B-Tech 4th Semester

End-Term Examination

May 2021

1 2 3 ✓ 1 2 continue

Subject: Introduction to Data Science [CS2205]

Date: 27th may 2021

Name: Abhiroop Mukherjee

Exam Roll No.: 510519109

G-Suite ID: 510519109.abhiroop@students.iiests.ac.in

No. of Sheets uploaded: 10

Q.1) a) given $A = \begin{bmatrix} 4 & 11 & 14 \\ 8 & 7 & -2 \end{bmatrix}$

A have is 2×3, ie non-square, so we do as follows

$$A^T \times A = \begin{bmatrix} 4 & 8 \\ 11 & 7 \\ 14 & -2 \end{bmatrix}_{3\times2} \begin{bmatrix} 4 & 11 & 14 \\ 8 & 7 & -2 \end{bmatrix}_{2\times3}$$

$$= \begin{bmatrix} 4\times4 + 8\times8 & 4\times11 + 8\times7 & 4\times14 - 8\times2 \\ 11\times4 + 7\times8 & 11\times11 + 7\times7 & 11\times14 - 7\times2 \\ 14\times4 - 8\times2 & 14\times11 - 7\times2 & 14\times14 + 2\times2 \end{bmatrix}$$

$$= \begin{bmatrix} 80 & 100 & 40 \\ 100 & 170 & 140 \\ 40 & 140 & 200 \end{bmatrix} = B \text{ (let)}$$

→ This is 3×3 symmetric matrix

→ Now lets find it's eigenvalues

$$\therefore \det(B - \lambda I) = 0$$

$$\therefore \begin{vmatrix} 80-\lambda & 100 & 40 \\ 100 & 170-\lambda & 140 \\ 40 & 140 & 200-\lambda \end{vmatrix} = 0$$

(1) (2) (3)

$$\det\left(\begin{bmatrix} 80-\lambda & 100 & 40 \\ 100 & 170-\lambda & 190 \\ 90 & 140 & 200-\lambda \end{bmatrix}\right) = 0$$

i.e $(80-\lambda)[(170-\lambda)(200-\lambda) - 190\times190] - 100[(200-\lambda)100 - 190\times40]$

$+ 40[100\times140 - 40(170-\lambda)] = 0$

i.e $(80-\lambda)[14400 + \lambda^2 - 370\lambda] - 100[8400 - 100\lambda] + 40[7200 + 40\lambda] = 0$

i.e $1152000 - 14400\lambda + 80\lambda^2 - \lambda^3 - 29600\lambda + 370\lambda^2 - 40000 + 10000\lambda$

$+ 288000 + 1600\lambda = 0$

or $\lambda(-\lambda^2 + 450\lambda - 32400) = 0$

or $\lambda(\lambda - 360)(\lambda - 90) = 0$

$\therefore \lambda_3 = 0 \quad \lambda_1 = 360 \quad \lambda_2 = 90 \longrightarrow$ eigen values

now Singular Values $\Rightarrow \sigma_i = \sqrt{\lambda_i}$

$\therefore \sigma_3 = \sqrt{0} = 0$

$\sigma_1 = \sqrt{360} = 18.9736$

$\sigma_2 = \sqrt{90} = 9.4868$

~~for $\lambda = 360$, $\lambda = 0$, eigen vector~~

so sorting in ascending order

$\sigma_1 = 18.9736$

$\sigma_2 = 9.4868$

$\sigma_3 = 0$

so $\Sigma = \begin{bmatrix} 18.9736 & 0 & 0 \\ 0 & 9.4868 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

→ from λ's we can find eigenvector $\vec{v}$'s from that we can find V

now we know $A = U\Sigma V^T$

→ so we can find U to complete full SVD

(2) (9) (18)

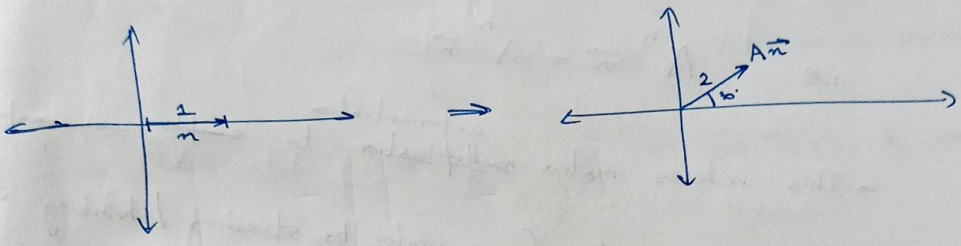
~~1) b)~~

1) b) consider a vector $\vec{x}$ and a matrix A

→ $A \times \vec{x}$ transforms vector $\vec{x}$ to some other vector (say $\vec{y}$), changing it's magnitude and generally direction as well

Eg consider $A = 2\begin{bmatrix} \cos 30^\circ & -\sin 30^\circ \\ \sin 30^\circ & \cos 30^\circ \end{bmatrix} = 2\begin{bmatrix} 0.8666 & -0.5 \\ 0.5 & 0.8666 \end{bmatrix}$

and $\vec{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$



→ for a transformation matrix A, there can exist special vectors $\vec{x}$, such that transformation of $\vec{x}$ using A doesn't change direction of resulting vector.

→ These special vectors are called eigenvectors of A

→ so using definition

$$A\vec{x} = \lambda\vec{x}, \text{ where } \lambda \text{ some constant}$$

or $$(A - \lambda I)\vec{x} = \vec{0}$$

→ for this to be true, we need,

$$\det(A - \lambda I) = 0$$

→ from this relation we can find λ, and their corresponding $\vec{x}$

→ λ's are called the eigenvalues of A, and $\vec{x}$'s corresponding to λ's are called eigenvectors of A

Property

1) if $\vec{x}$ is an eigenvector of A, $A\vec{x}$ will have same direction as $\vec{x}$

2) for an nxn matrix A, using $\det(A-\lambda I)=0$, we get n λ's and corresponding n eigenvectors $\vec{x}$'s

3) when A is powered, eigenvectors of resulting matrix stays same, but eigenvalue gets powered

i.e $A^n \vec{x} = \lambda^n \vec{x}$

→ This reduces matrix exponentiation to scalar exponentiation

4) → Symmetric matrix transforms vector by stretching/shrinking along it's eigenvectors

→ for an nxn Symmetric matrix, we get n linearly independant & orthogonal eigenvectors and corresponding n real eigenvalues

1) c) Limitation of Eigen Vector Decomposition method

→ This method is based on the properties of symmetric matrix [which is always square too]

→ So this method fails when matrix is square but non-symmetric, or is non-square.

→ We use Singular Value Decomposition in that case

Q2) a) ~~Drawbacks of K-Means Algorithm~~

→ K-Means is an unsupervised learning algorithm

→ Idea is to define k cluster center and iteratively update it to reduce overall distance between k centers and data points.

→ In the end it partitions the data set into k non-overlapping Clusters.

→ consider n datapoints and k clusters, we need to minimize the following

$$J(V) = \sum_{i=1}^{n} \sum_{j=1}^{k} (\|x_i - v_j\|)^2$$

where $x_i \Rightarrow$ data point
$v_j \Rightarrow$ cluster center

→ This is done as follows

i) Randomly define k cluster points.

ii) Assign all datapoint, the cluster center which is closest to it

iii) Update cluster point the mean of all the datapoint assigned to it

i.e $$v_j = \frac{1}{c} \sum_{c} x_i$$

where c are the no. of datapoints assigned to cluster v_i

iv) repeat step (ii) & (iii) until no. of epochs or until no substantial updates happen

v) Now you have k non-overlapping clusters.

Problem

1) need to define k manually
2) as it calculate distances, it generally does well on spherical clusters but not so well on non-spherical clusters
3) It cannot identify outliers, it has to be done manually

(Q2) b) I) SVM [Support Vector Machines]

→ SVM's are used for two-group classification problem [two class dataset]
→ for multiclass classification, we do SVM multiple time and report intersections of output

→ SVM → for a given two-group data, SVM will give the best decision boundary with highest margin
→ SVM's are sensitive to noise.

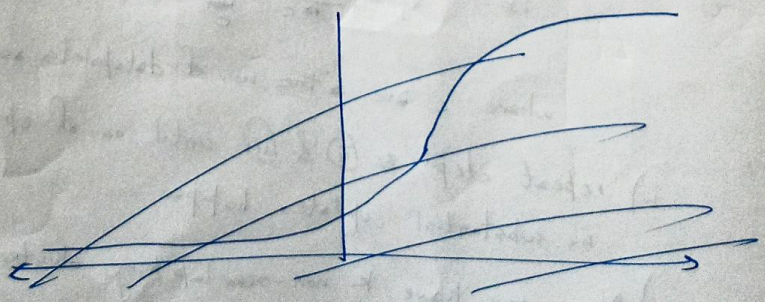
II) Logistic Regression

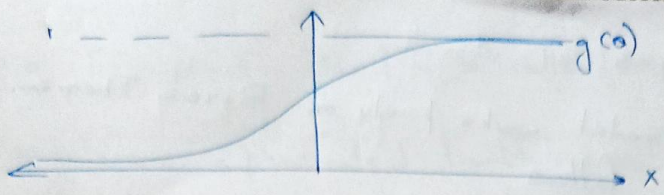
→ This model defines decision boundary as follows

$$g(\theta) = \frac{1}{1 + e^{-\theta^T x}}$$ [called sigmoid function]

→ for an input X, if value of $g(\theta) \geq 0.5$, it is defined as class A, if it is < 0.5, it is defined as class B [two-class dataset]

→ we train $g(\theta)$



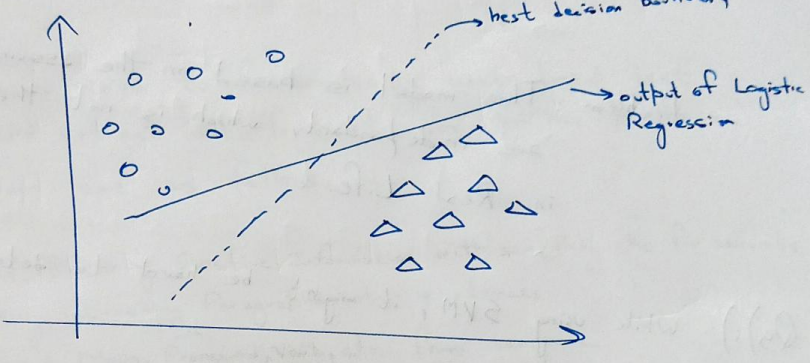


→ $g(\theta)$ gives probabilistic output between 0 and 1

→ we train θ using maximum likelihood estimate to get a decision boundary which best fits the given data.

→ This method doesn't confirm that decision boundary we have highest margin

Eg



→ This Algorith is also two class classification algorithm, so to do multiclass classification, we do this multiple time.

Eg: Three group : ~~Ball~~ Square, Triangle, Circle

1) Square vs not-Square
2) Triangle vs not-Triangle
3) Circle vs not-Circle.

→ we take intersection of these three to figure out what the actual classification is.

III) Naive Bayes

→ This model works purely on Bayes Theorem, which is as follows

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

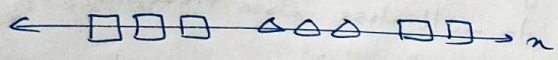
→ This model ~~is~~ can be easily used for multiclass dataset

→ given B, we calculate P(A|B) for multiple classes A, and select the A with highest P(A|B)

Problem: This model is based on the assumption that predictors are independant, which is not the case generally in Real Life.

Q5):) While using SVM, it might be hard to determine a decision boundary

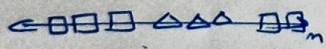
Eg



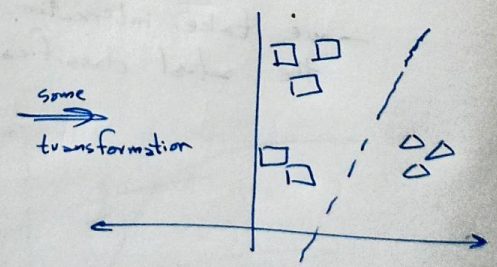
→ here it's hard to determine a decision boundary ~~for~~ here

→ we can, hopefully find some transformation, which can change given data to some higher order data, which might be ~~easier~~ easier on SVM

Eg



[Hard for SVM]



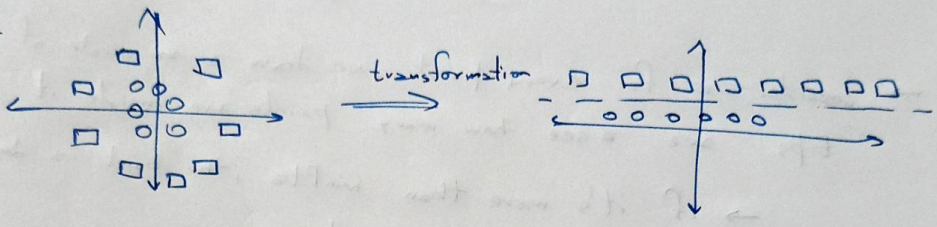
[Easy for SVM]

→ This Transformation Function is called Kernels

→ Kernels are sophisticated transformation function which maps given data to some other data space, generally of higher order

→ This is done in hope to make SVM classification easier

Eg



Q5) ii) There are many ways to for analysing unstructured text document and extract data from it for further purposes. Some of them are as follows

I) Tokenization : Replacing Senstive Information with something else for security
II) Sentence Breaking : Breaking Paragraphs into sentences
III) Chunking : Find Noun, Pronound, Verb, etc. from sentence
IV) Syntax Parsing : Analysing Meaning of Phrase to determine correct/incorrect
V) Language Identification : Identification of Language Used.
VI) Parts of Speech Tagging : Tagging words as parts of speech
VII) Sentence Chaining : Linking words in a sentence together.

Q5) iv) → DBSCAN [Density-Based Spatial Clustering of Applications with Noise] uses Density as a metric to calculate Clusters

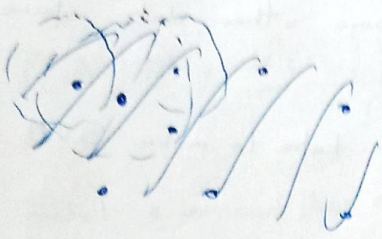
→ It works based on following two parameters

a) minPts : min. no. of points (a threshold) clustered together to ~~define~~ consider a region densed.

b) Eps : Radius metric to determine cluster.

How these two plays ~~together~~ together

→ Consider a ~~data~~ set of points as followed



→ ~~consider~~ for every point, we draw a ~~the~~ circle of radius Eps and see how many points are inside it

→ if it's more than minPts, the point is dense and considered a core point

→ if it's more than 1 and less than minPts, it's a border point

→ if it's only 1, it's a noise data.

→ Using above rules, we form cluster of a dataset, which is based on density.

v) given a $n \times n$ matrix A, it will have n eigen vectors and n corresponding eigen value

→ eigenvalues can be found by $\det(A - \lambda I) = 0$

→ corresponding eigenvectors can be found by

$$(A - \lambda I) \times \vec{v} = \vec{0}$$

→ ~~for~~ consider a case where A is symmetric matrix, we observe that ~~eigen~~ eigenvectors are orthogonal to each other and it stretch/shrinks any given vector ($\vec{m}$) along along it's eigenvector and magnitude of stretch/shrink is determined by eigen value.

→ The condition to get orthogonal ~~eigen~~ vectors eigen vector is that A should be symmetric