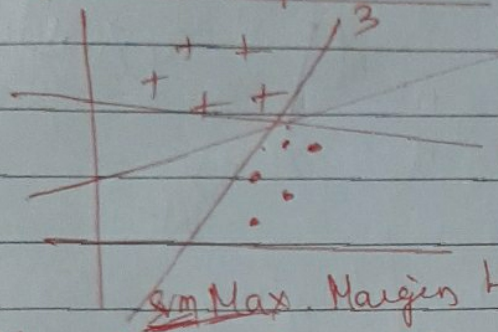
Support Vector Machine (SVM)

- Vapnik in 1992
- work well on reasonably sized DS not on large DS
- Comp. expensive

I Optimal Separation



2 → middle of separation blw data points from 2 class (equidistant)

3 → Max. Margin Linear classifier

① Margin & Support vectors.

→ Measure the distance that we have to travel away from the line before we hit a dp (datapt.)

- symmetric - The largest radius - Margin
- M → affects the speed at which (P) converges
- The datapoints in each class that lie closest to the classification line → support vectors

Best classifier -

1) M should be as large as possible
2) SV are the most useful DP: after training we can throw all of the data except SV.

Optimal decision boundary

weight vector, input vector x

$y = w \cdot x + b$ → bias weight

$w \cdot x = \sum_i w_i x_i$

the value → '+' class

-ve value → '0' class.

$x \to x_1$

$y \to x_2$

$ax_1 - x_2 + b =$

$w_1 x + b = 0$

$(a, -1)$

for a given marginal value M,

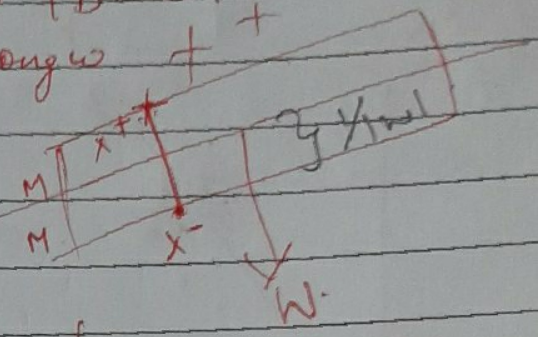
any point x where $w^T x + b \geq M \rightarrow +$

$w^T + b \leq -M \rightarrow -$

hyperplane $\Rightarrow w^T + b = 0$

direction from x^+ to x^- is along w

w is a unit vector $\frac{w}{\|w\|}$



Margin is $\frac{1}{\|w\|}$

To keep margin large (M) same as making $w^T w$ small

find boundary classifies well & also making $w^T w$ small

$\Downarrow$

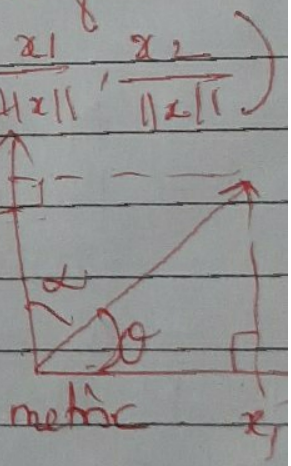
$\frac{1}{2} w^T w$ subject to some constraints

length of vector x $\Rightarrow$ norm

$\|x\| = \sqrt{x_1^2 + \dots + x_n^2}$

direction of v

$w = \left(\frac{x_1}{\|x\|}, \frac{x_2}{\|x\|} \right)$



$\Rightarrow$ absolute value of $(wx+b)$

But we fail to distinguish how good and bad hyperplane $\Rightarrow$ adjust this metric

② Constrained Optimisation Problem:

- Consider target answers $+1, -1$ — $t_i y_i$ — $+$ / $-$ $+$ $-$

$\min \frac{1}{2} w^T w$ subj to $t_i (w^T x_i + b) \geq 1$ for all i

$\Downarrow$

quadratic prog

problem of finding w & b is called Optimisation problem.

SVM Lagrange problem

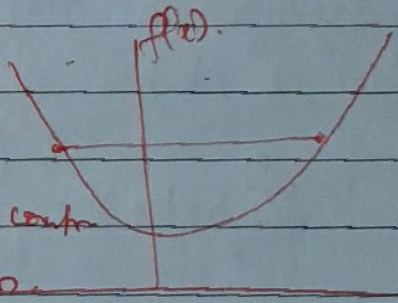
Lagrange state that if we want to find min of f under the equality constraint g, we just need to solve for

$$\nabla f(x) - \nabla g(x) = 0.$$

$\nabla f(x) \rightarrow \frac{1}{2} w^T w$

$\nabla g(x) \rightarrow$ lag. multiplier $\rightarrow t_i (w^T x_i + b) = 1$

To find unique optimum using Quad. prog $\rightarrow$ convex with linear constr.



unique minimum.

do this in poly time.

When we find that uniq opt.: KKT (Karush-Kuhn-Tucker) will be satisfied

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

$1 - t_i \ldots \leq 0.$

$\lambda_i^* \geq 0.$

$\lambda_i^* \neq 0$, so $(1 - t \ldots) = 0 \Rightarrow$ true for SV.

Lag fn.

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

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

(1)

diff w.r.t w^T & b

$$\nabla_w L = w - \sum \lambda_i t_i x_i .$$

$$\frac{\partial L}{\partial b} \text{ or } \nabla_b L = + \sum_{i=1} \lambda_i t_i$$

Set derivatives to zero to find maximum (saddle pts)

$$w^* = \sum_{i=1}^{n} \lambda_i t_i x_i \quad ; \quad \sum_{i=1}^{n} \lambda_i t_i = 0$$

Substitute this to (1)

$$L(w^*, b^*, \lambda) = \frac{1}{2} \sum_{i=1}^{n} \lambda_i t_i x_i \cdot \sum_{j=1}^{n} \lambda_j t_j x_j + \sum_{i=1}^{n} \lambda_i - \sum \lambda_i t_i [w^T x_i] - \sum_{i=1}^{n} \lambda_i t_i b$$

$$\sum_{i=1}^{n} \lambda_i t_i \to 0.$$

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

↓ This equation is dual problem (Wolfe dual lag fn)

'max it w.r.t. λ_i'.

Constraints are $\lambda_i \geq 0$ for all i.

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

then b^* is,

WKT, $t_i(w \cdot x_i + b) - 1 \geq 0$.

closest pt to HP will have functional margin 1.

$t_i(w \cdot x_i + b) = 1$

multiply both sides by t_i.

$$t_i^2(wx_i + b) = t_i \qquad \because t_i^2 = 1.$$

$$wx_i + b = t_i$$

$$b = t_i - (wx_i)$$

$$b^* = \sum_{j=SV} \left(t_j - \sum_{i=1}^{n} \lambda_i t_i x_i^T \cdot x_j \right)$$

$$= \frac{1}{N_s} \sum_{SV_j} \left(\quad \right)$$

In case of error, it is not stable, avg it o

To classify new data point(z), all SV

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

Slack Variables for Non-linearly separable problem

- Non linear data, Slack var $\eta_i \geq 0$

so the constraint becomes,

$$t_p(w^T x_i + b) \geq 1 - \eta_i$$

for correct inputs, set $\eta_i = 0$

SV → tells one classifier → mistake → put on wrong side
another → same but still longer
include this info in min. criterion
add a term $w^T w + Cx$ (distance of misclassified point from BL)

if small C → large margin with few errors

to min is,

$$L(w, \epsilon) = w^T w + C \sum_{i=1}^{n} \epsilon_p$$

dual problem is same except $0 \leq \lambda_i \leq C$.

SV with $\lambda_i > 0$

KT condition,

$$\lambda_i^* (1 - t_p(w^{*T} x_p + b^*) - \eta_i) = 0$$

$$(C - \lambda_i^*) \eta_i = 0 \rightarrow \lambda_i < C, \text{ the } \eta_i = 0$$

$$\sum_i \lambda_i^* t_i = 0$$

$\lambda_i = C$, if $\eta_i \geq 1$ then

Classifier make a mistake → SV

Slack Variables for Non-linearly separable problem

- Non linear data, Slack var $\eta_i \geq 0$

So the constraint becomes,

$$t_p(w^T x_p + b) \geq 1 - \eta_p$$

for correct inputs, set $\eta_i = 0$

→ SV → tells one classifier → mistake → put on wrong side
another → same but still longer

→ include this info in min. criterion

→ add a term $w^T w + Cx$ (distance of misclassified point from BL)

if small C → large margin with few errors

fn to min is,

$$L(w, \epsilon) = w^T w + C \sum_{i=1}^{n} \epsilon_p$$

derivation dual problem same except $0 \leq \lambda_i \leq C$.

& SV with $\lambda_i > 0$

KKT condition,

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

$$(C - \lambda_i^*) \eta_i = 0 \rightarrow \lambda_i < C, \text{ then } \eta_i = 0$$

$$\sum_i \lambda_i^* t_i = 0$$

if $\lambda_i = C$, if $\eta_i \geq 1$ then → SV

Classifier make a mistake

Kernels: new function $\phi(x)$ of our input features

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

Kernel function is defined as a function that corresponds to a dot product of two feature vectors in some expanded feature space.

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j).$$

Polynomial of degree 2 then.

$x_1 \dots x_n, x_1^2 \dots x_n^2, x_1x_2, x_1x_3 \dots x_{d-1}x_d$.

$\phi(x) \rightarrow d^2/2$ elements. $\left(n + n + \frac{n(n-1)}{2}\right)$

Using Kernel trick, $O(d^2) \xrightarrow{\text{reduced}} O(d)$

↳ Gram Matrix / Kernel Matrix.

Kernels:

linear kernel: $K(x,y) = 1 + x^T y$.

poly k: $K(x,y) = (1 + x^T y)^s$. s-degree

Sigmoid k: $K(x,y) = \tanh(\beta_0 x^T y - \beta_1)$.

RBF k: $K(x,y) = \exp(-(x-y)^2/2\sigma^2)$.

In general, fns that satisfy mercer's theorem is kernel fns.

SVM Algo:

Initialization

- for specified kernel & kernel param, Compute kernel of distances between datapoints
  - compute $K = XX^T$
  - for linear K, return K, for poly of degree d return $\frac{1}{\sigma} K^d$.
  - for RBF, compute $K = \exp(-(x-x')^2/2\sigma^2)$

Training

- assemble constraints as matrices
  min $\frac{1}{2} x^T t_i t_j K x + q^T x$, subject to. $Gx \leq h$, $Ax = b$.
- pass these matrices to solver.
- identify SV and dispose rest of tr. data
- Compute b^* using eqn

Classification

- for given test data z, use SV to classify data for relevant kernel using
  - → Compute inner prod. of test data & SV
  - → perform classification $\sum \lambda_i t_p K(x, z) + b^*$ yielding either label or value (soft dec).

21 Friday · August
Week 34 233-132

21 · 08 · 2015

SVM

	x_i	y_i
1	1	−1
2	2	−1
3	6	+1

1 2 3 4 5 6

$$L(w) = \frac{w^2}{2} - \sum_{i=1}^{3} \lambda_i \left(y_i (w x_i + b) - 1 \right)$$

$$= \frac{w^2}{2} - \lambda_1 (-1(w \cdot 1 + b) - 1)$$

$$- \lambda_2 (-1 (w \cdot 2 + b) - 1)$$

$$- \lambda_3 (+1 (w 6 + b) - 1).$$

$$L(w) = \frac{w^2}{2} - \lambda_1 (-w - b + 1) - \lambda_2 (-2w - b - 1)$$

$$- \lambda_3 (6w + b - 1)$$

$$\frac{\partial L}{\partial w} = \frac{2w}{2} - \lambda_1 (-1) - \lambda_2 (-2) - \lambda_3 (6)$$

$$= w + \lambda_1 + 2\lambda_2 - 6\lambda_3$$

$$w = -\lambda_1 + 2\lambda_2 + 6\lambda_3 \quad \text{——} (1)$$

0

$$\frac{\partial L}{\partial b} = 0 - \lambda_1(-1) - \lambda_2(-1) - \lambda_3(1)$$

$$= +\lambda_1 + \lambda_2 - \lambda_3$$

$$\lambda_1 + \lambda_2 - \lambda_3 = 0 \quad \text{--- (2)}$$

$$\frac{\partial L}{\partial \alpha_1} = +w + b + 1 \quad \text{(3)}$$

$$\frac{\partial L}{\partial \alpha_2} = 2w + b + 1 \quad \text{(4)}$$

$$\frac{\partial L}{\partial \alpha_3} = -6w - b + 1 \quad \text{(5)}$$

(4) − (3): $2w + b + 1 = 0$

$\quad w + b + 1 = 0$

$\quad w = 0$

(4) + (5): $2w + b + 1 = 0$

$\quad -6w - b + 1 = 0$

$\quad -4w + 2 = 0$

$\quad w = -2/-4$

$\quad w = 1/2$

Sub in (3)

$w + b + 1 = 0$

$\frac{1}{2} + b + 1 = 0$

$b + \frac{3}{2} = 0$

$b = -3/2$

$wx + b = 0$

$\frac{x}{2} - \frac{3}{2} = 0$

$x - 3 = 0$

$x = 3$

Sunday 23
235-130

Ugo:

T. im

24

Monday • August

Week 35 236-129

4 • 08 • 2015

w 1/2 b -3/2

July						August					
Weeks	27	28	29	30	31	Weeks	31	32	33	34	35
Monday		6	13	20	27	Monday	31	3	10	17	24
Tuesday		7	14	21	28	Tuesday		4	11	18	25
Wednesday	1	8	15	22	29	Wednesday		5	12	19	26
Thursday	2	9	16	23	30	Thursday		6	13	20	27
Friday	3	10	17	24	31	Friday		7	14	21	28
Saturday	4	11	18	25		Saturday	1	8	15	22	29
Sunday	5	12	19	26		Sunday	2	9	16	23	30

$g(x) = wx + b$.

~~$x = 1$~~ $g(1) = \frac{1}{2}(1) - \frac{3}{2} = \frac{1}{2} - \frac{3}{2} = -1$ (-ve)

$g(2) = \frac{1}{2}(2) - \frac{3}{2} = \frac{2}{2} - \frac{3}{2} = -\frac{1}{2}$ (-ve)

$g(6) = \frac{6}{2} - \frac{3}{2} = \frac{6-3}{2} = \frac{3}{2}$ (+ve).

— d —