaired.

For the purpose of one-class classification, we follow the operation in [1] to obtain five sub-datasets in our experiments. Specifically, take each of the five classes out of the dataset, and the other four classes are considered as target class. After this operation, five sub-datasets are achieved: UGS (1), UGS (2), UGS (3), UGS (4) and UGS (5). Here, the number in the bracket represents the class which is taken out of the dataset. What's more, for the data coming from the same distribution function, the outlier patterns are drawn from a Gaussian distribution with zero mean and standard deviation is added into some patterns in the dataset according to [10]. The outlier is added as a vector with the same number of dimensions as the dataset has. The standard deviation σ_i of the entire data along the dimension i is first obtained. In order to model the difference in outlier information on different dimensions, we define the standard deviation σ_i along dimension i, whose value is randomly drawn from the range $[0, 2\rho \cdot \sigma_i]$. In the experiment ρ is set as 1.5, Then, for dimension i, we add outlier from a random distribution with standard deviation σ_i. By this way, pattern $\mathbf{x}_j$ is added the outlier and regarded as an outlier pattern.

The accuracy of SVDD, K-mean-base SVDD, density-based SVDD and ASD-based SVDD in feature space are shown from Figure 5. We have the following observations:

(1) Our proposed framework can handle the multi-distribution data efficiently, after we use K-mean, density clustering method or ASD methods to determine the distribution of data in the dataset. The performance is improved compared with original SVDD.

(2) Compared with K-mean and density clustering method, our ASD method always achieves better performance.

5 Conclusions and Future Work

In SVDD for outlier detection, the performance will be reduced when handling multi-distribution data. In order to address this issue, our approach is proposed to detect outliers on multi-distribution data. Following the framework, an ASD method is proposed to determine the data distributions within the dataset, and then we propose multi-sphere method to enclose the detected distributions of data, respectively. Sufficient experiments on both artificial and real-world datasets show that our approach outperforms original SVDD.

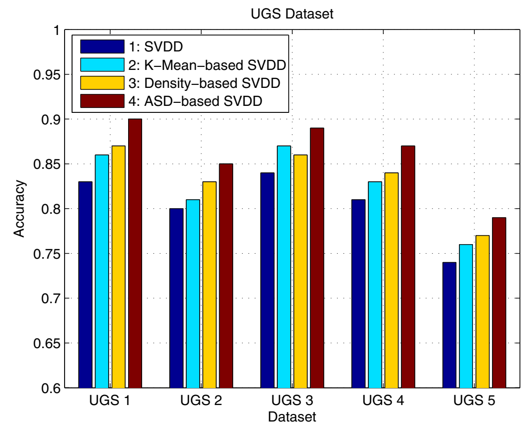

Figure 5. UGS Dataset

In the future, we will investigate on-line SVDD for multi-distribution data.

References

[1] D. Tax and R. Duin, "Support vector data description", *Machine Learning*., vol. 54, no. 1, pp. 45-66, Jan. 2004.

[2] V. N. Vapnik, "The Nature of Statistical Learning Theory", Springer-Verlag, London, UK, 1995.

[3] S. Saitoh, "Theory of reproducing kernels and its applications", Harlow, U.K.: Longman, 1988.

[4] J. Ma, and S. Perkins, "Time-series novelty detection using one-class support vector machines", *In Proceedings of the International Joint Conference on Neural Networks*, Vol. 3, pp. 1741 1745, 2003.

[5] M. Davy and S. Godsill, "Detection of abrupt spectral changes using support vector machines. an application to audio signal segmentation", *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002.

[6] K. A.Heller, K. M.Svore, A. D.Keromytis and S. J.Stolfo, "One Class Support Vector Machines for Detecting Anomalous Windows Registry Accesses", *In Proceedings of the workshop on Data Mining for Computer Security*, 2003.

[7] Y. Q. Chen, X. S. Zhou, T. S. Huang "one-class SVMs for learning in image retrieval", *ICIP*, pp. 34-37, 2001.

[8] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", *In Proceedings of ACM SSIGKDD*, pp. 226-231, 1999.

[9] C. W. Hsu "A comparison of methods for multiclass support vector machines", *IEEE Transactions on Neural Networks*., vol. 13, pp. 415-425, 2002.

[10] C. C. Aggarwal, P. Yu, "A framework for clustering uncertain data streams", *in Proc. ICDE08*, pp. 150-159, 2008.