Theory - SVM

Sou Sarathi
IPhD20

★ Topics :

1. SVMs
2. Hard Margin SVM
3. Soft Margin SVM
4. Projected Gradient Descent (incomplete)
5. Non Linear SVMs
6. Multi Class SVMs (Incomplete)

links :

1. Youtube - ritvikmath
2. Youtube - Statquest, ritvikmath
3. Washington winter '17 Feixia
4. cs.cmu Aarti & Geoff.

Sou Sarathi C.

Given the data, for supervised problem

↳ Choose a model (linear, Bilinear, Polynomial etc)

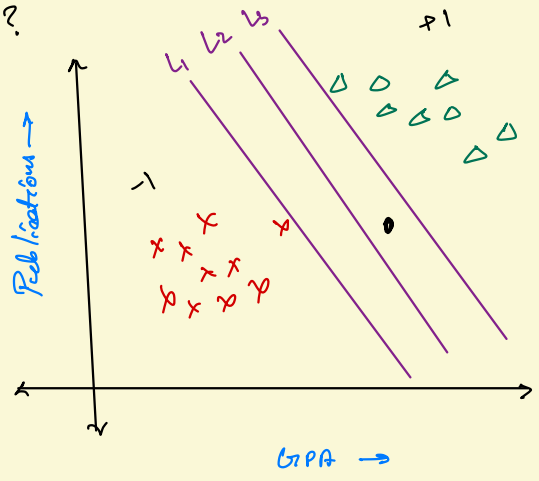
↳ Choose a suitable cost func. to find optimal parameters (least Sq. error, logistic Sigmoidal etc)

↳ Choose a suitable iterative tech. to solve (Grad. Descent, Stoch grad. descent)

Source: Ritvik math SVM youtube

2) SUPPORT VECTOR MACHINES:

Why?



PHD Positions

+1 : selected

-1 : not getting selected.

Question: Which plane would you choose to seperate the data? → L_2 hyperplane. Why?

What's the prob. with L_3? It also seperates the data.

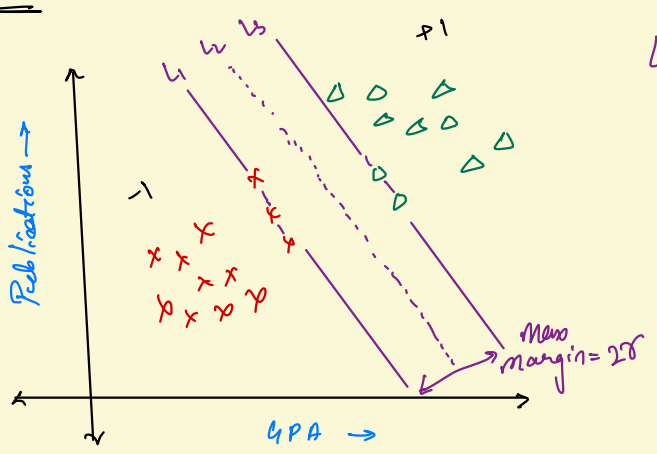
Suppose if we choose L_3 and we have a candidate in b/w L_2 & L_3 (-) he can get rejected despite having good Pub. & GPA. Same is the case with L_1

We choose L_2 i.e we choose an hyperplane which is not dangerously close to both data to avoid misclassification.

In other words, we want to maximize the margin.

(Margin: is the shortest dist b/w data & the chosen hyperplane) Margin = γ

* Support Vectors:



We choose L_2 hyperplane s.t it is midway b/w the margin.

L_2 = Decision boundary

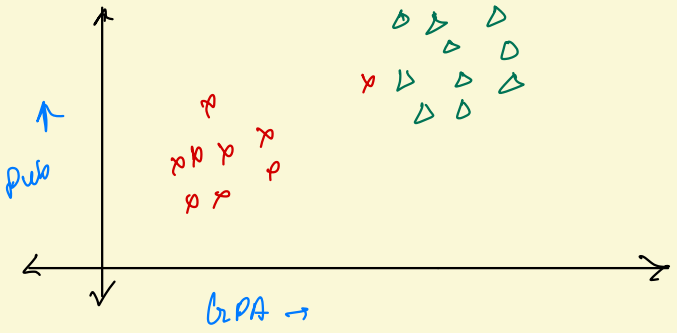
Support Vectors:

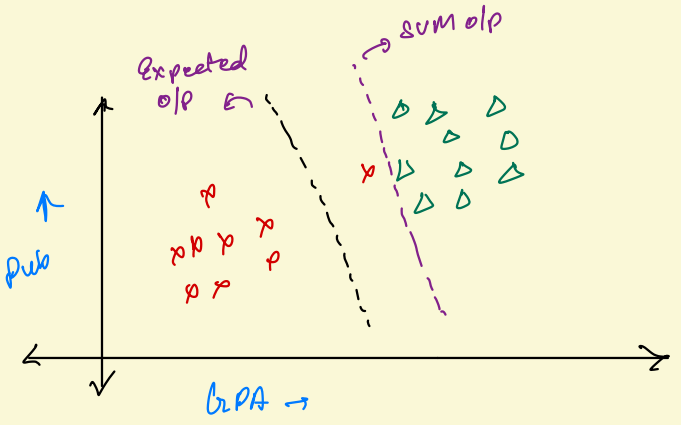
Some of the training data are called support vectors since they are the ones fully determining the margin.

Margin is supported by these support vectors. (data).

* Aim of SVM: is to maximize the margin.

One disadv. of SVM: in case of an outlying data point



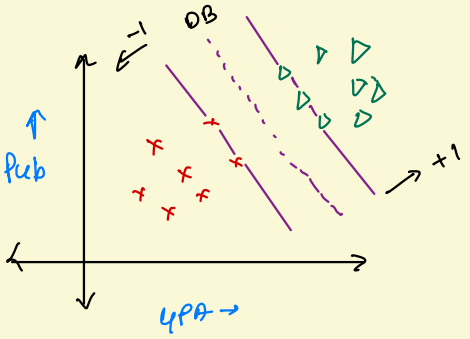


because SVM tries to maximize the margin

SVM

① Hard Margin

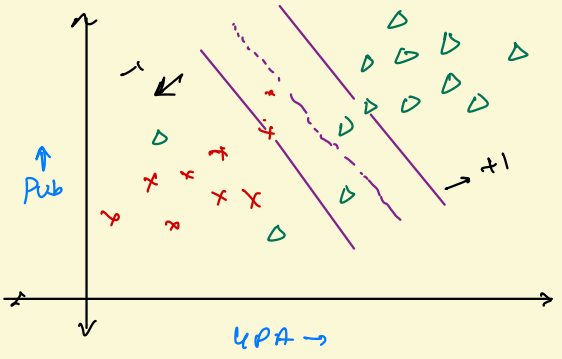
② SOFT MARGIN



To Right of DB: +1

To left of DB: -1

Works for linearly seperable



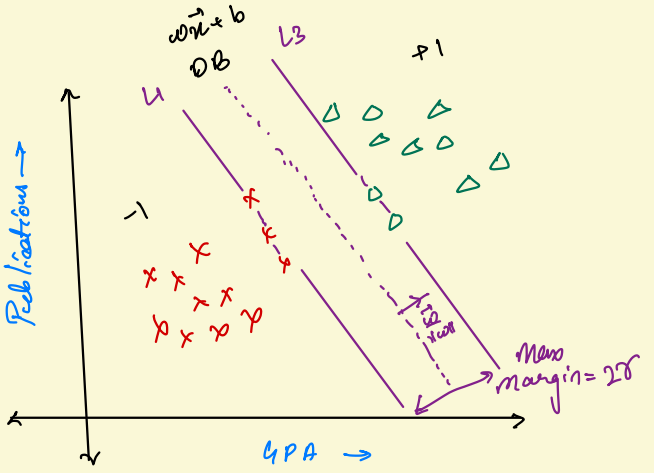
For data that are not linearly seperable.

Classify data so that type I is on one side of Bound & type II is on the other side but with some misclassifications With a penalty for every mistake depending upon the mistake

* SVM formulation:

- Margin If x on DB, $wx + b = 0$.

to determine: γ



let L_3 be the hyperplane

$\vec{w}\,x + b = 1$

Given a x on DB: $w\vec{x} + b$,

$\vec{w}\cdot\left(\vec{x} + K\frac{\vec{w}}{\|w\|}\right) + b = 1$ for some K.

$\vec{w}x + b + K\frac{(\vec{w},\vec{w})}{\|w\|} = 1$

$0 + K\frac{\|w\|^2}{\|w\|} = 1 \Rightarrow K = \frac{1}{\|w\|}$

$K = \frac{1}{\|w\|}$

So, $\gamma = \frac{1}{\|w\|}$ ∴ Problem: maximize 2γ = maximize $\frac{2}{\|w\|}$

≡ minimizing $\|w\| = \min \|w\|^2 = \min \frac{\|w\|^2}{2}$

Constraints: ① If $y_i = 1$ we need $\vec{w}\cdot\vec{x_i} + b \geq 1$

② If $y_i = -1$ we need $\vec{w}\cdot\vec{x_i} + b \leq -1$

Hence we require, $y_i(\vec{w}\cdot\vec{x_i} + b) \geq 1$ $\forall i$

- Optimization problem. :

SVM1 $\min_{w,b} \frac{1}{2} w^T w$ subject to

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

↳ (Convert as $1 - y_i(w^T x_i + b) \leq 0$)

Inequality constraint.

Step1

$L(w, b, \alpha_i) = \frac{1}{2} w^T w + \sum_{i=1}^{N} \alpha_i (1 - y_i(w^T x_i + b))$, $\alpha_i \geq 0$ $\forall i$,

$\alpha^*(\alpha_i) = \inf_{w,b} L(w, b, \alpha_i)$ → Dual prob.

Step 2: To find infimum of L^*,

$$\frac{\partial L}{\partial w} = 0, \quad \frac{\partial L}{\partial b} = 0$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{N} \alpha_i^* y_i^* x_i^* = 0 \quad \& \quad \frac{\partial L}{\partial b} = b \sum_{i=1}^{N} \alpha_i^* = 0$$

$$w^* = \sum_{i=1}^{N} \alpha_i^* y_i^* x_i^* \quad \text{and} \quad b \sum_{i=1}^{N} \alpha_i^* y_i^* = 0 \rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0$$

Step 3:

Substitute in $L^*(\alpha_i)$

$y_i^* =$ scalars

$x_i^* =$ vector

$$L^*(\alpha_i^*) = h(w^*, b^*, \alpha_i^*)$$

$$h(w, b, \alpha_i^*) = \frac{1}{2} w^T w + \sum_{i=1}^{N} \alpha_i^* (1 - y_i^* (w^T x_i^* + b))$$

$$h(w, b, \alpha_i^*) = \frac{1}{2} w^T w + \sum_{i=1}^{N} \alpha_i^* - \sum_{i=1}^{N} \alpha_i^* y_i w^T x_i^* + \sum_{i=1}^{N} \alpha_i^* b$$

- $\frac{1}{2} w^{*T} w^* = \frac{1}{2} \left(\sum_{i=1}^{N} \alpha_i^* y_i^* x_i^* \right)^T \left(\sum_{i=1}^{N} \alpha_i^* y_i^* x_i^* \right) = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i^* \alpha_j^* y_i^* y_j^* x_i^{*T} x_j$

- $\sum_{i=1}^{N} \alpha_i^* y_i w^T x_i^* = w^T \sum_{i=1}^{N} \alpha_i^* y_i^* x_i^* = \left(\sum_{i=1}^{N} \alpha_i^* y_i^* x_i^* \right)^T \left(\sum_{j=1}^{N} \alpha_j^* y_j^* x_j^* \right)$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i^* \alpha_j^* y_i^* y_j^* x_i^{*T} x_j$$

- $\sum_{i=1}^{N} b \alpha_i^* = 0$

Hence $$\lambda^*(\alpha_i^*) = \sum_{i=1}^{N} \alpha_i^* - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

∴ Our problem is reduced to:

$$\lambda^*(\{\alpha_i^*\}) = \inf_{w,b} h(w, b, \{\alpha_i^*\}) = \sum_{i=1}^{N} \alpha_i^* - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

Subject to $$\sum_{i=1}^{N} \alpha_i y_i^* = 0$$

$$\lambda^*(\{\alpha_i^*\}) = \sum_{i=1}^{N} \alpha_i^* - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

SUMMARY:

Step 4:

Primal Prob.

$$\min_{w,b} \frac{1}{2} w^T w \text{, subject to}$$

$$y_i (w^T x_i + b) \geq 1 \quad \forall i$$

minimizing over w, b

$\equiv$

Dual Prob.

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

$$\text{s.t} \sum_{i=1}^{N} \alpha_i y_i = 0$$

max over α_i

↳ dimension is red.

Recall $w^* = \sum_{i=1}^{N} \alpha_i y_i x_i$ But $r = \frac{2}{\|w\|}$ r is determined only by the support vectors.

But only vectors contributing to the margin are the support vectors.

It is necessary that $\alpha_i = 0$ for the non support vectors.

$\therefore \alpha_i = 0$ for non support vectors & dual problem simplifies.

$$b^* = y_i - w^{*T} x_i$$

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j = \min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{N} \alpha_i$$

$$\begin{bmatrix} x_1^T x_1 & x_1^T x_2 \\ x_2^T x_1 & x_2^T x_2 \end{bmatrix}$$

$$Q = [Q_{ij}]_{n \times n}$$

$$[Q_{ij}]_{n \times n} = [y y^T]_{n \times n} \odot \begin{bmatrix} x_1^T x_1 & \cdots & x_1^T x_N \\ & x_i^T x_j & \\ x_n^T x_1 & \cdots & x_N^T x_N \end{bmatrix}$$

↳ Inner product b/w x_i's

$$y y^T = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \begin{bmatrix} y_1 & y_2 \end{bmatrix} = \begin{bmatrix} y_1 y_1 & y_1 y_2 \\ y_2 y_1 & y_2 y_2 \end{bmatrix}$$

$$\therefore y y^T \odot \begin{bmatrix} x_1^T x_1 & x_1^T x_2 \\ x_2^T x_1 & x_2^T x_2 \end{bmatrix} = \begin{bmatrix} y_1 y_1 x_1^T x_1 & y_1 y_2 x_1^T x_2 \\ y_2 y_1 x_2^T x_1 & y_2 y_2 x_2^T x_2 \end{bmatrix}$$

$$\begin{bmatrix} \alpha_1 & \alpha_2 \end{bmatrix}_{1 \times 2} \begin{bmatrix} y_1 y_1 x_1^T x_1 & y_1 y_2 x_1^T x_2 \\ y_2 y_1 x_2^T x_1 & y_2 y_2 x_2^T x_2 \end{bmatrix}_{2 \times 2} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \alpha_1 y_1 y_1 x_1^T x_1 + \alpha_2 y_2 y_1 x_2^T x_1 & \alpha_1 y_1 y_2 x_1^T x_2 + \alpha_2 y_2 y_2 x_2^T x_2 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

$= \alpha_1^2 y_1 y_1 x_1^T x_1 + \alpha_1 \alpha_2 y_2 y_1 x_2^T x_1 + \alpha_2 \alpha_1 y_1 y_2 x_1^T x_2 + \alpha_2^2 y_2 y_2 x_2^T x_2 . = \sum_{i=1}^{2} \sum_{j=1}^{2} \alpha_i \alpha_j y_i y_j x_i^T x_j$

Hence $\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j = \alpha^T Q \alpha$

- linear SVMs are convex optimization $\Longrightarrow$ strong duality
- the unique solution, i.e. $(\mathbf{w}^*, b^*, \{\alpha_i^*\})$, is a **saddle point**

$$\underbrace{\text{SVM1}}_{\text{prime problem}} \iff \underbrace{\max_{\{\alpha_i\}} L^*(\alpha_i) \text{ s.t. } \sum_{i=1}^{N} \alpha_i y_i = 0}_{\text{dual problem}}$$

- solving the dual problem yields $\{\alpha_i^*\}$
- linear SVMs: $\mathbf{w}^* = \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i$ and $b^* = y_i - \mathbf{w}^{*\mathsf{T}} \mathbf{x}_i$

∴ Q for Dual prob. can be equ. written as:

$$\max_{\alpha} \; 1^T \alpha - \frac{1}{2} \alpha^T Q \alpha$$

subject to $y^T \alpha = 0$

$\alpha \geq 0$

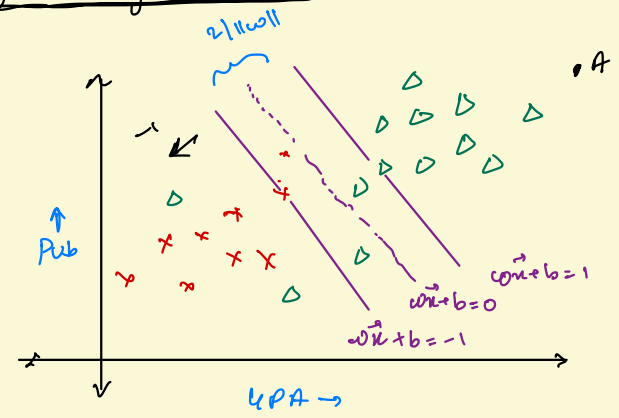
Imp. obs: For the optimization we just care about the inner product among the data

$Q = [Q_{ij}]_{n \times n}$

$[Q_{ij}]_{n \times n} = [y y^T]_{n \times n} \odot \begin{bmatrix} x_1^T x_1 & \cdots & x_1^T x_N \\ & x_i^T x_j & \\ x_n^T x_1 & \cdots & x_N^T x_N \end{bmatrix}$

Involves only inner product

② Soft Margin SVMs:



cannot use Hard margin SVM to separate

Spirit of Soft Margin SVM:

Classify the data as well as possible but also willing to accept certain level of mistakes with penalties based on how big the mistake is.

hinge loss: $\max[0, 1 - y_i^*(w x_i^* + b))$ — score

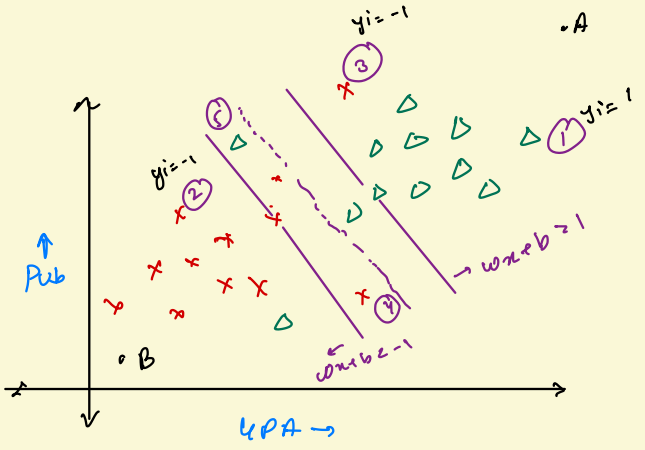
score: $w x_i^* + b$

If the obs. lies in b/w $wx+b=0$ and $wx+b=1$

$0 < \text{score} < 1$..

If obs lies in b/w $wx+b=0$ & $wx+b=-1$, $-1 < \text{score} < 0$.

@ point A score is $\gg 1$



score @ pt A: $\gg 1$

score @ pt B: $\ll 1$

① $\max(0, 1 - 1(>1)) = 0$

② $\max(0, 1 + 1(<(-1))) = 0$

③ $\max(0, 1 + 1(>1)) = >2$

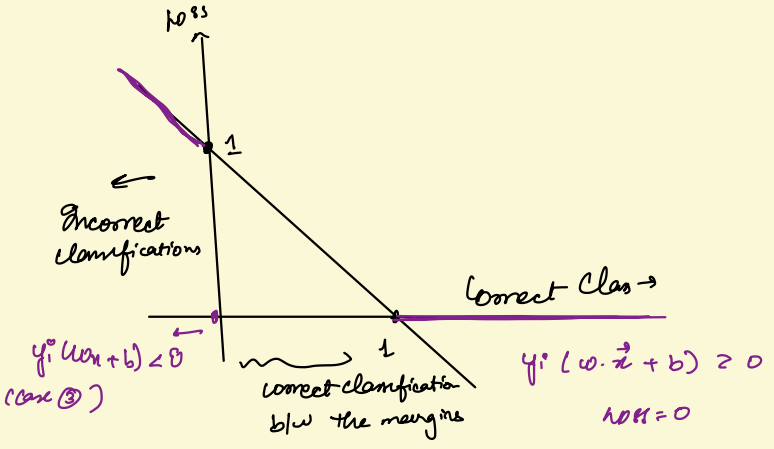
④ $\max(0, 1 + 1(\text{b/w } 0 \text{ \& } -1)) =$ something b/w 0 and 1

⑤ $\max(0, 1 - 1(\text{b/w } 0 \text{ \& } -1)) =$ something b/w 1 & 2

∴ for correct classifications, $y_i^*(wx+b) > 0$

& score = 0.

hinge loss



* Soft margin SVM Mathematical formulation:

2 things: ① Maximize the margin : minimize $\|w\|$

② Minimize the hinge loss.

- C = parameter that decides the tradeoff b/w max. margin or min. hinge loss.

C = small ⇒ hinge loss given less importance, C = large ⇒ focus on max. margin

$$\min_{w, b, \xi_i} \ \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i \quad , \quad \xi_i = \text{hinge loss} = \max \{ 0, \ 1 - y_i (w^T x_i + b) \}$$

Equivalently:

$$\text{minimize}_{w, b, \xi_i} \ \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i$$

$$\xi_i \geq 0 \quad \forall \ i \in \{1, \dots N\}$$

$$\xi_i \geq 1 - y_i (w^T x_i + b) \quad \forall \ i \in \{1, 2, 3, \dots N\}$$

Derivation: (Constraints: $-\xi_i \leq 0$, $1 - y_i (w^T x_i + b) - \xi_i \leq 0$)

Step-1 : Dual prob: for $\alpha_j, \beta_j \geq 0$,

$$L(w, b, \xi_i, \alpha_i, \beta_i) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i \xi_i + \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \alpha_i y_i (w^T x_i + b) - \sum_{i=1}^{N} \beta_i \xi_i$$

Step-2

i) $\frac{\partial L}{\partial w} = 0 \Rightarrow w - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \Rightarrow w^* = \sum_{i=1}^{N} \alpha_i y_i x_i$

ii) $\frac{\partial L}{\partial \xi_j} = 0 \quad \forall \ j = 1, 2, \dots N \Rightarrow C - \alpha_j - \beta_j = 0 \Rightarrow C = \alpha_j + \beta_j \quad \forall j$

iii) $\frac{\partial L}{\partial b} = 0 \Rightarrow - \sum_{i=1}^{N} \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0$

$\beta_i \geq 0 \Rightarrow \quad C - \alpha_i \geq 0 \quad \forall i \in \{1, 2, \dots N\}$

$C \geq \alpha_i \geq 0 \quad \forall i \in \{1, 2, \dots N\}$

Step 3 Substitute in lagrangian

$$L(w^*, b^*, \xi_i, \alpha_i, \beta_i) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{N} \underbrace{(C - \alpha_i - \beta_i)}_{0} \xi_i + \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \alpha_i y_i (w^T x_i + b)$$

$$L(w^*, b^*, \xi_i, \alpha_i, \beta_i) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{N} \alpha_i (1 - y_i (w^T x_i + b))$$

$$\sum_{i=1}^{N} \alpha_i y_i w^T x_i = w^T \sum_{i=1}^{N} \alpha_i y_i x_i = \left[\sum_{i=1}^{N} \alpha_i y_i x_i\right]^T \left[\sum_{j=1}^{N} \alpha_j y_j x_j\right]$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \qquad \text{and} \qquad \sum_{i=1}^{N} \alpha_i y_i b = 0$$

$$\therefore \quad L(w^*, b^*, \xi_i, \alpha_i, \beta_i) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

Dual Prob. : $L(w^*, b^*, \xi_i, \alpha_i, \beta_i) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$ subject to

$\sum_{i=1}^{m} \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$

- Or Dual prob. can be equ. written as:

$$\max_{\alpha} \; 1^T \alpha - \frac{1}{2} \alpha^T Q \alpha$$

subject to $y^T \alpha = 0$

$0 \leq \alpha \leq C$

Imp. obs: For the optimization we just care about the inner product among the data

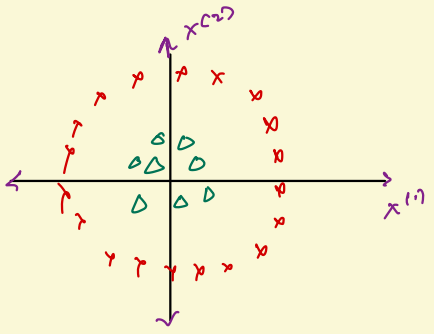
Involves only inner product

$Q = [Q_{ij}]_{n \times n}$

$[Q_{ij}]_{n \times n} = [y y^T]_{n \times n} \odot \begin{bmatrix} x_1^T x_1 & \dots & x_1^T x_N \\ & x_i^T x_j & \\ x_n^T x_1 & \dots & x_N^T x_N \end{bmatrix}$

★ PROJECTED GRADIENT DESCENT (For the constraint $0 \leq \alpha \leq C$)

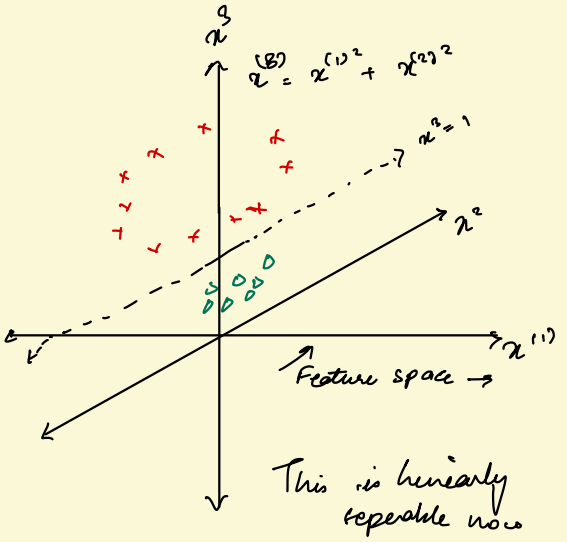
Non-linear SVMs:



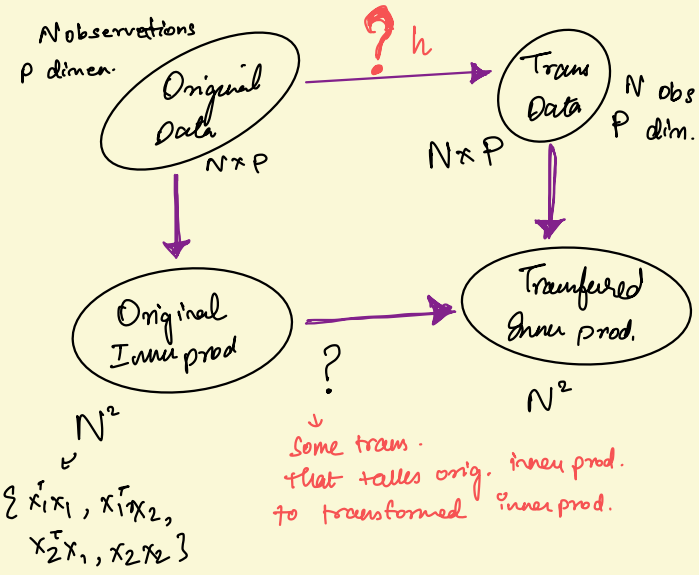
We can't seperate this by a hyperplane.

Adding a new dimension

$h(x^{(1)}, x^2)$



Idea: Project the data into a higher dim space so that it's linearly seperable in the higher dimension



Recall: For the Hard margin in SVM, we just cared about the inner products b/w the data. (for the Dual prob)

$x_i \cdot x_j \rightarrow h(x_i) \cdot h(x_j)$

Aim: We just need the inner products of the transformation.

for example

for an input point x_i^*,

$$\begin{bmatrix} x_i^{(1)} \\ x_i^{(2)} \end{bmatrix} = \begin{bmatrix} 1/2 \\ x_i^1 \\ x_i^2 \\ x_i^1 x_i^1 \\ x_i^1 x_i^2 \\ x_i^2 x_i^2 \\ x_i^2 x_i^1 \end{bmatrix}$$

→ all possible 2 way interactions.

Example of projection in 7 dc

$N \times P$ matrix in memory is computationally very expensive than the $N \times p$. Hence we need not add new data, instead we can go for the dual problem calculate N^2 & carry on.

Aim: To need $h(x_i^*) \cdot h(x_j) = \begin{bmatrix} 1/2 \\ x_i^1 \\ x_i^2 \\ x_i^1 x_i^1 \\ x_i^1 x_i^2 \\ x_i^2 x_i^2 \\ x_i^2 x_i^1 \end{bmatrix} \begin{bmatrix} 1/2 \\ x_j^1 \\ x_j^2 \\ x_j^1 x_j^1 \\ x_j^1 x_j^2 \\ x_j^2 x_j^2 \\ x_j^2 x_j^1 \end{bmatrix} = \frac{1}{4} + x_i^{*1} x_j^1 + x_i^{*2} x_j^2 + (x_i^{*1} x_j^1)^2 + (x_i^{*1} x_i^2 x_j^1 x_j^2) + (x_i^{*2} x_j^2)^2 + (x_i^{*2} x_i^{*1} x_j^2 x_j^1)$

We want to get the above terms without explicitly computing the transformation.

Let Polynomial kernel $\phi(x_i \cdot x_j) = (1 + x_i \cdot x_j)^2 = \left(\frac{1}{2} + x_i^1 x_j^1 + x_i^2 x_j^2\right)^2$

Polynomial Kernel

$$\left(\frac{1}{2} + x_i^1 x_j^1 + x_i^2 x_j^2\right)\left(\frac{1}{2} + x_i^1 x_j^1 + x_i^2 x_j^2\right) =$$

$$\frac{1}{4} + \frac{1}{2}x_i^1 x_j^1 + \frac{1}{2}x_i^2 x_j^2 + \frac{1}{2}x_i^1 x_j^1 + (x_i^1 x_j^1)^2 + (x_i^2 x_j^2 x_i^1 x_j^1) + \frac{1}{2}x_i^2 x_j^2 + (x_i^2 x_j^2 x_i^1 x_j^1) + (x_i^2 x_j^2)^2$$

$$= \frac{1}{4} + x_i^1 x_j^1 + x_i^2 x_j^2 + (x_i^1 x_j^1)^2 + (x_i^2 x_j^2)^2 + (x_i^2 x_j^2 x_i^1 x_j^1) + (x_i^2 x_j^2 x_i^1 x_j^1)$$

$h: \mathbb{R}^m \to \mathbb{R}^n$

$\phi(x_i, x_j)$.

$\mathbb{R}^m + \mathbb{R}^m \to \mathbb{R}$

Hence we found

ϕ: Original Ip $\longrightarrow$ Transformed Ip.

★ Polynomial Kernel: $\phi(x_i, x_j) = (x_i^T x_j + 1)^p$ p = dimension

↳ Overall picture:

① Introduce an additional dimension so that the non linear data is separable in the higher dimension. $h(x^{(1)}, x^{(2)})$ → o/p could be any no. of dimensions. [Not sure how to choose this dimension]

② Find a hyperplane in the higher dimension.

ie. finding an optimal w, b in the higher dimension. How do we do it?

If it's linearly separable we can find using hard margin SVM in the higher dim:

Primal Prob.

$\min_{w,b} \frac{1}{2} w^T w$, subject to

$y_i(w^T h(x_i) + b) \geq 1 \quad \forall i$

minimizing over w, b

Dual Prob.

$\max_{\{\alpha_i\}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j h(x_i)^T h(x_j)$

s.t $\sum_{i=1}^{N} \alpha_i y_i = 0$

max over α_i

≡

(dimension is increased)

We use the dual problem. So here we just need to compute $h(x_i)^T h(x_j)$ which could be found by using Kernel function.

③ After finding w, b project it onto the current dimension (I'm not sure how to do this but this is the idea.)

We have found the non-linear hyperplane.

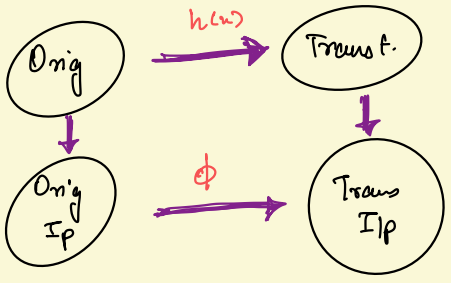
- **RADIAL BASIS KERNEL:**

$$\phi(x_i, x_j) = e^{-\frac{1}{2}\|x_i - x_j\|^2}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \\ x_1^2 \\ \vdots \\ x_1^3 \\ x_2^3 \\ x_1^3 x_2^2 \\ \vdots \\ \infty \end{bmatrix}$$

wishes to consider all possible interactions b/w the variables.

It's Infinite dimensional.



How to take the inner product of this ∞e dimensional vectors?

$$\begin{aligned}
\phi(x_i, x_j) &= e^{-\frac{1}{2}\|x_i - x_j\|^2} \\
&= e^{-\frac{1}{2}\left((x_i - x_j)^T (x_i - x_j)\right)} \\
&= e^{-\frac{1}{2}\left(x_i^T x_i - 2x_i^T x_j + x_j^T x_j\right)} \\
&= e^{-\frac{1}{2}\left(x_i^T x_i + x_j^T x_j\right)} \, e^{x_i^T x_j} \\
&= C' \, e^{x_i^T x_j}
\end{aligned}$$

$$\phi(x_i, x_j) = C' \, e^{x_i^T x_j + 1} \, e^{-1}$$

$$= C' \, e^{x_i^T x_j + 1} = e' \sum_{n=1}^{\infty} \frac{(1 + x_i^T x_j)^n}{n!} = C' \sum_{n=1}^{\infty} \frac{\phi_{poly}(x_i^T x_j)}{n!}$$

This captures the ∞e no. of terms involved in the inner product

* RBF Kernel: $\phi(x_i, x_j) = e\left(-\gamma \|x_i - x_j\|^2\right)$

• In summary, NON LINEAR SVMs :

Feature space(higher dimensional space)

• First idea was: Learning a non-linear classifier using SVM
– Calculate h(x)for each training example – Find a linear SVM in the feature space.

• Problems:
– Feature space can be high dimensional or even have infinite dimensions, so storing h(x) can be inefficient or even impossible

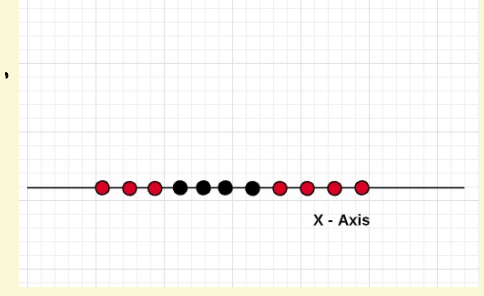
So we came up with the idea of “Kernels”:

-Kernels are similarity functions that return inner products between the images h(xi) of data points.
• Choosing kernel (phi) is equivalent to choosing h(x) in the feature space .

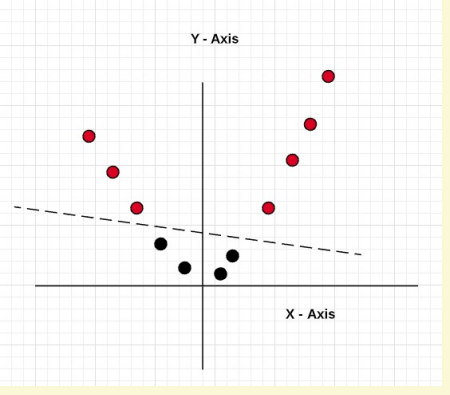
- $\phi : X \times X \rightarrow \mathbb{R}$

$\phi(x_i, x_j) = h(x_i)(h(x_j))^T$ (inner product b/w $h(x_i)$ & $h(x_j)$)

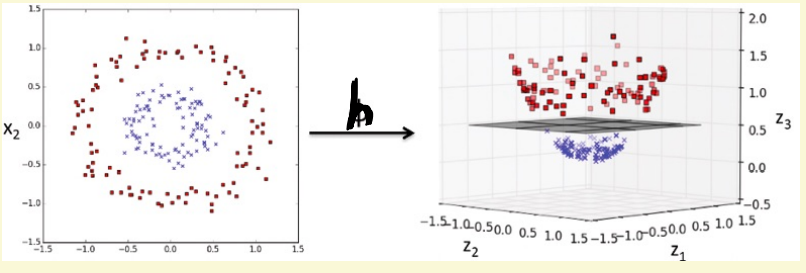
The kernel trick
• No need to know what h(x) is and what the feature space is.
• No need to explicitly map the data to the feature space.(memory constraint resolved)
• Define a kernel function phi and replace the dot produce <x,z> with a kernel function phi(x,z) in both training and testing.



Linearly inseparable data in : one-dimension



Applying kernel method to represent data using 2-dimensions



Linear

- Maximize

$$L(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \vec{x_i}, \vec{x_j} \rangle$$

Subject to $\alpha_i \geq 0$ and $\sum_i \alpha_i y_i = 0$

- Equivalently

$$\max_{\alpha} \; 1^T\alpha - \frac{1}{2}\alpha^T Q \alpha \quad \text{(Hard SVM)}$$

Sub. to $y^T\alpha = 0$, $\alpha \geq 0$

$Q = [Q_{ij}]_{n \times n}$

$[Q_{ij}]_{n \times n} = [yy^T]_{n \times n} \odot \begin{bmatrix} x_1^T x_1 & \cdots & x_1^T x_n \\ \vdots & x_i^T x_j & \vdots \\ x_n^T x_1 & \cdots & x_n^T x_n \end{bmatrix}$

- Plane. $w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i$

$b^* = y_i - w^{*T} x_i$

Plane: $w^* x + b^*$

$$= \sum_i \alpha_i y_i \langle \vec{x_i}, \vec{x} \rangle + b$$

Non linear case

- $L(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(\vec{x_i}, \vec{x_j})$

So problem is resolved without actually going to higher dimension, ∴ we need n't actually store $h(x_i)$ Just replace the orig. ip by ip in higher dim.

- Equivalently

$$\max_{\alpha} \; 1^T\alpha - \frac{1}{2}\alpha^T Q \alpha$$

Sub. to $y^T\alpha = 0$, $\alpha \geq 0$

$Q = [Q_{ij}]_{n \times n}$

$[Q_{ij}]_{n \times n} = [yy^T]_{n \times n} \odot \begin{bmatrix} h(x_1)^T h(x_1) & \cdots & h(x_1)^T h(x_n) \\ \vdots & h(x_i)^T h(x_j) & \vdots \\ h(x_n)^T h(x_1) & \cdots & h(x_n)^T h(x_n) \end{bmatrix}$

- Plane: $w^* = \sum_{i=1}^{N} \alpha_i^* y_i h(x_i)$

$b^* = y_i - w^{*T} x_i$

$w^* x + b^*$

$$f(\vec{x}) = \sum_i \alpha_i y_i \phi(\vec{x_i}, \vec{x}) + b$$

$\phi(x_i, x) = h(x_i)^T h(x)$

Summary so far

- Find the hyperplane that maximizes the margin.
- Introduce soft margin to deal with noisy data
- Implicitly map the data to a higher dimensional space to deal with non-linear problems.
- The kernel trick allows infinite number of features and efficient computation of the dot product in the feature space.
- The choice of the kernel function is important.

Common kernel functions

- Linear : $K(\vec{x}, \vec{z}) = < \vec{x}, \vec{z} >$
- Polynominal: $K(\vec{x}, \vec{z}) = (\gamma < \vec{x}, \vec{z} > + c)^d$
- Radial basis function (RBF): $K(\vec{x}, \vec{z}) = e^{-\gamma(||\vec{x}-\vec{z}||)^2}$

$$x - z = \begin{bmatrix} x_1 - z_1 \\ \vdots \\ x_N - z_N \end{bmatrix}$$

$$||x - z|| = \sqrt{(x_1 - z_1)^2 + \dots + (x_N - z_N)^2}$$

Example Linear Kernel: $h(x) = \alpha x$ for some α

$h(x_1, x_2, \dots, x_N) = [\alpha x_1, \alpha x_2, \dots \alpha x_N]$

$\therefore \phi(x, z) = (h(x), h(z)) = (h(x))^T h(z)$

$= [\alpha x_1, \dots \alpha x_N] \begin{bmatrix} \alpha z_1 \\ \vdots \\ \alpha z_N \end{bmatrix} = \alpha^2 x_1 z_1 + \alpha^2 x_2 z_2 + \dots + \alpha^2 x_N z_N$

The Kernel Matrix (a.k.a. the Gram matrix)

K(1,1)	K(1,2)	K(1,3)	…	K(1,m)
K(2,1)	K(2,2)	K(2,3)	…	K(2,m)
…				
…				
K(m,1)	K(m,2)	K(m,3)	…	K(m,m)

K(i, j) means $K(x_i, x_j)$,

where x_i means the i-th training instance.

$\phi(x_i, x_j) = (h(x_i))^T (h(x_j))$

Just storing all possible innerproducts in the form of a matrix

Mercer's Theorem

- The kernel matrix is symmetric positive definite.
- Any symmetric, positive definite matrix can be regarded as a kernel matrix; that is, there exists a ϕ such that K(x, z) = <h(x), h(z)>

SVM Learning Procedure (in a nutshell)

Given a training set as $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$

1. choose a kernel function $\Phi(\mathbf{x}_i, \mathbf{x}_j)$
2. build the matrices $\mathbf{Q}$, $\mathbf{y}$ and $\mathbf{1}$ from $\mathcal{D}$ and $\Phi(\mathbf{x}_i, \mathbf{x}_j)$
3. solve the quadratic programming problem to get:

$$\alpha^* = [\alpha_1^*, \cdots \alpha_N^*]^\mathsf{T} \text{ and } b^*$$

4. evaluate the learned model as:

$$y = \text{sign}\left(\sum_{i=1}^{N} \alpha_i^* y_i \Phi(\mathbf{x}_i, \mathbf{x}) + b^*\right)$$

↳ Multiclass SVMs

What if there are more than 2 classes for classification?

Assignment - 2

SVMs

$$l_i = \max(0,\ 1 - (y_i \cdot (w \cdot x_i + b)))$$

(where $y_i f(x_i) \geq 1$, $l_i = 0$)

$$J = \frac{1}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} l_i = \frac{1}{2} w^T w + \frac{1}{n}\sum_{i=1}^{n} l_i$$

$$\frac{\partial J}{\partial w_j} = w_j - \frac{1}{n}\left(\sum_{y_i f(x_i) < 1} y_i \cdot x_j^i\right)$$

$$\frac{\partial J}{\partial b} = -\frac{1}{n}\sum_{y_i f(x_i) < 1} y_i$$

$$y = w \cdot x + b$$

Plane: $y = w \cdot x + b$