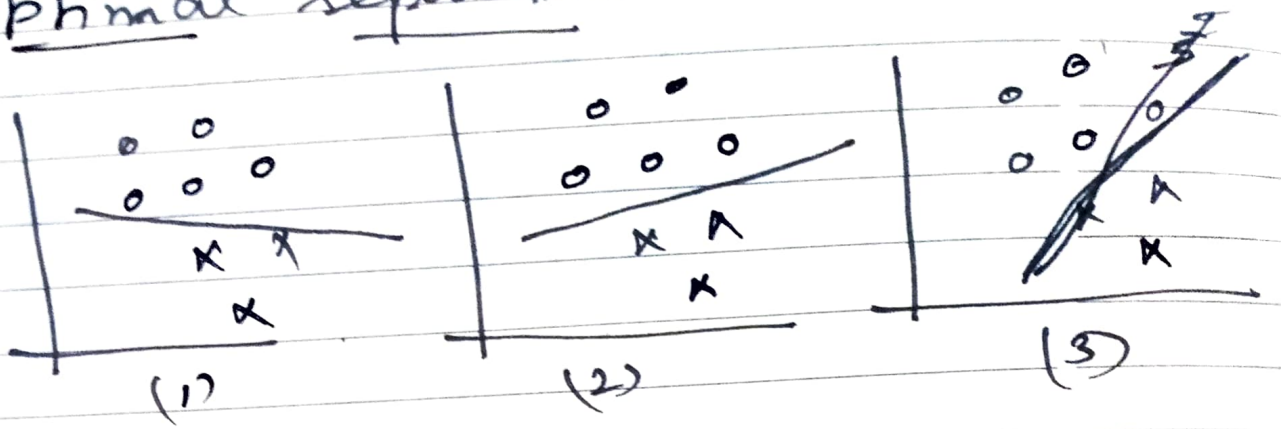
SVM

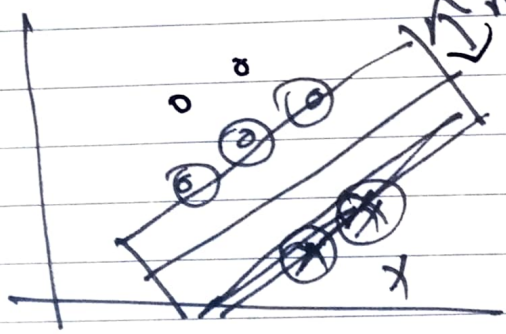
- Perceptron — learns the weights for the linearly separable data. If the data are not linearly separable, perceptron fails to learn.
- But, If the dimentionality (data representation) is changed (from lower to higher), perceptron is able to learn the weights.
- We did the XOR problem with perceptron by adding extra dimension and moved a point that could not classify properly into the additional dimension so that we could linearly separate the classes.
- kernal function is used to do the same.
- SVM is a ML alg used to provide better performance for ~~reasonably~~ reasonably sized data set.
- dont work well for the large size dataset.
- It is very expensive

Optimal separation



- If we pick a line from (1) / (2), there is a chance that a data point from one class will be on the wrong side of the line.

Margin and support vector.



maximum margin classifier (M).

The data point in class lies closes to the classification line is called support vectors.

- margin should be as large as possible
- Support vectors are the most useful

8 data points. we can keep only these support
9 vector data points after training and
throw away all other data points.
10
- Support vector data points alone used for
11 classification.

12 - We have weight vector w and input vector x.

13 - classifier line $y = wx + b$
- any x value gives +ve value for $y = wx+b$
14 is above a line $\in$ '+' ve class.
15 - any x value gives -ve value for $y = wx+b$
16 is below a line $\in$ '-' ve class.

17 $w.x = \sum_{i=1} w_i x_i = w^t x.$

18 - any point x where $w^t x + b \geq M$ is +ve
19 $\leq -M$ is -ve

- actual separating hyplane is given by
$w^t x + b = 0.$
- pick a x^+ point on the +ve class boundary
line $w^T x^+ = M$. . This is Support vector

- weight vector w is ⊥ to the classifier line $y = w^t x + b$.
- we make w a unit vector
- so margin is $1/\|w\|$.
- we need to find a and b from the training data, find the maximum margin M. with min of $w^t.w$
- set $w=0$, max margin M and unit

Constrained optimization problem

- target answer for 2 classes be ± 1 instead of 0/1.
- $t_i * y_i$ will be +ve if both are same else -ve.
- $t_i (w^t x + b) \geq 1$

t_i	y_i	
−	−	+
+	+	+
−	+	−

min $1/2 w^t.w$ subject to

$t_i (w^t x + b) \geq 1$ for all $i = 1, 2, \ldots$

Appointments

Karush - Kuhn - Tucker condition.

will be satisfied for the optimal parameters.

$$\lambda_i^* (1 - t_i (w^{*t} x_i + b^*)) = 0$$

$$1 - t_i (w^{*t} x_i + b^*) \leq 0$$

$$\lambda_i^* \geq 0$$

λ_i is lagrange multiplier.

if $\lambda_i \neq 0$ then $1 - t_i (w^{*t} x_i + b^*) = 0$

This is true for support vectors.

Lagrangian function

$$\mathcal{L}(w, b, \lambda) = \frac{1}{2} w^t w + \sum_{i=1}^{n} \lambda_i (1 - t_i (w^t x_i + b_i))$$

$$= \frac{1}{2} w^t . w + \sum_{i=1}^{n} \lambda_i - \sum_{i=1}^{n} t_i w_i^t x_i . \lambda_i + \sum_{i=1}^{n} \lambda_i b_i t_i \quad -(3)$$

diff w.r.t w.

$$\nabla_w \mathcal{L} = w - \sum_{i=1}^{n} \lambda_i t_i x_i = 0.$$

$$\boxed{w = \sum_{i=1}^{n} \lambda_i t_i x_i} \quad -(1)$$

diff. w.r.t. b

$$\nabla_b \mathcal{L} = \sum_{i=1}^{n} \lambda_i t_i = 0$$

$$\boxed{\sum_{i=1}^{n} \lambda_i t_i = 0} \quad -(2)$$

Appointment:

Sub (1) and (2) into eq (3)

$$L(w, b, \lambda) = \frac{1}{2} w^t . w + \sum_{i=1}^{n} \lambda_i - \sum_{i=1}^{n} \lambda_i t_i w^t . x_i - \sum_{i=1}^{n} \lambda_i t_i b$$

$$= \frac{1}{2} \sum_{i=1}^{n} \lambda_i t_i x_i \lambda_j t_j x_j - \sum_{i=1}^{n} \lambda_i t_i x_i \lambda_j t_j x_j + \sum_{i=1}^{n} \lambda_i \quad (0)$$

$$= -\frac{1}{2} \sum_{i=1}^{n} \lambda_i \lambda_j . t_i t_j . x_i x_j + \sum_{i=1}^{n} \lambda_i$$

This eq is dual problem. Constraint is $\lambda_i \geq 0$ for all i and $\sum_{i=1}^{n} \lambda_i t_i = 0$

- for the support vector $t_i(w^t x_i + b) = 1$

$$L = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum \lambda_i \lambda_j t_i t_j x_i x_j$$

22 Sunday

max L, always +ve

- if $\lambda_i = 0$, then x_i is not S.V.
- if λ_i is very high for S.V.
  - high influence for the position of hyperplane

8 If λ_i is extremely large, x_i is spurious.

9 $$b^* = \frac{1}{N_S} \sum_{\text{support vectors } j} \left(t_j - \sum_{i=1}^{n} \lambda_i t_i x_i^T x_j \right).$$

10

11 - to make a prediction, for a new point z

12 $$w^{*T} \cdot z + b^* = \left(\sum_{i=1}^{n} \lambda_i \cdot t_i x_i \right)^T z + b^*$$

13 Slack variables for non-linearly separable problems.

14

- Introduce slack variables $\eta_i \geq 0$ to resolve non-linearly separable prob., so constraint becomes: $t_i (w^T x_i + b) \geq 1 - \eta_i$, for inputs that are correct, we set $\eta_i = 0$

15

16

17

- minimize $w^t \cdot w + C \alpha$. (distance between misclassified pts from the boundary line)
- if C is small, large margin, err-less.
- if C is large, small " , err-high.
- turn this into soft margin classifier

18

19

$$\lambda(w, \epsilon) = w^t \cdot w + C \sum_{i=1}^{n} \epsilon_i$$

$0 \leq \lambda_i \leq C$., supports vectors for those with $\lambda_i \geq$

KKT constraint becomes.

$$\lambda_i^* (1 - t_i (w^{*T} x_i + b^*) - \eta_i) = 0 \quad -(1)$$

$$(C - \lambda_i^*) \eta_i = 0 \quad -(2)$$

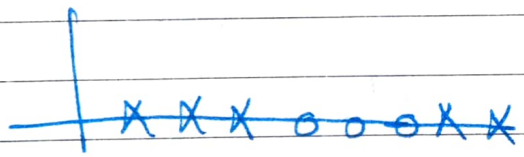
$$\sum_{i=1}^{n} \lambda_i^* t_i = 0. \quad -(3)$$

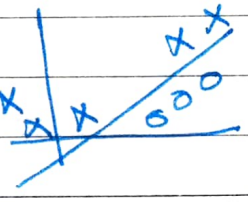
In (2) if $\lambda_i < C$, then $\eta_i = 0$. which means that they are support vectors.

Kernals

$$w^T x + b = \left(\sum_{i=1}^{n} \lambda_i t_i \phi(x_i) \right)^T \phi(z) + b$$

↑ the dimensions.





$x_1^2 + x_1$

$x_1, x_2 \cdots x_d$, input $x_1^2, x_2^2 \cdots x_d^2$

$x_1 x_2, x_1 x_3 \cdots x_{d-1} x_d$.

$\phi(x_i)$ — to replace x_i in higher dim.

If $d = 3$

$$\phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_3, x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3)$$

$$\phi(x)^T \cdot \phi(x) = 1 + 2\sum_{i=1}^{d} x_i y_i + \sum_{i=1}^{d} x_i^2 y_i^2 + 2 \cdot \sum_{i,j=1, i<j} x_i x_j y_i y_j$$

$\Downarrow$ reduced to

$$(1 + x^T y)^2 .$$

kernals.

1) polynomials up some degree s, in element x_k

$$k(x, y) = (1 + x^T y)^s .$$

for $s=1$, linear kernal.

2) sigmoid function of x_k's. with parameters $k \in \delta$

$$k(x, y) = \tanh(k x^t y - \delta)$$

3) rbf. kernal $k(x, y) = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right)$

kernels in 2-d.

Extension of SVM

C1 C2 C3.

C1 / C2, C3 (SVM1) C2 / C1, C3 (SVM2) C3 / C1, C2 (SVM3)

y longest (farthest away) from the boundary in +ve side.