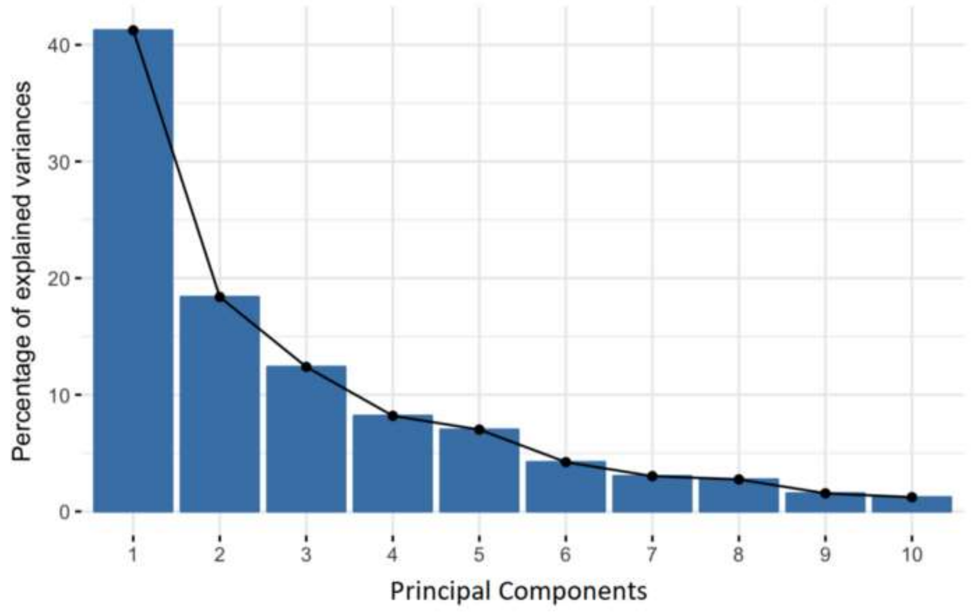
Principal Component Analysis (PCA) and Support Vector Machine (SVM)

Principal Component Analysis

- Principal component analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.
- Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.
- So, to sum up, the idea of PCA is simple — **reduce the number of variables of a data set, while preserving as much information as possible.**

Principal Component Analysis

- The idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on until having something like shown in the plot below.
- Geometrically speaking, principal components represent the directions of the data that explain a **maximal amount of variance**, that is to say, the lines that capture most information of the data.
- This enables you to remove those dimensions along which the data is almost flat. This decreases the dimensionality of the data while keeping the variance (or spread) among the points as close to the original as possible.
- To put all this simply, just think of principal components as new axes that provide the best angle to see and evaluate the data, so that the differences between the observations are better visible.



Principal Component Analysis

- Step-by-Step Explanation of PCA
  - 1. Normalize the data
  - 2. Build the covariance matrix
  - 3. Find the Eigenvectors and Eigenvalues
  - 4. Sort the eigenvectors in highest to lowest order and select the number of principal components.

STEP 1: STANDARDIZATION

- The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.
- Mathematically, this can be done by subtracting the mean and dividing by the standard deviation for each value of each variable.

$$z = \frac{value - mean}{standard\ deviation}$$

- Once the standardization is done, all the variables will be transformed to the same scale

STEP 2: COVARIANCE MATRIX COMPUTATION

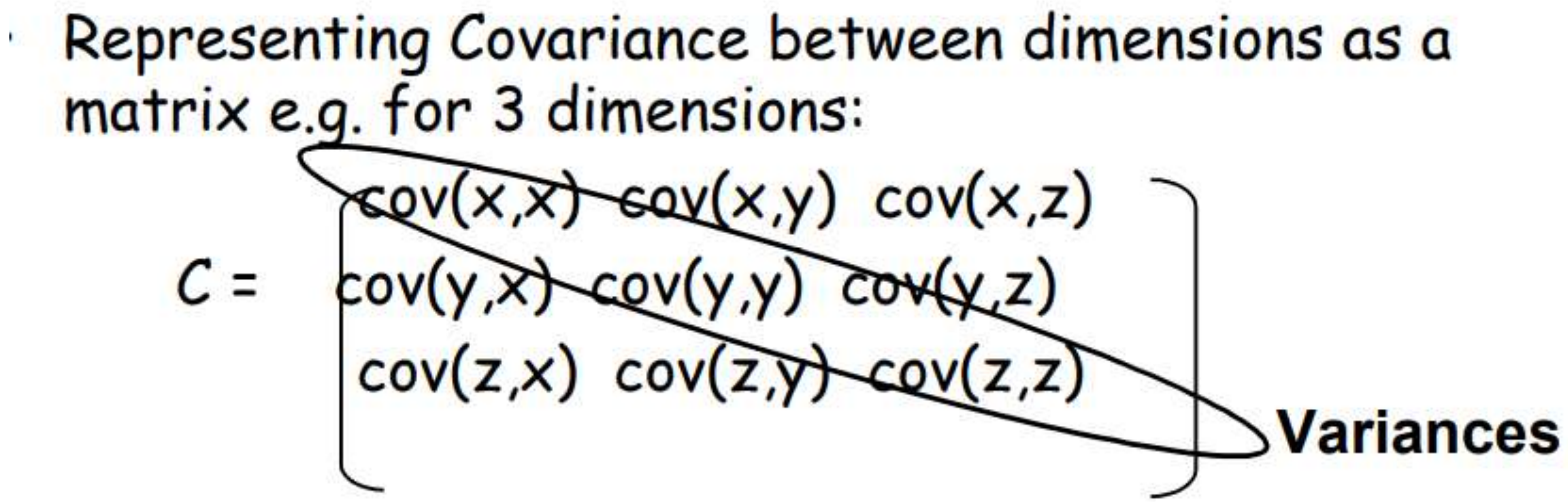
- The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.
- The covariance matrix is a $p \times p$ symmetric matrix (where p is the number of dimensions) that has as entries the covariance associated with all possible pairs of the initial variables.
- The covariance is commutative (Cov(a,b)=Cov(b,a)),
- Which means that the upper and the lower triangular portions are equal.

Covariance

- Variance and Covariance are a measure of the “spread” of a set of points around their center of mass (mean)
- Variance – measure of the deviation from the mean for points in one dimension e.g. heights
- Covariance as a measure of how much each of the dimensions vary from the mean with respect to each other
- Covariance is measured between 2 dimensions to see if there is a relationship between the 2 dimensions e.g. number of hours studied & marks obtained.
- The covariance between one dimension and itself is the variance

$$\text{covariance}\,(X,Y) = \frac{\Sigma_{i=1}^{n} (\overline{X_i} - X)\,(Y_i \overline{-} Y)}{(n - 1)}$$

Covariance



- Representing Covariance between dimensions as a matrix e.g. for 3 dimensions:

$$C = \begin{pmatrix} \mathrm{cov}(x,x) & \mathrm{cov}(x,y) & \mathrm{cov}(x,z) \\ \mathrm{cov}(y,x) & \mathrm{cov}(y,y) & \mathrm{cov}(y,z) \\ \mathrm{cov}(z,x) & \mathrm{cov}(z,y) & \mathrm{cov}(z,z) \end{pmatrix}$$

Variances

Covariance

- What is the interpretation of covariance calculations?
- e.g.: 2-dimensional data set
  - x: number of hours studied for a subject
  - y: marks obtained in that subject
  - covariance value is say: 104.53
  - what does this value mean?

Covariance

- The exact value is not as important as its sign
- A **positive** value of covariance indicates both dimensions increase or decrease together e.g. as the number of hours studied increases, the marks in that subject increase.
- A **negative** value indicates while one increases the other decreases or vice-versa e.g. active social life at PSU vs performance in CS dept.
- If the covariance is **zero**: the two dimensions are independent of each other e.g. heights of students vs the marks obtained in a subject

STEP 3: COMPUTE THE EIGENVECTORS AND EIGENVALUES OF THE COVARIANCE MATRIX

- The term eigenvalue can be termed as characteristic value, characteristic root, proper values, or latent roots as well.
- In simple words, the eigenvalue is a scalar that is used to transform the eigenvector. The basic equation is:
- $$Av = \lambda v$$
- The number or scalar value “λ” is an eigenvalue of A.v is an eigenvector of A corresponding to eigenvalue, λ.
- eigenvectors of the Covariance matrix are actually *the **directions** of the axes where there is the most variance*(most information) and that we call Principal Components. The eigenvalues are simply the coefficients attached to eigenvectors, which give the *amount of variance carried in each Principal Component*.
- By ranking your eigenvectors in order of their eigenvalues, highest to lowest, you get the principal components in order of significance

LAST STEP: RECAST THE DATA ALONG THE PRINCIPAL COMPONENTS AXES

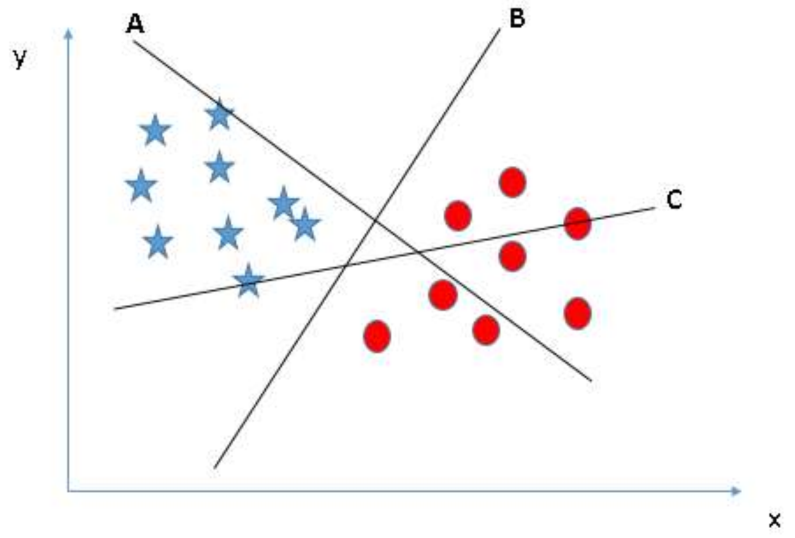
- This can be done by multiplying the transpose of the original data set by the transpose of the feature vector.

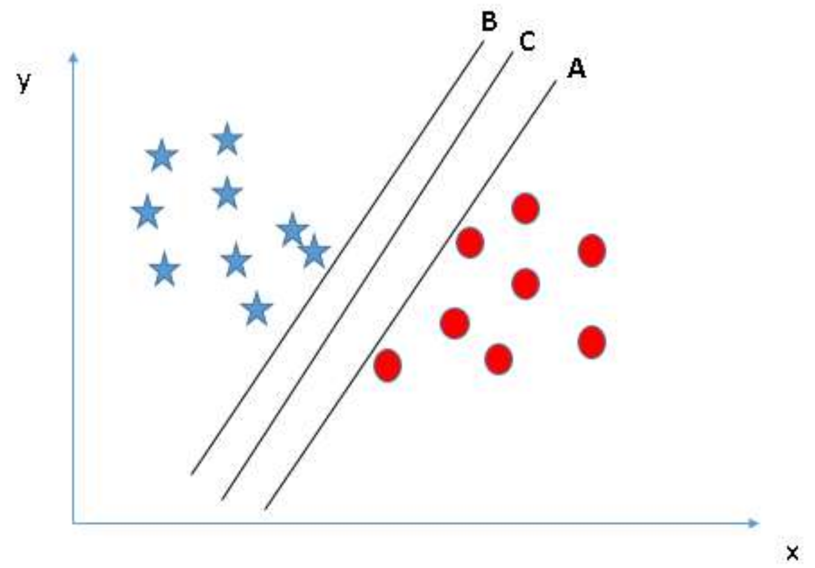
$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataSet^T$$

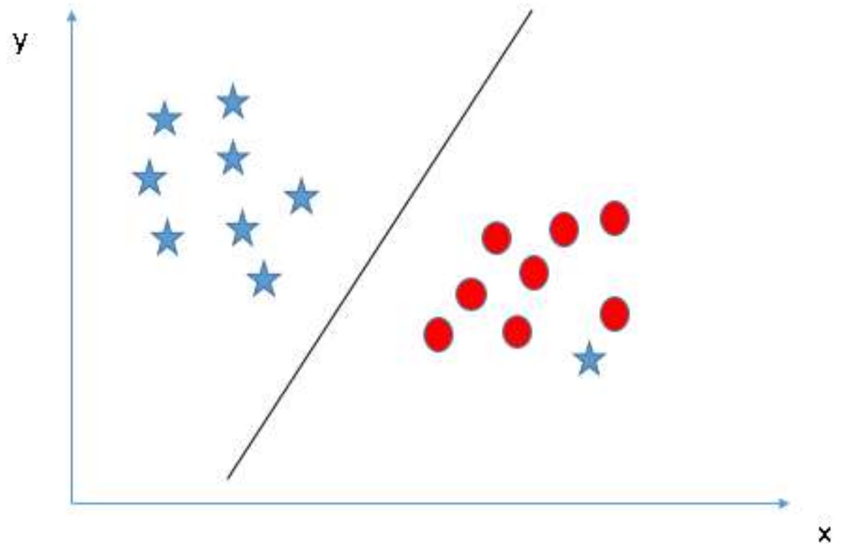
Support Vector Machine

- "Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems.
- 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)
- Then, we perform classification by finding the **hyper-plane** that differentiates the two classes very well

Support Vector Machine



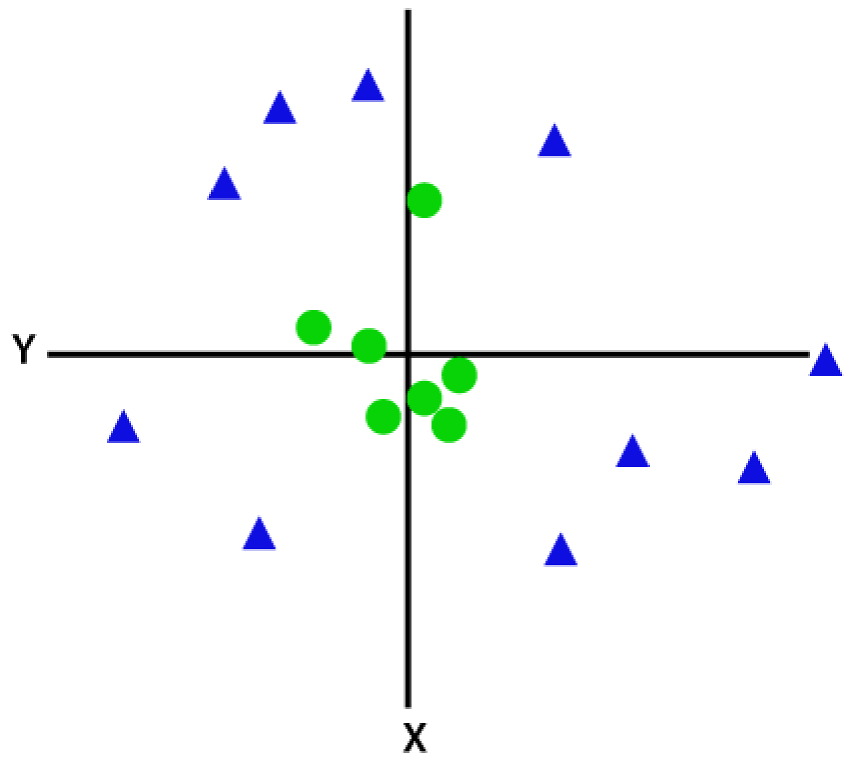


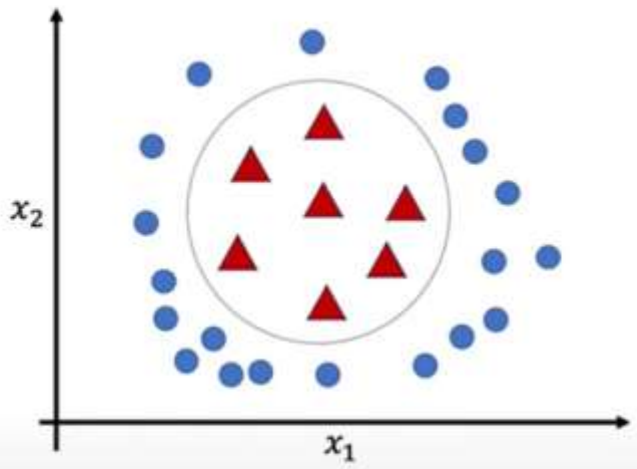


Support Vector Machine

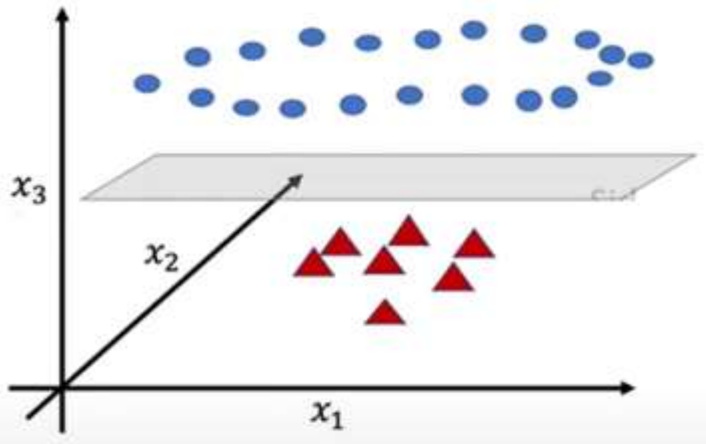
- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset **cannot** be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Support Vector Machine





SVM in 2 dimensions



SVM in 3 dimensions

Support Vector Machine

- Hyperplane
- Support Vectors
- Margin

Support Vector Machine

For $P_2(3,3)$

$w^T X = \begin{bmatrix} -1 \\ 0 \end{bmatrix} [3, 3]$

$=-3$ (negative)

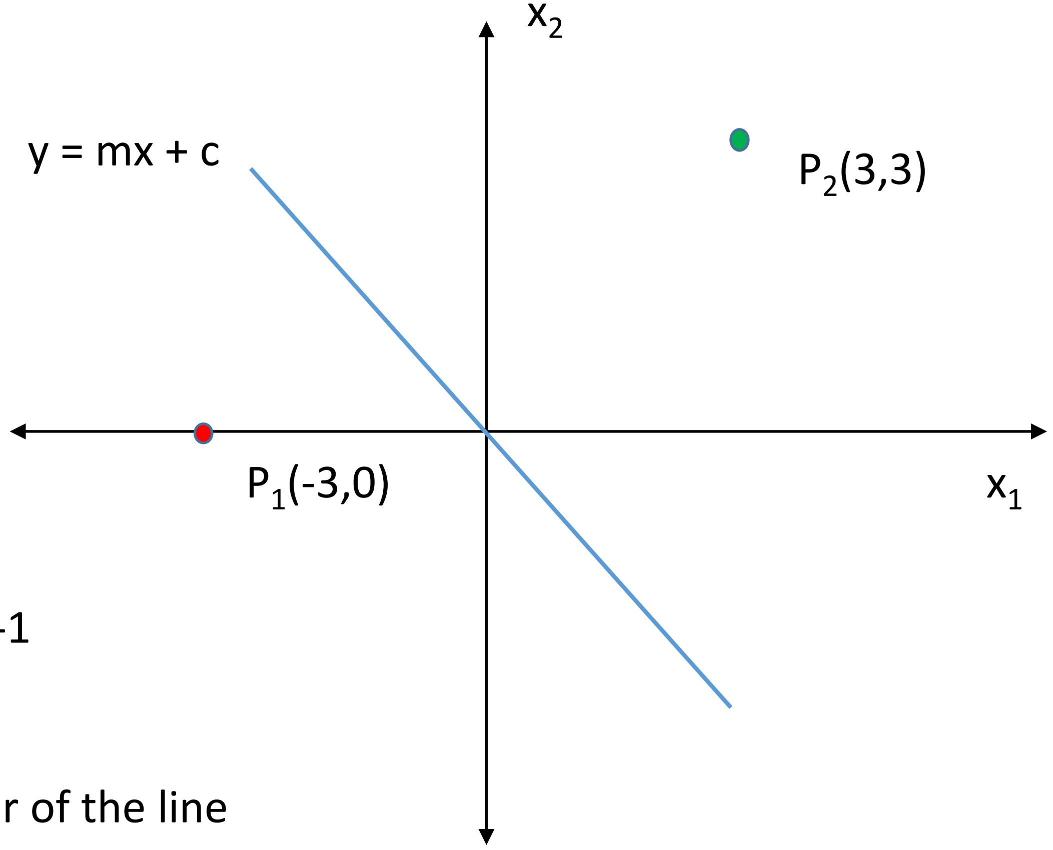
$X = P_1$ or P_2

$w^T X$

For $P_1(-3,0)$

$w^T X = \begin{bmatrix} -1 \\ 0 \end{bmatrix} [-3, 0]$

$=3$ (positive)



Inference= for all points which lie in the right side of hyperplane, $w^T X$ are positive

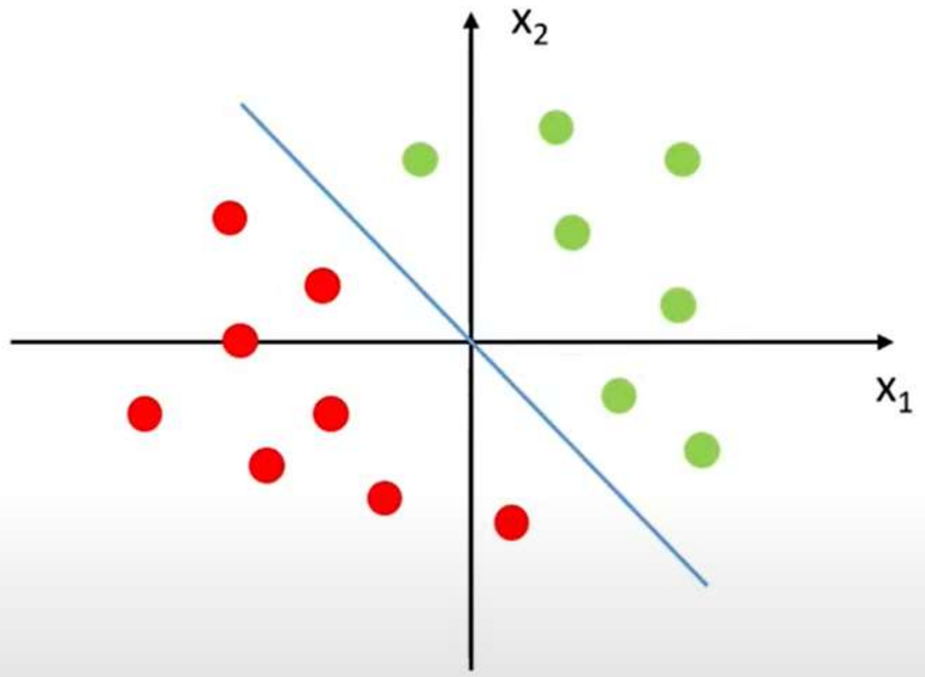
Inference= for all points which lie on the left side of hyperplane $w^T X$ are positive

Let $m = -1$

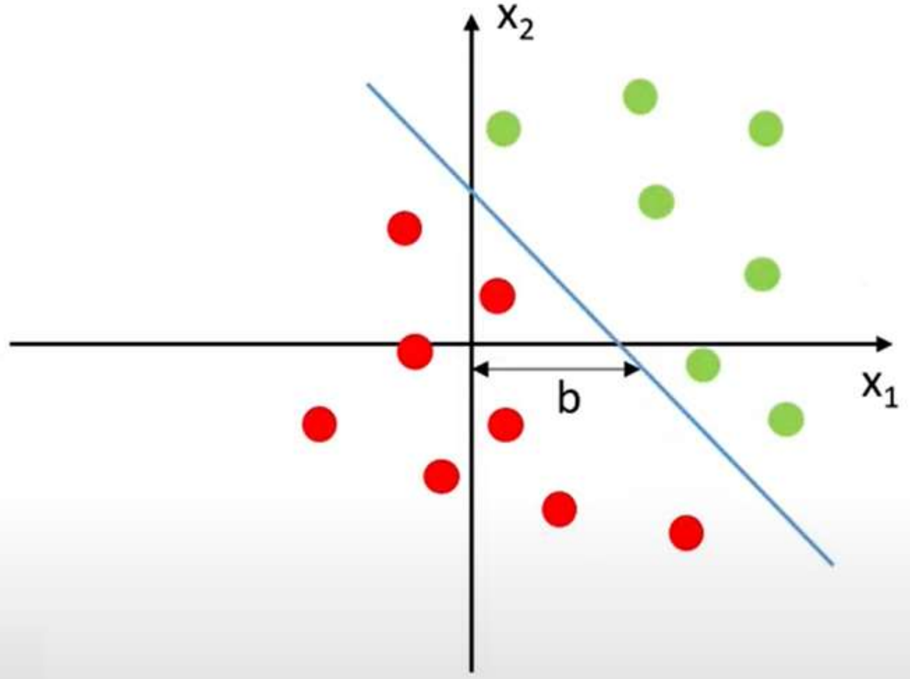
$c = 0$

$w \rightarrow$ parameter of the line

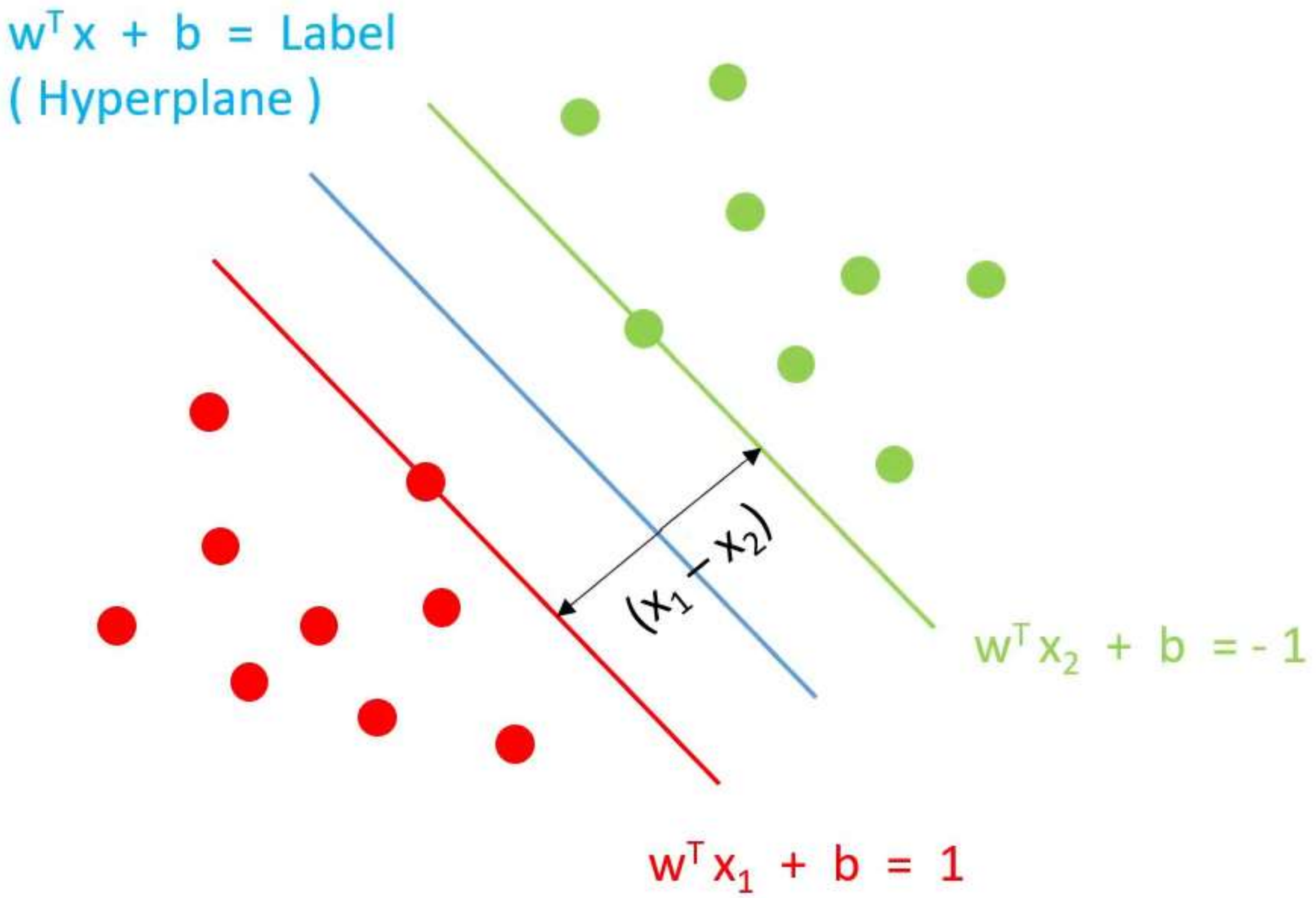
$(m, c) = (-1, 0)$



$w^Tx = \text{Label}$



$w^Tx + b = \text{Label}$



$$w^T x_1 + b = 1$$

$$(-)\quad w^T x_2 + b = -1$$

$$w^T (x_1 - x_2) = 2$$

$$\frac{w^T}{||w||}(x_1 - x_2) = \frac{2}{||w||}$$

$$\mathbf{(x_1 - x_2) = \frac{2}{||w||}} \qquad \text{(margin}$$

Support Vector Machine

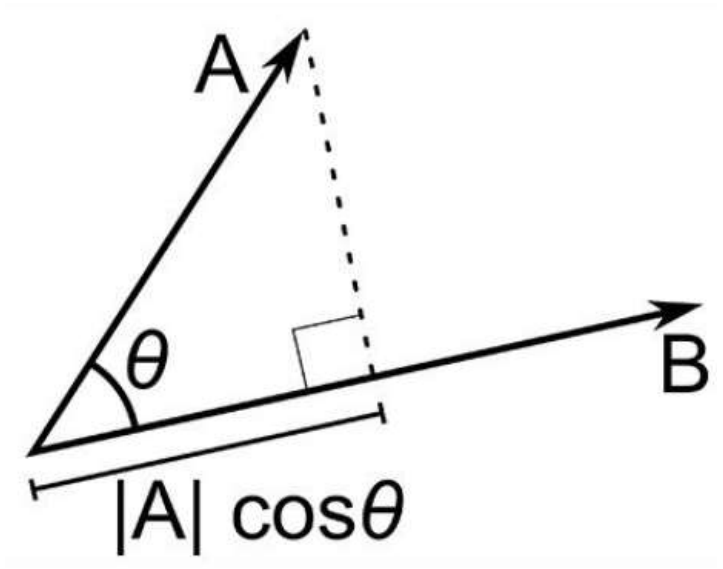
$$\max \left(\frac{2}{||w||} \right) \text{ Such that,}$$

$$y_i = \begin{cases} -1, & w^T x_1 + b \leq -1 \\ 1, & w^T x_1 + b \geq 1 \end{cases}$$

Mathematical Intuition behind Support Vector Machine

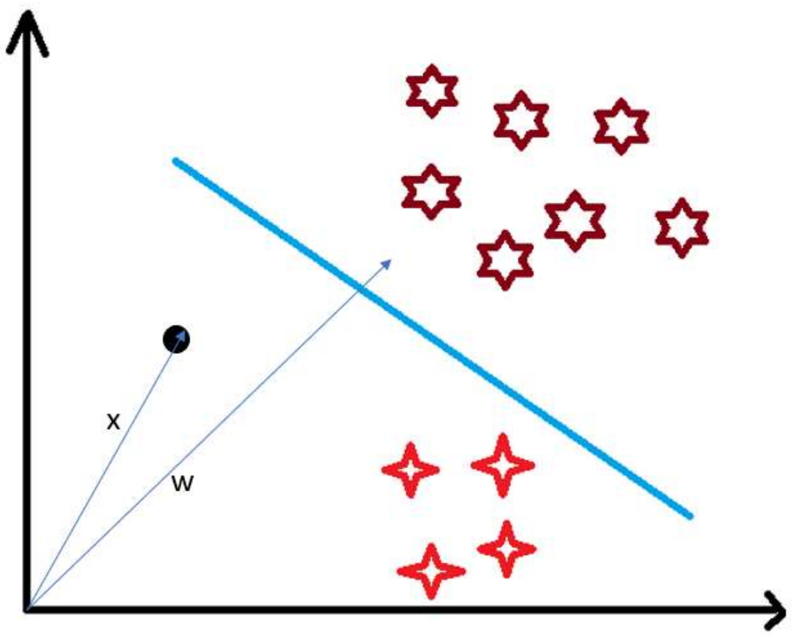
Dot-Product

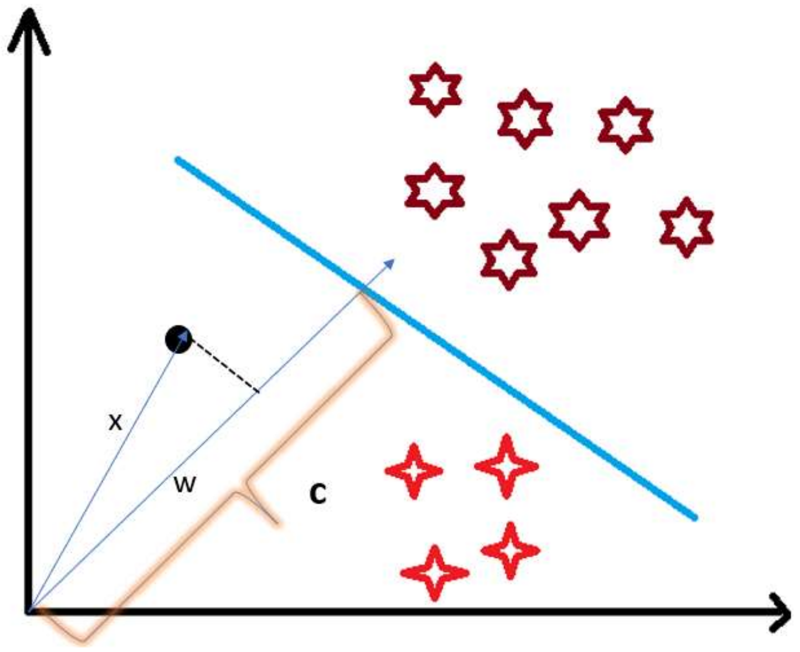
- The dot product is used to get a scalar value as a resultant whereas the cross-product is used to obtain a vector again.



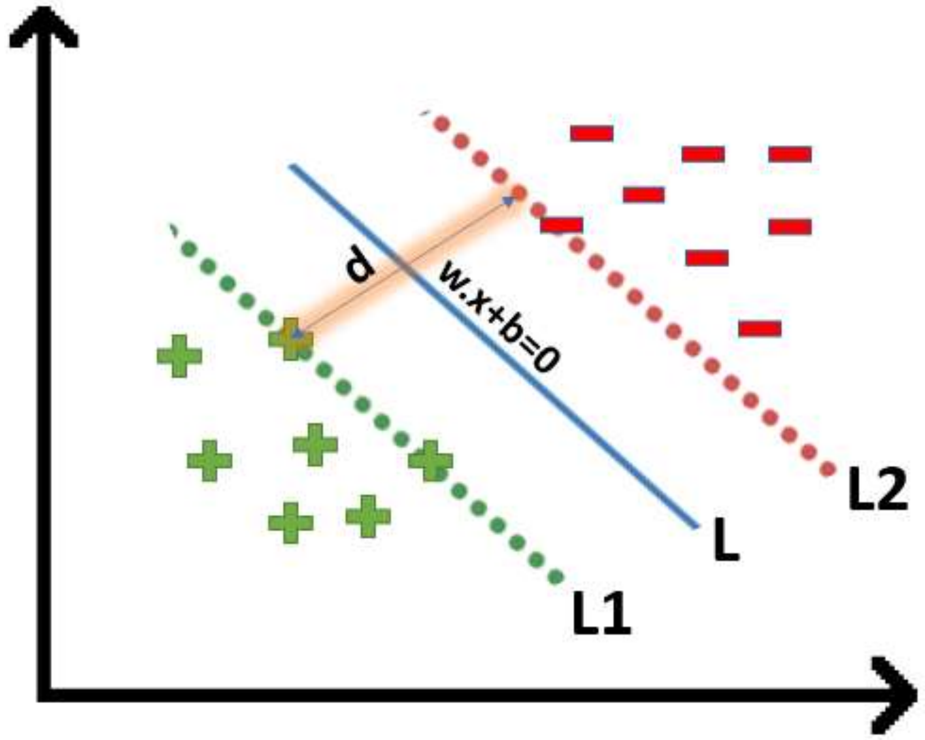
Use of Dot Product in SVM:

- Let’s say the distance of vector w from origin to decision boundary is ‘c’. Now we take the projection of X vector on w.
- If the dot product is greater than ‘c’ then we can say that the point lies on the right side. If the dot product is less than ‘c’ then the point is on the left side and if the dot product is equal to ‘c’ then the point lies on the decision boundary.

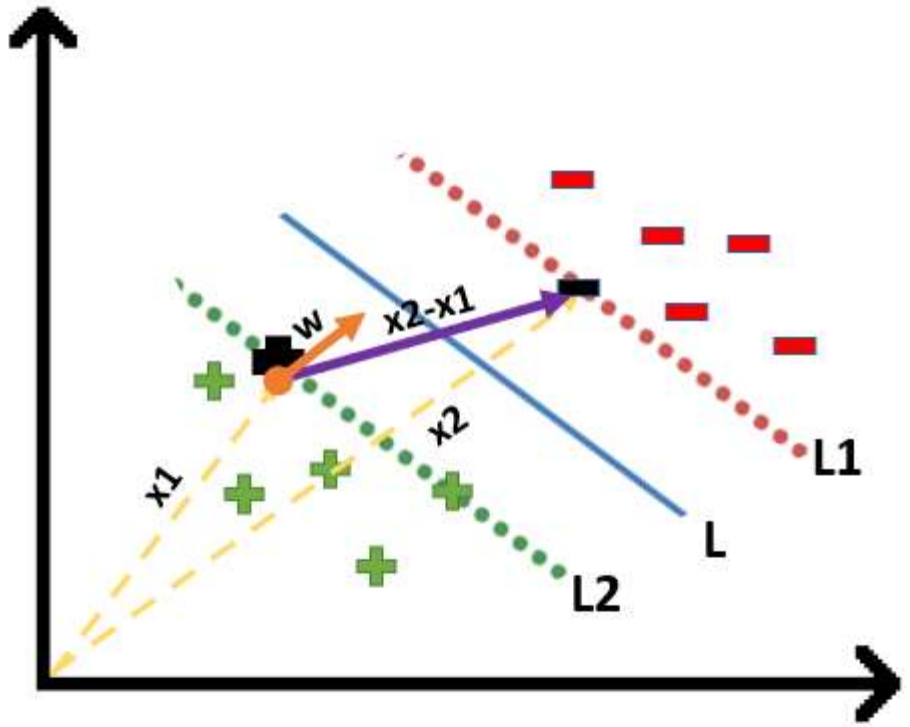




- If the value of w.x+b>0 then we can say it is a positive point otherwise it is a negative point. Now we need (w,b) such that the margin has a maximum distance. Let’s say this distance is ‘d’.



- 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 *(x2-x1) vector*.
- the shortest distance between these two points which can be found using a trick we used in the dot product. We take a vector 'w' perpendicular to the hyperplane and then find the projection of (x2-x1) vector on 'w'.



$$\Rightarrow (x2 - x1) . \frac{\overrightarrow{w}}{\|w\|}$$

$$\Rightarrow \frac{x2.\overrightarrow{w} - x1.\overrightarrow{w}}{\|w\|} \qquad - - - - (1)$$

Since x2 and x1 are support vectors and they lie on the hyperplane, hence they will follow **y_i* (2.x+b)=1** so we can write it as:

$$\text{for positive point } y = 1$$

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

$$\Rightarrow \overrightarrow{w}.x1 = 1 - b \qquad - - - - - (2)$$

$$\text{Similarly for negative point } y = -1$$

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

$$\Rightarrow \overrightarrow{w}.x2 = -b - 1 \qquad - - - - - (3)$$

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

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

Hyperspectral image

- Multispectral Image
- Hyperspectral Image

- **Multispectral vs Hyperspectral Imagery**
  - Multispectral: 3-10 wider bands.
  - Hyperspectral: Hundreds of narrow bands.

Hyperspectral image

- Hyperspectral sensors collect information as a set of ‘images’
- These images are then combined and form a 3-dimensional hyperspectral data cube for processing and analysis
- Hyperspectral imaging does not just measure each pixel in the image but also measures the reflection, emission, and absorption of electromagnetic radiation
- It provides a unique spectral signature for every pixel, which can be used by processing techniques to identify and discriminate materials.

Hyperspectral image

- Advantages?
- Disadvantages?

Point Cloud Data

- Point Cloud Data
- https://geoslam.com/point-clouds/