Ensemble Learning

09/09/2025

Combining weak learner to get a strong learner

Bagging - You can replace

Independent weak learner

b - no. of learners
n - no. of samples

n samples being picked n times
samples get replaced so we
always have n samples

$$P(\text{not being chosen}) = \left(1 - \frac{1}{n}\right)^n \doteq 0.37$$

Out of Bag - Those sample not being picked in training
↳ Acts as a validation set

If we don't add diversity, we will get same trees.

Subsampling - Subset of sample.

Bagging or Bootstrapping Algorithms

h_b = base learner (independent)
D_b = subset
For classification - majority voting
For regression - average

$$\text{Error} = (\text{Bias})^2 + \text{Variance} + \epsilon$$

Bias is already low in decision tree. It has more chances of overfitting ~~and~~ rather than underfitting.

So we reduce variance to reduce Error

$\boxed{B\uparrow \text{ Trees}\uparrow \text{ Variance}\downarrow \quad \text{Error}\downarrow}$

$$P(\underset{\text{error}}{\text{ensemble voting}}) = \sum_{k=\lceil \frac{B}{2} \rceil}^{B} \binom{B}{k} p^k (1-p)^{B-k}$$

linear regression - simple model → less variance - Bagging not helpful

Machine Learning (DSL-501)

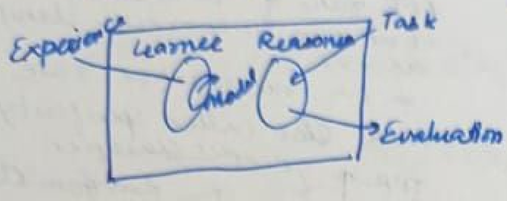
Theory (60%)
- Mid - 20%
- End - 30%
- Assignment - 10% (Best 2/3)

Lab (40%)
- Project
  - Phase I
    - 10% → 5% Presentation
    - Q1 Journal & CSE A/A* main conference
  - Phase - II
    - 5% -- Progress
  - Phase - III
    - 20% → Project Report (10 pages), Code, Presentation
    - Problem Statement
    - Related Work

Reference
- Pattern Recog & ML - Christopher Bishop
- An Intro to Statistical learning - Gareth James, Daniel Witten
- Hands On ML with Scikit learn by Aurelien Geron 3rd Ed
- Machine Learning - Tom Mitchell
- Deep Learning - Ian Goodfellow
- Essential Mathematics for Machine Learning

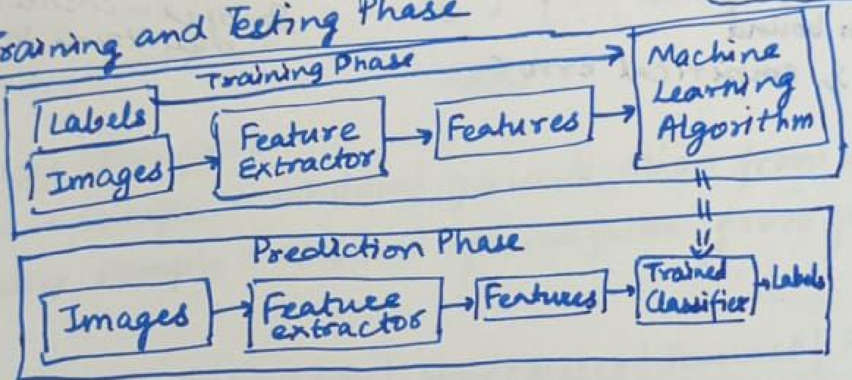
Learner System

Learner   Reasoner



Features
- ↳ Categorical (eg. "A", "AB ...")
- ↳ ~~Ordin~~ Ordinal (eg. "Large", "Medium", "Small")
- ↳ Integer valued
- ↳ Real Valued

Training and Testing Phase



Example (x, y) : Instance x with label y

Reinforcement Learning
The learning system, called an agent in this context, can observe the environment, select and perform actions and get rewards in return (or penalties)

Hyperparameter Tuning
- Manual Search
- Grid Search
- Random Search

$$y_i - \hat{y}_i$$

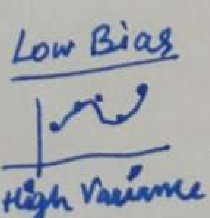
Expected Error → $EPE = \sum_{i=1}^{N}(f(x) - \hat{y}_i)$ → prediction function

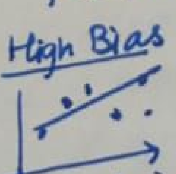
Prediction   True ~~prediction~~ function

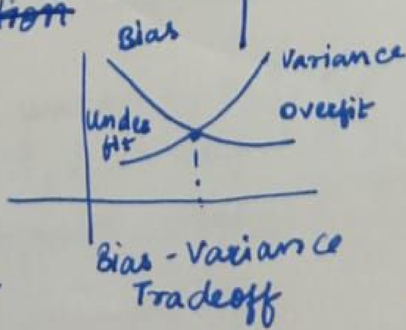
$$= E\left[\left(f(x_i) - \hat{y}(x_i)\right)^2\right]$$

$$= \underbrace{\left[f(x) - E[\hat{y}(x)]\right]^2}_{\text{Bias}} + \underbrace{E\left[\left(\hat{y}(x) - E[\hat{y}(x)]\right)^2\right]}_{\text{Variance}} + \underbrace{\sigma^2}_{\text{irreducible error}}$$

$$Var = E[\hat{f}^2(x)] - \left(E[\hat{f}(x)]\right)^2$$







Bias - Variance Tradeoff

$$precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ score = \frac{2}{\left(\frac{1}{Prec} + \frac{1}{Recall}\right)}$$

$x=2,\ f(x)=5$

$D_1 : \hat{y}_1(x) = 4.5$

$D_2 : \hat{y}_2(x) = 5.0$

$D_3 : \hat{y}_3(x) = 5.5$

Variance $= \frac{1}{N} \sum_{i=1}^{N} \frac{1}{M} \times \sum_{j=1}^{M} \left(\hat{y}^j(x_i) - E[\hat{y}(x_i)]\right)^2$

N → # of instances in Test

M → # of Models

Here $N=1$

Bias = 0

Hypothesis Space -

VC Dimension - how effectively select

Generalisation bound

True error $\geq$ empirical error

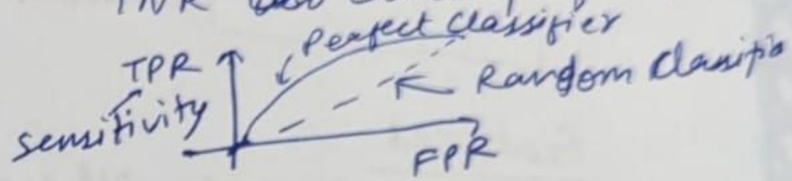
ROC Curve

True +ve rate (Recall) vs False +ve rate

FPR ratio of -ve instances that are incorrectly classified as +ve

= 1 − True negative rate

TNR also called specificity



Cross validation

K Fold → Each fold serves as validation set

Leave one out - Each observation

Stratified - Ensures each fold maintains same proportion of class

Concept Learning

Find-S | Candidate Elimination Algo

$D = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_n, y_n)\}$ $n \in \{1, 1000\}$

$m \in \{True, False\}$

$\Downarrow$

$(x_1, c(x_1)), (x_2, c(x_2)) \ldots$

$\Downarrow$

$h(x) = c(x) \ \forall \ x \in X$

Find-S — Not using -ve example

f_1, f_2, f_3

red, round, small +

yellow, long, medium −

green, round, medium +

$S_0 : \{\phi, \phi, \phi\}$

$S_1 : \{red, round, small\}$

$S_2 : \{?, round, ?\}$ ← If values are mismatch in +ve example than generalise(?)

~~$c(x)$~~

Candidate Elimination

Positive sample − minimal generalization from specific boundary

Negative sample − minimal specialization from general boundary

Sky	Air temp	Humidity	Wind	Water	Forecast	Playsport
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	"	"	"	Yes
Rainy	Cold	"	"	"	Change	No
Sunny	Warm	"	"	cool	"	Yes

$S_0 : \langle \phi, \phi, \phi, \phi, \phi, \phi \rangle$ $G_0 : \langle ?, ?, ?, ?, ?, ? \rangle$

$S_1 : \langle sunny, warm, normal, strong, warm, same \rangle$ $G_1 : \langle ?, ?, ?, ?, ?, ? \rangle$

$S_2 : \langle sunny, warm, ?, strong, warm, same \rangle$ $G_2 : \langle ?, ?, ?, ?, ?, ? \rangle$

$\boxed{G_3 : \langle sunny, ?, ?, \ldots, ? \rangle \mid \langle ?, warm, ? \ldots ? \rangle \mid \cancel{\langle ?, ?, Normal, ?, ?, ? \rangle} \mid \cancel{\langle ?, ?, ?, ?, cool, ? \rangle} \mid \cancel{\langle ?, ?, \ldots, Same \rangle}}$

eliminate inconsistent hypothesis

$S_3 = S_2$

$\boxed{S_4 : \langle sunny, warm, ?, strong, ?, ? \rangle}$

version space

- $\langle sunny, warm, ?, ?, ?, ? \rangle$
- $\langle sunny, ?, ?, strong, ?, ? \rangle$
- $\langle ?, warm, ?, strong, ?, ? \rangle$

→ $|V| = 3$

Decision Tree

How to split at Internal Nodes

- A pure group means that the majority of the inputs have the same label/output

Entropy and Information Gain

S - set of labelled inputs from C classes, P_c as fraction of class c inputs

Entropy $\rightarrow H(S) = -\sum_{c \in C} P_c \log P_c$

split S to S_1 & S_2

Reduction in entropy after split is called information gain

$$IG = H(S) - \frac{|S_1|}{|S|} H(S_1) - \frac{|S_2|}{|S|} H(S_2)$$

low entropy - High Information Gain - Good split

$$H(S) = 0.94$$

$S_{sunny} = [2+, 3-]$ $S_{rain} = [3+, 2-]$, $S_{overcast} = [4+, 0-]$

$H(S_{sunny}) = -\left(\frac{2}{5} \log_2 \frac{2}{5} + \frac{3}{5} \log_2 \frac{3}{5}\right) = +0.971$

$H(S_{rainy}) + 0.971$

$$H(S_{overcast}) = 0$$

$$IG(S, outlook) = 0.94 - 2 \times \frac{5}{14} \times 0.971 = 0.246$$

$S_{high} = [3+, 4-]$ $S_{normal} = [6+, 1-]$

$H(S_{high}) = \frac{3}{7} \log \frac{3}{7} + \frac{4}{7} \log \frac{4}{7} = 0.985$

$H(S_{normal}) = \frac{6}{7} \log \frac{6}{7} = 0.19$

$IG(S, humidity) = 0.94 - \frac{7}{14}(0.985) - \frac{7}{14} \times 0.19$

k-Nearest Neighbour Learning

18/08/2025

Instance based learning → "Lazy / Memory based"

↳ Model is not explicitly trained on a training dataset

Nearest neighbour defined in terms of distance

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2}$$

Here i and j are instances, a_r is r^{th} feature

Voronoi. ~~Voronoi~~ Diagram

Draw perpendicular bisectors of each pair of points.

1) **Training Algorithm**

- For each training algorithm $\langle x, f(x) \rangle$, add the example to list of training example

2) **Classification Algorithm**

- Given a query instance x_q to be classified
- Let $x_1, x_2 \dots, x_k$ denote the k instances from training examples that are nearest to x_q
- Return $f(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} \delta(v, f(x_i))$

$$\delta(a,b) = \begin{cases} 1 & ; \text{if } a=b \\ 0 & . \text{ else} \end{cases}$$

- For Regression $(f : R^n \to R)$

$$f(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_i)}{k}$$

Here we are assigning the class to unseen/new data points based on True/False (0 & 1), that is not good.

- **Distance weighted NN**

for classification

$$f(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} w_i \, \delta(v, f(x_i)); \text{ where } w_i = \frac{1}{d(x_q, x_i)^2} \text{ and } \delta(a,b) = \begin{cases} 1 & \text{if } a=b \\ 0 & \text{else} \end{cases}$$

for regression

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i (f_i)}{\sum_{i=1}^{k} w_i}, \text{ where } w_i = \frac{1}{d(x_q, x_i)^2}$$

Linear Regression

21/08/2025

$y = \theta_1 x_1 + \theta_0$ ← Inductive Bias

$x_0 = 1$

$\hat{y} = h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i$

$x \in \mathbb{R}^d$

$\theta \in \mathbb{R}^{d+1}$

$J(\theta) = r = y - \hat{y}$ ← cost function

residual

$= \frac{1}{2} \sum_{i=0}^{n} (\hat{y}_i - y_i)^2 = \frac{1}{2} \sum_{i=0}^{n} (h_\theta(x^i) - y^i)^2$

$= \frac{1}{2} \sum_{i=0}^{n} (\theta_i x_i - y_i)^2$

minimise residual based on the value of θ $\boxed{\arg_\theta(\min r)}$

or

$\arg_\theta \min J(\theta)$

To get to minima - 3 approach

(1) Gradient Descent Approach

$\theta^{(0)} = \text{Init}$

for all $j \in \{1, \ldots, d+1\}$

$\theta_j^{(1)} = \theta_j^{(0)} - \alpha \frac{\partial J(\theta^0)}{\partial \theta_j}$

$\theta_j^{(2)} = \theta_j^{(1)} - \alpha \frac{\partial J(\theta^1)}{\partial \theta_j}$

$\theta^{t+1} = \theta^t - \alpha \nabla_\theta J(\theta^t)$

$= \theta^t - \alpha \nabla_\theta \left[\frac{1}{2} \sum_{i=1}^{n} (h_\theta(x^i) - y^{(i)})^2\right]$

$= \theta^t - \alpha \nabla_\theta \left[\frac{1}{2} \sum_{i=1}^{n} (\theta^T x^{(i)} - y^{(i)})^2\right]$

diff wrt θ

$\theta^{t+1} = \theta^t - \alpha \left[\sum_{i=1}^{n} (\theta^T x^{(i)} - y^{(i)}) x^{(i)}\right]$

How many iterations?

Repeat Until Convergence

→ $\| \theta^{(t)} - \theta^{(t-1)} \|$

→ $| J(\theta^t) - J(\theta^{t-1}) |$

→ $\frac{\partial J(\theta^0)}{\partial \theta_j}$ is too small

For each Sample updating θ as we calculate

for i in n:
  θ = Init
  for j in d:
  $\theta_j \leftarrow \nabla_{\theta_j} J(\theta_j)$

$\boxed{\text{Stochastic Grad. Descent}}$

?? In large no. of sample, takes less time to converge

Considering n samples, storing gradients and then updating.

  θ = Init
  for i in n:
    for j in d:
      $\nabla_{\theta_j} J(\theta_j)$
  $\theta \leftarrow \nabla_{\theta_j} J(\theta_j)$

$\boxed{\text{Batch Grad. Descent}}$

(2) Closed form Sol^n

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n} (\theta^T x^i - y^i)^2$$

$$\underbrace{\theta X}_{\mathbb{R}^{n\times(d+1)}} - \underbrace{Y}_{\mathbb{R}^{n\times(d+1)}} \rightarrow \mathbb{R}^n$$

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n} (\theta X - Y)^T(\theta X - Y)$$

$$J(\theta) = 0$$

$$= \frac{1}{2}\sum_{i=1}^{n}\Big[(X\theta)^T X\theta - \overbrace{\boxed{Y^T X\theta - (X\theta)^T Y}}^{\text{scalar}} + Y^T Y\Big]$$

$$= \frac{1}{2}\sum_{i=1}^{n}\Big[\theta^T (X^T X)\theta - 2\theta^T (X^T Y) + Y^T Y\Big] \quad \Big\{ \theta^T A \theta \Rightarrow [A + A^T]\theta$$

$$= \frac{1}{2}\sum_{i=1}^{n}\Big[2(X^T X)\theta - 2\cancel{\theta^T}(X^T Y)\Big]$$

$$(X^T X)\theta - \cancel{\theta^T}(X^T Y) = 0$$

$$\boxed{(X^T X)\theta = \cancel{\theta^T}(X^T Y)} \Leftarrow \text{Normal eq}^n$$

$$\boxed{\theta = (X^T Y)(X^T X)^{-1}}$$

Diff between Probability and Likelihood

Probability - when We know the parameters

Likelihood - When we don't know the parameters

$$y^{(i)} = \theta^T x^{(i)} + \overset{\text{noise}}{\epsilon^{(i)}} \qquad \epsilon^{(i)} \in \overset{\text{gaussian}}{N(0,\sigma^2)}$$

ϵ iid

$$\epsilon^{(i)} = y^{(i)} - \theta^T x^{(i)} \sim N(0,\sigma^2)$$

If $y^{(i)} - \theta^T x^{(i)} \sim N(0,\sigma^2)$

$y^{(i)} \sim N(0,\sigma^2)$ → mean would actually not be 0 but depend on $\theta^T x^{(i)}$

$$y^{(i)} \sim N(\theta^T x^{(i)}, \sigma^2)$$

$$P(y^{(i)}|x^{(i)};\theta) \Rightarrow \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}\right)$$

(3) **Likelihood** $L(\theta) \Rightarrow \prod_{i=1}^{n} P(y^{(i)}|x^{(i)};\theta)$

$$= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y^{(i)}-\theta^T x^{(i)})^2\right)$$

$$\log(L(\theta)) = l(\theta) = \frac{-n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y^{(i)}-\theta^T x^{(i)})^2$$

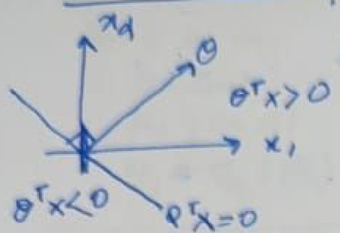
$$l(\theta) = k - \frac{1}{\sigma^2}\sum_{i=1}^{n}\frac{1}{2}(y^{(i)}-\theta^T x^{(i)})^2$$

$$\boxed{\arg\min_{\theta} J(\theta) = \arg\max_{\theta} l(\theta)}$$

Classification - using Perceptron (built on Linear regression)

$$y^{(i)} = g(\underbrace{\theta^T x^{(i)}}_{\text{linear regression}}) \qquad y^{(i)} \in \{0,1\}$$

Normal vector



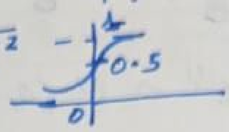
$D: \{(x_i, y_i)\}_{i=1}^{n},\ y_i \in \{0,1\}$

$y = \theta_0 + \theta_1 x_1$

$y = g(\theta^T x)$

Step function $\begin{cases} 1 & \text{if } g(\cdot) > 0 \\ 0 & \text{if } g(\cdot) < 0 \end{cases}$

$g(z) = \frac{1}{1+e^{-z}}$



$\begin{cases} 1 & \text{if } g(z) \geq 0.5 \\ 0 & \text{else} \end{cases}$

$h_\theta(x) = g(\theta^T x)$

$g(z) = \frac{1}{1+e^{-z}}$

$P(y^{(i)} = 1 \mid x^{(i)}; \theta) = h_\theta(x)$

$P(y^{(i)} = 0 \mid x^{(i)}; \theta) = 1 - h_\theta(x)$

$P(y \mid x; \theta) = (h_\theta(x))^y \times (1 - h_\theta(x))^{(1-y)}$

$\{(h_\theta(x))^1 \times (1 - h_\theta(x))^{(1-1)}\}$

$g(z') = g(z)(1 - g(z))$

$L(\theta) = \prod_{i=1}^{n} P(y^{(i)} \mid x^{(i)}; \theta)$

$l(\theta) = \sum_{i=1}^{n} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x)^{(i)})$

Grad Descent $\longrightarrow \nabla_\theta l(\theta)$

$\theta_{new} = \theta_{old} - \eta \nabla_\theta l(\theta)$

<u>For single example</u>

$y \log g(\theta^T x) + (1-y) \log(1 - g(\theta^T x))$

$\nabla_\theta l(\theta) = y \cdot \frac{1}{g(\theta^T x)} g'(\theta^T x) \cdot x + (1-y) \frac{1}{1 - g(\theta^T x)} (-1) g'(\theta^T x) \cdot x$

$= y \cdot \frac{1}{g(\theta^T x)} g(\theta^T x)(1 - g(\theta^T x)) x + (1-y) \frac{1}{1 - g(\theta^T x)} (-1) g(\theta^T x) \cdot (1 - g(\theta^T x)) x$

$= y(1 - g(\theta^T x)) x + (1-y)(-1) g(\theta^T x) x$

$= y(1 - g(\theta^T x)) + (1-y)(-1) g(\theta^T x))$

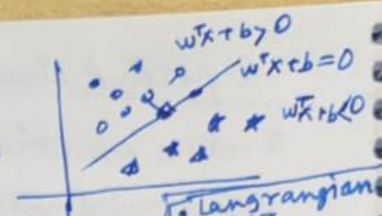
$= x(y(1 - g(\theta^T x)) + (1-y)(-1) g(\theta^T x))$

$= x(y - \cancel{y g(\theta^T x)} - g(\theta^T x) + \cancel{y g(\theta^T x)})$

$\nabla_\theta l(\theta) = x(y - g(\theta^T x))$

$\theta_{new} = \theta_{old} + \eta [x(y - g(\theta^T x))]$

SVM - Support Vector Machine



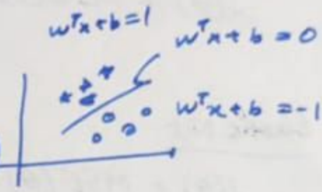
- Langrangian
- KKT
- Primal } Feasibility
- Dual

Along with classification also add confidence

γ = min Distance (point, hyperspace)

objective:- max γ constraint (w^Tx+b)

$$\underset{w^*,b^*}{\arg\max}\ \gamma$$



$$\{(x_i, y_i)\}_{i=1}^{n} \quad y \in \{-1, +1\}$$

Max distance with respect to Hyperplane (Decision Surface)

distance (x_i, w^Tx_i+b)

$$= \frac{|w^Tx_i+b|}{\|w\|}$$

$$dis(x_{i+}, w^Tx_i+b) = \frac{|w^Tx_{i+}+b|}{\|w\|}$$

$$y_i(w^Tx_i+b) \geq 1$$

for positive example $y_i = +1$

$$w^Tx_i+b \geq 1$$

for negative examples $y_i = -1$

$$(-1)(w^Tx_i+b) \geq 1$$

$$w^Tx_i+b < 1$$

$$dis(x_i, \text{hyperplane}) = \frac{y_i|w^Tx_i+b|}{\|w\|}$$

$$\hat{\gamma}_i = \min_i(dis(x_i, \text{hyperplane}))$$

(margin) $\rightarrow$ $\gamma = \max(\hat{\gamma}_i)$

Functional margin - $y_i(w^Tx_i+b)$

Geometric margin $\dfrac{y_i(w^Tx_i+b)}{\|w\|_2}$

$\gamma = \max\left(\frac{1}{\|w\|_2}\right) \Rightarrow \min\left(\|w\|_2\right)$, Primal

$$\boxed{\Rightarrow \min\left(\frac{1}{2}\|w\|_2^2\right) \text{ s.t. } y_i\left(w^T x_i + b\right) \geq 1}$$

$\frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{n} y_i\left(w^T x_i + b\right) \geq 1$ ← problem with this method

So we use Langrangian $\boxed{L = f(x) + \sum \lambda_i g_i(x) + \sum \mu_i h_i(x)}$

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i \left(1 - y_i\left(w_i^T x + b\right)\right), \quad \alpha \geq 0$$

$\frac{\partial L}{\partial w}$ & $\frac{\partial L}{\partial b}$

$$\min_{w,b}\left(\max_{\alpha \geq 0}\left(L(w,b,\alpha)\right)\right)$$

$$\max_{\alpha \geq 0}\left(\min_{w,b}\left(L(w,b,\alpha)\right)\right)$$

08/09/2025

$$\frac{\partial L}{\partial b} = 0 = \sum_{i=1}^{n} \alpha_i y_i$$

$$\frac{\partial L}{\partial w} = 0 = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\max_{\alpha \geq 0}\left(\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i \left(1 - y_i\left(w^T x_i + b\right)\right)\right)$$

$$\max_{\alpha \geq 0}\left(\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha - \sum_{i=1}^{n} \alpha_i y_i \left(w^T x_i + b\right)\right)$$

$$- \sum_{i=1}^{n} \alpha_i y_i \left(w^T x_i\right) - \sum_{i=1}^{n} \alpha_i y_i$$

$$\Downarrow$$

$$- \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$w(\alpha) = L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left(x_i^T x_j\right)$$

~~$w = \sum \alpha_i y_i x_i$~~ $\max_{\alpha} w(\alpha)$ $\qquad Q = y_i y_j \left(x_i^T x_j\right)$

KKT $\boxed{\max_{\alpha} w(\alpha) = 1^T \alpha - \frac{1}{2} \alpha^T Q \alpha}$

1. Primal Feasibility :- $y\left(w^T x_i + b\right) \geq 1$
2. Dual feasibility :- $\alpha_i \geq 0$
3. Stationarity :- $w = \sum_i \alpha_i y_i x_i$; $\sum_i \alpha_i y_i = 0$
4. Slackness :- $\alpha_i \left(y_i\left(w^T x_i + b\right) - 1\right) = 0$

$$f(x) = \text{sign}\left(\sum_{i}^{n} \alpha_i^* y_i \left(x_i^T x\right) + b\right)$$

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \quad s.t.\ y_i(w^T x_i + b) \geq 1-\xi_i$$

$$y_i(w^T x_i + b) \geq 1-\xi_i \qquad \xi_i \geq 0$$

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

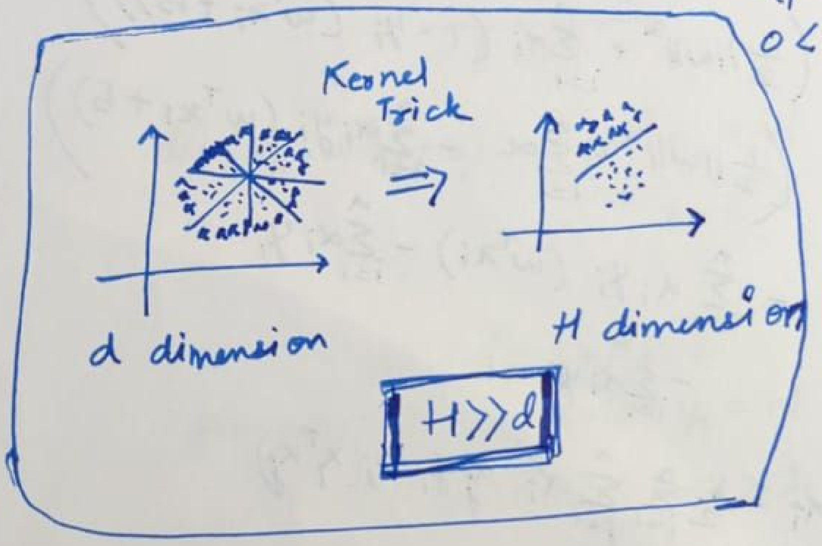
$$\min_{w,b,\xi_i,\mu,\alpha} \frac{1}{2}\|w\|^2 + \mu_j\left(\sum_i \xi_i\right) + \alpha_i\left(1-\xi_i - y_i(w^T x_i + b)\right)$$

$$\frac{\partial L}{\partial w}, \frac{\partial L}{\partial b}, \frac{\partial L}{\partial \xi}$$

$$\max_{\alpha} w(\alpha) = \sum \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i^T x_j)$$

$$s.t.\ 0 \leq \alpha_i \leq C \ ; \ \sum_i \alpha_i y_i = 0$$

$\alpha_i = 0$ non support vector
$\alpha_i = C$ inside margin / misclassified
$0 < \alpha_i < C$ on the margin



Pattern
MSQ
Fill in blank
Some straightforward
Some numerical
Senario based

Kernel function
Poly:- $K(x_i, x_j) = (x_i^T x_j + c)^q$
eg of more fxns: RBF, Gaussian, Sigmoid