International Journal of Fuzzy Logic and Intelligent Systems, vol. 3, no. 1, June 2003 pp. 100-104

One-Class Support Vector Learning and Linear Matrix Inequalities

Jooyoung Park[1], Jinsung Kim[2], Hansung Lee[3], Daihee Park[3]

Dept. of Control & Instrumentation Engineering, Korea University[1],
Dept. of Electrical Engineering, Korea University[2],
Dept. of Computer and Information Science, Korea University[3]

Abstract

The SVDD(support vector data description) is one of the most well-known one-class support vector learning methods, in which one tries the strategy of utilizing balls defined on the kernel feature space in order to distinguish a set of normal data from all other possible abnormal objects. The major concern of this paper is to consider the problem of modifying the SVDD into the direction of utilizing ellipsoids instead of balls in order to enable better classification performance. After a brief review about the original SVDD method, this paper establishes a new method utilizing ellipsoids in feature space, and presents a solution in the form of SDP(semi-definite programming) which is an optimization problem based on linear matrix inequalities.

Key words : One-class classification, Support vector learning, Ellipsoid, Linear matrix inequality, Semi-definite programming

1. Introduction

Recently, the support vector learning method has grown up as a viable tool in the area of intelligent systems.[1,2] Among the important application areas for the support vector learning, we have the one-class classification problems.[2,4-10]

In the problems of one-class classification, we are in general given only the training data for the normal class, and after the training phase is finished, we are required to decide whether each test vector belongs to normal class or abnormal class. The one-class classification problems are often called outlier detection problems or novelty detection problems. Obvious examples of this class include the fault detection for machines and the intrusion detection system for computers.[2] One of the most well-known support vector learning methods for the one-class problems is the SVDD(support vector data description).[4,5] In the SVDD, balls are used for expressing the region for the normal class. Among the methods related with SVDD are the nu one-class SVM of Schölkopf et al.[6,7,8], and the linear programming method of Campbell and Bennet.[9] Since balls in the input domain can express only limited class of regions, the SVDD in general enhances its expressing power by utilizing balls in the feature space instead of the balls on the input domain. However, recently it was shown that even with balls on the feature space, the SVDD still could have some limitations.[9] In this paper, we try to mitigate this limitation by utilizing ellipsoids defined on the feature space. The formulation of the presented method is given in the form of SDP(semi-definite programming), which belongs to a class of LMI(linear matrix inequality)-based optimization problems. As is well-known, SDP problems can be solved efficiently via recently developed interior point methods.[11,12]

The remaining parts of this paper are given in the following order: In Section 2, we present preliminaries about the SVDD and LMIs. In Section 3, we derive a modification of the SVDD toward the direction of utilizing ellipsoids and LMIs. Finally, in Section 4, concluding remarks are given.

2. Preliminaries

The SVDD method, which approximates the existence area of objects belonging to normal class, is derived as follows[4,5]: Consider a ball B with the center $a \in R^d$ and the radius R, and the training data set D consisting of objects $x_i \in R^d$, $i=1,\cdots,N$. We should note that since the training data are usually prone to noise, some part of the training data D could be abnormal objects. The main idea of the SVDD is to find a ball that can achieve the two conflicting goal simultaneously: First, it should be as small as possible, and more importantly it should contain as many training data as possible. Obviously, somewhat satisfactory balls satisfying these multiple objectives may be obtained by solving the following optimization problem:

$$\begin{aligned} \min \quad & L_o(R^2, a, \xi) = R^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & \|x_i - a\|^2 \le R^2 + \xi_i,\ \xi_i \ge 0,\ \forall i. \end{aligned} \tag{1}$$

Here, the slack variable ξ_i represents the penalty associated with the deviation of the i-th training pattern outside the ball. The objective function of the above optimization problem consists of the two conflicting terms, i.e., the square of radius R^2 and the total penalty $\sum_{i=1}^{N}\xi_i$. The constant C controls relative importance of each term; thus called the trade-off constant. The dual problem of the above can be derived as follows: First by introducing a Lagranger multiplier for each inequality condition, we obtain the following Lagrange function:

$$L = R^2 + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i[\,(x_i-a)^T(x_i-a) - R^2 - \xi_i] - \sum_{i=1}^{N}\eta_i\xi_i$$

where $\alpha_i \geq 0$, $\eta_i \geq 0$, $\forall i$.

From the saddle point condition[1], the optimal solution of (1) should satisfy the following:

$$\frac{\partial L}{\partial(R^2)} = 0: \ \sum_{i=1}^{N}\alpha_i = 1.$$

$$\frac{\partial L}{\partial a} = 0: \ a = (\sum_i \alpha_i x_i) / \sum_i \alpha_i = \sum_{i=1}^{N}\alpha_i x_i$$

$$\frac{\partial L}{\partial \xi_i} = 0: \ \alpha_i \in [0, C], \ \forall i. \tag{2}$$

With substitution of the above into L, the Lagrange function can be expressed in terms of the dual variables:

$$L = \sum_{i=1}^{N}\alpha_i\langle x_i, x_i\rangle - \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\langle x_i, x_j\rangle,$$

where $\sum_{i=1}^{N}\alpha_i = 1$, $\alpha_i \in [0, C]$, $\forall i$.

Thus, the dual problem can be written as follows:

$$\max_{\alpha} \sum_{i=1}^{N}\alpha_i\langle x_i, x_i\rangle - \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\langle x_i, x_j\rangle \quad s.t. \ \sum_{i=1}^{N}\alpha_i = 1, \ \alpha_i \in [0, C], \ \forall i. \tag{3}$$

Note that the above is equivalent to the following QP(quadratic programming) problem:

$$\min_{\alpha} \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\langle x_i, x_j\rangle - \sum_{i=1}^{N}\alpha_i\langle x_i, x_i\rangle \quad s.t. \ \sum_{i=1}^{N}\alpha_i = 1, \ \alpha_i \in [0, C], \ \forall i. \tag{4}$$

Also, note that from the Kuhn-Tucker complementarity condition, the following should hold true:

$$\alpha_i(\,\|x_i - a\|^2 - R^2 - \xi_i) = 0, \ \forall i \tag{5}$$

From the above, we can easily show that ultimately only the data points on the boundary or outside the ball can have the positive alpha values. These data points are called the support vectors. Once the α_i are obtained via solving the problem (4), the optimal center is given by the equation (2). Also, the optimal value of R^2 is acquired by applying the condition (5) to support vectors. After the training phase is over, we decide whether a given test point $x \in \mathbb{R}^d$ belongs to the normal class utilizing the following criterion:

$$f(x) = R^2 - \|x - a\|^2 = R^2 - \langle x, x\rangle - 2\sum_{i=1}^{N}\alpha_i\langle x_i, x\rangle + \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\langle x_i, x_j\rangle \geq 0$$

As mentioned before, balls can express only simple region. To enable to express more complex region, one can use balls defined on the feature space F. More precisely, consider the problem of finding relatively small ball $B_F \subset F$ that can contain large portion of the training data $D_F = \{\phi(x_i) \mid i = 1, \cdots, N\} \subset F$. With the arguments used for the SVDD, we can obtain the following mathematical formulation:

$$\min_{\alpha} \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j K(x_i, x_j) - \sum_{i=1}^{N}\alpha_i K(x_i, x_i) \quad s.t. \ \sum_{i=1}^{N}\alpha_i = 1, \ \alpha_i \in [0, C], \ \forall i. \tag{6}$$

When the gaussian function is chosen for the kernel, we always have $K(x, x) = 1$ for each $x \in \mathbb{R}^d$. Thus, the above problem can be further simplified as follows:

$$\min_{\alpha} \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j K(x_i, x_j) \quad s.t. \ \sum_{i=1}^{N}\alpha_i = 1, \ \alpha_i \in [0, C], \ \forall i. \tag{7}$$

Note that in this case, the criterion for the normality can be summarized as follows

$$\|\phi(x) - a\|^2 = 1 - 2\sum_{i=1}^{N}\alpha_i K(x_i, x) + \sum_{i=1}^{N}\sum_{j=1}^{N}K(x_i, x_j) \leq R^2 \tag{8}$$

Among the important tools of this paper are the linear matrix inequalities(LMIs), which mean the inequality constraints of the form:[11]

$$A(x) = A_0 + x_1A_1 + \cdots + x_NA_N < 0 \tag{9}$$

Here, $x = (x_1, \cdots, x_N)$ are the unknowns, $A_0, \cdots, A_N$ are given symmetric matrices, and "$<$" stands for the negative-definiteness. With SDP(semi-definite programming) problems, we mean the optimization problems having linear objective function and LMI constraints. SDP problems can be solved within the prescribed tolerance level by means of recently developed interior point methods. In the simulation part of this paper, we used the MATLAB LMI Control Toolbox[12] for solving SDP problems.

3. Main Results: A Learning Method Utilizing Ellipsoids and LMIs

As explained in Section 2, we can express shapes more complex than simple balls on the input domain by making use of the Mercer kernel. However, through many simulation works, we could observe that there certainly exist some limitations that cannot be overcome by simply utilizing balls on the feature space. For example, consider the result shown in Figure 1, which is yielded by the SVDD under the condition $\sigma = 4$, $C = 3.57$. (Note that this kind of example for an illustration of the problematic aspect of the SVDD was considered in [10].) This figure shows us the training data on

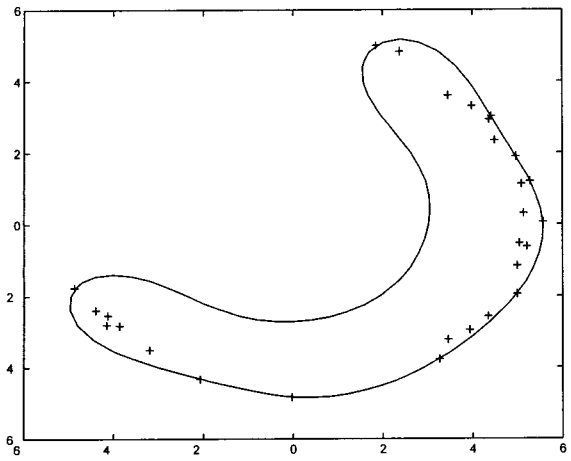

Figure 1. A classification result obtained by the SVDD method

R^2 together with the region for the normal class yielded by the Gaussian kernel-based SVDD. In the first look, the region for the normal class, which has a shape resembling banana, seems to describe the training data relatively well. However, with closer look, we can observe that there exists certain kind of defective unbalance between the areas near inner and outer surfaces of the banana shape. In other words, the figure illustrates the defect that could happen when applying the SVDD approach in the one-class problem; it could accept a significantly wide area where no training data reside as a region for the normal class.

In this paper, we try to mitigate this problematic aspect via a strategy utilizing ellipsoids and LMIs. For this, let us recall that the SVDD which uses balls on the feature space is formulated as follows:

$$\begin{aligned} \min \quad & L_o(R^2, a, \xi) = R^2 + C \sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & \|\phi(x_i) - a\|^2 \le R^2 + \xi_i,\ \xi_i \ge 0,\ \forall i. \end{aligned} \tag{10}$$

With the change of variables $\xi_i \triangleq R^2 z_i$,

$$\|\phi(x_i) - a\|^2 \le R^2 + \xi_i$$

can be rewritten as

$$\{\phi(x_i) - a\}^T (R^2 I)^{-1} \{\phi(x_i) - a\} \le 1 + z_i \tag{11}$$

Thus, problem (10) can be turned into the following equivalent form:

$$\begin{aligned} \min \quad & L_o(R^2, a, z) = R^2 + (CR^2) \sum_{i=1}^{N} z_i \\ \text{s.t.} \quad & \begin{bmatrix} R^2 I & \phi(x_i) - a \\ \star & 1 + z_i \end{bmatrix} \ge 0,\ z_i \ge 0,\ \forall i. \end{aligned} \tag{12}$$

(Here, the star represents the omitted part, which can be filled utilizing the symmetry of the matrix.)

Note that the first constraint of (12) is just a restatement of (11) utilizing the Schur complement, which is explained in [11]. For the modification toward the direction of utilizing ellipsoids, we make use of positive definite matrix P rather than $R^2 I$. Then with a slight modification of the objective function, we can obtain the following new formulation:

$$\begin{aligned} \min \quad & L_o(P, a, z) = \frac{1}{N} \operatorname{Tr}(P) + C \sum_{i=1}^{N} z_i \\ \text{s.t.} \quad & \begin{bmatrix} P & \phi(x_i) - a \\ \star & 1 + z_i \end{bmatrix} \ge 0,\ z_i \ge 0,\ \forall i,\ P > 0. \end{aligned} \tag{13}$$

(Here, Tr represents the trace operator.)

An immediately obvious disadvantage of the above formulation is that the feature vector $\phi(x_i)$ is in general very high dimensional; thus the number of unknowns getting huge. To relax this problem, we adopt the so-called empirical kernel feature map, which was recently proposed by Tsuda et al. in [13]. Since this concept plays an important role in our formulation, we describe more details about it. First, consider a function $k^*: \mathbb{R}^d \to \mathbb{R}^N$ defined as follows:

$$k^*(x) = \begin{bmatrix} k(x_1, x) \\ \vdots \\ k(x_N, x) \end{bmatrix} \in \mathbb{R}^N \tag{14}$$

This function seems to be a good choice for the feature map because it has the following attractive properties:

(a) Each entry of the feature vector $k^*(x)$ has clear meaning.

(b) The role of each training data is explicitly shown in $k^*(x)$.

(c) The dimension of the feature vector is not excessively high.

However, with direct application of the vector (14), the kernel trick cannot be satisfied as shown below:

$$\begin{aligned} & \langle k^*(x_i), k^*(x_j) \rangle \\ &= \left\langle \begin{bmatrix} k(x_1, x_i) \\ \vdots \\ k(x_N, x_i) \end{bmatrix}, \begin{bmatrix} k(x_1, x_j) \\ \vdots \\ k(x_N, x_j) \end{bmatrix} \right\rangle \\ &= \sum_{l=1}^{N} k(x_i, x_l) k(x_l, x_j). \end{aligned}$$

Hence, in order to get an appropriate feature map, we need to modify adequately. For this modification, we first define the kernel matrix[1] $K \in \mathbb{R}^{N \times N}$, whose (i, j)-th entry is $k(x_i, x_j)$. Of course, here the function k should be a Mercer kernel satisfying the Mercer condition. In many cases, the resulting kernel matrix becomes positive definite. Next, pre-multiply the vector $k^*(x)$ defined in (14) by matrix $K^{-1/2}$; then we get the following:

$$\phi(x) \triangleq K^{-1/2} k^*(x) = K^{-1/2} \begin{bmatrix} k(x_1, x) \\ \vdots \\ k(x_N, x) \end{bmatrix} \tag{15}$$

Note that with the above modification, the kernel trick is

1 It is often called as the Gram matrix.

now guaranteed as shown below:

$$\begin{aligned}\langle \phi(x_i), \phi(x_j) \rangle &= \{K^{-1/2}k^*(x_i)\}^T \{K^{-1/2}k^*(x_i)\} \\ &= k^*(x_i)^T K^{-1} k^*(x_j) \\ &= k^*(x_i)^T e_j \\ &= k(x_i, x_j)\end{aligned}$$

The above function $\phi: \mathbb{R}^d \rightarrow \mathbb{R}^N$ in (15) is often called the empirical feature map. By plugging the empirical feature map $\phi(x) = K^{-1/2}k^*(x)$ into (13), our derivation is now summarized as the following LMI-based optimization problem:

$$\begin{aligned}\min \quad & L_o(P, a, z) = \frac{1}{N}\,\mathrm{Tr}(P) + C\sum_{i=1}^{N} z_i \\ \text{s.t.} \quad & \begin{bmatrix} P & K^{-1/2}k^*(x_i) - a \\ \star & 1 + z_i \end{bmatrix} \geq 0, \\ & z_i \geq 0\ , \ \forall i\ , \\ & P > 0\ .\end{aligned} \tag{16}$$

When a test input $x \in \mathbb{R}^d$ is given after the training phase is over, an obvious choice for the acceptance criterion of x as a normal object would be

$$\begin{aligned}& 1 - (\phi(x) - a)^T P^{-1} (\phi(x) - a) \\ = \ & 1 - \|\phi(x) - a\|_P^2 \\ \geq \ & 0\ .\end{aligned}$$

However, some more flexibility would be possible with the following one:

$$f(x) = 1 + \rho - \|K^{-1/2}k^*(x) - a\|_P^2 \geq 0 \tag{17}$$

(Here, $\rho > 0$ is a constant chosen by users)

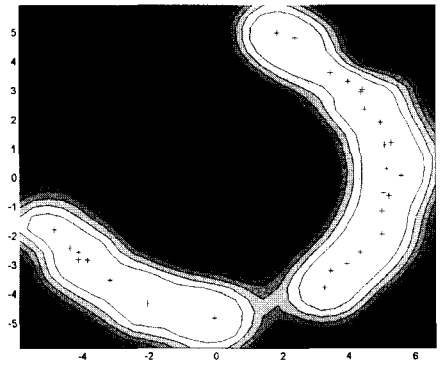

Figure 2. Contours obtained by the method of this paper

For the training data shown in Figure 1, we applied the method of this paper, and obtained the contour lines of the decision function f (under the conditions $\sigma = 1.5$, $C = 0.1$) shown in Figure 2. The contours show that with some more refining process, more meaningful results could follow.

4. Concluding Remarks

In this paper, we considered the problem of modifying the SVDD method into the direction that can make use of ellipsoids rather than simple balls. Along with a brief review over the conventional SVDD method, we presented a new method utilizing ellipsoids defined on the feature space and LMI-based solutions. The work performed in this paper is a kind of feasibility study toward a new direction; thus lacks practicality yet, and has a lot of things to be refined.

References

[1] N. Cristianini an J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods,* Cambridge University Press, 2000.

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

[3] C. Bishop, "Novelty detection and neural networks validation," *IEE Proceedings on Vision, Image, and Signal Processing, Special Issue on Applications of Neural Networks*, vol. 141, pp. 217-222, 1994.

[4] D. Tax and R. Duin, "Support Vector Domain Description," *Pattern Recognition Letters*, vol. 20, pp. 1191-1199, 1999.

[5] D. Tax, *One-class classification*, PhD Thesis, Delft University of Technology, 2001.

[6] B. Schölkopf, J. C. Platt, and A. J. Smola, *Kernel method for percentile feature extraction*, Technical Report MSR-TR-2000-22, Microsoft Research, WA, 2000.

[7] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, and A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol.13, pp. 1443-1471, 2001.

[8] G. Ratch, S. Mika, B. Schölkopf, and K.-R. Müller, "Constructing boosting algorithms from SVMs: An application to one-class classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1-15, 2002.

[9] C. Campbell and K. P. Bennett, "A linear programming approach to novelty detection," *Advances of NIPS 2000*, pp. 395-401, 2000.

[10] D. Tax and P. Juszczak, "Kernel whitening for one-class classification," *Pattern Recognition with Support Vector Machines*, pp. 40-52, 2002.

[11] S. Boyd, L. ElGhaoui, E. Feron and V. Balakrishnan, *Linear matrix inequalities in systems and control theory, SIAM Studies in Applied Mathematics, Vol. 15, SIAM, Philadelphia*, 1994.

[12] P. Gahinet, A. Nemirovski, A. J. Laub and M. Chilali, *LMI control toolbox*, MathWorks Inc., Natick, MA, 1995.

[13] K. Tsuda, "Support vector classifiers with asymmetric kernel functions," *Proceedings of ESANN*, pp. 183-188, 1999.

Jooyoung Park
received his BS degree in electrical engineering from Seoul National University, Korea, in 1983, and his PhD degree in electrical and computer engineering from the University of Texas at Austin, USA, in 1992. He joined Korea University in 1993, where he is currently a Professor in the Department of Control and Instrumentation Engineering. His research interests include neural networks and nonlinear systems.

Jinsung Kim
received his BS degree in control and instrumentation engineering and MS degree in applied electronics engineering from Korea University, Korea, in 1996 and 1998, respectively. He is now pursuing his PhD degree in the Department of Electrical Engineering, Korea University, Korea. His research interests include fuzzy modelling, intelligent control, and support vector machines.

Hansung Lee
received his BS and MS degrees in computer science from Korea University, Korea, in 1996 and 2002, respectively. He is now pursuing his PhD degree in the Department of Computer Science, Korea University, Korea. His research interests include machine learning and data mining.

Daihee Park
received his BS degree in mathematics from Korea University, Korea, in 1982, and his PhD degree in computer science from the Florida State University, USA, in 1992. He joined Korea University in 1993, where he is currently a Professor in the Department of Computer Science. His research interests include artificial intelligence and intelligence database.