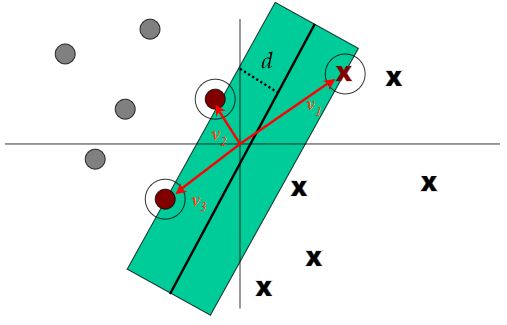
**Support vector Machines**

**What is support vector machine?**

**Important terminology related to support vector machine**

Support vectors are the data points that lie closest to the decision surface (or hyperplane)



**Margin of Separation** (d):- the separation between the hyperplane and the closest data point for a given weight vector w and bias b.

Few keywords – margin, gaps, optimal margin classifier, Lagrange duality, kernels, SMO algorithm

Margins –

If a point is far from the separating hyperplane, then we are quite confident on its belongingness to a particular class/category. Main idea is to find a decision boundary that allows us to make correct and confident predictions.

Let us think of a linear classifier hw,b(x) = g(wTx + b) . Hence g(z) = 1 if z>=0 else g(z) = -1 where w is similar to [Θ1, Θ2, Θ3, …,Θn-1, Θn]T and b is similar to Θ0

Functional margin – is not a good measure of confidence. Functional margin although tells you the sign of the label but it doesn’t tell the magnitude with reference on how far it is from decision plane

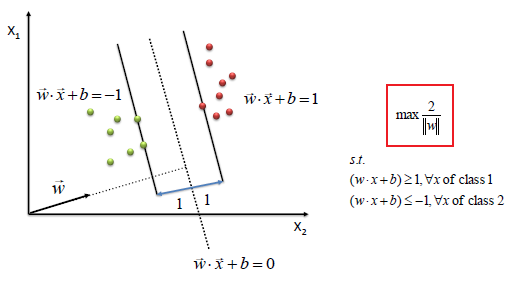
Functional margin (ϒ) of (w,b) wrt training set S of size m :-

ϒ(i)  = yi(wTxi + b)

& ϒ = min12..m ϒ(i)

Geometric margin is a better measure and so defined by and it is only a normalized version of functional margin with the help of weight vector. The geometric margin is telling you not only if the point is properly classified or not, but the magnitude of that distance in term of units of |w|

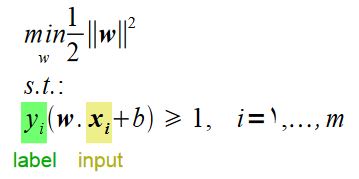
Another important discussion point is why it is profitable to find a wide margin instead of narrow margin.



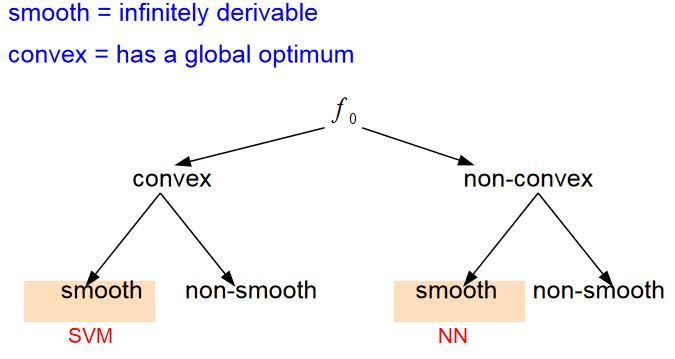
**Optimal Margin Classifier**

Here maximization 1/|w| problem has been converted to minimization ½ wTw problem.

where yi(wTxi + b) >=1.



Above is an optimization problem with convex quadratic objective and linear constraints and Its solution gives us the optimal margin classifier

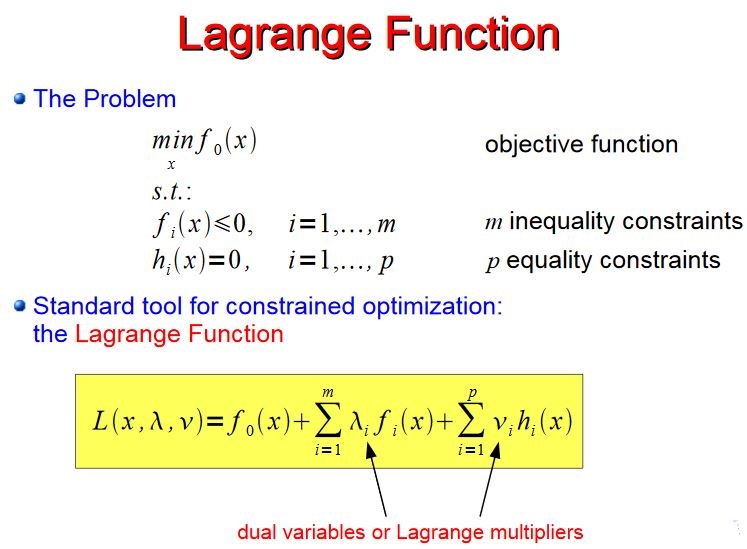


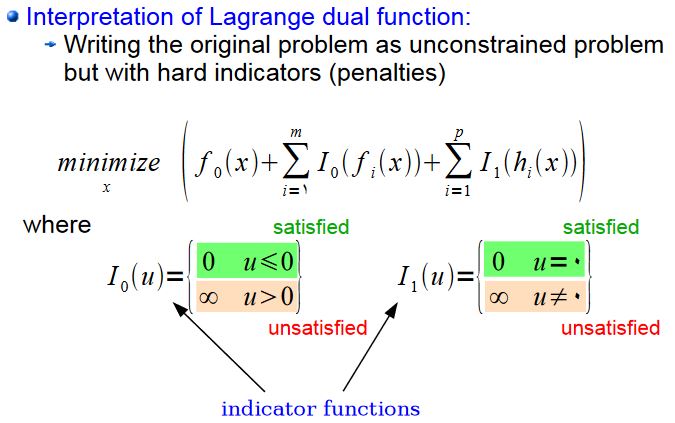
So, we need to ensure if the function is convex and smooth and only then a global solution will exist and SVM will be applicable for results with higher confidence.

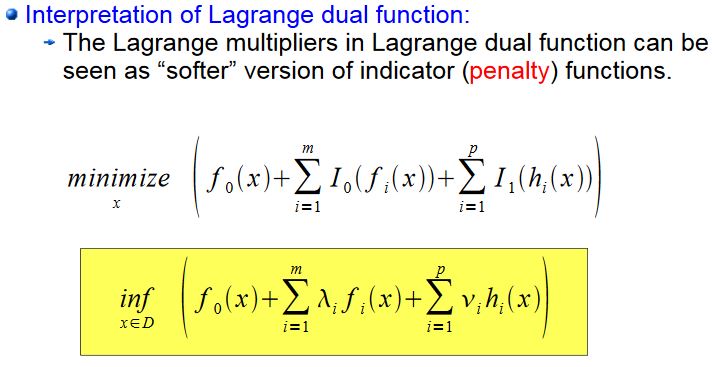
Until now we have been defined the problem in primal form.

Now we shall discuss what is dual problem and why we convert primal to dual. Why it is important to convert it into dual problem is that **it allows us to use kernels which helps us solving the problem efficiently in very high dimensional spaces**.

What is Lagrange function?





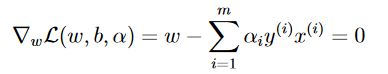


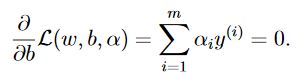
Lagrange dual is convex even if original problem is not

Karush-Kuhn-Tucker (KKT) conditions;-

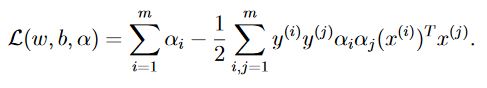
Karush–Kuhn–Tucker (KKT) conditions, also known as the Kuhn–Tucker conditions, are first-order necessary conditions for a solution in nonlinear programming to be optimal. Allowing inequality constraints, the KKT approach to nonlinear programming generalizes the method of Lagrange multipliers, which allows only equality constraints

Let’s take the derivative of L(w, λ, ν) w.r.t. w and b and set them to zero.

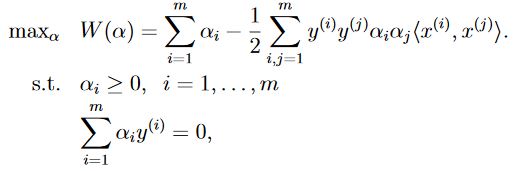




*After solving the above derivative equations*

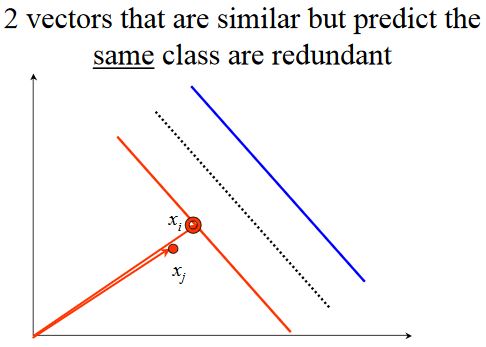
****

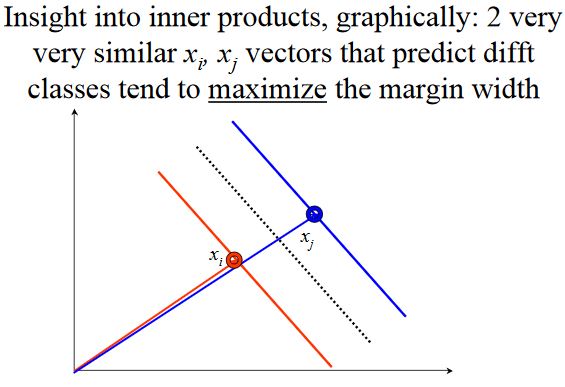
*Converting it into original problem format where L is the derivative wrt w and so putting it back with constraints αi ≥ 0*

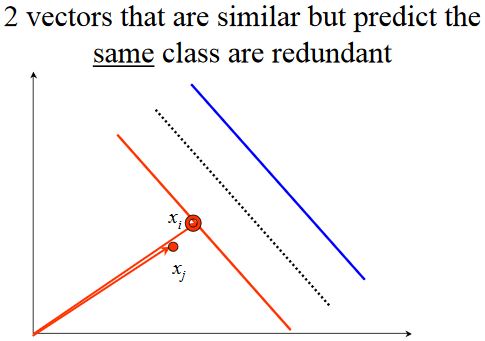
****

***Lagrange dual problem says instead of minimizing over weight w and bias b subjects to constraints involving a, we can maximize over a’s(dual variables).***

**Now the problem has become computationally achievable, as dual form just requires it to compute dot products of training points.**

******

******

****

**Kernels**

**What is a kernel?**

**K(x,z) = ΦT(x) \* Φ(z)**

**Kernels can be defined on strings, trees, structures etc.**

**What is kernel trick?**

Classifier can be learnt and applied without explicitly computing Φ(x). Complexity depends on O(N3) and not on D

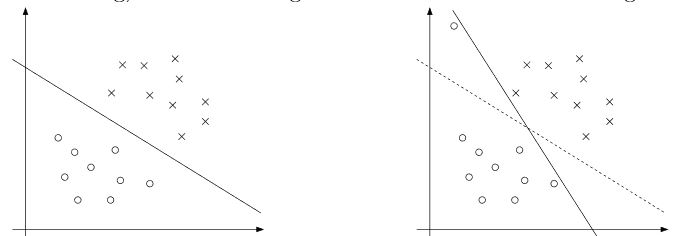
* Linear kernels,
* Polynomial kernels
* Gaussian kernels

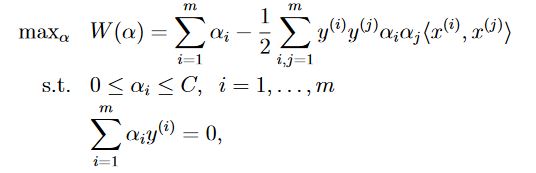
**How to check whether it is valid kernel or not**

**How SVMs avoid overfitting**

**L1 regularization:**

**For instance, we have a dataset in which we found an optimal margin classifier. Suppose we introduce an outlier in the dataset and it might cause decision boundary to make dramatic shift and resulting classifier has a much smaller margin.**

****



As per the new equation, the constraint has changed to including *αi ≤ C as well.*

**SMO Algorithm**

It gives an efficient way to solving the dual problem arising from the derivation of SVM.

Sequential minimal optimization (SMO) is an algorithm for solving the quadratic programming (QP) problem that arises during the training of support vector machines. The publication of the SMO algorithm in 1998 has generated a lot of excitement in the SVM community, as previously available methods for SVM training were much more complex and required expensive third-party QP solvers.

SMO is an iterative algorithm for solving the optimization problem. SMO breaks this problem into a series of smallest possible sub-problems, which are then solved analytically.

The algorithm proceeds as follows:

1. Find a Lagrange multiplier α1 that violates the Karush–Kuhn–Tucker (KKT) conditions for the optimization problem.
2. Pick a second multiplier α2 and optimize the pair ( α 1 , α 2 )
3. Repeat steps 1 and 2 until convergence.

When all the Lagrange multipliers satisfy the KKT conditions (within a user-defined tolerance), the problem has been solved.