International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:7, No:8, 2013

Kernel's Parameter Selection for Support Vector Domain Description

Mohamed EL Boujnouni, Mohamed Jedra, Noureddine Zahid

***Abstract*—**Support Vector Domain Description (SVDD) is one of the best-known one-class support vector learning methods, in which one tries the strategy of using balls defined on the feature space in order to distinguish a set of normal data from all other possible abnormal objects. As all kernel-based learning algorithms its performance depends heavily on the proper choice of the kernel parameter. This paper proposes a new approach to select kernel's parameter based on maximizing the distance between both gravity centers of normal and abnormal classes, and at the same time minimizing the variance within each class. The performance of the proposed algorithm is evaluated on several benchmarks. The experimental results demonstrate the feasibility and the effectiveness of the presented method.

***Keywords*—**Gravity centers, Kernel's parameter, Support Vector Domain Description, Variance.

I. Introduction

THE SVDD is kind of one-class classification method based on Support Vector Machine [1], which is proposed by Tax [2]-[4]. It tries to construct a boundary around the target data by enclosing the target data within a minimum hyper-sphere. Inspired by the support vector machines (SVMs), the SVDD decision boundary is described by a few target objects, known as support vectors (SVs). A more flexible boundary can be obtained with the introduction of kernel functions [5], [6], by which data are mapped into a high-dimensional feature space. The most commonly used kernel function is Gaussian kernel. This method has attracted many researchers from various fields. For example Liu et al. applied the SVDD techniques for novelty detection as part of the validation on an Intelligent Flight Control System (IFCS) [8]. Ji et al. discussed the SVDD application in gene expression data clustering [9]. Yu et al used SVDD for image categorization from internet images [10].

The performance of kernel methods strictly depends on their hyper parameters, especially the kernel parameters that directly control the non linear mapping of the features. Therefore, the tuning of parameters, known also as the model selection, plays an important role in kernel methods.

In the literature, there are two widely used approaches in choosing the values of kernel parameters in kernel-based methods [11]-[13]. The first approach empirically chooses a series of candidate values for the kernel parameter, executes the concerned method under these values again and again, and selects the one corresponding to the best performance as the final kernel parameter value. However, this approach suffers from the fact that only a very limited candidate values are considered, therefore the performance of the kernel-based methods may not be optimized. The second approach is the well-known cross-validation, which is also widely used in model selection. Compared with the first approach, cross-validation often yields better performance because it searches the optimal value for kernel parameter in a much wider range. However, performing cross-validation is often time-consuming and hence it cannot be used to adjust the kernel parameters in real time [12]. Furthermore, when there are only a limited number of training examples, the cross-validation approach can hardly ensure robust estimation. Another approach is to minimize some generalization bounds, such as the leave-one-out (LOO) error bounds, using numerical optimization methods [20], [14]. The numerical methods are generally more efficient than grid search. However, owing to the non-convexity of the generalization bounds, these methods may get stuck into local optimum and cause instabilities [17], [19]. Recently, some global stochastic optimization techniques, such as genetic algorithm (GA), particles warm optimization (PSO) and simulated annealing (SA) algorithm have been adopted to tune the SVM parameters for their better global search abilities [15], [18]. These methods, although can find the global solution in a high probability, are limited by the facts that they usually suffer from the problem of premature convergence, the slow convergence rate and the convergence to a single point [21].

In this paper we aim to find a feature space, in which the objects of each cluster are well separated. To do that we propose a new numerical optimization methods defined as the maximization of the distance between both gravity centers, of normal and abnormal classes and at the same time the minimization of the variance of each class in feature space.

To evaluate our approach, we run our algorithm on SVDD, we focus on optimizing the Gaussian kernel since it is widely used in pattern recognition, neural network and other fields, and shows good features and strong learning capability.

The rest of this paper is organized as follows. In Section II the theory behind the Support Vector Domain Description is presented. Section III gives a detailed description of our approach. In the last section we give several experiments results to show the validity of our proposed algorithm.

Mohamed EL Boujnouni, Mohamed Jedra, and Noureddine Zahid are with Faculty of Sciences, Mohammed V – Agdal University, Laboratory of Conception and Systems (Microelectronic and Informatics) Avenue Ibn Battouta B.P 1014, Rabat, Morocco (e-mail: med_elbouj@yahoo.fr, jedra@fsr.ac.ma, zahid@fsr.ac.ma).

II. Support Vector Data Description (SVDD)

The normal data description model [2]-[3], gives a closed boundary around the data: a hypersphere characterized by center a and radius R > 0. It minimizes the volume of the sphere by minimizing R^2, and demand that the sphere contains all training objects x_i.

Let $\{x_i\} \in \chi$ be a data set of N points, with, $x_i \in R^d$ the data space, we look for the smallest enclosing sphere of radius R which is described by the following constraints:

$$\|x_j - a\|^2 \leq R^2 \quad \forall j \tag{1}$$

where $\|.\|$ is the Euclidean norm. Soft constraints are incorporated by adding slack real and positive variable ε_j:

$$\|x_j - a\|^2 \leq R^2 + \varepsilon_j \quad \forall j \tag{2}$$

To solve this problem we introduce the Lagrangian:

$$L = R^2 - \sum_j \left(R^2 + \varepsilon_j - \|x_j - a\|^2\right)\alpha_j - \sum_j \varepsilon_j \mu_j + C\sum_j \varepsilon_j \tag{3}$$

where $\alpha_j \geq 0$ and $\mu_j \geq 0$ are Lagrange multipliers, C is a constant, and $C\sum_j \varepsilon_j$ is a penalty term. Setting the partial derivatives of L with respect to R, a, ε_i to zero gives the following constraints:

$$\frac{\partial L}{\partial R} = 0 \Rightarrow \sum_i \alpha_i = 1 \tag{4}$$

$$\frac{\partial L}{\partial a} = 0 \Rightarrow a = \sum_i \alpha_i x_i \tag{5}$$

$$\frac{\partial L}{\partial \varepsilon_i} = 0 \Rightarrow \alpha_i = C - \mu_i \tag{6}$$

The solution of the primal problem can be obtained by solving its dual problem [2].

Max:

$$W = \sum_j x_j^2 \alpha_j - \sum_{i,j} \alpha_i \alpha_j x_i x_j \tag{7}$$

$$Subject\ to\ \ 0 \leq \alpha_j \leq C\ \ \forall j\ \ and \sum_j a_j = 1$$

When negative examples (objects which should be rejected) are available, they can be incorporated in the training to improve the description. In contrast with the training (target) examples which should be within the sphere, the negative examples should be outside it. In the following, the target objects are enumerated by indices *i, j* and the negative examples by *l, m*. Again, we allow for errors in both the target and the outliers set and introduce slack real positive variables ε_i and ε_l [2]:

$$L(R, a, \varepsilon_i, \varepsilon_l) = R^2 + C1\sum_i \varepsilon_i + C2\sum_l \varepsilon_l \tag{8}$$

With the constraints:

$$\|x_i - a\|^2 \leq R^2 + \varepsilon_i \quad \|x_l - a\|^2 \geq R^2 - \varepsilon_l \quad \varepsilon_i, \varepsilon_l \geq 0\ \forall i, l$$

where C1, C2 are constants real positives, $C1\sum_i \varepsilon_i$ and, $C2\sum_l \varepsilon_l$ are penalty terms, these constraints are incorporated in (8) and the Lagrange multipliers α_i, α_l ,γ_i, γ_l are introduced as follow:

$$\begin{aligned} L(R, a, \varepsilon_i, \varepsilon_l, \alpha_i, \alpha_l, \gamma_i, \gamma_l) = R^2 + C1\sum_i \varepsilon_i + C2\sum_l \varepsilon_l - \sum_i \gamma_i \varepsilon_i - \sum_l \gamma_l \varepsilon_l \\ - \sum_i \alpha_i [R^2 + \varepsilon_i - \|x_i - a\|^2] \\ - \sum_l \alpha_l [\|x_l - a\|^2 - R^2 + \varepsilon_l] \end{aligned} \tag{9}$$

with $\alpha_i \geq 0, \alpha_l \geq 0, \gamma_i \geq 0, \gamma_l \geq 0$ are Lagrange multipliers. Setting the partial derivatives of L with respect to *R, a,* ε_i and ε_l to zero gives the following constraints:

$$\frac{\partial L}{\partial R} = 0 \Rightarrow \sum_i \alpha_i - \sum_l \alpha_l = 1 \tag{10}$$

$$\frac{\partial L}{\partial a} = 0 \Rightarrow a = \sum_i \alpha_i x_i - \sum_l \alpha_l x_l \tag{11}$$

$$\frac{\partial L}{\partial \varepsilon_i} = 0\ and\ \ \frac{\partial L}{\partial \varepsilon_l} = 0\ \Rightarrow\ \alpha_i = C1 - \gamma_i\ \ \alpha_l = C2 - \gamma_l\ \ \forall i, l \tag{12}$$

When (10) and (11) are substituted into (9) we obtain:

Max

$$W = \sum_i \alpha_i x_i x_i - \sum_l \alpha_l x_l x_l - \sum_{i,j} \alpha_i \alpha_j x_i x_j + 2\sum_{l,j} \alpha_l \alpha_j x_l x_j - \sum_{l,m} \alpha_l \alpha_m x_l x_m$$

Subject to: $0 \leq \alpha_i \leq C1\ \ and\ 0 \leq \alpha_l \leq C2\ \ \forall i, l$ (13)

$$\sum_i \alpha_i - \sum_l \alpha_l = 1$$

The formulations of SVDD can be extended to obtain a more flexible description. Data is mapped nonlinearly into a higher dimensional space where a hyperspherical description can be found. The mapping is performed implicitly, replacing all of the inner products by a kernel function K (x_i, x_j) [2], [3]. Table I describes some commonly used kernel functions.

TABLE I
Some Commonly Used Kernel Functions

Gaussian Radial Basis Function (RBF)	$k(x,y) = e^{\left(-(x-y)^2/2\sigma^2\right)}$
Exponential Radial Basis Function	$k(x,y) = e^{\left(-\lvert x-y\rvert/2\sigma^2\right)}$
Hyperbolic Tangent	$k(x,y) = \tanh(b(x,y) + c)$
Polynomial	$k(x,y) = (1 + x^T.x)^p$
Fourier Series	$k(x,y) = \dfrac{\sin\left(\delta + \frac{1}{2}\right)(x-y)}{\sin\left(\frac{1}{2}(x-y)\right)}$
Two-layer perception	$Tanh(s_0 x^T.x_i + s_1)$

For multiclass problems, to classify a test point *z*, we just investigate whether it is inside the hypersphere (a_k,R_k) constructed during the training and associated to the class *k* [2], [3], [7]. Namely the decision function is calculated as (14), if its value is positive for the k^{th} class and negative for the others we conclude that *z* belong to the class *k*.

$$f(z) = sgn(R_k^2 - \|z - a_k\|^2) \quad (14)$$

This function can be calculated as follows:

In the normal data description case we obtain:

$$\|z - a_k\|^2 = z.z - 2\sum_i \alpha_{ki} x_i z + \sum_{i,j} \alpha_{ki}\alpha_{kj} x_i x_j \quad (15)$$

$$R_k^2 = x.x - 2\sum_i \alpha_{ki} x_i x + \sum_{i,j} \alpha_{ki}\alpha_{kj} x_i x_j \quad (16)$$

with α_{kj} is the j^{th} Lagrangian multiplier corresponding to the k^{th} class. And $x \in SV$ the set of Support Vectors having $0 < \alpha_i < C$.

In the SVDD with negative examples case we obtain:

$$\| z - a_k \|^2 = z.z - 2\left(\sum_i \alpha_{ki} x_i z - \sum_l \alpha_{kl} x_l z\right) + \sum_{i,j} \alpha_{ki}\alpha_{kj} x_i x_j + \sum_{l,m} \alpha_{kl}\alpha_{km} x_l x_m - 2\sum_{i,l} \alpha_{ki}\alpha_{kl} x_i x_l \quad (17)$$

$$R_k^2 = x.x - 2\left(\sum_i \alpha_{ki} x_i x - \sum_l \alpha_{kl} x_l x\right) + \sum_{i,j} \alpha_{ki}\alpha_{kj} x_i x_j + \sum_{l,m} \alpha_{kl}\alpha_{km} x_l x_m - 2\sum_{i,l} \alpha_{ki}\alpha_{kl} x_i x_l \quad (18)$$

For any $x \in SV$ the set of support vectors having $0 < \alpha_i < C1$ (with x is a target object) or $0 < \alpha_l < C2$ (with x is negative object).

III. Optimization of Gaussian Kernel

A. Generalization Ability of a Classifier

Generalization is the ability that a trained model predicts the target value of an input sample which is not in the training set. Many indexes can be used to assess the generalization ability [27]. For example, the training process of the grid search uses the validation accuracy to indicate the generalization ability of the classifier, when the validation data are not available, k-fold cross validation can be used to acquire the validation accuracy [22].

Other indexes that estimate the generalization ability can be used. Takahashi [23], proposed the ratio of the numbers of SVs to the training samples as an index. Phetkaew [24] suggested using the SVM margin to identify a classifier that causes wrong classifications, Wu and Wang [25] introduced a separation index which indicates the separation of two classes in the feature space. The index is derived from inter-cluster distances δ4 which was used by Bezdek [26], for unsupervised data clustering. Bezdek and Pal mentioned several inter-cluster distance measures δ_i. They are the measurements of the distance between two clusters.

$$\delta_1(X_+, X_-) = \min_{\substack{x_+ \in X_+ \\ x_- \in X_-}} \mathrm{d}(x_+, x_-) \quad (19)$$

$$\delta_2(X_+, X_-) = \max_{\substack{x_+ \in X_+ \\ x_- \in X_-}} \mathrm{d}(x_+, x_-) \quad (20)$$

$$\delta_3(X_+, X_-) = \frac{1}{l_+ l_-} \sum_{\substack{x_+ \in X_+ \\ x_- \in X_-}} \mathrm{d}(x_+, x_-) \quad (21)$$

$$\delta_4(X_+, X_-) = \mathrm{d}(\overline{x_+}, \overline{x_-}) = \mathrm{d}\left(\frac{\sum_{x_+ \in X_+} x_+}{l_+}, \frac{\sum_{x_+ \in X_+} x_+}{l_-}\right) \quad (22)$$

$$\delta_5(X_+, X_-) = \frac{1}{l_+ + l_-}\left(\sum_{x_+ \in X_+} d(x_+, \overline{x_-}) + \sum_{x_- \in X_-} d(x_-, \overline{x_+})\right) \quad (23)$$

where X_+ and X_- are positive and negative classes, l_+ and l_- are sample sizes of X_+ and X_-, and $\overline{x_+}$and $\overline{x_-}$are the class means of X_+ and X_-. δ_1, δ2 and δ3 are the shortest, the longest and the average distance between two samples from different classes. δ_4 is the distance between two class means, and δ_5 is a combination of δ_3 and δ_4.

B. Our Approach

As mentioned previously our goal is to find a feature space induced by a Gaussian kernel, in which the objects of each cluster are well separated, to do that we will introduce a new separation index based on δ3 and on the variances within-class.

Contrarily to the approach mentioned by [27], who use those indexes to evaluate the generalization ability through grid search method, we will use our new index to calculate an optimal parameter of Gaussian kernel, by maximizing an objective function defined by (24).

In what follows a detailed description of our proposed algorithm is presented.

$$F(\sigma) = \left\| \frac{1}{N}\sum_{i=1}^{N} \Phi(\mathrm{x_i}) - \frac{1}{M}\sum_{k=1}^{M} \Phi(\mathrm{x_k}) \right\| - \beta\left(\frac{1}{N^2}\sum_{i}^{N}\sum_{j}^{N} \left\|\Phi(\mathrm{x_i}) - \Phi(\mathrm{x_j})\right\|^2 + \frac{1}{M^2}\sum_{k}^{M}\sum_{l}^{M} \|\Phi(\mathrm{x_k}) - \Phi(\mathrm{x_l})\|^2\right) \quad (24)$$

β is real and positive parameter used to control the variance. After substituting the inner product by RBF kenel, and expanding the equation 24, we obtain the following result:

$$F(\sigma) = \frac{1+2\beta}{N^2}\sum_{i}^{N}\sum_{j}^{N} \mathrm{e}^{\frac{-\|\mathrm{x_i}-\mathrm{x_j}\|^2}{2\sigma^2}} - \frac{2}{NM}\sum_{i}^{N}\sum_{k}^{M} \mathrm{e}^{\frac{-\|\mathrm{x_i}-\mathrm{x_k}\|^2}{2\sigma^2}} + \frac{1+2\beta}{M^2}\sum_{k}^{M}\sum_{l}^{M} \mathrm{e}^{\frac{-\|\mathrm{x_k}-\mathrm{x_l}\|^2}{2\sigma^2}} - 4\beta$$

$$F(\sigma) = \frac{1+2\beta}{N^2}\left(N + 2.\sum_{i}^{N-1}\sum_{i<j}^{N} \mathrm{e}^{\frac{-\|\mathrm{x_i}-\mathrm{x_j}\|^2}{2\sigma^2}}\right) - \frac{2}{NM}\sum_{i}^{N}\sum_{k}^{M} \mathrm{e}^{\frac{-\|\mathrm{x_i}-\mathrm{x_k}\|^2}{2\sigma^2}} + \frac{1+2\beta}{M^2}\left(M + 2.\sum_{k}^{M-1}\sum_{k<l}^{M} \mathrm{e}^{\frac{-\|\mathrm{x_k}-\mathrm{x_l}\|^2}{2\sigma^2}}\right) - 4\beta \quad (25)$$

The derivative of F(σ) with respect to σ:

$$\frac{dF(\sigma)}{d\sigma} = \frac{2(1+2\beta)}{N^2}\left(\sum_{i}^{N-1}\sum_{i<j}^{N}\frac{\|x_i - x_j\|^2}{\sigma^3} e^{\frac{-\|x_i-x_j\|^2}{2\sigma^2}}\right) - \frac{2}{NM}\sum_{i}^{N}\sum_{k}^{M}\frac{\|x_i - x_k\|^2}{\sigma^3} e^{\frac{-\|x_i-x_k\|^2}{2\sigma^2}} + \frac{2(1+2\beta)}{M^2}\left(\sum_{k}^{M-1}\sum_{k<l}^{M}\frac{\|x_k - x_l\|^2}{\sigma^3} e^{\frac{-\|x_k-x_l\|^2}{2\sigma^2}}\right) \quad (26)$$

The optimal values of the kernel parameters can be obtained through maximizing (25), i.e.

$$\sigma^* = \underset{\sigma}{argmax} F(\sigma) \quad (27)$$

In this paper, an iterative algorithm is employed to generate σ^*, According to the general gradient method, the updating equation for minimizing the objective function *F(σ) is* given by :

$$\sigma^{(n+1)} = \sigma^{(n)} + \eta\left(\frac{\partial F}{\partial \sigma}\right) \quad (28)$$

Where η is the learning rate and n is the iteration step. Our proposed method is summarized as follows [28]:

Step 1. Choose the value of β. Set the learning rate η , the maximum iteration number *N*, and ε to a very small positive number.

Step 2. Initialize the kernel parameters $\sigma = \sigma^{(0)}$and set the iteration step n = 0.

Step 3. Update the kernel parameters $\sigma^{(n)}$ using (28).

Step 4. If $\left|\sigma^{(n+1)} - \sigma^{(n)}\right| < \varepsilon$ or $n \geq N$ stop otherwise, set n=n+1, goto step 3.

IV. Experimental Results

A. Datasets and Experimental Setting

Before conducting our experiments on real datasets, we begin with two artificial ones; which are chessboard and Double-Spiral.

We fixe β at 0.03, then we calculate the optimal values of Gaussian width (σ*) using our proposed algorithm, after we classify the datasets through SVDD with C=200 and σ = σ*.

It can be seen in Fig. 1 that the two classes are well separated. More specifically, we observe that all elements corresponding to the first class appear in white, while those corresponding to the second class appear in gray.

We remark also that there are no overlaps between both classes, and the coloration follows the distribution of classes, this means that the data from different classes are projected successfully onto a suitable feature space, (implicitly a good value of σ).

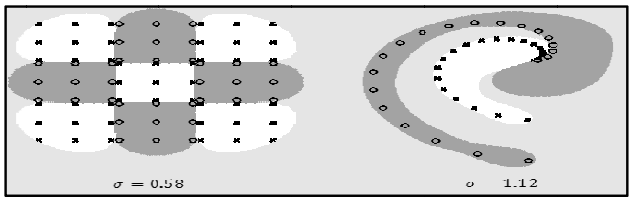

Fig. 1 Clustering results on Chessboard and Double-Spiral datasets, using SVDD with (σ*) found by our approach

To investigate the success of these results on real datasets, we conducted various tests in which our algorithm is applied on monk-1, monk-2, monk-3, iris flowers, wine, ionosphere; all of these datasets are taken from [16], further details of these datasets are provided in Table II.

TABLE II
Description of the Datasets Used in the Experiment, Training Samples and Testing Samples List the Rate of Data Used or Directly the Files Containing Data

dataset	Number of data	subset		Number of class	Feature
		Training set	Testing set		
monks	432	monks-1.train	monks-1.test	2	6
	432	monks-2.train	monks-2.test	2	6
	432	monks-3.train	monks-3.test	2	6
iris	150	80% of samples/each class	The remaining samples/each class	3	4
wine	178	80% of samples/each class	The remaining samples/each class	3	13
Ionosphere	351	The first 200 instances	The remaining 150 instances	2	34

Firstly, the three problems defined for monk's dataset were used in the experiment; monks-1 is in standard disjunctive normal form and is supposed to be easily learnable by most of the algorithms and decision trees. Conversely, monk's-2 is similar to parity problems. It combines different attributes in a way that makes it complicated to describe using the given attributes only; monks-3 serves to evaluate the algorithms under the presence of noise.

Secondly, the iris dataset consists of three classes, each of which has 50 samples. While one cluster is easily separable, it is difficult to achieve separation between the other two clusters. Data points correspond to the plants and attributes correspond to sepal and petal measurements.

Thirdly, the wine dataset is the results of a chemical analysis of wines grown in the same region but derived from three different cultivars. The analysis determines the quantities of constituents found in each of the three types of wines.

Fourthly, the Ionosphere dataset is a radar data; it consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets were free electrons in the ionosphere.

For monk's problem we use the files monks-(1, 2, 3).train, as training set and their corresponding files monks-(1, 2,

3).test, as testing set. Concerning Ionosphere, we train SVDD with the first 200 instances, which were split 50% positive and 50% negative. We use the remaining 150 instances as testing set.

For iris, and wine, datasets, we randomly split each one into 20 subsets, each subset contains training and testing sets, with the scheme described in Table II. Training and test sets do not intersect.

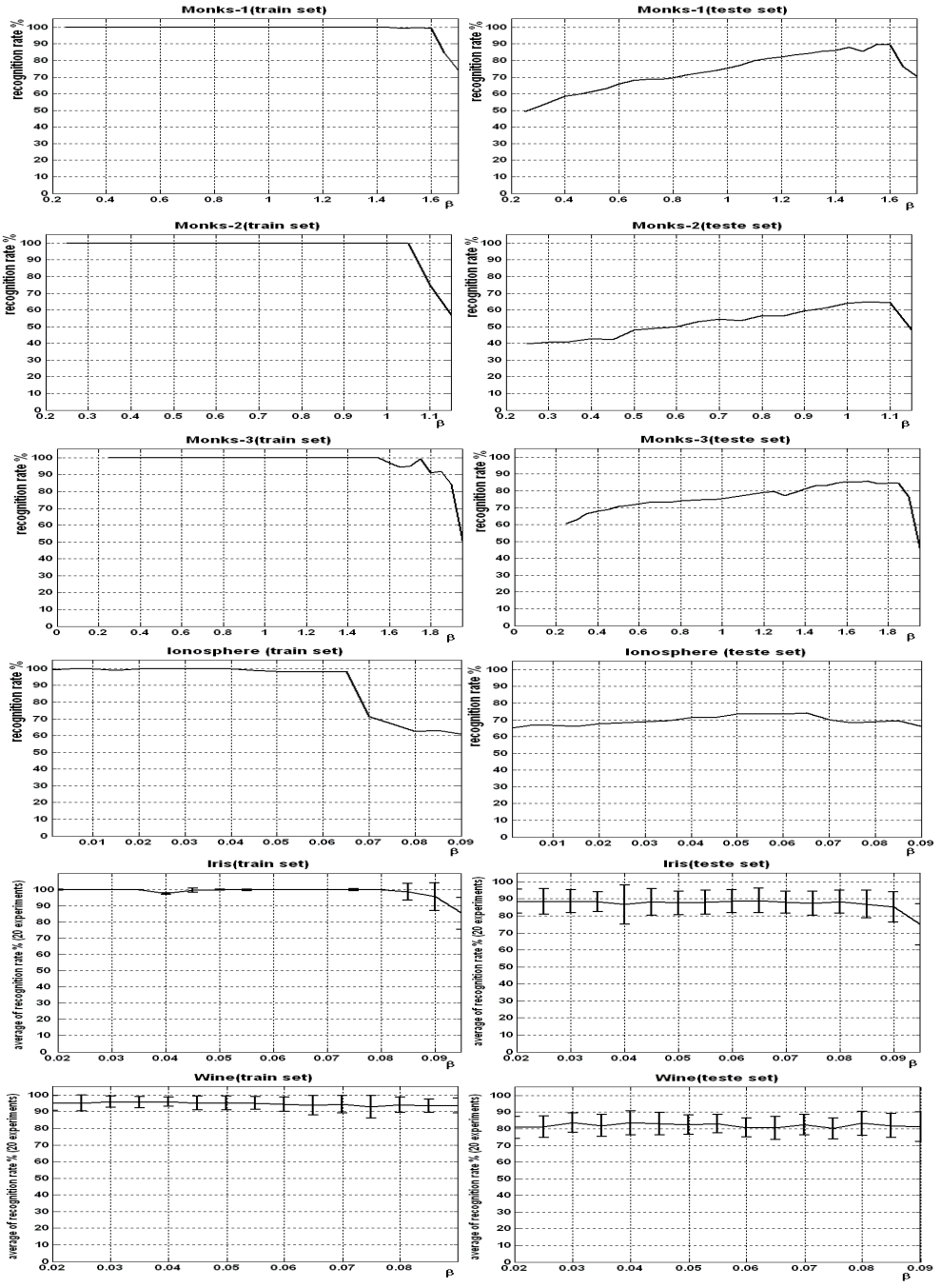

Fig. 2 Recognition rates (%) for the selected datasets, using the optimal values of Gaussian width (σ*) found by our approach, for different values of the parameter β

B. Numerical Results

In all experiments we fix C= 200 and we use one versus all method. For each dataset from monks-1, monks-2, monks-3, Ionosphere, iris, and wine: after setting the value of β, we run the algorithm described above to find the optimal value of σ for each class. Using that value, the algorithm SVDD will be trained by the training set and then, tested by the training and the corresponding testing set.

In the case of the Monk s (1,2,3) and Ionosphere dataset, we just calculate the recognition rate directly, for both training and testing set. For iris, and wine we repeat this experiment 20 times for all subsets and we calculate the mean and the standard deviation of the recognition rate. The results are shown in Fig. 2.

Fig. 2 shows that a good choose of β, which imply an optimal compromise between the distance inter cluster and the variance within cluster, gives a good value of gaussian width (σ*), which achieves an important classification rate.

V. Conclusion

In this paper, a novel approach for learning the kernel parameters is proposed and successfully applied to the SVDD classifier. An optimal value of the Gaussian kernel width is obtained by maximizing the distance between the gravity centers of both normal and abnormal clusters, and at the same time minimizing the variance of both clusters. The performance of the proposed algorithm is evaluated on two artificial datasets and six benchmark datasets from UCI repository [16]. The experimental results for different datasets show that our method achieves good performance.

References

[1] B. H. Asa, David Horn, T. H. Siegelmann, Vladimir. Vapnik, "Support vector clustering", *Journal of Machine Learning Research*. Vol. 2, No. 12, pp.125-137, 2001.

[2] D. Tax, R. Duin, "Data Domain Description Using Support Vectors". *Proceedings- European Symposium on Artificial Neural Networks Bruges*, pp 251-256, (Belgium, 1999a).

[3] D. Tax, R. Duin, "Support vector domain description". *Pattern Recognition Letters*, Vol. 20, No. 11–13, pp. 1191–1199, 1999b.

[4] D. Tax, R. Duin, "Support Vector Data Description", *Machine Learning*. Vol.54, pp.45–66, 2004.

[5] K. Lee, D.W Kim, D. Lee, and K. H Lee. "Improving support vector data description using local density degree". *Pattern Recognition*, Vol.38, No. 10, pp. 1768 – 1771, 2005.

[6] B. Schölkopf, A.J. Smola. "Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond". *Cambridge, Mass: MIT Press, London*, 2002.

[7] K.-R. Mäller, S. Mika, G. Rätsch, K. Tsuda, B. Schölkopf, "An introduction to kernel-based learning algorithms", *IEEE Transactions on Neural Networks*, Vol. 12, No. 2 , pp.181-201, 2001.

[8] Y. Liu, S. Gururajan, B. Cukic, T. Menzies, and M. Napolitano, "Validating an Online Adaptive System Using SVDD," in *15th IEEE Int. Conf. on Tools with Artificial Intelligence*, 2003.

[9] R. Ji, D. Liu, M. Wu, and J. Liu, "The Application of SVDD in Gene Expression Data Clustering," in *2nd Int. Conf. on Bioinformatics and Biomedical Engineering*, pp. 371-374, 2008.

[10] X. Yu, D. DeMenthon, and D. Doermann, "Support Vector Data Description for Image Categorization From Internet Images," in *19th Int. Conf. on Pattern Recognition*, 2008.

[11] J. Peng, D.R. Heisterkamp, H.K. Dai, "Adaptive quasiconformal kernel nearest neighbor classification", *IEEE Trans. PAMI* ,Vol. 26, No. 5, pp. 656-661, 2004.

[12] L. Wang, K.L. Chan, "Learning kernel parameters by using class separability measure", *NIPS'02 Workshop on Kernel Machines, Canada*, 2002.

[13] D.Q. Zhang, S.C. Chen, "Clustering incomplete data using kernel-based fuzzy c-means algorithm", *Neural Processing Letters*, Vol. 18, No. 3, pp. 155-162, 2003.

[14] S.S. Keerthi, V. Sindhwani, O. Chapelle, "An efficient method for gradient- based adaptation of hyperparameters in svm models", *NIPS **19*** pp. 673–680, 2007.

[15] A.C. Lorena, A.C.P.L.F. de Carvalho, "Evolutionary tuning of SVM parameter values in multiclass problems", *Neurocomputing,* Vol. 71, No. 16–18, pp. 3326–3334, 2008.

[16] UCI repository of machine learning databases. http://archive.ics.uci.edu/ml/.

[17] K.M. Chung, W.C. Kao, C.L. Sun, L.L. Wang, C.J. Lin, "Radius margin bounds for support vector machines with the RBF kernel", *Neural Comput*. Vol. 15, No. 11, pp. 2643–2681, November. 2003.

[18] S. Lin, Z. Lee, C. Chen, T. Tseng, "Parameter determination of support vector machine and feature selection using simulated annealing approach", *Appl. Soft Comput*. Vol. 8, No. 4, pp. 1505–1512, Sep. 2008.

[19] H. Frohlich, A. Zell, "Efficient parameter selection for support vector machines in classification and regression via model-based global optimization", *In Proc. Int. Joint Conf. Neural Networks*, pp. 1431–1436, 2005.

[20] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, "Choosing multiple parameters for support vector machines", *Mach. Learn*. Vol. 46, No. 1, pp. 131–159, January. 2002.

[21] Li. Shutao, Mingkui Tan, "Tuning SVM parameters by using a hybrid CLPSO–BFGS algorithm", *Neurocomputing.* Vol. 73, No 10–12, pp. 2089–2096, 2010.

[22] C.-W. Hsu, C.-C. Chang, C.-J .Lin,"A practical guide to support vector classification". [Online] Available from World Wide Web: http://www.csie. ntu.edu.tw/~cjlin/libsvm.

[23] F. Takahashi, S. Abe, "Optimizing directed acyclic graph support vector machines", *in: Proceedings of the Artificial Neural Networks in Pattern Recognition (ANNPR 2003)*, pp.166–170, September2003.

[24] T. Phetkaew, B. Kijsirikul, W. Rivepiboon, "Reordering adaptive directed acyclic graphs: an improved algorithm for multiclass support vector machines", *Proceedings of the International Joint Conference on Neural Networks (IJCNN2003)*, Vol. 2, pp.1605–1610, 2003.

[25] K.-P. Wu, S.-D. Wang, "Choosing the kernel parameters of support vector machines according to the inter-cluster distance", *Proceedings of the International Joint Conference on Neural Networks (IJCNN2006)*, 2006.

[26] C. Bezdek, N. R. Pal, "Some new indexes of cluster validity", IEEE Trans. Syst. Man Cybern. *Part B Cybern*. Vol. 28, No. 3, pp. 301–315, 1998.

[27] K.-P. Wu, S.-D. Wang, "Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space", *Pattern Recognition*, Vol. 42, No 5, pp. 710–717, 2009.

[28] D. Zhang, Chen, S., Zhou, Z., Learning the kernel parameters in kernel minimum distance classifier. *Pattern Recognition.* Vol. 39, No. 1, pp 133–135, January 2006.