Topic covered through ICT : Typical OLAP Operations **Date:21/2/2022**
Name of the Course Instructor: Michael Sadgun Rao Kona **Unit**: I
Course Name & Code : DWDM & 20CS10 **Academic Year**: 2021-22
Program/Sem./Section : B.Tech / IV Sem. / Section – A & B

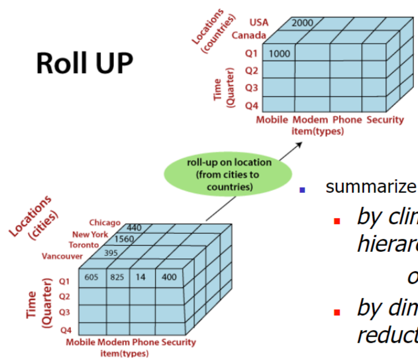
Typical OLAP Operations

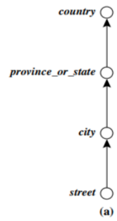
- Roll up (drill-up):
- Drill down (roll down):
- Slice and dice: *project and select*
- Pivot (rotate):
  - *reorient the cube, visualization, 3D to series of 2D planes*
- Other operations
  - *drill across: involving (across) more than one fact table*
  - *drill through:* Allows users to analyze the same data through different reports, analyze it with different features and even display it through different visualization methods

53

Typical OLAP Operations: Roll Up/Drill Up

Roll UP





- summarize data
  - *by climbing up hierarchy*

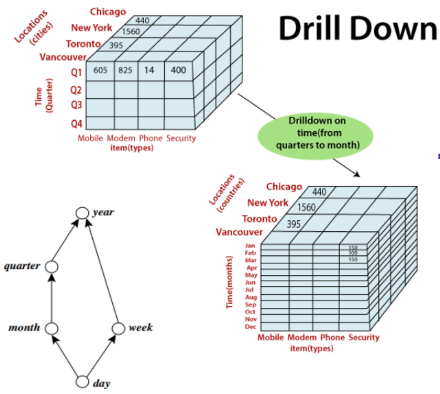
    or
  - *by dimension reduction*

Source & Courtesy: https://www.javatpoint.com/olap-operations

55

Typical OLAP Operations: Roll Down

Drill Down



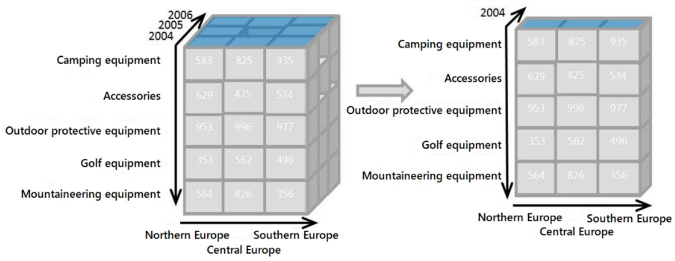
- reverse of roll-up
  - *from higher level summary to lower level summary or detailed data, or introducing new dimensions*

Source & Courtesy: https://www.javatpoint.com/olap-operations

56

Typical OLAP Operations: Slicing

- ***Slice*** is the act of picking a rectangular subset of a cube by choosing a single value for **one of its dimensions**, creating a new cube with one fewer dimension.
- Example: The sales figures of all sales regions and all product categories of the company in the year 2005 and 2006 are "sliced" out of the data cube.

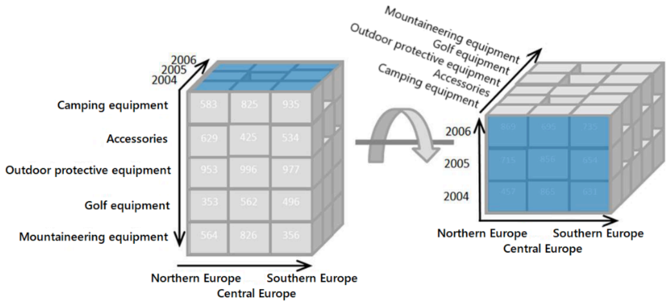


Source & Courtesy: https://en.wikipedia.org/wiki/OLAP_cube

57

Typical OLAP Operations:Pivot

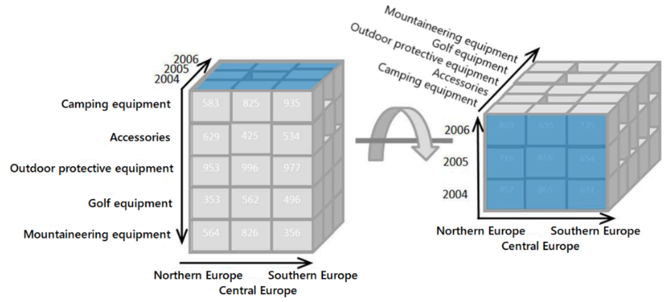
Pivot allows an analyst to **rotate the cube** in space to see its various faces. For example, cities could be arranged vertically and products horizontally while viewing data for a particular quarter.



Source & Courtesy: https://en.wikipedia.org/wiki/OLAP_cube

Typical OLAP Operations:Pivot

Pivot allows an analyst to **rotate the cube** in space to see its various faces. For example, cities could be arranged vertically and products horizontally while viewing data for a particular quarter.



Source & Courtesy: https://en.wikipedia.org/wiki/OLAP_cube

61

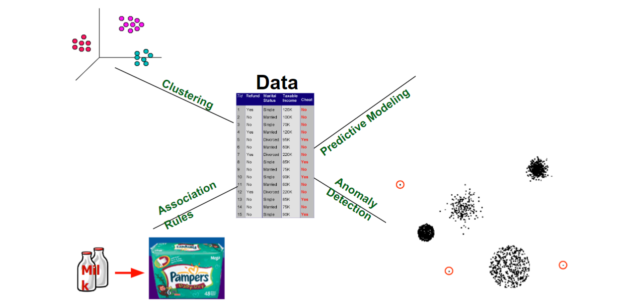
Course Instructor	**Module Co-ordinator**	**HOD**
Mr.Michael Sadgun Rao.K	Dr.K.Lavanya	Dr.B.Srinivasa Rao

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING
(AUTONOMOUS)
Accredited by NAAC & NBA (CSE, IT, ECE, EEE & ME)
Approved by AICTE, New Delhi and Affiliated to JNTUK, Kakinada
L.B.Reddy Nagar, Mylavaram-521230, Krishna Dist, Andhra Pradesh, India
DEPARTMENT OF INFORMATION TECHNOLOGY

Topic covered through ICT : Datamining Tasks **Date:4/4/2022**
Name of the Course Instructor: Michael Sadgun Rao Kona **Unit**: II
Course Name & Code : DWDM & 20CS10 **Academic Year**: 2021-22
Program/Sem./Section : B.Tech / IV Sem. / Section – A & B

Data Mining Tasks

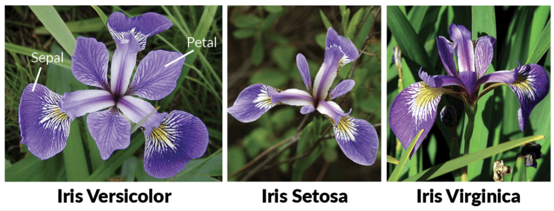
Press Esc to exit full screen



Introduction to Data Mining, 2nd Edition Tan, Steinbach, Karpatne, Kumar

Data Mining Tasks

- Example: (**Predicting the Type of a Flower**):



Data Mining Tasks

Example:
(**Predicting the Type of a Flower**)

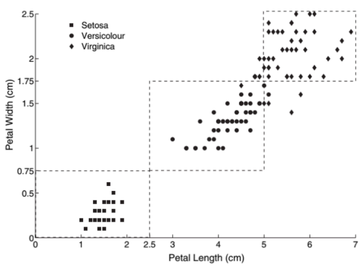


Figure 1.4. Petal width versus petal length for 150 Iris flowers.

Data Mining Tasks

- **Association analysis**
- **Example (Market Basket Analysis).**
  - **AIM:** *find items that are frequently bought together by customers.*
  - *Association rule {Diapers} −→ {Milk},*
    - suggests that customers who buy diapers also tend to buy milk.
- This rule can be used to identify potential cross-selling opportunities among related items.

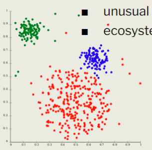
Table 1.1. Market basket data.

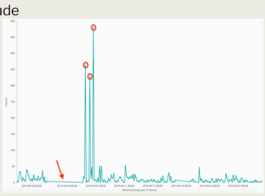
Transaction ID	Items
1	{Bread, Butter, Diapers, Milk}
2	{Coffee, Sugar, Cookies, Salmon}
3	{Bread, Butter, Coffee, Diapers, Milk, Eggs}
4	{Bread, Butter, Salmon, Chicken}
5	{Eggs, Bread, Butter}
6	{Salmon, Diapers, Milk}
7	{Bread, Tea, Sugar, Eggs}
8	{Coffee, Sugar, Chicken, Eggs}
9	{Bread, Diapers, Milk, Salt}
10	{Tea, Eggs, Cookies, Diapers, Milk}

The transactions data collected at the checkout counters of a grocery store.

Data Mining Tasks

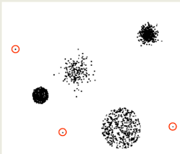
- **Anomaly Detection:**
  - Task of identifying observations whose characteristics are significantly different from the rest of the data.
  - Such observations are known as **anomalies or outliers**.
  - A good anomaly detector must have a high detection rate and a low false alarm rate.
  - Applications of anomaly detection include
    - the detection of fraud,
    - network intrusions,
    - unusual patterns of disease, and
    - ecosystem disturbances





https://commons.wikimedia.org/wiki/File:Anomalous_Web_Traffi c.png

Data Mining Tasks



- **Anomaly Detection:**
  - Example 1.4 (Credit Card Fraud Detection).
  - A credit card company records the transactions made by every credit card holder, along with personal information such as credit limit, age, annual income, and address.
  - Since the number of fraudulent cases is relatively small compared to the number of legitimate transactions, anomaly detection techniques can be applied to **build a profile of legitimate transactions for the users.**
  - When a new transaction arrives, it is compared against the profile of the user. If the characteristics of the transaction are very different from the previously created profile, then the transaction is flagged as potentially fraudulent.

Data Mining Tasks

- **Cluster analysis**
  - Example 1.3 (Document Clustering)
  - Each article is represented as a set of word-frequency pairs (w, c),
    - where w is a word and
    - c is the number of times the word appears in the article.
  - There are two natural clusters in the data set.
  - First cluster -> first four articles (news about the economy)
  - Second cluster-> last four articles (news about health care)
  - A good clustering algorithm should be able to identify these two clusters based on the similarity between words that appear in the articles.

Table 1.2. Collection of news articles.

Article	Words
1	dollar: 1, industry: 4, country: 2, loan: 3, deal: 2, government: 2
2	machinery: 2, labor: 3, market: 4, industry: 2, work: 3, country: 1
3	job: 5, inflation: 3, rise: 2, jobless: 2, market: 3, country: 2, index: 3
4	domestic: 3, forecast: 2, gain: 1, market: 2, sale: 3, price: 2
5	patient: 4, symptom: 2, drug: 3, health: 2, clinic: 2, doctor: 2
6	pharmaceutical: 2, company: 3, drug: 2, vaccine: 1, flu: 3
7	death: 2, cancer: 4, drug: 3, public: 4, health: 3, director: 2
8	medical: 2, cost: 3, increase: 2, patient: 2, health: 3, care: 1

Data Mining Tasks

- Predictive modeling refers to the task of **building a model** for the target variable as a function of the explanatory variables.
- 2 types of predictive modeling tasks:
  - **Classification:** Used for discrete target variables
  - **Regression**: used for continuous target variables.

	Continuous	Discrete		
	Quantitative data	Qualitative / Categorical / Attribute data		
Measurement	Units (example)	Ordinal (example)	Nominal (example)	Binary (example)
Time of day	Hours, minutes, seconds	1, 2, 3, etc.	N/A	a.m./p.m.
Date	Month, date, year	Jan., Feb., Mar., etc.	N/A	Before / After
Cycle time	Hours, minutes, seconds, month, date, year	10, 20, 30, etc.	N/A	Before / After
Speed	Miles per hour/centimeters per second	10, 20, 30, etc.	N/A	Fast / Slow
Brightness	Lumens	Light, medium, dark	N/A	On / Off
Temperature	Degrees C or F	10, 20, 30, etc.	N/A	Hot / Cold
<Count data>	Number of things	10, 20, 30, etc.	N/A	Large / Small
Test scores	Percent, number correct	F, D, C, B, A	N/A	Pass / Fail
Defects	N/A	Number of cracks	N/A	Good / Bad
Defects	N/A	N/A	Oversized, missing	Good / Bad
Color	N/A	N/A	Red, blue, green	N/A
Location	N/A	N/A	East, West, South	Domestic / International
Groups	N/A	N/A	HR, Legal, IT	Exempt / Non-exempt
Anything	Percent	10, 20, 30, etc.	N/A	Above / Below

Course Instructor	Module Co-ordinator	HOD
Mr.Michael Sadgun Rao.K	Dr.K.Lavanya	Dr.B.Srinivasa Rao

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING

(AUTONOMOUS)

Accredited by NAAC & NBA (CSE, IT, ECE, EEE & ME)

Approved by AICTE, New Delhi and Affiliated to JNTUK, Kakinada

L.B.Reddy Nagar, Mylavaram-521230, Krishna Dist, Andhra Pradesh, India

DEPARTMENT OF INFORMATION TECHNOLOGY

Topic covered through ICT : Decision Tree Induction **Date:22/4/2022**

Name of the Course Instructor: Michael Sadgun Rao Kona **Unit**: III

Course Name & Code : DWDM & 20CS10 **Academic Year**: 2021-22

Program/Sem./Section : B.Tech / IV Sem. / Section – A & B

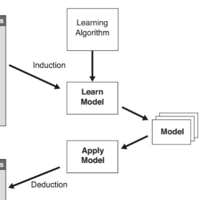
Press Esc to exit full screen

General approach to solving a classification problem

- Classification technique (or classifier)
  - Systematic approach to building classification models from an input data set.
  - Examples
    - Decision tree classifiers,
    - Rule-based classifiers,
    - Neural networks,
    - Support vector machines, and
    - Naive bayes classifiers.
- Learning algorithm
  - Used by the classifier
  - To identify a model
    - That best fits the relationship between the attribute set and class label of the input data.

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes



Test Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Figure 4.3. General approach for building a classification model.

DECISION TREE INDUCTION

Working of Decision Tree

- Three types of nodes:
  - **Root node**
    - No incoming edges
    - Zero or more outgoing edges.
  - **Internal nodes**
    - Exactly one incoming edge and
    - Two or more outgoing edges.
  - **Leaf or terminal nodes**
    - Exactly one incoming edge and
    - No outgoing edges.
- Each leaf node is assigned a class label.
- **Non-terminal nodes** (root & other internal nodes) contain attribute test conditions to separate records that have different characteristics.

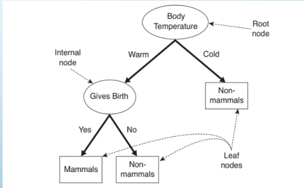


Figure 4.4. A decision tree for the mammal classification problem.

DECISION TREE INDUCTION

Working of Decision Tree

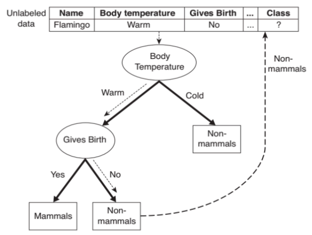


Figure 4.5. Classifying an unlabeled vertebrate. The dashed lines represent the outcomes of applying various attribute test conditions on the unlabeled vertebrate. The vertebrate is eventually assigned to the `Non-mammal` class.

DECISION TREE INDUCTION

Buiding Decision Tree

- **Example**:-predicting whether a loan applicant will repay or not (defaulted)
  - Construct a training set by examining the records of previous borrowers.

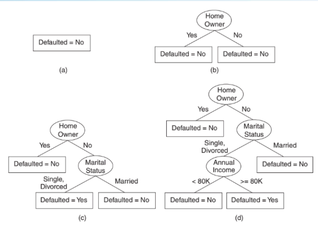


Figure 4.7. Hunt's algorithm for inducing decision trees.

	binary	categorical	continuous	class
Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training set for predicting borrowers who will default on loan payments.

DECISION TREE INDUCTION

Measures for Selecting the Best Split

- selection of best split is based on the degree of impurity of the child nodes
- Node with class distribution (0, 1) has **zero impurity**,
- Node with uniform class distribution (0.5, 0.5) has the **highest impurity**.
- p - fraction of records that belong to one of the two classes.
- P – maximum(0.5) – class distribution is even
- P- min. (0 or 1)– all records belong to the same class

$$\begin{aligned} \text{Entropy}(t) &= -\sum_{i=0}^{c-1} p(i|t)\log_2 p(i|t), \\ \text{Gini}(t) &= 1-\sum_{i=0}^{c-1}[p(i|t)]^2, \\ \text{Classification error}(t) &= 1-\max_i[p(i|t)], \end{aligned}$$

where c is the number of classes and $0\log_2 0 = 0$ in entropy calculations

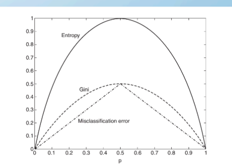
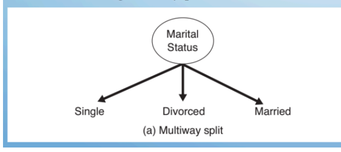


Figure 4.13. Comparison among the impurity measures for binary classification problems.

DECISION TREE INDUCTION

Methods for Expressing Attribute Test Conditions

- **Test condition for Nominal Attributes**
  - nominal attribute can have many values
  - Test condition can be expressed in two ways
    - Multiway split - number of outcomes depends on the number of distinct values
    - Binary splits(used in CART) - produces binary splits by considering all $2^{k-1} - 1$ ways of creating a binary partition of k attribute values.



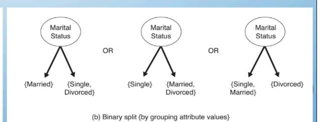


Figure 4.9. Test conditions for nominal attributes.

DECISION TREE INDUCTION

Methods for Expressing Attribute Test Conditions

- **Test condition for Ordinal Attributes**
  - Ordinal attributes can also produce binary or multiway splits.
  - values can be grouped without violating the order property.
  - 4.10© is invalid

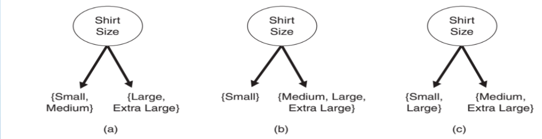


Figure 4.10. Different ways of grouping ordinal attribute values.

DECISION TREE INDUCTION

Methods for Expressing Attribute Test Conditions

- **Test condition for Continuous Attributes**
  - Test condition - Comparison test ($A < v$) or ($A \geq v$) with **binary** outcomes,

    or
  - Test condition - a range query with outcomes of the form $v_i \leq A < v_{i+1}$, for $i = 1,\ldots, k$.
    - Multiway split
    - Apply the discretization strategies

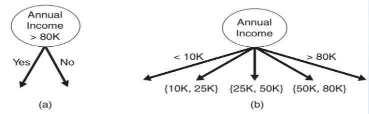


Figure 4.11. Test condition for continuous attributes.

Course Instructor	**Module Co-ordinator**	**HOD**
Mr.Michael Sadgun Rao.K	Dr.K.Lavanya	Dr.B.Srinivasa Rao

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING
(AUTONOMOUS)
Accredited by NAAC & NBA (CSE, IT, ECE, EEE & ME)
Approved by AICTE, New Delhi and Affiliated to JNTUK, Kakinada
L.B.Reddy Nagar, Mylavaram-521230, Krishna Dist, Andhra Pradesh, India
DEPARTMENT OF INFORMATION TECHNOLOGY

Topic covered through ICT : Apriori Algorithm **Date:20/5/2022**
Name of the Course Instructor: Michael Sadgun Rao Kona **Unit**: IV
Course Name & Code : DWDM & 20CS10 **Academic Year**: 2021-22
Program/Sem./Section : B.Tech / IV Sem. / Section – A & B

Frequent Itemset Generation

- **Lattice structure** - list of all possible itemsets
- itemset lattice for
  - **I = {a, b, c, d, e}**
- Data set with k items can generate up to **2^k − 1** frequent itemsets (without null set)
  - Example:- 2^5−1=32
- So, search space of itemsets in practical applications is exponentially large

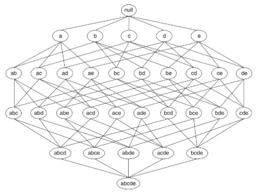


Figure 6.1. An itemset lattice.

Frequent Itemset Generation

Transactions

TID	Items
1	Bread, Milk
2	Bread, Diapers, Beer, Eggs
3	Milk, Diapers, Beer, Coke
4	Bread, Milk, Diapers, Beer
5	Bread, Milk, Diapers, Coke

N Candidates M

Figure 6.2. Counting the support of candidate itemsets.

- A **brute-force approach** for finding frequent itemsets
  - determine the support count for every candidate itemset in the lattice structure.
- compare each candidate against every transaction
- Very expensive
  - requires **O(NMw)** comparisons,
  - N- No. of transactions,
  - M = 2^k − 1 is the number of candidate itemsets
  - w - maximum transaction width.

Frequent Itemset Generation

The Apriori Principle

If an itemset is frequent, then all of its subsets must also be frequent.

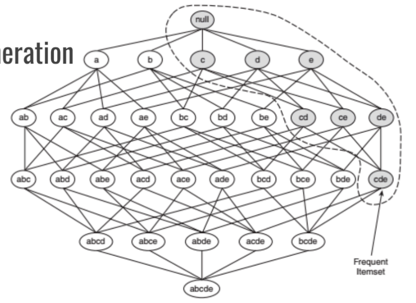


Figure 6.3. An illustration of the *Apriori* principle. If {c, d, e} is frequent, then all subsets of this itemset are frequent.

Frequent Itemset Generation

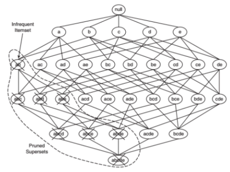


Figure 6.4. An illustration of support-based pruning. If {a, b} is infrequent, then all supersets of {a, b} are infrequent.

Support-based pruning:

- strategy of **trimming** the exponential **search space** based on the support measure is known as support-based pruning.
- It uses anti-monotone property of the support measure.
- Anti-monotone property of the support measure
  - support for an itemset never exceeds the support for its subsets.
- Example:
  - {a, b} is infrequent,
  - then all of its supersets must be infrequent too.
  - entire subgraph containing the supersets of {a, b} can be pruned immediately

Frequent Itemset Generation in the Apriori Algorithm

Candidate 1-Itemsets

Item	Count
Beer	3
Bread	4
Cola	2
Diapers	4
Milk	4
Eggs	1

Minimum support count = 3

Candidate 2-Itemsets

Itemset	Count
{Beer, Bread}	2
{Beer, Diapers}	3
{Beer, Milk}	2
{Bread, Diapers}	3
{Bread, Milk}	3
{Diapers, Milk}	3

Itemsets removed because of low support

Candidate 3-Itemsets

Itemset	Count
{Bread, Diapers, Milk}	3

brute-force strategy

$\binom{6}{1}+\binom{6}{2}+\binom{6}{3}=6+15+20=41$

With the *Apriori* principle, this number decreases to

$\binom{6}{1}+\binom{4}{2}+1=6+6+1=13$

ure 6.5. Illustration of frequent itemset generation using the *Apriori* algorithm.

Min. support 50% (i.e., 2tx's) BE=>C conf.:66%

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan → C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan → C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3rd scan → L_3

Itemset	sup
{B, C, E}	2

https://towardsdatascience.com/apriori-association-rule-mining-explanation-and-python-implementation-290b42afdfc6

Frequent Itemset Generation in the Apriori Algorithm

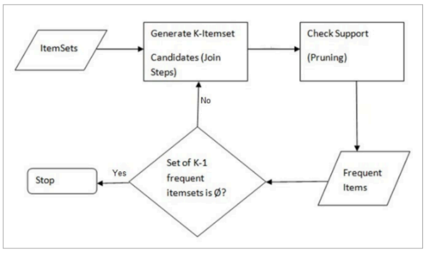
C_k-set of k-candidate itemsets

F_k - set of k-frequent itemsets

Algorithm 6.1 Frequent itemset generation of the *Apriori* algorithm.

$k = 1$.
$F_k = \{ i \mid i \in I \wedge \sigma(\{i\}) \geq N \times minsup \}$. {Find all frequent 1-itemsets}
repeat
  $k = k + 1$.
  $C_k = \text{apriori-gen}(F_{k-1})$. {Generate candidate itemsets}
  for each transaction $t \in T$ **do**
    $C_t = \text{subset}(C_k, t)$. {Identify all candidates that belong to t}
    for each candidate itemset $c \in C_t$ **do**
      $\sigma(c) = \sigma(c) + 1$. {Increment support count}
    end for
  end for
  $F_k = \{ c \mid c \in C_k \wedge \sigma(c) \geq N \times minsup \}$. {Extract the frequent k-itemsets}
until $F_k = \emptyset$
Result $= \bigcup F_k$.

Frequent Itemset Generation in the Apriori Algorithm



https://www.softwaretestinghelp.com/apriori-algorithm/#:~:text=Apriori%20algorithm%20is%20a%20sequence,is%20assumed%20by%20the%20user.

Example

Transaction	List of items
T1	A,B,C
T2	B,C,D
T3	D,E
T4	A,B,D
T5	A,B,C,E
T6	A,B,C,D

Support threshold=50% => 0.5*6=3 => min_sup=3

Item	Count
A	4
B	5
C	4
D	4
E	2

After Pruning

Item	Count
A	4
B	5
C	4
D	4

Item	Count
A,B	4
A,C	3
A,D	2
B,C	4
B,D	3
C,D	2

after pruning

Item	Count
A,B	4
A,C	3
B,C	4
B,D	3

Item	Count
A,B,C	3
A,B,D	2
A,C,D	1
B,C,D	2

after pruning

Item	Count
A,B,C	3

{A, B} => {C}

Confidence = support {A, B, C} / support {A, B} = (3/ 4)* 100 = 75%

{A, C} => {B}

Confidence = support {A, B, C} / support {A, C} = (3/ 3)* 100 = 100%

{B, C} => {A}

Confidence = support {A, B, C} / support {B, C} = (3/ 4)* 100 = 75%

{A} => {B, C}

Confidence = support {A, B, C} / support {A} = (3/ 4)* 100 = 75%

{B} => {A, C}

Confidence = support {A, B, C} / support {B = (3/ 5)* 100 = 60%

{C} => {A, B}

Confidence = support {A, B, C} / support {C} = (3/ 4)* 100 = 75%

This shows that all the above association rules are strong if minimum confidence threshold is 60%.

< 26 >

Course Instructor	Module Co-ordinator	HOD
Mr.Michael Sadgun Rao.K	Dr.K.Lavanya	Dr.B.Srinivasa Rao

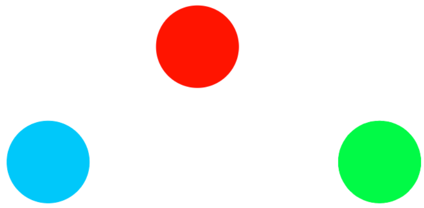
Topic covered through ICT : Types of Clusters **Date:6/6/2022**
Name of the Course Instructor: Michael Sadgun Rao Kona **Unit**: V
Course Name & Code : DWDM & 20CS10 **Academic Year**: 2021-22
Program/Sem./Section : B.Tech / IV Sem. / Section – A & B

Types of Clusters

- Well-separated clusters
- Prototype-based clusters
- Contiguity-based clusters
- Density-based clusters
- Described by an Objective Function

Types of Clusters: Well-Separated

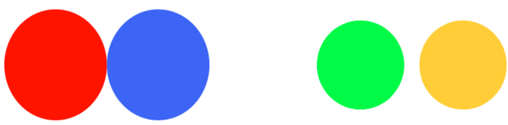
- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

Types of Clusters: Prototype-Based

- Prototype-based (or center based)
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the prototype or "center" of a cluster, than to the center of any other cluster
  - Data - Continuous - Centroid/mean
  - Data - Categorical - Medoid (Most Representative point)



4 center-based clusters

Types of Clusters: Contiguity-Based (Graph)

- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.
  - Graph (Data-Nodes, links - Connections),Cluster is group of connected objects.No connections with outside group.

8 contiguous clusters

- Useful when clusters are irregular or intertwined
- Trouble when noise is present
  - a small bridge of points can merge two distinct clusters.

Types of Clusters: Density-Based

- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Used when the clusters are irregular or intertwined, and when noise and outliers are present.

The two circular clusters are not merged, as in Figure, because the bridge between them(previous slide figure) fades into the noise.

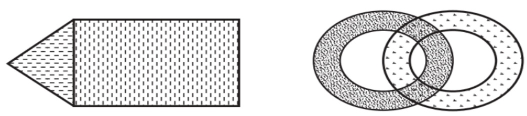
6 density-based clusters

Curve that is present in previous slide Figure also fades into the noise and does not form a cluster

A density based definition of a cluster is often employed when the clusters are irregular or intertwined, and when noise and outliers are present.

Types of Clusters: Density-Based

- Shared property(Conceptual Clusters)
  - a cluster as a set of objects that share some



(e) Conceptual clusters. Points in a cluster share some general property that derives from the entire set of points. (Points in the intersection of the circles belong to both.)

A clustering algorithm would need a very specific concept (sophisticated) of a cluster to successfully detect these clusters. The process of finding such clusters is called conceptual clustering.

Clustering Algorithms

- K-means and its variants
- Hierarchical clustering
- Density-based clustering

K-means Clustering

- Partitional clustering approach
- Number of clusters, K, must be specified
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- The basic algorithm is very simple

```
Select K points as the initial centroids.
repeat
    Form K clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

Course Instructor	**Module Co-ordinator**	**HOD**
Mr.Michael Sadgun Rao.K	Dr.K.Lavanya	Dr.B.Srinivasa Rao