29/10/2022

Aniket
Date
Page

Sessional – 2 (2021)

Sol 1. Initial centroids are $A_2(3,5)$, $B_2(7,5)$, $C_2(4,9)$.

Now using Euclidean Distance formula, $d = \sqrt{(y_2-y_1)^2 + (x_2-x_1)^2}$

Distance from each centroid in order (A_2, B_2, C_2) over first iteration:

Points	Distance from A2	Distance from B2	Distance from C2
$A_1(4,11)$	6.08	6.70	2.0
$A_2(3,5)$	0	4.0	4.12
$A_3(8,4)$	5.09	1.41	6.40
$B_1(5,8)$	3.6	3.6	1.41
$B_2(7,5)$	4.0	0	5.0
$B_3(6,4)$	3.1	1.4	5.3
$C_1(1,2)$	3.6	6.7	7.6
$C_2(4,9)$	4.1	5.0	0

cluster 1 (A_2) : $\{A_2(3,5), C_1(1,2)\}$
cluster 2 (B_2) : $\{A_3(8,4), B_2(7,5), B_3(6,4)\}$
Cluster 3 (C_2) : $\{A_1(4,11), B_1(5,8), C_2(4,9)\}$

Now, calculating new clusters centroids :-

$$\left(\frac{3+1}{2}, \frac{5+2}{2}\right), \left(\frac{8+7+6}{3}, \frac{4+5+4}{3}\right), \left(\frac{4+5+4}{3}, \frac{11+8+9}{3}\right)$$

$\Rightarrow$ (Cen1) $(2, 3.5)$, (Cen2) $(7, 4.33)$, (Cen3) $(4.33, 9.33)$

cen → Centroid

which are the required clusters centroids after 1st round of execution

Ans 1(a)

Now, re-calculating distances from new centroids over 2nd iteration, we get :-

points	Distance from Cen1	Distance from Cen2	Distance from Cen3
A1	7.7	7.31	1.69
A2	1.8	4.05	4.33
A3	6.02	1.05	6.47
B1	8.4	4.17	1.49
B2	5.2	0.66	3.08
B3	4.03	1.03	3.38
C1	1.8	6.43	8.03
C2	5.83	5.54	0.47

cluster 1 :-
cluster 2 :-
cluster 3 :-
} all same as prev.

re-calculating centroids :-

(2, 3.5), (7, 4.33), (4.33, 9.33)

As, center of newly formed clusters does not changed as it is in 1st iter. Also, the points (data points) remain present in the same cluster. Hence, these 3 centroids are the final centers of the final cluster.

Ans 1(b)

Aniket
Date
Page

Sol 2. The K-means algorithm is not an ideal option to find the global optimum because to make sure it reaches the global optimum we have to repeat the clustering exercise multiple times i.e., we have to iterate over all possible clusterings and then take the most optimum solution out of it. But this gives the results in exponential runtime.

Ex:- (for justification)

Assume a cluster, $A = \{1, 2, 3, 4\}$
Atleast two diff^{t} settings satisfy the stationary condition of K-means :-

setting 1

center 1 = 1 , cluster 1 = {1}
center 2 = 3 , cluster 2 = {2, 3, 4}

As the objective is 2 & it is a saddle point, consider epsilon.

setting 2

center 1 = 1.5 , cluster 1 = {1, 2}
center 2 = 3.5 , cluster 2 = {3, 4}

$\Rightarrow$ objective = $\frac{1}{4}$

If K-means was used as the first (setting 1), it would be stuck without giving us the global optimum.

Hence justified.

chi-Squared

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

observed value, expected value

Aniket

Sol 3. Chi-square test :-

↳ It's a statistical test used to examine the diff^n b/w categorical variables from a random sample in order to judge goodness of fit b/w expected and observed results.

Using Formula, expected counts = $\frac{(\text{row total}) * (\text{column total})}{\text{table total}}$

The expected count table :-

	High School	Bachelor	Masters	Phd	Total
Female	50.75	51.765	53.238	47.19	203
Male	49.25	50.235	51.171	45.8	197
Total	100	102	105	93	400

$$\therefore \chi^2 = \frac{(60-50.75)^2}{50.75} + \frac{(57-51.76)^2}{51.76} + \frac{(50-53.28)^2}{53.28} + \frac{(36-47.19)^2}{47.19} + \frac{(40-49.25)^2}{49.25} + \frac{(45-50.23)^2}{50.23} + \frac{(55-51.71)^2}{51.71} + \frac{(57-45.8)^2}{45.8}$$

$$= 9.26$$

The critical value of χ^2 with 3 degree of freedom is 7.815. (from standard Table of χ^2_{critical} values)

$\therefore$ 9.26 > 7.815

we reject the null hypothesis and conclude that the education level depends on gender at a 5% level of significance

Sessional-2 (2019)

sol 1 → not in sessional-2 syllabus. (except only Bayesian Belief Network)
sol 3 → " " " " " " .

sol 2 (a.) "Overfitting is a danger when learning a classifier, but not when doing unsupervised learning."

Yes, it's true. Overfitting has little to do with whether the setting is supervised or unsupervised. Essentially, you can break your data points into two components — pattern + stochastic noise.

Now, the goal of machine learning is to model the pattern & ignore the noise. Anytime an algorithm is trying to fit the noise in addition to the pattern, it is overfitting.

~~In the supervised learning,~~
In the supervised setting, you typically want to match the output of a prediction function to your training labels. So in the driving example, you want to accurately predict the steering angle and speed. As you keep adding more and more variable — like curvature of the road, model of car, experience of driver, weather, mood of driver, etc — you tend to make better & better predictions on the training data. However, beyond a point, adding more variables is not helping in modelling the pattern, but only trying to fit the noise. Since, the noise is stochastic, this doesn't generalise well to unseen data & therefore, you have low training error & high test error.

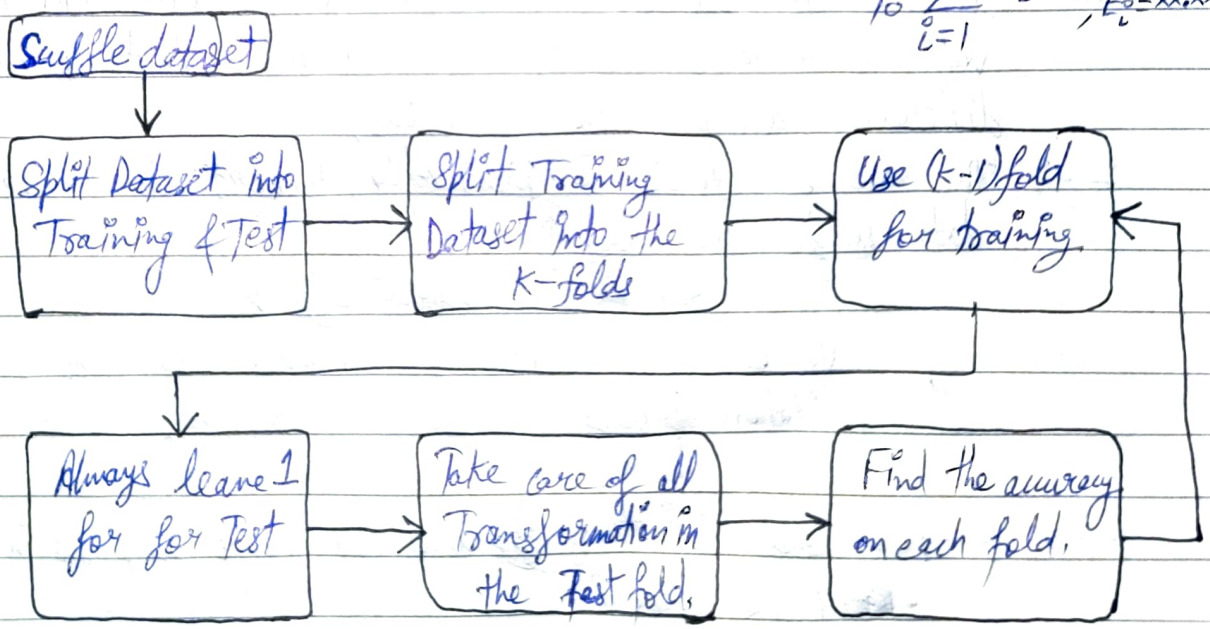
In the unsupervised ~~learning~~ setting, you have some notion of the quality of the solution. For instance, the classic unsupervised problem is clustering, where one measure of quality of the solution is the similarity of points within a cluster. As you keep forming more and more clusters, the similarity keeps increasing, but again, instead of clustering points with very similar values into a single cluster, you tend to assign them to finer clusters, at which point you are fitting the noise.

Ofcourse, you have the underfitting problem at the other end of spectrum, but on unsupervised learning overfitting is not a danger.

point can be added → Distinction b/w noise & pattern is not obvious in most cases. So, you don't have foolproof methods to model only the pattern & ignore the noise completely.

sol 2 (b.) "With K-fold cross-validation, larger K is always better."

$$E = \frac{1}{10}\sum_{i=1}^{10} E_i \quad , E_i = xx.xx$$



Life cycle of K-fold Cross Validation

→ In each set (fold) training & the test would be performed precisely once during this entire process.
→ It helps us to avoid overfitting.
→ As we know, when a model is trained using all of the data in a single short and give the best performance accuracy.
→ To resist this, k-fold cross-validation helps us to build the model as a generalised one.

Thumb Rules Associated with K Fold :-

→ K should be always >= 2 & = number of records. (LOOCV)
  → If 2 then 2 Iterations only
  → if K = no. of records in dataset, then 1 for testing (say n) & (n-1) for training.

Ans 26.)
→ optimised value for K is 10 and commonly used with the data of good size.

→ If the K value is too large, then this will lead to less variance across the training set and limit the model currency difference across the iterations.
→ The no. of folds is indirectly proportional to the size of the data set. ie, folds ↑ ∝ $\frac{1}{\text{dataset size} \downarrow}$
→ Larger values of K eventually increase the running time of the cross-validation process.

→ K-Fold Cross Validation used for many purposes in ML stream.
- Model selection
- Parameter tuning
- Feature selection.

→ The ans is NO. Good choice of K (say 10) is good for large dataset but that's not the case always as increasing folds reduces biases or variance in training sets but it increases the runtime of solution.
Hence, inspite of this slow execution runtime K-Fold cross-validation plays a critical role in ML world. But with K-Fold CV, larger K is not always better.)

Ans 2(b)

Sol 1. (in short) Bayesian classification is based on Baye's Theorem. Bayesian classifiers are the statistical classifiers. Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.

Acc^{n} to Baye's Theorem, $P(H/X) = \frac{P(X/H) \cdot P(H)}{P(X)}$

$P(H/X)$: Posterior Probability; $P(H)$: Prior Probability; $P(X)$: a data tuple; and H → Hypothesis

Bayesian Belief Networks

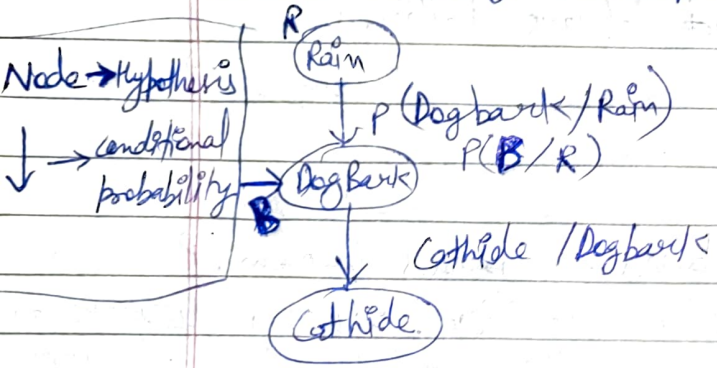
→ Bayesian Belief Networks specify joint conditional probability distributions.
(also known as Bayesian networks, probabilistic Networks).

→ A Belief Network allows class conditional independencies to be defined b/w subsets of variables.
→ It provides a graphical model of casual relationship on which learning can be performed.
→ We can use a trained Bayesian Network for classification.

→ Bayesian B.N. is convinient for representing probabilistic relation b/w multiple events.

→ two components that define a Bayesian Belief Network:-
- → Directed Acyclic Graph.
- → A set of conditional probability tables.

(1) Directed Acyclic Graph:-

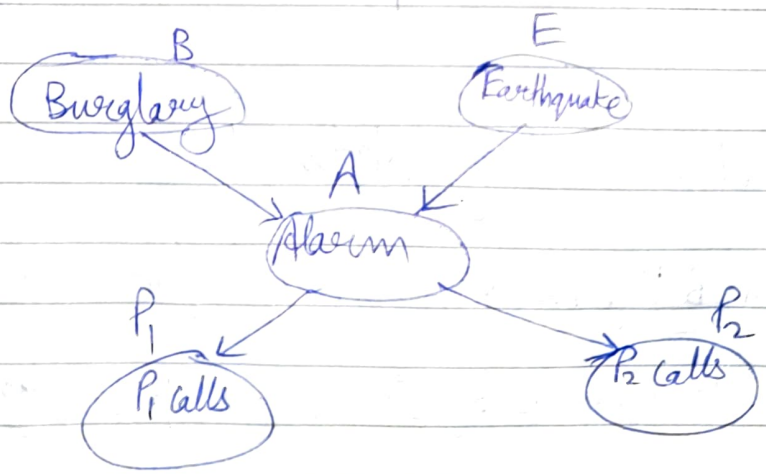


(2) Conditional probability Table:

	R	~R
B	9/48	18/48
~B	3/48	18/48

(B=T & R=T) = 0.19
(B=T & R=F) = 0.375
(B=F & R=T) = 0.06
(B=F & R=F) = 0.375

eg:- (DAG)



P(B=T) = 0.001 , P(E=T) = 0.002
P(B=F) = 0.999 , P(E=F) = 0.998

Probabilities of Parents of A(Alarm).

B	E	P(A=T)	P(A=F)
T	T	0.95	0.05
T	F	0.99	0.06
F	T	0.29	0.71
F	F	0.001	0.999

A	P(P1=T)	P(P1=F)
T	0.90	0.10
F	0.05	0.95

A	P(P2=T)	P(P2=F)
T	0.70	0.30
F	0.01	0.99

$$\Rightarrow P(P_1, P_2, A, \sim B, \sim E) = P(P_1/A) \cdot P(P_2/A) \cdot P(A/\sim B, \sim E) \cdot P(\sim B) \cdot P(\sim E)$$

$$= (0.90) \cdot (0.70) \cdot (0.001) \cdot (0.999) \cdot (0.998)$$
$$= 0.000628$$
$$\simeq 0.00063$$

means $P_1 \to T$, $P_2 \to T$, $A \to T$, $(\sim B) \to$ No Burglary (means $B \to F$), $(\sim E) \to$ // $(E \to F)$

→ Bayesian Belief Network vs Naive Bayes classification:-

A Bayesian network models relationships b/w features in a very general way. If you know what these relationships are, or have enough data to derive them, then it may be appropriate to use a Bayesian network.

A Naive Bayes classifier is a simple model that describes particular class of Bayesian Network — where all of the features are class-conditionally independent. Because of this, there are certain problems that Naive Bayes cannot solve.

(example next page)

Aniket
Date
Page

However, its simplicity also makes it easier to apply, and it requires less data to get a good result in many cases.

ex:- XOR, learning problem with binary features x_1 & x_2 & a target variable $y = x_1 \text{ XOR } x_2$

In a Naive Bayes classifier, x_1 & x_2 must be treated independently — so you would compute things like "The probability that $y=1$ given that $x_1=1$" — hopefully you can see that this isn't helpful, because $x_1=1$ doesn't make $y=1$, any more or less likely. Since a Bayesian network does not assume independence, it would be able to solve such a problem.

→ Naive Bayes is just a constrained form of a more general Bayesian Network.