Chapter 11

Support Vector Machines

"Science is the systematic classification of experience."
— **George Henry Lewes, Physical Basis of Mind**

Support vector machines (SVM) are extremely popular classifiers. SVM is the default choice classifier for most practical applications. It can be extended to solve regression problems as well.

Learning Objectives

- Introduce the concept of support vector machines and its advantages
- Derive hyperplanes and margins
- Introduce separable support vector machines
- Outline Lagrangian primal and dual optimization problems
- Explain soft margin support vector machines
- Illustrate the use of kernels and its types
- Introduce the concept of support vector Kernel Regression

11.1 INTRODUCTION TO SUPPORT VECTOR MACHINES

Scan for information on *'Decision Functions'* and for *'Additional Examples'*

SVM is a supervised learning algorithm that takes a labelled data as input and creates learning functions that can be used for classification of unknown test data as shown in Figure 11.1.

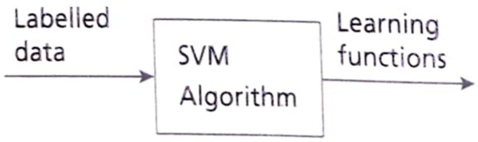

Figure 11.1: SVM Algorithm

Vladimir Vapnik, one of the main developers of the **Vapnik**–Chervonenkis theory of statistical learning, and the co-inventor of the support-vector machine method and support-vector clustering algorithm, provided a sound theoretical foundation for SVM. SVM is an extremely popular classifier and used in many practical applications. There are many advantages of SVM:

1. SVM can perform as a linear and non-linear classifier.
2. SVM can perform regression.
3. The decision boundary can be constructed using a small set of support vectors and can be constructed using less training samples. But this factor is dependent on the nature of the given application.
4. SVM works with higher dimensional data.
5. Generally, SVM classifiers are very robust and immune to the data features or dimensions.
6. SVM can be implemented in many applications such as object recognition, face recognition, Iris classification, and Pedestrian recognition successfully.

The aim of SVM is to produce a decision plane that defines the boundary between classes to classify the data points. In contrast to the decision theoretic minimum distance classifier that uses only one decision boundary, SVM uses a reference hyperplane and two decision boundaries to classify the points. This is the major difference between SVM and all other classifiers. This is shown in Figure 11.2 where there is a reference hyperplane shown with a dark line and two parallel boundaries in dotted lines.

The decision boundaries are drawn with the following characteristics:

- The decision boundary should be as far away from the data points, and hence the distance should be maximized between the line and the nearest data point.
- Should avoid misclassifications of data.

The first constraint indicates that the hyperplane should provide a wide margin. SVM is different from other classifiers as the focus of SVM is to maximize the margin as much as possible. A margin is defined as the amount of space between two classes as defined by the hyperplane. Formally, the distance between the hyperplanes is known as the margin of the classifier. The margin should be as large as possible. The focus of SVM is to obtain maximal margin classifier (MMH) often called 'hard SVM'.

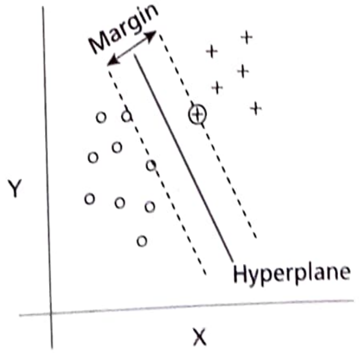

Figure 11.2: Margin Shown with Supporting Boundary Lines

It can be observed from Figure 11.1 that the selection of hyperplane is dependent on 'Support vectors'. Support vectors are those data points that fall on the supporting boundary lines. In a training dataset, all training samples that fall on the boundary lines are called support vectors.

A SVM implements a binary classifier. This means that there are only two classes, say +1 and −1. Later on, the idea of extending the two class SVM to multiclass SVM is discussed in the chapter. As of now, let us consider only two class problems.

Let us consider a dataset:

$D = \{(x_1, y_1), (x_2, y_2) \ldots, (x_i, y_i)\}, x \in \Re^n$, and y is the target label. $y = \{-1, +1\}$

The aim of a linear maximal margin support vector machine or Hard-margin SVM is to find a hyperplane that maximally separates the classes. From Box 11.1, one can check out that a hyperplane is an n-dimensional generalization of a line which is a set of points that satisfy the hyperplane equation:

$$h(x) = b + w_1x_1 + w_2x_2 + \cdots + w_nx_n = 0$$

$$\text{or } b + w^Tx = 0 \tag{11.1}$$

where, b is the intercept and $a_1, a_2, \cdots, a_n$ are coefficients and n is the dimension of the data point.

For a simple 2D, the hyperplane can be written as:

$$w \cdot x + b = 0 \tag{11.2}$$

Then, the hyperplane equation separates the data points x_i into two classes +1 and −1. This hyperplane equation can also be written as: $w^Tx = 0$ (See Box 11.1). Here, w is the weight vector and b is the bias or offset from the origin. This equation can be expressed better in terms of the dot vector product as $w \cdot x = 0$ for dimensions three and above.

The difference between SVMs and all other classifiers such as a decision theoretic classifier is that the SVM uses two decision lines constructed using the reference hyperplane. Two lines act as two classifiers. This is written as follows:

$$H_1: w \cdot x_i + b \geq 0 \text{ for } y = +1 \text{ and} \tag{11.3}$$

$$H_2: w \cdot x_i + b < 0 \text{ for } y = -1$$

This classifier is suitable for two class problems. One can also bring the output y into the Eq. (11.3) and that changes the equation to:

$h(x) = y_i(w \cdot x_i + b) \geq 1$ for $i = 1, 2, \cdots, n$ or one can alternatively can find predictions as:

$$h(x_i) = sign(w \cdot x_i + b) \tag{11.4}$$

The sign of $h(x)$ is always positive if it is correctly classified. Its value is negative if it is wrongly classified. The basics of hyperplane are given in Box 11.1.

11.2 OPTIMAL HYPERPLANE

What is an optimal hyperplane? Consider the following Figure 11.3, where a decision function separates the samples.

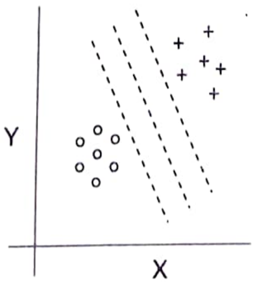

Figure 11.3: Multiple Hyperplanes for Separating the Data

There can be many hyperplanes separating the samples. Recollect that a hyperplane divides the input space into two half spaces, so that all samples (°) fall below the hyperplane and samples (+) fall above the line. The aim is to find the maximum margin. Each boundary is associated with a pair of hyperplanes, say h_1 and h_2. These parallels hyperplanes are obtained by moving the hyperplane h_1 till it touches (+), and h_2 till it touches (°). The hyperplane that creates the maximum margin is called optimal hyperplane.

The basics required for constructing the hyperplane are referred in Box 11.1.

Box 11.1: Basics of Constructing a Hyperplane

Any point $v = (v_1, v_2)$, $v \neq 0$, can be defined as a vector in a plane starting at the origin and ending at the given point. For example, the point (4, 3) can be represented as a vector shown in Figure 11.4.

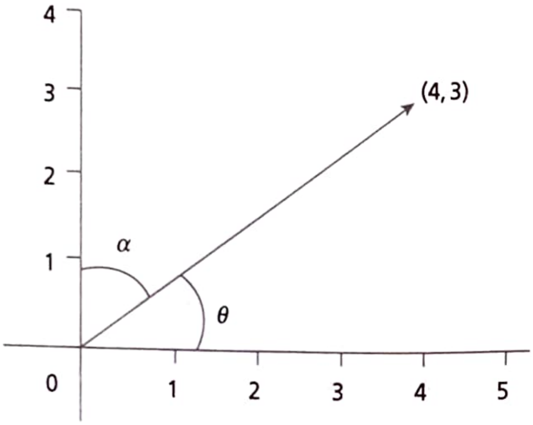

Figure 11.4: A Point (4, 3) in Vector Form

The vector, as discussed earlier, is an object that has both length and direction. The calculation of size of length of the vector is often required in machine learning. The length of the vector is called vector norm or vector magnitude. The length of the vector is often a positive number. The form of the vector norm is given as $\|v\|$. The L_2 norm or Euclidean norm is calculated as the Euclidean distance from the origin. It is expressed as the square root of the sum of squared vector values:

$$\|v\|_2 = \sqrt{a_1^2 + a_2^2} \tag{11.5}$$

Hyperplane Equations

All are familiar with line equation:

$$y = ax + b \tag{11.6}$$

Rewriting x and y as x_1 and x_2 in Eq. (11.6) gives the equation,

$$x_2 = a \times x_1 + b, a \times x_1 - x_2 = 0. \tag{11.7}$$

This Eq. (11.7) can be written as $w^T x = 0$. Here, the vector w determines the orientation of the hyperplane and b is the bias or intercept or offset from the origin. If the weight vectors $w = (b, a, -1)^T$ and $x = (1, x_1, x_2)^T$, then one can check $w^T x = y - ax - b = 0$. Thus, these equations are equivalent.

(Continued)

One can use vector dot notation $w \cdot x = 0$ instead of $w^T x = 0$ as the above vectors w and x are equivalent to $w \cdot x = 0$. This is suitable for data whose dimension is 3 or more.

The distance of a point x and the hyperplane, say $h(x)$, is given as $r = \frac{h(x)}{\|w\|}$. Here, r is called directed distance. The aim of SVM is to maximize r for the points that are closest to the optimal hyperplane. These points are called support vectors. The intuition is that the best or optimal hyperplane is that falls exactly in the middle of the two classes of data. 'r' is positive if x is on the positive side of the hyperplane and it is negative if it is on the negative side of the hyperplane. One can multiply it by class labels as [+1, −1] for binary classifiers. This gives:

$$yr = \frac{yh(x)}{\|w\|} \tag{11.8}$$

The expression can be simplified as the numerator just indicates the class +1 or −1 and the denominator does not change at all. So, one can take the absolute value to remove the numerator of Eq. (11.8) as 1. So, the simplified expression is:

$$|r| = \frac{1}{\|w\|} \tag{11.9}$$

The aim is, therefore, to maximize r of the points that are closest to the hyperplane. Such data points are called support vectors.

11.3 FUNCTIONAL AND GEOMETRIC MARGIN

Given the dataset D, one can compute the functional margin f. Functional margin is a function that can indicate the classification or misclassification of a data point, but it cannot indicate whether the point is close or far from the hyperplane. The functional margin of a data or example is computed as follows:

$$f = \min_{i=1\ldots n} f_i$$

Here, f is called the functional margin of example or data.

And functional margin of the entire dataset F can be computed as follows:

$$F = \min_{i=1\ldots n} y(w \cdot x_i + b) \tag{11.10}$$

The closeness or farness of the data point from the hyperplane is indicted through the geometric margin by the magnitude of the distance in terms of $\|w\|$. In short, the geometric margin is a scaled version of the functional margin by a factor of $\|w\|$. The normalized equation of $f = y(w \cdot x) + b$ becomes:

$$\gamma = y\left(\frac{w}{\|w\|} \cdot x + \frac{b}{\|w\|}\right) \tag{11.11}$$

For the dataset, one can say, $M = \min_{i=1\ldots n} \gamma_i$. Here, γ is called the geometric margin of dataset D. The relationship between the functional and geometric margin is given as:

$$M = \frac{F}{\|w\|} \tag{11.12}$$

Here, M is called geometric margin of the entire dataset.

Maximizing the geometric margin (M) does not depend on the scale of w and b (Geometric margin is scale invariant) and hence one can rescale w and b such that $F = 1$. Therefore, the geometric margin of Eq. (11.12) can be given as:

$$M = \frac{F}{\|w\|} = \frac{1}{\|w\|} \qquad (11.13)$$

The distance between any point and hyperplane is $\frac{1}{\|w\|}$, where, $\|w\|$ is the norm. This is equal to the distance from any point on the other side as also $\frac{1}{\|w\|}$. Therefore, the maximum margin is given as $\frac{2}{\|w\|}$. The maximal margin classifier now can be expressed as an optimization problem as:

$$\text{Max} \frac{2}{\|w\|} \text{ subjected to the constraint } y_i\,(w \cdot x_i + b) \geq 1,\ \forall x_i \in D. \qquad (11.14)$$

Maximizing $\frac{2}{\|w\|}$ is equivalent to minimizing $\|w\|$. If $\|w\|$ is large, then $\frac{2}{\|w\|}$ is small and when $\|w\|$ is small, then $\frac{2}{\|w\|}$ is large. Since our aim is to maximize $\frac{2}{\|w\|}$, $\|w\|$ serves as an error function called distance error function.

If the norm of the line is large, then $\frac{2}{\|w\|}$ is small and vice versa. Since large distance is preferable, one must find a line whose norm is small. If so, the constructed classifier is good. The following problem will help to understand the role of the distance error function.

Example 11.1: The hyperplane function for two variables is $b + a_1x_1 + a_2x_2$. If two hyperplanes given are for classifier 1 as $5 + 2x_1 + 5x_2$ and $5 + 20x_1 + 50x_2$, for classifier 2. Find the distance error function and pick a good classifier constructed using these hyperplanes?

Solution: This norm is the distance error. It is given as:

$$\sqrt{a_1^2 + a_2^2} \qquad (11.15)$$

The weight vector for the first equation omitting the intercept is (2, 5), and using Eq. (11.15), one get the norm as:

$\|w\| = \sqrt{2^2 + 5^2}$, which is approximately 5.39.

For the second equation with weight vector (20, 50), using Eq. (11.15),

$\|w\| = \sqrt{20^2 + 50^2} = 53.85$

The distance between the lines can be calculated for classifier 1, as $\frac{2}{\|w\|} = \frac{2}{5.39} = 0.37$, and for the second one as $\frac{2}{53.85} = 0.037$. It can be noticed that if the distance error is large, then $\frac{2}{\|w\|}$ is small and vice versa. Since distance error 5.39 is smaller than 53.85, the first equation is preferable and hence the classifier that uses this hyperplane is a good classifier.

Example 11.2: Points (4, 1), (4, –1) and (6, 0) belong to class positive and points (1, 0), (0, 1) and (0, –1) belong to negative class. Draw an optimal hyperplane to classify the points.

Solution: The scatter plot of the data points is shown in Figure 11.5.

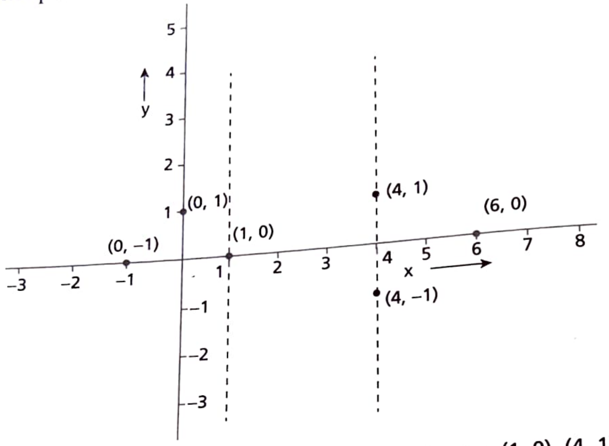

Figure 11.5: Scatter Plot of the Points with the Support Vectors (1, 0), (4, 1) and (4, –1)

It can be observed that the support vectors are (1, 0), (4, 1) and (4, –1) as shown below:

$$s_1 = \begin{pmatrix}1\\0\end{pmatrix}, s_2 = \begin{pmatrix}4\\1\end{pmatrix}, s_2 = \begin{pmatrix}4\\-1\end{pmatrix}$$

The augmented vector can be obtained by adding the bias given as follows:

$$\tilde{s}_1 = \begin{pmatrix}1\\0\\1\end{pmatrix}, \tilde{s}_2 = \begin{pmatrix}4\\1\\1\end{pmatrix}, \tilde{s}_2 = \begin{pmatrix}4\\-1\\1\end{pmatrix}$$

From these, a set of three equations can be obtained based on these three support vectors as follows:

$$\begin{aligned}\alpha_1\tilde{s}_1\tilde{s}_1 + \alpha_2\tilde{s}_2\tilde{s}_1 + \alpha_3\tilde{s}_3\tilde{s}_1 &= -1\\ \alpha_1\tilde{s}_1\tilde{s}_2 + \alpha_2\tilde{s}_2\tilde{s}_2 + \alpha_3\tilde{s}_3\tilde{s}_2 &= +1\\ \alpha_1\tilde{s}_1\tilde{s}_3 + \alpha_2\tilde{s}_2\tilde{s}_3 + \alpha_3\tilde{s}_3\tilde{s}_3 &= +1\end{aligned} \qquad (11.16)$$

It can be observed that in equation 1, $\tilde{s}_1$ is constant, in equation 2, $\tilde{s}_2$ is constant and in equation 3, $\tilde{s}_3$ is constant. The first equation is for (1, 0) that belongs to the negative class –1 and equation 2 and equation 3 are for the positive class +1.

Substituting the augmented support vectors in the Eq. (11.16) yields:

$$\alpha_1\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha_2\begin{pmatrix}4\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha_3\begin{pmatrix}4\\-1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix}$$

$$= 2\alpha_1 + 5\alpha_2 + 5\alpha_3 = -1$$

$$\alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}4\\1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}4\\1\\1\end{pmatrix}\begin{pmatrix}4\\1\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}4\\-1\\1\end{pmatrix}\begin{pmatrix}4\\1\\1\end{pmatrix}$$

$$= 5\alpha_1 + 18\alpha_2 + 16\alpha_3 = +1$$

$$\alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}4\\-1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}4\\1\\1\end{pmatrix}\begin{pmatrix}4\\-1\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}4\\-1\\1\end{pmatrix}\begin{pmatrix}4\\-1\\1\end{pmatrix}$$

$$= 5\alpha_1 + 16\alpha_2 + 18\alpha_3 = +1$$

Solving these three simultaneous equations with three unknowns yields the values:

$$\alpha_1 = -3$$

$$\alpha_2 = +1$$

$$\alpha_3 = 0$$

The optimal hyperplane vectors are given as:

$$w = \sum_{1}^{3} \alpha_i \times \tilde{s}_i$$

$$= -3 \times \begin{pmatrix}1\\0\\1\end{pmatrix} + 1 \times \begin{pmatrix}4\\1\\1\end{pmatrix} + 0 \times \begin{pmatrix}4\\-1\\1\end{pmatrix} = \begin{pmatrix}1\\1\\-2\end{pmatrix} \qquad (11.17)$$

The hyperplane is (1, 1) with an offset –2. The optimal hyperplane is shown in Figure 11.6.

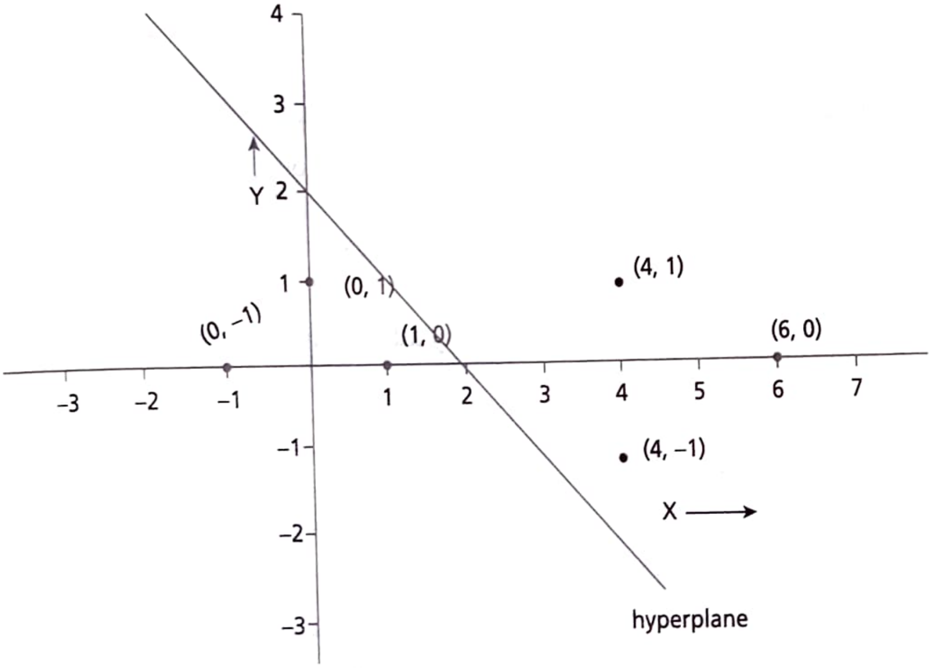

Figure 11.6: Scatter Plot of the Points with Hyperplane

11.4 HARD MARGIN SVM AS AN OPTIMIZATION PROBLEM

In example 11.2, the data points are less. So, one can visually find out the support vectors easily and construct the hyperplane. In practice, it is not feasible as involved data is huge. So, the method

The rest of the implementation of soft margin SVM is same as the hard margin SVM.

As usual, the primal Lagrangian optimization can now be formulated as:

$$L = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left(y_i(w\cdot x_i + b) - 1 + \xi_i - \sum_{i=1}^{n}\beta_i\xi_i\right) \tag{11.30}$$

Here, α_i and β_i are independent undetermined Lagrangian coefficients. Since the constraints involved are inequality constraints, like hard-margin SVM classifiers, KKT conditions are used. Using the KKT conditions, one can compute w and b using the partial derivatives as in the hard-margin SVM classifier. KKT conditions are discussed in section 11.4.1.

Now, differentiating the Lagrange function with respect to KKT conditions of Eq. (11.23), one gets:

$$\nabla_w L = w - \sum_{i=1}^{n}\alpha_i y_i x_i \text{ and}$$
$$\therefore w = \sum_{\alpha_i>0}\alpha_i y_i x_i \text{ and} \tag{11.31}$$

Differentiating with respect to b and equating it to zero, one gets:

$$\frac{\delta L}{\delta b} = -\sum_{\alpha_i}^{n}\alpha_i y_i = 0$$

And differentiating with respect to slack variables, one gets:

$$\frac{\delta L}{\delta \xi} = C - \alpha_i - \beta_i = 0$$
$$\therefore C = \alpha_i + \beta_i \tag{11.32}$$

This can be substituted back to get the dual formulation and the dual Lagrangian objective function is given as follows:

$$\max_\alpha \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i \cdot x_j \tag{11.33}$$

Subjected to the constraints:

$$0 \leq \alpha_i \leq C,\ x_i \in D \text{ and } \sum_{i=1}^{n}\alpha_i y_i = 0.$$

This is same as the hard-margin SVM classifiers. The only difference is the additional constraint, $C = \alpha_i + \beta_i$, that is, $\alpha + \beta = C$. Since, $\alpha, \beta \geq 0$, C serves as the upper bound. This is called box constraint. The value of C should be selected properly. So, there should be a trade-off between the accuracy of data fit and regularization. Optimal choice of C depends on the nature of dataset and given problem. The optimal value of C can be found from cross-validation.

For a new point z, one can compute using Eq. (11.34) as follows:

$$\hat{y} = \text{sign}(w \cdot z + b) \tag{11.34}$$

where, $w = \sum_{i=1}^{n}\alpha_i y_i x_i$ and $b = \text{avg}_{\alpha_i>0}\left(\frac{1}{y_i} - wx_i\right)$.

11.6 INTRODUCTION TO KERNELS AND NON-LINEAR SVM

In machine learning applications, the data can be text, image, sequence, or video. So, there is a need to extract features from these data prior to classification. Hence, in the real world, many same as the hard margin SVM.

be formulated as:

$1 + \xi_i - \sum_{i=1}^{n} \beta_i \xi_i \Big)$ (11.30)

gian coefficients. Since the constraints classifiers, KKT conditions are used. sing the partial derivatives as in the n section 11.4.1.

ect to KKT conditions of Eq. (11.23),

ne gets:

(11.31)

:

(11.32)

nd the dual Lagrangian objective

(11.33)

y difference is the additional the upper bound. This is called e should be a trade-off between pends on the nature of dataset -validation.

(11.34)

NEAR SVM

nce, or video. So, there is a ce, in the real world, many

classification models are complex and mostly require non-linear hyperplanes. Consider the Figure 11.9 where the samples cannot be separated by a linear hyperplane and require a non-linear hyperplane. One solution is to map the data into a higher-dimensional space and define a separating hyperplane.

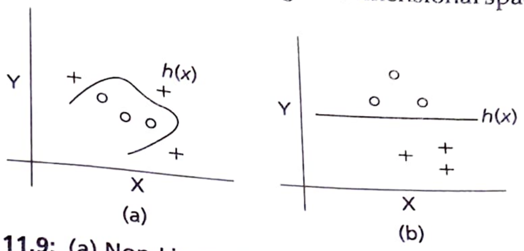

Figure 11.9: (a) Non-Linear Hyperplane (b) Need for Mapping

The mapping process, denoted as $\phi(x)$, is the vector representation of the feature x. In short, mapping transforms a data point in the input space and maps to another point in a space called feature space. Normally, mapping functions are used to map data from a lower dimension to a higher dimension. In Figure 11.9 (b), it can be observed that when the data is mapped from 2D to another feature space, the data points are nicely segregated in different planes and hence ca be separated by a plane. So, mapping functions play an important role in non-linear classificatio

For example, one mapping function $\varphi: \Re^2 \rightarrow \Re^3$ used to transform a 2D data to 3D data is give as follows:

$$\varphi(x,y) = (x_1^2, \sqrt{2}xy, y^2) \quad (11.3$$

Example 11.3: Consider a point (2, 3) and apply the mapping $\varphi(x, y) = (x^2, \sqrt{2}xy, y^2)$ to g 3D data.

Solution: One can get 3D data by plugging in the values as $x = 2$ and $y = 3$ in Eq. (11.35 $\varphi(2, 3) = (2^2, \sqrt{2} \times 2 \times 3, 3^2) = (4, 6\sqrt{2}, 9)$.

While mapping functions play an important role, there are many disadvantages, as map involves more computations and learning costs. Also, the disadvantages of transformations ar there is no generalized thumb rule available describing what transformations should be ap and if the data is large, mapping process takes huge amount of time.

In this context, only kernels are useful. Kernels are used to compute the value withou forming the data. What is a Kernel? Kernels are a set of functions used to transform data fro dimension to higher dimension and to manipulate data using dot product at higher dimen

The use of kernels is to apply transformation to data and perform classification at th dimension as shown in Figure 11.10.

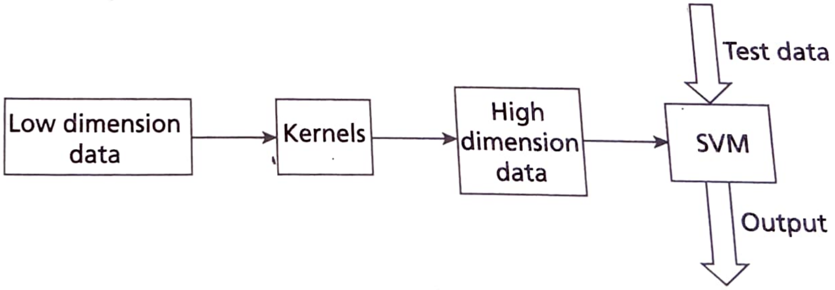

Figure 11.10: Role of Kernels

Kernels are of different types, as discussed below.

Linear Kernel

Linear kernels are of the type $k(x, y) = x^T y$ where x and y are two vectors. Therefore,

$$k(x, y) = \phi(x)\ \phi(y) = x^T y \qquad (11.36)$$

Polynomial Kernel

Polynomial kernels are of the type:

$$k(x, y) = (x^T \cdot y)^q. \qquad (11.37)$$

This is called homogeneous kernel. Here, q is the degree of the polynomial. If $q = 2$, then it is called quadratic kernel. For inhomogeneous kernels, this is given as:

$$k(x, y) = (c + x^T \cdot y)^q. \qquad (11.38)$$

Here, c is a constant and d is the degree of the polynomial.

If c is zero and degree is one, the polynomial kernel is reduced to a linear kernel. The value of degree d should be optimal as more degree may lead to overfitting.

Example 11.4: Consider two data points $x = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and $y = (2,3)$. Apply homogeneous and inhomogeneous kernels $k(x, y) = (x^T \cdot y)^q$.

Solution: For this, if $q = 1$ in Eq. (11.37), one gets a linear kernel. If $q = 2$, then the resultant of Eq. (11.37) is called a quadratic kernel.

Thus, the linear kernel is given by substituting $q = 1$ in Eq. (11.37) as:

$$\begin{aligned} k(x, y) &= x^T \cdot y \\ &= \begin{pmatrix} 1 \\ 2 \end{pmatrix}^T \cdot (2,3) \\ &= 8 \end{aligned}$$

The inhomogeneous kernel is given by substituting the value of c as 1 and x in Eq. (11.38) to get,

$$\begin{aligned} k(x, y) &= (1 + x^T \cdot y) \\ &= \left(1 + \begin{pmatrix} 1 \\ 2 \end{pmatrix}^T (2,3)\right) \\ &= (1 + 8) = 9 \end{aligned}$$

If $q = 2$, then the kernels are called quadratic kernels. The quadratic kernel based on the result of the linear kernel is given as follows:

$$\begin{aligned} k(x,y) &= (x^T \cdot y)^2 \\ &= (8)^2 = 64 \end{aligned}$$

And the inhomogeneous kernel is given as:

$$\begin{aligned} k(x, y) &= (1 + x^T \cdot y)^2 \\ &= (9)^2 = 81 \end{aligned}$$

Gaussian Kernel

RBFs or Gaussian kernels are extremely useful in SVM. RBF stands for radial basis functions. The RBF function is shown as below:

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \quad (11.39)$$

Here, γ is an important parameter. If γ is small, then the RBF is similar to linear SVM and if γ is large, then the kernel is influenced by more support vectors. The RBF performs the dot product performed in $\Re^{\infty}$, and therefore, it is highly effective in separating the classes and is often used.

Example 11.5: Consider two data points $x = (1, 2)$ and $y = (2, 3)$. Apply RBF kernel and find the value of RBF kernel for these points.

Solution: Substitute the value of x and y in RBF kernel as given in Eq. (11.39). The squared distance between the points (1, 2) and (2, 3) is given as:

$$(1 - 2)^2 + (2 - 3)^2 = 2$$

If $\sigma = 1$, then $k(x, y) = \exp\{-2/2\} = \exp\{-1\} = 0.3679$.

Sigmoid Kernel

The sigmoid kernel is given as:

$$k(x_i, x_j) = \tanh\,(kx_i x_j - \delta) \quad (11.40)$$

Kernel Operations

Every kernel operation has an equivalent mapping function. For example, the norm of the vecto can be performed as:

$$k(x, x) = x^T x \quad (11.41$$

Example 11.6: Find the norm of the point (1, 2) and find the distance between the points (1, and (2, 3).

Solution: The norm of the point (1, 2) is given by conventional means as $\sqrt{1^2 + 2^2} = \sqrt{5}$. By usi Eq. (11.41), one gets:

$$k(1,\ 2) = \sqrt{k(x,\ x)} = \sqrt{\begin{pmatrix}1\\2\end{pmatrix}^T (1,\ 2)} = \sqrt{5}.$$

Kernel Trick

Kernel trick means replacing the dot product in mapping functions with a kernel functio example, the kernel $k(x, y)$ given in Eq. (11.42) below and mapping functions are the same.

$$k(x, y) = \phi(x) \cdot \phi(y) \quad ($$

As usual, kernels help as mapping functions in the non-linear separation of samples a mapping data from input space to higher-dimensional feature space with least computa

The gain is done by replacing the dot product of the mapping function with the kernel operation, as performing the kernel operation is much easier. This is illustrated in the following numerical example.

Example 11.7: Consider two data points (1, 2) and (2, 3) Apply a polynomial kernel $k(x, y) = (x^T y)^2$ and show that it is equivalent to mapping function $\phi = (x^2, y^2, \sqrt{2}xy)$.

Solution: The mapping function is given as $\phi = (x^2, y^2, \sqrt{2}xy)$. (11.43)

Let us apply the mapping function first for the first data point (1, 2) using Eq. (11.44) given as:

$$\phi(x) = (1^2, 2^2, \sqrt{2} \times 1 \times 2) = (1, 4, 2\sqrt{2})$$

For the second data point (2, 3) the mapping is given as per Eq. (11.43):

$$\phi(y) = (3^2, 4^2, \sqrt{2} \times 3 \times 4) = (9, 16, 12\sqrt{2})$$

Now the mapping function $\phi(x)\phi(y)$ yields:

$$\phi(x)\phi(y) = (1, 4, 2\sqrt{2})\begin{pmatrix} 9 \\ 16 \\ 12\sqrt{2} \end{pmatrix} = (1 \times 9 + 4 \times 16 + 24(2) = 121$$

It can be seen the computation involves many multiplication operations. The operation can be computed quickly using kernel functions. Now, using polynomial kernel function, $k(x, y) = (x^T y)^2$, it can be computed as:

$$\left(\begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot \begin{pmatrix} 3 & 4 \end{pmatrix}\right)^2 = 11^2 = 121$$

It can be observed that kernel operations are easy as compared to plain mapping functions. Therefore, kernel trick can be applied to replace the dot product operations of mapping functions with the kernels. The clever selection of kernels can solve the problem of non-linearity. This is the reason why kernels are referred to as measures of similarity.