Unit-3

→ Mining Data Streams :- Process of extracting knowledge structures from continuous rapid data.
Ex:- ATM transaction, web searches, Phone Conversation, Sensor data.

→ Introduction to Stream Concepts
  ↳ Temporally ordered, fast changing, massive
  ↳ Huge volume of continuous data
  ↳ Fast changing

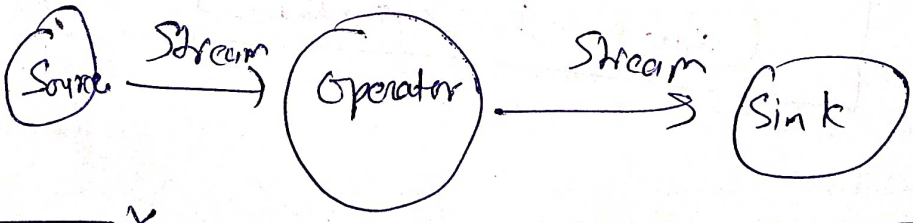
Data Stream mining is the process of extracting knowledge from continuous rapid data records which comes to the system in a stream.

→ A data stream is an ordered sequence of instances in time.
→ Data Stream transmit from a source & recieves at processing end.
→ Also refers to communication of bytes

Ex:-
→ Monitoring & detection
→ analysis of social media
→ Watching online video lecture (Rewind, forward)

Data Stream Model.

→ Stream is data in model
→ Three approaches for updating the end points



Types of Data Stream

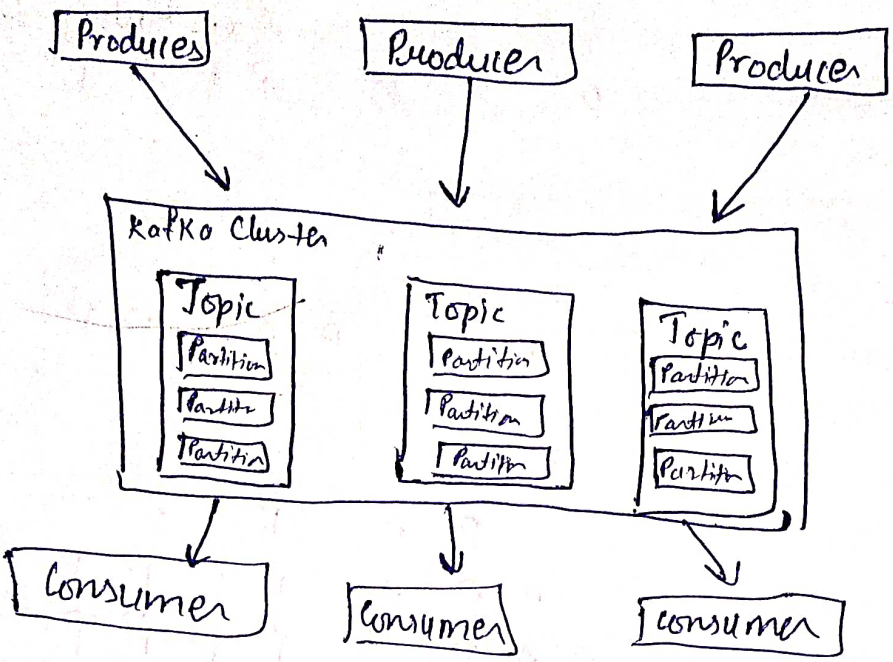
- Data Stream :- Sequence of tuples
- Transaction data Stream:- Credit Card, Web
- Measurement data Stream :- Sensor data, IP Network, Earth Climate

Model & Architecture

Continuous flow
Stream management
Can be done externally.

Tweets

Architecture

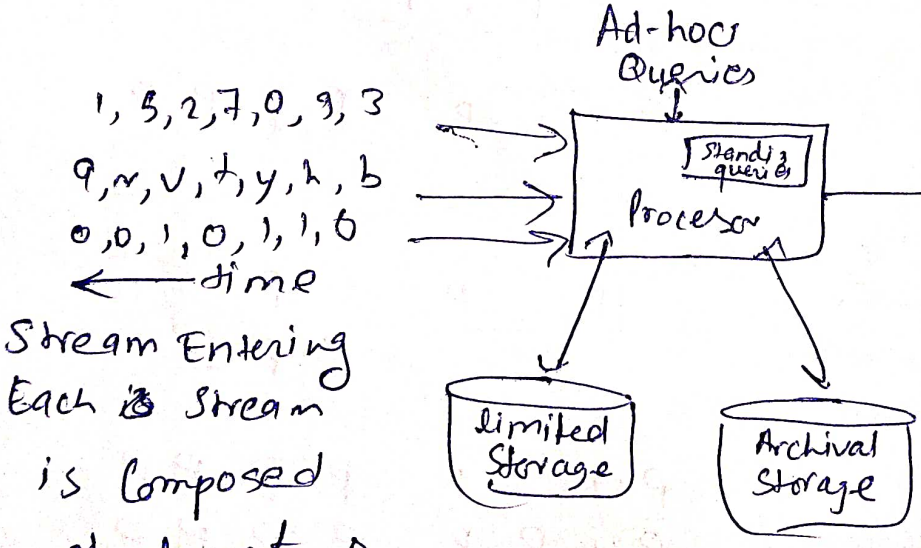


Stream Processing
- Data scope
- Data size
- Performance

Model:-

→ The input element enter at a rapid rate at one or more input ports.

→ The system cannot store the entire stream accessibly.



Stream Entering Each is Stream is Composed of element of tuples.

→ Types of queries
→ filtering a data stream
→ Counting distinct elements
→ Estimating moment
→ finding frequent element

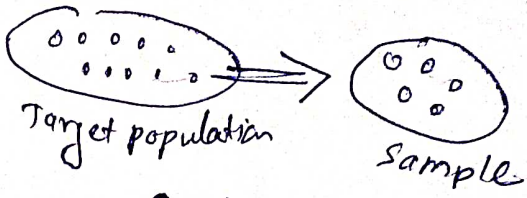
Sec-A
63
7S - B

~~Architecture~~ Sampling data in a stream

→ Since we can not store the entire stream, one obvious approach is to store a sample → random selection

→ Two different problems:- Probability, Non Probability
- (1) Sample a fixed proportion of element in stream. → every member has equal chance to be selected
- (2) maintain a random sample of fixed size over a potentially infinite stream

(1) Sampling fixed proportion



Search engine query

- Stream of tuple (user, query, time)
- Answer question such as: How often did a user run the same query in a single days
- Have space to store 1/10th of query stream
- Random integer

Covid → Vaccine ⇒ different bg people

Ex:- Say $s = 2$

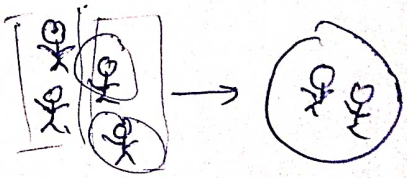
Stream:- a x c y z k c d e g ....

At $n = 5$, each of the first 5 tuples is included in the sample S with equal prob.

At $n = 7$, each of the first 7 tuples is included in the sample S with equal prob

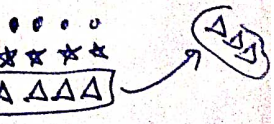
Probability Sample

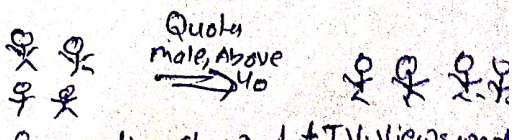
- Simple Random:- equal chance of selection
- Systematic sampling:- every nth element selected
- Cluster Sampling:- randomly selecting cluster
- Stratified Sampling:- dividing the population in groups



Non Probability Sampling

- Convenience Sampling:- sample based on convenience
- Snowball Sampling:- Select sample & ask them to refer them
- Quota Sampling:- Some characteristics
- Judgemental Sampling:- based on judgement





* Rule Induction

→ actions are given and we have to discover the rules.

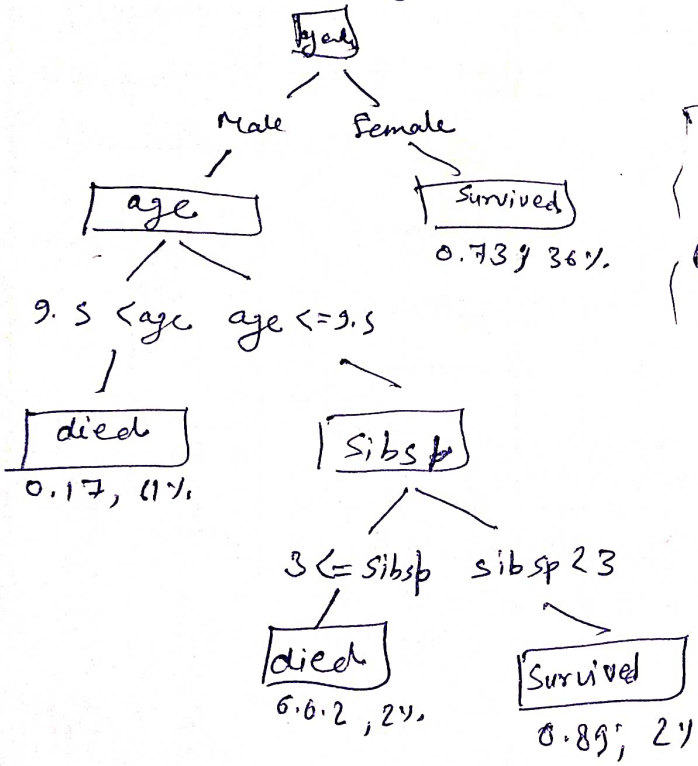
→ ex:- Do not give discount on 2 items. At 8 a.m on weekday, traffic is heavy. At 8 p.m on Sunday the traffic may be light.

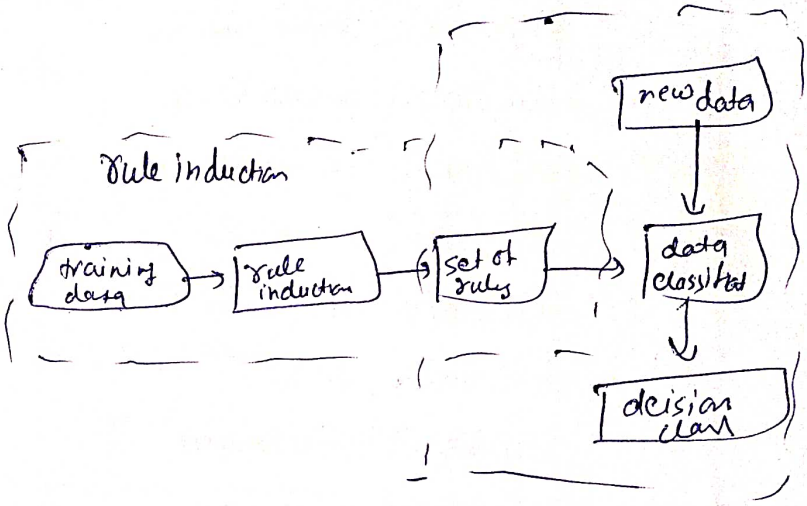
→ This scheme make use of If -Then rule.

→ This is the reverse of a rule-based agent, where the rules are given and the agent must act.

- Prevalence = Probability that LHS & RHS occur together
- Predictability = probability of RHS given LHS.
- Builds a Decision tree representing rule for distinguish between different cases.

Survival of Passengers in Titanic





Neural Networks

→ Collection of neurons that take input and, in conjuction with information from other nodes, develop output without programmed rules.

→ A simple neural network has an input layer, output layer and and hidden layer.

→ These neurons are computational units, These networks transform data like the pixels in an image or the word in an document until they can classify it as an output, such as naming an object in an image or tagging unstructured text data.

→

Learning and Generalizations

Neural Networks learn by adjusting the strength of their connections to better convey input signals through multiple layer of to neurons associated with the right general concept.

→ Learning :- The network must learn decision surfaces from a set of training patterns so that these training patterns are classified correctly.

→ Generalization :- After training, the network must also be able to generalize, i.e. correctly classify test patterns it was never seen before.

~~Competitive learning Rule~~

~~→ In CL, neurons compete among themselves to be activated.~~

Learning
- → Supervised :- Applications :- Speech Recognition, Bioinformatics, Spam Detection, object Recognition
- → Unsupervised. Amazon using clustering.

Supervised

- → are trained using labeled data
- → takes direct feedback to check if the output is correct or not.
- → Simpler method
- → Predicts the Output
- → More Accurate
- → Categorized in Classification & Regression problems
- → ex:- House prices, Cheating Weather

Unsupervised

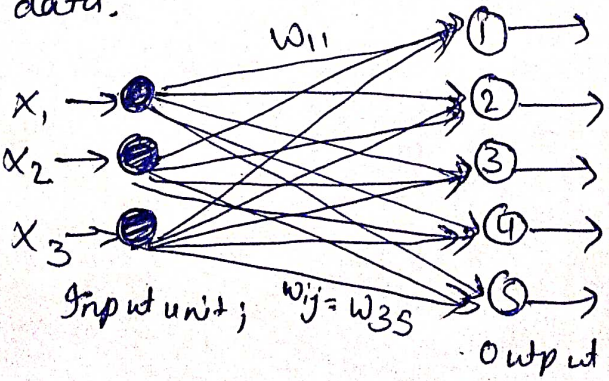
- → using unlabeled data
- → does not take any feedback
- → Computationally Complex
- → Hidden pattern in data
- → less Accurate
- → Clustering & Association problems.
- → finding customer segment in Marketing data [gender, age, location]

Competitive learning

→ form of unsupervised learning,

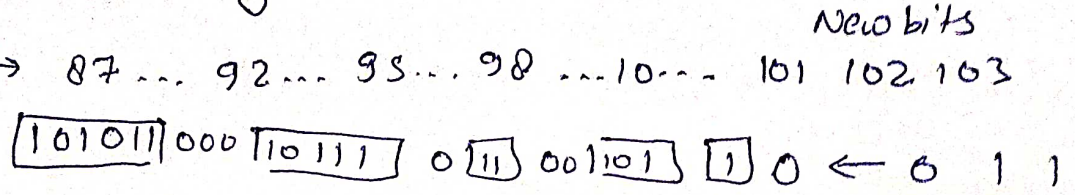
→ In which nodes compete for the right to respond to a subset of input data.

→



In the give data stream, let us assume the new bit arrives from the right

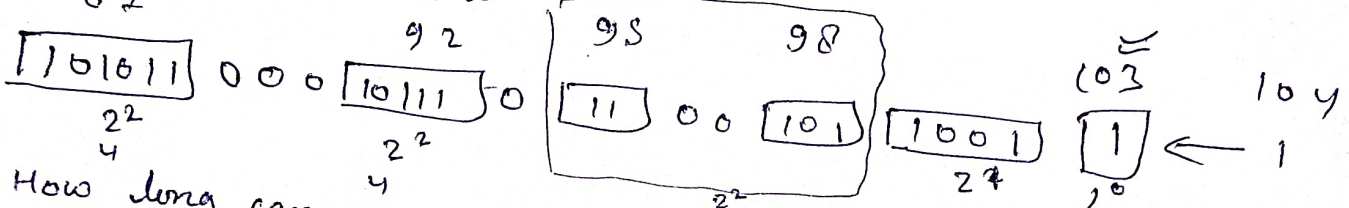
If new bit = 0 ⇒ No change in bucket,



when new bit = 0 enters
It does not effect bucket

when new bit = 1 enters
Combine

when next bit = 1 enters
87



How long can you continue doing this....

- Current time stamp − left most bucket timestamp < N (window size = 24 hours)

Eg :- 103 − 87 = 16 < 24 so I Continue

finally answer to query

How many no. of 1's in last 20 bits?

⇒ 20 bit

→ No. of 1s. in last 20 bits = 11

Real time Analytics Platform (RTAP)

→ Allows Business to get insights.
→ ex :- Continuously updated Customer activity like page views and shopping cart

- Components of RTAP :-
  - → Aggregator → Compiles real time streaming data
  - → Broker − Makes data in Real time available for use.
  - → Analytics Engine → Correlates values & blends data stream together during analysis
  - → Stream Processor → Executes Real time app analytics & logic by receiving & sending data stream.

→ Amazon uses Amazon kinesis

Real Time Analytic ~~Problem~~ Platform

→ Refers to finding meaningful patterns in data at the actual time of receiving
→ Analyse the data, correlate
→ Predicts the outcome in real time
→ Manages & Process data
→ evaluation Business intelligence
→ Apache Spark Streaming - a Big Data platform for data stream analytics in real time
→ Cisco Connected Streaming Analytics (CSA) - delivers insights from high-velocity streams of live data from multiple sources
→ Oracle Stream Analytics (OSA) - a platform provides a graphical interface to fast data
→ SAP HANA → streaming analytics tools
→ IBM stream Computing → a data streaming tool that analyze a broad range of streaming data - unstructured text, audio, video, sensor - to spot the risk & opportunities

Applications

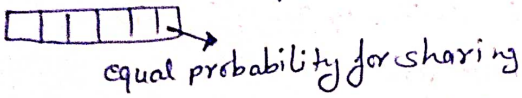
1) Frauds detection systems for online trasactions,
2) Log analysis for understanding usage pattern.
3) Click analysis for online Recommendation
4) Social media Analytics
5) Push notification to the customers for location based platforms

disadvantage
→ we don't know the value of n
→ we need to define window size (by storage capacity)

Sampling :-

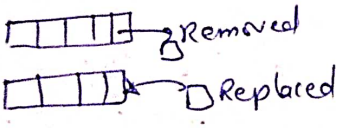
Unit	1	2	3	4	5	6
X	1	1	2	2	2	3

- Simple Random sample



- Simple Random sample Without Replacement

- S R S With Replacement



- Stratified Sampling

Counting distinct element (Flajolet martin Algo.) ⇒ Count the element in every windows of size k

a = [1, 2 ... 2, 1, 3, 1, 1, 3]
k = 4

Output

2
3
3
2
2

Total windows = $((n-k+1) \times k^2)$

Q Stream :- 1, 4, 2, 1, 2, 4, 4, 4, 1, 2, 4, 1, 7 = x
$h(a) = (3x+1) \bmod 8$

1. After $h(a)$ = 4, 3, 2, 4, 2, 3, 3, 3, 4, 2, 3, 4, 7
2. Convert it to binary = 100, 011, 010, 100, 010, 011, 011, 011, 100, 010, 011, 100, 010
3. Count no. of 0s (Trailing zero ⇒ end zero's)
   $r(a)$ = 2, 0, 1, 2, 1, 0, 0, 0, 2, 1, 0, 2, 1
4. $R = \max[r(a)] = 2$
5. $E = 2^R = 2^2 = 4$

There are 4 distinct elements in Stream
1, 2, 4, 7

Estimating moments (Mean Mode Median) ↓
measuring data uniformity

- moments for stream: 1, 3, 2, 1, 2, 3, 4, 3, 1, 2, 3, 1
- occurence = (1×4), 2(×3), 3(×4), 4(×1)
- 0th moment = $4^0 + 3^0 + 4^0 + 1^0 = 4$ (Distinct items)
- 1st moment = $4^1 + 3^1 + 4^1 + 1^1 = 12$ (Stream length n)
- 2nd moment = $4^2 + 3^2 + 4^2 + 1^2 = 42$ (Suprise Number)

Moments :-
Stream distinct elements (frequency, how they are distributed)
→ A is ordered set
→ m_i → no. of times value i occur.
→ k^{th} moment is

$$\sum_{i \in A} (m_i)^k$$

→ 0^{th} moment = no. of distinct element.
→ 1^{st} moment = length of the stream.
→ 2nd moment = Surprise no. S = a measure of how uneven the distribution is

ex:- Stream of length 100 1 3 4 5 ... 100
11 distinct values
find 2nd moment

→ Item Count : 10, 9, 9, 9, 9, 9, 9, 9, 9, 9
Surprise S = 910
$= 10 \times (1)^2 + 9 \times (10)^2$

→ Item count : 90, 1, 1, 1, ..., 1, 1, 1, 1, 1,
= 8,110

Ams method
→ Alon - matias - Szegedy
→ To calculate second moment.
→ estimated moments provided, gives an unbiased estimate
→ X.val X.el

Ex:- a, b, c, b, d, a, c, d, a, b, d, c, a, a, b
(3) (2) (13)

a = 5 b = 4 c = 3, d = 3

$(5)^2 + (4)^2 + (3)^2 + (3)^2 = 59$

A/c to Ams algorithm
n = 15 bits ~~Lets assume~~ no. of stream

$f(x) = n(2c - 1)$

we will calculate variables in different time interval.

$X_1 = \{c, 3\}$
$X_2 = \{d, 2\}$
$X_3 = \{a, 2\}$

Calculate estimate = $f(X_1) = n(2c - 1)$
$= 15(2 \times 3 - 1)$
$= 15 \times 5 = 75$

$f(X_2) = 15(4 - 1)$
$= 15 \times 3 = 45$

$f(X_3) = 15(3)$
$= 45$

final estimate = $\frac{75 + 45 + 45}{3} = \frac{165}{3} = 55$

Unit- 4

Frequent itemset:- Set of words that occur together in some minimum fraction of document in a cluster.

Mining frequent itemsets (Association Rule Mining)

→ Searches for frequent items in the data-set.

→ Frequent mining shows which item appear together in a transaction or relation.

Need of Association Mining:

→ ['milk'] ^ ['bread'] ⇒ ['butter']

Important Definitions:-

1. Support:- It is one of the measure of interestingness

Unit-4

Frequent Itemsets & Clustering.

⇒ Mining frequent itemsets.

- Association Mining →
- If there are two items X & Y purchased frequently then its good to put them together in stores or provide some discount. It increase Sale

  [milk] ^ [bread] => [butter]

- Support :- measure of interestingness. tells about usefulness & Certainty of rule.

  Support (A → B) = Support_count (A ∪ B)

- Confidence: A Confindence of 60%. means that 60% of Customer who purchased a milk & bread also bought butter.

  Confidence (A → B) = Support_count (A ∪ B) / Support_count (A)

- Maximal Itemset :- An itemset is maximal frequent if none of its supersets are frequent
- Closed Itemset :- if none of immediate superset have same support.
- k- itemset :- contains k- itemset

Ex :-

TransactionId	items
1	{A, C, D}
2	{B, C, D}
3	{A, B, C, D}
4	{B, D}
5	{A, B, C, D}

- Let say minimum support Count is 3
- hold maximal frequent ⇒ Closed ⇒ frequent

→ Market basket Model

→

Associate Rule Mining (ARM)

→ Market Basket Analysis
→ Affirmity Analysis
→ Set of items in a transaction is called Market Basked
→ Mostly used in Retail
→ if 'A' then 'B' {A ⇒ B}

- **Support (s):** % of transaction of both 'A' & 'B' — measure frequency
  ("A" → "B") = P(A ∩ B) } measures frequency of association
- **Confidence (c):** transaction set 'T' if 'c' is %. of times 'B' is present in all the transaction in — strength of association

$$C = P(B/A) = \frac{P(A \cap B)}{P(A)}$$

Parameters:-
(i) finding all items that appear frequently in transaction } min. support
(ii) finding strong association among frequent items } Count Confidence value.

Apriori Algorithm:-
generate candidate item set
(i) All Singleton item are Candidate
(ii) This algo is given by R. Agrawal & R. Srikant in 1994 for finding frequent itemset in a data set.

Eg:-

Tid	item
1	2, 3
2	1, 3, 5
3	1, 2, 4
4	2, 3

min Support = 2

item	Support
1	2
2	3
3	3
4	1
5	1

→ eliminated

itemsets	Support
{1, 2}	1
{1, 3}	1
{2, 3}	2

eliminated

Steps:-
1. Create a table Containing Support Count of each item.
2. Compare Candidate set item's Support Count
3. Generate Candidate set Using Condition
4. Check all subset

Apriori Algorithm (To generate association rule)

- Min Support = 50%.
- Threshhold Confidence = 70%.

eg:-

Tid	items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

I

Item set	Support
1	2/4 → 50%.
2	3/4 → 75%.
3	3/4 → 75%.
4	1/4 → 25%.
5	3/4 → 75%.

{ item set → 1, 2, 3, 5 }

II

Itemset	Support
{1,2}	1/4 → 25%
{1,3}	2/4 → 50%
{1,5}	1/4 → 25%
{2,3}	2/4 → 50%
{2,5}	3/4 → 75%
{3,5}	2/4 → 50%

III

Itemset	Support
{1,3,5}	1/4 = 25%
{2,3,5}	2/4 = 50%
{1,2,3}	1/4 = 25%

IV

{2,5,3} = This holds the Minimum support

Now generate Rules:

Rules	Support	Confidence
(2^3) → 5	2	2/2 = 100%
(3^5) → 2	2	2/2 = 100%
(2^5) → 3	2	2/3 = 66%
2 → (3^5)	2	2/3 = 66%
5 → (2^3)	2	2/3 = 66%
3 → (2^5)	2	2/3 = 66%

final Association rules are

(2^3) → 5

(3^5) → 2

Confidence = S(AUB)/S(A)

= 2^3 → 5 (A) (B) = S((2^3)U 5)/S(2^3)

= 2/2 = 100%.

→ Handling large data sets in main m/m.

- Allocate more memory
- Work with a Smaller Sample
- Use a Computer with more memory
- Change the data format
- Stream data or use progressive loading
- Use a Relational database
- Use big data platform

→ Limited Pass Algorithm :- finding frequent itemsets

- Algorithms uses to compute exact collection of frequent itemsets of size k in k passes
  - → A Priori
  - → PCY
  - → multistage
  - → multihash
- Algo. that find all frequent itemsets
  - → sampling
  - → SON
  - → Toivonen's Algo.

→ SON algorithm (Savasere, Omiecinski & Navathe)

- known as Partition algorithm
- MapReduce
  - → first map fnctn
  - → first Reduce functn
  - → second map
  - → second Reduce

Limited pass Algo. (PCY Algorithm) Frequent item set Counting

Park Chen Yun

Min support = 2

Tid	Items
1	1, 3, 4
2	2, 3, 5
3	1, 2, 3, 5
4	2, 5

PASS 1

Itemset	Support
1	2
2	3
3	5
4	1
5	3

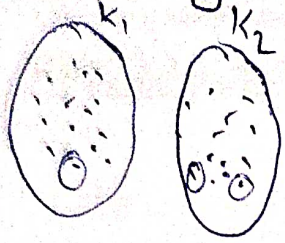
T_1	{1,3}$_2$ {1,4}$_1$ {3,4}$_1$;
T_2	{2,3}-2 {2,5}-3 {3,5}-2
T_3	{1,2} {1,3} {1,5} {2,3} {2,5} {3,5}
T_4	(2,5)

Pass 2 Apply hash fn,

Bucket -	1	2	3	4	5
Count -	4	6	3	0	0

- Construct Transaction Id table
- Counting frequent itemset in a stream
  → Count Sketch Algorithm.

Clustering Techniques.



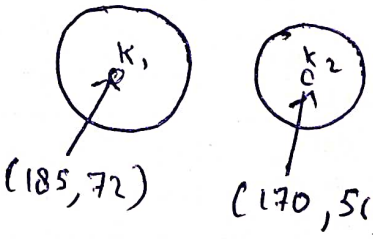
- Partition Based → k-means
- Hierarchical Based
- Density Based
- Grid Based → CLIQUE
- Model Based

k-means clustering:

	Height	Weight
①	185	72
②	170	56
③	168	60
④	179	68
⑤	182	72
⑥	188	77
⑦	180	71
⑧	180	70
⑨	183	84
⑩	180	88
⑪	180	67
⑫	177	76

~~Euclinear distance~~

Euclidean Distance $= \sqrt{(x_0 - x_c)^2 + (y_0 - y_c)^2}$

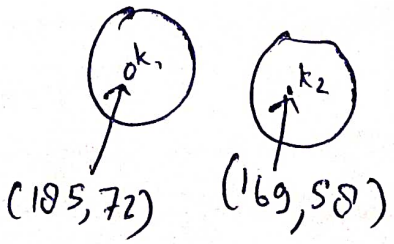


ED for ③ → $k_1 \rightarrow \sqrt{(168-185)^2 + (60-72)^2} = 20.80$

$k_2 \rightarrow \sqrt{(168-170)^2 + (60-56)^2} = 4.48$

New centeroid Calculation

for $k_2 = \left(\frac{170+168}{2}, \frac{60+56}{2}\right) = (169, 58)$



ED for ④ → $k_1 \Rightarrow \sqrt{(179-185)^2 + (68-72)^2} = 6.32$

$k_2 = \sqrt{(179-169)^2 + (68-58)^2} = 14.14$

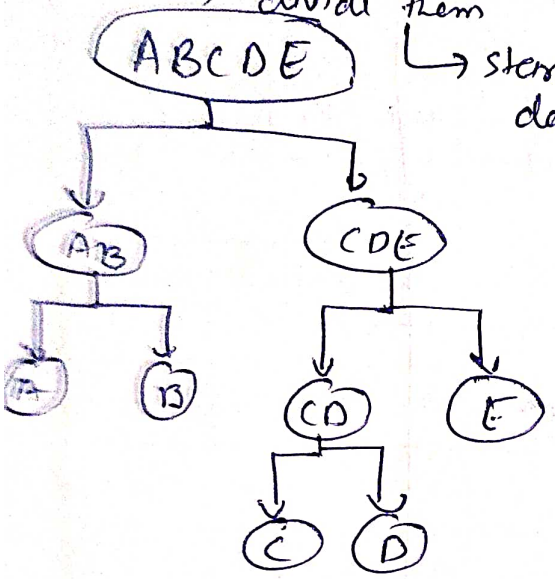
$k_1 \rightarrow \{1, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$

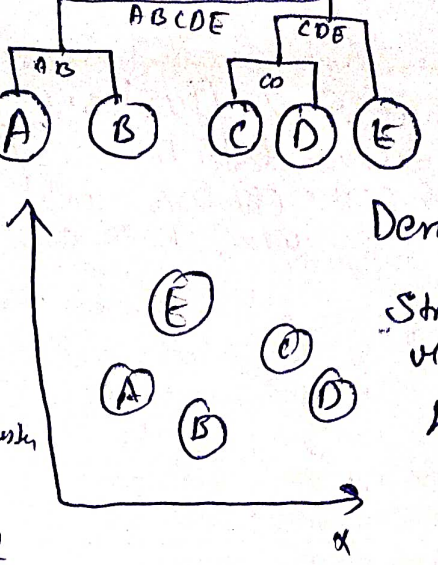
$k_2 \rightarrow \{2, 3\}$

Hierarchical Clustering

→ Agglomerative
→ Divisive
  → Start with individual data item
  → bottom to up

→ top to down
→ Put all the data points in one cluster
  divide them
  → starts with whole data points





Dendogram :-
Structure which represent the hierachical Clustering

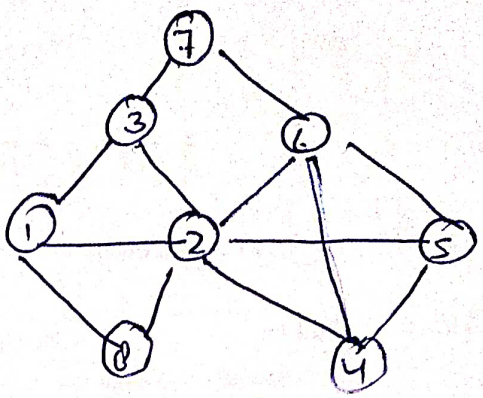
→ Clustering high-dimensional data.
  → Subspace clustering :- search for clusters already existing
    • Such as :- CLIQUE, ProClus
  → Dimensionality Reduction approaches :-
    Construct much lower dimensional space
    • Spectral Clustering
  → Clustering should not only consider dimension

CLIQUE Cluster

→ Grid based cluster & DensityBased
→ estimate the density by points

Clique & Percolation Method

Use this method to find the communities for k=3



Step 1:- Divide the graph into small clusters (cliques)

Step 2:- Write down all the cliques and group them into different categories

Step 3:- 6 cliques found here

a = (1, 2, 3)
b = (2, 8)
c = (2, 4, 6)
d = (2, 4, 5)
e = (4, 6, 5)
f = (2, 6, 5)

Clique

k = no. of nodes

N=2

Step 4:- Community 1 = (1, 2, 3, 8)

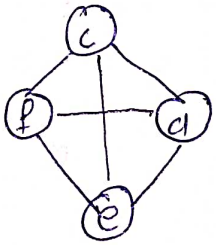
C2 = (2, 4, 5, 6)

To find community [k-1 node should be common]

Community is group of Clique.

for k = 3

Step 5:- Now create New Graph



$C_1 = \{a, b\}$

$C_2 = \{c, d, e, f\}$

Step 6:- formulae is $k-1$

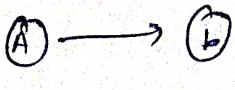
$3-1=2$

Step 7:- find the community

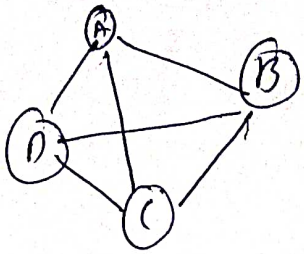
$C = \{C_1, C_2\}$

Diagram for Communities

(i) Community = {a,b}



(ii) Community = {a,b,c,d}



~~Quiz~~ ~~P to CLUS~~

Frequent Pattern Based Clustering Methods.

To remove drawback of apriori algorithm.

- It is expensive operation for calculation

Ex:-

Tid	item	min_sup=2
T100	I1, I2, I5	I2, I1
T200	I2, I4	I2, I4
T300	I2, I3	I2, I3
T400	I1, I2, I4	I2, I1, I4
T500	I1, I3	I1, I3

Step1

item	Sup.
I1	3
I2	4
I3	2
I4	2
I5	1

~~descending order~~

Check min Support & write in decreasing order

	item	Sup
T_1	I5	1
T_2	I4	2
T_3	I3	2
T_4	I1	3
T_5	I2	4

Step2

Null

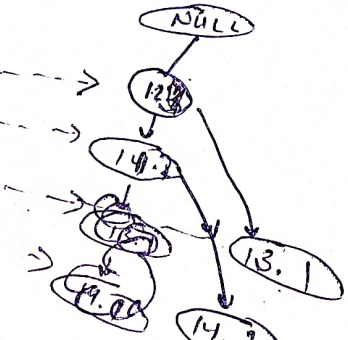
I

I2	4
I1	3
I3	2
I4	2

↑ frequent pattern

Step3 Construct FP Tree :- ordered item set

ITEM ID	Sup	N.L
I2	4	
I1	3	
I3	2	
I4	2	



Step4 Conditional Pattern based

Steps:- finding frequent patterns

I4	{I2,I1,I4} {I2,I4}
I3	{I1,I3} {I2,I3}
I1	{I1,I3} {I2,I1,I4} {I2,I4}
I2	{

Frequent Pattern Growth Algorithm

TID	item	L1
T100	I1 I2 I5	I2 I1 I5
T200	I2 I4	I2 I4
T300	I2 I3	I2 I3
T400	I1 I2 I4	I2 I1 I4
T500	I1 I3	
T600	I2 I3	
T700	I1 I3	
T800	I1 I2 I3 I5	I2 I1 I3 I5
T900	I1 I2 I3	I2 I1 I3

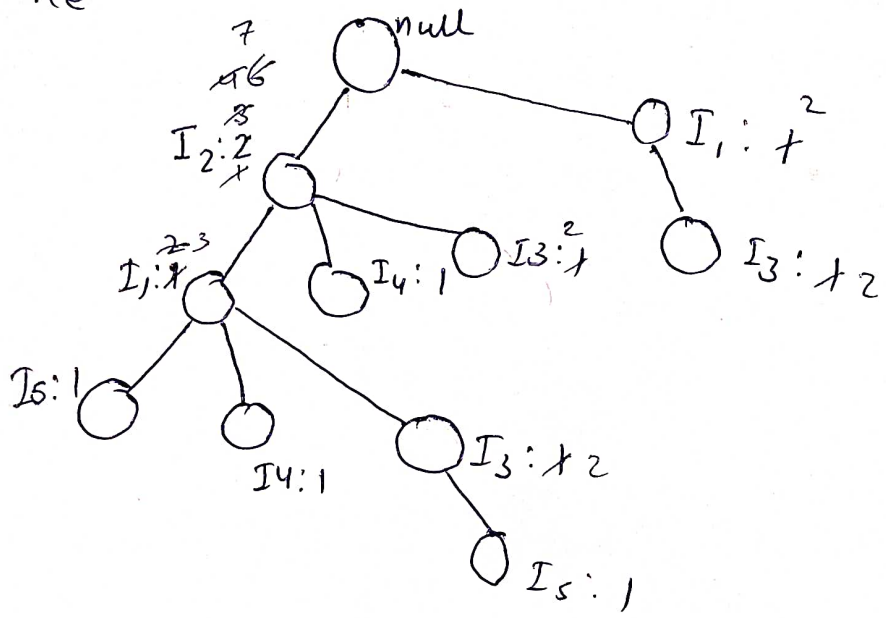
L1

Item set	Sup_count
I1	6
I2	7
I3	6
I4	2
I5	2

⇒

Items	Sup_count
I2	7
I1	6
I3	6
I4	2
I5	2

FP Tree



min_sup= 2

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I5	{I2, I1:1} {I2, I1, I3:1}	{I2:2, I1:2}	{I2, I5:2} {I1, I5:2} {I2, I1, I5:2}
I4	{I2, I1:1} {I2:1}	(I2:2)	{I2, I4:2}
I3	{I2, I1:2} {I2:2}, {I1:2}	(I2:4, I1:2), (I1:2)	{I2, I3:4} {I1, I3:4} {I2, I1, I3:2}
I1	{I2:4}	(I2:4)	{I2, I1:4}

Fuzzy decision Tree

ID3 Algorithm

Age	Competition	Type	Profit
old	Yes	S/W	Down
old	No	S/W	Down
old	No	H/W	Down
mid	Yes	S/W	Down
mid	Yes	H/W	Down
mid	Yes	H/W	Up
mid	No	S/W	Up
new	No	S/W	Up
new	Yes	S/W	Up
new	No	H/W	Up
new	No	S/W	Up

Age :

	Down	Up
old	3	0
mid	2	2
new	0	3

IG

$$\text{Info. gain} = \frac{-P}{P+N}\log_2\left(\frac{P}{P+N}\right) - \frac{N}{P+N}\log_2\left(\frac{N}{P+N}\right)$$

$$\text{Entropy}(A) = \sum_{i=1}^{v} \frac{P_i+N_i}{P+N} I(P_iN_i)$$

(Attribute)

$$\text{Gain} = IG - E(A)$$

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

Target attribute

$$I(\text{old}) = -\left[\frac{3}{3}\log_2\left(\frac{3}{3}\right) + \frac{0}{3}\log_2\left(\frac{0}{3}\right)\right]$$

$$= 0 \times \frac{3}{10} = 0$$

$$I(\text{mid}) = -\left[\frac{2}{4}\log_2\left(\frac{2}{4}\right) + \frac{2}{4}\log_2\left(\frac{2}{4}\right)\right] = 0.$$

$$I(\text{new}) = 0$$

$$E(\text{Age}) = 0 + 0.4 + 0 = 0.4$$

For Profit

$$IG = -\left[\frac{5}{10}\log_2\left(\frac{5}{10}\right) + \frac{5}{10}\log_2\left(\frac{5}{10}\right)\right]$$

$$= -[0.5 \times \log_2 2^{-} + 0.5\log_2 2^{-1}]$$

$$= -[0.5 \times (-1\log_2 2) + 0.5 \times (-1\log_2 2)]$$

$$= -[-0.5 - 0.5] = -[-1]$$

$$I.G. = 1$$

$$\text{Gain} = 1 - 0.4$$

$$= 0.6$$

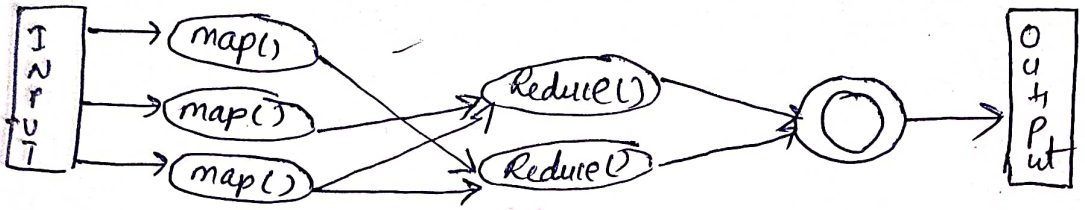
~~Clustering in non euclidian Space~~

Unit-5

MapReduce

→ Word Count program :- In Hadoop architechture
→ Key value Pair → Parallely Process
→

Map Reduce Component



→ key/value → Combines key values and make set of tuples

(1) one master Job tracker
  (1) managing Resources
  (2) Resource management
  (3) Scheduling task
  (4) Monitoring task

(2) many Slave Job tracker
  ① Executes the task
  ② Provide task status