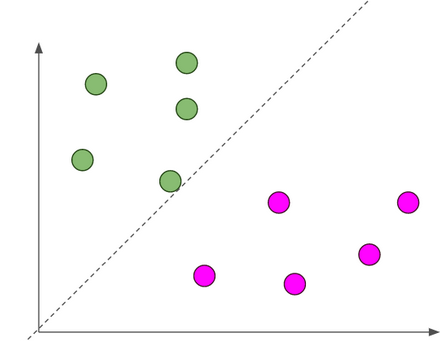
**SVM (Support Vector Machines)**

**Largange Multiplier, Primal and Dual Concept**

SVM is one of the most popular, versatile supervised machine learning algorithm. It is used for both classification and regression task. But, in this thread we will talk about classification task. It is usually preferred for medium and small sized data-set.

The main objective of SVM is to find the optimal hyperplane which linearly separates the data points in two components by maximizing the margin.

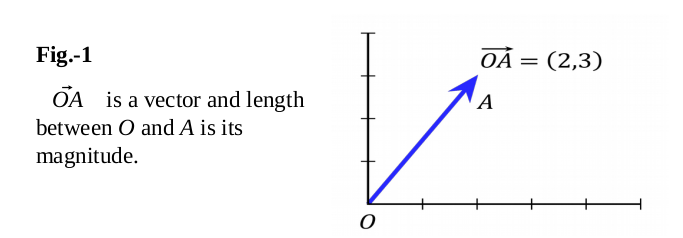


Dotted line is hyperplane, separating blue and pink classes circles.

## Basic Linear Algebra

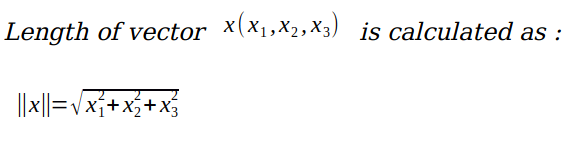
## Vectors

Vectors are mathematical quantity which has both magnitude and direction. A point in the 2D plane can be represented as a vector between origin and the point.

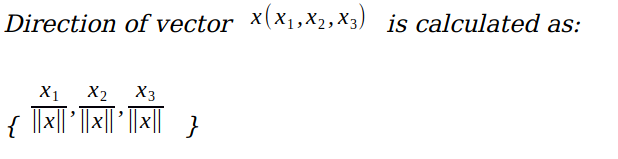


## Length of Vectors

Length of vectors are also called as norms. It tells how far vectors are from the origin.

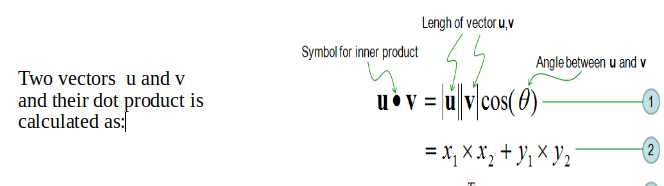


## Direction of vectors



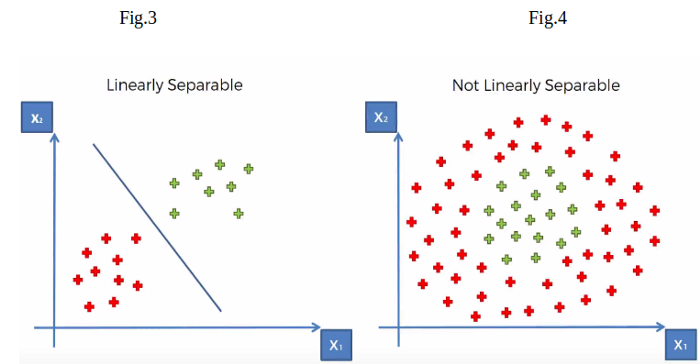
## ****Dot Product****

Dot product between two vectors is a scalar quantity. It tells how to vectors are related.

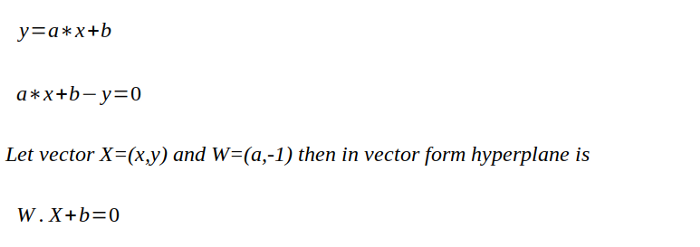


## Hyper-plane

It is plane that linearly divides the n-dimensional data points in two components. In case of 2D, hyperplane is line, in case of 3D it is plane. It is also called as n-dimensional line. Fig.3 shows, a blue line(hyperplane) linearly separates the data point in two components.

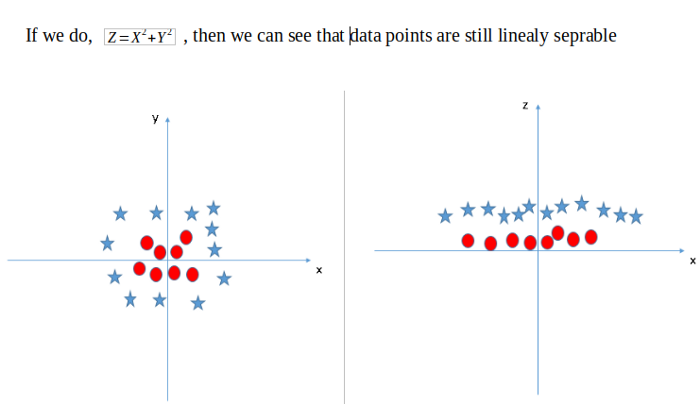


In the Fig.3, hyperplane is line divides data point into two classes (red & green), written as



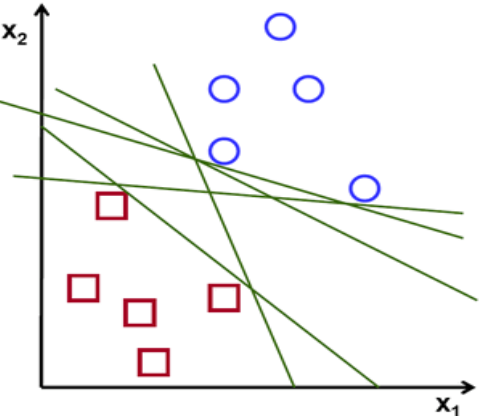
## What if data points are not linearly separable?

Look Fig.4 how can we separate the data-points linearly? This type of situation comes very often in machine learning world as raw data are always non-linear here. So, Is it do-able? yes!!. we will add one extra dimension to the data points to make it separable.



So, the above process of making non-linearly separable data point to linearly separable data point is also known as **Kernel Trick**.

## Optimal Hyperplane



If you look above **Fig** there are numbers of hyperplane that can separate the data points in two components*.* **So optimal hyperplane is one which divides the data points very well***.*

**Why it is needed to choose optimal hyper plane?**

So, if you choose sub-optimal hyperplane, possibly after number of training iteration, training error will decrease but during testing when an unseen instance(test data) will come, it will result in high test error. In that case it is must to choose an optimal plane to get good accuracy.

## How to choose Optimal Hyperplane?

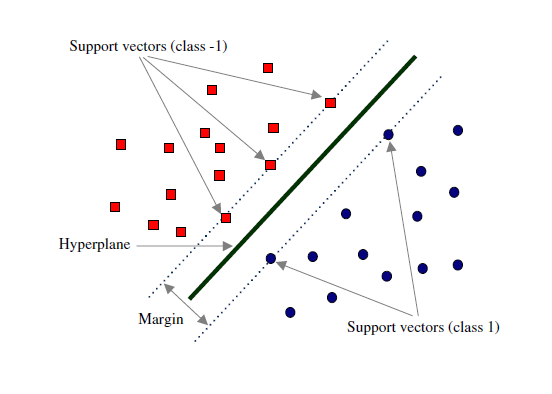


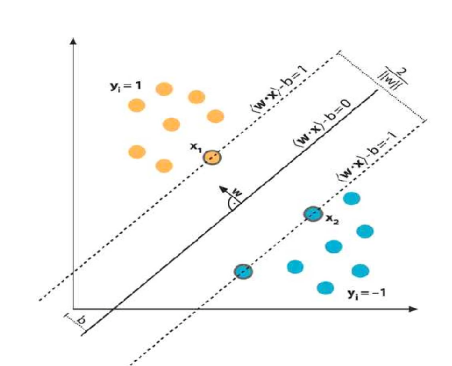
Fig.7 **Margin and Support Vectors**

Let’s assume that solid black line in above Fig.7 is optimal hyperplane and two dotted lines are some hyperplanes, which are passing through nearest data points to the optimal hyperplane. Then distance between hyperplane and optimal hyperplane is known as **margin**, and the closest data-points are known as **support vectors**. Margin is an area which should not contains any data points. There will be some cases when we have data points in margin area but right now, we stick to margin as no data points lands.

So, when we are choosing the optimal hyperplane, we will choose the one among set of hyperplanes which is at highest distance from the closest data points. If optimal hyperplane is very close to the 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.**

**SVM Objective/Intuition - In SVM our goal is to choose an optimal hyperplane which maximizes the margin.**

## Mathematical Interpretation of Optimal Hyperplane



we have L training examples where each example x is of D dimension and each have labels of either **y=+1or y= -1** class, and our examples are linearly separable. Then, our training data is form,



We consider **D=2** to keep explanation simple and data points are linearly separable, The hyperplane **w.x+b = 0** can be described as:

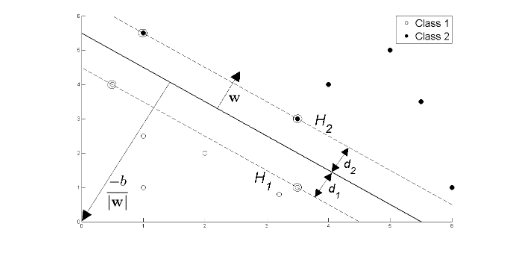
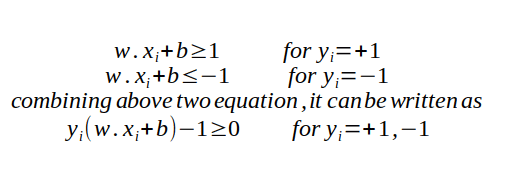


Fig.8

Support vectors examples are closest to optimal hyperplane and the aim of the SVM is to orientate this hyperplane as far as possible from the closest member of the both classes.

From the above Fig , **SVM problem can be formulated as,**



From the Fig.8 we have two hyperplane H1 and H2 passing through the support vectors of +1 and -1 class respectively. so

**H1: w.x+b = -1**

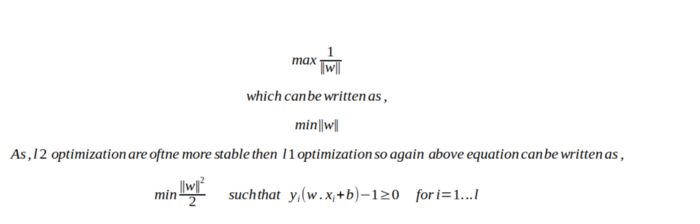
**H2: w.x+b = 1**

And distance between H1 hyperplane and origin is (-1-b)/|w| and distance between H2 hyperplane and origin is (1–b)/|w|. So, margin can be given as

**M = (1-b)/|w|-(-1-b)/|w|**

**M = 2/|w|**

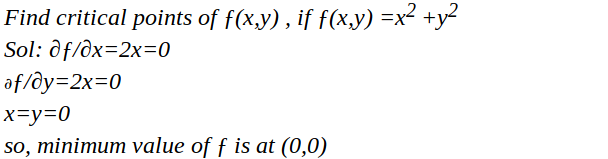
Where M is nothing but twice of the margin. So margin can be written as **1/|w|**. As, **Optimal hyperplane maximize the margin, then the SVM objective is boiled down to fact of maximizing the term 1/|w|,**



## Basic optimization algorithm terms

## Unconstrained optimization

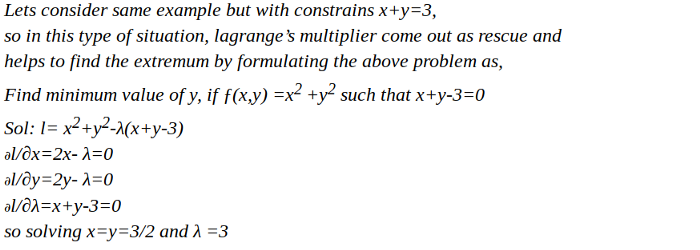
Example will be more intuitive in explaining the concept,



So here we have used Calculus for finding maxima and minima of a function. Only difference is at that time we were calculating for univariate variable, but now we are calculating for multivariate variables.

## Constrained optimization

Again it will become clear with an example,



So basically, we calculate the maxima and minima of a function by taking into the consideration of given constraints on the variables.

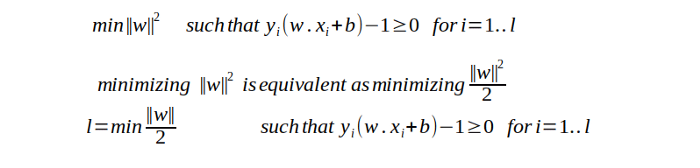
## Primal and Dual Concept

Any optimization problem can be formulated into two way, primal and dual problem. First, we use primal formulation for optimization algorithm, but if it does not yield any solution, we go for dual optimization formulation, which is guaranteed to yield solution.

## ****Optimization****

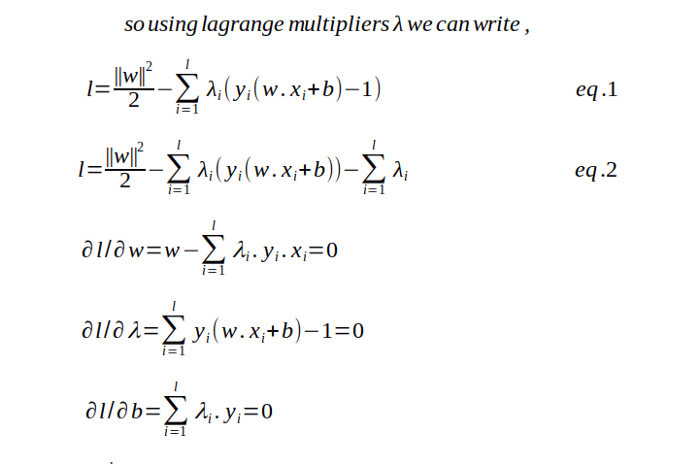
Since it is constrained optimization problem **Lagrange multipliers** are used to solve it, which is described below. It looks like, will be more mathematical but it is not, it is just few steps of finding gradient. We will divide the complete formulation into three parts.

1. In first we will formulate SVM optimization problem mathematically
2. we will find gradient with respect to learning parameters.
3. we will find the value of parameters which minimizes ||w||



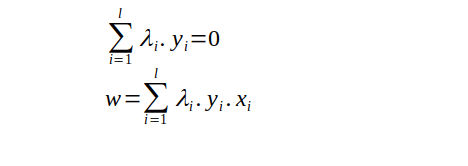
**PART I — Problem formulation**

The above equation is **Primal optimization problem**. **Lagrange method** is required to convert constrained optimization problem into unconstrained optimization problem. The goal of above equation to get the optimal value for w and b.

­

**PART II - Finding the gradient with respect to w ,b and lambda.**

May be above equation is looking tricky? but it is not, its just high school math of finding minima with respect to variable.



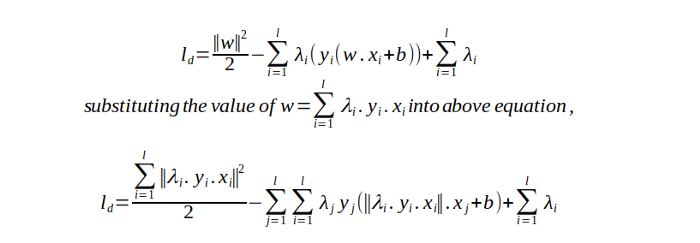
**PART III - We will get the value of w**

As, from the above formulation we are only able to find the optimal value of w and that is too dependent on λ, so we need to find the optimal value of λ also. And finding optimal value of b needs both w and λ. So, finding the value of λ will be the important for us.

**So how do we find the value of λ?**

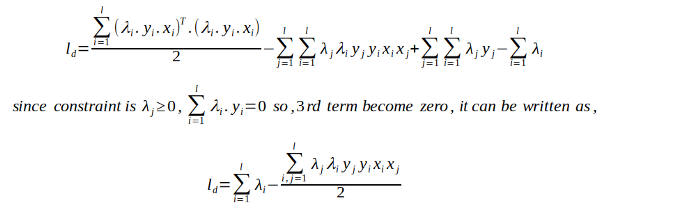
Above formulation is itself an optimization algorithm, but it not helpful to find the optimal value. **It is primal optimization problem**. As we read above that if Primal optimization doesn’t result in solution, we should use dual optimization formulation, which has guaranteed solution. Also, when we move from primal to dual formulation, we switch minimizing to maximizing the loss function. Again, we will divide the complete formulation into three parts to easier to understand.

1. Problem formulation and substitution value from primal
2. Simplify the loss function equation after substitution
3. final optimization formulation to get the value of λ



**PART I - Formulation from primal and Substitution**

Above equation is a **dual optimization problem**.



**PART II - Simplification**

It is simplified equation of above dual optimization problem.



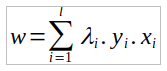


**PART III - Final optimization**

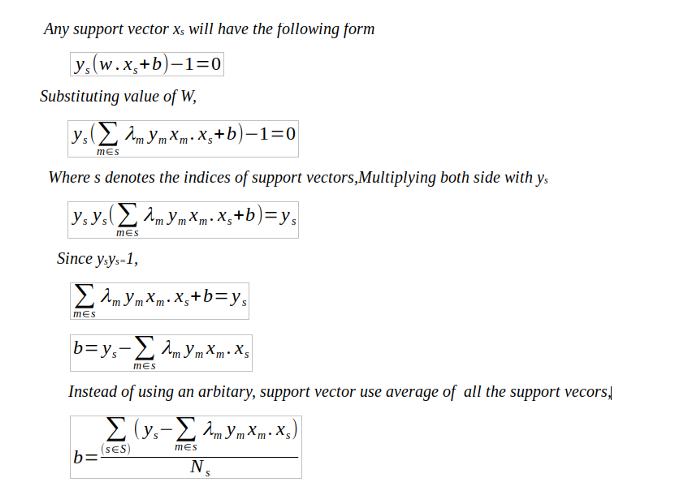
So, this is the final optimization problem, to find the maximum value of λ. Here one more term K is there, which is nothing but dot product of input variable x. (This K will be very important in future when we will learn about kernel trick and non-linear data points).

**Now, how do we Solve the above problem??**

Above maximization operation can be solved with the **SMO (Sequential minimization optimization) algorithms**. There are various library supports online for this optimization. Once we get the value of λ we can get w from below equation

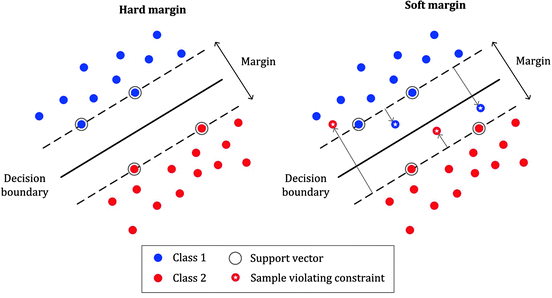


and using value of w , λ we will calculate b as following,

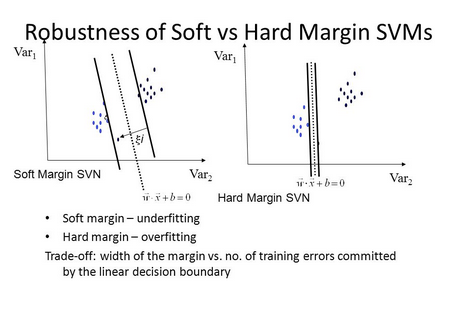


**Hard and Soft Margin Classifiers**

So far, we have seen how to find an optimal hyperplane from the set of hyperplane which separate the two classes and stays as far as from closest data-points (support vector). No data points are allowed in the margin areas. This type of linear classification is known as Hard margin classification.



**But here is the problem with hard margin classification!**



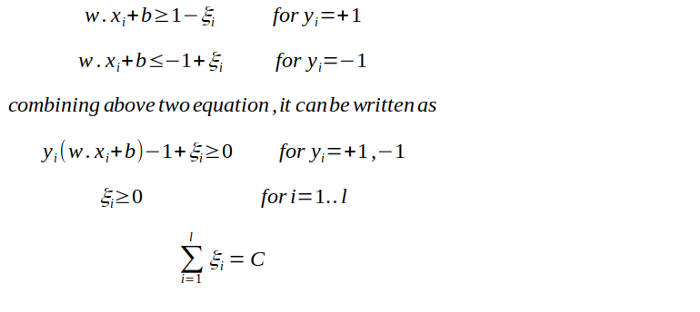
If we strictly impose that hyperplane stays as far as from the closest data points, then the data points will be linearly separable but there will be also narrow margin, in that case model will be extremely sensitive to noisy data points, and model will lack generalization capability i.e., chance of under-fitting will increase. Also, it will only work with linearly separable data-points(training example). data points and training example are interchangeable words referring to same thing.

**So, what is the way out?**

To avoid these issues, it is preferable to use more flexible model. As most of the real-world data are not fully linearly separable, we will allow some margin violation to occur. It is better to have large margin, even though some constraints are violated. Margin violation means choosing a hyperplane, which can allow some data points to stay in either in between the margin area or in the incorrect side of hyperplane, which is contrast to hard margin classification task. This type of classification is called as **Soft margin classification**.

**What does it mean Mathematically?**

We will relax the constrains of the equation slightly to allow the margin violation to occur with the help of positive **slack variable** ***ξi***. so now above equation can be written as,



***ξi*** actually tells where the **ith** observation is located relative to hyperplane and margin, for **0*<ξi*≤1,** then observation is between incorrect side of margin and correct side of hyperplane. This is margin violation. For *ξi>1*, observation is on the incorrect side of both hyperplane and margin, and point is misclassified. And in the soft margin classification task, the observation is on the incorrect side of margin have penalty which increases as distance from it increases.

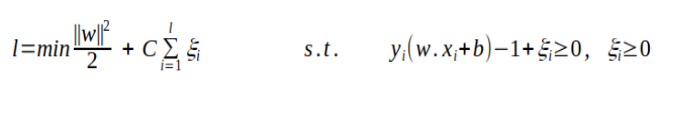
*C* is the parameter which controls the trade-off between length of margin and number of the misclassification on the training data. For C=0, It is not letting any misclassification to occur which is a nothing but hard margin classification and it result in narrow margin, *For C>0,* it means no more than C observation can violate the margin, as C increases margin also widens.

The correct value of *C* is decided by cross-validation and it is this parameter only, that can result in bias-variance trade-off in SVM. **If the value of *C =0*, which results in high variance but if the value of *C= ∞* it results in high bias.**

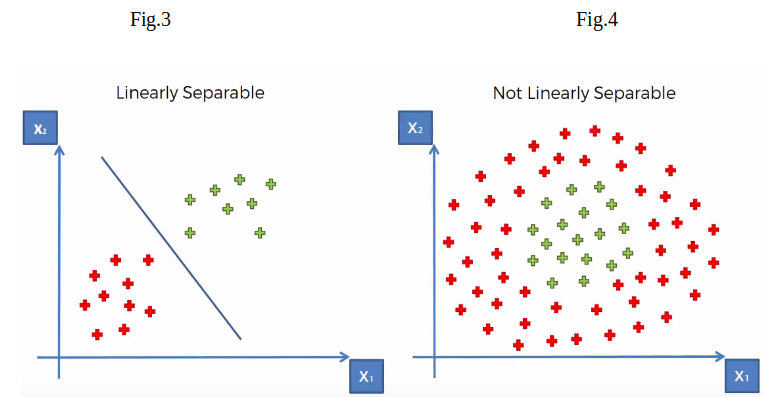
**Optimization**

As we seen in last part of SVM, the learning problem of Hard margin classifier is formulated as Dual quadratic Programming Problem. We have also understand the concept of primal and dual optimization problem.

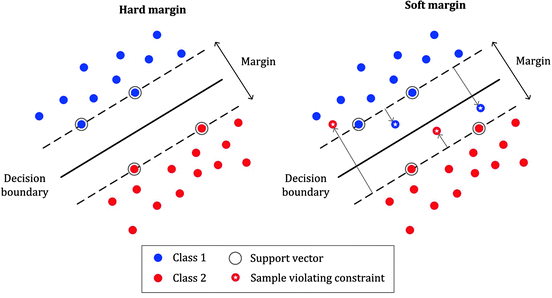
The soft margin optimization will be similar to that hard margin optimization task, but it will include an additional penalty term which penalize margin when a data point is mis-classified. The loss function is given as,



**Kernel Trick**



Last two part on SVM, we talk about the data points are either partially or fully linearly separable, knows as soft and hard margin classification task respectively.



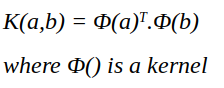
What if data points are not at all linearly separable? What if data points are scattered in shape of circle?

So, we need a generalized version of SVM which can perform well on any type of data shape, it doesn’t care if your data points have linear shape, circular, ellipsoid or anything, it just cares about finding an optimal decision boundary that divides both the two class.

Before diving deeper into concept, lets discuss few mathematical theorem and relation which are very crucial to understand this whole story.

## Mercer’s Theorem:

It says if a function **K(a,b)** satisfy all the constrains called mercer’s constrains, then there exists a function that maps **a** and **b** into higher dimension.

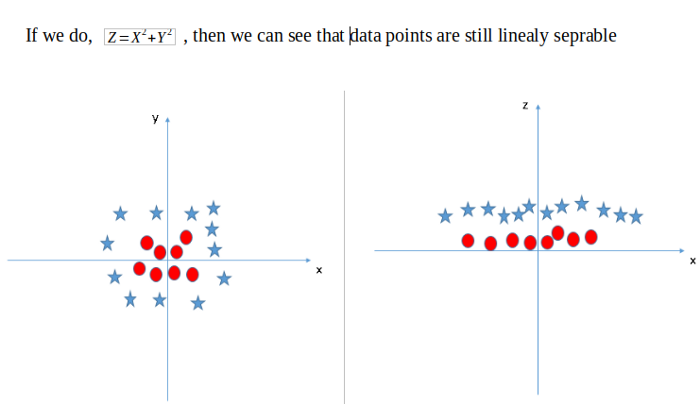


So, in this story we will talk about formulation of SVM which can deal with any shape of data points.

**Kernel trick!**

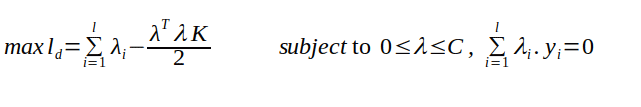
It is method of using linear classifier to classify non-linear data points. Mathematically, it is above explained Mercer’s theorem, which maps non-linear input data points into higher dimension where they can be linearly separable. And kernel is a function which actually perform the above task for us. There are different types of kernel like ‘linear’, ‘polynomial’, ‘redial basis function’ etc. Selecting a right kernel which can best suit your data is obtained by cross-validation.

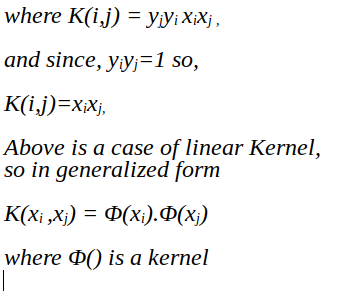
Small visualization of how ‘polynomial’ kernel can separates non-linear data into linearly separable in higher dimension. From the below figure you can see 2D plane data is linearly separated in 3D plane.



**So how do you actually use this kernel trick?**

In the PART I of SVM, Dual optimization problem is broken down to below maximization problem,





So solving the above minimization problem. you will get the value of λ, then using it, we will compute w and b. The λ dependent equation of w can be seen in PART I of the SVM. And from w we will compute b.

As we now have the value for both w and b, then optimal hyperplane that can separates the yedata points can be written as,

w.x + b = 0

And a new example x\_ can be classified as sign(w.x\_ +b)

## Popular kernel

## Gaussian Redial basis function

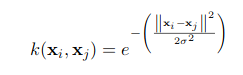
It is one of the most popular kernel used in SVM. It is used when there is no prior knowledge about data.



Gaussian redial basis function

## Gaussian function

It is written as,



## Polynomial Kernel Function

It is written as,



## Linear Kernel

It is just the normal dot product,

