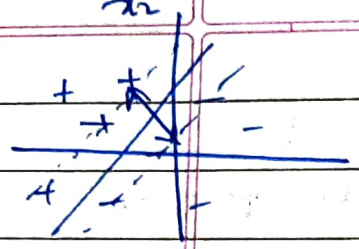
Week 7 SVM:

SVM for distribution free learning:



We try to minimize Empirical Risk: $\Leftrightarrow$

Find the boundary that margin maximization.

$t_i \in \{-1, +1\}$ for every classified point

$t_i (w^T x_i + b) > 0$ (Mirror fold) Subject to $\|w\| = 1$

Subject to: $\|w\| = 1 \Rightarrow$

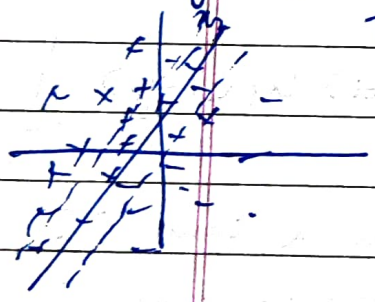
1. $\underset{w,b}{\text{Arg max}} \left[\frac{1}{\|w\|} \min_i \left[t_i (w^T x_i + b) \right] \right]$ subject to $\|w\| = 1$ (when no normalization)

2. $\underset{w,b}{\text{arg min}} \|w\|^2$ st $t_i (w^T x_i + b) \geq 1$

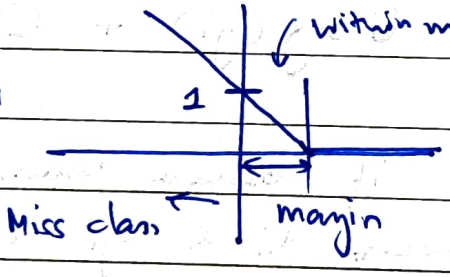
Stretch the points from margin by ~~increasing~~ shrinking w.

Constrained Convex Problem : Quadratic Programming

Soft Margin for SVM for non seperable data:



Tradeoff parameter = λ Slack variable ξ_i

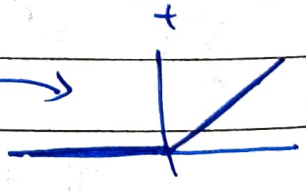


$$\min_{w,b,\xi} \left[\lambda \|w\|^2 + \frac{1}{N} \sum_i^N \xi_i \right]$$

Such that $\forall i \quad t_i (w^T x_i + b) \geq 1 - \xi_i \quad (\xi_i \geq 0)$

$$\min_{w,b} \left[\lambda \|w\|^2 + \frac{1}{N} \sum_{i=1}^{N} \left[1 - t_i (w^T x_i + b) \right] \right] \quad \text{(ReLU)}$$

ReLU $= \max(0, a)$



SVM - Also keep loss function

Such that margin is one & shrink w st margin maximized

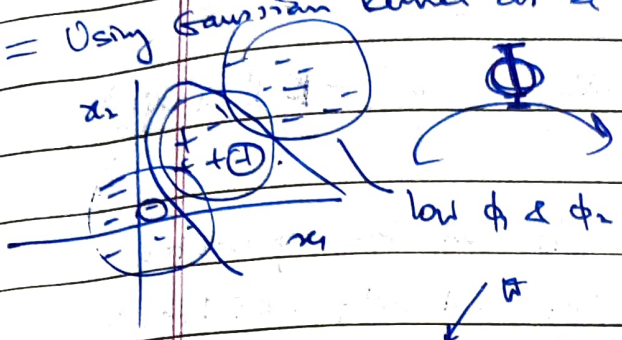
(Robustness $\Leftrightarrow$ Maximizing the margin.) Different loss fn in different properties

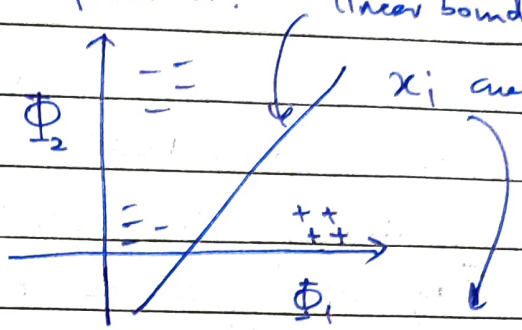
Perceptron loss fn: NOT convex relaxation of step fn. (Not always step function)

6.1.) Kernelized SVMs :-

- Derive dual form of SVM
- Show raw data replaced with kernel
- Derive SV version objective

= Using Gaussian kernel as a Basis function.





lose ϕ_1 & ϕ_2

$$\phi(x) = \exp\left(\frac{\|x - x_1\|^2}{2\sigma^2}\right) \quad \phi_2^{(x)} = \exp\left(-\frac{\|x - x_2\|}{2\sigma^2}\right)$$

High ϕ_1 for x_1 (points)

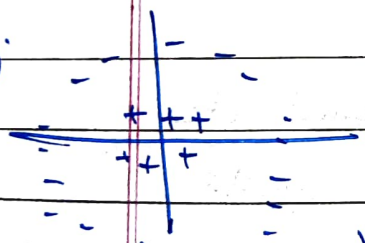
2 data points :) x where we want to compute feature

x_i is the referral data points

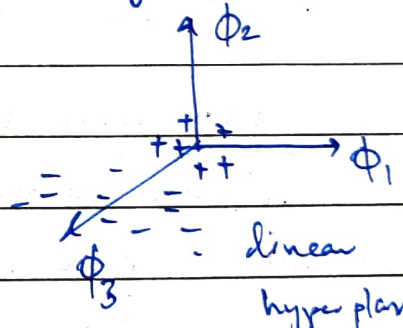
Why use a Kernel ?

- Using right kernel $\Phi(\cdot)$ can increase the dimension from x to that of $\phi(x)$
- More importantly, transformed data can be linearly separable.

Eg.



$$\phi(x) = \begin{bmatrix} x_1 \\ x_2 \\ \sqrt{x_1^2 + x_2^2} \end{bmatrix}$$



We may even need to transform into ∞ dimensions to have a linearly seperable boundary.

linear hyperplane

$K(x_i, x_j) = K(x_j, x_i)$ [Symmetric]

High value when $x_i = x_j$ and low when x_i & x_j are not alike at all.

Also, we want K to form a positive semi definite. (Non negative eigen-values.) Reproducing Kernel or Mercer Kernel in Hilbert space

So, we have: (Mercel Kernel)

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

PAGE No.
DATE / /

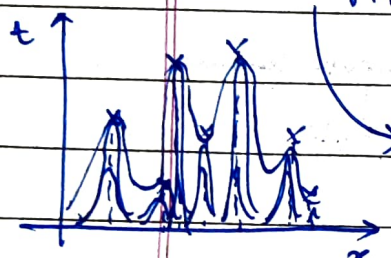
Example: $K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$

or equivalently $\exp\left(-k \|x_i - x_j\|^2\right)$

This gives a Gram Matrix.

* Change equation of SVM such that it can be represented as $x_i^T x_j$ as opposed to x [We want only DOT PRODUCT term]

If we have this, we may replace x with any ϕ and replace the dot product with a kernel. Lot of flexibility to design non linear SVM.



Fit a kernel (localised fn)

Estimate a fn that smoothly interpolates these pts

Same everywhere but shifted to different training pts

Scaled to different values.

Adding the linearly scaled functions we get a smooth function

$$f(x) = \sum_{i=1}^{N} a_i K(x, x_i)$$ [K is the gaussian kernel]

If regularly spaced we may have a triangle kernel etc

Different kernels give different smoothness effects.

We use a sparse set of points $\Rightarrow$ Take out all non SV points.

Change form, introduce dot products and replace kernel with dot product

Change primal form into Dual form. So that it has $x_i^T x_j$

Then we replace dot product with a ϕ to get output of Kernel to be a scalar

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$ Has +ve semi definit gram matrix.

6.2.1 original SVM into dual form :-

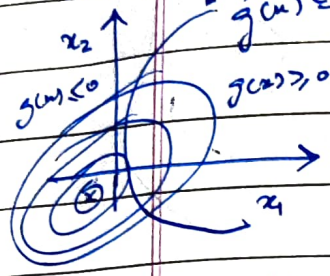
$$L(w,b) = \frac{1}{2}\|w\|^2 - \sum_i a_i [t_i(w^T x_i + b) - 1] \quad a_i \geq 0$$

↑ shrink weights ↑ lagrange multipliers ↑ Margin.

[Lagrange form of constraint opt over w] Subject to constraint $a_i \geq 0 \ \forall i$

$\min L(x, \lambda) = f(x) + \lambda g(x)$; $\lambda \geq 0$ ①

This function encodes minimizing $f(x)$ subject to $g(x) \geq 0$ ②



We want $g(x) \geq 0$ (Right side of curve)

Solution will be at yellow curve tangential to red curve.

aligned with 0 or 180°

Minimize and set $\nabla = 0$ $\quad \nabla f + \lambda \nabla g = 0$

Thus lagrange multiplier gives constrained optimized solution.

We want to replace x to convert into dual form replace with $x^T x$

$$\min L(w,b,a) = \frac{1}{2}\|w\|^2 - \sum_i a_i [t_i(w^T x_i + b) - 1] \quad a_i \geq 0$$

Set derivatives $\partial L/\partial w = 0 \implies w \text{ value} = \sum_i a_i t_i x_i$

$\partial L/\partial b = 0 \implies 0 = \sum a_i t_i$

Plug these in to eliminate w & b

$$L(a) = \sum_i a_i - \frac{1}{2}\sum_i \sum_j a_i a_j t_i t_j x_i^T x_j$$ ← $K(x_i, x_j)$

This on minimization gives values of a_i

Then y

$$y = \sum_i a_i t_i x^T x_i + b$$

Put values of a_i that minimized

Test ↑ training

($a_i = 0$ then those aren't support vectors)

$K(x, x_i)$

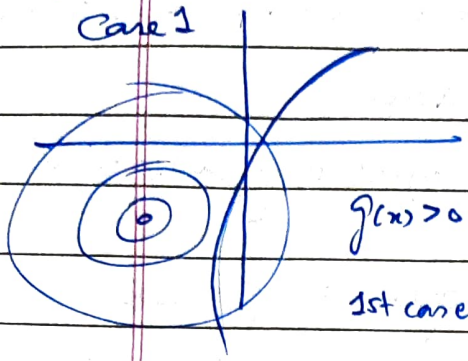
6.2.2 Solving the dual lagrangian L(a)

PAGE No. / DATE / /

$\min L(x) = f(x) + \lambda g(x)$

$\rightarrow g(x) \geq 0 \quad \lambda \geq 0 \quad \& \quad \lambda g(x) = 0$ [Either $g(x) = 0$ or $\lambda = 0$]

Case 1



$g(x) > 0$

1st case

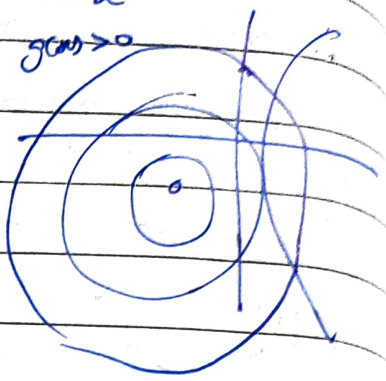
Ideal solution will be tangential to $g(x) = 0$

∇f & ∇g are aligned.

In this case $g(x) = 0$ & used to cancel out.

Case 2

$g(x) > 0$



$g(x)$ does not matter

$\Rightarrow \lambda = 0$

Case 1	Case 2
$g(x) = 0$	$\lambda = 0$
$\lambda g(x) = 0$	$\lambda g(x) = 0$

Then, $L(a) = \sum_i a_i - \frac{1}{2} \sum_i \sum_j a_i a_j t_i t_j x_i^T x_j$

Using the observations from earlier, we have:

$a_i \geq 0 \quad \& \quad t_i y(x_i) - 1 \geq 0$

$a_i [t_i y(x_i) - 1] = 0$ are the conditions

↳ Using this, what we get for b:

$$t_i \left[\sum_j a_j t_j x_i^T x_j + b \right] = 1$$

$$\therefore b = \frac{1}{N_S} \sum_{i \in S} \left[t_i - \sum a_j t_j x_i^T x_j \right]$$

KKT conditions (Karush Kuhn Tucker condition)

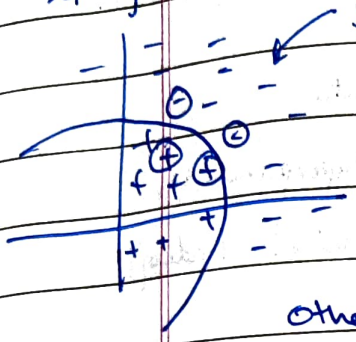
- Points where $a_n = 0 \notin S$ (Not SV)
  - $t_n (y(x_n) - 1)$ need not be 0
  - Do not define decision boundary - Can be moved around w/o changing the decision boundary.

- If $a_n > 0$ then $t_n(y(x_n) - 1) = 0$ (They are SV)
  - They define the decision boundary
  - These points need to be retrained on for classification of test data

PAGE No. / DATE / /

Visualizing Gaussian Kernel:

$x_i^T x_j$ $\quad K(x_i, x_j) = \exp(-k\|(x_i - x_j)\|^2)$



Support Vectors

Gaussian Kernel from these points

Decision boundary linear in ϕ space

Other useful Kernels: $-(x_i^T x_j + c)^d$ - Polynomial

Cosine - $x_i^T x_j / \|x_i\| \|x_j\|$ and linear kernels etc.

6.3.) What if the classes are not seperable? (Allow some misclass^n)

- An error function that gives ∞ error on misclassif^n needs to be modified
- Introduce slack variable $\xi_n \geq 0$ for each point n st:
  - $\xi_n = 0$ if the point is on correct side
  - $|t_n - y(x_n)|$ for points outside the margin

For this formulation :-

$$L(w, b, a, \xi) = \frac{1}{2}\|w\|^2 - \sum_i a_i [t_i y(x_i) - 1 + \xi_i] + C \sum_i \xi_i - \sum_i \mu_i \xi_i$$

Now we will have 6 KKT cond^{n}s:

• $a_i \geq 0$	• $t_i y(x_i) - 1 + \xi_i \geq 0$	• $a_i(t_i y(x_i) - 1 + \xi_i = 0$
• $\mu_i \geq 0$	• $\xi_i \geq 0$	• $\mu_i \xi_i = 0$

Now we eliminate w and have everything term of x_i & x_j dots

$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_i a_i t_i x_i$ $\quad \frac{\partial L}{\partial b} = 0 \Rightarrow \sum a_i t_i = 0$

$dL/d\xi = 0 \Rightarrow a_i = C - \mu_i$ (Eliminate ξ, w & b)

Consequently get dual form of soft SVM

$$\tilde{\mathcal{L}}(a) = \sum_i a_i - \frac{1}{2}\sum_i \sum_j a_i a_j b_i t_j \underbrace{x_i^T x_j}$$

PAGE No. / DATE / /

Condition: $0 \le a_i \le C$ & $\sum a_i t_i = 0$

↳ Boxed into 0 and C Therefore 3 types of a_is.

1) $a_i = 0$ Not support vectors, dont matter

↳ $\xi_i \le 1$ correct points but inside margin

2) $a_i = C$; $\xi_i \ge 1$ ⟹ Incorrectly classified points

Analyzing further & Support Vector Regression: → Hyperparameter

∴ We finally minimize $\tilde{\mathcal{L}}(a)$ when $C > 0$ controls tradeoff between slack variable penalty & the margin.

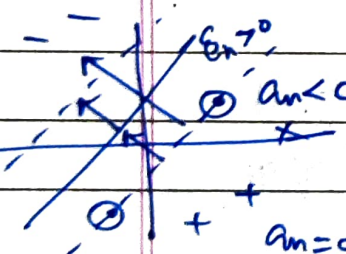
→ Note that this is sensitive to outliers as the penalty is linearly proportional to distance from the margin.

Large C → More emphasis on reducing ξ_i, fewer misclassified

~~Small~~ C → More SV ⟹ Smoother boundary. (OVO)

6.4) SVM for Regression:

Box Constraints: Four types of points:



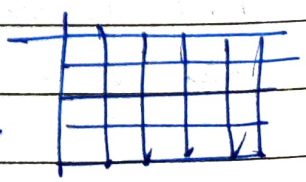
- $a_n = 0$ (Inside margin & correctly class)
- $a_n < C$, $\xi_n = 0$ (On the margin)
- $a_n = C$, $\xi_n \le 1$ (Across margin, correctly class)
- $a_n = C$, $\xi_n > 1$ (Across margin, misclassified)

Influence of point across margin is capped by C. [Capped]

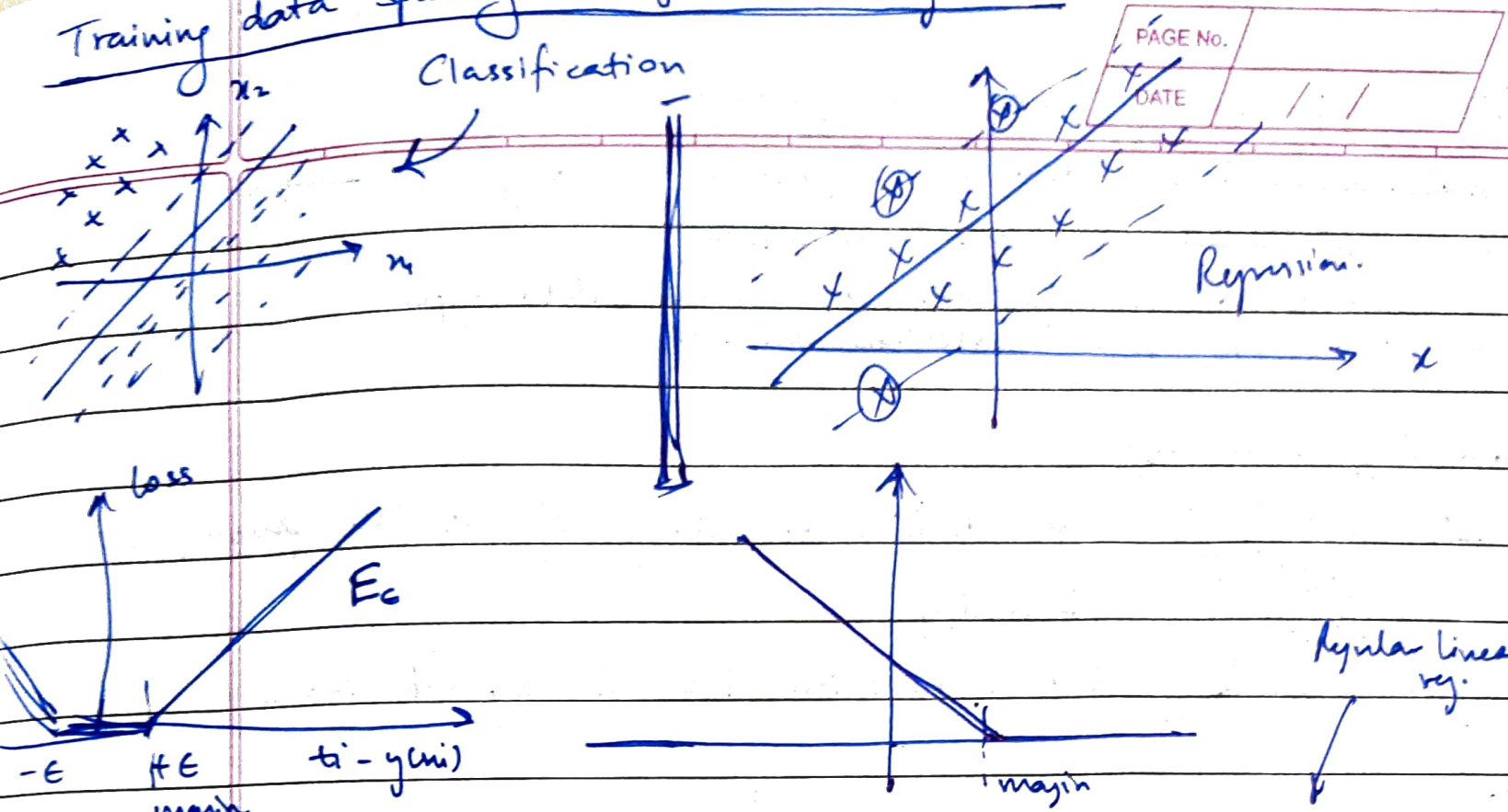
Hyperparameters:

C, & Kernel specific: K, d matrix

And then we choose the best combination of C, K



Training data sparsity in regression using SVM:



Mean Square Error $= \frac{1}{2} \sum_i (y_i - t_i)^2 + \frac{\lambda}{2} \|w\|^2$

SVR $= C \sum_i E_\xi (y_i - t_i) + \frac{1}{2} \|w\|^2$

$E_\xi = \max(0, |y_i - t_i| - \epsilon)$

Two slack variables $t_i \leq y_i + \epsilon + \xi_i \quad / \quad t_i \geq y_i - \epsilon - \hat{\xi}_i$

$C \sum_i (\xi_i + \hat{\xi}_i) + \frac{1}{2} \|w\|^2$