Page :
Date : / /

* If ~~not~~ data is not linearly separable

→ Cannot go for zero training error.
→ Still learn a maximum margin hyperplane
1) Allow some examples to be misclassified.
2) Allow some examples to fall inside the margin

* Introducing Slack Variables.

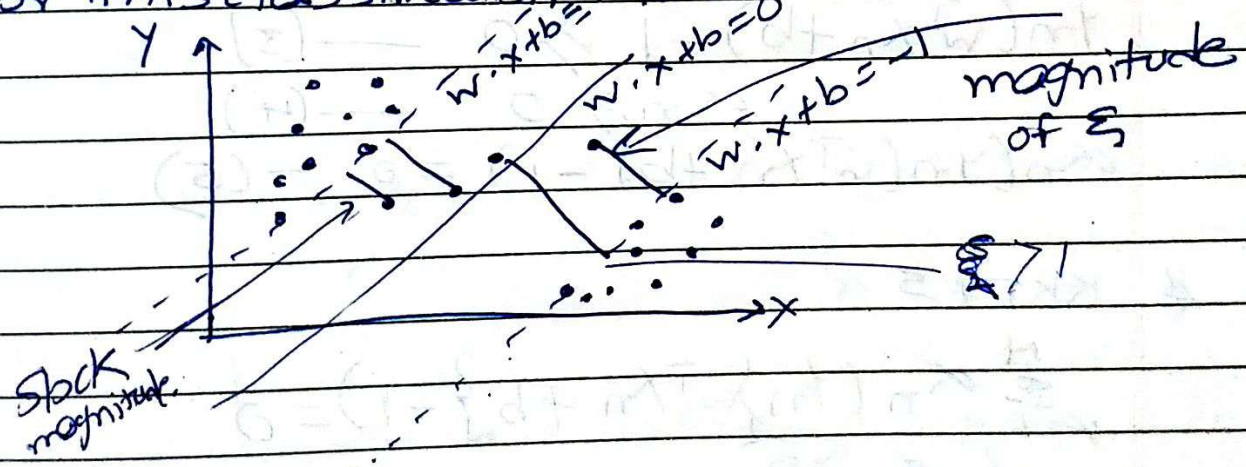
→ Separable case: To ensure zero training loss, constraint was:

$$y_n(w^T x_n + b) \geq 1 \quad \forall n = 1 \cdots N$$

→ Non-separable case: Relax the constraint

$$y_n(w^T x_n + b) \geq 1 - \xi_n \quad \forall n = 1 \cdots N$$

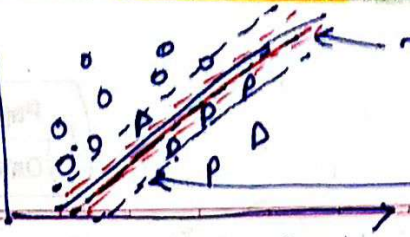
→ ξ_n is called slack variable ($\xi_n \geq 0$) is introduced for non separable
→ for misclassification, $\xi_n \geq 1$



* Relaxing the Constraint.

→ It is ok to have some misclassified training example
  → Some ξ_n is will be non-zero.
→ minimize the number of such example
  → minimize $\sum_{n=1}^{N} \xi_n$
→

C control the two
If we want margin larger but slack is less



margin is small $\frac{2}{\|w\|^2}$ small
ξ_n - slack penalty - small
$\frac{2}{\|w\|^2}$ is larger ξ_n is large

Optimization Problem for Non-Seperable Case:

$$\underset{w,b}{\text{maximize}} \quad f(w,b) = \|w\|^2 + C \sum_{n=1}^{N} \xi_n$$

Subject to $y_n(w^T x_n + b) \geq 1 - \xi_n, \quad \xi_n \geq 0 \quad n = 1 \cdots N$

C controls the impact of margin & the margin error

* The Kahrun-Kuhn-Tucker Conditions (5)

$$\frac{\partial}{\partial w} L_p(w,b,\alpha) = w - \sum_{n=1}^{N} \alpha_n y_n x_n = 0 \quad -(1)$$

$$\frac{\partial}{\partial b} L_p(w,b,\alpha) = -\sum_{n=1}^{N} \alpha_n y_n = 0 \quad -(2)$$

$$y_n(w^T x_n + b) - 1 \geq 0 \quad -(3)$$

$$\alpha_n \geq 0 \quad -(4)$$

$$\alpha_n(y_n(w^T x_n + b) - 1) = 0 \quad -(5)$$

* KKT #5

$$\sum_{n=1}^{N} \alpha_n (y_n\{w^T x_n + b\} - 1) = 0$$

& $\alpha_n \geq 0$

$$\alpha_1(y_1\{w^T x_1 + b\} - 1) + \alpha_2(y_2\{w^T x_2 - b\} - 1) + \cdots = 0$$

$\forall n \; \alpha_n \geq 0$

& $\forall n \; (y_1(w^T x_1 + b) - 1) \geq 0$

$\Rightarrow \alpha_n(y_n\{w^T x_n + b\} - 1) = 0$

1) If $\alpha_n = 0$ then $(y_n\{w^T x_n + b\} - 1)$ can be ≥ 0

2) If $\alpha_n > 0$ then $(y_n(w^T x_n + b) - 1)$ will be 0

x_n is on the margin.

* Key Observation from Dual formulation

→ KKT conditions $\neq 5$

$\alpha_n (y_n (w^T x_n + b) - 1) = 0$

→ If x_n not on margin

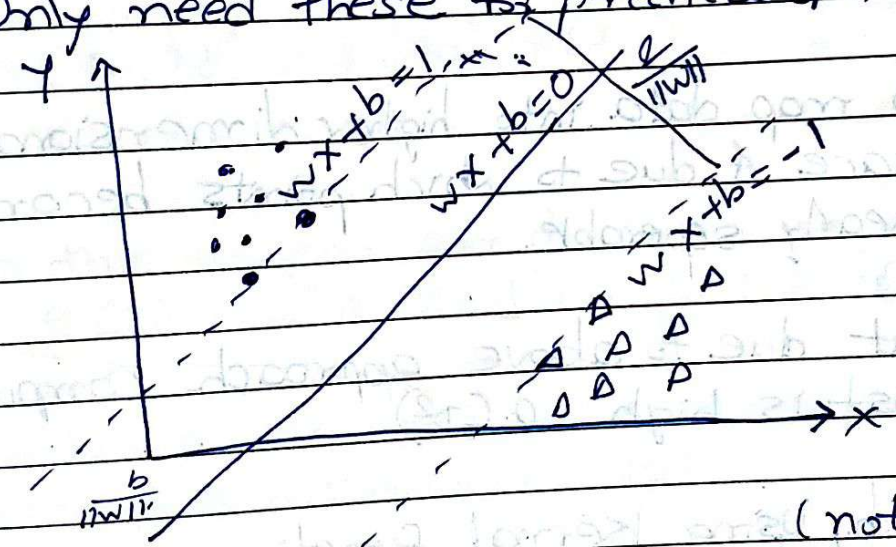
$y_n (w^T x_n + b) > 1$

$\Rightarrow \alpha_n = 0$

→ $\alpha_n \neq 0$ only for x_n on margin

→ These are the Support vectors

→ Only need these for predication.



* In SVM training need to save α_n & b (not w)

2) But not all α_n only the $\alpha > 0$ means the α for Support vector

$$= \sum_{\alpha_n > 0} \alpha_n^* y_n (x_n^T x^*) + b$$

↑ for only Support vector

Estimating weights

$$w = \sum_{n=1}^{N} \alpha_n x_n y_n$$

Page :
Date : / /

1) Points on the margin ($\xi_n = 0$)
2) Inside the margin but on the correct side ($0 < \xi_n < 1$)
3) On the wrong side of the hyperplane ($\xi_n > 1$)

Nonlinear SVM & Kernel function SVM

1) x is input feature attribute which are truly non seperable.

2) Suppose x is our ilp features. We can use mapping to map x to $\phi(x)$

$x \rightarrow \phi(x)$ means transform into new feature space.

3) Due to this it is possible ilp feature are linearly seperable in the transformed feature space.

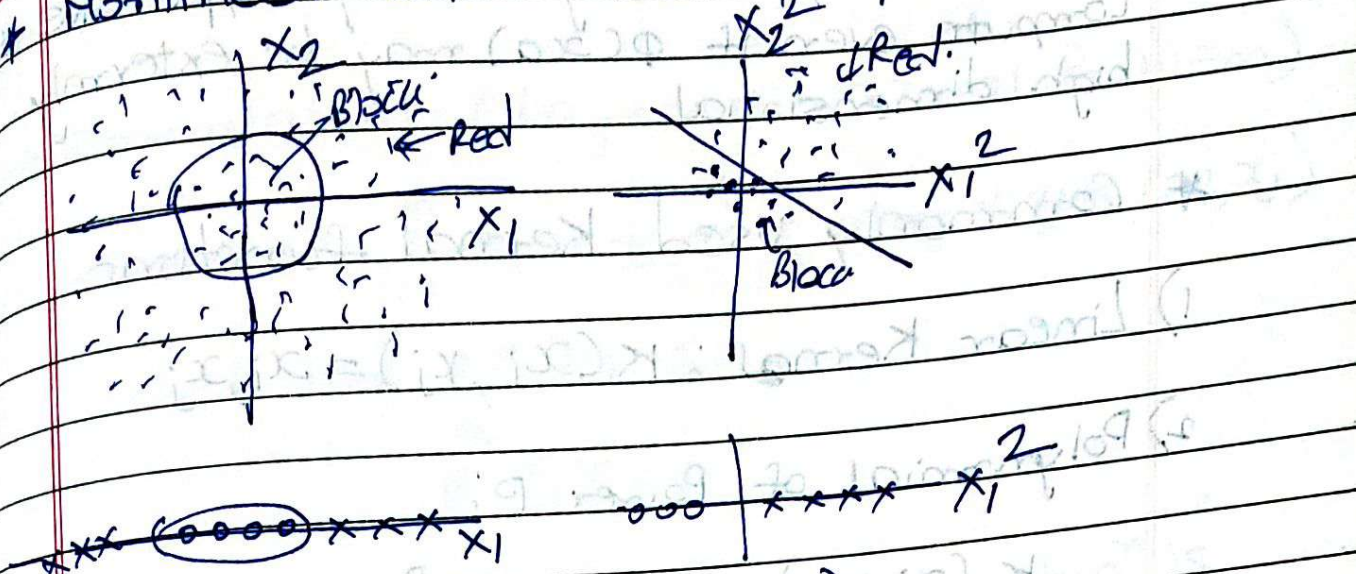
4) So map data into higher dimensional feature space & due to which points become linearly seperable.

5) But due to above approach computational cost is high $O(D^2)$

6) So by using Kernal function we can achie this transformation without any major implication on computational cost

Page :
Date : / /

* Nonlinear SVM - feature Space.



$$\phi : x \rightarrow \phi(x)$$

So here we can observe if conver feature into high dimensional feature space, the points become linearly seperable.

* Kernal function.

$$K(x_a, x_b) = \phi(x_a) \cdot \phi(x_b)$$

The Kernal Trick

1) With this mapping, our discriminant function

$$g(x) = w^T \phi(x) + b = \sum_{i \in SV} \alpha \phi(x_i)^T \phi(x) + b$$

2) We only use the dot product of feature vectors in both the trainning & test.

3) A Kernal function is def^n as a function that correspond to a dot product of two feature vectors in some expanded feature space

$$K(x_a, x_b) = \phi(x_a) \cdot \phi(x_b)$$

often $K(x_a, x_b)$ may be very inexpensive to compute even if $\phi(x_a)$ may be extermly high dimensional

* Commonly used Kernal functions

1) Linear Kernel : $K(x_i, x_j) = x_i, x_j$

2) Polynomial of Power P:

$$K(x_i, x_j) = (1 + x_i, x_j)^P$$

3) Gaussian Radial-Basis function

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

4) Sigmoid

$$K(x_i, x_j) = \tanh(\beta_0 x_i \cdot x_j + \beta_1)$$

* In general, functions that qualify mercers cond^n. is kernal fun^n.

* Kernal function measures some similarity bet^n instances x_i & x_j & this kernal functions use.

* mercer's cond^n states that any positive semi-definite kernel $f(x, y)$ ie

$$\sum_{i,j} K(x_i, x_j) c_i c_j \geq 0$$

can be expressed as a dot product in a high dimensional space

* Nonlinear SVM - Optimization.

→ Formulation (Lagrangian Dual Problem)

maximize $\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$

such that $0 \leq \alpha_i \leq C$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

The solution of the discriminant fun^n is

$$g(x) = \sum_{i \in SV} \alpha_i K(x_i, x_j) + b$$

* Multi - Class Classification.

→ SVM can only handle two - class outputs.

→ Learn N SVM's.

- SVM 1 learns Class1 vs Rest
- SVM 2 learns Class2 vs Rest
- :
- SVM N learns ClassN vs Rest.

Then to predict O/P for a new i/p, just predict with each SVM & find out which one puts the prediction the furthest into the positive region