FEATURE SELECTION OF SUPPORT VECTOR DOMAIN DESCRIPTION USING GAUSSIAN KERNEL

[1]MOHAMED EL BOUJNOUNI, [2]MOHAMED JEDRA, [3]NOUREDDINE ZAHID

[1]Doctoral Student, Conception & Systems Laboratory, FSR, Morocco

[2]Prof, Conception & Systems Laboratory, FSR, Morocco

[3]Prof, Conception & Systems Laboratory, FSR, Morocco

E-mail: [1]med_elbouj@yahoo.fr ,[2] jedra@fsr.ac.ma , [3]zahid@fsr.ac.ma

ABSTRACT

The performance of the kernel-based learning algorithms, such as support vector domain description, depends heavily on the proper choice of the kernel parameter. It is desirable for the kernel machines to work on the optimal kernel parameter that adapts well to the input data and the pattern classification tasks. In this paper we present a novel algorithm to optimize the Gaussian kernel parameter by maximizing a classical class separability criterion, and the problem is solved through a steepest descent algorithm, Simulation results on six benchmark datasets have successfully validated the effectiveness of the proposed method.

Keywords: *Support Vector Domain Description, RBF Width, Class Separability.*

1. INTRODUCTION

In domain description the task is not to distinguish between classes of objects like in clustering problems, or to produce a desired outcome for each input object like in regression problems, but to give a description of a set of objects. This description should be able to distinguish between the classes of objects represented by the training set, and all other possible objects in the object space [1, 2]. Recently, a Support Vector Domain Description (SVDD) (also called One-class Classification), inspired by support vector machine was invented by Tax and Duin [2, 3, 4]. In a SVDD the compact description of target data is given as a hyper sphere with minimal volume containing most of normal data, rejecting most of negative data. It has the possibility of transforming the data to new feature spaces without much extra computational cost using kernel functions [5, 6], see Table 1. However, there is no theoretical method for determining a suitable kernel function. Also, there is no a priori knowledge for setting the kernel parameter. Therefore, choosing an appropriate kernel, which is a model selection problem [7], is crucial to ensure good performance since the geometrical structure of the mapped samples is determined by the selected kernel and its parameters.

The most common and reliable approach to features selection is to decide on parameter ranges, and to then do an exhaustive grid search [14,20] over the parameter space to find the best setting. However, this type of search is a local search and prone to a local optimality. Additionally, setting the search interval is a problem. Too large a search interval wastes computing power, while too small a search interval might render a satisfactory outcome impossible, in addition to the commonly adopted grid search technique, other techniques are used in SVM to improve the possibility of an appropriate choice of parameter values, those techniques can be categorized as filter models and wrapper models [8]. Filter models [8] utilize statistical techniques, such as principal component analysis (PCA), factor analysis (FA), independent component analysis (ICA), and discriminate analysis (DA) in the investigation of other indirect performance measures, mostly based on distance and information measures. Chen and Hsieh [9] presented latent semantic analysis (LSA), Gold et al. [10] developed a Bayesian viewpoint of SVM classifiers to tune hyper-parameter values in order to determine useful criteria for pruning irrelevant features. Chapelle et al. [11] developed an automatically tuning multiple parameters and applied principal components to obtain features for SVM. The wrapper models [12], adopt the accuracy rate of the classifier as the performance measure. Some researchers argue that if the highest predictive accuracy is obtained by minimizing the classifier error rate and equalizing the measurement cost for all features, wrapper models are more suitable. A classifier is constructed with the aim of maximizing the predictive accuracy. The features utilized by the classifier are then selected as the optimal features. The wrapper models often apply meta-heuristic approaches to help in searching for the optimal feature subset. Although meta-heuristic approaches are slow, they produce the (near) optimal feature subset.

In this paper we focus on a filter model technique defined as the optimization of the kernel function. In [11,13], a radius-margin quotient is used as a criterion to tune kernel parameters for the support vector machine (SVM) classifier, and it is applicable to two-class classification problems only. Xiong et al. [15] proposed to optimize a kernel function in the so called empirical feature space by maximizing a class separability measure defined as the ratio between the trace of the between-class scatter matrix and the trace of the within-class scatter matrix, which corresponds to the class separability criterion J4 in [18]. Promising results have been reported on a set of two-class classification problems. Jie Wang et al. [17] proposed a kernel optimization algorithm by maximizing the J1 class separability criterion in [18], defined as the trace of the ratio between the between-class scatter matrix and the within-class scatter matrix. Which is equivalent to the criterion used in the classical Fisher's discriminant analysis [18, 19, 21]. In this paper we propose to maximize a class separability measure defined as the difference between the between-class variance and the within-class variance.

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 optimization is solved using the well-known Steepest Descent algorithm. The results are compared with grid search approach.

The rest of this paper is organized as follows. In Section 2 the theory behind the Support Vector Domain Description is presented. Section 3 gives a detailed description of our approach and the optimization using the Steepest Descent method. In the last section we give several experiments results to show the validity of our proposed algorithm.

2. SUPPORT VECTOR DOMAIN DESCRIPTION (SVDD)

2.1 Normal Data Description

The normal data description model [1, 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:

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

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 constraint we introduce the Lagrangian:

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

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:

$$\partial_R L = 0 \Rightarrow \sum_i \alpha_i = 1 \tag{4}$$

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

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

The solution of the primal problem can be obtained by solving its dual form [1, 3]:

Max:

$$W = \sum_j x_j^2 \alpha_j - \sum_{i,j} \alpha_i \alpha_j x_i x_j$$
$$\textit{Subject to :} \tag{7}$$
$$0 \leq \alpha_j \leq C \quad \forall j \;\; and \;\; \sum_j \alpha_j = 1$$

2.2. SVDD with Negative Examples

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 [1, 3]:

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

with the constraints:

$$\|x_i - a\|^2 \le R^2 + \varepsilon_i \qquad \|x_l - a\|^2 \ge R^2 - \varepsilon_l \qquad \varepsilon_i \ge 0 \qquad \varepsilon_l \ge 0 \qquad \forall i,l$$

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

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

With $\alpha_i \ge 0, \alpha_l \ge 0, \gamma_i \ge 0, \gamma_l \ge 0$ 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:

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

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

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

When Eqs (10),(11) are substituted into Eq (9) we obtain :

Max:

$$W = \sum_i \alpha_i x_i x_i - \sum_l \alpha_l x_l x_l - \sum_{i,j} \alpha_i \alpha_j x_i x_j + 2\sum_{l,j} \alpha_l \alpha_j x_l x_j - \sum_{l,m} \alpha_l \alpha_m x_l x_m$$
$$Subject \quad to \quad : 0 \le \alpha_i \le \mathrm{C1} \quad and \quad 0 \le \alpha_1 \le \mathrm{C2} \quad \forall i,l$$
$$\sum_i \alpha_i - \sum_l \alpha_l = 1 \tag{13}$$

2.3 Decision Function

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* [1, 3].Namely the decision function is calculated as eq(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*.

$$f(z) = \mathrm{sgn}(R_k^2 - \|z - a_k\|^2) \tag{14}$$

Where in the normal data description case:

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

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

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 $0 \prec \alpha_i \prec C$.

And in the SVDD with negative examples case we obtain:

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

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

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

3. OPTIMIZATION OF GAUSSIAN KERNEL

3.1. Flexible Descriptions

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) [1, 3]. Table 1 describes some commonly used kernel functions:

Table 1: Some Commonly Used Kernel Functions

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

3. 2 Our Approach

The objective of SVDD is to find a sphere with minimum volume containing all or most of training objects, and rejecting all or most of negative examples. It‘s obvious that this description become easier if the target examples are close to each other, in the same time the training and the negative examples are distant, it is the principle of discriminate analysis which highlights differences between observations belonging to different classes. Our idea, is to find a future space (σ in the RBF kernel), in which these conditions are the most verified possible.

Our aim is to maximize the between-class variance described by :

$$\frac{1}{NM}\sum_i^N\sum_l^M \left\|\phi(x_i)-\phi(x_l)\right\|^2 \qquad (19)$$

In the same time to minimize the within-class variance described by:

$$\frac{1}{2N(N-1)}\sum_i^N\sum_j^N \left\|\phi(x_i)-\phi(x_j)\right\|^2 \qquad (20)$$

Where the target objects are enumerated by indices *i*, *j* and the negative ones by *l*. N and M define the total number of target and negative objects, respectively.

Combining the two conditions we can maximize:

$$f(\sigma) = \frac{\alpha}{NM}\sum_i^N\sum_l^M \left\|\phi(x_i)-\phi(x_l)\right\|^2 - \frac{\beta}{2N(N-1)}\sum_i^N\sum_j^N \left\|\phi(x_i)-\phi(x_j)\right\|^2 \qquad (21)$$

The role of both parameters α and β is to control maximization and minimization of the two terms 19 and 20 respectively.

After expanding the equation 21 we obtain:

$$f(\sigma) = \frac{\alpha}{NM}.\sum_i^N\sum_l^M \left[\phi(x_i).\phi(x_i) - 2.\phi(x_i).\phi(x_l) + \phi(x_l).\phi(x_l)\right]$$
$$-\frac{\beta}{2N(N-1)}.\sum_i^N\sum_j^N \left[\phi(x_i).\phi(x_i) - 2.\phi(x_i).\phi(x_j) + \phi(x_j).\phi(x_j)\right] \qquad (22)$$

Using the RBF kernel the formula (22) becomes:

max:

$$f(\sigma) = 2.\frac{\alpha}{NM}.\sum_i^N\sum_l^M \left[1-\exp\left(\frac{-\left\|x_i-x_l\right\|^2}{2\sigma^2}\right)\right]$$
$$-\frac{\beta}{N(N-1)}.\sum_i^N\sum_j^N \left[1-\exp\left(\frac{-\left\|x_i-x_j\right\|^2}{2\sigma^2}\right)\right] \qquad (23)$$

$$g(\sigma) = \frac{\partial f(\sigma)}{\partial \sigma} = -2.\frac{\alpha}{NM}.\sum_i^N\sum_l^M \left[\frac{\left\|x_i-x_l\right\|^2}{\sigma^3}.\exp\left(\frac{-\left\|x_i-x_l\right\|^2}{2\sigma^2}\right)\right]$$
$$+\frac{\beta}{N(N-1)}.\sum_i^N\sum_j^N \left[\frac{\left\|x_i-x_j\right\|^2}{\sigma^3}.\exp\left(\frac{-\left\|x_i-x_j\right\|^2}{2\sigma^2}\right)\right] \qquad (24)$$

To maximize f (σ), we use the steepest descent algorithm:

Given an initial σ_0, and a convergence tolerance ε, and the maximum number of iterations MAX

```
for k=0 to MAX
        $\sigma_{k+1} \leftarrow \sigma_k + \alpha_k\, g(\sigma_k)$
    Compute $g(\sigma_{k+1})$
    if $\left\|g(\sigma_{k+1})\right\| < \varepsilon$ then
      converged
   end if
end for
```

4. EXPERIMENTAL RESULTS

4.1 Datasets and Experimental Setting

Six datasets were used to test the Small Sphere and Parametric Volume for SVDD. The datasets describe various characteristics from studying monk-1, monk-2, monk-3, iris flowers, wine, glass; all of these datasets are taken from [16], further details of these datasets are provided in Table 2.

Table 2 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
glass	214	80% of samples/each class	The remaining samples/each class	6	9

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 determined the quantities of constituents found in each of the three types of wines.

Fourthly, the glass dataset is the study of classification of types of glass was motivated by criminological investigation data points correspond to the type of glass and attributes correspond to their oxide content (i.e. Na, Fe, K, etc).

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. For iris, wine, and glass datasets, we randomly split each one into 20 subsets, each subset contains training and testing sets, with the scheme described in Table 3.Training and test sets do not intersect.

4.2 Numerical results

In all experiments we fix C=1000, and we use the one versus all method.

For each dataset from monks-1, monks-2, monks-3, iris, wine, and glass: after setting the values of σ and β, we run the algorithm described above to find the optimal value of (σ) for each class. Using those values, 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 problems, we just calculate the recognition rate directly, for both training and testing set, unlike the remaining datasets where we repeat this experiment 20 times for all subsets and we select the values of σ which gives the median value of the recognition rate.

To prove the efficiency, of our method we run SVDD on monk-1, monk-2, and monk-3, datasets, using a set of discrete values of σ1 and σ2, then we plot the variation of the recognition rate against both σ1 and σ2 (see fig 1).Table 3 shows that when we use the optimal Gaussian width σ found by the proposed algorithm, SVDD gives a good recognition rate for both training and testing set,

Table 3 Recognition Rates (%) For Different Datasets, Using The Optimal Values Of Σ_i Found By The Approach Described Above

data sets	α	β	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	Recognition rate % Train set	Recognitio n rate % Test set
monks-1	0.48	1.00	2.07	5.45					100	83.10
monks-2	0.53	1.00	0.53	1.01					100	66.07
monks-3	0.46	1.00	2.54	2.11					100	80.33
iris	0.49	1.00	1.47	1.32	1.81				100	90.00
wine	0.70	1.00	214.98	153.23	123.90				100	82.35
glass	0.80	1.00	0.42	0.55	0.48	1.56	0.01	1.48	100	55.00

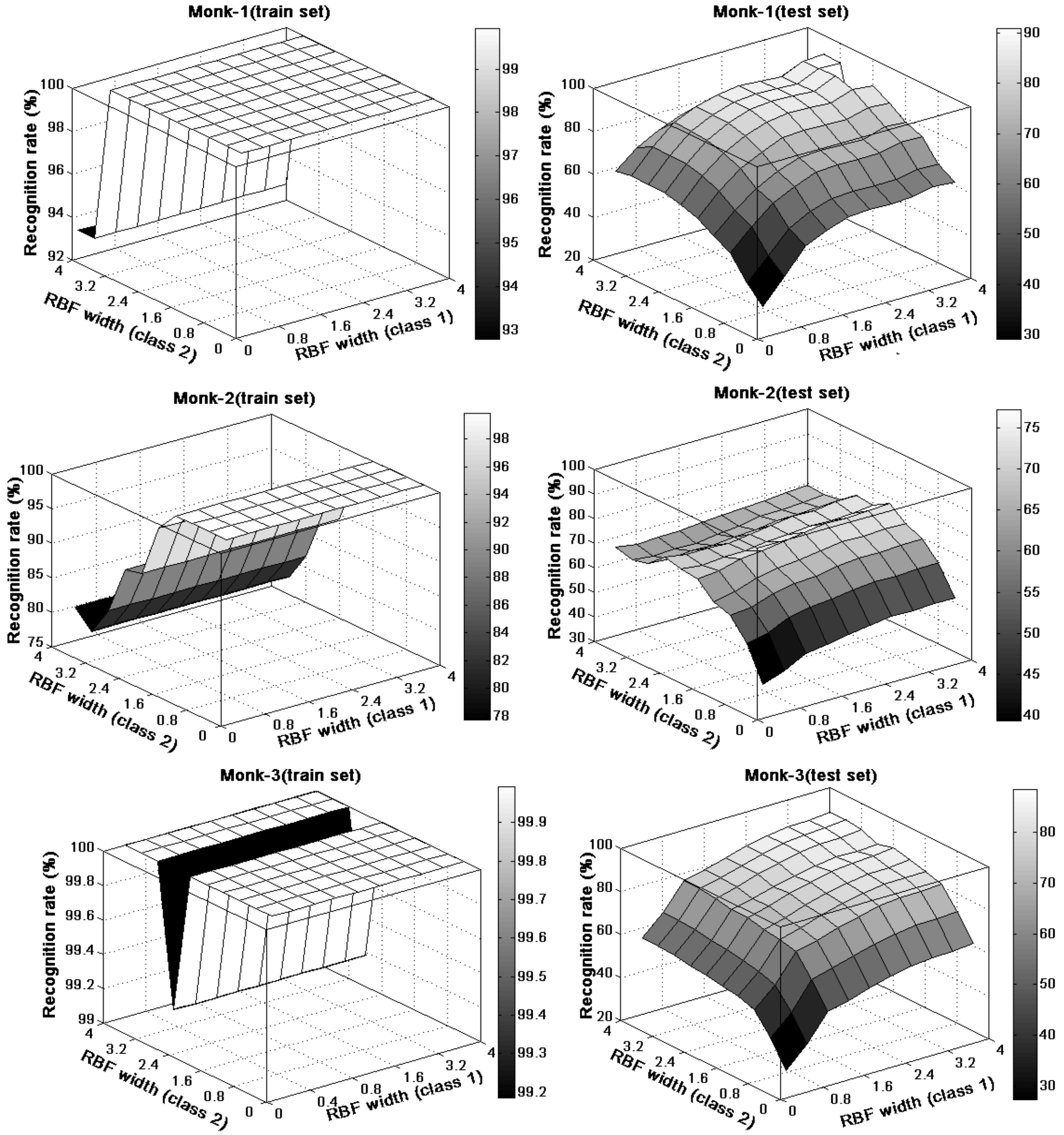

Fig. 1 Recognition Rate For Training And Testing Set, Using Monk-1, Monk-2, And Monk-3, Against A Set Of Discrete Values RBF Width For Both Class 1 And 2.

5. CONCLUSION

We propose an approach for optimizing the kernel parameters, based on a class separability measure, defined as the difference between the variance between-class and the variance within-class. We focus on the Gaussian kernel since it is a widely used and we use the steepest descent algorithm for the optimization. To evaluate the performance of our algorithm, we run the kernel Support Vector Domain Description classifier using the optimal values of σ and we calculate the recognition rate.

By computer simulations using six benchmark data sets, we demonstrated that our method can find optimal values of σ which gives an important recognition rates.

REFERENCES:

[1]Asa Ben-Hur, David Horn, Hava T.Siegelmann, Vladimir Vapnik , “Support vector clustering”,. Journal of Machine Learning Research. Vol. 2, No. 12, 2001, pp.125-137.

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

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

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

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

[6] Schölkopf, B., and 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. 2,2001, pp.181-201.

[8] H. Liu, H. Motoda, “Feature Selection for Knowledge Discovery and Data Mining”, Kluwer Academic, Boston, 1998.

[9] R.-C. Chen, C.-H. Hsieh, “Web page classification based on a support vector machine using a weighed vote schema”, Expert Syst. Appl 31 2006, pp 427–435.

[10] C. Gold, A. Holub, P. Sollich, “Bayesian approach to feature selection and parameter tuning for support vector machine classifiers”, Neural Networks. Vol. 18, No. 5-6, 2005, pp. 693–701.

[11] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, “Choosing multiple parameters for support vector machines”, Mach. Learn. Vol. 46, No. 1-3, 2002, pp.131–159.

[12] R. Kohavi, G.H. John, “Wrappers for feature subset selection”, Artif. Intell, Vol. 97, No.1-2 ,1997, pp. 273–324.

[13] T. Glasmachers, C. Igel, Gradient-based adaptation of general gaussian kernels, Neural Computation, Vol. 17,No. 10, 2005,pp 2099-2105.

[14] C.-W. Hsu, C.-C. Chang, C.-J. Lin, “A practical guide to support vector classification”. Technical Report, University of National Taiwan, Department of Computer Science and Information Engineering , July 2003, pp. 1–12.

[15]H.Xiong,M.N.S.Swamy,M.Omair, “Optimizing the kernel in the empirical feature space”, IEEE Transactions on Neural Networks, Vol. 16, No. 2, 2005, pp460-474.

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

[17] Jie Wang, Haiping Lu, K N Plataniotis, Juwei Lu, “Gaussian kernel optimization for pattern classification”, Pattern Recognition, Vol 42, No. 7, 2009, pp1237-1247.

[18] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd Edition, Academic Press, (Boston, 1990).

[19] P. N. Belhumeur, J. P. Hespanha, D. J. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711-720.

[20] J. Wang, X. Wu, C. Zhang, Support vector machines based on k-means clustering for real time business intelligence systems, Int. J. Business Intell. Data Mining, 1(1) (2005) 54–64.

[21] Shih-Wei Lin , Zne-Jung Lee , Shih-Chieh Chen , Tsung-Yuan Tseng, Parameter determination of support vector machine and feature selection using simulated annealing approach ,Applied Soft Computing 8 (2008) 1505–1512.