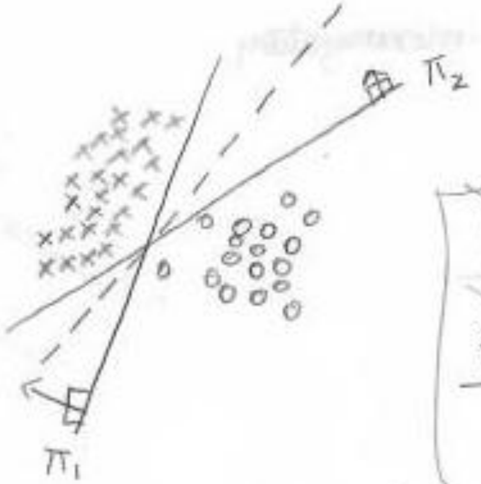
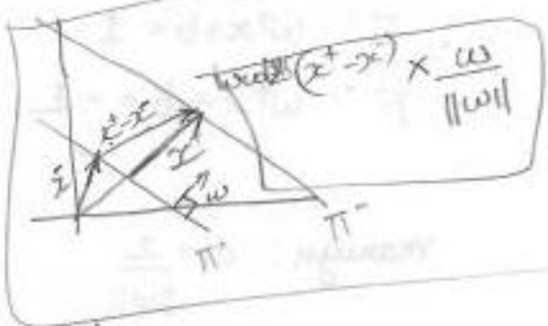
Support Vector Machines

◇ Geometric-Intution

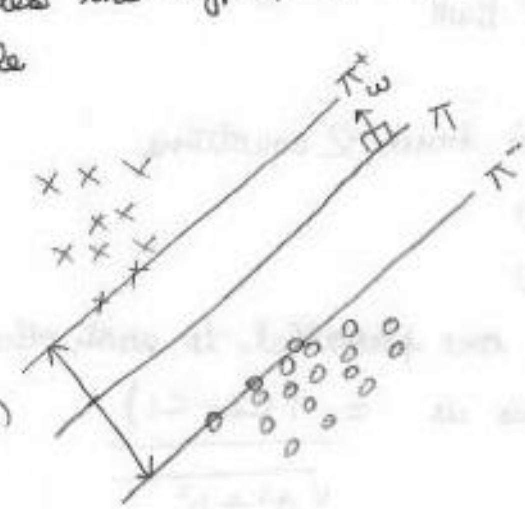


× +ve pts
o -ve pts



many πs that seperates +ve pts. and -ve pts.
A point that are very close to hyperplane would be low.

☆ Key idea of Support Vector Machine
π seperates the hyperplanes from -ve and +ve pts as widely as possible



π : margin maximising hyperplane

π^+ and π^- are both parallel to π.

SVM : Try to find a π that maximizes the margin = dist (π^+, π^-)

If my margin is high chance of misclassification decreases.
Points through which π^+ and π^- goes through are called <u>support vectors</u>.

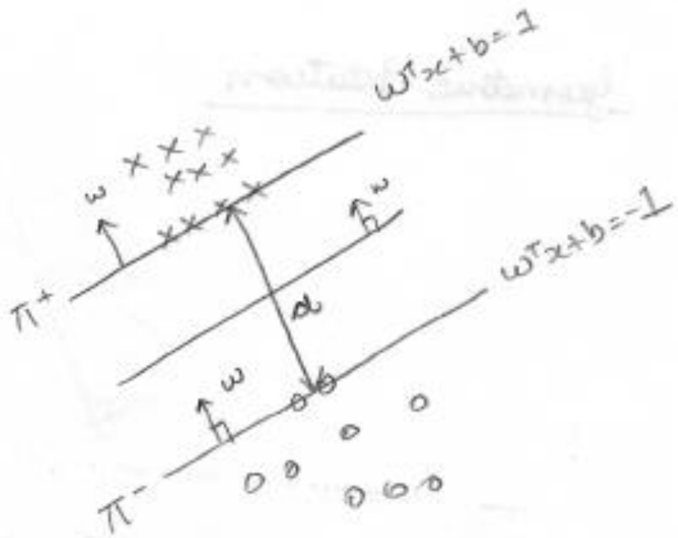
✧ Mathematical derivation

π :- margin - maximization

$\pi : w^T x + b = 0$

$\pi^+ : w^T x + b = 1$

$\pi^- : w^T x + b = -1$



margin : $d = \frac{2}{\|w\|}$

Objective :- $(w^*, b^*) = \underset{w,\,b}{\operatorname{argmax}} \frac{2}{\|w\|}$

s.t $(w^*, b^*) = \underset{w,\,b}{\operatorname{argmax}} \frac{2}{\|w\|}$ = margin.

Proof:

for example, if we have 2 equations

$ax + by + c1 = 0$

$ax + by + c2 = 0$

here both lines are parallel to each other;

So distance between lines is $= \frac{|c2 - c1|}{\sqrt{a^2 + b^2}}$

now we are using 2 hyperplanes equations

$w^T x + b = 1$

$w^T x + b = -1$

$w^T x + b - 1 = 0$

$w^T x + b + 1 = 0$

Comparing eq of plane to eq. of lines

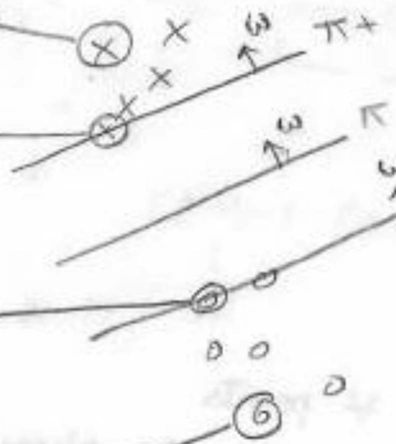
$$\text{distance between 2 planes} = \frac{|\text{difference of vector components}|}{\|w\|}$$

$$\text{diff of vector components} = |w^T x + b - 1 - w^T x - b - 1|$$
$$= |-2|$$

$$\boxed{\text{distance between 2 planes} = \frac{2}{\|w\|}}$$

$+y_i(w^T x_i + b) > 1$

$y_i(w_i^T x_i + b) = 1$



$y_i(w^T x_i + b) = 1$

$\underset{(-1)}{y_i}\ \underset{(-1)}{(w^T x_i + b)} > 1$

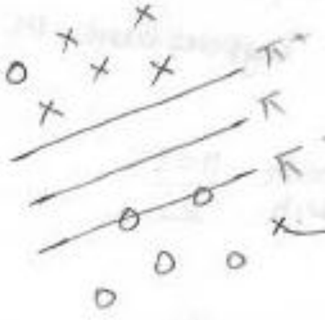
$$(w^*, b^*) = \underset{w, b}{\operatorname{argmax}} \frac{2}{\|w\|} = \text{margin}$$

$$\text{s.t. } y_i(w^T x_i + b) \geq 1 \text{ for all } x_i$$

} Constr optimization problem of SVM

works when data is linearly seperable

$\underset{+ve}{(w^T x_i + b)}\underset{-ve}{(-y_i)} < -1$

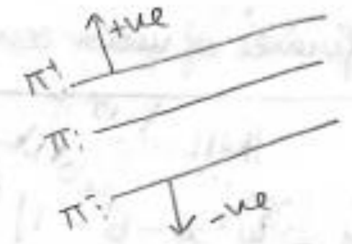


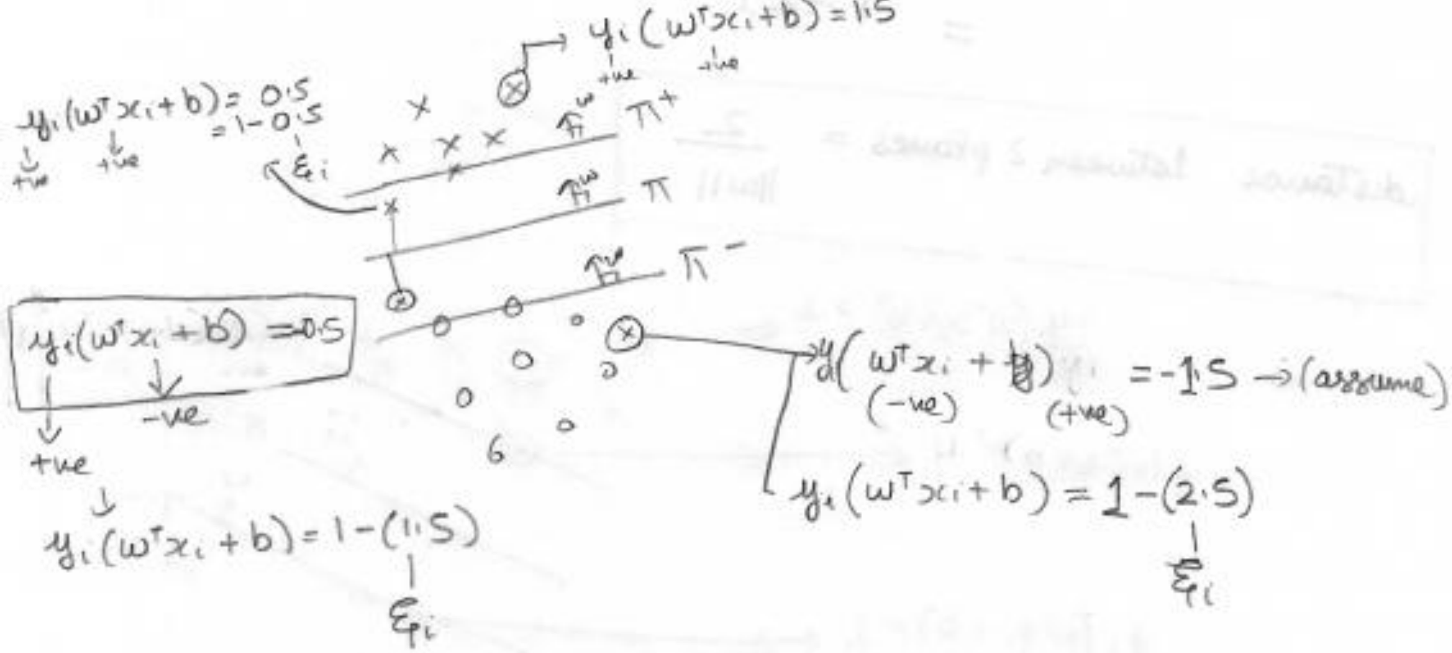
$\underset{-ve}{(w^T x_i + b)}\underset{+ve}{(y_i)} < -1$

No errors allowed

This is known as hard margin SVM

1) no point between the margins.
2) no -ve point on +ve side of hyperplane and vise versa as these are not ~~two~~ seperable by hard margin SVM.





ξ_i $\forall$ points

s.t. +ve point lies above π^+ $\xi_i=0$

-ve point lies below π^- $\xi_i=0$

$\Rightarrow$ $\xi_i\uparrow$, pt. is farther away from correct π in incorrect direction.

$\xi_i=0$ if $y_i(w^Tx_i+b)\geq 1$

$\xi_i>0$ else it is equal to the some units of dist away from correct hyperplane in incorrect direction

Objective:

$$(w^*,b^*)=\underset{w,b}{\arg\max}\ \frac{2}{\|w\|}=\underset{w,b}{\arg\min}\ \frac{\|w\|}{2}$$

ξ_i's

$$(w^*, b^*) = \underset{w,b}{\operatorname{argmin}} \underbrace{\frac{\|w\|}{2}}_{\substack{\Downarrow \\ \text{margin} \\ \text{(regularization)}}} + \underbrace{C}_{\text{hyperparameter}} \cdot \underbrace{\frac{1}{n} \sum_{i=1}^{n} \xi_i}_{\text{average distance of misclassified pts from } \pi\text{'s / loss}}$$

$$\text{s.t } y_i (w^T x_i + b) \geq 1 - \xi_i \quad \forall i$$
$$\xi_i \geq 0$$

Correctly classified pt $\xi_i = 0$

minimize errors / misclassifications

$$\Updownarrow$$

$$\min \sum \xi_i$$

$$\cancel{\text{s.t } y_i (w^T x_i + b) \geq 1 - \xi_i \ \forall i \ \xi_i \geq 0}$$

$C \uparrow$ giving more importance to make mistakes (reduces) on training data which leads to overfitting (variance)

$C \downarrow$ high bias or underfit model.

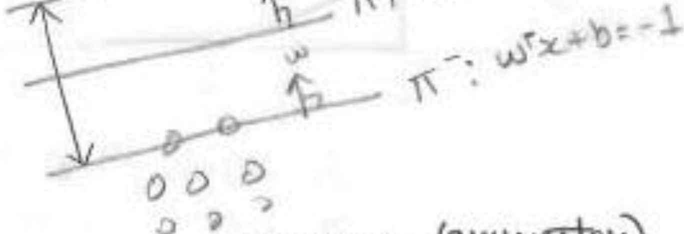
$$\boxed{(w^*, b^*) = \underset{w,b}{\operatorname{argmin}} \frac{\|w\|}{2} + C \cdot \frac{1}{n} \sum_{i=1}^{n} \xi_i}$$

Soft margin SVM

Q SVM: Why we take values +1 and −1 for SVM vector planes for hard margin SVM?

margin $= \frac{2}{\|w\|}$

$$w^*, b^* = \arg \max_{w,b} \frac{2}{\|w\|}$$



$\|w\| \neq 1$ (any vector) need not to be unit vector.

① $\Pi^+ :- \ w^T x + b = K$ $\qquad K > 0$

$\Pi^- :- \ w^T x + b = -K$

We are taking +K and −K as we want 2 hyperplanes Π^+ and Π^- to lie equally far away from Π.

$$\text{margin} = \frac{2K}{\|w\|}$$

$$\underset{w,b}{\text{argmax}} \ \frac{2}{\|w\|} = \underset{w,b}{\text{argmax}} \frac{2K}{\|w\|} = \frac{8}{\|w\|}$$

② $\Pi^+ : \ w^T x + b = K$ $\qquad w \perp \Pi$

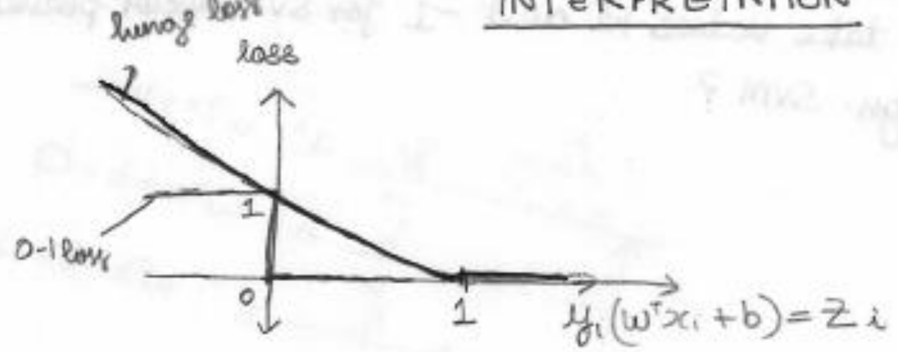
$$\left(\frac{w}{K}\right)^T x + \frac{b}{k} = 1$$

$$(w')^T x + b' = 1$$

Reason to take +1 and −1

Is to simple the math.

Loss Function (Hinge Loss) Based Interpretation



$\begin{cases} Z_i > 0 \ ; \ x_i \text{ is correctly classified} \\ Z_i < 0 \ ; \ x_i \text{ is incorrectly classified} \end{cases}$

hinge loss is not differenciable as it is not continous

approximated 0-1 loss by hinge loss

hinge loss :- $\begin{cases} z_i \geq 1; & \text{hinge loss} = 0 \\ z_i < 1; & \text{hinge loss} = 1 \end{cases}$

$\rightarrow \max(0, 1 - z_i)$

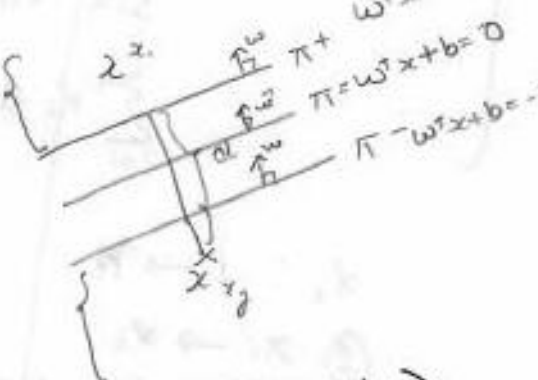
Case I : $z_i \geq 1$; $1 - z_i$ is -ve value $\Rightarrow \max(0, 1 - z_i) = 0$

Case II : $z_i < 1$, $1 - z_i$ is +ve value $\Rightarrow \max(0, 1 - z_i) = 1 - z_i$

☆ Geometric Formulation & loss minimization

$\xi_i = 0 \leftarrow x_i$

$d = 1 - y_i(w^T x_i + b)$



Soft SVM

$$\min_{w, b} \frac{\|w\|}{2} + \underbrace{C}_{\text{loss (hyperparamater)}} \sum_{i=1}^{n} \xi_i$$

$$\text{s.t } (1 - y_i(w^T x_i + b)) \geq \xi_i \ \forall i$$

$$\xi_i \geq 0$$

loss min

$$\min_{w, b} \sum_{i=1}^{n} \max(0, 1 - y_i(w^T x_i + b)) + \underbrace{\lambda \|w\|^2}_{\text{reg.}}$$

$\|w\| \geq 0 \Rightarrow \min \|w\|$ is same as $\min \|w\|^2$

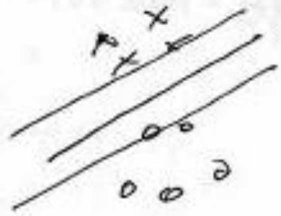
Note → If we multiply hyperparamater with loss function it will cause overfitting $C \uparrow$ = overfit

$\lambda \uparrow \Rightarrow$ underfit

3) Hard And Soft margin SVMs

a) <u>Hard margin SVM</u>

1.) Key idea of SVM maximise the margin



For the supports vectors, we know:

$y_i * (wx + b) = 1$

for non support vectors in the above image

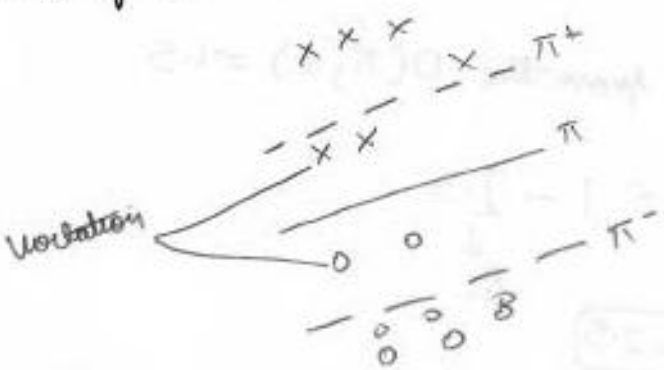
$y_i * (wx + b) > 1$

So optimization problem becomes:

$$\max (w) \left\{ \frac{2}{\|w\|} \right\} \text{ such that } y^* (wx_i + b) \geq 1$$

As we know the hard margin SVMs are optimal svm for linearly seprable data where +ve pts are above π^+ and -ve pt are below π^-. There is no points in the margin area or we can say no points voilating the margin.

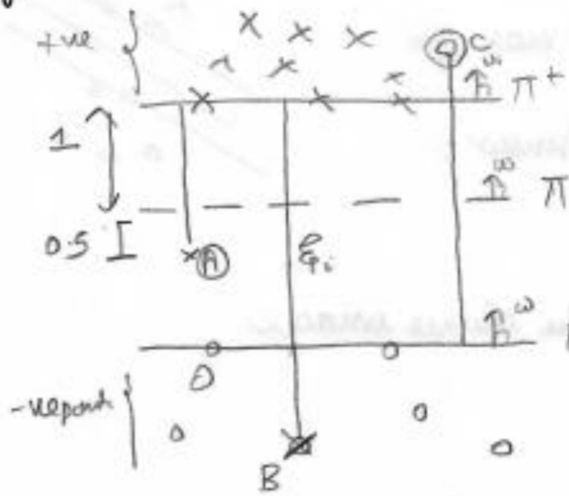
◇ what if pts lie between the margin?



Q The solution for this SVM is soft margin SVM?
Soft margin SVM.

ii) Soft margin SVM

When the points are almost linearly separable we consider soft-max SVM as the best idea to implement.



$$\max_{(w)} \left\{ \frac{2}{\|w\|} \right\} \text{ such that}$$

$$y_i (w x_i + b) \geq 1$$

For point A: The distance from the π, $D\{\pi, A\} = -0.5$

We can write this distance as $D\{\pi, A\} = 1 - (1.5)$

lets call this 1.5 as zeta {i}

Symbolized as $\xi_i = 1.5$

For point B: The distance from the, $D(\pi, B) = -1.5$

$$\xi_i = 2.5$$

$$D(\pi, B) = 1 - \underbrace{2.5}_{\xi_i}$$

For point C: The distance from the $D(\pi, C) = -1.5$

$$D(\pi, C) = 1 - \underbrace{2.5}_{\xi_i}$$

$$\boxed{\xi_i = 2.5}$$

Note for the SVM support vectors and the correctly classified pts we have $z_i = 0$ and for misclassified pts $z_i > 0$.

So the total error in our case is : (Soft margin)

$$C \sum_{i=1}^{m} \xi_i$$

As we know $\max(f(x)) = \min(-f(x))$

$$\max(f(x)) = \frac{1}{\min f(x)}$$

So we can write $\max(w) \left\{ \frac{2}{\|w\|} \right\}$ as $\min(w) \frac{1}{2} \|w\|$

Minimize $\frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \xi_i$

subject to $y_i (w^T x_i + b) \geq 1 - \xi_i \quad \forall \xi_i \geq 0$

C: hyperparameter : It is the hyperparameter which tunes how much error rate we have to optimize. As C value ↑ the importance of loss term increases.

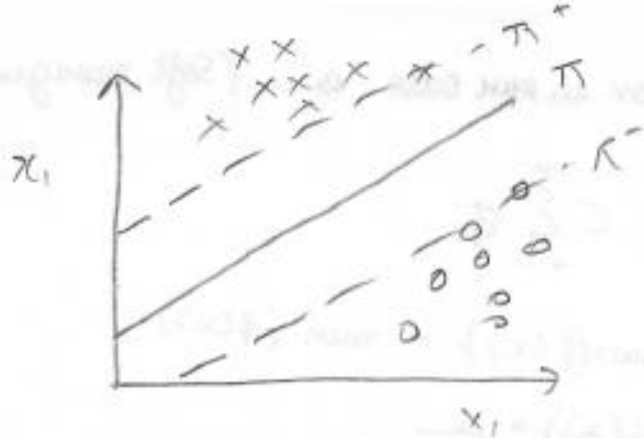
C ↑ overfitting ↑
C ↓ underfitting.

☆ Primal and Dual Forms of SVMs

Hard and Soft Margin is called constrained optimization problem

To get Dual form of the optimization problem we use Lagrange's multipliers

1.) Hard Margin SVM Primal form and Dual Form:



optimization problem for hard SVM

$$\min \frac{1}{2}\|w\|^2$$

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

This is primal form of SVM

$\Downarrow$ convert

Dual form we use (Lagrange multiplier α)

<u>rewriting the primal form with lagrange multiplier</u>

$$\underset{(w,b)}{\arg\min} \ \frac{1}{2}\|w\|^2 - \alpha_n \left\{ \text{Constraint eq}^n \right\}$$

where cons eq^n

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

$$\underset{(w,b)}{\arg\min} \left\{ \frac{1}{2} w^T w - \sum_{i=1}^{n} \alpha_n \left(y_i (w^T x_i + b) - 1 \right) \right\}$$

we min wrt (w,b) & max^n wrt $\alpha_n \geq 0$

$$L(w,b,\alpha) = \left\{ \frac{1}{2} w^T w - \sum_{i=1}^{n} \alpha_n \left(y_i (w^T x_i + b) - 1 \right) \right\}$$

Please note here we are minimizing the L wrt to (w,b) but we always maximize L w.r.t α

Gradient definations → derivative of the fn.

After finding the gradient w.r.t to w we get:

$$\nabla_w L = w - \sum_{n=1}^{N} \alpha_n y_n x_n = 0 \quad \text{(1)}$$

$$\frac{\partial L}{\partial b} = -\sum_{n=1}^{N} \alpha_n y_n = 0 \quad \text{(2)}$$

using lagrange

$$w = \sum_{n=1}^{N} \alpha_n y_n x_n \quad \& \quad \sum_{n=1}^{N} \alpha_n y_n = 0$$

In (1), (2) → partial derivative w.r.t w and partial derivative w.r.t b.

Putting w in $\alpha(w, b, \alpha)$

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{n=1}^{N} \alpha_n (y_n (w^T x_i + b) - 1)$$

$$\Rightarrow \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \alpha_n (y_n (w^T x_n))$$

$$\sum_{n=1}^{N} \alpha_n - \cancel{\frac{1}{2}} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m$$

$$\Rightarrow \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \alpha_n y_n x_n w^T$$

$$\sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m$$

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

subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$, $\alpha_i \geq 0 \; \forall i$ Dual form of hard margin

KKT condition   We have for $n = 1, \dots, N$

$$\alpha_n \left(y_n \underbrace{(w^T x_n + b_0)} - 1 \right) = 0$$

Interior points

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

$$\Downarrow$$

$$\alpha = 0$$

SVs

$$w^T x_n + b = 1$$

$$\downarrow$$

$$\alpha_n > 0$$

so $$w^* = \sum_{n \in SV}^{N} \alpha_n y_n x_n$$

Solve for b:-

using $y_n (w^T x_n + b) = 1$ for SVs

Put $w^* = \sum_{n \in SV}^{N} \alpha_n y_n x_n$ and get 'b'

$$y_n \left(\sum_{n \in SV}^{N} \alpha_n y_n x_n \cdot x_n + b \right) = 1$$

$$\sum_{n \in SV}^{N} \alpha_n (y_n)^2 x_n^T x_n + b = 1$$

$$\boxed{b = \left(\frac{1}{y_n}\right) - \sum_{n \in SV}^{N} \alpha_n (y_n) * x_n^T x_n}$$

Now we have optimal w & b but we donot have α value

☆ How to find α values?

We compute α using the concept of quadratic programming which takes minimization problems.

$$\min_{\alpha} \frac{1}{2} \sum \sum \alpha_n \alpha_m y_n y_m x_n^T x_m - \sum \alpha_y$$

Solution of quadratic programming gives

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - 1^T \alpha \quad \text{subject to} \quad y^T \alpha = 0 ; \alpha \geq 0$$

α is a vector where each α_i corresponds to each x_i

☆ Soft - Max Margin SVM Primal and Dual form.

$$\text{Minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

$$\text{subject to } y_i (w^T x_i + b) \geq 1 - \xi_i \quad \forall i \ \xi_i \geq 0$$

2 lagrange multiplier α and β.

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n - \sum_{n=1}^{N} \alpha_n (y_n (w^T x_n + b) - 1 + \xi_n)$$

$$- \sum_{n=1}^{n} \beta_n \xi_n$$

minimize w.r.t w, b & ξ maximum w.r.t $\alpha_n \geq 0$ & $\beta_n \geq 0$

gradient w.r.t 'w' and partial derivative wrt 'b' and ξ_i

$$\nabla_w \alpha = w - \sum_{n=1}^{n} \alpha_n y_n x_n = 0 \quad \text{—(I)}$$

$$\frac{\partial L}{\partial b} = - \sum_{n=1}^{} \alpha_n y_n = 0 \quad \text{—(II)}$$

$$\frac{\partial L}{\partial \xi_n} = (C - \alpha_n - \beta_n) = 0 \quad \text{—(III)}$$

Plug in equation into $L\{w, b, \alpha, \beta\}$

We got same dual form as we computed for hard margin SVM.

$$\max_{\alpha} L(\alpha) = \max_{\alpha} \left\{ \cdot \alpha_n - \frac{1}{2} \sum_{n=1} \sum_{n=1} y_n y_m \alpha_n \alpha_m x_n^T x_m \right.$$

$$\text{s.t } 0 \leq \alpha \leq C \ , \sum \alpha_n y_n = 0$$

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

$$\text{Subject to } \sum_{i=1}^{n} \alpha_i y_i = 0 \ , \ C \geq \alpha_i \geq 0 \ \forall_i$$

Hard margin SVM to separate the linearly separable dataset and soft margin SVM to separate the almost linearly separable dataset.

Kernels TRICK

Kernel. → It takes data points in X space in d dimensional and transform the points in Z space in d' dimension.

$$\max_{\alpha_i} \sum \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$ → similarity function → $K(x_i, x_j)$

$$\text{st } \sum_{i=1}^{n} \alpha_i y_i = 0 \ ; \ \alpha_i \geq 0$$

$$f(x_a) = \sum_{i=1}^{n} \alpha_i y_i \underbrace{K(x_i, x_q)} + b$$

Kernalization ÷ SVM handle non-linear seperate datasets

1.) <u>Polynomial Kernels</u>

$$K(x_1, x_2) = (x_1^T x_2 + c)^d$$

eg $K(x_1, x_2) = (1 + x_1^T x_2)^2$
$\Downarrow$
quadratic Kernel

$$x_1 = \langle x_{11}, x_{12} \rangle$$
$$x_2 = \langle x_{21}, x_{22} \rangle$$

$$= 1 + x_{11}^2 x_{21}^2 + x_{12}^2 x_{22}^2 + 2x_{11} x_{21} + 2x_{12} x_{22} + 2 x_{11} x_{21} x_{12} x_{22}$$

$$= [1, x_{11}^2, x_{12}^2, \sqrt{2}x_{11}, \sqrt{2}x_{12}, \sqrt{2}x_{11}x_{12}] : x_1'$$
$$[1, x_{2}^2, x_{22}^1, \sqrt{2}x_{21}, \sqrt{2}x_{22}, \sqrt{2}x_{21}x_{22}] : x_2'$$

$$= (x_1')^T (x_2')$$

Kernalization doing internally is feature transformation

<u>Radial Basis Functions (RBF)</u>

SVM: most popular/general purpose : RBF

$$(x_1, x_2) \quad K_{RBF}(x_1, x_2) = \exp\left(\frac{-\|x_1 - x_2\|^2}{2\sigma^2}\right)$$

hyperplane parameter

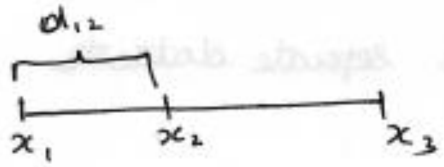
$$d_{12} = \|x_1 - x_2\|_2$$

$$K(x_1, x_2) = \exp\left(\frac{-d_{12}^2}{2\sigma^2}\right)$$

1.) $d_{12}\uparrow ; K(x_1, x_2) \downarrow$

$d\uparrow, d^2\uparrow, e^{d^2}\uparrow$

$\frac{1}{e^{d^2}}\downarrow$



$K(x_1, x_2) > K(x_1, x_3)$

(2) σ

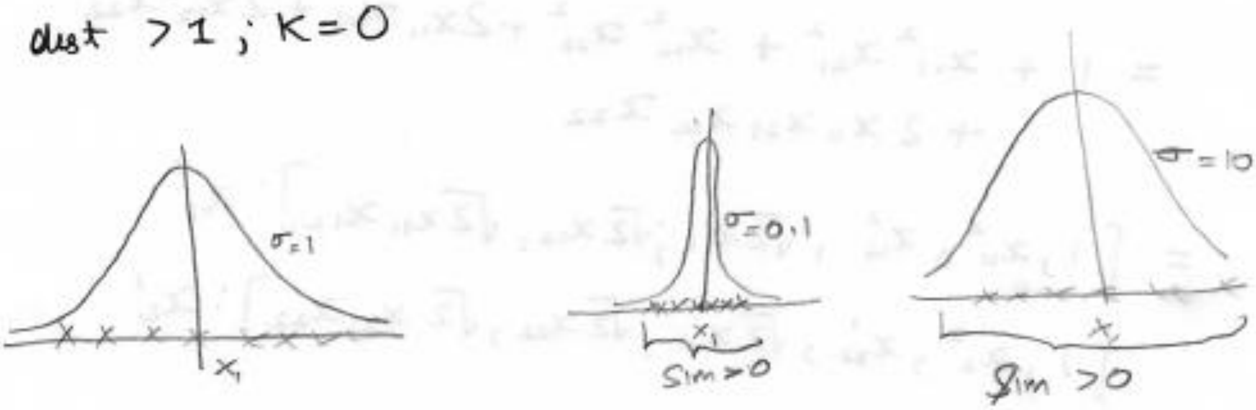
RBF ~ guassian kernel (μ, σ^2)

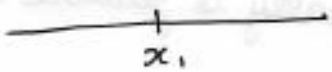
$\sigma = 1$ to $\sigma = 0.1$

$\sigma = 0.1$

$\sigma^2 = 0.01$

dist > 1 ; $K = 0$





RBF approximation is similar to KNN.

If you don't know which kernel to use you use RBF kernels

4 Domain specific Kernels ⇒ These are specilized kernels for specified tasks

for eg Graph kernel → specific kernel → lot of specilized graph kernel

Ex 1.8

Train & Run Time Complexity of SVM.

Train → SGD
↳ Specialized algo( ) → Sequential minimal optimization (SMO)

Training Time ~ $O(n^2)$ for kernel SVM

more opt algo ~ (2007) ÷ $O(nd^2)$ if $d < n$.

if you have large data n is large → $O(n^2)$ ↑↑
↑ Typically donot use SVM when n is large.
↓ internet applications

$O(Kd)$

---

nu-SVM

alternative formulation of SVM

$\boxed{0 \le nu \le 1}$

nu
hyperparameter

nu ≥ fraction of errors

nu ≤ fraction of SV's

lower bound

nu = 0.01 ⇒ %age of errors = 1%
# SVs ≥ 1% of n points

SVM D_train
10% errors
$\boxed{nu = 0.1}$
1% error
nu = 0.01

Run Time complexity

SVM Regressions

SVM Classification : SVC $\quad y_i \in \{+1, -1\}$

Mathematical formulation:

In form of SVR

$$\min_{w, b} \underbrace{\frac{1}{2}\|w\|^2}_{\text{regularization}}$$

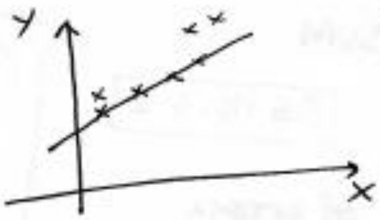
$$\text{s.t} \quad y_i - (w^T x_i + b) \le \underbrace{\varepsilon}_{\text{hyperparameter}}.$$

$$(w^T x_i + b) - y_i \le \varepsilon.$$

$$\varepsilon \ge 0$$

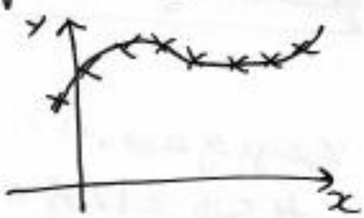
$$\boxed{f(x_i) = w^T x_i + b} = \hat{y}_i$$

error $\boxed{\begin{array}{l} = y_i - \hat{y}_i \le \varepsilon \\ \hat{y}_i - y_i \le \varepsilon. \end{array}}$



If not kernelized

If kernelized



Kernel SVM

$\mathcal{E}\downarrow \Rightarrow$ errors are low on training data

$\Rightarrow$ overfitting $\uparrow$

$\mathcal{E}\uparrow \Rightarrow$ errors on $D_{train}\uparrow \Rightarrow$ underfit $\uparrow$.

RBF SVR $\rightarrow$ $\boxed{\text{KNN-reg}}$