CHAPTER 2

Relevant Background

This chapter provides relevant backgrounds which form the bases for the proposed works in Chapter 3. First, we describe some notations in Section 2.1 which will also be used later throughout this thesis. In order to use hyperellipsoids in the hypothesis set of learning machines, we explain the representation of an ellipsoid and the formulation of the minimum volume covering ellipsoid (MVCE) in Section 2.2 and 2.3. Section 2.4 is particularly devoted to support vector based classifiers which use hyperplanes in their hypothesis set, while Section 2.5 focuses on the ones which use hyperspheres.

2.1 Notation

In this thesis, all column vectors are denoted using boldfaced lowercase letters, while matrices are in boldfaced capital letters. Given m training examples, $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_m \in \mathbb{R}^n$, we also define the corresponding matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ as $\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_m]$. $\mathbf{e}$ is a column vector of ones whose dimension is determined from the context. We always denote α_i as the i-th Lagrange multiplier. $\boldsymbol{\alpha} \in \mathbb{R}^m$ is the column vector whose element consists of α_i for i = 1, 2, …, m. We also denote $\mathbf{A} \in \mathbb{R}^{m \times m}$ to represent the diagonal matrix whose main diagonal is composed of the elements of $\boldsymbol{\alpha}$, i.e.

$$\mathbf{A} = \begin{bmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_m \end{bmatrix}.$$

In some cases, we also separate training data into classes. For p classes, we use $\mathbb{I}_1$, $\mathbb{I}_2$, …, $\mathbb{I}_p$ to represent class 1 to class p, respectively. The label of each class is y_i for i = 1, 2, …, p. The data in each class are also written in the matrix form $\mathbf{X}_i$ where each column of $\mathbf{X}_i$ represents one example in class i.

2.2 Representations of an Ellipsoid

A general representation of an ellipsoid in *n*-dimensional space whose center is at point **d** and its orientation is defined by a positive definite matrix **F** can be defined as follows.

Definition 2.1: *Given* $\mathbf{F} \in \mathbb{S}_{++}^{n}$, *and* $\mathbf{d} \in \mathbb{R}^{n}$, *an arbitrarily oriented ellipsoid* $E_{\mathbf{F},\mathbf{d}}$ *in* $\mathbb{R}^{n}$ *is represented as*

$$E_{\mathbf{F},\mathbf{d}} = \{\mathbf{x} \in \mathbb{R}^{n} : (\mathbf{x}-\mathbf{d})^{T}\mathbf{F}(\mathbf{x}-\mathbf{d}) \le 1\}. \tag{2.1}$$

Alternatively, suppose $\mathbf{F} = \mathbf{E}^{T}\mathbf{E}$ *and* $\mathbf{d} = \mathbf{E}^{-1}\mathbf{c}$, *ones can rewrite* $E_{\mathbf{F},\mathbf{d}}$ *with another representation,*

$$\mathbb{E}_{\mathbf{E},\mathbf{c}} = \{\mathbf{x} \in \mathbb{R}^{n} : \|\mathbf{E}\mathbf{x}-\mathbf{c}\| \le 1\}. \tag{2.2}$$

According to [71], the volume of an ellipsoid can be represented in terms of the volume of the unit ball in $\mathbb{R}^{n}$. Suppose the volume of the unit ball is V_n and let Γ be the gamma function, the volume of $E_{\mathbf{F},\mathbf{d}}$ is equal to $V_n(\det \mathbf{F})^{-1/2}$ or, in term of **E**, $V_n(\det \mathbf{E})^{-1}$. For the exact formulation of the unit ball's volume, the reader is kindly referred to [71]. However, for this thesis, it suffices to consider the volume of the unit ball as a scaling factor, since we are not interested in computing the actual volume of the ellipsoid.

2.3 Minimum Volume Covering Ellipsoid

Given a set of *m* examples $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_m$ where $\mathbf{x}_i \in \mathbb{R}^{n}$. A fundamental question is to find the minimal ellipsoid covering all the given points as illustrated in Figure 2.1.

Although there are many possible criteria in defining an ellipsoid to be minimal, such as using the trace or determinant of the ellipsoid's matrix **E**, this thesis specifically focuses on using the determinant since it represents the volume of the ellipsoid, and using

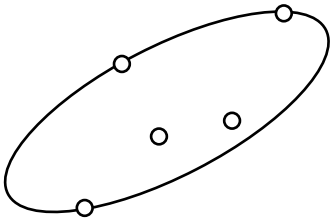


Figure 2.1 Minimal ellipsoid covering a set of examples

volumes is rather intuitive to control the size of an ellipsoid.

Since, for a given set of examples, it is possible that examples are not adequately distributed in all dimensions of $\mathbb{R}^n$. We consider the following assumption prior to the construction of the minimum volume covering ellipsoid in order to avoid the so-called *degenerate case* where a particular dimension has no example causing the covering ellipsoid to have zero volume or become deflated in that dimension.

Assumption 2.1: *The affine hull of the m given examples* $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m$ span $\mathbb{R}^n$.

Definition 2.2: *Given* $S = \{\mathbf{x}_i\}_{i=1}^m$, *the minimum volume covering ellipsoid* $\mathsf{E}^*_{\mathbf{E},\mathbf{c}}$ *is the smallest ellipsoid with the minimum volume to cover S. It is the solution to the following optimization problem,*

$$\begin{aligned} \min_{\mathbf{E},\mathbf{c}} \quad & 2\log\det(\mathbf{E}^{-1}) \\ \text{s.t.} \quad & \|\mathbf{E}\mathbf{x}_i - \mathbf{c}\| \le 1, \quad i = 1, \ldots, m \\ & \mathbf{E} \succ 0. \end{aligned} \tag{2.3}$$

The dual formulation of (2.3) is

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \log\det(\mathbf{X}\mathbf{A}\mathbf{X}^T - \frac{\mathbf{X}\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{X}^T}{\mathbf{e}^T\boldsymbol{\alpha}}) - \mathbf{e}^T\boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{\alpha} \ge 0 \end{aligned} \tag{2.4}$$

which can be derived using the standard Lagrange multiplier technique. That is from (2.3), the Lagrangian is

$$L(\mathbf{E}, \mathbf{c}, \boldsymbol{\alpha}) = -2\log\det(\mathbf{E}) + \sum_{i=1}^{m} \alpha_i \left(\|\mathbf{E}\mathbf{x}_i - \mathbf{c}\|^2 - 1 \right) \tag{2.5}$$

where the Lagrange multiplier $\alpha_i \ge 0$ for $i = 1, 2, \ldots, m$.

Since

$$\frac{\partial \|\mathbf{E}\mathbf{x} - \mathbf{c}\|^2}{\partial \mathbf{c}} = -2(\mathbf{E}\mathbf{x} - \mathbf{c}) \tag{2.6}$$

$$\frac{\partial \|\mathbf{E}\mathbf{x} - \mathbf{c}\|^2}{\partial \mathbf{E}} = (\mathbf{E}\mathbf{x} - \mathbf{c})\mathbf{x}^T + \mathbf{x}(\mathbf{E}\mathbf{x} - \mathbf{c})^T \tag{2.7}$$

$$\frac{\partial \log\det(\mathbf{E})}{\partial \mathbf{E}} = \mathbf{E}^{-1}, \tag{2.8}$$

the first-order derivatives are

$$\frac{\partial L}{\partial \mathbf{c}} = 2\left(\mathbf{c}\sum_{i=1}^{m}\alpha_i - \mathbf{E}\sum_{i=1}^{m}\alpha_i \mathbf{x}_i \right) \tag{2.9}$$

$$\frac{\partial L}{\partial \mathbf{E}} = -2\mathbf{E}^{-1} + \sum_{i=1}^{m}\alpha_i \left[(\mathbf{E}\mathbf{x}_i - \mathbf{c})\mathbf{x}_i^T + \mathbf{x}_i(\mathbf{E}\mathbf{x}_i - \mathbf{c})^T \right]. \tag{2.10}$$

In this thesis, we prefer using matrix representation when applicable for conciseness. Therefore, it is worth noting some useful relations for later uses.

$$\sum_{i=1}^{m}\alpha_i = \mathbf{e}^T\boldsymbol{\alpha}\ , \qquad \sum_{i=1}^{m}\alpha_i\mathbf{x}_i = \mathbf{X}\boldsymbol{\alpha}\ , \qquad \sum_{i=1}^{m}\alpha_i\mathbf{x}_i\mathbf{x}_i^T = \mathbf{X}\mathbf{A}\mathbf{X}^T \tag{2.11}$$

From (2.9), the optimal $\mathbf{c}$ of the MVCE is

$$\mathbf{c} = \frac{\mathbf{E}\sum_{i=1}^{m}\alpha_i\mathbf{x}_i}{\sum_{i=1}^{m}\alpha_i} = \frac{\mathbf{E}\mathbf{X}\boldsymbol{\alpha}}{\mathbf{e}^T\boldsymbol{\alpha}}. \tag{2.12}$$

With (2.10) and (2.12), under the first-order necessary condition of optimality, (2.10) can be rewritten as

$$\mathbf{E}^{-1} = \frac{1}{2}(\mathbf{E}\mathbf{S} + \mathbf{S}\mathbf{E}) \tag{2.13}$$

where

$$\mathbf{S} = \mathbf{X}\mathbf{A}\mathbf{X}^T - \frac{\mathbf{X}\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{X}^T}{\mathbf{e}^T\boldsymbol{\alpha}}. \tag{2.14}$$

According to Zhang & Gao [72], for $\mathbf{S} \in \mathbb{S}_{++}^n$, the unique solution to (2.13) is

$$\mathbf{E} = \mathbf{S}^{-\frac{1}{2}}. \tag{2.15}$$

At optimality, it can be shown that $\sum_{i=1}^{m}\alpha_i \left\|\mathbf{E}\mathbf{x}_i - \mathbf{c}\right\|^2 = n$ (See Lemma A.1 in Appendix A), Thus, the Lagrangian (2.5) can be reduced to

$$L(\boldsymbol{\alpha}) = \log\det(\mathbf{X}\mathbf{A}\mathbf{X}^T - \frac{\mathbf{X}\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{X}^T}{\mathbf{e}^T\boldsymbol{\alpha}}) - \mathbf{e}^T\boldsymbol{\alpha} + n. \tag{2.16}$$

As a result, we reach the dual formulation (2.4).

The dual formulation of MVCE can be more succinct when we also try to minimize the Mahalanobis distance in the primal problem (2.3). That is, suppose we rewrite (2.3) to

$$\begin{aligned} \min_{\mathbf{E},\mathbf{c},r} \quad & nr^2 + 2\log\det(\mathbf{E}^{-1}) \\ \text{s.t.} \quad & \left\|\mathbf{E}\mathbf{x}_i - \mathbf{c}\right\| \le r, \qquad i = 1,\ldots,m \\ & \mathbf{E} \succ 0. \end{aligned} \tag{2.17}$$

Then, instead of the Lagrangian in (2.5) , the Lagrangian of (2.17) is

$$L(\mathbf{E},\mathbf{c},r,\boldsymbol{\alpha}) = nr^2 - 2\log\det(\mathbf{E}) + \sum_{i=1}^{m}\alpha_i\left(\left\|\mathbf{E}\mathbf{x}_i - \mathbf{c}\right\|^2 - r^2\right). \tag{2.18}$$

The first order derivative of the Lagrangian (2.18) with respect to r is

$$\frac{\partial L}{\partial r} = 2r\left(n - \sum_{i=1}^{m}\alpha_i\right) = 2r(n - \mathbf{e}^T\boldsymbol{\alpha}). \tag{2.19}$$

Hence, under the first order optimality condition, either $r = 0$ or $\mathbf{e}^T\boldsymbol{\alpha} = n$. However, r cannot be zero due to the positive definiteness of $\mathbf{E}$ imposed by the log-determinant term in the objective function. Therefore, from (2.19), we have the constraint $\mathbf{e}^T\boldsymbol{\alpha} = n$. Since $\frac{\partial L}{\partial \mathbf{c}}$ and $\frac{\partial L}{\partial \mathbf{E}}$ of (2.18) are also equal to (2.9) and (2.10), respectively, the Lagrangian (2.18) is reduced to

$$L(\boldsymbol{\alpha}) = \log\det(\mathbf{X}\mathbf{A}\mathbf{X}^T - \frac{\mathbf{X}\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{X}^T}{\mathbf{e}^T\boldsymbol{\alpha}}). \tag{2.20}$$

Thus, the corresponding dual formulation of (2.17) becomes

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \log\det(\mathbf{X}\mathbf{A}\mathbf{X}^T - \frac{\mathbf{X}\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{X}^T}{\mathbf{e}^T\boldsymbol{\alpha}}) \\ \text{s.t.} \quad & \mathbf{e}^T\boldsymbol{\alpha} = n \\ & \boldsymbol{\alpha} \ge 0. \end{aligned} \tag{2.21}$$

It can be seen that the dual problem (2.4) and (2.21) differ only on the constraint $\mathbf{e}^T\boldsymbol{\alpha} = n$.

It is also possible to further succinctly rewrite the objective function (2.21). Since $\mathbf{XAX}^T - \frac{\mathbf{X}\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{X}^T}{\mathbf{e}^T\boldsymbol{\alpha}}$ is a Schur complement of $\begin{bmatrix} \mathbf{XAX}^T & \mathbf{X}\boldsymbol{\alpha} \\ \boldsymbol{\alpha}^T\mathbf{X}^T & \mathbf{e}^T\boldsymbol{\alpha} \end{bmatrix}$, and by using the fact that $\boldsymbol{\alpha} = \mathbf{Ae}$ and $\mathbf{A} = \mathbf{A}^T$, then we have

$$\begin{bmatrix} \mathbf{XAX}^T & \mathbf{X}\boldsymbol{\alpha} \\ \boldsymbol{\alpha}^T\mathbf{X}^T & \mathbf{e}^T\boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{XAX}^T & \mathbf{XAe} \\ \mathbf{e}^T\mathbf{A}^T\mathbf{X}^T & \mathbf{e}^T\mathbf{Ae} \end{bmatrix} = \begin{bmatrix} \mathbf{X} \\ \mathbf{e}^T \end{bmatrix} \mathbf{A} \begin{bmatrix} \mathbf{X}^T & \mathbf{e} \end{bmatrix} = \tilde{\mathbf{X}}\mathbf{A}\tilde{\mathbf{X}}^T \tag{2.22}$$

where $\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \mathbf{e}^T \end{bmatrix}$.

As a result, instead of solving (2.21) for a given $\mathbf{X}$, we can lift all examples $\mathbf{x}_i$ to $\tilde{\mathbf{x}}_i = \begin{bmatrix} \mathbf{x}_i & 1 \end{bmatrix}^T$ for all i, and solve

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \log\det(\tilde{\mathbf{X}}\mathbf{A}\tilde{\mathbf{X}}^T) \\ \text{s.t.} \quad & \mathbf{e}^T\boldsymbol{\alpha} = n \\ & \boldsymbol{\alpha} \geq 0. \end{aligned} \tag{2.23}$$

2.4 Support Vector based Classifiers

Support vector classifiers have a distinctive feature in that no assumption are made on the distribution of the data. They are normally formulated as a convex optimization problem and then transformed into their corresponding dual form to implement the kernel trick [15]. The reason why they are collectively called support vector based classifiers is that all of them embrace the notion of "support vectors". We primarily focus on linear SVM and linear TWSVM in this section. Support vector based classifiers which are based on hypersurfaces are introduced in Section 2.5

2.4.1 Support vector machine

Given a set of tuples of training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$ where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{+1, -1\}$, the pre-defined set of feasible decision functions that SVM implements is

$$\mathbf{w}^T\mathbf{x} + b = 0\,. \tag{2.24}$$

The goal of SVM is to minimize $\mathbf{w}^T\mathbf{w}$ which is reciprocal to the margin between two convex hulls [73] of two classes, $y_i = 1$ and $y_i = -1$.

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1,...,m \end{aligned} \tag{2.25}$$

In (2.25), ξ_i is a slack variable representing the empirical loss from the misclassification of example $\mathbf{x}_i$. Hence, the term $\sum_{i=1}^{m}\xi_i$ corresponds to empirical risk which is also another objective of minimization. The scalar $C \geq 0$ is the hyperparameter controlling the trade-off between the misclassification and the margin's width.

By Karush–Kuhn–Tucker (KKT) conditions, Lagrange's duality can be obtained as in (2.26),

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j \\ \text{s.t.} \quad & \sum_{i=1}^{m}\alpha_i y_i = 0, \qquad i = 1,...,m \\ & 0 \leq \alpha_i \leq C. \end{aligned} \tag{2.26}$$

Solving this optimization yields the Lagrange's multiplier $\boldsymbol{\alpha} \in \mathbb{R}^m$ where the optimal weight vector $\mathbf{w}$ can be found by

$$\mathbf{w} = \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i \tag{2.27}$$

Therefore, $\mathbf{w}$ is a linear combination of the training examples. Generally, $\boldsymbol{\alpha}$ is sparse i.e. most of its elements are zero. Consequently, only the non-zero ones contribute to $\mathbf{w}$. The training examples $\mathbf{x}_i$ whose α_i are non-zero are so-called "support vectors", and they construct the optimal hyperplane separating two classes of examples.

The optimal value of b in (2.24) which defines the optimal decision hyperplane can be obtained by the complementary slackness condition, $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$, using any examples whose α_i satisfies $0 < \alpha_i < C$. It can

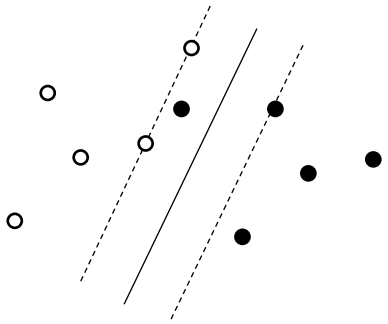


Figure 2.2 Rough sketch of SVM's decision boundary.

be observed that the hyperparameter C defines the so-called "box constraints" on the Lagrange multipliers $\boldsymbol{\alpha}$ in the dual problem (2.26). Since $\boldsymbol{\alpha}$ defines $\mathbf{w}$ in (2.27), it implies that C also controls the size of the hypothesis set of SVM. A sketch of SVM's decision boundary can be depicted in Figure 2.2 where ○ and ● represent the examples from two different classes. The solid line and the two dash lines show the decision boundary and its maximum margin, respectively.

2.4.2 Twin support vector machine

The idea of TWSVM entirely differs from SVM, although it also has the notion of support vectors. Therefore, some authors refer the family of TWSVM-like classifiers as the nonparallel plane classifiers [27] or the best fitting hyperplanes [28]. In fact, each hyperplane of TWSVM can be viewed as a data descriptor for one class of data. For binary classification, TWSVM solves two quadratic programming problems (QPP) where each problem tries to find the optimal hyperplane closest to one class but far from the other class with the distance at least one. Suppose the training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ are separated into two classes, class I_1 and I_2, according to their labels, with $m_1 = |\mathsf{I}_1|$ and $m_2 = |\mathsf{I}_2|$. Two QPPs of TWSVM can be formulated as shown in (2.28) and (2.29),

$$\begin{aligned} \min_{\mathbf{w}_1, b_1, \xi_2} \quad & \frac{1}{2}\sum_{i\in \mathsf{I}_1}(\mathbf{x}_i^T\mathbf{w}_1 + b_1)^2 + C_1\sum_{j\in \mathsf{I}_2}\xi_{2j} \\ \text{s.t.} \quad & -\sum_{j\in \mathsf{I}_2}(\mathbf{x}_j^T\mathbf{w}_1 + b_1) \geq 1 - \xi_{2j}, \\ & \xi_{2j} \geq 0,\ j \in \mathsf{I}_2 \end{aligned} \tag{2.28}$$

$$\begin{aligned} \min_{\mathbf{w}_2, b_2, \xi_1} \quad & \frac{1}{2}\sum_{j \in \mathrm{I}_2} (\mathbf{x}_j^T \mathbf{w}_2 + b_2)^2 + C_2 \sum_{i \in \mathrm{I}_1} \xi_{1i} \\ \text{s.t.} \quad & \sum_{i \in \mathrm{I}_1} (\mathbf{x}_i^T \mathbf{w}_2 + b_2) \geq 1 - \xi_{1i}, \\ & \xi_{1i} \geq 0, \; i \in \mathrm{I}_1 \end{aligned} \tag{2.29}$$

where C_1 and C_2 are the hyperparameters. $\boldsymbol{\xi}_1 \in \mathbb{R}^{m_1}$ and $\boldsymbol{\xi}_2 \in \mathbb{R}^{m_2}$ are the vectors of slack variables. $f_1(\mathbf{x}) = \mathbf{w}_1^T \mathbf{x} + b_1 = 0$ and $f_2(\mathbf{x}) = \mathbf{w}_2^T \mathbf{x} + b_2 = 0$ are the hyperplanes for class I_1 and I_2, respectively, with $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^n$ and $b_1, b_2 \in \mathbb{R}$. The examples in class I_1 are supposed to be closest to the hyperplane $f_1(\mathbf{x})$ while the examples in class I_2 are closest to $f_2(\mathbf{x})$ with the distance more than 1.

Suppose $\mathbf{X}_1$ and $\mathbf{X}_2$ are the matrices composed of the column vectors of the training data from class I_1 and I_2, respectively. The matrix forms of (2.28) and (2.29) thus become

$$\begin{aligned} \min_{\mathbf{w}_1, b_1, \xi_2} \quad & \frac{1}{2}\| \mathbf{X}_1^T \mathbf{w}_1 + \mathbf{e}_1 b_1 \|^2 + C_1 \mathbf{e}_2^T \boldsymbol{\xi}_2 \\ \text{s.t.} \quad & -(\mathbf{X}_2^T \mathbf{w}_1 + \mathbf{e}_2 b_1) \geq \mathbf{e}_2 - \boldsymbol{\xi}_2, \quad \boldsymbol{\xi}_2 \geq 0 \end{aligned} \tag{2.30}$$

$$\begin{aligned} \min_{\mathbf{w}_2, b_2, \xi_1} \quad & \frac{1}{2}\| \mathbf{X}_2^T \mathbf{w}_2 + \mathbf{e}_2 b_2 \|^2 + C_2 \mathbf{e}_1^T \boldsymbol{\xi}_1 \\ \text{s.t.} \quad & \mathbf{X}_1^T \mathbf{w}_2 + \mathbf{e}_1 b_2 \geq \mathbf{e}_1 - \boldsymbol{\xi}_1, \quad \boldsymbol{\xi}_1 \geq 0 \end{aligned} \tag{2.31}$$

where $\mathbf{e}_1$ and $\mathbf{e}_2$ are the vectors of ones with appropriate dimensions.

The dual formulations of (2.30) and (2.31) are, respectively, as follows,

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \mathbf{e}_2^T \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^T \tilde{\mathbf{X}}_2^T (\tilde{\mathbf{X}}_1 \tilde{\mathbf{X}}_1^T)^{-1} \tilde{\mathbf{X}}_2 \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq C_1 \mathbf{e}_2 \end{aligned} \tag{2.32}$$

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \mathbf{e}_1^T \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^T \tilde{\mathbf{X}}_1^T (\tilde{\mathbf{X}}_2 \tilde{\mathbf{X}}_2^T)^{-1} \tilde{\mathbf{X}}_1 \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq C_2 \mathbf{e}_1 \end{aligned} \tag{2.33}$$

where $\tilde{\mathbf{X}}_1$ and $\tilde{\mathbf{X}}_2$ are the augmented matrices with $\tilde{\mathbf{X}}_1 = [\mathbf{X}_1^T, \mathbf{e}_1]^T$ and $\tilde{\mathbf{X}}_2 = [\mathbf{X}_2^T, \mathbf{e}_2]^T$. Since $\tilde{\mathbf{X}}_1\tilde{\mathbf{X}}_1^T$ and $\tilde{\mathbf{X}}_2\tilde{\mathbf{X}}_2^T$ are generally singular, it is important to add a small identity matrix to avoid the ill-conditioned situation, i.e. we replace $\tilde{\mathbf{X}}_1\tilde{\mathbf{X}}_1^T$ with $\tilde{\mathbf{X}}_1\tilde{\mathbf{X}}_1^T + \delta\mathbf{I}$ and $\tilde{\mathbf{X}}_2\tilde{\mathbf{X}}_2^T$ with $\tilde{\mathbf{X}}_2\tilde{\mathbf{X}}_2^T + \delta\mathbf{I}$ where $\mathbf{I}$ is an identity matrix of appropriate dimensions and $\delta > 0$ is a rather small scalar.

After solving (2.32), the optimal hyperplane for class I_1 is obtained from

$$[\mathbf{w}_1^T, b_1]^T = -(\tilde{\mathbf{X}}_1\tilde{\mathbf{X}}_1^T + \delta\mathbf{I})^{-1}\tilde{\mathbf{X}}_2\boldsymbol{\alpha}. \tag{2.34}$$

Similarly, solving (2.33) and computing

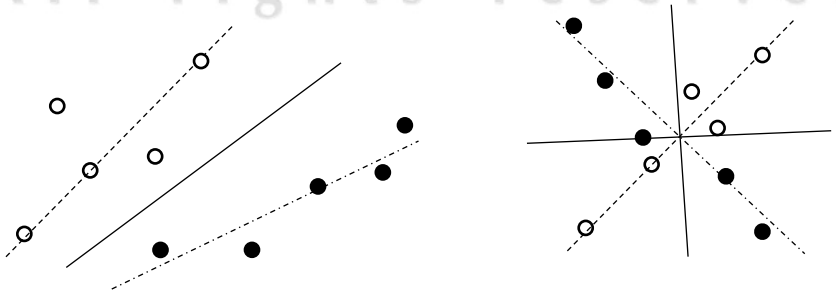
$$[\mathbf{w}_2^T, b_2]^T = -(\tilde{\mathbf{X}}_2\tilde{\mathbf{X}}_2^T + \delta\mathbf{I})^{-1}\tilde{\mathbf{X}}_1\boldsymbol{\alpha} \tag{2.35}$$

yield the optimal hyperplane for class I_2.

The classification rule for a testing example **x** is simply

$$\text{Class } i = \arg\min_{k=1,2} \frac{|\mathbf{w}_k^T\mathbf{x} + b_k|}{\|\mathbf{w}_k\|} \tag{2.36}$$

which is the shortest distance from a test example to the closest hyperplane. Each minimization of TWSVM is quite different from SVM in that the number of its constraints is equal to the number of examples in just one class as can be seen in (2.32) and (2.33). Thus, TWSVM requires solving smaller QPPs and is claimed to run approximately four time faster than SVM [10].



(a) Separable data (b) Cross-plane data

Figure 2.3 Rough sketch of TWSVM's decision boundary

Two naïve illustrations of the TWSVM's hyperplanes are shown in Figure 2.3. Figure 2.3(a) and 2.3(b) display two different sets of examples which are a simple separable case and a cross-plane case [24] of data. The examples from class I_1 and I_2 are denoted using ○ and ●, respectively. The dash lines and dot-dash lines illustrate the descriptive hyperplanes for class I_1 and I_2, and the solid line shows the decision boundary. According to Figure 2.3(b), it can be easily observed that linear TWSVM has a natural ability over linear SVM in classifying the cross-plane dataset.

2.5 Support Vector based Classifiers with Hypersurface

In the previous section, we introduce the supervised classifiers which construct the decision boundary using hyperplanes. To create more sophisticated decision rules while still not being overly-complicated, a hypersurface is considered one step further for the candidacy. Therefore, in this section, support vector based classifiers based on hyperspheres which include SVDD and THSVM are introduced.

2.5.1 Support vector data description

The idea of SVDD is to find the best hypersphere which covers a given set of examples. The area covered by the hypersphere can be interpreted as the domain which best describes the data. However, finding the hypersphere which exactly fits the entire set of examples is not a good option since some examples are possibly mere outliers. Therefore, the formulation of SVDD also allows some errors in the training examples. Given a set of examples $\{\mathbf{x}_i\}_{i=1}^m$ without labels, the simplest form of SVDD is simply the minimum enclosing hypersphere whose radius is r and center is at $\mathbf{c}$, formulated with soft margins. That is

$$\begin{aligned} \min_{r,\mathbf{c},\xi} \quad & r^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & \left\|\mathbf{x}_i - \mathbf{c}\right\|^2 \le r^2 + \xi_i, \\ & \xi_i \ge 0,\ i = 1,2,...,m \end{aligned} \tag{2.37}$$

where C is the hyperparameter. r is the radius of the hypersphere to be minimized with the constraints to enclose all the examples with some tolerable errors defined using the slack variables ξ_i.

Since SVDD uses the minimal hypersphere for data description, its solution is always a closed spherical boundary. Therefore, it is inherently suitable for one-class classification. The formulation of SVDD in (2.37) is the initial form of SVDD [12], and it is later extended to incorporate examples with labels as either targets or outliers [31]. In fact, the later version of SVDD is called SVDD with negative examples (nSVDD). Hence, it becomes a binary classifier where one class has abundant examples while examples in the other class, or also called outliers/novelties or negative examples, are so scarce or hard to be collected.

Given a set of m examples, $\{\mathbf{x}_i\}_{i=1}^m$, where each example is labeled with $y_i = 1$ for the target class and $y_i = -1$ for the outlier class, the formulation of nSVDD is to find a hypersphere whose radius is $r > 0$ and center is at $\mathbf{c} \in \mathbb{R}^n$ is as follows.

$$\begin{aligned} \min_{r,\mathbf{c},\boldsymbol{\xi}} \quad & r^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y_i \left\|\mathbf{x}_i - \mathbf{c}\right\|^2 \le y_i r^2 + \xi_i, \\ & \xi_i \ge 0,\ i = 1, 2, ..., m \end{aligned} \tag{2.38}$$

The use of slack variables $\boldsymbol{\xi} \in \mathbb{R}^m$ is again to form the so-called "soft margins". The hyperparameter $C > 0$ is also to control the trade-off between the volume of the hypersphere and the misclassification.

The dual formulation of (2.38) can be obtained as

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j - \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i^T\mathbf{x}_i \\ \text{s.t.} \quad & \sum_{i=1}^{m}\alpha_i y_i = 1, \\ & 0 \le \alpha_i \le \frac{C}{m},\ i = 1, 2, ..., m \end{aligned} \tag{2.39}$$

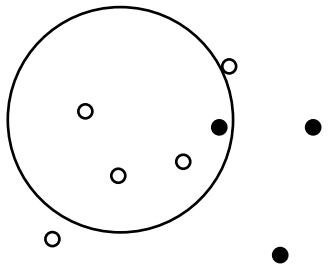


Figure 2.4 Rough sketch of nSVDD's descriptive boundary

by following the method of Lagrange multipliers where $\boldsymbol{\alpha} \in \mathbb{R}^m$ is the vector of Lagrange multipliers. The support vectors of nSVDD are also defined by any example whose value of Lagrange multiplier is in the set $(0, \frac{C}{m}]$.

For a test example $\mathbf{x} \in \mathbb{R}^n$, the classification rule of nSVDD is

$$f(\mathbf{x}) = \text{sign}\left(2\sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} - \mathbf{x}^T \mathbf{x} + h \right) \tag{2.40}$$

where $h = r^2 - \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ with r being the distance from the center $\mathbf{c}$ to any support vector whose α_i is in the set $0 < \alpha_i < C/m$. It is worth noting that, in order to obtain r, the actual value of $\mathbf{c}$ of the optimal hypersphere is not required to be computed. Let $\mathbf{x}_k$ be a support vector with $\alpha_k < C/m$, then we can obtain r^2 by

$$r^2 = \mathbf{x}_k^T \mathbf{x}_k - 2\sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i^T \mathbf{x}_k + \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j . \tag{2.41}$$

The dual formulation of nSVDD in (2.39) has similar form as the dual formulation of SVM. That is, it includes the box constraints and also the inner products between examples. The illustration of nSVDD is shown in Figure 2.4 where the circle depicts the descriptive boundary. ○ and ● are targets and outliers, respectively.

As a remark, in some works, instead of formulating SVDD as in (2.38), the distinction between allowing outliers to be inside the hypersphere and

targets to be outside the hypersphere is also made by using separate hyperparameters as in (2.42).

$$\begin{aligned} \min_{\mathbf{c},r,\boldsymbol{\xi},\tilde{\boldsymbol{\xi}}} \quad & r^2 + \frac{C_1}{m_1}\sum_{i\in \mathsf{I}_1}\xi_i + \frac{C_2}{m_2}\sum_{j\in \mathsf{I}_2}\tilde{\xi}_j \\ \text{s.t.} \quad & \left\|\mathbf{x}_i - \mathbf{c}\right\|^2 \le r^2 + \xi_i \qquad \xi_i \ge 0 \qquad i \in \mathsf{I}_1 \\ & \left\|\mathbf{x}_j - \mathbf{c}\right\|^2 \ge r^2 - \tilde{\xi}_j \qquad \tilde{\xi}_j \ge 0 \qquad j \in \mathsf{I}_2 \end{aligned} \tag{2.42}$$

Here, C_1 and C_2 are two hyperparameters. The first and second constraints are defined so as to allow misclassification on the targets and outliers, respectively. $\boldsymbol{\xi}$ and $\tilde{\boldsymbol{\xi}}$ are two separate slack variables for each constraint.

A simple multiclass classification scheme using SVDD is accomplished by performing SVDD defined in (2.38) on every class and the classification rule is based on the proximity to the closest hypersphere [33]. Some authors [32] even combine two SVDD formulations of (2.42) into one in order to solve general binary classification problems. That is, they simultaneously try to find two hyperspheres by using the following formulation,

$$\begin{aligned} \min_{\mathbf{c}_1,\mathbf{c}_2,r_1,r_2,\boldsymbol{\xi}_1,\boldsymbol{\xi}_2,\tilde{\boldsymbol{\xi}}_1,\tilde{\boldsymbol{\xi}}_2} \quad & r_1^2 + r_2^2 + \frac{C_1}{m_1}\sum_{i\in \mathsf{I}_1}\xi_{1i} + \frac{C_2}{m_2}\sum_{j\in \mathsf{I}_2}\tilde{\xi}_{1j} + \frac{C_3}{m_1}\sum_{i\in \mathsf{I}_1}\xi_{2i} + \frac{C_4}{m_2}\sum_{j\in \mathsf{I}_2}\tilde{\xi}_{2j} \\ \text{s.t.} \quad & \left\|\mathbf{x}_{1i} - \mathbf{c}_1\right\|^2 \le r_1^2 + \xi_{1i}, \qquad \xi_{1i} \ge 0, \qquad i \in \mathsf{I}_1 \\ & \left\|\mathbf{x}_{2j} - \mathbf{c}_1\right\|^2 \ge r_1^2 - \tilde{\xi}_{1j}, \qquad \tilde{\xi}_{1j} \ge 0, \qquad j \in \mathsf{I}_2 \\ & \left\|\mathbf{x}_{1i} - \mathbf{c}_2\right\|^2 \le r_2^2 + \xi_{2i}, \qquad \xi_{2i} \ge 0, \qquad i \in \mathsf{I}_2 \\ & \left\|\mathbf{x}_{2j} - \mathbf{c}_2\right\|^2 \ge r_2^2 - \tilde{\xi}_{2j}, \qquad \tilde{\xi}_{2j} \ge 0, \qquad j \in \mathsf{I}_1 \end{aligned} \tag{2.43}$$

where $(\mathbf{c}_1, r_1)$ and $(\mathbf{c}_2, r_2)$ denote the tuples of the center and radius of the two hyperspheres. C_1, C_2, C_3, and C_4 are hyperparameters.

2.5.2 Twin-hypersphere support vector machine

Developed by Peng and Xu [19], THSVM borrows the same concept from TWSVM. Instead of using two nonparallel hyperplanes, it implements two hyperspheres centered at $\mathbf{c}_1$ and $\mathbf{c}_2$ with radius r_1 and r_2, respectively. Although the formulation of THSVM is also closely related to the

formulation of SVDD as both methods use a hypersphere to represent a class of data, it is specifically designed for binary classification. In fact, its idea was initiated differently from the formulation (2.43) of SVDD. In TWSVM, a hyperplane is used as a class descriptor. However, in THSVM, the hyperplane is replaced or reformulated with a hypersphere. One hypersphere of THSVM, therefore, not only fits one class, but also is as far as possible from the other class.

Given a set of m examples, $\{\mathbf{x}_i\}_{i=1}^{m}$, where each example is a member of either class $\mathbf{I}_1$ or $\mathbf{I}_2$. Let the numbers of examples in each class be m_1 and m_2, respectively. The formulation of THSVM is to find two hyperspheres in $\mathbb{R}^n$ by solving a pair of quadratic programs. One hypersphere of THSVM to describe class $\mathbf{I}_1$ can be formulated as

$$\begin{aligned} \min_{r_1,\mathbf{c}_1,\xi} \quad & r_1^2 - \frac{\nu_1}{m_2}\sum_{j\in \mathbf{I}_2}\left\|\mathbf{x}_j - \mathbf{c}_1\right\|^2 + \frac{C_1}{m_1}\sum_{i\in \mathbf{I}_1}\xi_i \\ \text{s.t.} \quad & \left\|\mathbf{x}_i - \mathbf{c}_1\right\|^2 \le r_1^2 + \xi_i, \\ & \xi_i \ge 0,\ i \in \mathbf{I}_1 \end{aligned} \tag{2.44}$$

where r_1 and $\mathbf{c}_1$ are the radius and center of the hypersphere. $C_1 > 0$ and $\nu_1 > 0$ are the hyperparameters. Like other support vector based classifiers, THSVM also uses slack variables ξ to create the soft margins in addition to minimizing the hypersphere's radius. The constraints in (2.44) is defined such that feasible hyperspheres enclose the examples from class $\mathbf{I}_1$ with some tolerable errors allowing some examples to stay outside the hypersphere. In fact, the formulation (2.44) of THSVM is more or less modified from the formulation (2.38) of SVDD by entirely changing the excluding constraints for negative examples, or class $\mathbf{I}_2$, into the optimization objective in (2.44). In other words, in addition to find a hypersphere around class $\mathbf{I}_1$, it also tries to place the hypersphere as far as possible from class $\mathbf{I}_2$ controlled by the hyperparameter ν_1.

Suppose $\boldsymbol{\alpha} \in \mathbb{R}^{m_1}$ is the vector of Lagrange multipliers. The dual form of THSVM is as follows.

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & -\sum_{i_1 \in \mathbf{I}_1} \sum_{i_2 \in \mathbf{I}_2} \alpha_{i_1} \alpha_{i_2} \mathbf{x}_{i_1}^T \mathbf{x}_{i_2} + \sum_{i \in \mathbf{I}_1} \alpha_i \left[\frac{2\nu_1}{m_2} \sum_{j \in \mathbf{I}_2} \mathbf{x}_i^T \mathbf{x}_j + (1-\nu_1) \mathbf{x}_i^T \mathbf{x}_i \right] \\ \text{s.t.} \quad & \sum_{i \in \mathbf{I}_1} \alpha_i = 1, \\ & 0 \le \alpha_i \le \frac{C_1}{m_1}, \ i \in \mathbf{I}_1 \end{aligned} \tag{2.45}$$

when ν_1 is set to zero, the formulation of THSVM will become SVDD with only positive examples (or class $\mathbf{I}_1$). That is, all negative examples (or class $\mathbf{I}_2$) are completely ignored. The optimal center $\mathbf{c}_1$ can be computed from

$$\mathbf{c}_1 = \frac{1}{1-\nu_1} \left(\sum_{i \in \mathbf{I}_1} \alpha_i \mathbf{x}_i - \frac{\nu_1}{m_2} \sum_{j \in \mathbf{I}_2} \mathbf{x}_j \right) \tag{2.46}$$

and the optimal radius r_1 is

$$r_1 = \left\| \mathbf{x}_p - \mathbf{c}_1 \right\| \tag{2.47}$$

i.e. the distance from $\mathbf{c}_1$ to a support vector $\mathbf{x}_p$ whose α_p is the member of $0 < \alpha_p < \frac{C_1}{m_1}$.

After the optimal hypersphere describing class $\mathbf{I}_1$ is obtained, the optimal hypersphere the other class can also be similarly found by switching the role of $\mathbf{I}_1$ and $\mathbf{I}_2$. That is

$$\begin{aligned} \min_{r_2, \mathbf{c}_2, \xi} \quad & r_2^2 - \frac{\nu_2}{m_1} \sum_{j \in \mathbf{I}_1} \left\| \mathbf{x}_j - \mathbf{c}_2 \right\|^2 + \frac{C_2}{m_2} \sum_{i \in \mathbf{I}_2} \xi_i \\ \text{s.t.} \quad & \left\| \mathbf{x}_i - \mathbf{c}_2 \right\|^2 \le r_2^2 + \xi_i, \\ & \xi_i \ge 0, \ i \in \mathbf{I}_2. \end{aligned} \tag{2.48}$$

As a result, given a testing example $\mathbf{x} \in \mathbb{R}^n$, the decision rule of THSVM is defined as

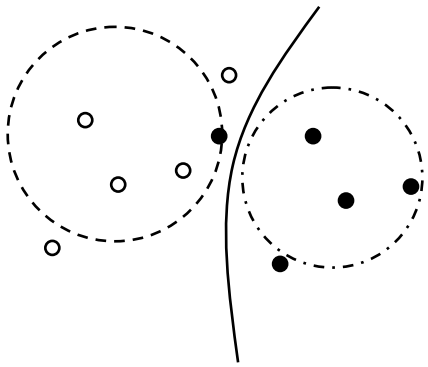


Figure 2.5 Rough sketch of THSVM's decision boundary

$$\text{Class } k = \arg\min_{k=1,2} \frac{\|\mathbf{x} - \mathbf{c}_k\|}{r_k}. \tag{2.49}$$

It is worth nothing that, although the decision rule (2.49) requires the calculation of $\mathbf{c}_k$ and r_k which can be obtained from (2.46) and (2.47), respectively, we can entirely avoid the direct computations by substituting (2.46) and (2.47) into (2.49) and rewriting them in terms of inner products between examples. The two hyperspheres of THSVM can be roughly illustrated as in Figure 2.5 where ○ and ● are the examples from class l_1 and l_2, respectively. The dash line is the descriptive boundary for class l_1 and the dot-dash line is the boundary for class l_2. The decision boundary is shown in the solid line.