MACHINE LEARNING

UNIT-3

LINEAR MODELS

Topics

- **The least-squares method**
  - Univariate/Simple Linear Regression
  - Multivariate/Multiple Linear Regression
- **The Perceptron**
- **Support Vector Machines**
- **Obtaining probabilities from linear classifiers**

Linear Models

- We can use geometric concepts such as lines and planes to impose structure on the space, for instance in order to build a classification model.

- Models that can be understood in terms of lines and planes, commonly called *linear models.*

Characteristics of Linear Models

- Linear models are *parametric, meaning that they have a fixed form with a small* number of numeric parameters that need to be learned from data. This is different from tree or rule models, where the structure of the model (e.g., which features to use in the tree, and where) is not fixed in advance.

- Linear models are stable, which is to say that small variations in the training data have only limited impact on the learned model. Tree models tend to vary more with the training data, as the choice of a different split at the root of the tree typically means that the rest of the tree is different as well.

- Linear models are less likely to overfit the training data than some other models, largely because they have relatively few parameters. The flipside of this is that they sometimes lead to *underfitting*: *e.g., imagine you are learning where the* border runs between two countries from labelled samples, then a linear model is unlikely to give a good approximation.

Linear Models

- The last two points can be summarized by saying that linear models have low variance but high bias.
- Linear models are often preferable when you have limited data and want to avoid overfitting.
- High variance–low bias models such as decision trees are preferable if data is abundant.
- It is usually a good idea to start with simple, high-bias models such as linear models and only move on to more elaborate models if the simpler ones appear to be under-fitting.

Linear Models

- Linear models exist for all predictive tasks, including classification, probability estimation and regression.

- Linear regression, in particular, is a well-studied problem that can be solved by the least-squares method

Least-Squares Method

- Recall that the regression problem is to learn a function estimator ˆ *f :X →R from examples (xi , f (xi))* .

- *The differences* between the actual and estimated function values on the training examples are called *residuals i = f (xi)− ˆ f (xi).*

- *The least-squares method, introduced by Carl* Friedrich Gauss, consists in finding ˆ *f such that* $\sum_{i=1}^{n} \epsilon_i^2$ *is* minimised.

- In case of a single feature, the regression is called *univariate regression.*

UNIVARIATE REGRESSION.

Example 7.1 (Univariate linear regression). Suppose we want to investigate the relationship between people's height and weight. We collect n height and weight measurements $(h_i, w_i), 1 \leq i \leq n$. Univariate linear regression assumes a linear equation $w = a + bh$, with parameters a and b chosen such that the sum of squared residuals $\sum_{i=1}^{n}(w_i - (a + bh_i))^2$ is minimised. In order to find the parameters we take partial derivatives of this expression, set the partial derivatives to 0 and solve for a and b:

$$\frac{\partial}{\partial a}\sum_{i=1}^{n}(w_i - (a + bh_i))^2 = -2\sum_{i=1}^{n}(w_i - (a + bh_i)) = 0 \qquad \Rightarrow \hat{a} = \overline{w} - \hat{b}\overline{h}$$

$$\frac{\partial}{\partial b}\sum_{i=1}^{n}(w_i - (a + bh_i))^2 = -2\sum_{i=1}^{n}(w_i - (a + bh_i))h_i = 0$$

$$\Rightarrow \hat{b} = \frac{\sum_{i=1}^{n}(h_i - \overline{h})(w_i - \overline{w})}{\sum_{i=1}^{n}(h_i - \overline{h})^2}$$

UNIVARIATE REGRESSION.

- In other words, univariate linear regression can be understood as consisting of two steps:

1. normalization of the feature by dividing its values by the feature's variance;

2. calculating the covariance of the target variable and the normalized feature

Example for UNIVARIATE REGRESSION.

Feature	Target
"h"	"w"
height	Weight
2	4
3	5
5	7
7	10
9	15

Univariate Linear Regression Equation:

w = a + bh

$$\Rightarrow \hat{b} = \frac{\sum_{i=1}^{n}(h_i - \overline{h})(w_i - \overline{w})}{\sum_{i=1}^{n}(h_i - \overline{h})^2}$$

$$\Rightarrow \hat{a} = \overline{w} - \hat{b}\overline{h}$$

Example for UNIVARIATE REGRESSION.

	Feature	Target				
	"h"	"w"				
	height	Weight	h-mean(h)	w-mean(w)	(h-mean(h))* (w-mean(w))	(h-mean(h))^2
	2	4	-3.2	-4.2	13.44	10.24
	3	5	-2.2	-3.2	7.04	4.84
	5	7	-0.2	-1.2	0.24	0.04
	7	10	1.8	1.8	3.24	3.24
	9	15	3.8	6.8	25.84	14.44
Mean	5.2	8.2		SUM	49.8	32.8

Example for UNIVARIATE REGRESSION.

	Feature	Target
	"h"	"w"
	height	Weight
	2	4
	3	5
	5	7
	7	10
	9	15

Univariate Linear Regression Equation:

$$w = a + bh$$

$$\Rightarrow \hat{b} = \frac{\sum_{i=1}^{n}(h_i - \overline{h})(w_i - \overline{w})}{\sum_{i=1}^{n}(h_i - \overline{h})^2}$$

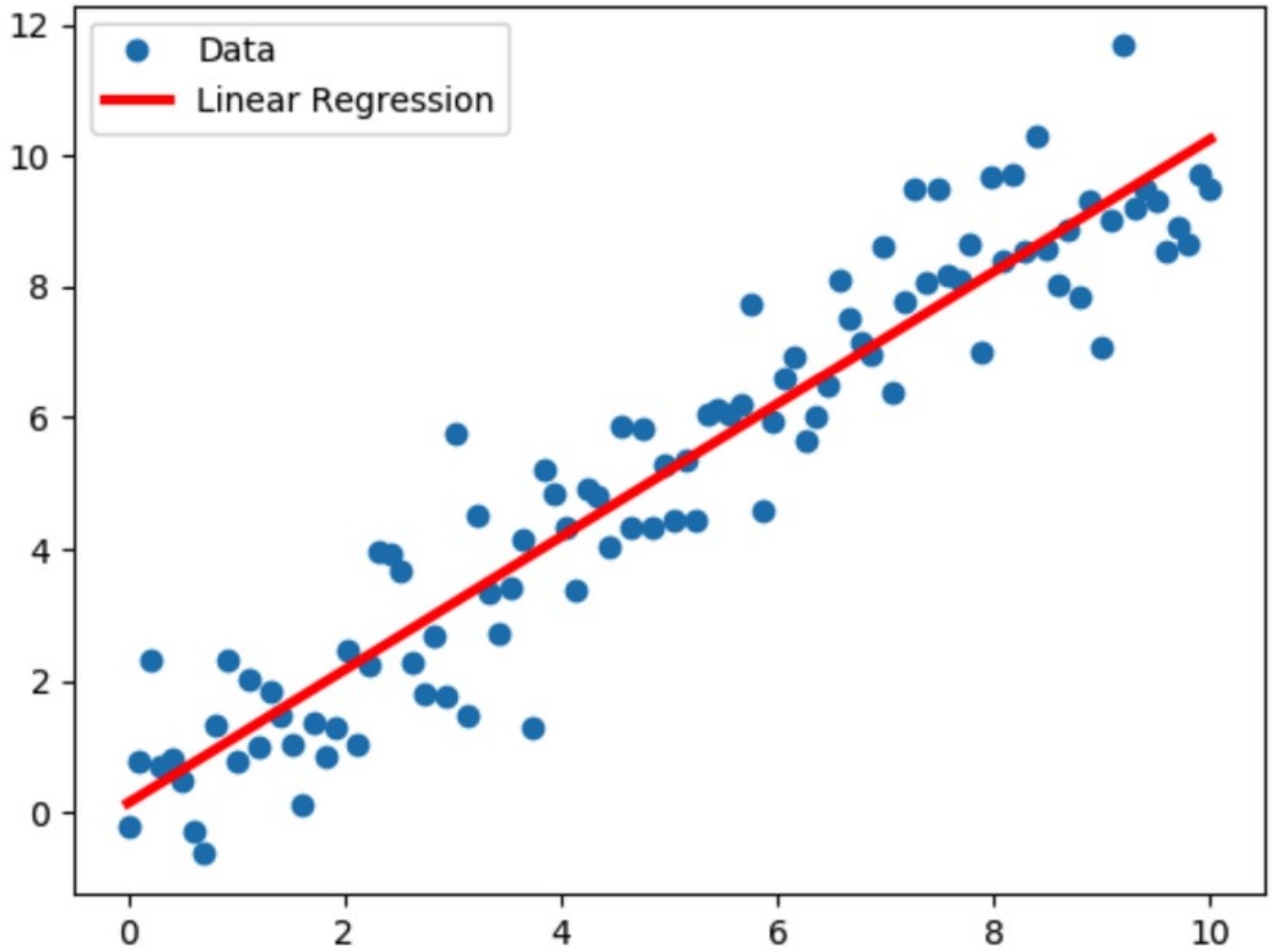
b = 49.8/32.8 = 1.52

$$\Rightarrow \hat{a} = \overline{w} - \hat{b}\overline{h}$$

a = 8.2 – 1.52 * 5.2 = 0.3

Univariate Linear Regression Equation: ***w = 0.3 + 1.52 * h***

Example for UNIVARIATE REGRESSION.



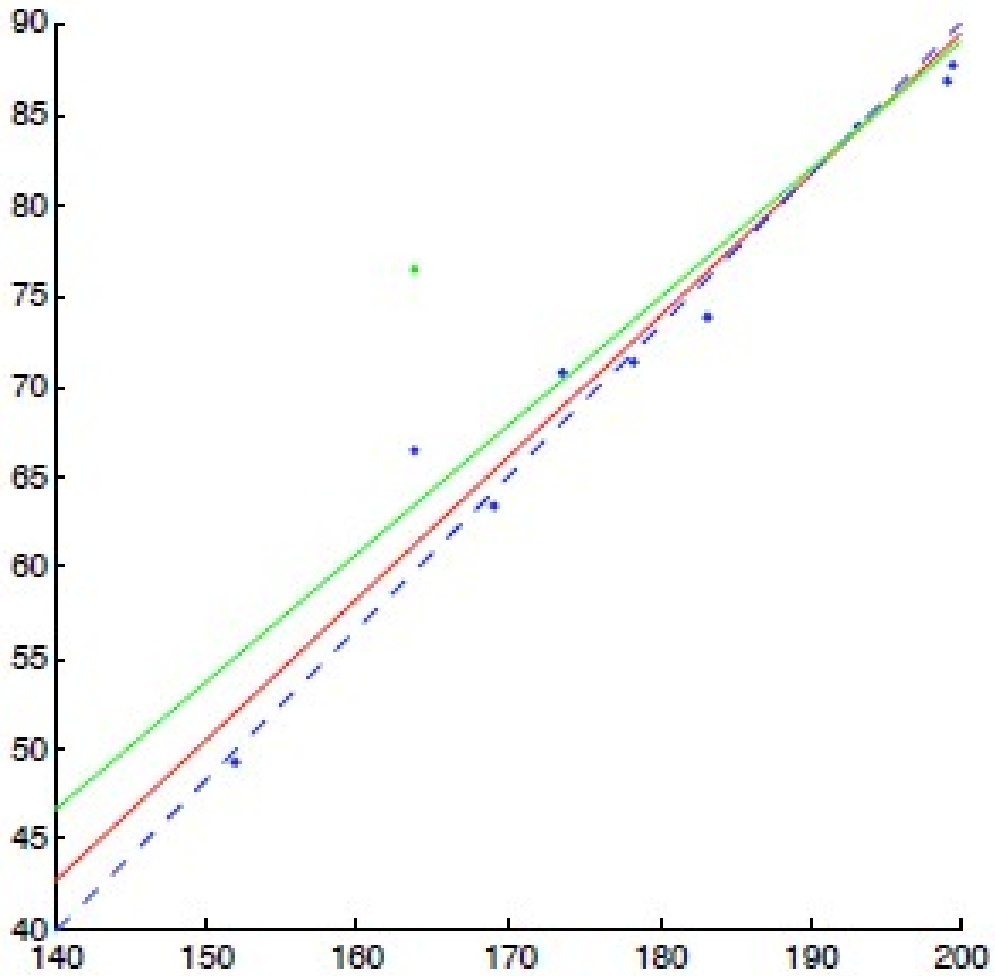
Linear regression

- linear regression is susceptible to *outliers:* *points that are far moved from the regression line, often* because of measurement errors.

- Suppose that, as the result of a transcription error, one of the weight values in is increased by 10 kg.

- A considerable effect on the least-squares regression line will be visible.

Effect of Outliers on linear regression



Multivariate linear regression

This is quite similar to the simple linear regression model we have discussed previously, but with multiple independent variables contributing to the dependent variable and hence multiple coefficients to determine and complex computation due to the added variables. Jumping straight into the equation of multivariate linear regression,

$$Y_i = \alpha + \beta_1 x_i^{(1)} + \beta_2 x_i^{(2)} + \ldots + \beta_n x_i^{(n)}$$

Y_i is the estimate of i^{th} component of dependent variable y, where we have n independent variables and x_i^j denotes the i^{th} component of the j^{th} independent variable/feature. Similarly cost function is as follows,

$$E(\alpha, \beta_1, \beta_2, \ldots, \beta_n) = \frac{1}{2m} \sum_{i=1}^{m} (y_i - Y_i)$$

MLR with 2 independent variables

The estimated linear regression equation is: $\hat{y} = b_0 + b_1 * x_1 + b_2 * x_2$

The formula to calculate b_0 is: $\overline{y} - b_1\overline{x}_1 - b_2\overline{x}_2$

$$b_1 = \frac{(\sum x_2^2)(\sum x_1 y) - (\sum x_1 x_2)(\sum x_2 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)^2}$$

$$b_2 = \frac{(\sum x_1^2)(\sum x_2 y) - (\sum x_1 x_2)(\sum x_1 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)^2}$$

- x_1 and x_2 variance from actual values. We need to calculate them using X1 and X2 (independent variables) values in the given data.

$$\sum x_1^2 = \sum X_1 X_1 - \frac{(\sum X_1)(\sum X_1)}{N}$$

$$\sum x_2^2 = \sum X_2 X_2 - \frac{(\sum X_2)(\sum X_2)}{N}$$

$$\sum x_1 y = \sum X_1 Y - \frac{(\sum X_1)(\sum Y)}{N}$$

$$\sum x_2 y = \sum X_2 Y - \frac{(\sum X_2)(\sum Y)}{N}$$

$$\sum x_1 x_2 = \sum X_1 X_2 - \frac{(\sum X_1)(\sum X_2)}{N}$$

Example for Multivariate linear regression

Suppose we have the following dataset with one response variable *y* and two predictor variables X_1 and X_2:

y	X_1	X_2
140	60	22
155	62	25
159	67	24
179	70	20
192	71	15
200	72	14
212	75	14
215	78	11

Example for Multivariate linear regression

Step 1: Calculate X_1^2, X_2^2, X_1y, X_2y and X_1X_2.

	y	X_1	X_2
	140	60	22
	155	62	25
	159	67	24
	179	70	20
	192	71	15
	200	72	14
	212	75	14
	215	78	11
Mean	181.5	69.375	18.125
Sum	1452	555	145

X_1^2	X_2^2	X_1y	X_2y	X_1X_2
3600	484	8400	3080	1320
3844	625	9610	3875	1550
4489	576	10653	3816	1608
4900	400	12530	3580	1400
5041	225	13632	2880	1065
5184	196	14400	2800	1008
5625	196	15900	2968	1050
6084	121	16770	2365	858
Sum 38767	2823	101895	25364	9859

Example for Multivariate linear regression

Step 2: Calculate Regression Sums.

Next, make the following regression sum calculations:

- $\Sigma x_1^2 = \Sigma X_1^2 - (\Sigma X_1)^2 / n = 38{,}767 - (555)^2 / 8 =$ **263.875**
- $\Sigma x_2^2 = \Sigma X_2^2 - (\Sigma X_2)^2 / n = 2{,}823 - (145)^2 / 8 =$ **194.875**
- $\Sigma x_1 y = \Sigma X_1 y - (\Sigma X_1 \Sigma y) / n = 101{,}895 - (555*1{,}452) / 8 =$ **1,162.5**
- $\Sigma x_2 y = \Sigma X_2 y - (\Sigma X_2 \Sigma y) / n = 25{,}364 - (145*1{,}452) / 8 =$ **-953.5**
- $\Sigma x_1 x_2 = \Sigma X_1 X_2 - (\Sigma X_1 \Sigma X_2) / n = 9{,}859 - (555*145) / 8 =$ **-200.375**

Example for Multivariate linear regression

Step 3: Calculate b_0, b_1, and b_2.

The formula to calculate b_1 is: $[(\Sigma x_2^2)(\Sigma x_1 y) - (\Sigma x_1 x_2)(\Sigma x_2 y)] / [(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2]$

Thus, $\mathbf{b_1} = [(194.875)(1162.5) - (-200.375)(-953.5)] / [(263.875)(194.875) - (-200.375)^2] = \mathbf{3.148}$

The formula to calculate b_2 is: $[(\Sigma x_1^2)(\Sigma x_2 y) - (\Sigma x_1 x_2)(\Sigma x_1 y)] / [(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2]$

Thus, $\mathbf{b_2} = [(263.875)(-953.5) - (-200.375)(1152.5)] / [(263.875)(194.875) - (-200.375)^2] = \mathbf{-1.656}$

The formula to calculate b_0 is: $\overline{y} - b_1\overline{X}_1 - b_2\overline{X}_2$

Thus, $\mathbf{b_0} = 181.5 - 3.148(69.375) - (-1.656)(18.125) = \mathbf{-6.867}$

Example for Multivariate linear regression

Step 4: Place b_0, b_1, and b_2 in the estimated linear regression equation.

The estimated linear regression equation is: $\hat{y} = b_0 + b_1 * x_1 + b_2 * x_2$

In our example, it is $\hat{y} = -6.867 + 3.148x_1 - 1.656x_2$

Multivariate linear regression

Now let us talk in terms of matrices as it is easier that way. As discussed before, if we have n independent variables in our training data, our matrix X has $n+1$ rows, where the first row is the 0^{th} term added to each vector of independent variables which has a value of 1 (this is the coefficient of the constant term α). So, X is as follows,

$$X = \begin{bmatrix} X_1 \\ .. \\ X_m \end{bmatrix}$$

X^i contains n entries corresponding to each feature in training data of i^{th} entry. So, matrix X has m rows and $n+1$ columns ($0^{th} column$ is all 1^s and rest for one independent variable each).

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ .. \\ Y_m \end{bmatrix}$$

and coefficient matrix C,

$$C = \begin{bmatrix} \alpha \\ \beta_1 \\ .. \\ \beta_n \end{bmatrix}$$

and our final equation for our hypothesis is,

$$Y = XC$$

Multivariate linear regression

$$C = (X^T X)^{-1} X^T y$$

$(\mathbf{X^T X})^{-1}$ ***acts*** *as a transformation that decorrelates, centres and normalises the features.*

uncorrelated features effectively decomposes a multivariate regression problem into d univariate problems.

Topics

- **The least-squares method**
  - Univariate Linear Regression
  - Multivariate linear regression

- **Support vector machines**

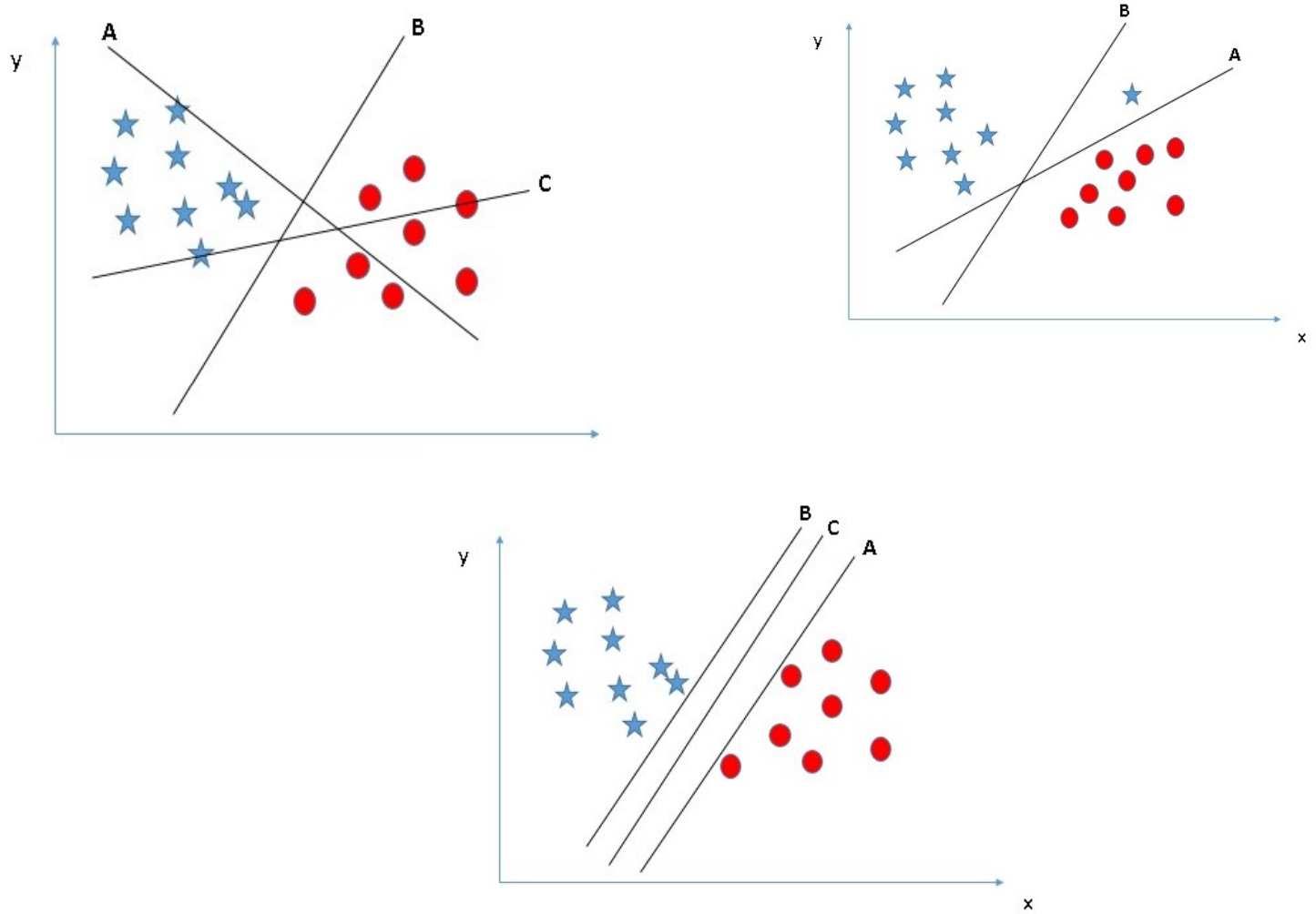
Support Vector Machine

- "Support Vector Machine" (SVM) is a supervised learning machine learning algorithm that can be used for both classification or regression challenges.

- In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have), with the value of each feature being the value of a particular coordinate.
- Then, we perform classification by finding the optimal hyper-plane that differentiates the two classes very well.

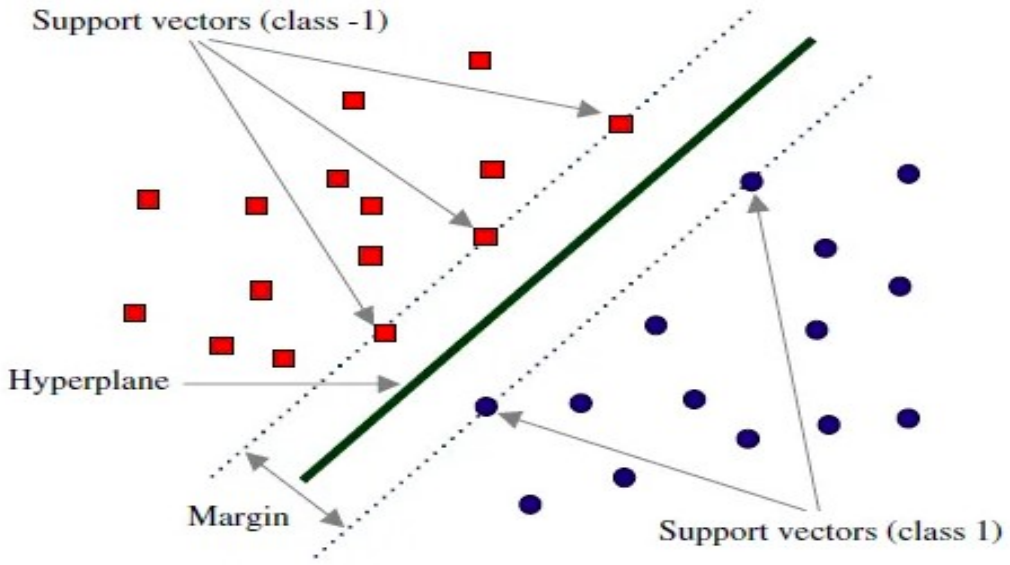
- **Hyper-plane:** It is plane that linearly divide the n-dimensional data points in two component. In case of 2D, hyperplane is line, in case of 3D it is plane.It is also called as *n-dimensional line.*
- ***optimal hyperplane is one which divides the data points very well***

How to choose a hyperplane

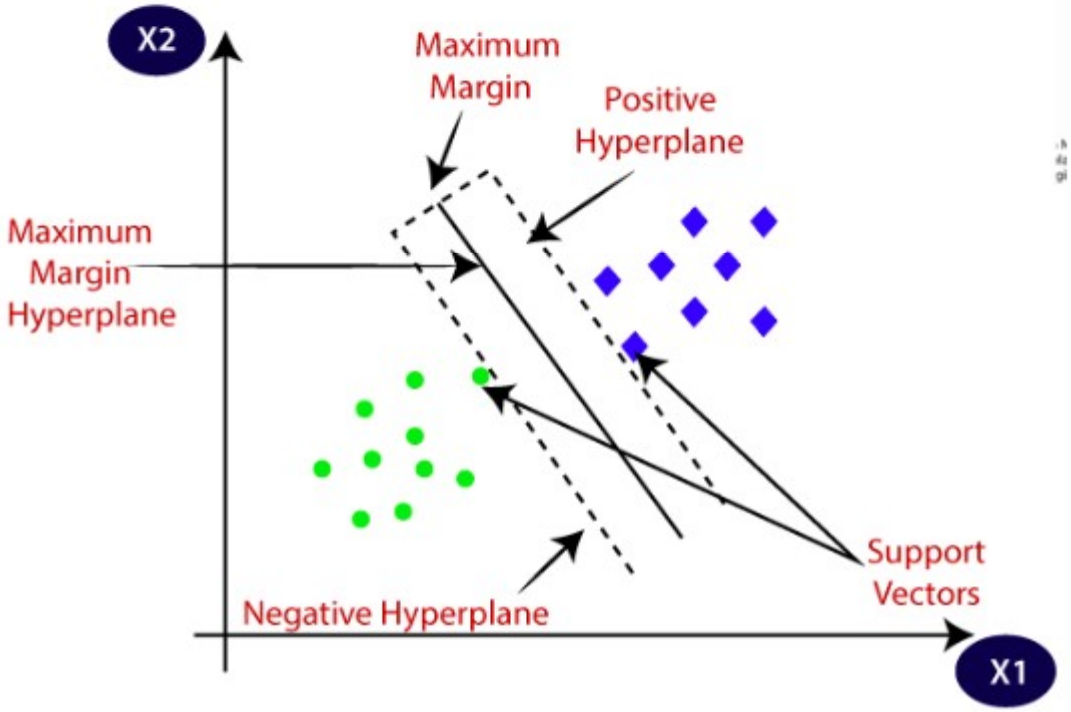


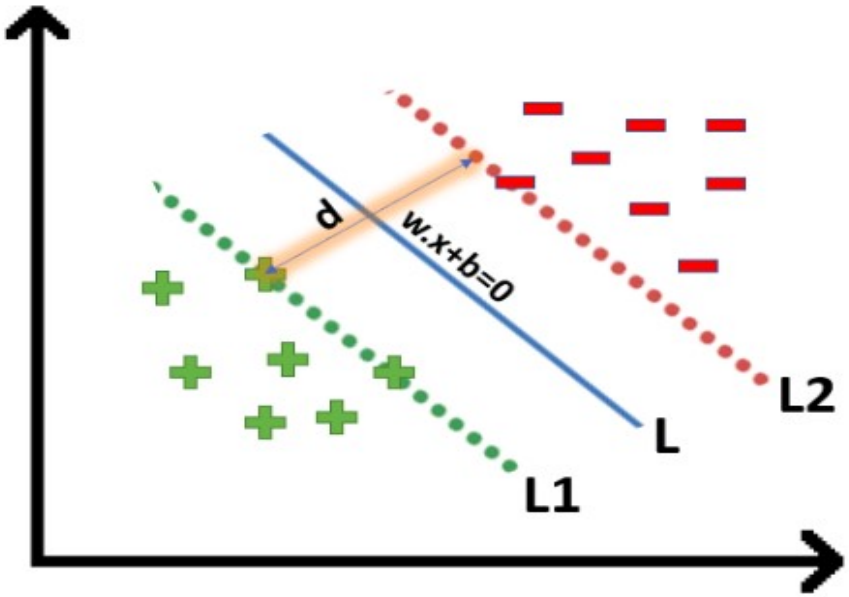
Margin and Support Vectors

- Let's assume that solid black line is optimal hyperplane and two dotted line is some hyperplane, which is passing through nearest data points to the optimal hyperplane.
- Then distance between hyperplane and optimal hyperplane is know as margin, and the closest data-points are known as support vectors. Margin is an area which do not contains any data points.



- So, when we are choosing optimal hyperplane we will choose one among set of hyperplane which is highest distance from the closest data points.
- If optimal hyperplane is very close to data points then margin will be very small and it will generalize well for training data but when an unseen data will come it will fail to generalize well as explained above.
- So our goal is to maximize the margin so that our classifier is able to generalize well for unseen instances.


X2
Maximum
Margin
Positive
Hyperplane
Maximum
Margin
Hyperplane
Support
Vectors
Negative Hyperplane
X1



$$\vec{X}.\vec{w} + \mathrm{b} \geq 0$$

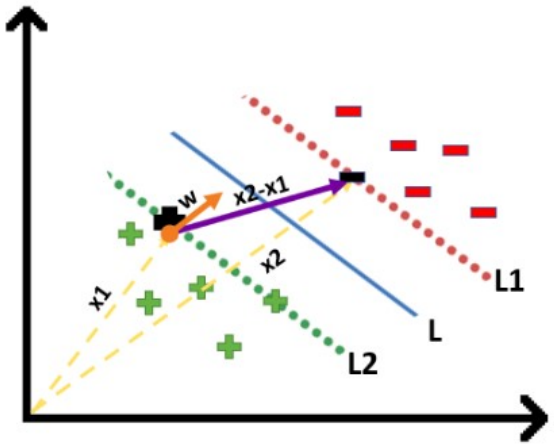
hence

$$\mathrm{y} = \begin{cases} +1 & \text{if } \vec{X}.\vec{w} + \mathrm{b} \geq 0 \\ -1 & \text{if } \vec{X}.\vec{w} + \mathrm{b} < 0 \end{cases}$$

To find maximum margin

- That constraint is that **"We'll calculate the distance (d) in such a way that no positive or negative point can cross the margin line".**
- **for negative points,** $\vec{w}.\vec{X}+b \leq -1$
- **For positive points,** $\vec{w}.\vec{X}+b \geq 1$
- We assume that negative classes have ***y=-1*** and positive classes have ***y=1.***
- ***Common expression*** $y_i(\vec{w}.\vec{X}+b) \geq 1$

- We will take 2 support vectors, 1 from the negative class and 2nd from the positive class. The distance between these two vectors x1 and x2 will be *(x1-x2) vector*.
- To find the distance , We take a vector ‘w’ perpendicular to the hyperplane and then find the projection of (x1-x2) vector on ‘w’.
- This perpendicular vector should be a unit vector. So, to make this ‘w’ a unit vector we divide this with the norm of ‘w’.



- So, we need to find (X1-X2). $\frac{\vec{w}}{\|w\|}$
- Distance d = x1. $\frac{\vec{w}}{\|w\|}$ - x2. $\frac{\vec{w}}{\|w\|}$ (1)

for positive point $y = 1$

$$\Rightarrow 1 \times (\vec{w}.x1 + b) = 1$$

$$\Rightarrow \vec{w}.x1 = 1 - b \quad ------(2)$$

Similarly for negative point $y = -1$

$$\Rightarrow -1 \times \left(\vec{w}.x2 + b\right) = 1$$

$$\Rightarrow \vec{w}.x2 = -b - 1 \quad ------(3)$$

Putting equations (2) and (3) in equation (1) we get:

$$\Rightarrow \frac{(1-b)-(-b-1)}{\|w\|}$$

$$\Rightarrow \frac{1-b+b+1}{\|w\|} = \frac{2}{\|w\|} = d$$

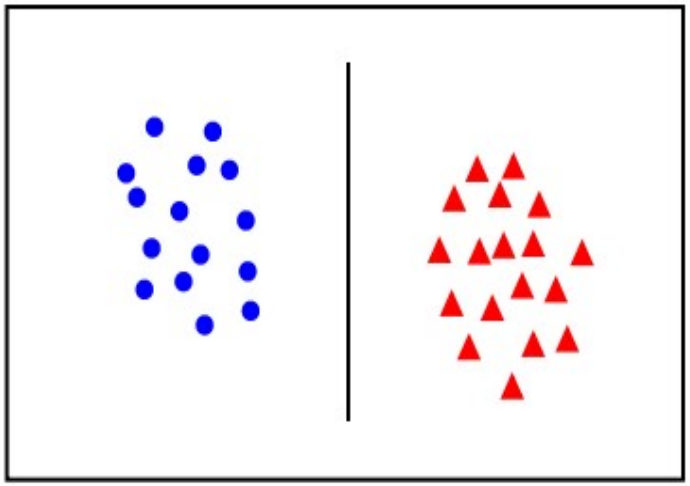
Hence the equation which we have to maximize is:

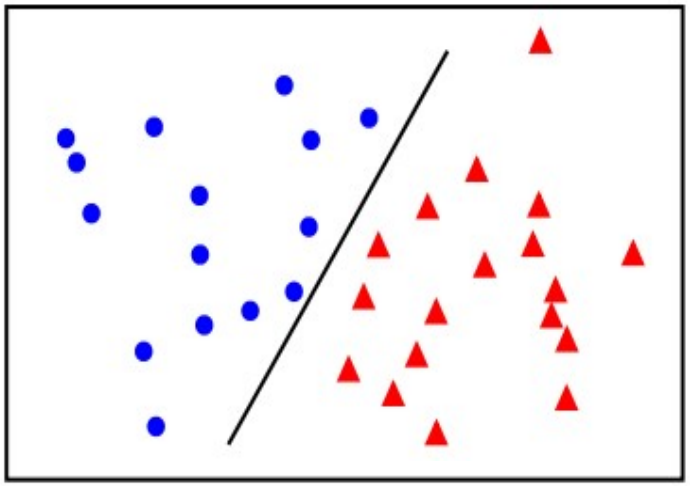
$$\text{argmax}(w^*, b^*) \frac{2}{\|w\|} \text{ such that } y_i\left(\vec{w}.\vec{X} + b\right) \geq 1$$

https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

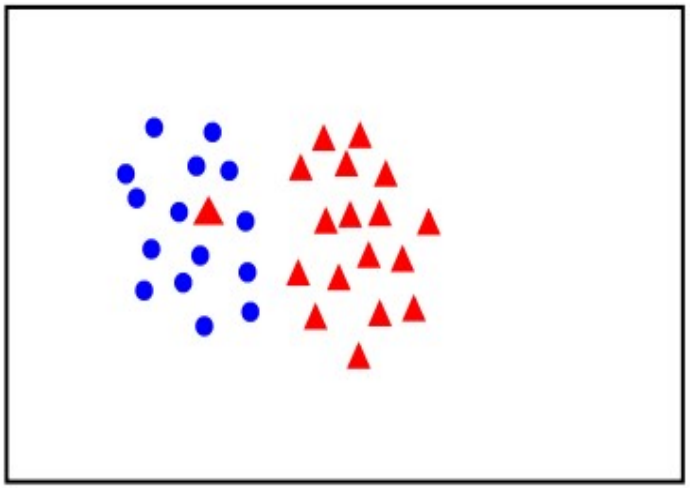
Linearly separable/not

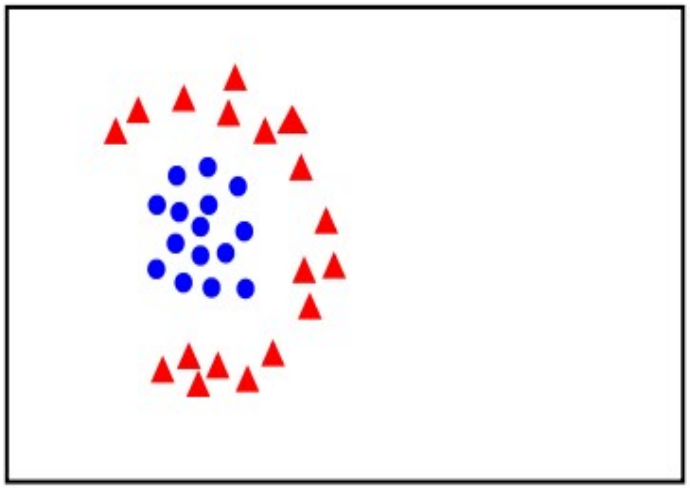
linearly separable





not linearly separable





Soft Margin SVM

- In real-life applications we don't find any dataset which is linearly separable, what we'll find is either an almost linearly separable dataset or a non-linearly separable dataset. In this scenario, we can't use the trick we proved above because it says that it will function only when the dataset is perfectly linearly separable.
- To tackle this problem what we do is modify that equation in such a way that it allows few misclassifications that means it allows few points to be wrongly classified.

- To make a soft margin equation we add 2 more terms to this equation which is **zeta** and multiply that by a **hyperparameter 'c'**
-

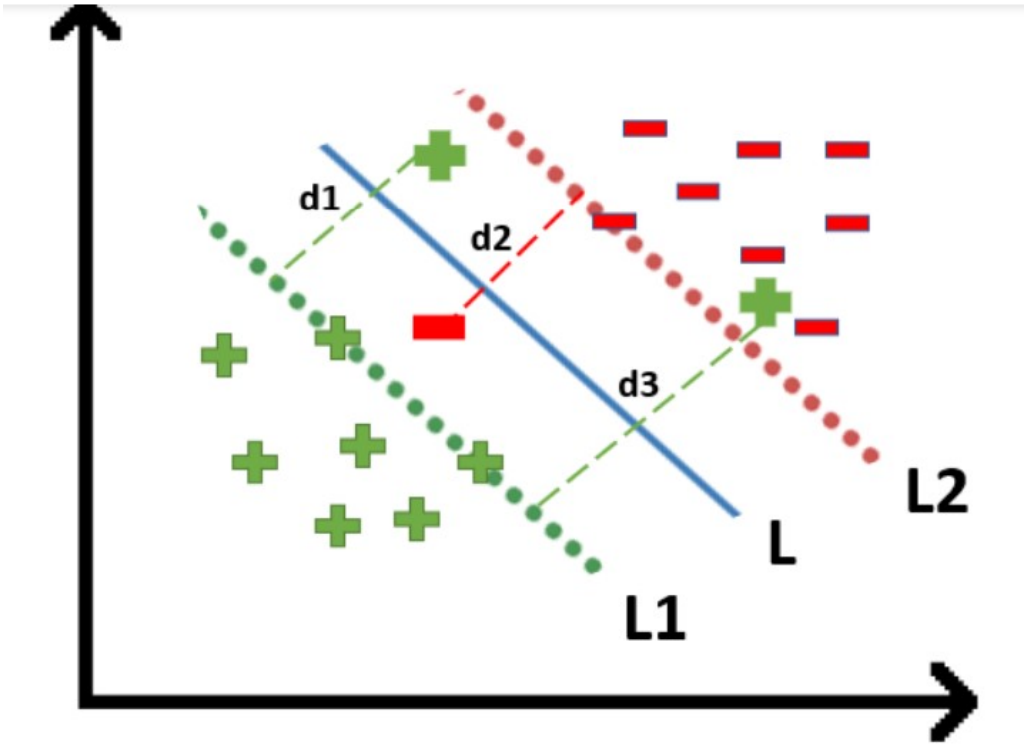
$$\operatorname{argmin}\left(\mathrm{w}^{*}, \mathrm{b}^{*}\right) \frac{\|\mathrm{w}\|}{2} + c \sum_{i=1}^{n} \zeta_i$$

For all the correctly classified points our zeta will be equal to 0 and for all the incorrectly classified points the zeta is simply the distance of that particular point from its correct hyperplane

- We know that *max[f(x)]* can also be written as *min[1/f(x)]*, it is common practice to minimize a cost function for optimization problems; therefore, we can invert the function.

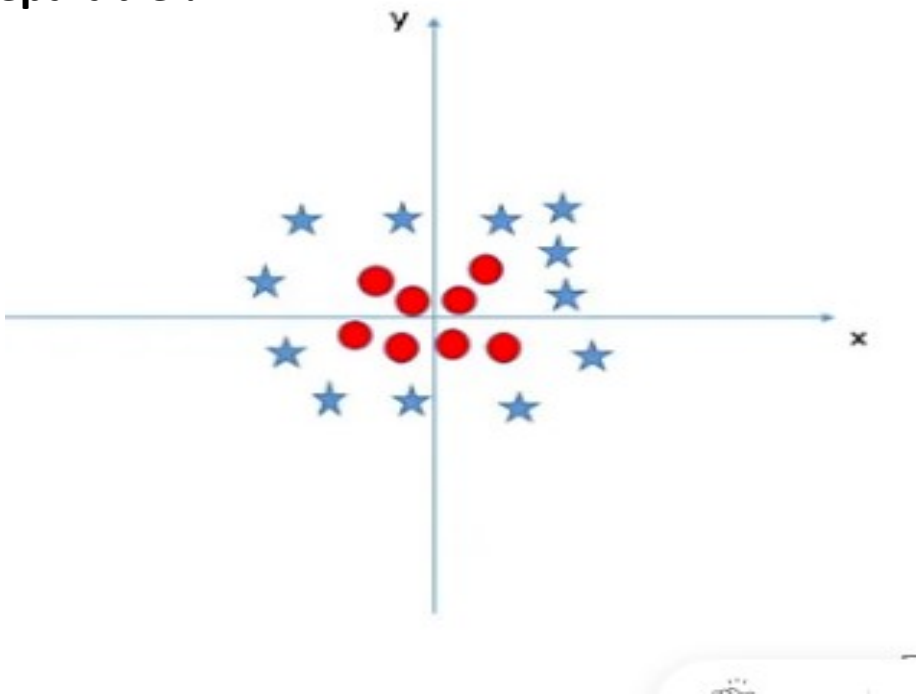
$$\text{argmin}(\text{w}^*, \text{b}^*) \ \frac{\|\text{w}\|}{2} \text{ such that } \text{y}_\text{i}\left(\overrightarrow{w}.\overrightarrow{X} + b\right) \geq 1$$

- **SVM Error = Margin Error + Classification Error.** The higher the margin, the lower would-be margin error, and vice versa.



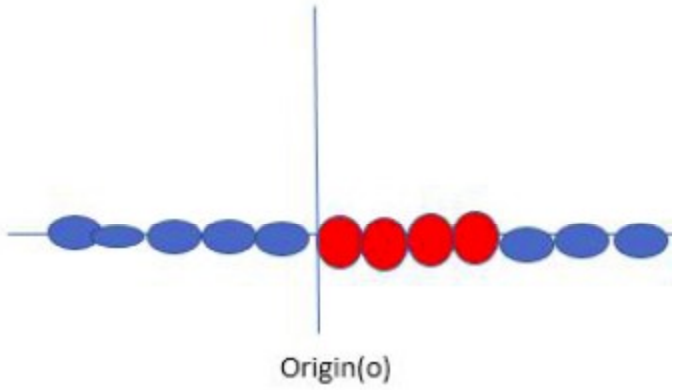
Nonlinear SVM

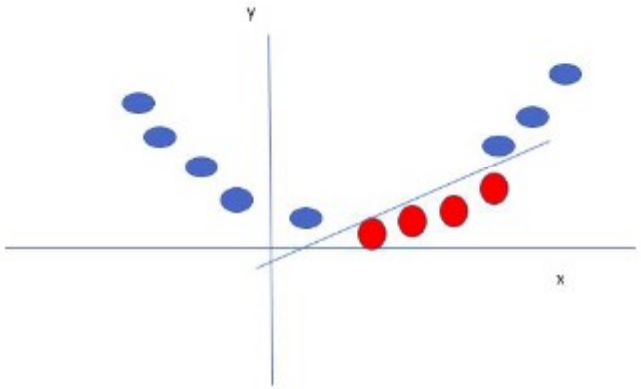
What if data points is not linearly separable ?
raw data are always non-linear



We will add one extra dimension to the data points to make it separable.

SVM solves this by creating a new variable using a kernel. We call a point xi on the line and we create a new variable yi as a function of distance from origin o.
so if we plot this we get something like as shown below



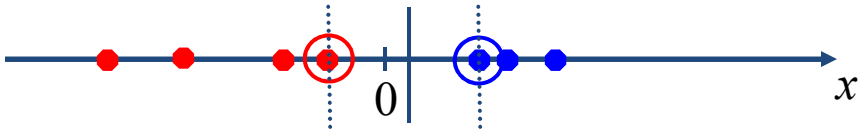


SVM Kernel

- A non-linear function that creates a new variable is referred to as kernel.
- The SVM kernel is a function that takes low dimensional input space and transforms it into higher-dimensional space, ie it converts non separable problem to separable problem.
- It is mostly useful in non-linear separation problems. Simply put the kernel, it does some extremely complex data transformations then finds out the process to separate the data based on the labels or outputs defined.

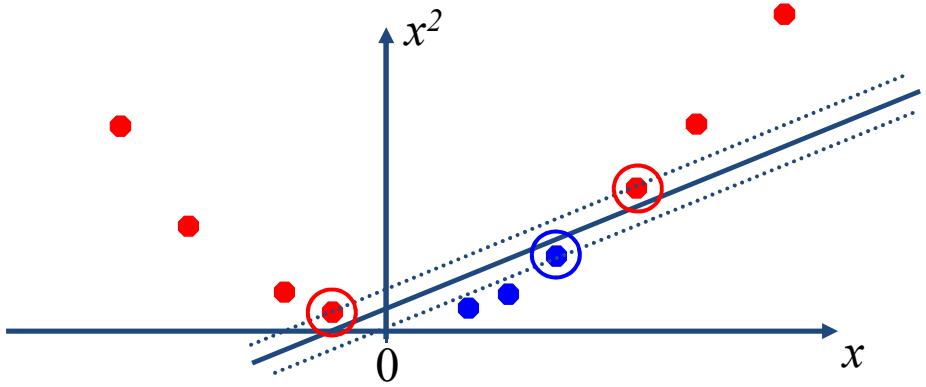
Non-linear SVMs

- Datasets that are linearly separable (with some noise) work out great:



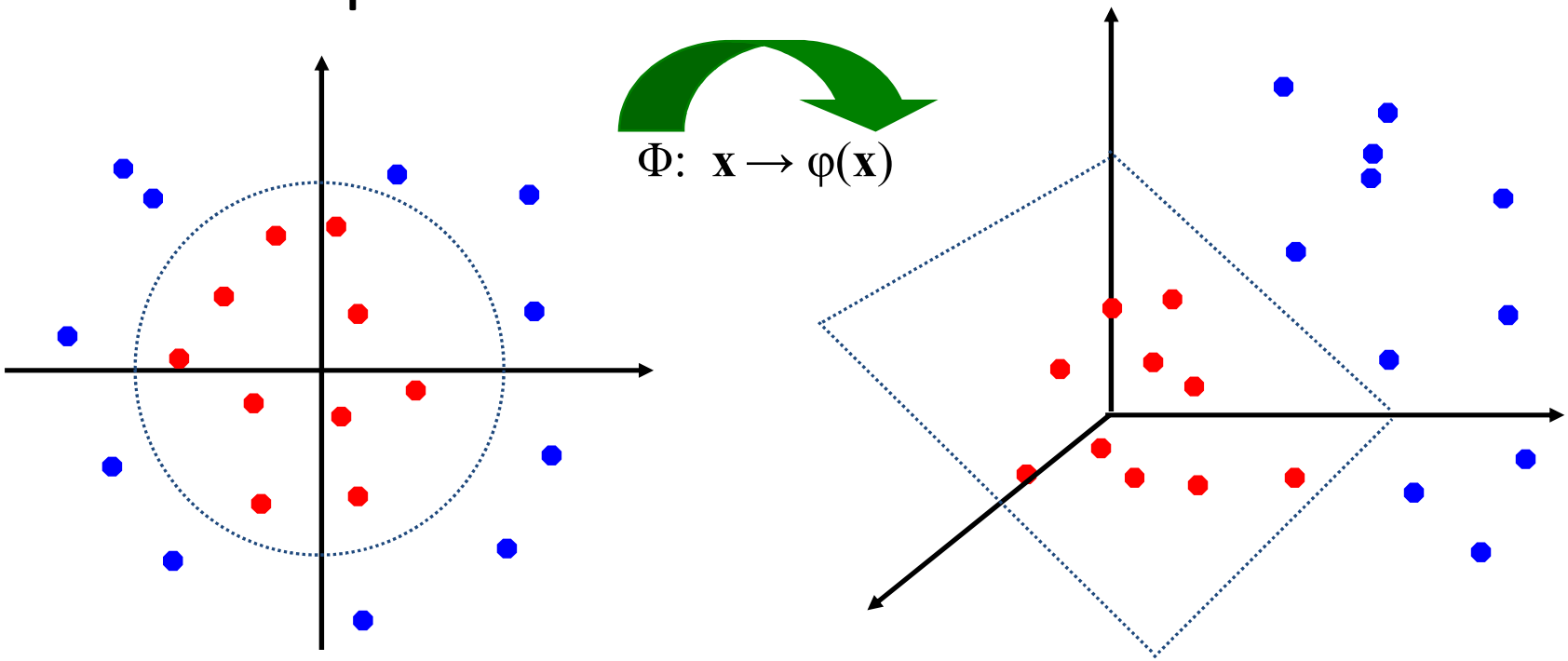
- But what are we going to do if the dataset is just too hard?

- How about … mapping data to a higher-dimensional space:



Non-linear SVMs: Feature spaces

- General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:



Kernels

- Mathematical definition: K(x, y) = <f(x), f(y)>
  - Here K is the kernel function
  - x, y are n dimensional inputs.
  - f is a map from n-dimension to m-dimension space.
  - < x,y> denotes the dot product.
  - Usually m is much larger than n.

Different Kernel Functions

- Different types of kernels
  - Linear
  - Polynomial
  - Gaussian (RBF)

Linear kernel: $K(x_i, x_j) = x_i . x_j$

Polynomial of power p:

$$K(x_i, x_j) = (1 + x_i . x_j)^p$$

Gaussian (radial-basis function):

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

Example for Linear SVM

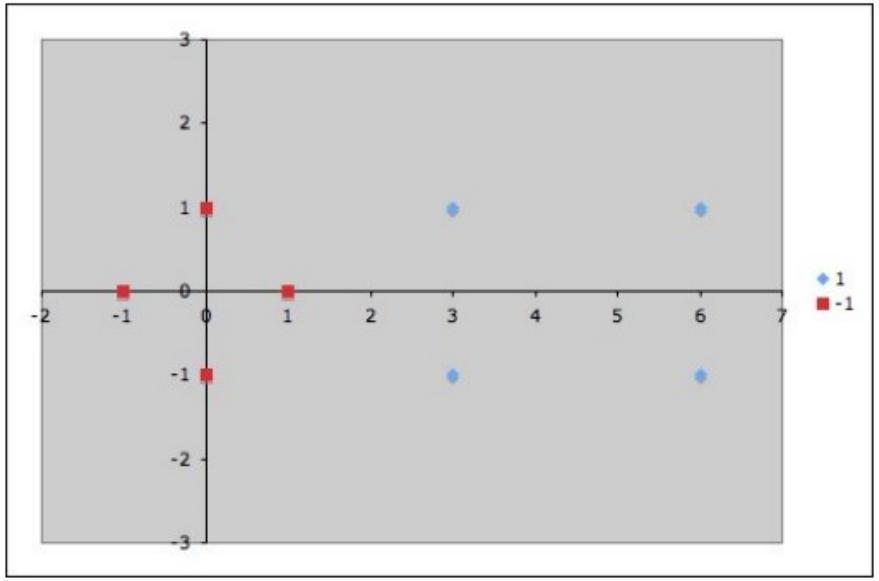
- We would like to discover a simple SVM that accurately discriminates the two classes.

Suppose we are given the following positively labeled data points in $\Re^2$:

$$\left\{\begin{pmatrix}3\\1\end{pmatrix},\begin{pmatrix}3\\-1\end{pmatrix},\begin{pmatrix}6\\1\end{pmatrix},\begin{pmatrix}6\\-1\end{pmatrix}\right\}$$

and the following negatively labeled data points in $\Re^2$ (see Figure 1):

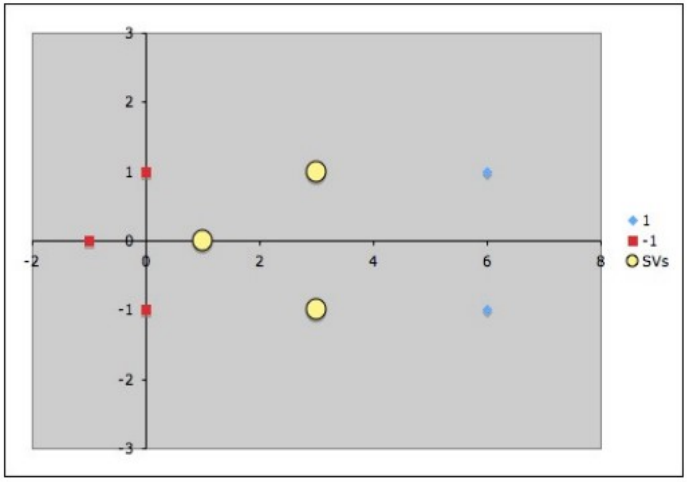
$$\left\{\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}0\\-1\end{pmatrix},\begin{pmatrix}-1\\0\end{pmatrix}\right\}$$



Example for Linear SVM

The support vectors are:

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



Two +ve support vectors and one –ve support vector. Hence frame 3 equations:

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

Assume each support vector as 3dimensions (since (n+1) coefficients are to be identified):
i.e., S1=(1, 0, 1), s2=(3, 1, 1) and s3= (3, -1, 1)

- Substitute values

$$\alpha 1\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix}+\alpha 2\begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix}+\alpha 3\begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix}=-1$$

$$\alpha 1\,(1+0+1)+\alpha 2\,(3+0+1)+\alpha 3\,(3+0+1)=-1$$

$$2\alpha 1+4\alpha 2+4\alpha 3=-1$$

$$\alpha 1\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix}+\alpha 2\begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix}+\alpha 3\begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix}=1$$

$$\alpha 1\,(3+0+1)+\alpha 2\,(9+1+1)+\alpha 3\,(9-1+1)=1$$

$$4\alpha 1+11\alpha 2+9\alpha 3=1$$

$$\alpha 1\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix}+\alpha 2\begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix}+\alpha 3\begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix}=1$$

$$\alpha 1\,(3+0+1)+\alpha 2\,(9-1+1)+\alpha 3\,(9+1+1)=1$$

$$4\alpha 1+9\alpha 2+11\alpha 3=1$$

Example for Linear SVM

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

A little algebra reveals that the solution to this system of equations is
α1 = −3.5, α2 = 0.75 and α3 = 0.75.

Calculating the weights based on α1, α2 and α3:

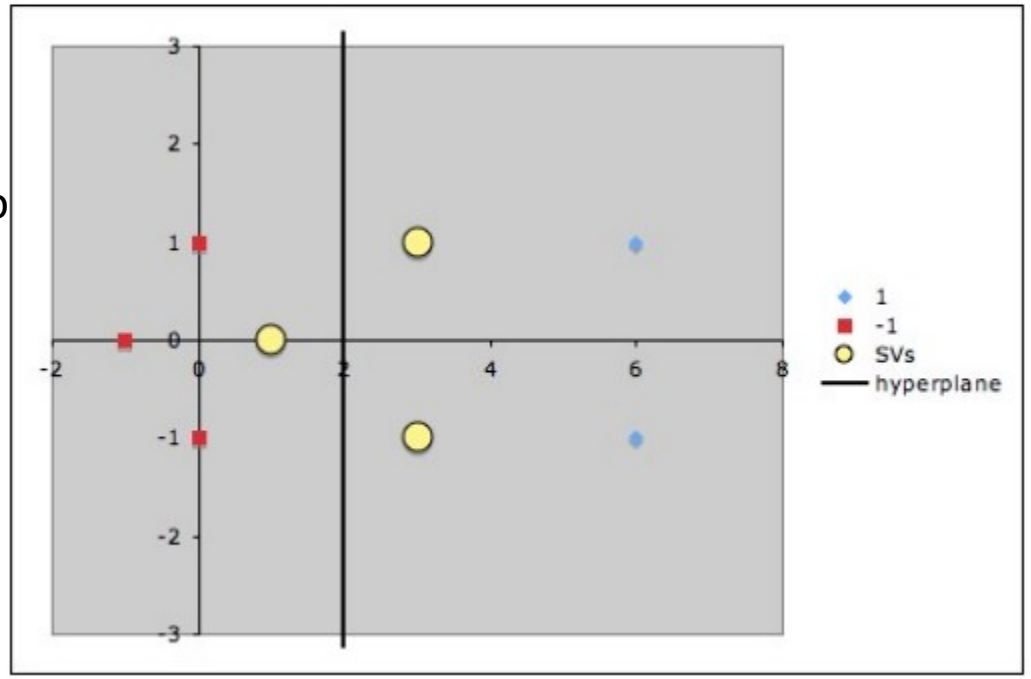
$$
\begin{aligned}
\tilde{w} &= \sum_i \alpha_i \tilde{s}_i \\
&= -3.5 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}
\end{aligned}
$$

Example for Linear SVM

Finally, remembering that our vectors are augmented with a bias, we can equate the last entry in $\tilde{w}$ as the hyperplane offset b and write the separating hyperplane equation $y = wx + b$ with $w = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $b = -2$. Plotting the line gives the expected decision surface

origin 0 is at a distance of −b /||w|| = 2/1=2 from hyperplane.

ttps://www.youtube.co
ı/watch?v=TPVzIKJOcN



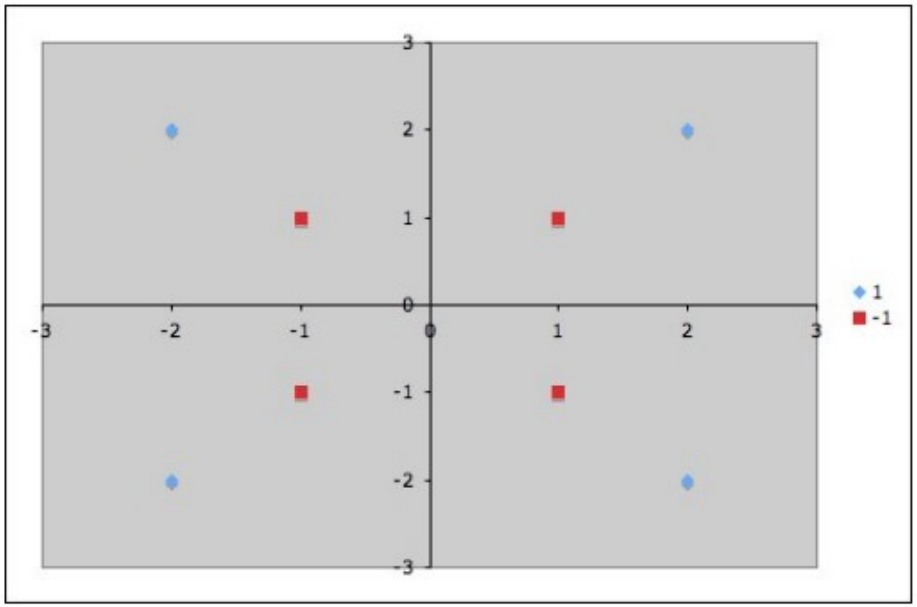
Example for Non-Linear SVM

Now suppose instead that we are given the following positively labeled data points in $\Re^2$:

$$\left\{\begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \end{pmatrix}\right\}$$

and the following negatively labeled data points in $\Re^2$ (see Figure 5):

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



Example for Non-Linear SVM

- we must use a nonlinear SVM (that is, one whose mapping function Φ is a nonlinear mapping from input space into some feature space).

Define

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

- Given data

$$\left\{ \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \end{pmatrix} \right\}$$

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

Example for Non-Linear SVM

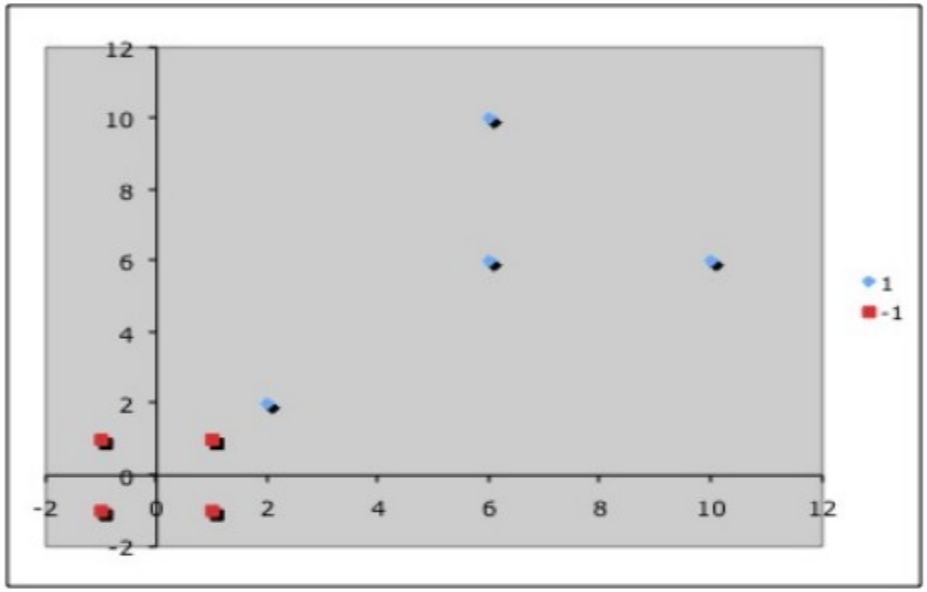
- we can rewrite the data in feature space as

$$\left\{\begin{pmatrix}2\\2\end{pmatrix},\begin{pmatrix}6\\2\end{pmatrix},\begin{pmatrix}6\\6\end{pmatrix},\begin{pmatrix}2\\6\end{pmatrix}\right\}$$

for the positive examples and

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

for the negative examples



Example for Non-Linear SVM

Supporting vectors:

$$\left\{ s_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, s_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \right\}$$

$$\begin{aligned} \alpha_1 \tilde{s_1} \cdot \tilde{s_1} + \alpha_2 \tilde{s_2} \cdot \tilde{s_1} &= -1 \\ \alpha_1 \tilde{s_1} \cdot \tilde{s_2} + \alpha_2 \tilde{s_2} \cdot \tilde{s_2} &= +1 \end{aligned}$$

Now, computing the dot products results in

$$\begin{aligned} 3\alpha_1 + 5\alpha_2 &= -1 \\ 5\alpha_1 + 9\alpha_2 &= +1 \end{aligned}$$

Solving the equations gives $\alpha 1 = -7$ and $\alpha 2 = 4$.

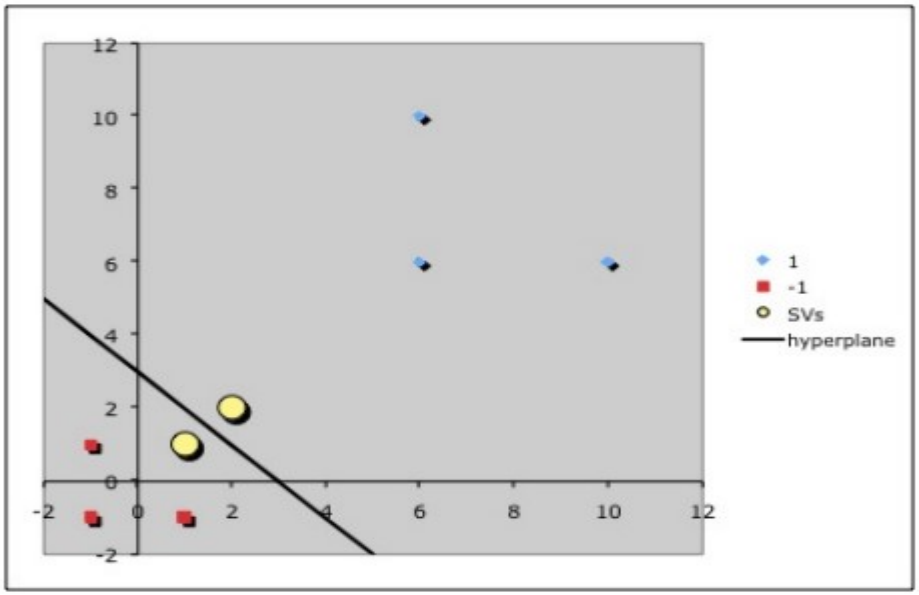
Example for Non-Linear SVM

$$\tilde{w} = \sum \alpha_i \tilde{s}_i$$

$$= -7 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + 4 \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 \\ 1 \\ -3 \end{pmatrix}$$

giving us the separating hyperplane equation $y = wx + b$ with $w = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $b = -3$. Plotting the line gives the expected decision surface .



Pro's and Con's of SVM

- **Pro's:**
  - It is really effective in the higher dimension.
  - Effective when the number of features are more than training examples.
  - Best algorithm when classes are separable
  - The hyperplane is affected by only the support vectors thus outliers have less impact.
  - SVM is suited for extreme case binary classification.
- **Con's:**
  - For larger dataset, it requires a large amount of time to process.
  - Does not perform well in case of overlapped classes.
  - Selecting the appropriate kernel function can be tricky.

Important points to remember

- The SVM's are less effective when the data is noisy and contains overlapping points

- The effectiveness of an SVM depends upon:
  - Selection of Kernel
  - Kernel Parameters
  - Soft Margin Parameter