Mid 2021

1) ML?
* subfield of AI that uses statistical methods to determine some outputs corresponding to some inputs like classification problems. make use of patterns in Data and use indirect instructions and learn from exper

2] classification → the output is some specific value from a set of values, classify cat or Dog

Regression → the output is a continuous quantity like house price or temperature level

3] Feature space → All possible values that a feature can take the feature can be mapped in N-Dim

Decision Region → constructed by finding a boundary bet. classes that can seperate them clearly
eg linear & Non-Linear boundaries

4] KNN → it calculates the distance between the test point and all the points in the space, and decide based on the majority of the K nearst points

Adv → not very sensetive to the outliers because it takes K-points, not affected by rouge patterns

Dis → sensitive to K, if K is too large it may take points from another regions!

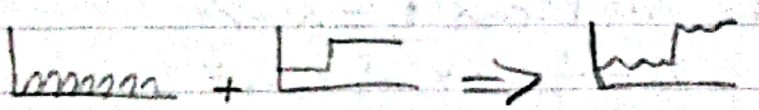
5] Bayes classifier → It's an optimum classifier as it takes the class with the highest probability! So, no other classification rule is better anyway.

* It's assumes that the Prob. density is known, which is not usually the case. Practically, only a training data is available and from it we can estimate densities
* the density estimation have some errors that is higher if the training set size is smaller.

6] Kernel density [Parzen window] → used to get the density of data by using bump func for each point, then it gets the estimation density through the avg of their sum. it's continues and give a weight for the points. Hence better than the histogram & naive estimators

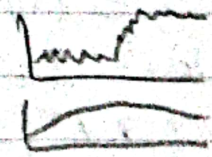
$\varphi_h(x) = \frac{1}{h} g\left(\frac{x}{h}\right)$

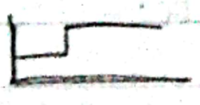
$\hat{p}(x) = \frac{1}{M} \sum_{1}^{M} \varphi_h(x - x(m))$



& $g(x)$ should be a PDF!

7] Kernel width (h) → too small → too bumpy
too large → Details lost



For 

$h)_{opt} = \frac{1}{N} \sum H_i \rightarrow H_i = \sigma_i \left[\frac{4}{(N+2)M}\right]^{\frac{1}{N+4}}$

8] Filter Type → select features regardless of the used classifier. Based on [Corr bet. Features, Distance bet. class means]

Adv → Fast exe, general

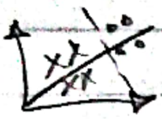
Dis → Tendency to select large subsets

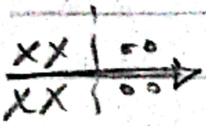
Wrapper type → select features by taking the classifier into consideration. e.g: SFS, SBS ($\downarrow$ seq, $\downarrow$ seq)

Adv → Accuracy ↑

Dis → slow exe, lack of generality ↓

9] PCA → A feature extraction method, in which we want to rotate and shift the data distribution to the direction of max variance. Hence, the covariance matrix is diagonal [No corr] and we can select (L) features with the highest impact [variance or λ]





1] Transform the problem to zero mean $\underline{Y} = \underline{X} - \hat{\underline{\mu}}$

2] Estimate Σ from the data, $\Sigma = \frac{1}{M} \sum \underline{Y(m)}\, \underline{Y(m)}^T$

3] Compute Eigen values and vectors of Σ,
we want Σ_Z to be diagonal [uncorrelated] for $Z = AY$

$$\Sigma_Z = E[ZZ^T] = E[AYY^TA^T] = AE[YY^T]A^T = A\Sigma_Y A^T$$

$\therefore Bu = \lambda u$, let $B = \Sigma$, $A = U^T \rightarrow \Sigma_Z = A\Sigma A^T = U^T \Sigma U = \Omega$

$\therefore \Sigma_Z$ is diagonal $\rightarrow u^T B u = \lambda$

4] Choose the eigenvectors corresponding to big Eigen Values, reorder values such that $\lambda_1 >> \lambda_2 \cdots >> \lambda_n$ and take (L) values

5] Rewrite the transformation as: $Z = \begin{bmatrix} U_1^T \\ U_2^T \\ U_L^T \end{bmatrix} Y$

Q:10 $C_1 \Rightarrow \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}1.5\\0.7\end{pmatrix}, \begin{pmatrix}1.5\\1\end{pmatrix}$

$C_2 \Rightarrow \begin{pmatrix}0\\-1\end{pmatrix}, \begin{pmatrix}0.1\\0.3\end{pmatrix}, \begin{pmatrix}1.2\\0\end{pmatrix}$

a] min dist

$\hookrightarrow M_1 = \left(\frac{4}{3}, \frac{9}{10}\right)$

$\hookrightarrow M_2 = \left(\frac{13}{30}, \frac{-7}{30}\right)$

$C = \begin{pmatrix}1.3\\0\end{pmatrix}$

$d_1^2 = 0.811 >$ } C_2 ✓

$d_2^2 = 0.8055$

KNN

(1,1) → 1.09 ④

(1.5,0.7) → 0.53 ②

(1.5,1) → 1.04 ③

(0,-1) → 2.69 ⑥

(0.1,0.3) → 1.53 ⑤

(1.2,0) → 0.01 ①

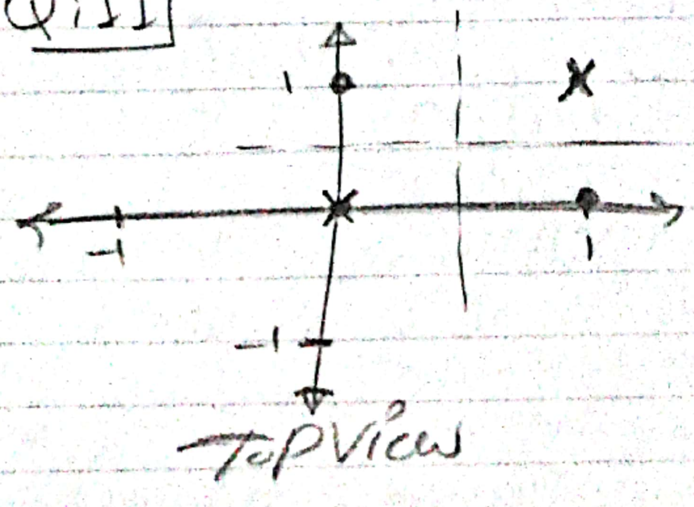
For K=3

Points ①, ②, ③ → C2, C1, C1

so C1 ✓

* The result of the KNN is more accurate, as (1.2,0) in C2 is an outlier that affects the result. KNN solved this by taking the majority vote!

Q:11



Top view

$M_1 \Rightarrow (0,0)$] C_1

$M_2 \Rightarrow (1,1)$

$M_3 \Rightarrow (1,0)$] C_2

$M_4 \Rightarrow (0,1)$

$\boxed{Q:12}$ $\mu = [0.85, 0.16667]$

$\Sigma = \frac{1}{M} \sum (\underline{x} - \mu_1)(\underline{y} - \mu_2)$

X	Y	XY
1	1	1
1.3	0.7	0.91
1.5	-0.1	-0.15
0	-1	0
0.1	0.3	0.03
1.2	0.1	0.12

$E(x) = 0.85$, $E(Y) = 1/6$

$E(xy) = \frac{191}{600}$

$Cov(x, y) = E(xY) - E(x) E(Y)$
$= \frac{191}{600} - 0.85 * 1/6$
$= \frac{53}{300} \approx 0.17667$

$Cov(x) = E(x^2) - E(x)^2$
$= \frac{213}{200} - 0.85^2 = 0.3425$

$Cov(y) = E(y^2) - E(y)^2$
$= \frac{13}{30} - 1/36 = 0.4055$

$$\Sigma = \begin{bmatrix} Cov(x) & Cov(x,y) \\ Cov(x,y) & Cov(y) \end{bmatrix} \checkmark$$

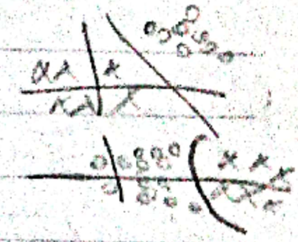
Mid 2018

1) Supervised learning → The goal is to learn a function that maps inputs to outputs, Model $P(C_i|\underline{x})$
we feed it the labeled data
e.g. classify written characters

Reinforcement learning → The input is a state, the model then takes an action and learn from score/Penalty
e.g. give a toy to a child!

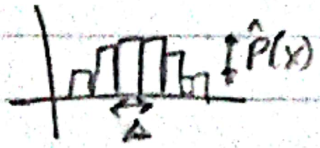
2) obj Detection is Hard → due to the variability of obj, same obj can look diff [rotated, diff size diff angle, —]

3) Linear classifier → use a hyperplane [or line] to divide the features



Non-linear → use a non linear plane or curve

4) histogram → for each bin we count the points that fall in it



$$\hat{P}(x) = \frac{m}{N * \text{binsize}}, \quad K = N P(x) \Delta$$

Dis Cont.

Kernel Density → we use bump func
→ smooth, add weight diff

5] Feature selection → Choose a subset of L (<<N) Features
SFS, SFB — From N, Get rid of irrelevant Features and correlated ones

Feature extraction → Transform the available N features
PCA — into a smaller no of L Features through certain transformation

more compact, no Phy meaning, not suitable to all prob; diff to explain

$\boxed{Q:8}$ $C_1, C_2 \rightarrow$ classify $C = (1.3, 0)^T$

$h = 0.5 \rightarrow N(0, 0.5) = \frac{1}{\sqrt{2\pi}h} \exp\left(\frac{-x^2}{2h^2}\right)$

We want $\max P(C_i|x) \rightarrow \max P(x|C_i) \underbrace{P(C_i)}_{0.5}$

N-Dim $\rightarrow \frac{1}{\sqrt{2\pi}h} \exp\left[\frac{-1}{2h^2} \Sigma x_i^2\right]$ ~~I removed all the const~~

$$P(x|C_i) = \frac{1}{Mh^n} \sum^{M} g\left(\frac{x - x(m)}{h}\right) = \frac{1}{3 \times (0.5)^2 \times 2\pi} \Sigma \exp\left[\frac{-1}{2 \times .5^2} \times \Sigma[x_i - x_i(m)]^2\right]$$

$$= \frac{2}{3\pi}\left[e^{\frac{-8}{9} \Sigma(x_i - x_i(m))^2}\right]$$

$$\Rightarrow P(x|C_1) = \frac{2}{3\pi}\left[e^{-2 \times [(1-1.3)^2 + (1)^2]} + e^{-2 \times [(1.3-1.3)^2 + 0.7^2]} + e^{-2 \times [(1.5-1.3)^2 + .1^2]}\right]$$

$-2 \times [(1-1.3)^2 + (1)^2] \rightarrow \frac{.109}{50} \approx -2.18$

$-2 \times [(1.3-1.3)^2 + 0.7^2] \rightarrow \frac{-49}{50} \approx -0.98$

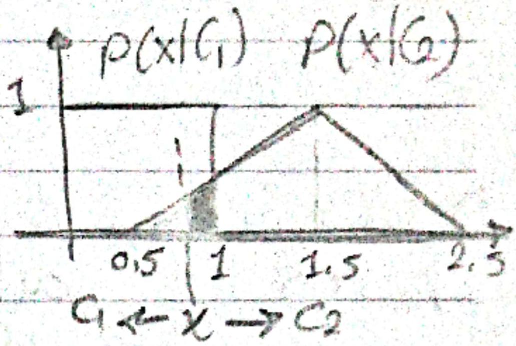
$-2 \times [(1.5-1.3)^2 + .1^2] \rightarrow \frac{-1}{10} \approx -0.1$

$$= \frac{2}{3\pi}[1.393] = 0.2956$$

Same for $P(x|C_2)$ and Choose the max!

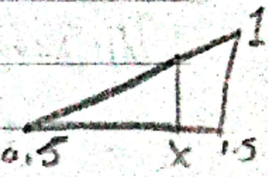
6] $P(C_1) = 0.3$, $P(C_2) = 0.7$

a) Decision region?
b) classification error?



Dec reg → $P(x|C_1)P(C_1) = P(x|C_2)P(C_2)$

$$\underset{y}{1} * 0.3 = \underset{y}{(x-0.5)} * 0.7$$



a) so $x = \frac{13}{14} \simeq 0.928$

$y = x - 0.5$

b) $P(e) = \int_x^{\infty} P(x|C_1)P(C_1)\,dx + \int_{-\infty}^{x} P(x|C_2)P(C_2)\,dx$

$= \int_x^1 1 * 0.3\,dx + \int_{0.5}^{x} (x-0.5) * 0.7\,dx$

$= 0.3[1-x] + 0.7(x-0.5)[x-0.5]$

$= 0.3\left[1-\frac{13}{14}\right] + 0.7(x-0.5)^2$

$= 0.3\Big[\underbrace{1-\frac{13}{14}}_{1/14}\Big] + 0.7\Big(\underbrace{\frac{13}{14}-0.5}_{9/49}\Big)^2$

OR

$P(x|C_2) * P(C_2)$

$\Rightarrow \frac{1}{2} B * H$

$\frac{1}{2} * [x-0.5] * 0.3$

Mid 2019

1) supervised → the input is the labeled data and the out is the expected label.
Model P(y/x) e.g classify letters

unsupervised → learn generative model of input data that tries to describe possible patterns and features of these inputs without the need for labeled output.
Model P(X) e.g clustering and Association

Apriori, Freq Pattern Growth ← dependencies

2] SFS

* Examine each feature x_i, taken alone, design the classifier and select the best Feature

* From the remaining Features, select the one that together with the selected group, give best perform

* Continue in this manner until you have selected (L) features