Support Vector Data Description by Using Hyper-ellipse Instead of Hyper-sphere

Yahya Forghani
Computer Department,
Mashhad Branch, Azad University,
Mashhad, Iran
yahyafor2000@yahoo.com

Hadi Sadoghi Yazdi
Computer department
Ferdowsi University of Mashhad
Mashhad, Iran
h-sadoghi@um.ac.ir

Sohrab Effati
Applied Mathematics department
Ferdowsi University of Mashhad
Mashhad, Iran
s-effati@um.ac.ir

Reza Sigari Tabrizi
Faculty of Management
Multimedia University
Malaysia
reza.tabrizi@live.com

***Abstract*—Support Vector Data Description (SVDD) describes data by using a hyper-sphere. In this paper, we propose an extended SVDD (ESVDD) which describes data by using a hyper-ellipse. Clearly, ESVDD can describe data better than SVDD in the input space. Both hyper-sphere and hyper-ellipse are very rigid for data description. The kernel ESVDD which will be proposed in this paper and the kernel SVDD enhance the ability of ESVDD and SVDD for data description, respectively. The formulation of SVDD/ESVDD contains a penalty term C which controls the tradeoff between the volume of hyper-sphere/hyper-ellipse and the training errors. We show that the ESVDD can control this tradeoff better than the SVDD.**

Keywords- Kernel; ESVDD; Data description; Hyper-ellipse.

I. INTRODUCTION

The one-class classification problem is an interesting field in pattern recognition and machine learning researches. In this kind of classification, we assume the one class of data as the target class and the rest of data are classified as the outlier. One-class classification is particularly significant in applications where only a single class of data objects is applicable and easy to obtain. Objects from the other classes could be too difficult or expensive to be made available. So we would only describe the target class to separate it from the outlier class.

The SVDD is a kind of one-class classification method based on Support Vector Machine [1, 2] which proposed by Tax. 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, 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 the 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) [3]. Ji et al. discussed the SVDD application in gene expression data clustering [4]. Yu et al. used SVDD for image categorization from internet images [5].

Recently, some efforts have been expended to improve the SVDD method. Guo et al. proposed a simple post-processing method which tries to modify the SVDD boundary in order to achieve a tight data description [6]. As another example Cho apply the orthogonal filtering as a preprocessing step is executed before SVDD modeling to remove the unwanted variation of data [7].

In this paper, we propose an extended SVDD (ESVDD) which describes data by using a hyper-ellipse instead of a hyper-sphere. Clearly, ESVDD can describe data better than SVDD. The kernel ESVDD which will be proposed in this paper enhance the ability of ESVDD for data description. The formulation of SVDD/ESVDD contains a penalty term C which controls the tradeoff between the volume of hyper-sphere/hyper-ellipse and the training errors. We show that the ESVDD can control this tradeoff better than the SVDD.

Our proposed method has two major differences with the ellipse SVDD proposed by GhasemiGol et al. [8]. Firstly, they used a hyper-ellipse whose diameters are parallel with the axes. Secondly, they have not used the kernel trick for non-linear data description which makes their method more time-consuming. In our proposed method, both drawbacks of their method will be removed.

The paper is organized as follows. In the next section we review the support vector data description (SVDD). Our proposed method is explained in Section 3. Finally, in section 4 the experimental results are presented and we conclude in the last section.

II. SVDD

Let $x_i (i = 1,2,...,n)$ be p-dimensional training samples belonging to one class. We consider approximating the class region by the minimum hyper-sphere with center $e = (e_1, e_2, ...)^T$ and radius R in a high dimensional feature space (HDS), excluding the outliers. Therefore, the problem is

$$\min_{R,e,\xi} R^2 + C\sum_{i=1}^{n}\xi_i$$
$$s.t.\begin{cases} \|\varphi(x_i) - e\|^2 \le R^2 + \xi_i, & i = 1,2,...,n, \\ \xi_i, & i = 1,2,...,n, \end{cases} \tag{1}$$

where $\varphi(x)$ is the mapping function that maps x into a high dimension feature space (HDS), $\xi = (\xi_1, \xi_2, ..., \xi_n)^T$, ξ_i is the slack variable of i-th training sample and C is a constant which determines the trade-off between the hyper-sphere volume and training errors.

The Lagrangian dual form of program (1) can be restated as follows:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i K(x_i, x_i) - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j K(x_i, x_j)$$
$$s.t.\begin{cases} \sum_{i=1}^{n} \alpha_i = 1, \\ 0 \le \alpha_i \le C, \quad i = 1,2,...,n, \end{cases}$$

where $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$. From the optimality condition of the program (2.1), we obtain

$$R^2 = K(x_i, x_i) - 2\sum_{j=1}^{n} \alpha_j K(x_i, x_j) + \sum_{k=1}^{n}\sum_{j=1}^{n} \alpha_k \alpha_j K(x_k, x_j),$$

and the unknown datum x is inside the hyper-sphere if $\|\varphi(x_i) - e\|^2 \le R^2$ or equivalently if

$$K(x,x) - 2\sum_{j \in SV}^{n} \alpha_j K(x, x_j) + \sum_{k \in SV}^{n}\sum_{j \in SV}^{n} \alpha_k \alpha_j K(x_k, x_j) \le R^2,$$

where SV is the set of indices of training samples whose $\alpha \ne 0$ (See [1, 2] for more information).

III. OUR PROPOSED METHOD

The SVDD describes data by using a hyper-sphere. It's clear that the description of data in the input space by using a hyper-ellipse is better than a hyper-sphere (see Figure 1). Each point on the hyper-ellipse satisfies the following equation:

$$\|x - C_1\| + \|x - C_2\| = d,$$

where d is a constant value, and C_1 and C_2 are its focuses (see Figure 2). So, the formulation of our novel method called extended SVDD (ESVDD) is as follows:

$$\min_{d,C_1,C_2,\xi} d + C\sum_{i=1}^{n}\xi_i$$
$$s.t.\begin{cases} \|x_i - C_1\| + \|x_i - C_2\| \le d + \xi_i, & i = 1,2,...,n, \\ \xi_i \ge 0, & i = 1,2,...,n. \end{cases} \tag{2}$$

The above program is convex (See appendix A). Therefore, its global optimal solution can be obtained easily [9].

Clearly, ESVDD can describe data better than SVDD in the input space. If samples of a dataset have been distributed normally (with a Gaussian distribution), then ESVDD in the input space can describe it better than SVDD in the input space which uses a hyper-sphere and needn't use the kernel version of SVDD for data description. Using the kernel version of SVDD can cause to obtain many support vectors and to increase testing time.

The formulation of kernel ESVDD is as follows:

$$\min_{d,C_1,C_2,\xi} d + C\sum_{i=1}^{n}\xi_i$$
$$s.t.\begin{cases} \|\varphi(x_i) - C_1\| + \|\varphi(x_i) - C_2\| \le d + \xi_i, & i = 1,2,...,n, \\ \xi_i \ge 0, & i = 1,2,...,n. \end{cases} \tag{3}$$

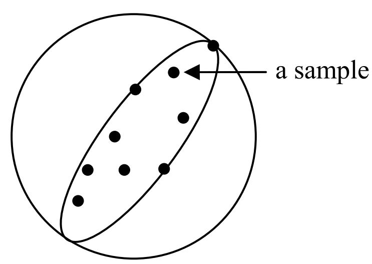


Figure 1. Description of data by using circle and ellipse.

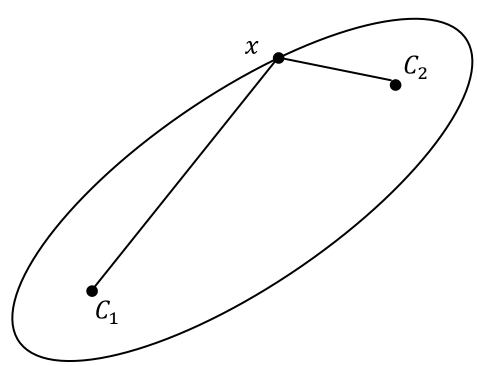


Figure 2. An ellipse and its focuses.

We will prove in appendix B that

$$C_1 = \sum_{j=1}^{n} \alpha_j \varphi(x_j), \qquad C_2 = \sum_{j=1}^{n} \beta_j \varphi(x_j),$$

where $\forall j : \alpha_j, \beta_j$ are real scalars. Therefore, the program (3) can be restated as follows:

$$\min_{d, C_1, C_2, \xi} d + C\sum_{i=1}^{n} \xi_i$$

$$s.t. \begin{cases} \left\| \varphi(x_i) - \sum_{j=1}^{n} \alpha_j \varphi(x_j) \right\| + \left\| \varphi(x_i) - \sum_{j=1}^{n} \beta_j \varphi(x_j) \right\| \\ \qquad \le d + \xi_i, \quad i = 1,2,\ldots,n, \\ \xi_i \ge 0, \quad i = 1,2,\ldots,n, \end{cases}$$

or equivalently

$$\min_{d, C_1, C_2, \xi} d + C\sum_{i=1}^{n} \xi_i$$

$$s.t. \begin{cases} \sqrt{K(x_i, x_i) - 2\sum_{j=1}^{n} \alpha_j K(x_i, x_j) + \sum_{j=1}^{n}\sum_{k=1}^{n} \alpha_j \alpha_k K(x_j, x_k)} + \\ \sqrt{K(x_i, x_i) - 2\sum_{j=1}^{n} \beta_j K(x_i, x_j) + \sum_{j=1}^{n}\sum_{k=1}^{n} \beta_j \beta_k K(x_j, x_k)} \le \\ \qquad d + \xi_i, \quad i = 1,2,\ldots,n, \\ \xi_i \ge 0, \quad i = 1,2,\ldots,n. \end{cases}$$

The unknown datum x is inside the hyper-ellipse if

$$\|\varphi(x) - C_1\| + \|\varphi(x) - C_2\| \le d,$$

or equivalently if

$$\sqrt{K(x,x) - 2\sum_{j \in SV1} \alpha_j K(x, x_j) + \sum_{j \in SV1,} \sum_{k \in SV1} \alpha_j \alpha_k K(x_j, x_k)} + \sqrt{K(x,x) - 2\sum_{j \in SV2} \beta_j K(x, x_j) + \sum_{j \in SV2,} \sum_{k \in SV2} \beta_j \beta_k K(x_j, x_k)} \le d$$

where $SV1$ is the set of indices of training samples whose $\alpha \ne 0$ and $SV2$ is the set of the indices of training samples whose $\beta \ne 0$.

IV. EXPRIMENTAL RESULTS

In this section, the superiority of our proposed algorithm with respect to the SVDD is studied by using some numerical examples. Here, for ease of evaluation, 2-dimensional data is used. Moreover, the Gaussian kernel function, namely $K(x, y) = \exp\left(\frac{\|x - y\|^2}{2\sigma^2} \right)$, is utilized in the kernel SVDD and kernel ESVDD.

A. Example 1.

Figure 3-8 plot the description of data obtained by using the kernel SVDD and our novel method (kernel ESVDD) for the first training set and for $\sigma = 20$ and different value of penalty term C. As it can be seen, one of the training samples plotted in each of these figures is far from the other samples and may be a noisy sample. By decreasing the value of penalty term, the kernel ESVDD could ignore such noisy sample, properly (see Figure 5 and 6). But, the boundary of kernel SVDD either has ignored some other training samples too or has contained the area between the noisy training sample and the other samples when the value of penalty term is decreased. The SVDD cannot ignore such noisy sample by changing the value of penalty term *C*.

This problem occurs when the value of σ is not small enough. Choosing a small value for σ can remove the mentioned problem of kernel SVDD but may produce a very tight decision boundary which may cause to occur the over-fit problem (see Figure 7).

B. Example 2.

Figure 8-11 plots the problem of kernel SVDD for the second training set. One of the training samples plotted in each of these figures is far from the other samples and may be a noisy sample. Again, by decreasing the value of penalty term, the kernel ESVDD could ignore such noisy sample, properly (see Figure 9). But, the boundary of kernel SVDD either has ignored some other training samples too or has contained the area between the noisy training sample and the other samples when the value of penalty term is decreased. The SVDD cannot ignore such noisy sample by changing the value of penalty term *C*.

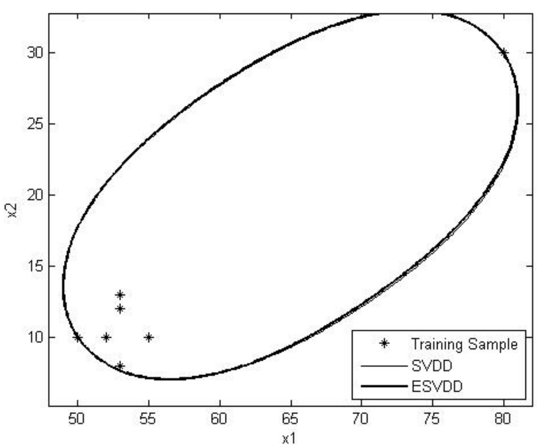


Figure 3. Description of data for $\sigma = 20, C = 0.6$

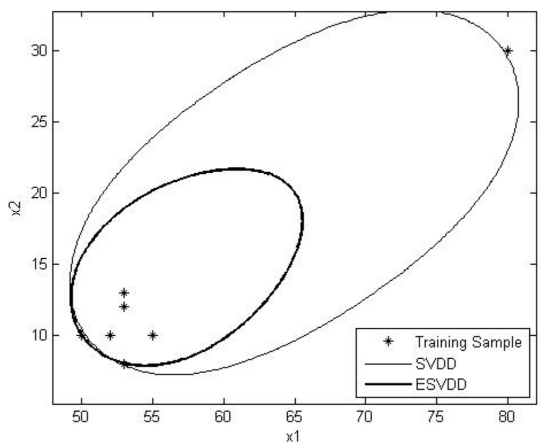


Figure 4. Description of data for $\sigma = 20, C = 0.5$

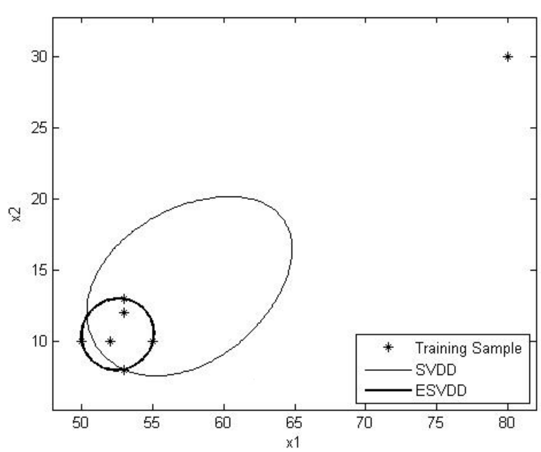


Figure 5. Description of data for $\sigma = 20, C = 0.4$

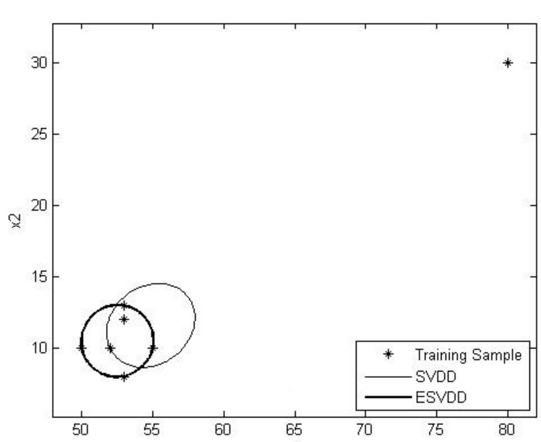


Figure 6. Description of data for $\sigma = 20, C = 0.3$

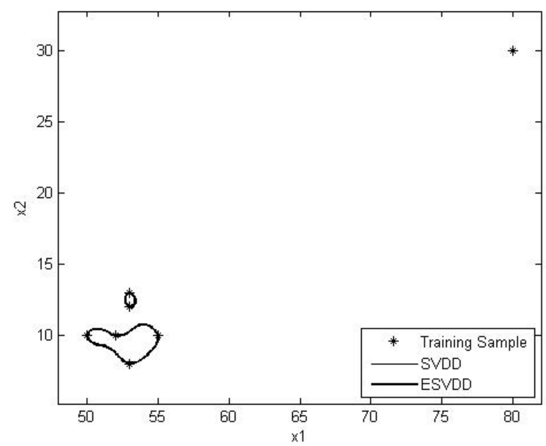


Figure 7. Description of data for $\sigma = 1.5, C = 0.2$

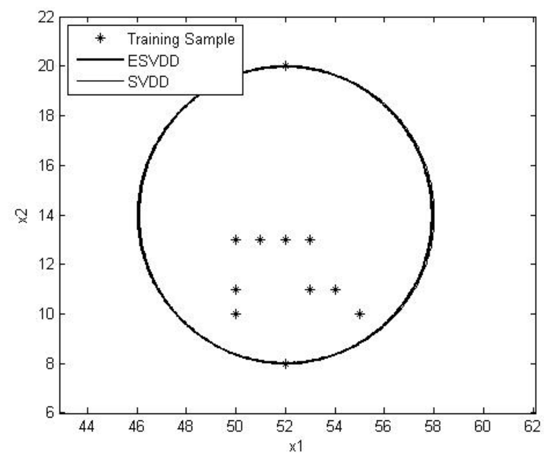


Figure 8. Description of data for $\sigma = 50, C = 0.55$

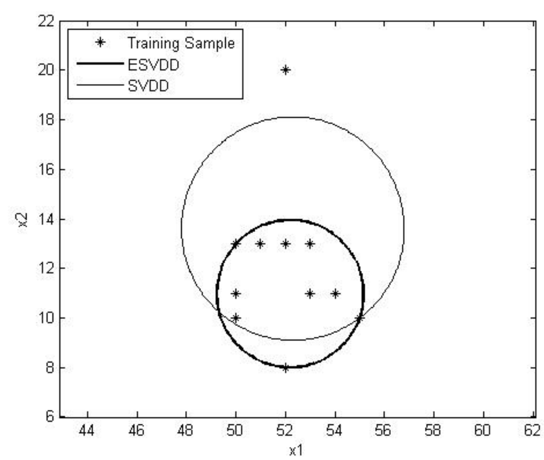


Figure 9. Description of data for $\sigma = 50, C = 0.45$

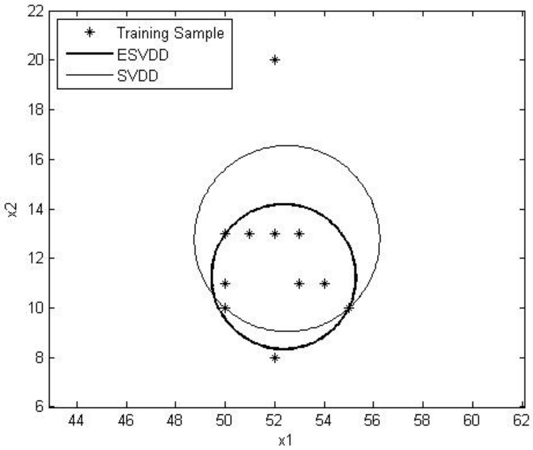


Figure 10. Description of data for $\sigma = 50, C = 0.35$

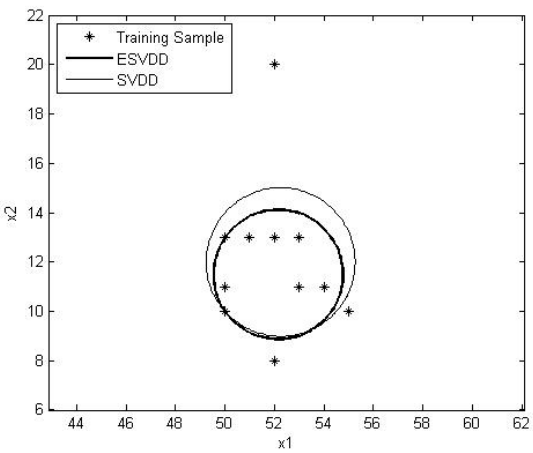


Figure 11. Description of data for $\sigma = 50, C = 0.25$

V. Conclusion

In this paper, we proposed an extended SVDD (ESVDD) which describes data by using a hyper-ellipse. We extended it in high dimensional feature space, too.

Clearly, ESVDD can describe data better than SVDD in the input space. If samples of a dataset have been distributed normally (with a Gaussian distribution), then ESVDD in the input space can describe it better than SVDD in the input space which uses a hyper-sphere, and needn't use the kernel SVDD for data description. Using the kernel SVDD or kernel ESVDD can cause to obtain many support vectors and to increase testing time.

The formulation of kernel SVDD/ESVDD contains a penalty term C which controls the tradeoff between the volume of hyper-sphere/hyper-ellipse and the training errors. We showed that the kernel ESVDD could control this tradeoff better than the kernel SVDD.

Appendix A

Theorem 1. The program (2) is convex.

Proof. The cost function this program and the second set of their constraints are linear and therefore are convex. Thus, to prove the convexity of this program it suffices to prove the convexity of the first set of its constraints [9]. Let $\|.\|$ be a vector norm. For each $x, y \in \Re^p$ and for each $\lambda \in [0,1]$, we have

$$\|\lambda x + (1-\lambda)y\| \le \|\lambda x\| + \|(1-\lambda)y\| = \lambda\|x\| + (1-\lambda)\|y\| .$$

Therefore, $\|.\|$ is convex. Thus, we conclude that the second set of the constraints of the program (2) are convex.

Appendix B

Theorem 2. Consider the program (3.2). Each focuses of the objective hyper-ellipse can be stated as follows

$$C_1 = \sum_{j=1}^{n} \alpha_j x_j , \qquad C_2 = \sum_{j=1}^{n} \beta_j x_j ,$$

where $\forall j : \alpha_j , \beta_j$ are real scalars, and $x_j \, (j = 1,2,...,n)$ are training samples in the input space or high dimensional feature space.

Proof. If we have at least p linearly independent p-dimensional training samples, each point of the p-dimensional space can be expressed by using these training samples and the proof is complete. For example, each point x of a 2-dimensional space can be expressed by using two linearly independent points x_1 and x_2, namely by utilizing one of the following relations:

$x = A_1 x_1 + A_2 x_2$, $x = -A_1 x_1 + A_2 x_2$, $x = A_1 x_1 - A_2 x_2$ or $x = -A_1 x_1 - A_2 x_2$, where $A_1, A_2 \in \Re^+ \cup \{0\}$ or equivalently x can be expressed using the relation $x = \sum_{j=1}^{2} \alpha_j x_j$, where $\alpha_j \in \Re$ (See Figure 12).

Now, suppose that we have $n < p$ linearly independent training samples and some dependent training samples in the p-dimensional space. Then the objective hyper-ellipse is on a p-dimensional hyper-plane (Figure 13 and 14 show this fact for $p=2$ and $p=3$, respectively). Therefore, since each point of the n-dimensional space can be expressed using n linearly independent training samples, the proof is complete.

References

[1] D. M. J. Tax and R. P. W. Duin, "Support vector domain description," *Pattern Recognition Letters,* vol. 20, pp. 1191-1199, 1999.

[2] D. M. J. Tax and R. P. W. Duin, "Support Vector Data Description," *Mach. Learn.,* vol. 54, pp. 45-66, 2004.

[3] 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.

[4] 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*, 2008, pp. 371-374.

[5] 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.

[6] S. M. Guo, L. C. Chen, and J. S. H. Tsai, "A boundary method for outlier detection based on support vector domain description," *Pattern Recognition,* vol. 42, pp. 77-83, 2009.

[7] H. W. Cho, "Data description and noise filtering based detection with its application and performance comparison," *Expert Systems with Applications,* vol. 36, pp. 434-441, 2009.

[8] M. GhasemiGol, R. Monsefi, and H. Sadoghi Yazdi, "Ellipse Support Vector Data Description," *Communications in Computer and Information Science,* vol. 43, pp. 257-268, 2009.

[9] M. S. Bazara, H. D. Sherali, and C. M. Shetty, *Nonlinear programming*, 3 ed.: John Willey & Sons, Inc., 2006.

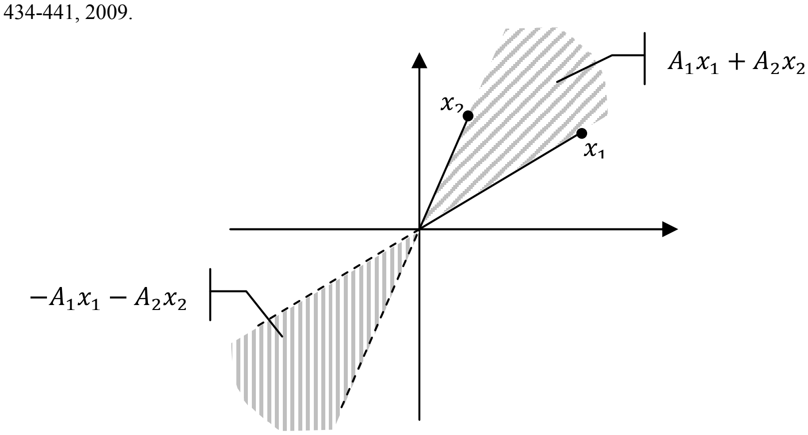
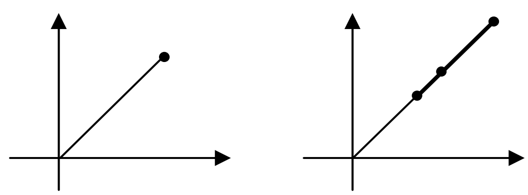
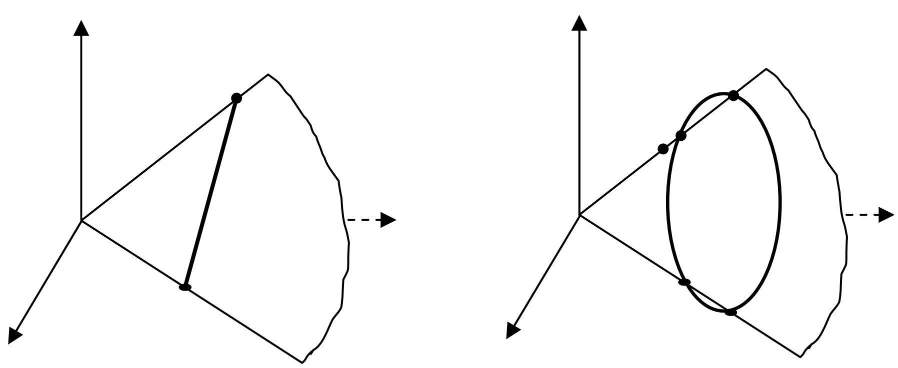


Figure 12. Dividing the 2-dimensional space by using two linearly independent points x_1 and x_2.



(a) (b

Figure 13. (a) The objective ellipse (shown by a point) in the 2-dimensional space when we have only n=1 independent training sample;
(b) The objective ellipse (shown by a solid segmented line) in the 2-dimensional space when we have n=1 independent training sample and some dependent training samples.



(a) (b)

Figure 14. a) The objective ellipse shown by solid segmented line in 3-dimensional space when we have only n=2 independent training sample;
b) The objective ellipse in 3-dimensional space when we have n=2 independent training sample and some dependent training sample