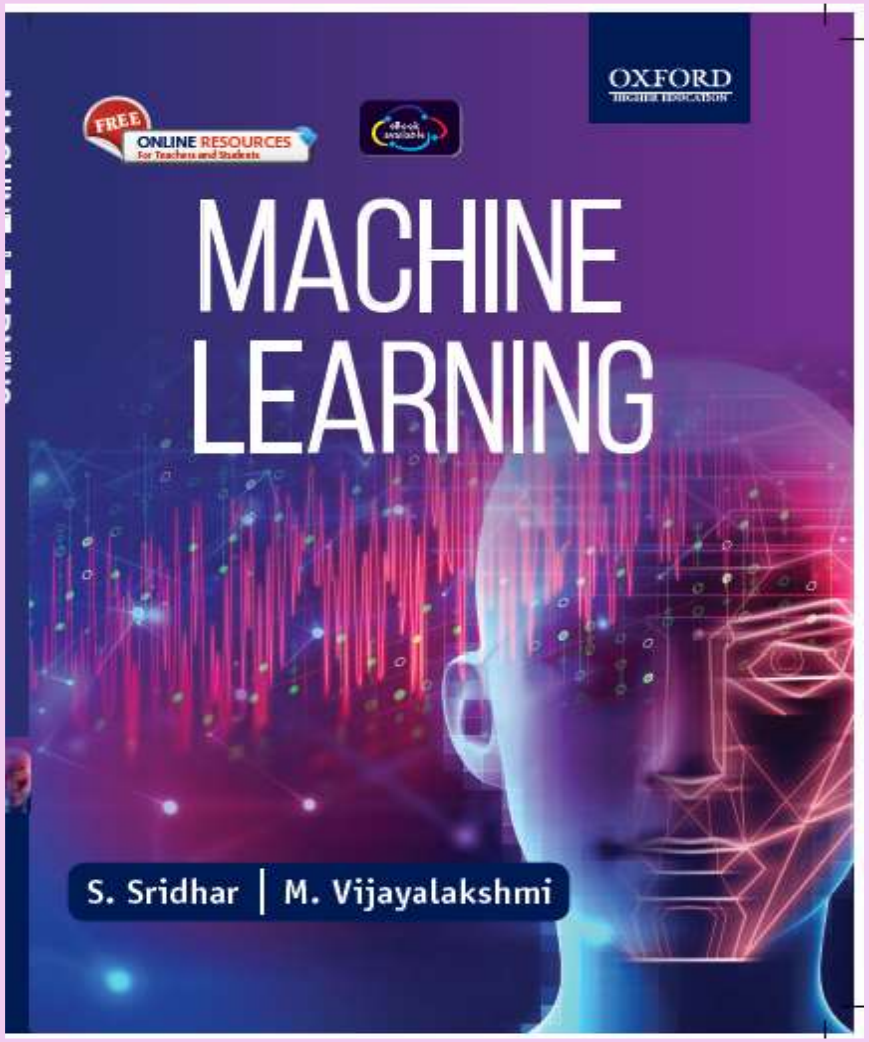


OXFORD
UNIVERSITY PRESS

Machine Learning

S. Sridhar and M. Vijayalakshmi

Chapter 11

Support Vector Machines

What is SVM?

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.

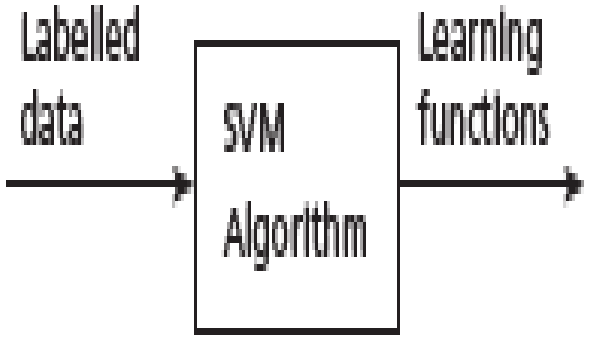


Figure 11.1: SVM Algorithm

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.

Applications of Regression Analysis

There are many applications of regression analysis. Some of the applications of regressions include predicting:

1. Sales of a goods or services
2. Value of bonds in portfolio management
3. Premium on insurance companies
4. Yield of crops in agriculture
5. Prices of real estate

Widest Margin Classifier

SVM CLASSIFIERS HAVE WIDEST MARGIN HYPERPLANES

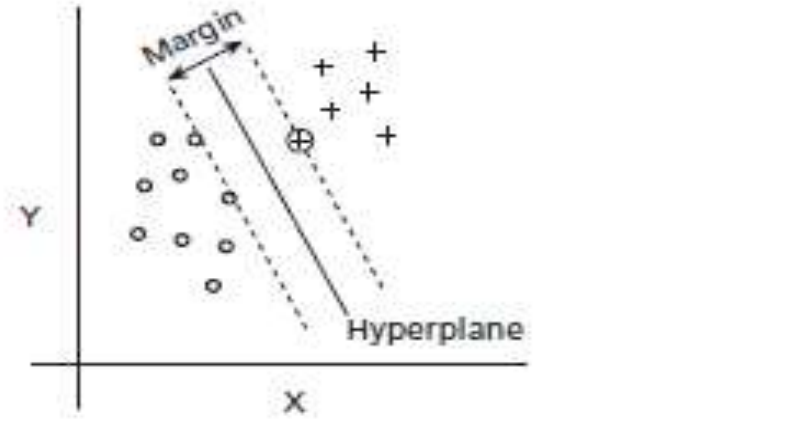


Figure 11.2: Margin Shown with Supporting Boundary Lines

Hyperplane

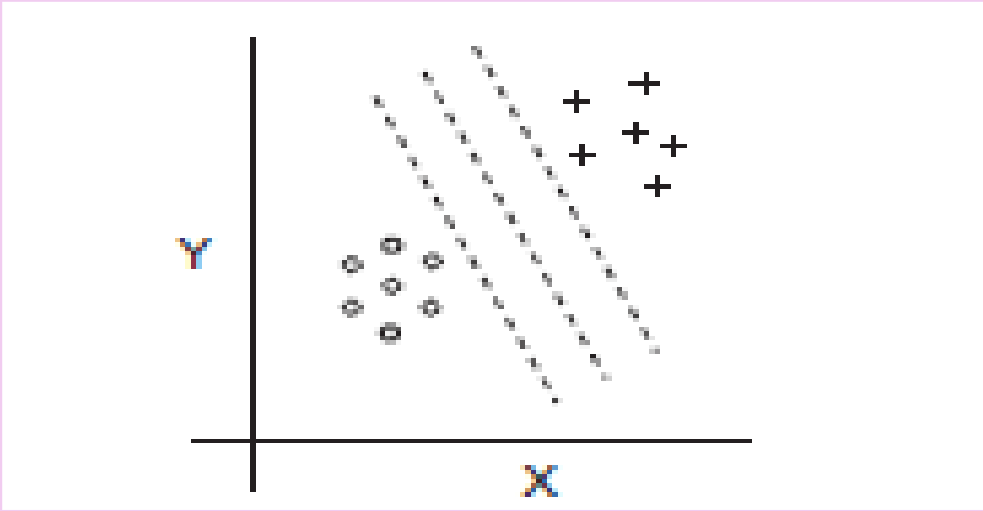
$$h(x) = b + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = 0$$
$$\text{or } b + w^T x = 0$$

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

$$h(x_i) = sign(w \cdot x_i + b) \quad (11.4)$$

Hyperplanes

- MANY HYPERPLANES ARE POSSIBLE. BUT SVM WILL HAVE AN OPTIMAL HYPERPLANE



Functional Margin

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

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

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

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

Geometric Margin

$$M = \frac{F}{\|w\|} = \frac{1}{\|w\|}$$

SVM as Optimization Problem

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

Example

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:

$$\sqrt{a_1^2 + a_2^2} \tag{11.15}$$

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

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.

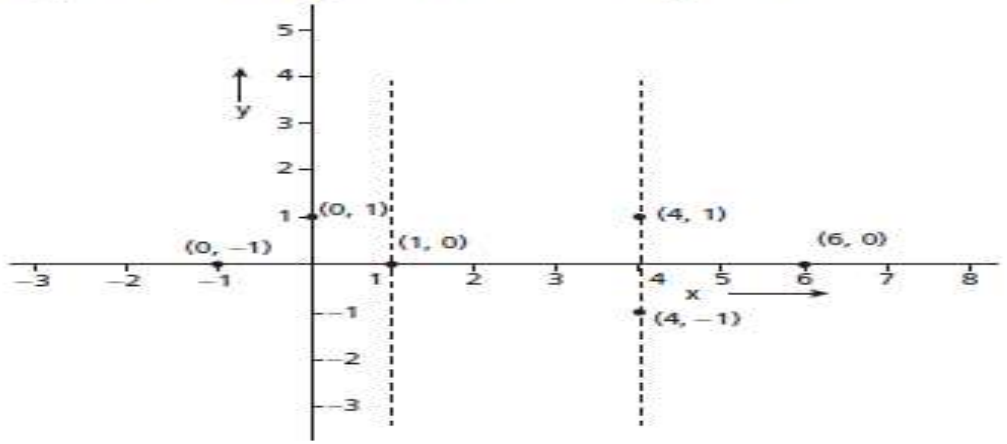


Figure 11.5: Scatter Plot of the Points with the Support Vectors (1, 0), (4, 1) and (4, −1)

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:

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

Example

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

$$\begin{aligned}
\alpha_1\bar{s}_1\bar{s}_1 + \alpha_2\bar{s}_2\bar{s}_1 + \alpha_3\bar{s}_3\bar{s}_1 &= -1 \\
\alpha_1\bar{s}_1\bar{s}_2 + \alpha_2\bar{s}_2\bar{s}_2 + \alpha_3\bar{s}_3\bar{s}_2 &= +1 \\
\alpha_1\bar{s}_1\bar{s}_3 + \alpha_2\bar{s}_2\bar{s}_3 + \alpha_3\bar{s}_3\bar{s}_3 &= +1
\end{aligned} \tag{11.16}$$

It can be observed that in equation 1, $\bar{s}_1$ is constant, in equation 2, $\bar{s}_2$ is constant and in equation 3, $\bar{s}_3$ 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:

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

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

Example

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

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

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

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

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:

$$w = \sum_1^3 \alpha_i \times \bar{s}_i$$

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

Hyperplane

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

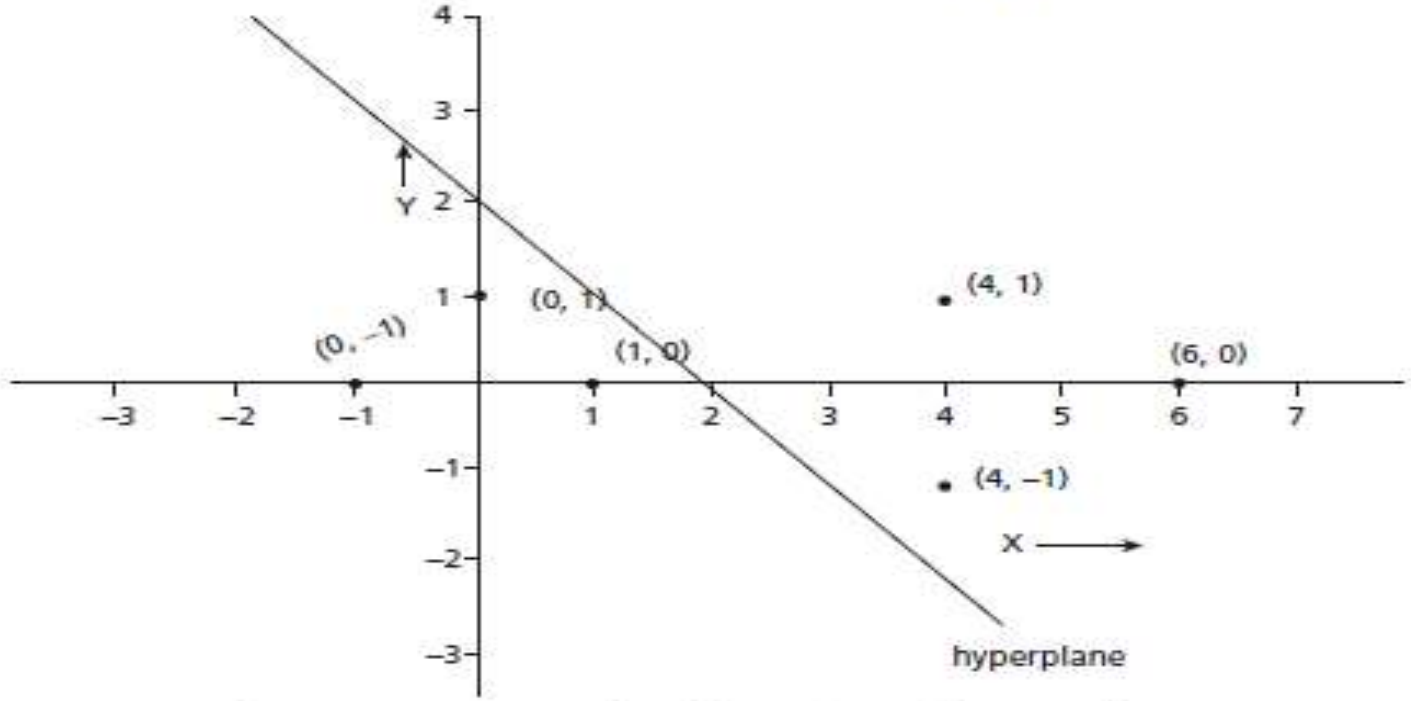


Figure 11.6: Scatter Plot of the Points with Hyperplane

Hard Margin SVM

1. State the problem as a primal optimization problem.
2. Convert that to dual Lagrangian problem. Finally, using the solutions, construct the hyperplanes.

The optimization problem now can be stated by adding the constraint to Eq. (11.18) as follows:

$$\text{Minimize}_{w,b} \frac{1}{2}\|w\|^2$$

Subject to the constraint:

$$y_i(w \cdot x_i) + b - 1 \geq 0, \text{ for } \forall x_i \in D$$

This is called *primal optimization problem.*

Lagrangian Problem

The Lagrangian problem can now be formulated as a problem as follows:

$$\text{Minimize}_{w,b} \max_{\alpha} \mathcal{L}(w, b, \alpha)$$
$$\text{subject to } \alpha_i \geq 0 \text{ For } \forall_i, i = 1 \ldots n. \quad (11.20)$$

KKT Conditions

The KKT conditions are given below:

1. $\nabla_{w} \mathcal{L}(w, b, \alpha) = 0$ and $\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0$ (Stationary Condition)
2. $\alpha_i \geq 0$ (Dual Feasibility Condition)
3. $\alpha_i (y_i(wx_i + b) - 1) = 0$ (Complementary Slackness condition)
4. $y_i (wx_i + b) - 1) \geq 0$ (Primal feasible Condition) (11.23)

Wolfe Dual Problem

$$\max \mathcal{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

Subject to $\alpha_i \geq 0$, for $\forall_i, i = 1 \ldots n$ and

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \qquad (11.24)$$

This is called Wolfe dual problem of the optimization problem.

Dual Lagrangian Problem

The dual Lagrangian objective function can be given as Eq. (11.25).

$$\max \mathcal{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

Subject to $\alpha_i \geq 0$, for $\forall_i, i = 1 \ldots n$ and

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \qquad (11.25)$$

Solution of Optimization Problem

From the Lagrangian multipliers, α_i, one can find w and b as follows:

$$w = \sum_{\alpha_i > 0} \alpha_i y_i x_i, \text{ and}$$

$$y_i(wx_i + b) = 1$$

This implies:

$$b = avg_{\alpha_i > 0} \frac{1}{y_i} - w_i x_i \qquad (11.26)$$

Combining w and b, one can get the hyperplane equation as:

$$h(x) = w \cdot x + b \qquad (11.27)$$

The new point x can be predicted as class based on the sign of $h(x)$ in Eq. (11.27) as $\hat{y}_i = sign(h(x))$.

Soft Margin SVM

1. The decision boundary should be far away from the data points, and hence, the distance should be maximized between the line and the nearest data point. In short, this constraint concerns more about the margin and hence deals with distance error.
2. To avoid misclassifications of data, hence, dealing with classification error.

In SVM, the error is the sum of distance error and classification error.

Justification for Soft SVM

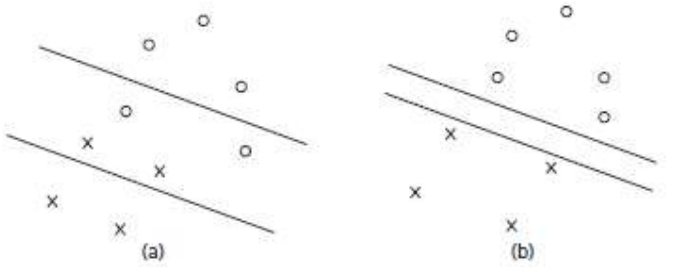


Figure 11.7: (a) Wide Margin but with Misclassification Due to Outliers
(b) Small Margin but with Correct Classification

Soft SVM Problem

The new objective function is given as:

$$\min_{w,b,\xi_i}\left\{\frac{\|w\|^2}{2}+C\sum_{i=1}^{n}(\xi_i)\right\} \tag{11.28}$$

This objective function is subjected to the constraints:

$y_i(w \cdot x_i + b) \geq 1 - \xi_i$, for all the points of x in the dataset and

$$\xi_i \geq 0, \forall x_i \in D$$

Here, ξ represents slack variables, C is the regularization constant, and k is a parameter that determines the error loss functions. It can be observed that the objective function has margin component same as Hard SVM and C times the classification error component. Recollect the fact that errors in SVM are distance error and classification error.

Soft SVM Problem

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

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

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

Subjected to the constraints:

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

Solution of Soft SVM

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 $b = \text{avg}_{\alpha_i > 0}\left(\frac{1}{y_i} - wx_i\right)$.

Kernel

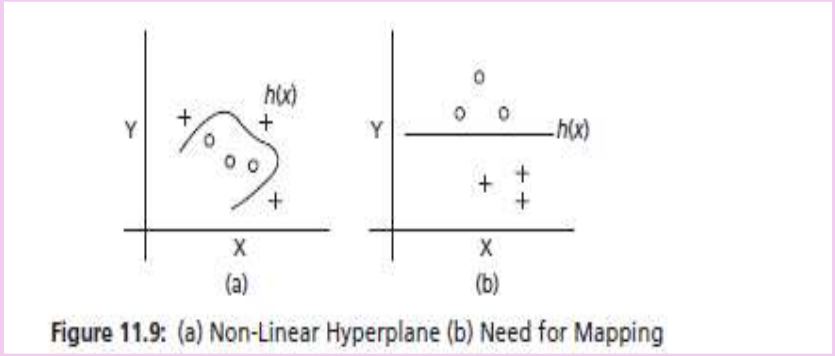


Figure 11.9: (a) Non-Linear Hyperplane (b) Need for Mapping

Mappings

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

$$\varphi(x,y) = (x_1^2, \sqrt{2}xy, y^2) \tag{11.35}$$

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

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

Role of Kernels

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

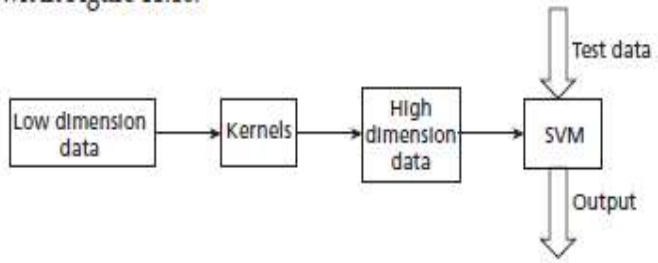


Figure 11.10: Role of Kernels

Types of Kernels

Linear Kernel

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

$$k(x, y) = \phi(x)\,\phi(y) = x^T y \tag{11.36}$$

Polynomial Kernel

Polynomial kernels are of the type:

$$k(x, y) = (x^T \cdot y)^q. \tag{11.37}$$

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:

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

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

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

Gaussian Kernel

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

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

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

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

Sigmoid Kernel

The sigmoid kernel is given as:

$$k(x_i, x_j) = \tanh(kx_i x_j - \delta) \tag{11.40}$$

Kernel Operations

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

$$k(x, x) = x^T x \tag{11.41}$$

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

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

$$k(x, y) = \phi(x) \cdot \phi(y) \tag{11.42}$$

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:

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

Kernel Based Non-Linear Classifier

The objective function of kernel-based SVM can be written now as follows:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

Subjected to the constrains:

$0 \leq \alpha_i \leq C \; \forall_i = 1..n$

And, $\sum_{i=1}^{n} \alpha_i y_i = 0$ (11.44)

Kernel-Based Classifier

The weight w can be derived as discussed earlier in terms of α from KKT conditions as:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{11.45}$$

And the classification rule for a new point z is given as:

$$\begin{aligned} y &= sign(w \cdot z + b) \\ &= sign\left(\sum_{i=1}^{S} \alpha_i y_i x_i z + b\right) \end{aligned} \tag{11.46}$$

Here, S is the set of support vectors, α_i is the Lagrangian multiplier, y_i is the class label, b is the bias and z is the new point to be classified.

The parameter 'b' can be calculated for support vectors $\alpha_i > 0$ and slack variables of zero. These points, say j, are only used to compute b as:

$$\begin{aligned} b &= y_j - wx_j \\ &= y_j - \sum_{i=1}^{n} \alpha_i y_i x_i x_j \end{aligned} \tag{11.47}$$

The decision boundary using the kernel is given as:

$$\hat{y} = sign\left(\sum_{i=1}^{n} \alpha_i y_i k(x_i \cdot z) + \left(y_i - \sum_{i=1}^{n} \alpha_i y_i k(x_i, x_j)\right)\right) \tag{11.48}$$

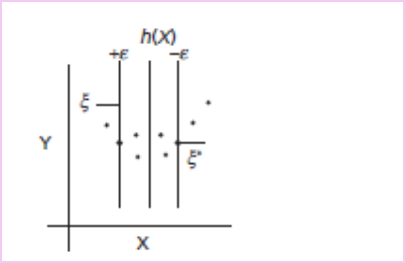
Here, z is the new data and x_i is the support vector found during the training period and x_j is the test vector. y_i is the class label and b is the bias.

Multi-class Problems

Extending Binary Classifier to Multiclass Problems

The binary classifier can be extended for a multi class classifier too if one category is split out and all other categories are merged, and also by using the strategy of one against one, where $k(k-1)/2$ models are constructed. Here, k is the number of categories.

Support Vector Regression



In SVR, ε – insensitive loss function is given as follows:

$$\varepsilon - \text{insensitive loss function} = \begin{cases} 0 \text{ if } |y - f(x)| \leq \varepsilon \\ |y - f(x)| - \varepsilon \text{ otherwise} \end{cases} \quad (11.49)$$

or ε – tube.

Support Vector Regression

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i + \xi_i^*$$

Subjected to the constrains:

$$\begin{aligned} y_i - w_i x_i - b &\le \xi + \xi_i \\ w_i x_i + b - y_i &\le \xi + \xi_i^* \\ \xi_i, \xi_i^* &\ge 0 \end{aligned} \qquad (11.51)$$

$$y = \sum_{i=1}^{s}(\alpha_i - \alpha_i^*)x_i^t x_j + b$$

The non-linear SVR in terms of kernel function is given as:

$$y = \sum_{i=1}^{s}(\alpha_i - \alpha_i^*)K(x_i, x_j) + b \qquad (11.56)$$

Here, s is the set of support vectors, α_i is the Lagrangian multiplier, x_i is the support vector found during the training period, x_j is the test vector and b is the bias.

Relevant Vector Machines

RVM uses a kernel like Gaussian process model as given in Eq. (11.57).

$$k(x, x') = \sum_{j=1}^{n} \frac{1}{\alpha_j} \phi(x, x_j)\phi(x', x_j) \tag{11.57}$$

Here, ϕ is a kernel function, $x_1, x_2, \cdots x_n$ are input vectors of the training set, and α_j are the variances of the prior on the weight vector w. The parameters of the RVM can be initialized using the EM algorithm.

Summary

- SVM is a supervised learning algorithm that uses a labelled data as input and learns learning functions.
- The aim of SVM is to produce a decision plane that defines the boundary between the classes. The focus is to maximize the margins as much as possible.
- The maximal margin problem can be formulated as an optimization problem.
- The convex optimization problem can be solved using Lagrangian multipliers.
- The dual problem can be formulated. Its solution is $\leq$ the solution of the primal problem.
- Soft margin SVM can be used to solve non-linearly separable problems.
- The optimization problem can be solved by introducing slack variables.

Summary

- Binary classification can be extended for multiclass classifiers by including one category and when all other categories are merged. Similarly, the multiclass classifiers can be constructed by constructing $k(k-1)/2$ models, where 'k' represents categories.
- One way of transforming a low-dimensional data to a high-dimensional data is with kernels. Kernels are functions that perform dot product in some other higher dimension.
- Types of kernels are linear kernel, polynomial kernel and Gaussian RBF kernels.
- Kernel trick replaces dot product using kernels.
- Kernel-based classifiers can be constructed for non-linear classifications.
- Support vector regression uses ε – margin to fit a linear or non-linear line for regression problems.