# Support Vector Machines(SVM)

DR. BHARGAVI
PROFESSOR
VIT CHENNAI

# Support Vector Machines - Introduction

- SVMs are very powerful and popularly used Classification and Regression.
- Can be used for Linearly separable data, also used for non-linear data using Kernel trick.
- Maximum Margin Classifier – Hard Margin
- Soft Margin – Support Vector Classifier (SVC)
- Support Vector Machines (SVM) - Kernel
- Logistic Regression and Perceptron picks the linear decision boundary that results in minimum error. But, need not be the ***Best boundary***
- ***Best*** boundary is the one that separates two classes the most.

# Hyperplane

An m-dimensional hyperplane is defined by mathematical equation as follows

$w_0x_0 + w_1x_1 + w_2x_2 + \cdots + w_mx_m = 0$ ------ (Equation1)

Any x = $(x_1, x_2, \ldots, x_m)^T$ satisfying Equation1 is a data point lying on the hyper plane.

Now consider the other equations

$w_0x_0 + w_1x_1 + w_2x_2 + \cdots + w_m > 0$ ------ (Equation2)

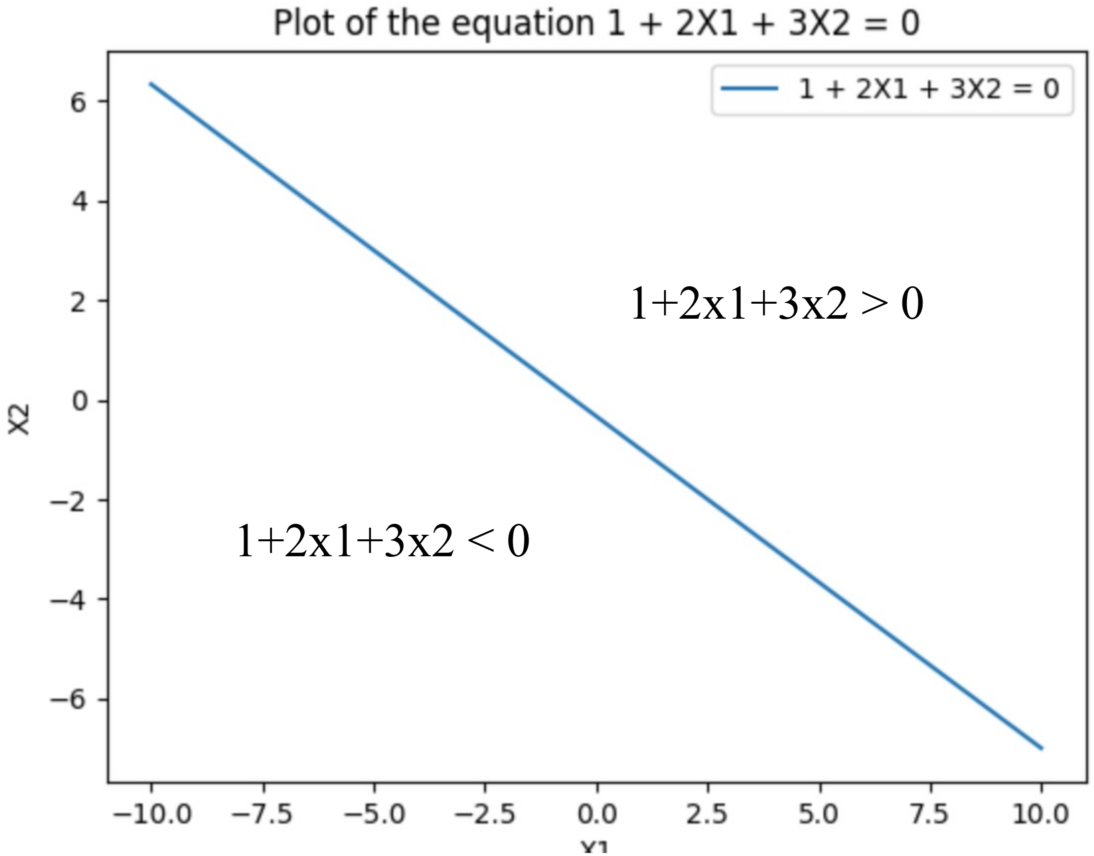
$w_0x_0 + w_1x_1 + w_2x_2 + \cdots + w_m < 0$ ------ (Equation3)

Any x = $(x_1, x_2, \ldots, x_m)^T$ satisfying Equation2 lies on one side of the hyper plane.

Any x = $(x_1, x_2, \ldots, x_m)^T$ satisfying Equation3 lies on the other side of the hyper plane.

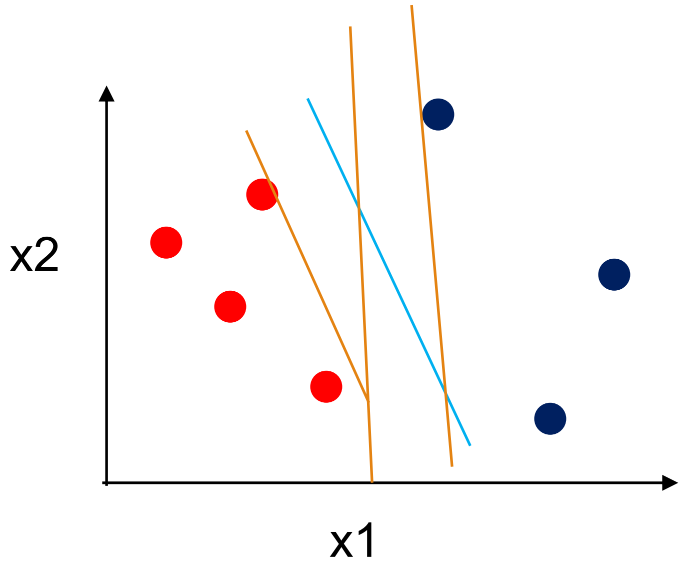
Hence the hyperplane is dividing p-dimensional space into two halves. Sign of the LHS of Eq 1 tells which side of the hyperplane the point is lying.

# Hyperplane (cont…)



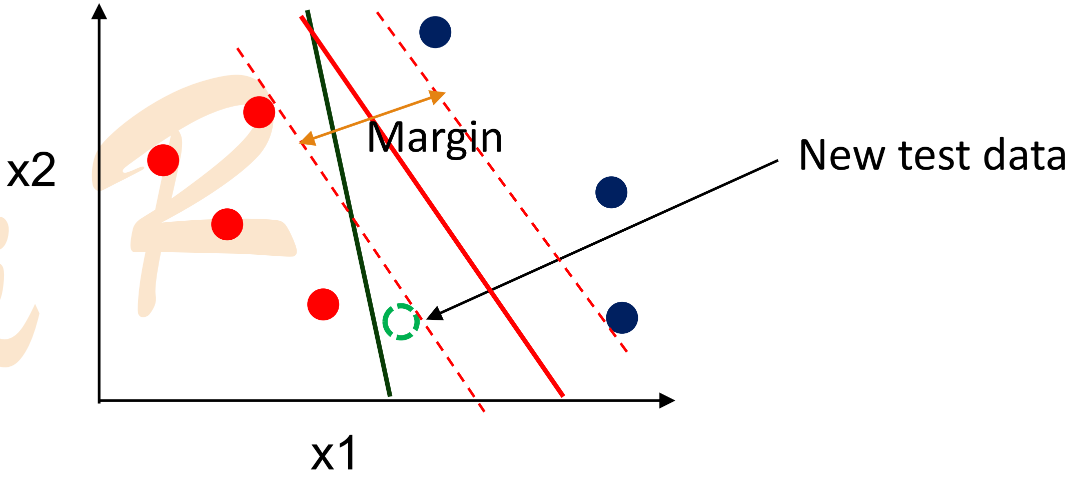
# Linearly separable data

- Consider two dimensional data as shown in the plot.
- x1, x2 are two features.
- Red coloured points belong to one class (or -ve class).
- Blue coloured points belong another class (or +ve class).
- Both the classes can be separated by drawing a straight line (hyper plane in higher dimensions).
- There exists number of separating hyperplanes indicated by orange and blue coloured lines in the figure.
- **Best boundary** is indicated by blue coloured line



# Maximal margin Hyper plane

- A classifier which maximizes the margin is a better classifier.
- The maximal margin hyperplane is the separating hyperplane for which the margin is largest—that is, it is the hyperplane that has the farthest minimum distance to the training observations.



- SVM not only maximizes the margin but also minimizes the prediction error
- In Figure two classes of data are shown - represented with solid dots
- Black coloured solid line is the decision boundary resulted from Logistic Regression or perceptron model.

# Maximal margin Hyper plane

- Consider new test data ( represented as the dotted circle in green colour) to be classified
- As seen from the figure, LR misclassifies the test data as blue class.
- In the figure, maximum margin hyperplane is indicated by red coloured line with margin indicated by the dotted red lines.
- The test data is classified correctly by the hyperplane as it lies to the left side of the hyperplane.
- Hypothesis Function

$$\boxed{y = sign(w^T X + b)}$$

# Hard Margin - Optimization problem

- Maximum margin classifier with hard margin does not allow any misclassifications on the training data.
- All the training data samples lies on either side of the margin or on the boundary of the margin

Margin Equation is $\frac{2}{\|w\|^2}$ (squaring is done for some mathematical convenience)

Here the objective is to maximize the Margin , i.e $\max_{w,b} \frac{2}{\|w\|^2}$

Or equivalently $\min_{w,b} \frac{\|w\|^2}{2}$ ------ (1)

Subject to the constraints : $y_i(\mathbf{wx}_i + b) -1 >= 0$ (zero tolerance)

# Hard margin (cont…)

The objective function is constrained quadratic optimization problem.

solve this using method of Legrange multipliers.

Introduce the Lagrangian function $L_p$ as follows

$L_p = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i(y_i(wX_i + b) - 1)$ such that $\alpha_i \geq 0$ for i = 1,2 ,,,, , n

The problem can now be rewritten as

$\min_{w,b}[\frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \max_{\alpha_i} \alpha_i(y_i(wX_i + b) - 1)\ ]$ such that $\alpha_i \geq 0$

$\min_{w,b} \max_{\alpha} \ \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i(y_i(wX_i + b) - 1)$ such that $\alpha_i \geq 0$ (primal problem)

# Hard margin (cont…)

Dual of the primal problem is as follows

$\max_{\alpha} \min_{w,b} \ \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i (wX_i \ + b) - 1)$ such that $\alpha_i \geq 0$

Or $\max_{\alpha} \left( \min_{w,b} \ L(w, b, \alpha) \right)$ such that $\alpha_i \geq 0$

Since it is an optimization problem we can solve it by setting the derivatives w.r.t w and b equal to 0 i.e

$\frac{\partial}{\partial w} L(w, b, \alpha) = 0$

$\frac{\partial}{\partial b} L(w, b, \alpha) = 0$

# Hard margin (cont…)

Solving

$$\frac{\partial}{\partial w} L(w, b, \alpha) = w - \sum_{i=1}^{n} \alpha_i y_i \, x_i \ = 0$$

$$w = \sum_{i=1}^{n} \alpha_i y_i \, \boldsymbol{x}_i$$

$$\frac{\partial}{\partial b} L(w, b, \alpha) = \ - \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

Substituting the value of $W$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$ in L we get

$$\max_{\alpha_i} \left[ \sum \alpha_i \ - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left( \boldsymbol{x}_i^T . \boldsymbol{x}_j \right) \right] \text{ such that } \ \alpha_i \ \geq 0$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

# Hard margin (cont…)

(Note - Method of Lagrange multipliers is used for solving problems with equality constraints, and here we are using them with inequality constraints. Hence, there is an additional requirement that the solution must also satisfy the Karush-Kuhn-Tucker (KKT) conditions. if a solution satisfies the KKT conditions, we are guaranteed that it is the optimal solution.)

Karush-Kuhn-Tucker (KKT) conditions:

**Stationarity condition:**

$$\frac{\partial}{\partial w} L(w, b, \alpha) = w - \sum_{i=1}^{n} \alpha_i y_i \, x_i = 0$$

$$\frac{\partial}{\partial b} L(w, b, \alpha) = - \sum_{i=1}^{n} \alpha_i y_i = 0$$

**Primal feasibility condition:** $y_i(\boldsymbol{w}\boldsymbol{x}_i + b) - 1 \geq 0 \; for\ all\ i = 1,2, \ldots\ldots, n$

**Dual feasibility condition:** $\alpha_i \geq 0 \; for\ all\ i = 1,2, \ldots\ldots, n$

# Hard margin (cont…)

**Complementary slackness condition:**

$\alpha_i(y_i(\boldsymbol{wx}_i + b) - 1) = 0 \; for \; all \; i = 1,2, \ldots \ldots, n$

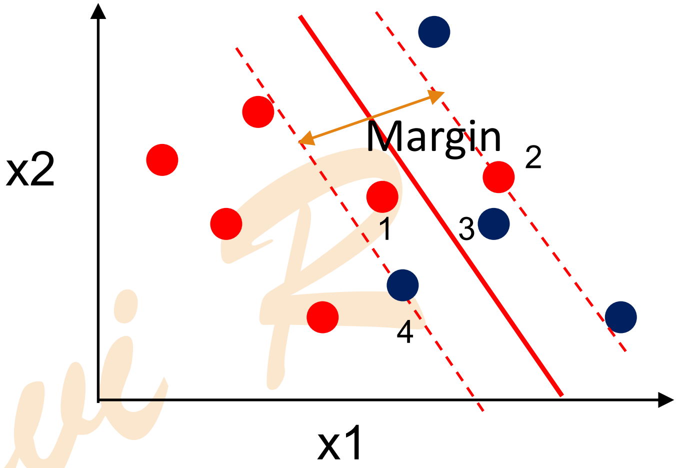
From this condition either $\alpha_i = 0$ or $y_i(wX_i + b) - 1 = 0$.

Examples/instances having a positive Lagrange multiplier are called as the Support Vectors.

$b = \frac{1}{s}\sum_{i=1}^{s}(y_i - W.X_i)$ (there are other alternative ways of computing b)

# Linearly non-separable data - Soft Margin

- Most of the datasets in practical real applications are not linearly separable.
- “zero training loss” is practically impossible.
- Solution – Soft margin with misclassifications by allowing some margin violations



- Classifier with soft margin is called as Support Vector Classifier.
- With Soft margin two kinds of violations are allowed.
- Margin violation without violating the decision boundary (Data points 1, 3 in the figure)
- Decision boundary violation (Data points 2, 4).

# Soft Margin – Objective Function

- The optimization problem now becomes

$$\min_{w,b} \frac{\|w\|^2}{2} + C \sum_{i=1}^{n} \varepsilon_i \quad \text{s.t: } y_i(w^T x + b) >= 1 - \varepsilon_i \text{ , } \varepsilon_i >= 0$$

- $C \sum_{i=1}^{n} \varepsilon_i$ - Regularization term
- $\varepsilon_i$ - slack variable associated with i<sup>th</sup> observation indicating the amount of violation
- $\varepsilon_i = 0$ : Points on the correct side of the margin
- $0 < \varepsilon_i <= 1$ : Points inside the margin but on the correct side
- $\varepsilon_i > 1$ : Points on the wrong side of the hyperplane.
- Solution to this Objective function models the Support Vector classifier.

# Soft Margin – Objective Function (cont…)

**Low C value (Wider margin)**: The model will prioritize maximizing the margin, more tolerant misclassifications.

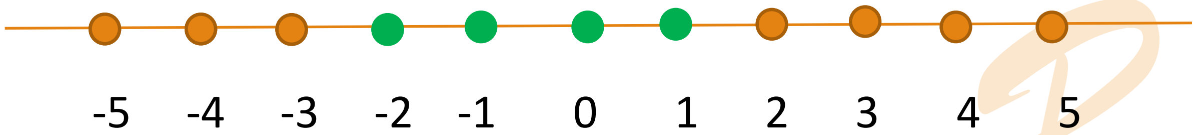
This is useful when the data is noisy or there are overlapping classes. A small C leads to a wider margin, which can lead to a simpler and more generalizable model reducing the risk of overfitting.

**High C value (Narrow margin)**:When **C is large**, the penalty for misclassification is high.

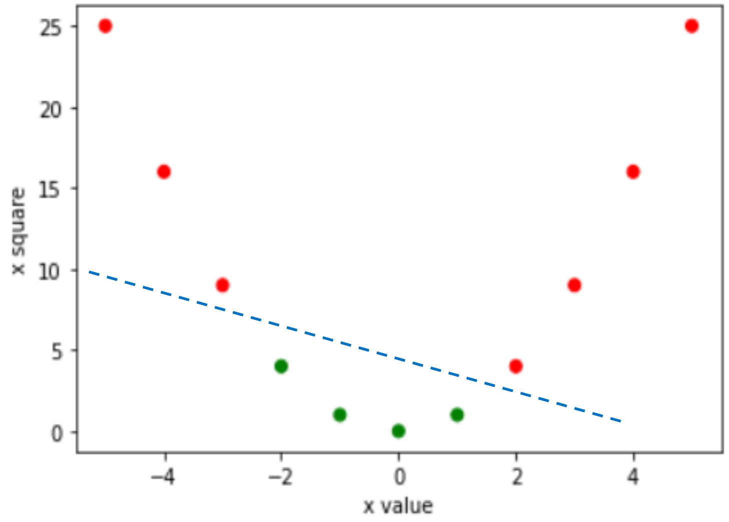
The model is forced to be much more strict, trying to classify as many training points correctly as possible. This results in a **narrower margin** and a more complex model that is very sensitive to the training data. This can lead to a lower training error but an increased risk of overfitting, where the model performs poorly on new, unseen data. The model has a lower bias and higher variance.

# Kernel Functions – Non-linear Decision Boundary

- Consider the one dimensional data (feature x ) shown below which is not linearly separable



- Map x to $\emptyset(x)$ where $\emptyset(x) = x^2$.
- Now this high dimensional transformed data becomes linearly separable as shown in the figure



# Need for Kernel

- Original feature space which is not linearly separable may become linearly separable by extending the feature space
- Feature space extension
  - Higher-degree polynomials of features
  - Some functions of features
  - Interaction terms of the form $X_jX_j{'}$ for $j \neq j'$ etc.
- Feature extension results in huge memory and computational requirements.
- Use of kernel function reduces the memory and computational requirements.

# Objective function

The linear support vector classifier problem's objective function is

$$\max_{\alpha_i}\left[\sum \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \langle \boldsymbol{x}_i^T . \boldsymbol{x}_j \rangle\right] \text{ such that } \alpha_i \geq 0$$

Dot product. Can be replaced with kernel

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

Define a kernel as: $k(x_{i,} x_j) = \langle x_i^T . x_j \rangle$ , Objective function now becomes

$$\max_{\alpha_i}\left[\sum \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(x_{i,} x_j)\right] \text{ such that } \alpha_i \geq 0$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

# Kernel Trick - Example

Let f(x) = (x1x1, x1x2, x1x3, x2x1, x2x2, x2x3, x3x1, x3x2, x3x3), the kernel is K(x, y ) = (<x, y>)$^2$.

Consider two vectors x = (x1, x2, x3); y = (y1, y2, y3)

Let x = (1, 2, 3); y = (4, 5, 6)

Now we want extend the feature space as in f(x)

So, f(x) = (1, 2, 3, 2, 4, 6, 3, 6, 9)

f(y) = (16, 20, 24, 20, 25, 30, 24, 30, 36)

and <f(x), f(y)> = 16 + 40 + 72 + 40 + 100+ 180 + 72 + 180 + 324 = 1024

Observation - Many computations

# Kernel Trick - Example

$K(x, y) = (<x, y>)^2 = (4 + 10 + 18)^2 = 32^2 = 1024$

Here $K(x, y) = (<x, y>)^2$ is called as Quadratic Kernel

# Kernel functions - Examples

**Linear Kernel**: This is the simplest kernel and is equivalent to using a linear SVM. The kernel function is simply the dot product of two vectors.

$$k(x, z) = x^T z$$

**Polynomial Kernel**: This kernel allows us to create non-linear decision boundaries by using polynomial functions of the input features. It computes the dot product of two vectors raised to a power d (degree of the polynomial)

$$k(x, z) = (x^T z + c)^d$$

The parameter *d* controls the degree of the polynomial, and *c* is a constant. Higher values of d lead to more complex decision boundaries.

# Kernel functions – Examples (cont…)

Radial Basis Function (RBF) Kernel / Gaussian Kernel: This is the most commonly used kernel in practice. It maps the data into an infinite-dimensional space and is especially useful for separating complex datasets

$$k(x, z) = \exp(-\frac{\|x - z\|^2}{2\sigma^2})$$

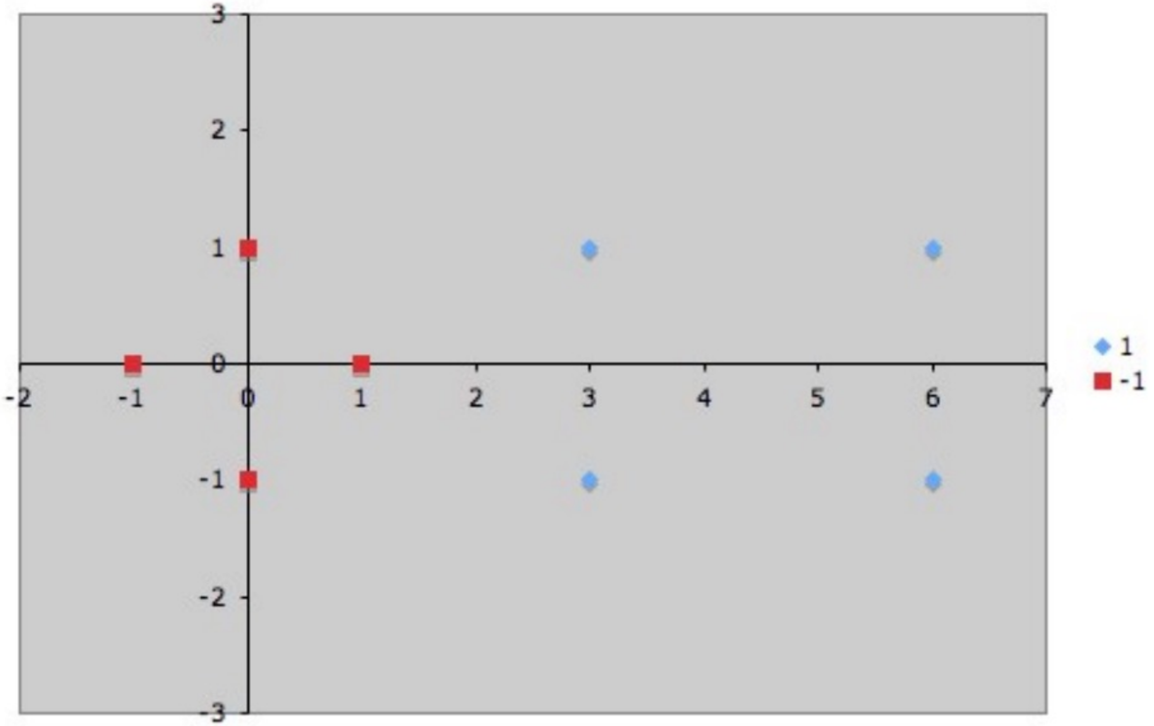
Sigmoid Kernel: The sigmoid kernel is related to neural networks and is defined as:

$$k(x, z) = \tanh(\alpha x^T z + c)$$

# Example – Linearly Separable

Consider the following 2-dimensional dataset

| Feature1 | Feature2 | Class |
|---|---|---|
| 3 | 1 | + |
| 3 | -1 | + |
| 6 | 1 | + |
| 6 | -1 | + |
| 1 | 0 | - |
| 0 | 1 | - |
| -1 | 0 | - |
| 0 | -1 | - |



# Example (cont…)

From the given data the support vectors are

$$\left\{ s_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, s_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, s_3 = \begin{pmatrix} 3 \\ -1 \end{pmatrix} \right\}$$

Augment the bias input 1 to the input vectors ((1,0) becomes (1,0,1) etc,.)

Introducing the mapping function $\Phi$ ( Here Identity function).

Our task is to find the $\alpha_{\mathrm{I}}$'s such that

$$\begin{aligned}
\alpha_1\Phi(s_1)\cdot\Phi(s_1)+\alpha_2\Phi(s_2)\cdot\Phi(s_1)+\alpha_3\Phi(s_3)\cdot\Phi(s_1) &= -1 \\
\alpha_1\Phi(s_1)\cdot\Phi(s_2)+\alpha_2\Phi(s_2)\cdot\Phi(s_2)+\alpha_3\Phi(s_3)\cdot\Phi(s_2) &= +1 \\
\alpha_1\Phi(s_1)\cdot\Phi(s_3)+\alpha_2\Phi(s_2)\cdot\Phi(s_3)+\alpha_3\Phi(s_3)\cdot\Phi(s_3) &= +1
\end{aligned}$$

# Example (cont…)

Since mapping function $\Phi$ Identity function (I).

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

For the augmented support vectors the above equations becomes

$$\begin{aligned} 2\alpha_1 + 4\alpha_2 + 4\alpha_3 &= -1 \\ 4\alpha_1 + 11\alpha_2 + 9\alpha_3 &= +1 \\ 4\alpha_1 + 9\alpha_2 + 11\alpha_3 &= +1 \end{aligned}$$

On solving the above equations we get $\alpha_1$= -3.5, $\alpha_2$= 0.75, $\alpha_3$= 0.75

# Example (cont…)

Now, We can find the Hyperplane defined by $\mathbf{w}_i$

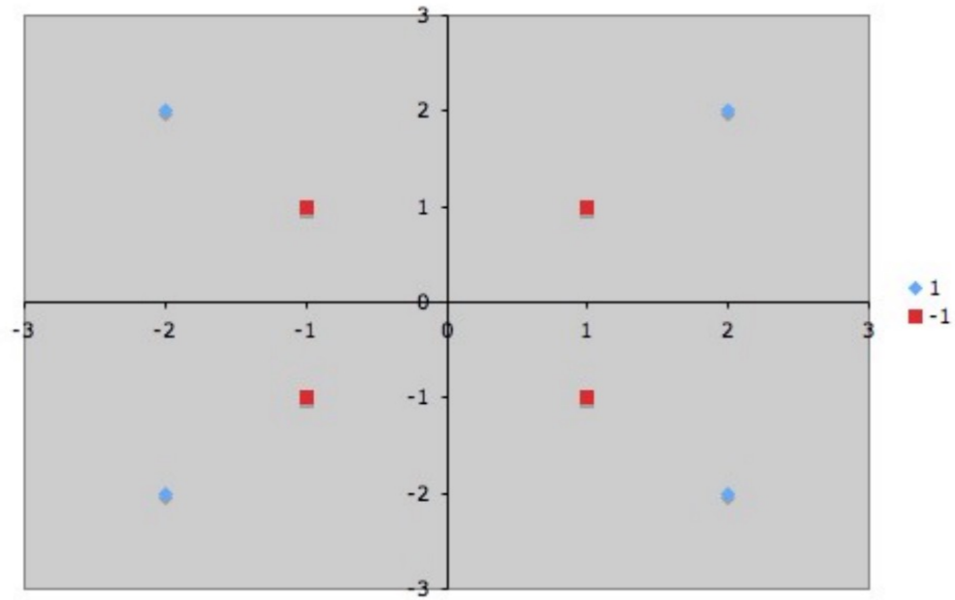
$$\begin{aligned}
\tilde{w} &= \sum_i \alpha_i \tilde{s}_i \\
&= -3.5 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}
\end{aligned}$$

$$w = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } b = -2.$$

# Example – Linearly non-separable

Consider the following data : +ve samples $\left\{ \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \end{pmatrix} \right\}$

-ve samples $\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\}$



# Example (cont…)

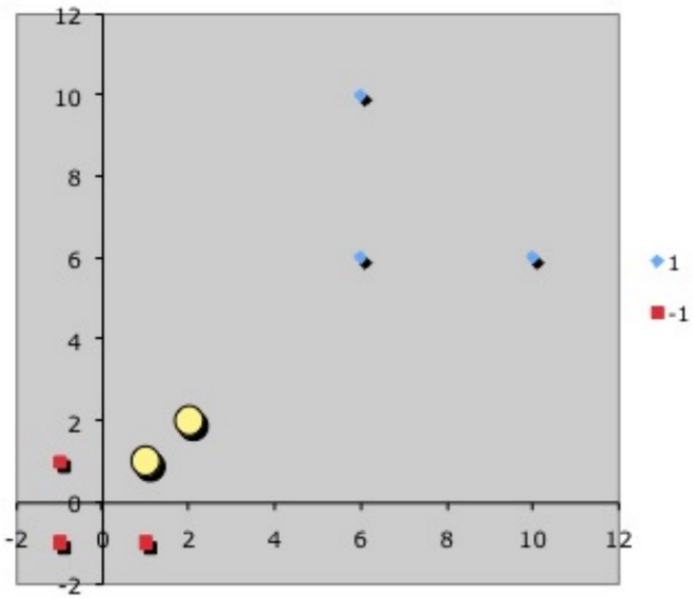
Now, we use feature transformation

$$\Phi_1\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{cases}\begin{pmatrix}4-x_2+|x_1-x_2|\\4-x_1+|x_1-x_2|\end{pmatrix} & \text{if } \sqrt{x_1^2+x_2^2}>2\\ \begin{pmatrix}x_1\\x_2\end{pmatrix} & \text{otherwise}\end{cases}$$

Transformed +ve samples $\left\{\begin{pmatrix}2\\2\end{pmatrix},\begin{pmatrix}10\\6\end{pmatrix},\begin{pmatrix}6\\6\end{pmatrix},\begin{pmatrix}6\\10\end{pmatrix}\right\}$

Transformed -ve samples

$$\left\{\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}1\\-1\end{pmatrix},\begin{pmatrix}-1\\-1\end{pmatrix},\begin{pmatrix}-1\\1\end{pmatrix}\right\}$$


12
10
8
6
4
2
0
-2
-2
0
2
4
6
8
10
12
◆ 1
■ -1