DAY:_____ DATE:_____

IDS-15 (20/03/24)

KNN ⇒ reg + classification.

K-mean ⇒ clustering.

Classification (supervised learning)

↳ label = categorical.

KNN ⇒ classification + reg.

↳ if numbers ⇒ % of yes/no

↳ actual − calculated = error.

↳ no learning (just store data during training)

↳ for reg ⇒ own exercise.

↳ for classification.

⇒ data is used as it is

⇒ lazy learner (learn while predicting).

⇒ k ⇒ usually odd value selected.

⇒ relatively better at binary classification rather than multi class.

1- Find distance from all pts 2- Sort distance

3- select top k values.

↳ usually euclidean distance used.

$\sqrt{(7-3)^2+(7-7)^2} = \sqrt{16+0} = 4, A$ — 3

$\sqrt{(7-3)^2+(4-7)^2} = \sqrt{16+9} = 5, A$ — 4

$\sqrt{(3-3)^2+(4-7)^2} = \sqrt{0+9} = 3, B.$ — 1

$\sqrt{(1-3)^2+(4-7)^2} = \sqrt{4+9} = 3.61, B.$ — 2

3, B
~~3.61, B~~ } → B
4, A
5, A

So point is B. — B

DAY: _____ DATE: _____

IDS-16 (22/03/24)

Unsupervised learning:

Clustering ⇒ K-mean.

↑ Inter cluster distance: distance b/w pts in d/f clusters

↓ Intra cluster distance: " " " within cluster

K >= 2

↳ particular method used to find value of k.

(x,y) (x,y) (x-x, y-y) ⇒ distance

1) k = 3.

2) c_1, c_2, c_3 ⇒ select randomly but have difference.

3) Iterations
- distance of each point from every centroid.
- assign point to centroid with min distance.
- Avg the distance, update the centroid.

↳ repeat until no updation in centroid values

Elbow Method (for value of k)

↳ generally used to find k.

Expectation-Maximization

$$\sum_{j=1}^{c} \sum_{i=1}^{n} \| x_i - c_j \|^2$$

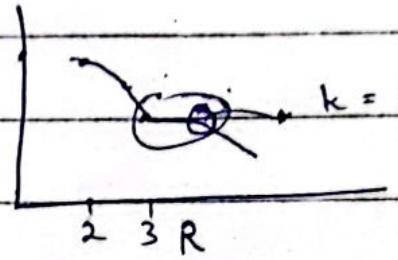
x_i ⇒ data points, c_j ⇒ no of cluster

8:35
9.50
10:45
11:40

DAY:_____ DATE:_____

1) Elbow Method:

- sum((distance of point from cluster-assigned)^2)
  - ↳ for d/f values of k. (e.g 1 to 10)



2) Silhouette Method:

↳ usually used for verification.

a = distance of point from centroid it belongs.

b = distance // // it does not belong.

more +ve value ⇒ good no of k

more -ve value ⇒ worst no of k.

from -1 to +1.

IDS-17 (27/03/24)

Evaluation Metric

→ Confusion Matrix (for classification)

↳ False alarm (FP)

overfit ⇒ Type 1 error ⇒ all FP

DAY:_____ DATE:_____

TP FP
FN TN

Confusion Matrix Example.

TP: 3 TN: 4 FP: 1 FN: 2

Positive class: Dog

3	1
2	4

Accuracy: $\frac{3+4}{3+4+1+2} = \frac{7}{10} = 70\%$

Precision: $\frac{TP}{TP+FP} = \frac{3}{3+1} = 75\%$

Recall: $\frac{TP}{TP+FN} = \frac{3}{3+2} = 60\%$

IDS-18 (29/03/24)

Naive Bayes ⇒ collection of algos.

Conditional Probability:

→ 1) Assume features are independent.

Naive ⇒ Basic ↓ equal contribution on each other

$$P(y|x) = \frac{P(x|y) * P(y)}{P(x)}$$

$$P(y|x_1, x_2) = \frac{P(x_1, x_2|y) * P(y)}{P(x_1, x_2)}$$

DAY:_____ DATE:_____

$$P(y|x_1, x_2) = \frac{P(x_1|y)\,P(x_2|y)\,P(y)}{P(x_1)\,P(x_2)}$$

⇒ neglecting denominator, as always dividing by 1 value (fixed), errors (diff) will be same for all values

Example:

⇒ $P(\text{sunny}|\text{yes}) = \frac{\text{len}(\text{sunny \& yes})}{\text{len}(\text{yes})}$

↳ model will have calculated all individual probabilities.

↳ convert strings to numbers.

⇒ All probabilities are multiplied, if one is 0 ⇒ Ans 0.

Laplacian Smoothing:

→ Start with 1 instead of 0.

1) • Multinomial Naive Bayes ⇒ doc classification.
2) • Bernoulli Naive Bayes ⇒ Boolean value (2 types)
3) • Gaussian Naive Baye ⇒ when normally dist

Example:

P(Read) = 4/7	P(Exploitable) = 5/7	P(small) = 2/7
P(write) = 3/7	P(Non-Exec) = 2/7	P(Medium) = 3/7
		P(Large) = 2/7

DAY:_____ DATE:_____

P(write|infected) = 1/4 P(write|clean) = 2/3

P(exec|infected) = 4/4 P(exec|clean) = 0/3

P(large|infected) = 0/4 P(large|clean) = 2/3

Apply Laplacian Smoothing

P(Write|Infected) = 2/5 P(Write|clean) = 3/4

P(Exec|Infected) = 5/5 P(Exec|clean) = 1/4

P(Large|Infected) = 1/5 P(large|clean) = 3/4

P(Read) = 5/8

P(write) = 4/8

P(Exec) = 5/8

P(Large) = 3/8

P

IDS-19 (03/04/24)

Spam filter

P = p1 * p2 * pn multiply probability of n words

1-P ⇒

Example:

$$P(S|W) = \frac{P(W|S) * P(S)}{P(W)}$$

Rev.

DAY: _____ DATE: _____

P(S | "Review us Now") = ?

P(S) =

=> If probability = 0 in prior probabilities

then apply laplasian smoothing.

↳ Add 2 in denominator (total)

total records + = no of classes

$P_r(\{1, 0, 1, 0, 00\})$ — P, 1-P

↳ Each index indicating presence/absence of word in table (for multiplication).

=> Unknown word => ignore.

=> If prob of spam > threshold => spam

spam 12.3%

ham 87.6%

or spam > ham => spam

$$\frac{0.0625 \times 2/6}{0.0044 \times 4/6 + 0.0625 \times 2/6}$$

IDS-20 (05/04/24)

Data Preprocessing:

↳ Most imp part of DS.

Data collection + preprocessing => 80% time.

DAY:_____ DATE:_____

- Remove Irrelevant Data:

100% same (variance = 0) or 100% d/t (emails) => no info

Standardize Capitalization

↳ when working with text.

↳ solv: generally all text converted to lower case.

Convert Data Type:

Handle Missing Values

↳ some models don't work on missing values.

• MCAR	• MAR	• MNAR
↳ no relationship of missing data	↳ you can identify why data is missing.	↳ ~~identification~~ ✓
↳ why missing? X	↳ missing within range	↳ investigate
		↳ only not related with collected data

Transformation:

- Label encoding => biasness to large values.

- One hot encoding => vector to each category.

↳ No one hot encode for label

↳ only for feature

DAY:_____ DATE:_____

Feature Scaling:

- Standardization
  - mean = 0
  - variance = 1
  - (prefer) as we assume data is normalized.
- Normalization.
  - (0 - 1)

Unsup learning: IDS-21 (19/07/24)

→ zero or near zero variance:
1- find variance of all columns in features.
2- If variance ≈ 0 ⇒ remove.

high-correlation ⇒ same behaviour.
↓
multicolinearity ⇒ use one feature.
$f_1 \uparrow\downarrow f_2$

Supervised Learning:

→ Filters Methods ⇒ single feature anglysis with y.
highly correlated is more good in classification

feature → label

(All feat) cat, num → cat, num (cat, reg)

categorical ⇒ numerical (rare)
cat ⇒ cat (social science)

Num ⇒ Num (Correlation)
↳ pick top n highly correlated features.
↳ Accuracy may decrease a bit.

DAY:_____ DATE:_____

2) Num → Cat. (Info Gain / Mutal Info)

↳ Also for cat to cat

→ **Wrapper**

↳ Time consuming.

1) forward Selection

KNN(f1, y)

KNN(f2, y)

KNN(f3, y) ✓ → Highest Acc

(cannot go back)

KNN(f1, f3, y) ✓

KNN(f2, f3, y)

2) Backward Elimination.

3)

Decision Tree ⇒ Reg + Classification.

Entropy ⇒ Randomness.

$$\log_2(x) = \frac{\log_{10}(x)}{\log_{10}(2)}$$

1st find entropy of label

$$E(5,9) = \left(\frac{-9}{14}\log_2\frac{9}{14}\right)_{\text{Yes}} + \left(\frac{-5}{14}\log_2\frac{5}{14}\right)_{\text{No}}$$

$\sum -P_i \log_2 P_i$

DAY:_____ DATE:_____

↳ for multiple attribute:

↳ create table like Naive Baise.

↳ entropy of label with individual attribute col.

$$E(T,X) = \sum_{c \in X} P(c) E(c)$$

$$P(\text{sunny}) * E(\underset{\text{Yes, No}}{\overset{\text{Sunny, Sunny}}{}})$$

$$= \frac{5}{14} * E(3,2)$$

$$= \frac{5}{14} * \left[\left(\frac{-3}{5}\log_2\frac{3}{5}\right) + \left(\frac{-2}{5}\log_2\frac{2}{5}\right)\right]$$

$$\boxed{E(a,b) = \left(\frac{-a}{a+b}\log_2\frac{a}{a+b}\right) + \left(\frac{-b}{a+b}\log_2\frac{b}{a+b}\right)}$$

Info Gain:

Entropy of whole sys − entropy of 1 attribute

Example: Select most imp col:

1) Entropy of whole sys:
2) Entropy of each attr. with label.
3) Gain(T,X) = Entropy(T) − Entropy(T,X)
4) Separate table & again check for entropy.
   - ↳ if no entropy ⇒ direct decision.
   - ↳ if entropy ⇒ repeat.

DAY:_____ DATE:_____

IDS-22 (24/04/24)

Dimensionality Reduction:

⇒ Linear Transformation:

In Maths / Straight line ⇒ Straight line.

↓ Eigen vector ↓ (stretch, compress)

Matrix * Vector ⇒ Vector

↓

gets scaled up/down by int value.

⇓

Eigen value (non zero)

& the vector is called eigen vector.

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} * \begin{bmatrix} 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \end{bmatrix}$$

$$2 \begin{bmatrix} 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \end{bmatrix}$$

2 ← Eigen value. $\begin{bmatrix} 4 \\ 5 \end{bmatrix}$ ← Eigen vector

⇒ Matrix multiplied by its eigen vector, its shape is not changed.

$$A\vec{v} = \lambda\vec{v}$$

↓ Matrix ↓ vector ↓ scale no / eigen value

DAY:_____ DATE:_____

$$A\vec{v} - \lambda\vec{v} = 0.$$

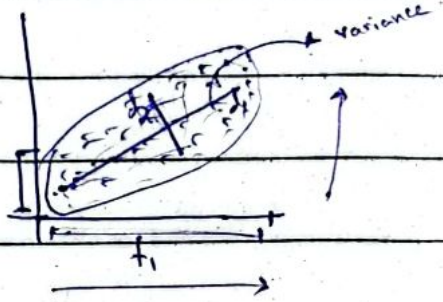
$$\vec{v}(A - \lambda I) = 0 \qquad I = \text{Identity}.$$

Principal Component Analysis

$f_1\ f_2\ f_3\ f_4\ f_5 \Rightarrow pf_1\ pf_2$

$pf_1 = 0.3f_3 + 0.15f_2 + 0.13f_2 + 0 + 0.005f_2$

pf is not selection but transformation (linear).



Always 90°

V_1 V_2

$\lambda = \lambda_1 + \lambda_2$

PC1 PC2

$\Rightarrow d_1$ is greater $\Rightarrow$ greater variance.

$\Rightarrow$ greater variance $\Rightarrow$ more info

variance high $\rightarrow f_3$ (more learning)

variance low $\rightarrow f_7$ (no learning).

S.D $\Leftrightarrow$ Eigen value (biggest) $\rightarrow$ Eigen vector

λ_1

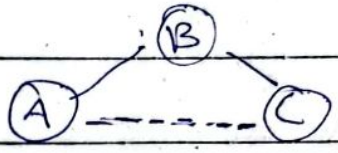
DAY:_____ DATE:_____

IDS- (7/05/24)

Graph Data Science

1) Adjacency List.
2) Adjacency Matrix.
3) Edge List

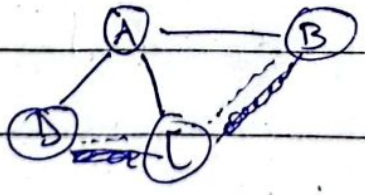
Clustering Coefficient:



↳ separate for each node.

No of edges of node = $\frac{k(k-1)}{2}$ k = no of connected nodes.

$\frac{n_i}{\frac{k(k-1)}{2}} \Rightarrow \frac{2n_i}{k(k-1)}$



$C_i = \frac{2 \times 1}{3 \times (3-1)} = 0.33$