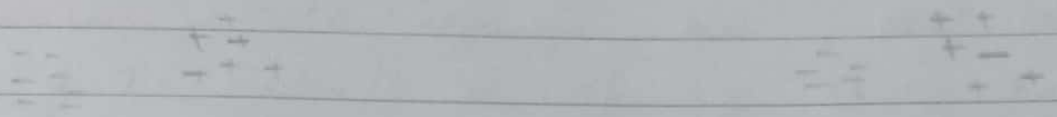
Here we are imposing a strict constraint that no data point is closer than 1 from the D.B. This is called hard margin classifer

With primal approach, if data is not linearly separable by hyperplane, then this method gives no solution

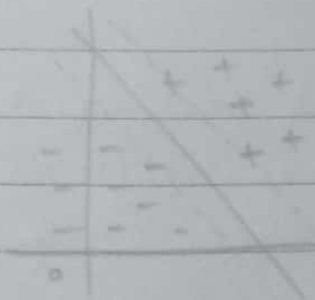


linearly separable Not linearly separable

↓

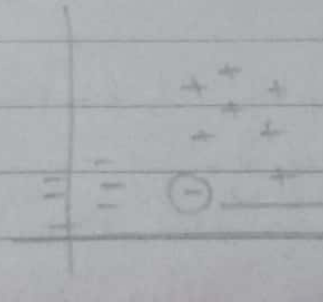
At least for some points, the constraint $y^{(i)}(\beta^T x^{(i)} + \beta_0) \geq 1$ is not satisfied

In order to solve this problem when data is not linearly separable, we use softmargin classifier. (SVM with regularisation)

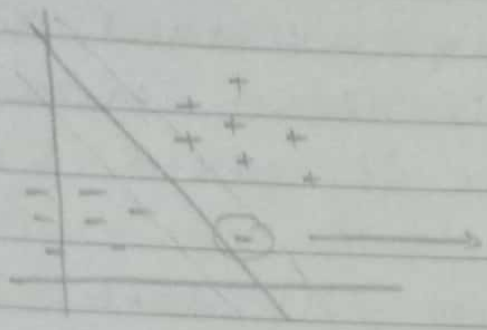


$\beta^T x + \beta_0 = 0$

→ when we do not have circle a nice case with clear boundary, use soft margin



→ outlier : far from its group (makes margin very narrow)



we let go and make an error in this case. But in turn we get a linear decision boundary with a reasonably good margin

so when no. of outliers and the points lying in completely wrong group are less, we ignore those as errors and still go for a linear D.B

Soft margin SVM (SVM with regularisation)

Problem formulation - we still get hyperplane taking care of outliers and non linear data

$\min_{\beta} \frac{1}{2} \|\beta\|^2$ such that $y^{(i)} (\beta^T x^{(i)} + \beta_0) \geq 1 - \xi_i$ for all $i = 1 \dots m$.

$\xi_i \geq 0$ is slack parameter

But here we do not get unique β^* and β_0^* because we can have different ξ_i. Hence, we add other term (regularisation)

$$\min_{\beta, \xi_i} \left[\frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^{m} \xi_i \right]$$

Penalize when they occur i.e. add weight to a function which is to be minimised

C: Regularisation parameter (chosen, not estimated)

outliers and misclassified

For softmax :—

→ Objective function : $\min \frac{\|\beta\|^2}{2} + C \sum_{i=1}^{m} \xi_i$

→ constraints : (1) $y_i (\underline{\beta}^T x_i + \beta_0) \geq 1 - \xi_i$

(2) $\xi_i \geq 0 \quad , \forall i$

Primal variables — $\underline{\beta}$, β_0, ξ_i

* Dual variables — α_i, λ_i (2 constraints)

→ Lagrangian is given as follows —

$$L(\underline{\beta}, \beta_0, \xi_i, \alpha_i, \lambda_i) = \frac{1}{2}\|\underline{\beta}\|^2 + C \sum_{i=1}^{m} \xi_i +$$

$$\sum_{i=1}^{n} \alpha_i \left[1 - \xi_i - y_i (\beta^T x_i + \beta_0) \right] + \sum_{i=1}^{m} \lambda_i \underbrace{\left[-\xi_i \right]}_{\leq 0}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \lambda_i = 0$$

For dual problem formulation, replace the primal variables.

$$L = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \underline{x_i^T x_j} + C\sum_{i=1}^{m} \xi_i + \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \alpha_i \xi_i - \sum_{i=1}^{m} \lambda_i \xi_i - \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \underline{x_i^T x_j}$$

$$= \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \underline{x_i^T x_j} + (\underbrace{C - \alpha_i - \lambda_i}_{=0}) \sum_{i=1}^{m} \xi_i$$

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

with $\alpha_i \geq 0$

$\lambda_i \geq 0$. But no λ_i in this expression so using $\lambda_i = C - \alpha_i$ we get $C - \alpha_i \geq 0$

Hence $\alpha_i \leq C$.

So $0 \leq \alpha_i \leq C$ → (the λ_i written in terms of α_i)

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

KKT Conditions

$$\frac{\partial L}{\partial \beta} = 0 \; ; \; \frac{\partial L}{\partial \beta_0} = 0 \; ; \; \frac{\partial L}{\partial \xi_i} = 0$$

$y_i (\beta^T x_i + \beta_0) \geq 1 - \xi_i$
$\xi_i \geq 0$
Primal feasibility

$\alpha_i \geq 0$
$\lambda_i \geq 0$
Dual feasibility

$\alpha_i [y_i (\beta^T x_i + \beta_0) - 1 + \xi_i] = 0$
$\lambda_i \xi_i = 0$
Complementary slackness

$\alpha_i > 0 \Rightarrow y_i (\beta^T x_i + \beta_0) = 1 - \xi_i$
Two possibilities here

A) $\xi_i > 0 \Rightarrow \lambda_i = 0 \Rightarrow \alpha_i = \gamma$
B) $\xi_i = 0 \Rightarrow \lambda_i$ could be greater than 0
so $\gamma - \alpha_i > 0$
$\alpha_i < \gamma$

so all $\alpha_i > 0$ are support vectors.
Those in B) are on the margin
and those that don't violate margin
Those in A) may violate margin

→ As ξ_i is close to zero, we approach hard margin.

→ We want to minimize the entire function, so when c is very large, ξ_i s are small hence we are close to hard margin (close to zero)

→ As the value of c decreases, the margin becomes narrower.

For hard margin problem, dual problem formulation is:

$$\max_{\lambda_i} \sum_{i=1}^{m} \lambda_i - \sum_i \sum_j d_i d_j y^{(i)} y^{(j)} x^{(i)T} x^{(j)}$$

with $\lambda_i \geq 0$

$\sum \lambda_i y_i = 0$.

→ For soft margin, dual problem is:

$$\max_{\lambda_i} \sum_{i=1}^{m} \lambda_i - \sum_i \sum_j \lambda_i \lambda_j y^{(i)} y^{(j)} x^{(i)T} x^{(j)}$$

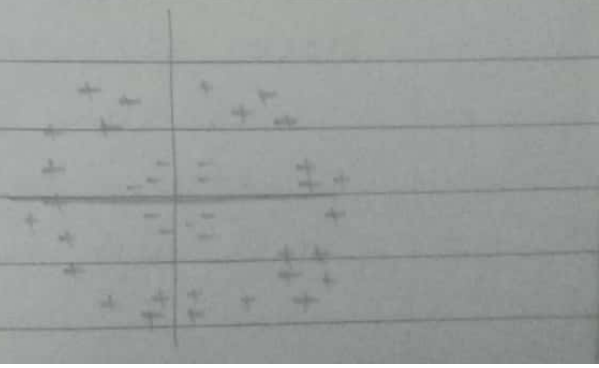
with $0 \leq \lambda_i \leq c$

$\sum \lambda_i y_i = 0$

Proof:

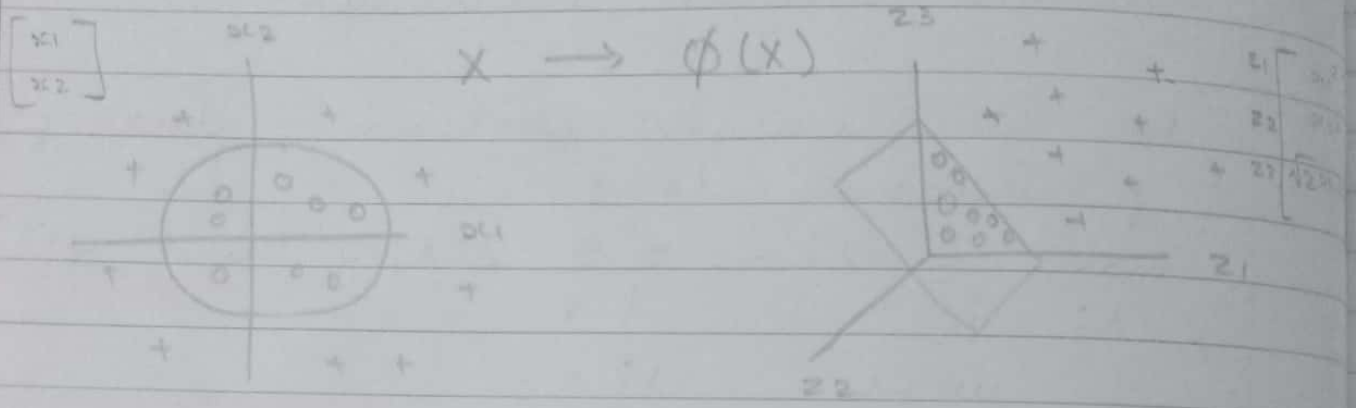
SVM with Kernel function

→ This is used for nonlinearly separable data point. (The best solution - a general method that can be used



→ The trick used here is kernel trick - convert data from lower dimension to higher dimension [Eg. from a vector in 2D to a vector in 4D]

→ After the conversion, the data now becomes separable and hence we can fit a decision boundary.



(2D) (3D)

(Non linear D.B) (linear plane DB)

→ ϕ function need not be known

→ Kernel function — Radial basis function (RBF)

Hard margin :

→ Data has to be linearly separable

Soft margin :

→ Slack variables ξ_i are used

→ Takes care of non linear separation and outliers can be handled well

→ Still not the best classifier

Using Kernel function :

→ Best SVM classifier.

→ When data is not linearly separable project

onto higher dimensional space where the data becomes linearly separable (Then we can use concepts of hard margin / soft margin classifier for problem formulation) The data now used is $\phi(x)$. But ϕ itself is not used due to the use Kernel function

A Kernel function is given by:

$K(\underline{x}, \underline{z}) = [\phi(\underline{x})]^T [\phi(\underline{z})]$

Thus K performs dot product of $\underline{x}$ and $\underline{z}$ not in lower dimension but in higher dimension. (Here $\underline{x}$ and $\underline{z}$ are 2 examples)

Eg. $\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ $\underline{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$

$\phi(\underline{x}) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2} x_1 x_2 \end{bmatrix}$ $\phi(\underline{z}) = \begin{bmatrix} z_1^2 \\ z_2^2 \\ \sqrt{2} z_1 z_2 \end{bmatrix}$

$\phi(\underline{x})^T \phi(\underline{z}) = x_1^2 z_1^2 + x_2^2 z_2^2 + 2 x_1 x_2 z_1 z_2$

$= (x_1 z_1 + x_2 z_2)^2$

Let us consider $K(\underline{x}, \underline{z}) = (x^T z)^2$

So $K(\underline{x}, \underline{z}) = (x_1 z_1 + x_2 z_2)^2$

Thus, the kernel operates on lower dimension data, but is actually doing dot product in higher dimension data

One of the popular kernel $K(\underline{x}, \underline{z})$ is called radial basis function (RBF) or

$K(x, z) = e^{-\|x - z\|^2 / 2\sigma^2}$ where σ^2 must be known

Intuitively, Kernel function is finding out similarity in x and z

When x and z are similar, $x - z$ is small (close to zero) hence $K(x, z)$ will be close to one (highest possible). This makes sense because when x and z are similar, their dot product should be high.

Similarly, $K(x, z)$ is small when x and z are not similar

After converting in higher dimension, we formulate using dual approach in hard margin

$$\max_{\lambda_i} \sum_{i=1}^{m} \lambda_i - \sum_{i=1}^{m} \sum_{i=1}^{m} \lambda_i \lambda_j y^{(i)} y^{(j)} \underbrace{K(x^{(i)}, x^{(j)})}_{\downarrow}$$

$\phi(x^{(i)})^T \phi(x^{(j)})$

($x^{(i)T} x^{(j)}$ becomes $\phi(x^{(i)})^T \phi(x^{(j)})$ in higher dimension now)

(We do not worry about what the higher dimension is. Kernel function will ultimately give a single value only)

Now, $\underline{\beta} = \sum_{i=1}^{m} \lambda_i^* y^{(i)} \phi(x^{(i)})$

But now, there is a problem here because ϕ is unknown. Not a major problem because we actually want the decision boundary not β and β_0

Hence $\sum_{i=1}^{m} \lambda_i^* y^{(i)} [\phi(x^{(i)})]^T \phi(z) + \beta_0 < 0$

$\phi(x^{(i)})]^T \phi(z) \rightarrow K(x^{(i)}, z)$

test example

So now if $\beta \phi^T(z) + \beta_0 > 0$ — classify as 1

$\beta \phi^T(z) + \beta_0 < 0$ — classify as 0

We still need to find β_0.

suppose $x^{(i)}$ is an example in the training set with non zero λ_i, i.e. $x^{(i)}$ is a support vector.

So $\beta^T \phi(x^{(i)}) + \beta_0 = 1$

$\sum_{j=1}^{m} \lambda_j y^{(j)} [\phi(x^{(j)})]^T \phi(x^{(i)}) + \beta_0 = 1$

$[\phi(x^{(j)})]^T \phi(x^{(i)}) \rightarrow K(x^{(j)}, x^{(i)})$

$\beta_0 \rightarrow$ Find from here

Here we are using gaussian kernel.

There are other types of kernel as well:

① Polynomial kernel $K(x^{(i)}, x^{(j)}) = (1 + x^{(i)T} x^{(j)})^p$

where p is an integer > 0

② linear kernel $K(x^{(i)}, x^{(j)}) = x^{(i)T} x^{(j)}$

(used in original problem)