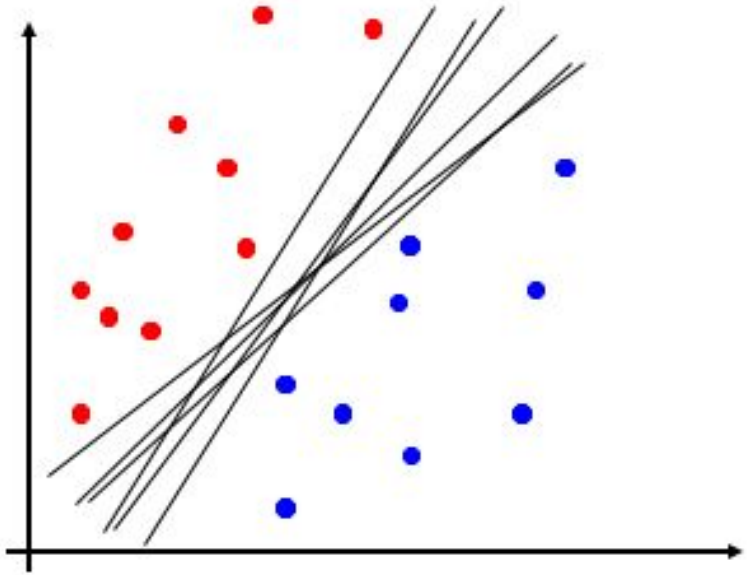
Support Vector Machine

Introduction

- Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems.
- However, primarily, it is used for Classification problems in Machine Learning.
- The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.
- SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

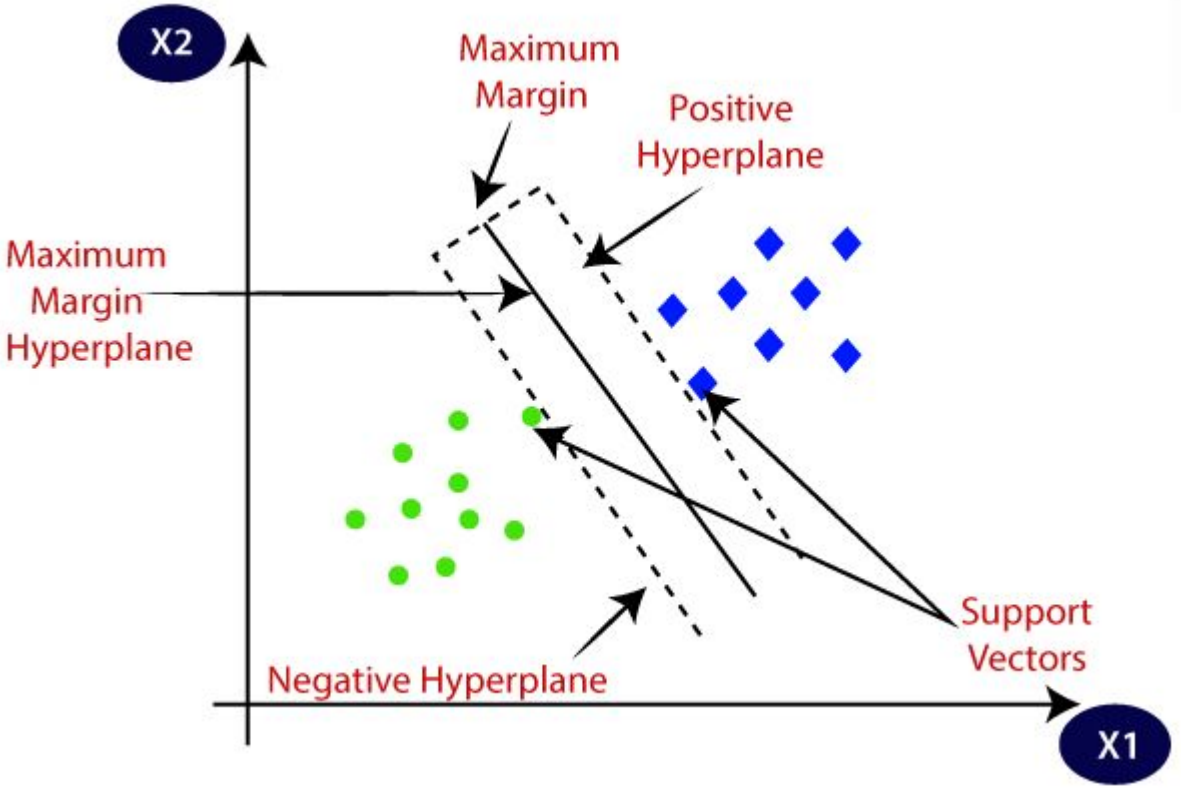
Introduction -Linear Separators

- Which of the linear separators is optimal?

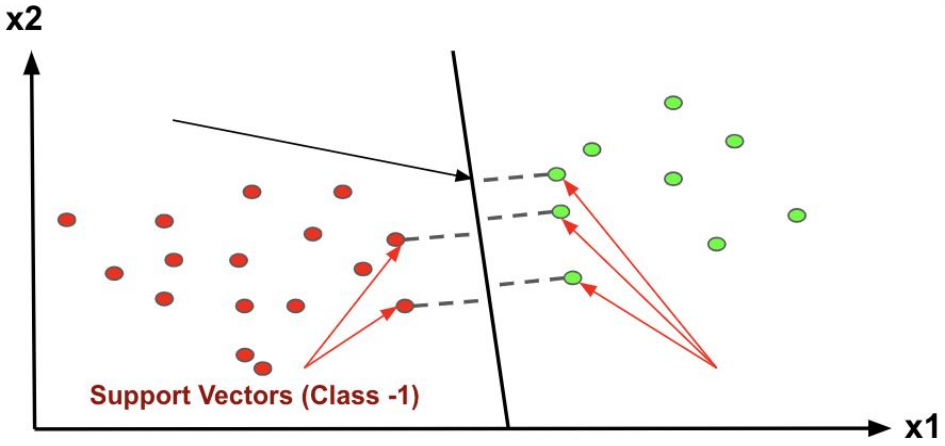


Introduction

Consider this diagram in which there are two different categories that are classified using a decision boundary or hyperplane

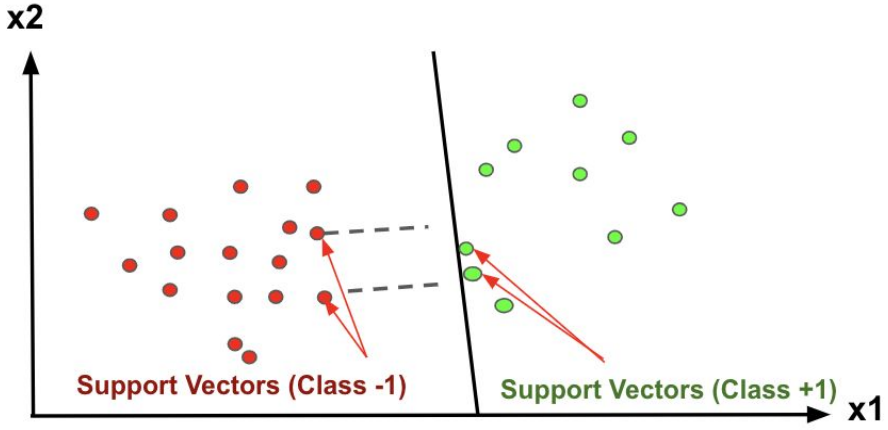


Margin



Good Margin

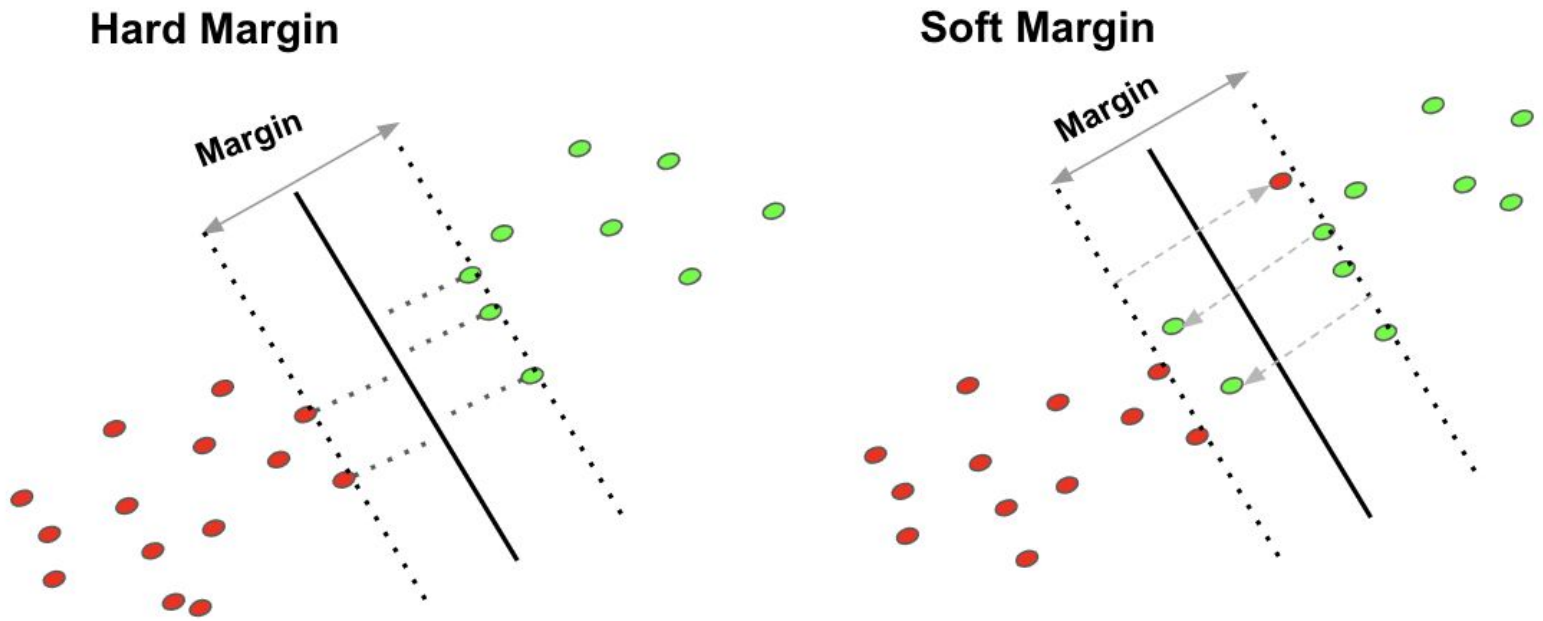
- all sector vectors have the same distance with the maximum margin hyperplane



Bad Margin

- very close to either class -1 support vectors or class +1 support vectors

Hard Margin and Soft Margin



Hard Margin and Soft Margin

Hard Margin

If the training data is linearly separable, we can select two parallel hyperplanes that separate the two classes of data, so that the distance between them is as large as possible.

Soft Margin

As most of the real-world data are not fully linearly separable, we will allow some margin violation to occur, which is called soft margin classification. It is better to have a large margin, even though some constraints are violated. Margin violation means choosing a hyperplane, which can allow some data points to stay in either the incorrect side of the hyperplane and between the margin and the correct side of the hyperplane.

In order to find the **maximal margin**, we need to maximize the margin between the data points and the hyperplane.

Important concepts in SVM

Hyperplane: It is a decision plane or space which is divided between a set of objects having different classes.

There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the **best decision** boundary that helps to classify the data points. This best boundary is known as the **hyperplane** of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in diagram), then hyperplane will be a **straight line**. And if there are 3 features, then hyperplane will be a **2-dimension plane**.

We always create a hyperplane that has a **maximum margin**, which means the maximum distance between the data points.

Important concepts in SVM

Support Vectors −The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.Separating line will be defined with the help of these data points.

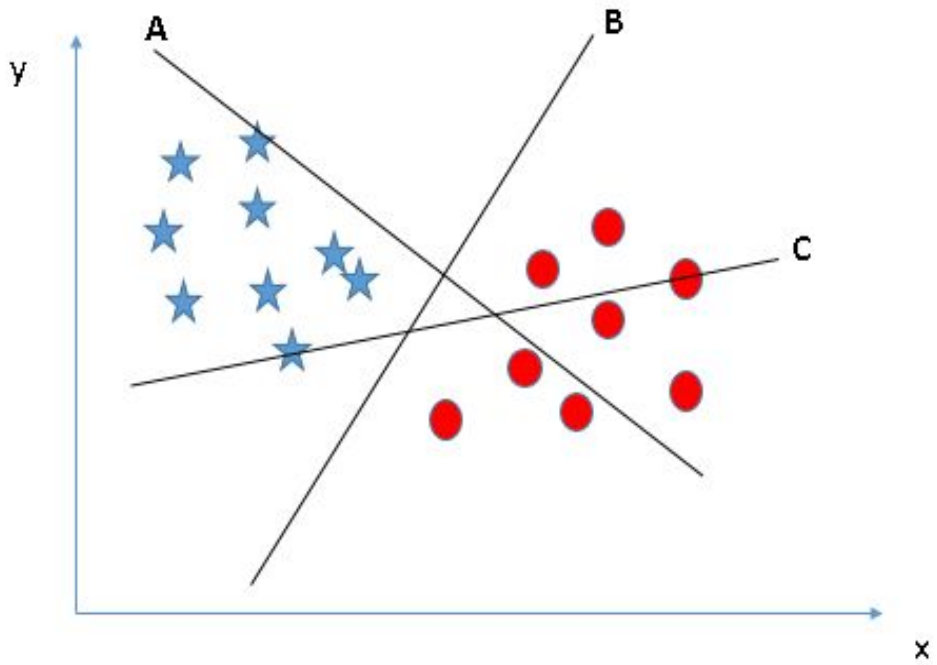
Margin − It may be defined as the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

The separating line with the maximum margin is called the “maximum margin line” or the “optimal separating line”.

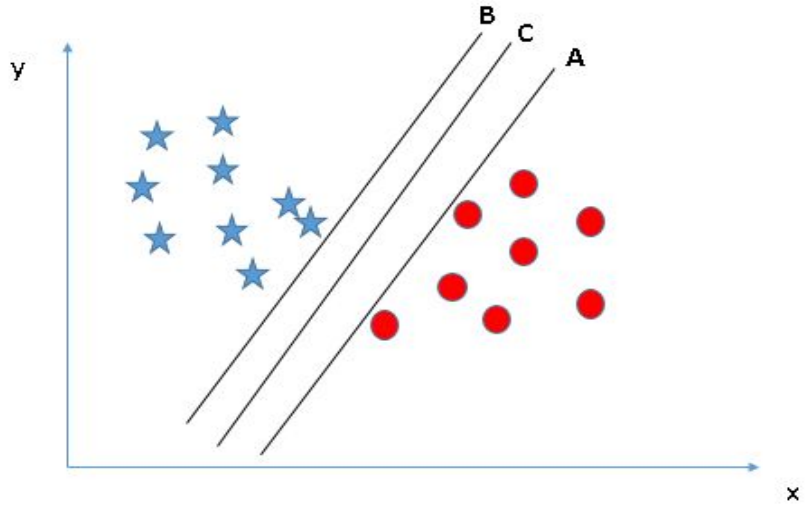
How can we identify the right hyper-plane(Optimal Hyperplane)?”

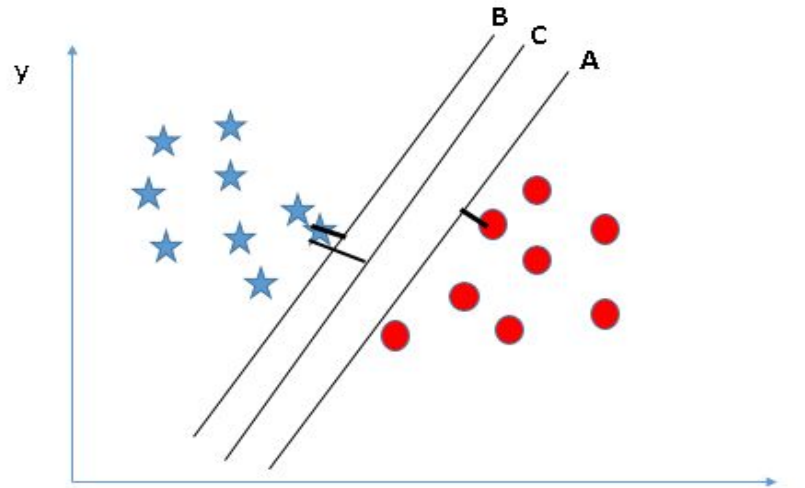
Select the hyper-plane which segregates the two classes better”

hyper-plane “B”



How can we identify the right hyper-plane(Optimal Hyperplane)?”





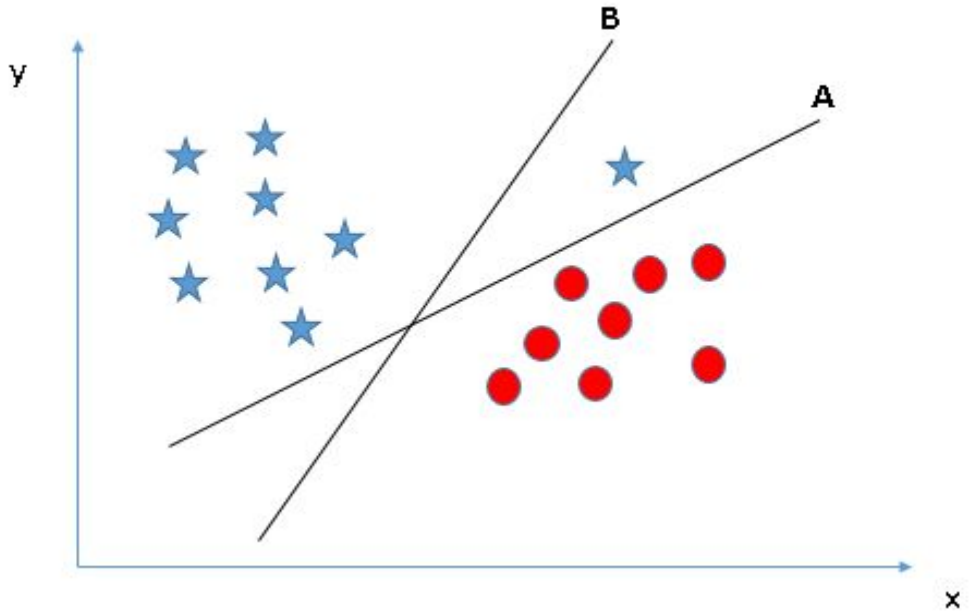
The margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C.

Another reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

How can we identify the right hyper-plane(Optimal Hyperplane)?”

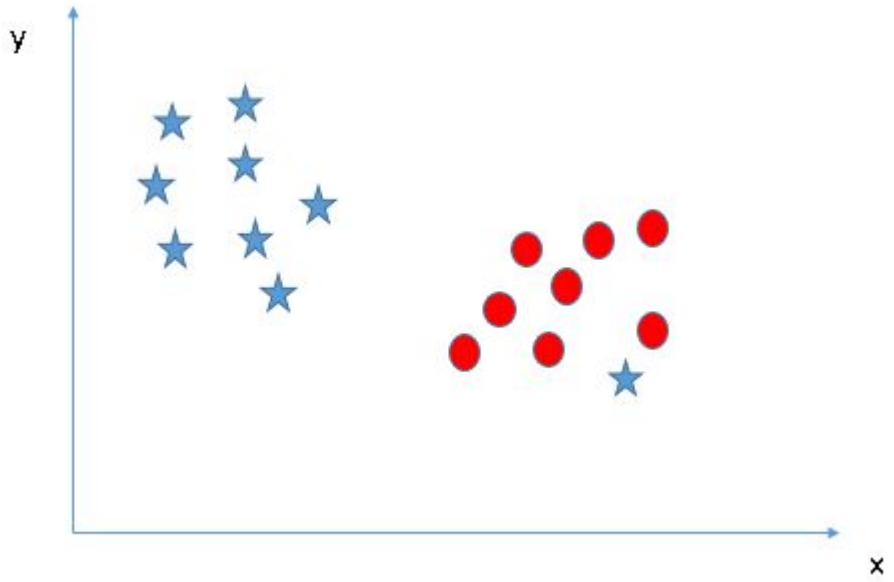
SVM selects the hyperplane which classifies the classes accurately prior to maximizing margin.

Hyperplane B has a classification error and A has classified all correctly. Therefore, the right hyperplane is **A.**

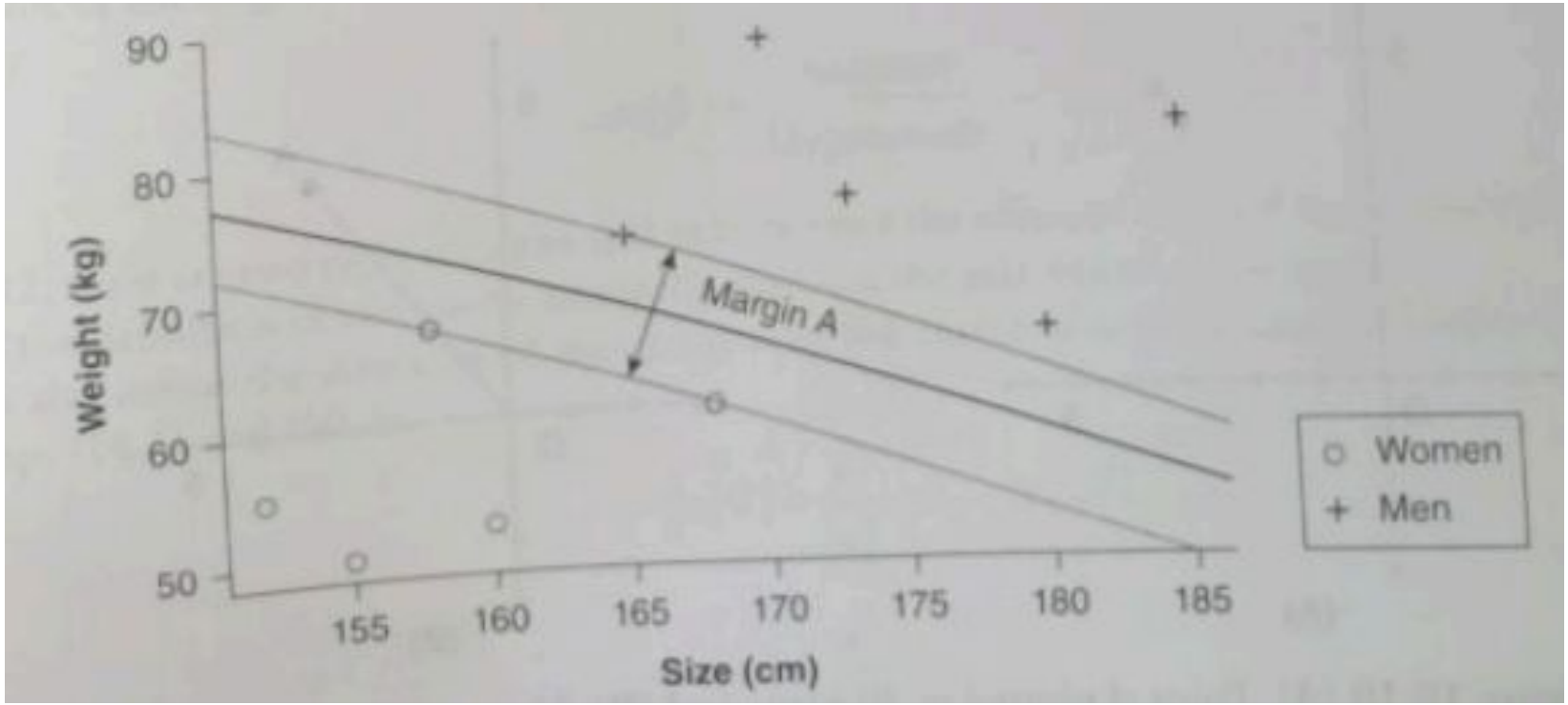


How can we identify the right hyper-plane(Optimal Hyperplane)?”

one star at other end is like an outlier for star class. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.



Relationship between Margin and Optimal Hyperplane



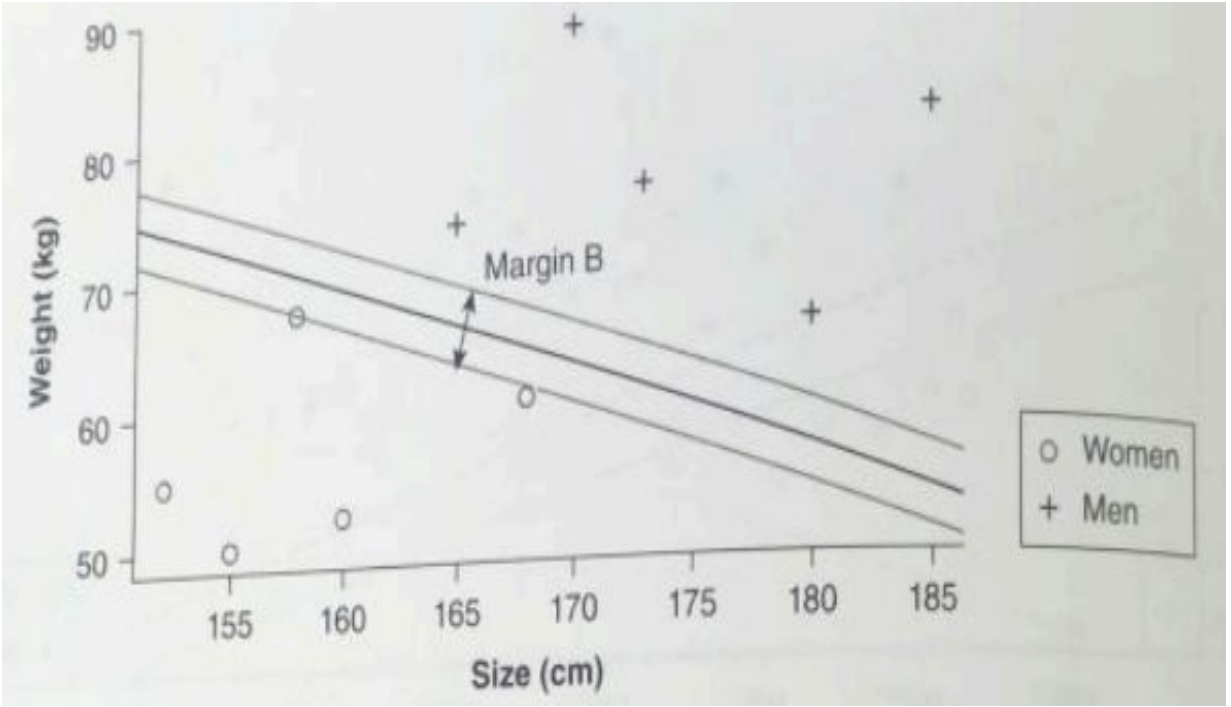
Margin of a Optimal Hyperplane

Relationship between Margin and Optimal Hyperplane

Margin B is smaller than Margin A.

Two inferences:

- If a hyperplane is very close to a datapoint, its margin will be small.
- And if hyperplane is farther away from datapoint, its margin will be larger.



Hyperplane with a narrow margin

A separating hyperplane is given by

$W.X + b = 0$

where, W is the weight vector namely $W= w_1, w_2,.... w_n$, n is the number of attributes and b is the scale often referred to as bias.

If training tuples are 2D with $X =x_1, x_2$ where x_1 and x_2 are the values of attributes A1 and A2 respect for X. And if we consider b as additional weight w_0 then,

$w_0+ w_1x_1+w_2x_2 =0$

Thus, any point that lies above the separating hyperplane satisfies

$w_0+ w_1x_1+w_2x_2>0$

And the point below the separating hyperplane is given as $w_0+ w_1x_1+w_2x_2<0$

Any type that falls on hyperplanes H1 or H2 satisfy the above equations are called support vectors.

The weight can be adjusted so that the hyperplane defining the sides of the margin can be written as

H1: $w_0 + w_1x_1 + w_2x_2 >= 1$ for yi =+1

H2 : $w_0 + w_1x_1 + w_2x_2 <= -1$ for yi=-1

Combining the 1 inequalities we get

$yi(w_0 + w_1x_1 + w_2x_2) >= 1$, for all values of i

Distance Measurement

The distance from the separating hyperplane to any point on H1 is $\frac{1}{\|W\|}$

Where IIWII is the Euclidean norm of W, that is $\sqrt{W.\ W_2}$

This is the distance from any point on H2 to the separating hyperplane.

So, maximal margin is $\frac{2}{\|W\|}$, i.e. Margin is twice the distance to nearest samples

- The task of maximization M is same as compared to minimizing a function L(w) subject to some conditions.
- The conditions used to model the requirements for correct classification of all training samples x_i by the hyperplane is given by ,

  $$\min_{w,b} L(w) = \frac{1}{2} \|W\|^2 \text{ subject to } y_i(w^t x + b) >= 1 \text{ for all i.}$$
- This problem is solved by lagranges multiplier to calculate weight vector w and bias b of the optimal hyperplane

Based on lagranges formulation the MMH can be rewritten as the decision boundary

$d(X_T) = y_i \alpha_i X_i X_T + b_0$

Where y_i is the class label of support vector X_i

X_T is the test tuple

α_i and b_0 are numeric parameters that are determined automatically by SVM algorithm

l is the number if support vectors

Quadratic Programming solution to find Maximum Margin separator

A hyperplane is defined as

$F(x) = w^t x + b$

where, w is the weight and b is the bias

Distance = $\frac{|\, w^t x + b \,|}{\|W\|}$

$\frac{1}{\|W\|}$

Where, the numerator is equal to 1 for the hyperplane that is being satisfied or considered appropriate .

Types of SVM

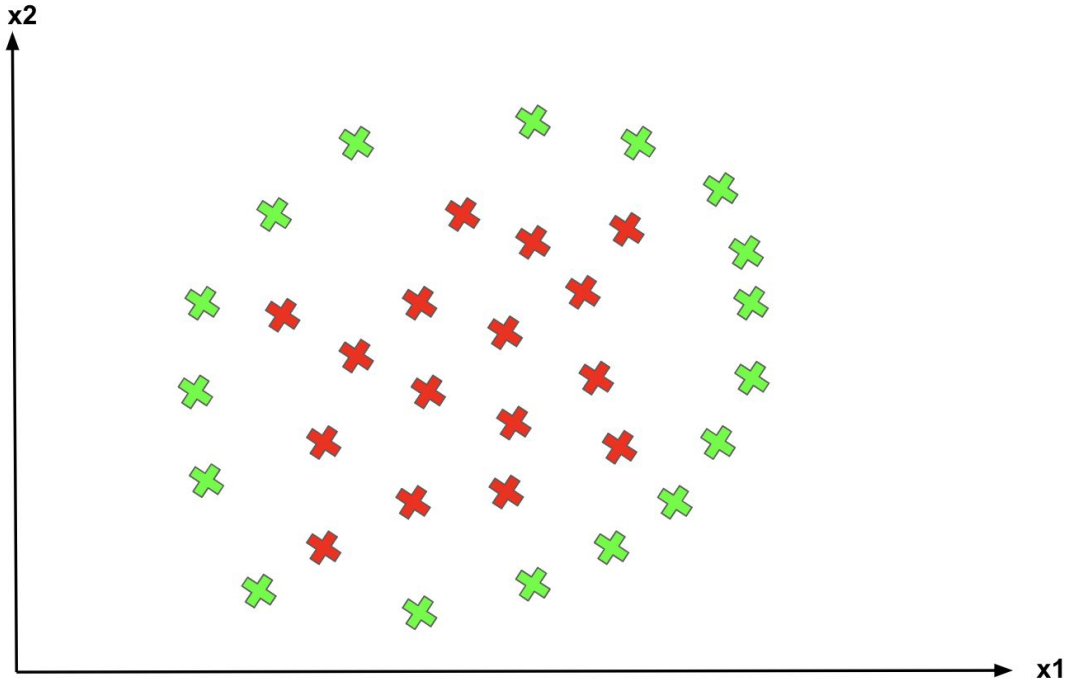
SVM can be of two types:

- **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/ Kernel:** 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.

Classifying non-linear data

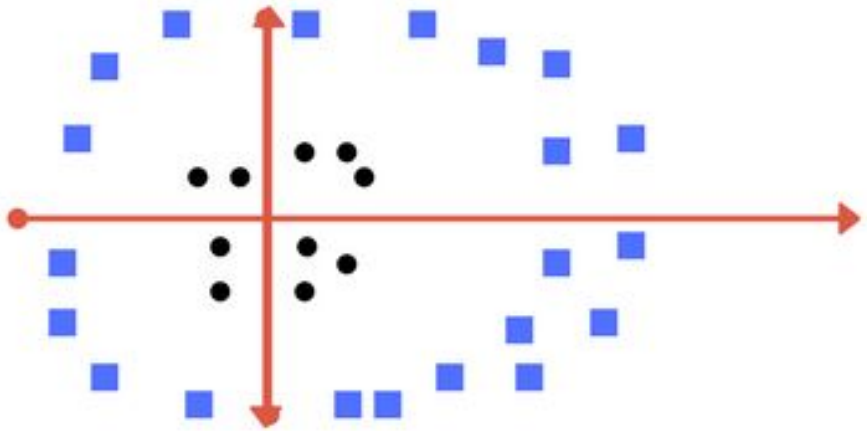
What about data points are not linearly separable?



Non-linear separate data points.

Classifying non-linear data

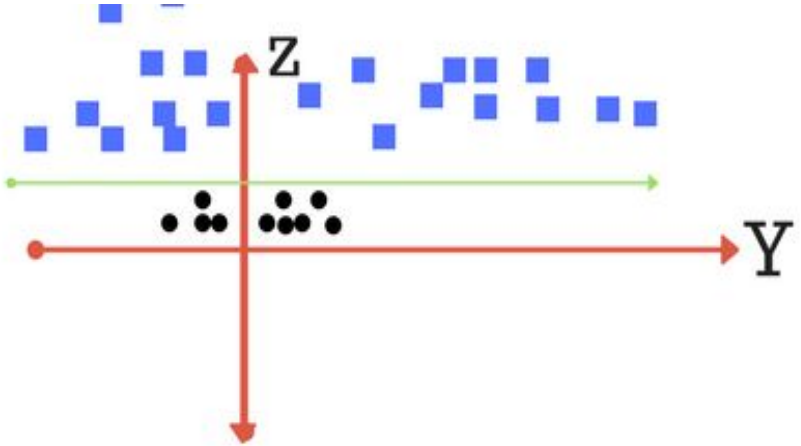
As shown in figure there is no line that can separate the two classes in this x-y plane.
We apply transformation and add one more dimension as we call it z-axis.
Let's assume value of points on z plane, $w = x^2 + y^2$.



Classifying non-linear data

In this case we can manipulate it as distance of point from z-origin.

Now if we plot in z-axis, a clear separation is visible and a line can be drawn .

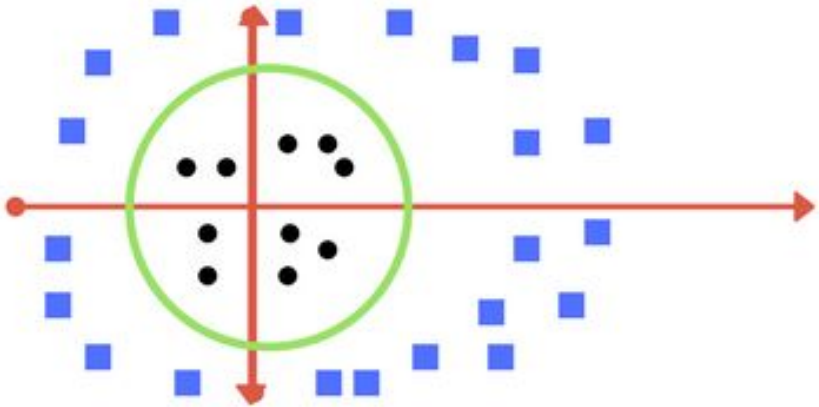


plot of zy axis. A separation can be made here

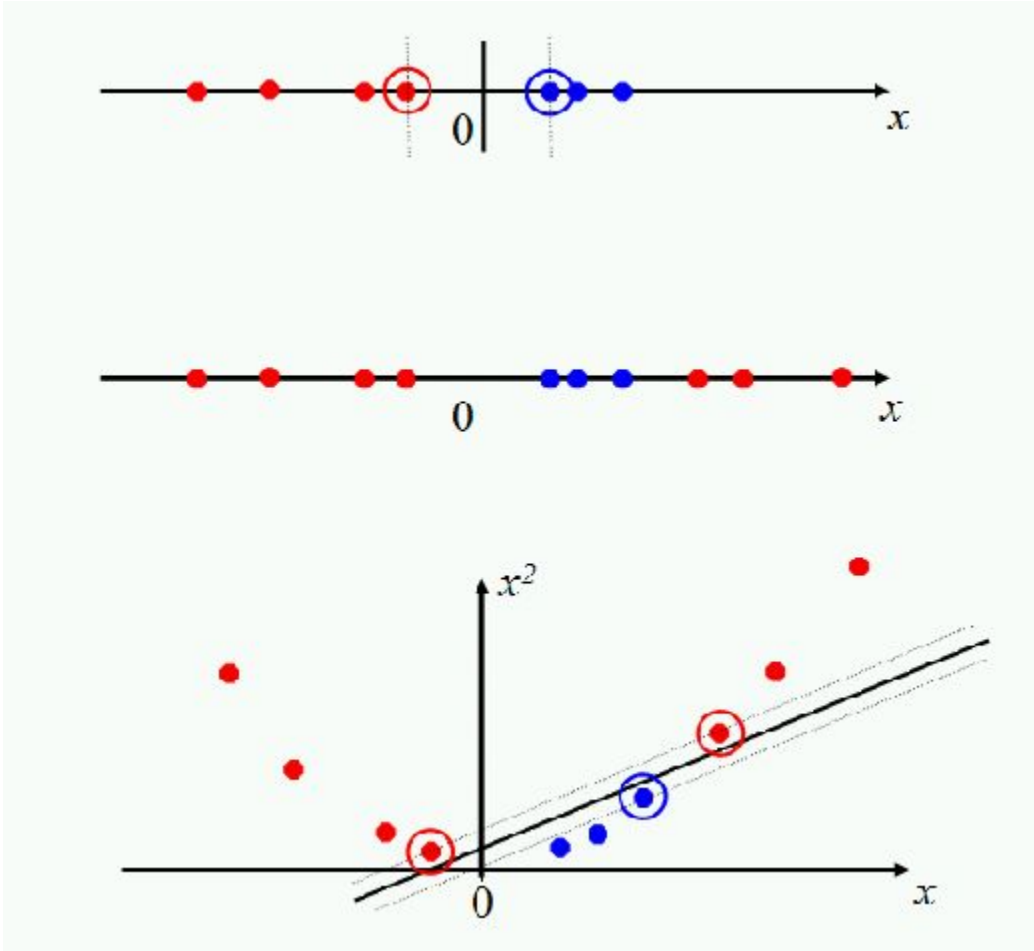
Classifying non-linear data

When we transform back this line to original plane, it maps to circular boundary as shown in *image E*.

These transformations are called ***kernels.***



Transforming back to x-y plane, a line transforms to circle.


0
x
0
x
x²
0
x

Classifying non-linear data

SVM has a technique called the **kernel trick**.

These are functions that take low dimensional input space and transform it into a higher-dimensional space, i.e., it converts not separable problem to separable problem.

It is mostly useful in non-linear separation problems.

Classifying non-linear data

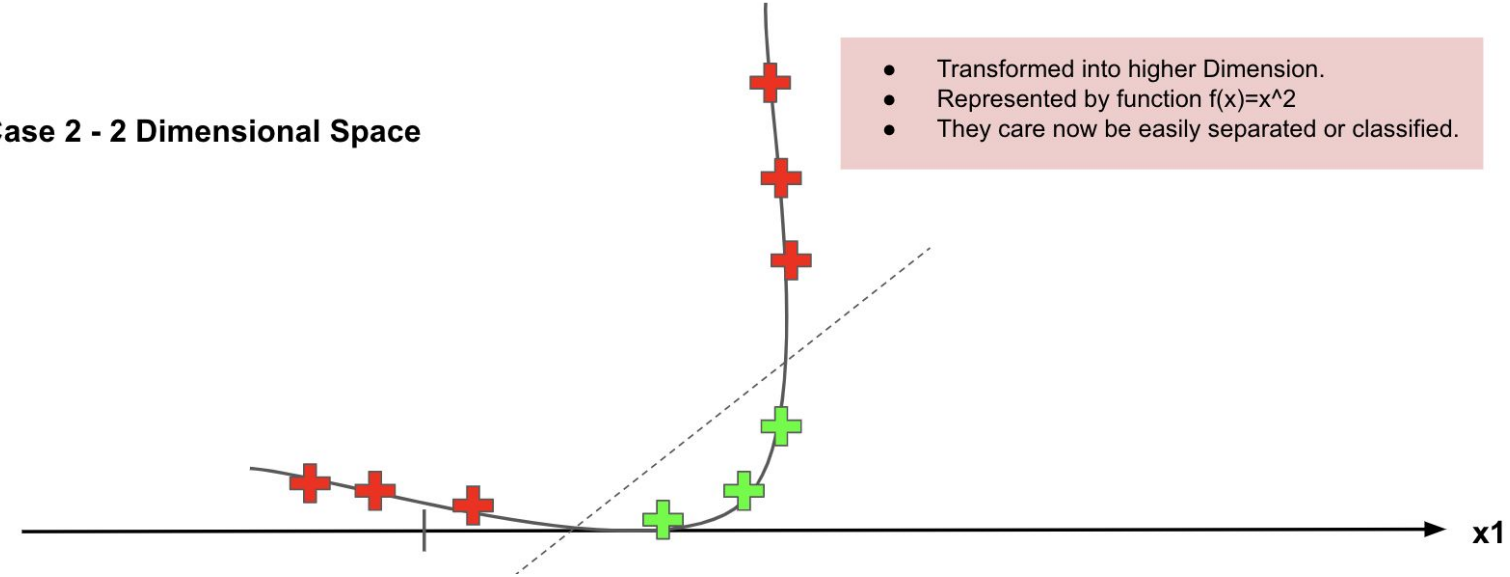
Mapping to a Higher Dimension

Case 1 - 1 Dimensional Space

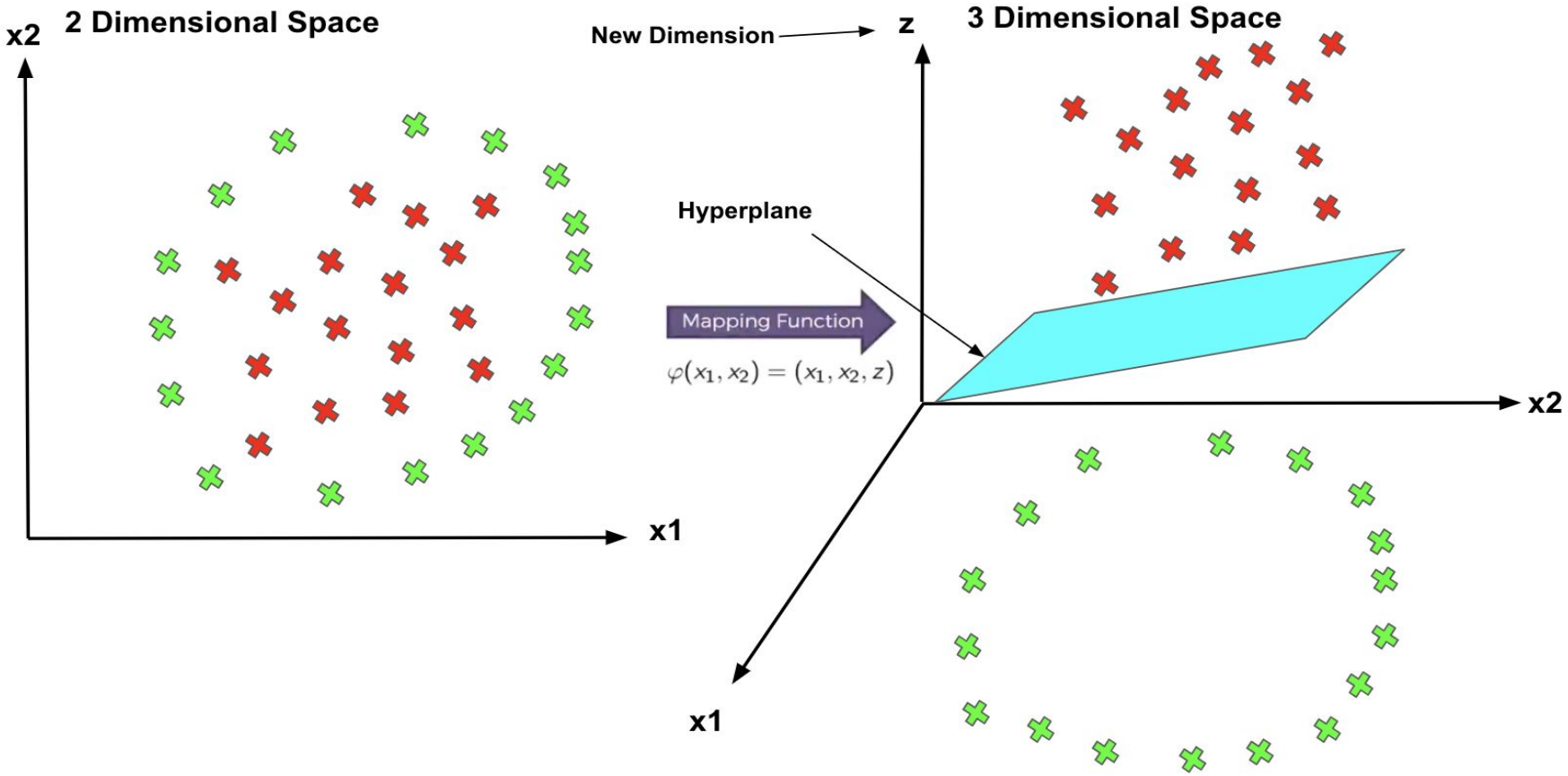
- Points in 1 Dimension Plan.
- Represented by function f(x)=x
- They cannot be separated or classified.

Case 2 - 2 Dimensional Space

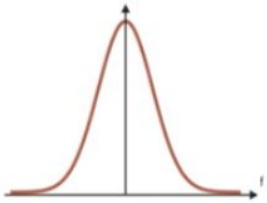
- Transformed into higher Dimension.
- Represented by function f(x)=x^2
- They care now be easily separated or classified.



Classifying non-linear data

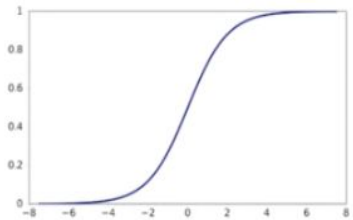


Some Frequently Used Kernels



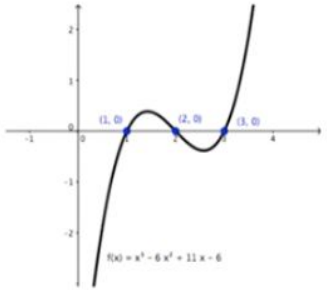
Gaussian RBF Kernel

$$K\left(\vec{x}, \vec{l}^{i}\right) = e^{-\frac{\left\|\vec{x}-\vec{l}^{i}\right\|^{2}}{2\sigma^{2}}}$$



Sigmoid Kernel

$$K(X, Y) = \tanh(\gamma \cdot X^{T}Y + r)$$



Polynomial Kernel

$$K(X, Y) = (\gamma \cdot X^{T}Y + r)^{d}, \gamma > 0$$

Regularization parameter and **gamma.**

Which line among image 1 and 2 should considered?

Both are correct.

The first one tolerates some outlier points.

The second one is trying to achieve 0 tolerance with perfect partition.

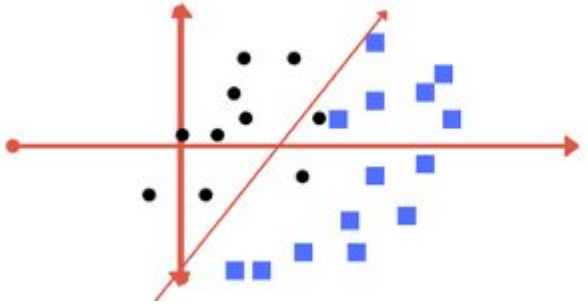


Image 1

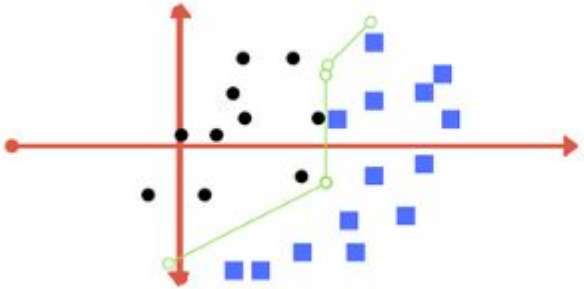


Image 2

Regularization parameter and **gamma.**

But, there is trade off.

In real world application, finding perfect class for millions of training data set takes lot of time.

we define two terms **regularization parameter** and **gamma.** These are tuning parameters in SVM classifier.

Varying those we can achieve considerable non linear classification line with more accuracy in reasonable amount of time.

Tuning parameters: Regularization

The Regularization parameter (often termed as C parameter) tells the SVM optimization how much you want to avoid misclassifying each training example.

For **large** values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly.

Conversely, a very **small** value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

Tuning parameters: Regularization

Left one has some misclassification due to lower regularization value. Higher value leads to results like right one.

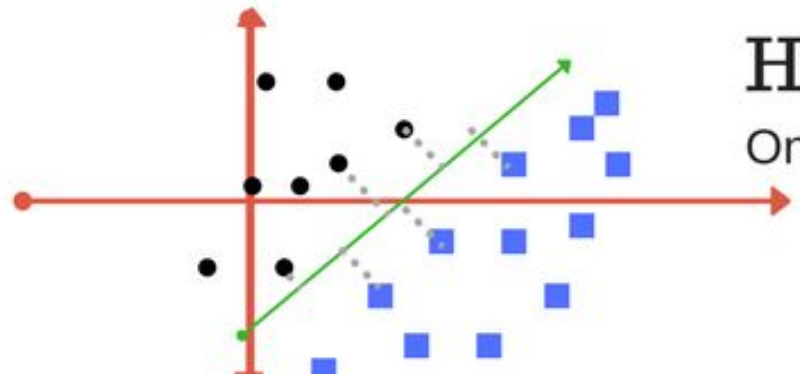
Left: low regularization value, right: high regularization value

Gamma

The gamma parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’.

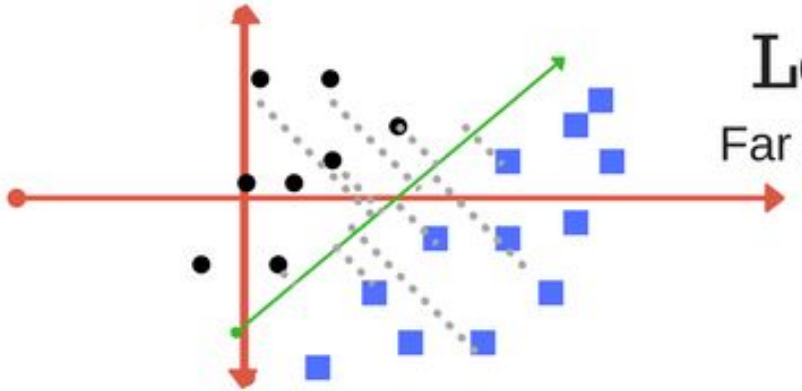
In other words, with **low** gamma, points far away from plausible separation line are considered in calculation for the separation line. Where as **high** gamma means the points close to plausible line are considered in calculation.

Gamma



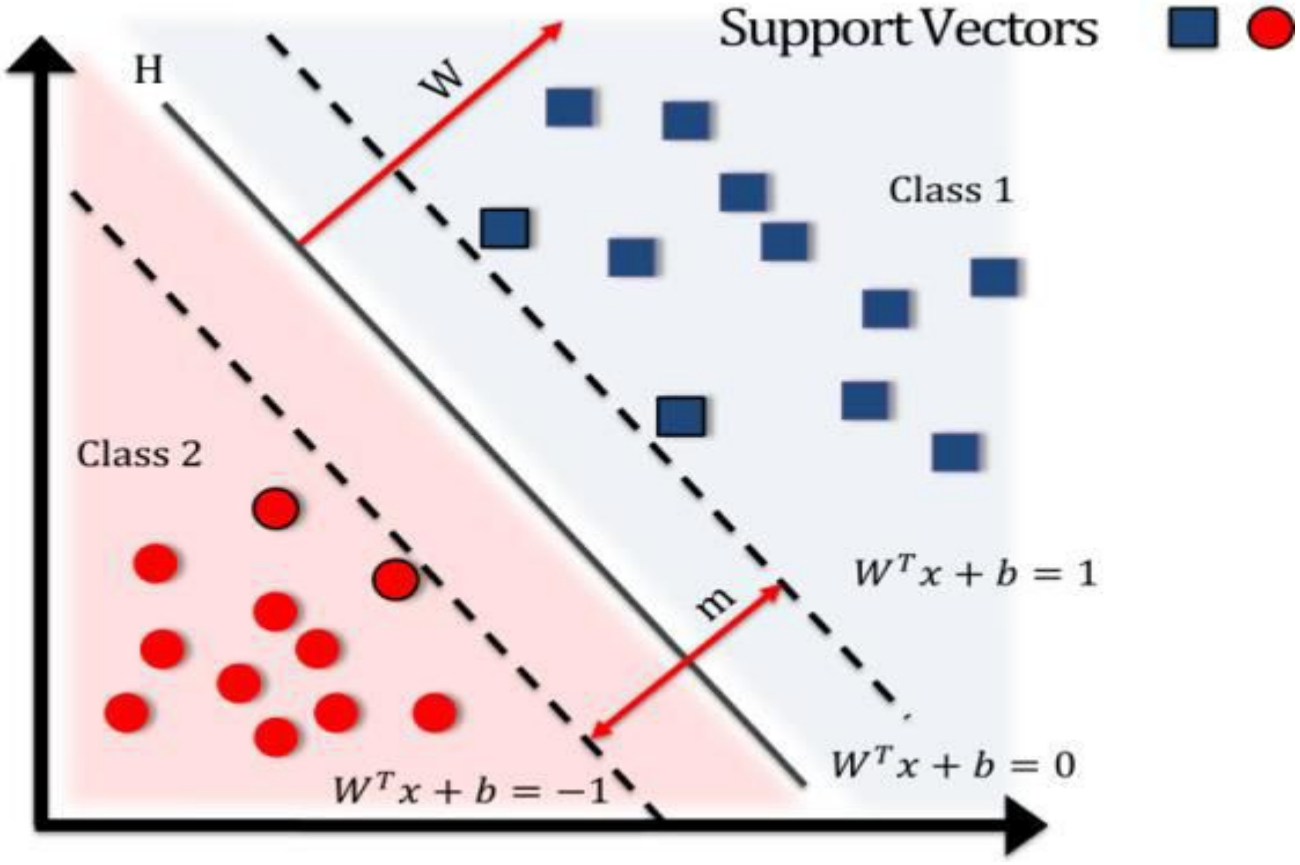
High Gamma

Only nearby points are considered.



Low Gamma

Far away points are also considered.


Support Vectors
H
W
Class 1
Class 2
$W^T x + b = 1$
m
$W^T x + b = 0$
$W^T x + b = -1$

Why SVM?

- SVMs are used in applications like handwriting recognition, intrusion detection, face detection, email classification, gene classification, and in web pages. This is one of the reasons we use SVMs in machine learning. It can handle both classification and regression on linear and non-linear data.

- Another reason we use SVMs is because they can find complex relationships between your data without you needing to do a lot of transformations on your own. It's a great option when you are working with smaller datasets that have tens to hundreds of thousands of features. They typically find more accurate results when compared to other algorithms because of their ability to handle small, complex datasets.

SVM: Pros and Cons

Pros

- Effective on datasets with multiple features, like financial or medical data.
- Effective in cases where number of features is greater than the number of data points.
- Uses a subset of training points in the decision function called support vectors which makes it memory efficient.
- Different kernel functions can be specified for the decision function. You can use common kernels, but it's also possible to specify custom kernels.

Cons

- If the number of features is a lot bigger than the number of data points, avoiding over-fitting when choosing kernel functions and regularization term is crucial.
- SVMs don't directly provide probability estimates. Those are calculated using an expensive five-fold cross-validation.
- Works best on small sample sets because of its high training time.

Kernel functions

Linear

These are commonly recommended for text classification because most of these types of classification problems are linearly separable.

The linear kernel works really well when there are a lot of features, and text classification problems have a lot of features. Linear kernel functions are faster than most of the others and you have fewer parameters to optimize.

Here's the function that defines the linear kernel:

$$W^T x + b = f(x)$$

In this equation, **w** is the weight vector that you want to minimize, **X** is the data that you're trying to classify, and **b** is the linear coefficient estimated from the training data. This equation defines the decision boundary that the SVM returns.

Polynomial

The polynomial kernel isn't used in practice very often because it isn't as computationally efficient as other kernels and its predictions aren't as accurate.

Here's the function for a polynomial kernel:

```
f(X1, X2) = (a + X1^T * X2) ^ b
```

This is one of the more simple polynomial kernel equations you can use. **f(X1, X2)** represents the polynomial decision boundary that will separate your data. **X1** and **X2** represent your data.

Gaussian Radial Basis Function (RBF)

One of the most powerful and commonly used kernels in SVMs. Usually the choice for non-linear data.

Here's the equation for an RBF kernel:

```
f(X1, X2) = exp(-gamma * ||X1 - X2||^2)
```

In this equation, **gamma** specifies how much a single training point has on the other data points around it. **||X1 - X2||** is the dot product between your features.

Sigmoid

More useful in neural networks than in support vector machines, but there are occasional specific use cases.

Here's the function for a sigmoid kernel:

```
f(X, y) = tanh(alpha * X^T * y + C)
```

In this function, **alpha** is a weight vector and **C** is an offset value to account for some mis-classification of data that can happen.

SVM as constrained Optimization Problem

- Constrained Optimization Problem are the problems for which a function f(x) has to be minimized or maximized subject to the given constraints ø(x)
- Here $f:R^n \longrightarrow R$ is called the objective function and ø(x) is a Boolean valued formula
- The constraints ø(x) is an arbitrary combination of equations
  g(x) = 0, Weak inequalities g(x) >=0, strict inequalities g(x) >0 and x Ɛ· Z statements
- Notations used are

  Min.f(x)

  S.t ø(x)

  This states, “Minimize f(x) subject to constraints ø(x)”

  Max.f(x)

  S.t ø(x)

  This states, “Maximize f(x) subject to constraints ø(x)”
- A point u Ɛ· $R^{n,}$ satisfies the constraints ø if ø(u)is true

SVM as constrained Optimization Problem : Global Optimization

- A point u Ɛ· $R^{n,}$ is said to be a global minimum of f subject to the constraints ø if u satisfies the constraint and for any point v that satisfies the constraints .

  f(u) <= f(v)
- A value a Ɛ· R is said to be a global minimum of f subject to the constraints and for any point v that satisfies the constraints, a <= f(v)
- The global minimum value a exists for any f and ø . The global minimum value is attained if there exists a point u such that ø(u) is true and f(u) = a
- If f is a continuous function and the set of points satisfying the constraints ø is compact(ie. Closed and bounded) and non empty, then a global minimum exists. Otherwise, global minimum may not exist.

SVM as constrained Optimization Problem : Local Optimization

- A point u Ɛ· $R^{n,}$ is said to be a **local minimum** of f subject to the constraints ø if u satisfies the constraint and for some r>0, if v satisfies Iv-uI < r ^ ø (v), then f(u) <= f(v)
- A local minimum may not be a global minimum
- A global minimum is always a local minimum .

Solving Optimization Problems

- Depending on the type of functions involved , they are divided into Linear and NonLinear optimization problem
- Functions for constrained optimization include Minimize, maximize , N minimize and N maximise for global constrained optimization .
- **Find minimum** for local constrained optimization and Linear Optimization for efficient and direct access to linear optimization methods.

Quadratic Programming or Quadratic Optimization

- It is a method of solving the **mathematical optimization problems** involving quadratic functions .
- Aim is to minimize or maximise a multivariate quadratic function subject to linear constraints on the variables.
- It refers to non linear programming, programming indicates a formal procedure for solving mathematical problem.

Kernel Trick

- If the problem is non linear, instead of trying to fit a non linear model , we can map the problem into a new space by doing a non linear transformation using suitable chosen basis function and then use a linear model in this new space.
- This linear model in the new space corresponds to a nonlinear model in the original space.
- This approach can be used for both classification and Regression problems.

SVM for Linear and Non Linear Classification

Support Vector Regression

- Regression analysis is a statistical procedure used to identify the degree and type of a connection between one dependent variable and a set of other factors.
- Aim of linear regression models is to reduce the sum of squared errors.
- A linear regression is represented by : $y = mx+c$ and
  Polynomial regression : $y= mx^n+c$

  Both do not provide optimal line ,

  So, in order to determine the optimum regression equation , SV regression is used.

- SV regression chooses the best fitting model equation mathematically and applies it to the model .

Support Vector Regression

- **A margin of tolerance (ε= epsilon)** is supplied in case of regression as an approximate estimate to SVM.
- Aim is to reduce the errors by customising the hyperplane to maximize the margin.
- Introduction of a insensitive zone around the function , known as the **tube ,** allows SVM to be a generalized to SVR
- This tube reformulates the optimization problem to find the tube that best approximates the continuous-valued function, while balancing model complexity and prediction error.
- More specifically, SVR is formulated as an optimization problem by :-
  - First defining a convex *ε*-insensitive loss function to be minimized and
  - Finding the flattest tube that contains most of the training instances.

  Hence, a multiobjective function is constructed from the loss function and the geometrical properties of the tube.

SVR problem formulation

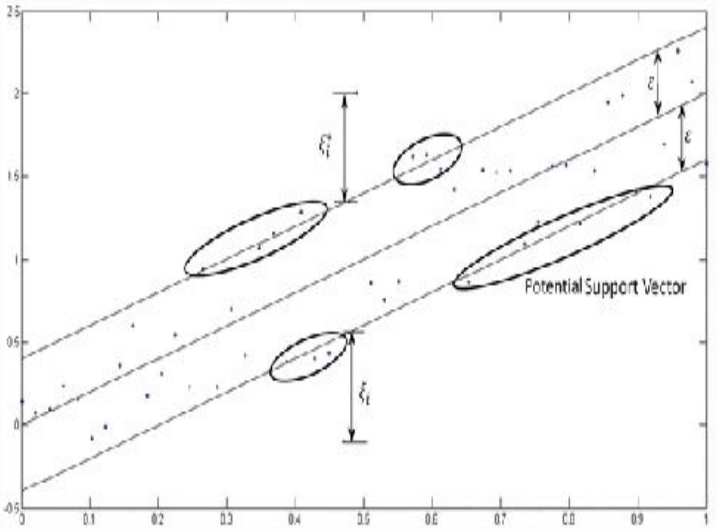
SVR problem formulation is often best derived from a one-dimensional geometrical perspective,as represented in Figure

The continuous-valued function being approximated can be written as

$$y = f(x) = \langle w, x \rangle + b = \sum_{j=1}^{M} w_j x_j + b,\ y, b \in \mathbb{R}, x, w \in \mathbb{R}^M$$

$$f(x) = \begin{bmatrix} w \\ b \end{bmatrix}^T \begin{bmatrix} x \\ 1 \end{bmatrix} = w^T x + b \quad x, w \in \mathbb{R}^{M+1}$$

For multidimensional data, you augment *x* by one and include *b* in the *w* vector to simply the mathematical notation, and obtain the multivariate regression in Equation 2.



SVR problem formulation

- SVR attempts to find the narrowest tube centered around the surface, while minimizing the prediction error, that is, the distance between the predicted and the desired outputs.
- This is given by the objective function

$$\min_w \frac{1}{2}\|w\|^2 .$$

where $||w||$ is the magnitude of the normal vector to the surface that is being approximated:

https://link.springer.com/chapter/10.1007/978-1-4302-5990-9_4#Fig1

Multiclass Classification