Support Vector Machines

Lagrangian dual formulation

Formulation of Dual problem

$$\text{Maximize } \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$\text{subject to } \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0 \;\; \forall i$$

This a Quadratic Programming problem (QP)

This means that the parameters form a parabolloidal surface, and an optimal can be found(since it is convex function).

SVM training phase

Maximize L_D w.r.t. α_i such that $\alpha_i \geq 0$ with the solution $w = \sum_{i=1}^{n} \alpha_i y_i x_i$

In the solution those points for which $\alpha_i > 0$ are support vectors and they lie on Hyperplanes H_1 or H_2.

For other points $\alpha_i = 0$ and lie on the either sides of H_1 or H_2 such that

$$y_i(w^T x_i + b) > 1, \forall i$$

SVM test phase

In order to classify a new instance z we just have to compute.

$$sign(w^T z + b) = sign(\sum_{\forall i \in SV} \alpha_i y_i x_i^T z + b)$$

This means that w does not need to be explicitly computed

SVM:Example

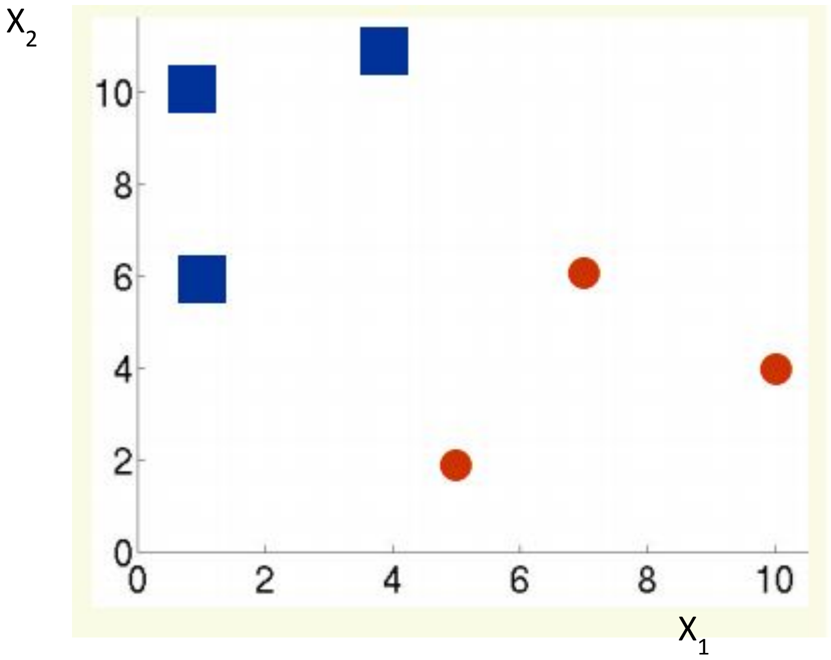
Given data :

Class-1 : [1,6] ,[1,10],[4,11]

Label y_i = 1

Class-2 : [5.2],[7,6],[10,4]

Label y_i = -1



SVM:Example

To represent all inputs consider array X

$$X = \begin{pmatrix} 1 & 6 \\ 1 & 10 \\ 4 & 11 \\ 5 & 2 \\ 7 & 6 \\ 10 & 4 \end{pmatrix} \quad \text{and labels } Y = \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{pmatrix}$$

SVM:Example

$$Y^T.Y = \begin{pmatrix} 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 \end{pmatrix}$$

$$X.X^T = \begin{pmatrix} 37 & 61 & 70 & 17 & 43 & 34 \\ 61 & 101 & 114 & 25 & 67 & 50 \\ 70 & 114 & 137 & 42 & 94 & 84 \\ 17 & 25 & 42 & 29 & 47 & 58 \\ 43 & 67 & 94 & 47 & 85 & 94 \\ 34 & 50 & 84 & 58 & 94 & 116 \end{pmatrix}$$

SVM:Example

Therefore Matrix H with $H_{ij} = y_i y_j x_i^T x_j$ can be written as

$$H = Y^T Y . X^T X = \begin{pmatrix} 37 & 61 & 70 & -17 & -43 & -34 \\ 61 & 101 & 114 & -25 & -67 & -50 \\ 70 & 114 & 137 & -42 & -94 & -84 \\ -17 & -25 & -42 & 29 & 47 & 58 \\ -43 & -67 & -94 & 47 & 85 & 94 \\ -34 & -50 & -84 & 58 & 94 & 116 \end{pmatrix}$$

SVM:Example in Matlab

Matlab expects quadratic programming to be stated in the canonical (standard) form which is

minimize $L_D(\alpha) = f^T \alpha + ½ \alpha^T H \alpha$ constrained to $A\alpha \leq a$ and $B\alpha = b$.

where A,B,H are n by n matrices and f, a, b are vectors

Need to convert our optimization problem to canonical form

Maximize $$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}^t H \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}$$ constrained to

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0 \ \forall i$$

SVM:Example in Matlab

Multiply by –1 to convert to minimization: $L_D(\alpha) = -\sum_{i=1}^{n} \alpha_i + \frac{1}{2} \alpha^t H \alpha$

Let f = $\begin{bmatrix} -1 \\ \vdots \\ -1 \end{bmatrix}$ = -ones(6,1), then we can write

Minimize $L_D(\alpha) = f^T \alpha + \frac{1}{2} \alpha^T H \alpha$

First constraint is $\alpha_i \geq 0 \ \forall i$

A = $\begin{bmatrix} -1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -1 \end{bmatrix}$ = -eye(6), a = $\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$ = zeros(6,1)

Rewrite the first constraint in canonical form : $A\alpha \leq a$

SVM:Example in Matlab

Our Second constraint is $\sum_{i=1}^{n} \alpha_i y_i = 0, \ \ \alpha_i \geq 0 \ \ \forall i$

Let B = [[Y] ; [zeros(5,6)]

and b = $\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$ = zeros(6,1)

Second constraint in canonical form is **Bα = b.**

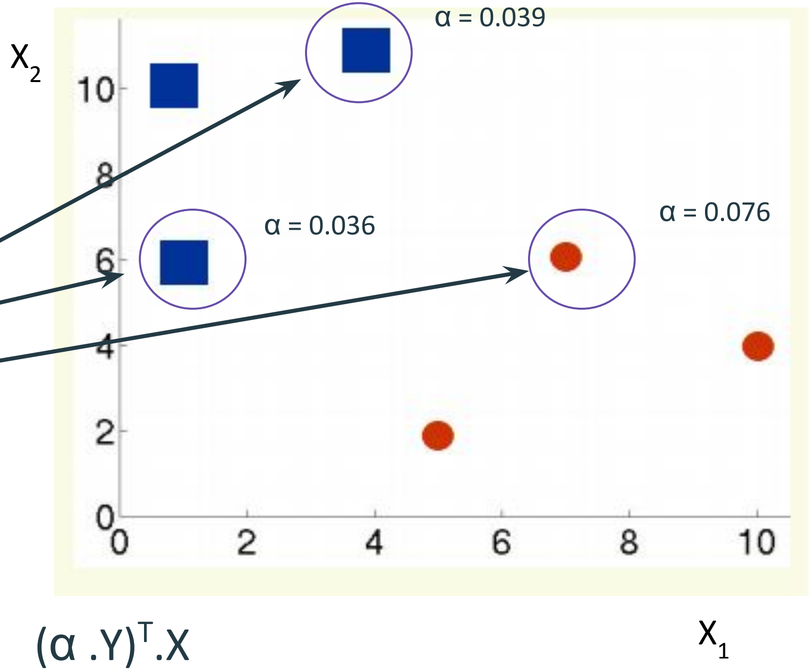
α = quadprog(H, f, A, a, B, b) %%in matlab

SVM:Example

Solution

$$\alpha = \begin{pmatrix} 0.036 \\ 0 \\ 0.039 \\ 0 \\ 0.076 \\ 0 \end{pmatrix}$$

Support Vectors



Find W using $w = \sum_{\forall i \in SV} \alpha_i y_i x_i$ = $(\alpha\ .Y)^T.X$

SVM:Example

$\alpha .Y = [0.036\ ,\ 0\ ,\ 0.04\ ,\ -0\ ,\ -0.076\ ,\ -0]$

$W = (\alpha .Y)^T.X = [-0.33\ ,\ 0.20] \Rightarrow w_1 = -0.33 \quad w_2 = 0.20$

We can find b using $w^T x_i + b = 1$ for positive support vector i =1

$b = 1 - w^T x_1 = 1 -(-0.33x1 + 6x0.20) = 1 -(-0.33 + 1.20)$

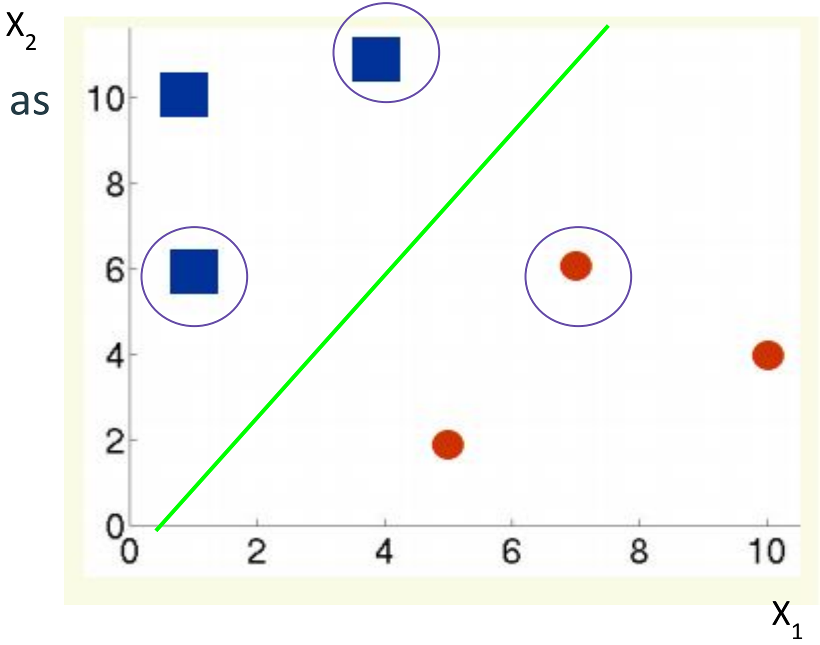
$= 0.13$

SVM:Example

Equation of the line can be written as

$-0.33\ X_1 + 0.20\ X_2 + 0.13 = 0$

(since $w^T x_i + b = 0$)



SVM:Example

Testing Phase:

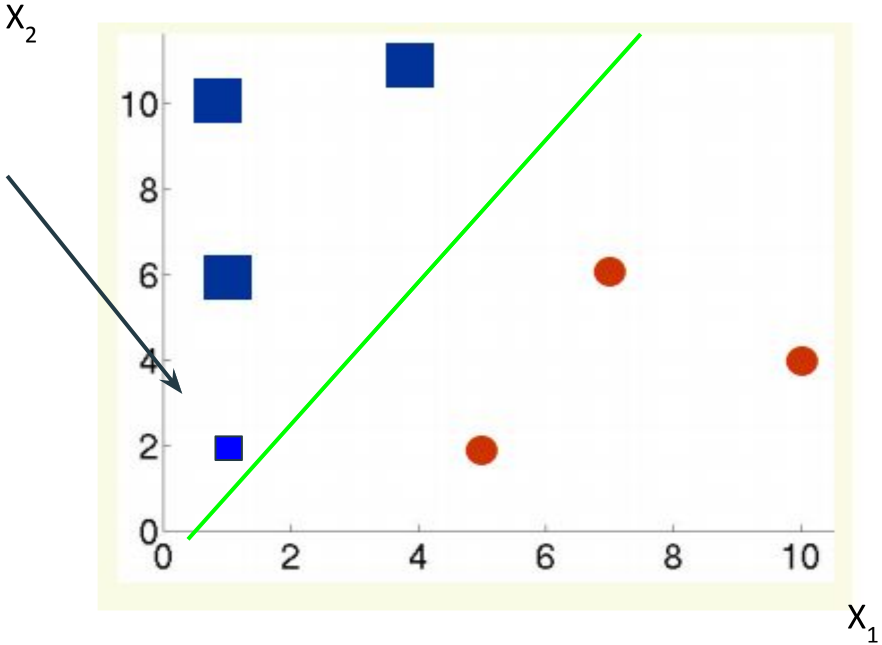
Suppose give a new point z = [1,2]

Find $sign(w^T z + b)$

sign(-0.33x1 0.20x2 + 0.13)

sign(-0.07 + 0.13) = positive

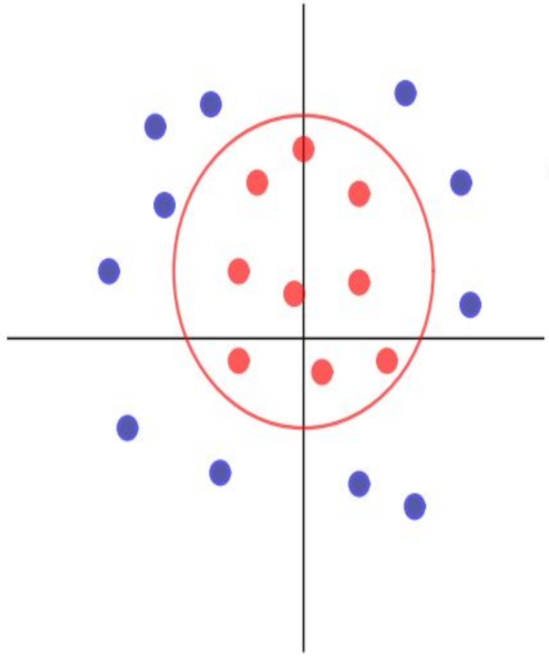
Therefore z is in class 1



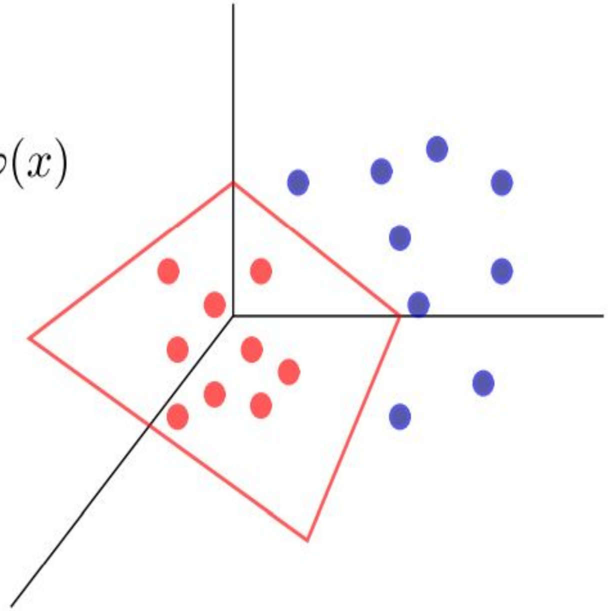
SVM: Non-linear classification

- So far we have only considered large margin classifiers that use a linear boundary.
- In order to have better performance we have to be able to obtain non-linear boundaries.
- The idea is to transform the data from the input space (the original attributes of the examples) to a higher dimensional space using a function $\varphi(x)$.
- This new space is called the feature space .
- The advantage of the transformation is that linear operations in the feature space are equivalent to non-linear operations in the input space.

Feature Space transformation



$$\phi : x \Rightarrow \varphi(x)$$



XOR - Problem

The XOR problem is not linearly separable in its original definition, but we can make it linearly separable if we add a new feature $x_1 \cdot x_2$

x_1	x_2	$x_1 \cdot x_2$	x_1 XOR x_2
0	0	0	1
0	1	0	0
1	0	0	0
1	1	1	1

The linear function $h(x) = 2x_1x_2 - x_1 - x_2 + 1$ classifies correctly all the examples.

Thank you