DATA PREPROCESSING

UNIT- 2

1.7 Need for preprocessing the Data :-

Data in the real World is incomplete, noisy & inconsistent

Incomplete — lacking attribute values or certain attributes of interest or containing only aggregate data

Noisy — containing errors or outlier values

Inconsistent — containing discrepancies

All these things occur in large databases & Data-Warehouses.

Reasons for Incomplete data :-

- Attributes of interest may not be available
- Data not considered as unimportant at time of entry
- Relevant data is not recorded due to misunderstanding
- Data may be deleted.
- Modifications of data may be overlooked.

Reasons for Noisy Data :-

- Data collection instruments may be faulty
- Errors in data Transmission
- Technology limitations — limited buffer size

* Incorrect data may result from inconsistencies in naming conventions of data code used or Inconsistent format for i/p fields

* Data Cleaning — Cleans data by filling in missing values, smoothing noisy data, removing outliers, resolving inconsistencies If data is dirty mining results an unreliable output.
Some mining methods deals with incomplete, noisy data but they are not always robust So preprocessing step is to run your data through some cleaning routines.

Data Integration — Integrating multiple databases, data cubs, files Attributes have diff names in diff databases. Causes inconsistencies and redundancies

Cust_ID in one data str. and CustomerID in another store. So naming inconsistencies arise.

> Large no of redundant data slowdown or confuse discovery proc
So Data Integration need to be performed to detect and remove redundancies.

Data Transformation

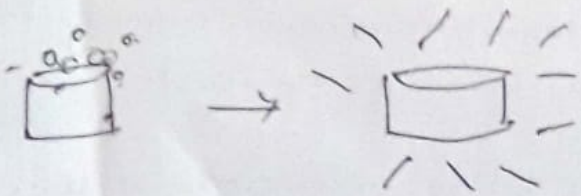
Neural N/w, Nearest neighbour classifiers, clustering provide better results if the data is normalized ie scaling to [0.0, 1.0]. Data Transformation operations are normalization, aggregation.

↳ provides or moves towards Succ of mining process

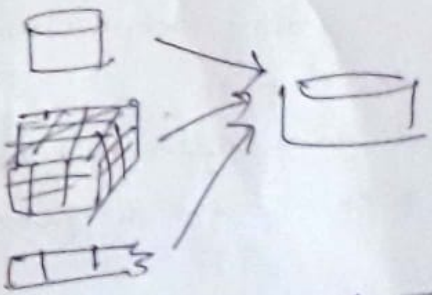
Data Reduction

Obtains a reduced representation of data set ie much smaller in volume yet produces same analytical results. Includes Data aggregation, — building data cube.
Attribute subset selection — removing irrelevant attributes thru correla
dimensionality reduction — uses encoding schemes
Numerosity reduction — replace data by clusters or parametric mo

Data can also be reduced by generalization with concept hierarc
ie low level concepts are replaced with high level concepts
Data discretization is also form of reduction.

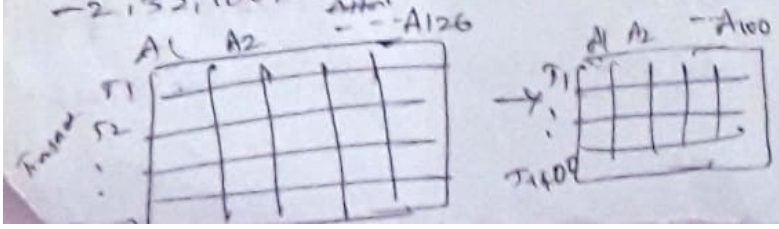


→ Data cleaning



→ Data Integration.

-2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48 → Data Transform



→ Data Reduction

⇒1.8 DATA CLEANING:-

This process fills missing values, smooths data, detect outliers, correct inconsistency in data.

1.8.1 Missing values:-

There are some methods for filling missing values

1) Ignore tuple → done when class label is missing
   - > not effective unless several attributes are missed.
   - > poor when percentage of missing values per attribute varies considerably.

2) Fill in Missing values manually → - > Time consuming
   - > Not feasible in large data set with many missing values

3) Use global constant to fill missing values → - > Replace missing attributes with same constant like unknown or -∞
   - > But use may mislead by seeing unknown
   - > Simple, not foolproof

4) Use attribute mean to fill missing values → Calculate mean of attribute and replace missing value of that attribute with that mean

5) Use attribute mean for all samples belonging to same class as given tuple → If we classify customers according to Credit risk we can replace mean of income attribute value to Credit risk Category of given tuple

6) Use most probable value to fill missing values → determined with regression, Bayesian formalism or Decision tree Induction
   - > Most popular one

1.8.2 Noisy Data:-

Noise is a random error or variance in a measured variable

Some of Smoothing techniques → Binning
- Regression
- Clustering

Binning :- 4, 8, 15, 21, 21, 24, 25, 28, 34

partition into equal frequency bins

Bin 1 : 4, 8, 15
Bin 2 : 21, 21, 24
Bin 3 : 25, 28, 34

Smoothy by bin means

Bin 1 : 9, 9, 9
Bin 2 : 22, 22, 22
Bin 3 : 29, 29, 29

Smoothing by bin boundaries

Bin 1 : 4, 4, 15
Bin 2 : 21, 21, 24
Bin 3 : 25, 25, 34

* smoothy by bin median

> Binning method smooths sorted data by considering neighbours.

> sorted values are distributed into bins

> performs local smoothing

→ Each bin is replaced by mean value of bin

→ Min, max values in bin are identified as bin boundaries. Each value is replaced by closest boundary value

— each bin value is replaced by bin median.

Regression :-

* Data is smoothed by fitting data to some fn (Regression)
* Linear Regression — finding best line to fit 2 attributes so that one attribute is used to predict other.
* Multiple linear Regression — More than 2 attributes are involved and data are fit to multidimensional surface

Clustering :

* Outliers are detected with clustering where similar values are grouped into clusters.
* values that fall outside clusters are outlier.

data cleaning as a process :-

* First step is Discrepancy detection
* Discrepancies are caused due to several factors
  - > poorly designed data entry forms that has more fields
  - > Human Error in entry
  - > deliberate Errors
  - > data decay (outdated address)
  - > Inconsistent data representation
  - > Inconsistent use of code
  - > Error in instrument devices
  - > System Errors
  - > in data integration ie many conventions

* Dealing discrepancy detection —— we use metadata
  What are domain, datatype of attribute
  What are acceptable values of attribute
  What is range of length of values
  Do all values fall within range

* We should look at inconsistent code and representation.
  Field overloading is an source of Error:— new attributes are squeezed into unused portion of already defined attribute

* Data should be examined regarding uniqueness rule, consecutive rule, Null rule
  uniqueness rule — each value in attribute must be dif from others
  consecutive rule — there should be no missing values b/w lowest & highest values of attribute
  Null rule — handling of blanks, unknown, Special Characters.

* There are number of tools for discrepancy detection
  Data Scrubbing tools —— use simple domain knowledge to detect and correct Errors in data.
  —— rely on parsing & fuzzy matching for detection of syntax Errors (works with grammar, logic)
  techniques when Cleaning

  Data Auditing tools — find discrepancies by analyzing data to discover rules, relationship & detect data that violate such Condn

2nd step

* Some Data inconsistencies are corrected manually bw or semiautomatic data Transformation.. After finding discrepancies we need to define & apply series of Transformation to correct them
  Data Migration tools —— used for Transformations.
  ETL tools —— Specify it through GUI

* All above process is Error prone & time consuming

* There are many new approaches to data cleaning
Potter's Wheel tool — public tool
Integrates discrepancy detect & Tx
Discrepancy checking is done automatically
User can develop & refine Tx
* Development of declarative languages for Tx operators is a new approach

1.9 DATA INTEGRATION AND TRANSFORMATION:

DM requires integration — merging of data from multiple data stores. Data need to be transformed into forms appropriate for mining.

Data Integration:

* Combines data from multiple sources such as multiple databases, data cubes or flat files
* There are many issues to consider during Integration
> Schema Integration, object matching are tricky
> Entity Identification problem — how entities from one multiple sources are matched?
i.e how analyst is sure of Customer_id in one database and Cust_number in another database are same
Meta data is used to solve problem. Ie for each attribute it include name, meaning, data type, range of values permitted for attribute, Null values for handling zero, or null values.
* Metadata is used to help transform data

Redundancy — Attribute is redundant if it can be derived from another attribute & set of attributes.
Cause of redundancy — Inconsistency in attribute or dimension naming inconsistent

* Redundancies are detected by Correlation analysis ie they measure how strongly one attribute implies other.

For Numerical att → we use Correlation coefficient (or) pearson product moment coefficient

$$r_{A,B} = \frac{\sum_{i=1}^{N}(a_i - \bar{A})(b_i - \bar{B})}{N\sigma_A\sigma_B} = \frac{\sum_{i=1}^{N}(a_i b_i) - N\bar{A}\bar{B}}{N\sigma_A\sigma_B}$$

N — no of tuples
a_i, b_i — values of A, B
$\bar{A}, \bar{B}$ — mean of A, B
σ_A, σ_B — SD of A, B
$\sum(a_i b_i)$ — sum of AB Crossproduct

* $-1 \leq r_{A,B} \leq +1$

If $r_{A,B} > 0$ A, B are +vely correlated
higher value stronger correlation.
higher value indicate A or B may be removed as redundancy

$r_{A,B} = 0$ A, B are independent
i.e no correlation

$r_{A,B} < 0$ A, B are -vely correlated

* Scatter plots are used to view Correlation
* Not always A causes B or B causes A.
Ex in demogeaphic database if A is no of hospitals & B is no of thefts in area. Here A will not cause B But both are linked to 3rd attribute population.

Correlation Categorical Attributes

Correlation is discovered by χ^2 (chi square) test

If A has c distinct values $a_1, a_2, \ldots a_c$
B has r " " $b_1, b_2, \ldots b_r$

Contingency table — has c values of A as Column
r values of B as rows

$$\chi^2 = \sum_{i=1}^{c}\sum_{j=1}^{r}\frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

o_{ij} — observed frequency
e_{ij} — Expected "

pearson χ^2 statistic

$$e_{ij} = \frac{\text{Count}(A = a_i) \times \text{Count}(B = b_j)}{N}$$

2×2 Contingency table

	male	female	Total
fiction	250(90)	200(360)	450
Non fiction	50(210)	1000(840)	1050
Total	300	1200	1500

(r-1)*(c-1)
↓
degree of freedom

no in parantheses — expected frequencies

we can verify Expected frequency

$$e_{11} = \frac{\text{count(male)} \times \text{count(fiction)}}{N} = \frac{300 \times 450}{1500} = 90$$

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

Third issue Detection and resolution of data value Conflicts

Attribute value from diff sources may differ

i.e weight is stored in metric units in one system and British Emperial Units in another.

Data Transformation :-

data are transformed int appropriate form for mining.

It involves

> Smoothing — remove noise ← Binning, Regression, Clustering

> Aggregation — Summary & aggregation operations are applied.
Daily sales is aggregated to compute monthly and annual total amount.

> Generalization — low level data are replaced by higher level by using concept hierarchies.
Street is generalized to City, Country
age " " " Young, Middle, Senior

> Normalization — data are scaled so as to fall with specified range i.e -1.0 to 1.0 or 0.0 to 1.0

> Attribute Construction — New attributes are constructed and added to help mining process.

Smoothing : Users specify Transformations to Correct data inconsistencies.

Normalization — Attribute is normalized by scaling its value
- so they fall within specified range such as 0.0 to 1.0,
- useful for classification alg involving NN, nearest neighbors classifier and clustering.

In NN normalizing up for each attribute speed up learning Phase.

In Distance Based methods — larger range values are scaled to smaller range (ie Binary attributes)

Min-Max Normalization

performs linear ⨍ⁿ on original data

min_A, max_A — minimum and maximum value of attribute A

A value of A is mapped to v' in range $[new\text{-}min_A, new\text{-}max_A]$

$$v' = \frac{v - min_A}{max_A - min_A}(new\,max_A - new\,min_A) + new\text{-}min_A$$

(Ex) min, max of income are ₹12,000 and ₹98,000

Range is [0.0, 1.0]

₹73,600 is Transformed to $\frac{73,600 - 12,000}{98,000 - 12,000}(1.0 - 0) + 0 = 0.716$

Z-score Normalization (or Zero-Mean Normalization)

Values are normalized based on mean & Standard deviation of A

v is mapped to v' by computing

$$v' = \frac{v - \bar{A}}{\sigma_A}$$

$\bar{A}$ — mean

σ_A — SD

this method is useful when min, max are unknown

Ex — Mean of A ₹54,000

SD of A ₹16,000

₹73,600 is Transformed to $\frac{73,600 - 54,000}{16,000} = 1.225$

Normalization by decimal Scaling :-

normalizes by moving decimal points of values of attribute A

No of decimal points moved depend on maximum absolute value of A. v is transformed to v'

$$v' = \frac{v}{10^j}$$

j = smallest integer such that $Max(|v'|) < 1$

Ex — Values of A range from −986 to 917.

Max absolute value of A is 986

∴ j = 3 (1000)

−986 is normalized to −0.986

917 " " 0.917

Attribute Construction — new attributes are constructed from given attr and are added to improve accuracy & understanding of structure in high dimensional data

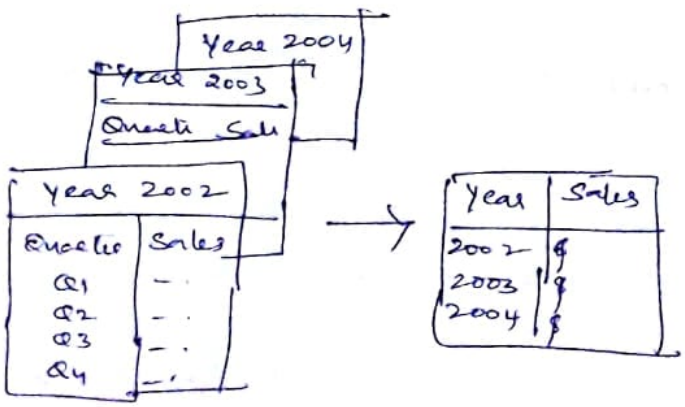
1.10 DATA REDUCTION :-

Complex data analysis and mining on huge amounts of data can take a long time, making such analysis impractical or infeasible. Data reduction is applied to obtain a reduced representation of data set ie much smaller in volume yet maintain integrity of original data set. Thus mining on reduc data set should be more efficient yet produce same analytical results

Strategies of Data reduction

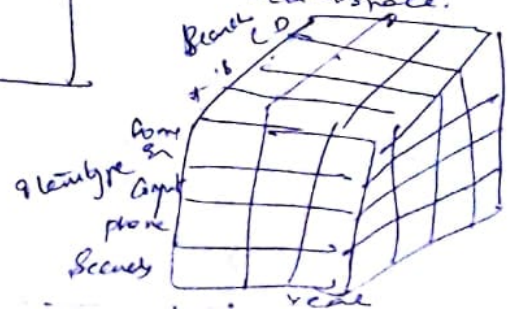
Data Cube Aggregation : — Aggregation operations are applied to the data in construction of data cube

Sales data from 2002 to 2004 is present which is in sales per Quarter but we are interested in annual sales. So data can be aggregated. Resulting data is smaller in volume w/o loss of information necessary for analysis.



Data Cube store multidimensional aggregated information

* Each cell holds aggregated data value corresponding to data point in multidimensional data space.



* Concept hierarchies allows data to be analyzed at multiple levels of abstraction.

Data Cube provides fast access to precomputed, summarized data which benefit OLAP & Data mining

Cube at low level — Base Cuboid
└ Corresponds to Individual Entity

at High level — Apex Cuboid
Ex-It may give total sales for all 3 years for all item types & for all branches

Data Cube — called as lattice of Cuboids

Attribute Subset Selection :-

Data set may contain hundreds of attributes, many of which may be irrelevant to task or redundant. The domain Expert can pick useful attributes but which is difficult & time Consuming

Attribute Subset selection reduces data size by removing irrelevant or redundant attributes

Goal — To find minimum set of attributes such that resulting probability distribution of data is as close as possible to original distribution.

How to find good subset of attributes —

for n attributes there are 2^n possible subsets

* Heuristic methods are used for Attribute Subset Selection
↳ Greedy methods
Make locally optimal choice which lead to globally optimal solution.

* Best (& worst) attributes are determined using statistical tests which assume attribute are independent to one another
Some of them methods are

Stepwise forward Selection — Starts with Empty set of attributes
Best attributes are determined & added to reduced set

(60) At each step best of remaining attributes is added to set

Ex Initial attribute set {A1, A2, A3, A4, A5, A6}

Initial reduced set : { }
{A1}
{A1, A4}

reduced attribute set :
{A1, A4, A6}

Stepwise Backward Elimination — starts with full set of attributes
— at each step removes worst attribute in set

Ex — Initial attribute set {A1, A2, -- A6}

⇒ {A1, A3, A4, A5, A6}

⇒ {A1, A4, A5, A6}

⇒ {A1, A4, A6}

Combination of forward selection & Backward Elimination — The procedure selects the best attribute & removes worst attribute from any remaining attributes

Decision Tree Induction — alg such as ID3, C4.5, CART are used for classification. ↳ constructs flow chart like structure where each internal node denotes test on attribute each branch correspond to the outcome of test & each leaf node denotes class prediction.
At each node alg chooses "best attribute to partition data into individual classes.
Attribute in tree form reduced subset of attribute.

⇒ Dimensionality Reduction :-

↳ Encoding or tx are applied to get reduced representation of original data

Lossless — If original data is reconstructed from compressed data w/o any loss of info then it is lossless

Lossy — if we reconstruct only an approximation of original data then it is lossy

Two popular and effective methods of lossy dimensionality reduction are wavelet transforms & principal component analysis.

Curse of dimensionality
When dimen ↑ data become sparse
Density, distance b/w point are critical to clustering / outlier analys.

Dimensionality reduct — avoid curse of dimension
— reduce time & space in
— eliminate irrelevant feature
reduce noise

Wavelet Transforms (used for image compression) (18) (20)

* Discrete wavelet Transform (DWT) is linear signal processing technique when it is applied to data vector X it is transformed to X'
  / X' → with wavelet co-efficients
  x and x' → same length
  $X = (x_1, x_2, \ldots x_n)$ ie n dimensional vector (n attributes)
* Both vectors are of same length but still this method is useful for data reduction
  > these wavelet Transformed data can be truncated
  > Compressed data is retained only with strongest wavelet co-effic ie co-efficients larger than user specified threshold are retained, all other co-efficients are set to zero (0).
  > Now data is sparse and operations are performed very fast in wavelet space
  > Wavelet TX is used to remove noise w/o smoothing.
* DWT is closely related to DFT - discrete fourier Transform which involves sines & cosines.
  DWT achieves better lossy compression ie DWT provides more accurate approximation of original data than DFT. It also requires less space than DFT.
* There are several families of DWT such as Haar-2 and Daubechies-4
* General procedure uses pyhierarchial pyramid algorithm and is given below
  ① length L of i/p vector must be integer power of 2
  If necessary pad vector with zeros as necessary
  ② each TX involves two fn — data smoothing
  — weighted difference (bring detailed feature of data)

③ 2 fn are applied to all pairs of data which results in 2 sets of data of length l/2

④ 2 fn are recursively applied to data obtained in prev loop until datasets are of length 2.

⑤ Selected values obtained from above loops are designated as wavelet co-efficients of fx data.

* Matrix multiplication is applied to I/p data to get Wavelet Co-efficients

Matrix — orthonormal
- cols are unit vectors
- orthogonal ie inverse is its transpose

* Alg has complexity of O(n)
n - I/p vector

* Haarlet fx are applied to data cube. First apply fx to 1st dimension, then to second & so on

Principal Component Analysis (PCA)

* Data consist of vectors described by n attributes ∀ dimens.
PCA also called as Karhunen-Loeve or K-L method
Search for k n-dimensional orthogonal vectors (k < n)

* Original data is projected into smaller space

Basic procedure

① I/p data is normalized so attribute falls within same range
So that attributes with large domain will not dominate attribute with smaller domains.

② PCA computes K orthonormal vectors
- unit vectors to each other.
- known as principal components

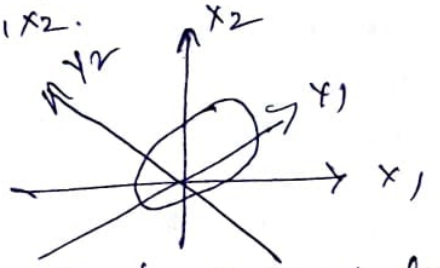
I/p data as linear combination of principal components

(3) Principal Components are sorted in decreasing significance of strength. These serve as new set of axes providing info about variance.

1st axis — More variance

2nd axis — next highest variance

Y_1, Y_2 — principal components of data originally mapped to X_1, X_2.



(4) Size of data can be reduced by eliminating weakes components. ie with low variance

* PCA is computationally inexpensive
* Can be applied to ordered & unordered data sparse and skewed data

Numerosity Reduction:-

Parametric — a model is used to estimate data. only parameters are stored instead of actual data

Nonparametric — Do not assume model for storing reduced representation of data include histogram, Clustering & Sampling -

Regression and Log linear Models:-

used to approximate given data

linear Regression — data is modelled to fit in st-line

. Y (response variable) is modeled as linear fn of ~~random variable~~ Predictor (~~Response~~)

$$Y = wx + b$$

(64)

w, b } regression co-efficients

x, y } numerical database attributes

Co-efficients are solved by method of least Squares

Multiple linear Regression — linear fn of 2 or more predictor Variables

Log linear models — approximate discrete multidimensional probability distribution

n dimensions are present

Each tuple is considered as point in n-dimensional space

* These models are used to estimate probability of each point in multidimensional space
* used to construct higher dimensional dataspace from lower dimensional space
* Also used for dimensionality reduction, data smoothing
  ↓
  lower dimensional point occupy less space than original
* used on sparse data

Histograms:-

* Use binning to approximate data distributions
* attribute A is partioned into disjoint subsets or buckets
  If Each bucket has single value then it is called singleton buckets

Partioning rules :-

① Equal-width — width of Each bucket range is uniform — $w = \frac{B-A}{N}$ A,B low & high of attribute

② Equal frequency or equal depth — frequency of each bucket is constant (same no. of samples)

③ V-optimal — V-optimal histogram is one with least Variance. Histogram Variance is weighted Sum of original values of each bucket.

④ Max-Diff — conside diff b/n adjacent values. Bucket boundary is established b/n each pair for pairs have β-1 largest difference

β — User specified no of buckets

* V-optimal, MaxDiff — more accurate

Clustering :-

* Considers data tuples as objects. partition objects into clusters. objects within clusters are similar to each other and dissimilar to other objects in other clusters.
* Similarity is based on distance fn
  Quality of cluster — its diameter — max distance b/w 2 objet in cluster
  — Cluster centroid (average point in space of cluster)
* clustering is more effective for data that can be organized into distinct clusters than for smeared data
* Multidimensional Index trees are used for providing fast data access which is used for hierarchical data reduction and also provide approximate answers to queries
* Index tree recursively partitions multidimensional space with root node representing entire space. Tree are balanced. parent node has keys and pointers to childNodes. Leaf node has pointers to data tuples they represent
* Index trees store aggregate as well as detailed data at diff levels of abstraction, which provide hierarchy of clusters.
  Each child node is bucket then tree is hierarchical histogram
* There are many measures for defining Clusters and Cluster Quality.

Sampling :-

↳ one form of Data Reduction
. larger data set is represented by smaller random sample (subset) of data

Data set → D
Total tuples → N

We draw 's' sample of N tuples from D

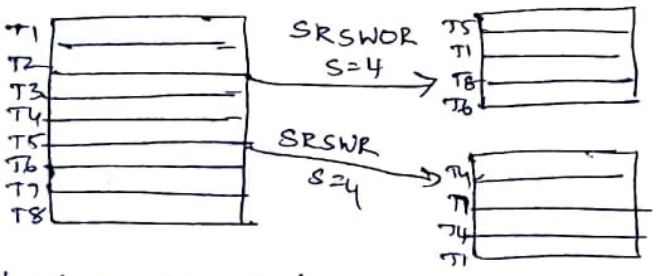
's' < N

probability of drawing any tuple is $\frac{1}{N}$

ie all tuples are equally likely to be sampled

Simple Random Sample With Replacement (SRSWR) of size s :-

Similar to SRSWOR bt each time a tuple is drawn from D it is recorded and replaced. ie it placed back in D and so we can draw it again.



Cluster Sample :-

D is grouped to M disjoint clusters then SRS of 's' clusters are obtained where $\frac{s}{\text{tuples}} < M$

Eg- tuples in database are retrieved one page at a time so each page is considered as clusters.

Stratified Sample :-

> D is divided into mutually disjoint parts called "strata"

> Apply SRS at each stratum and generate Stratified sample

> useful when data is skewed

Eg — a stratified sample is obtained from Customer data where ~~each~~ stratum is created for each Customer age group.

Advantage

> Cost of obtaining sample is proportional to size of sample s instead of N

> Sampling complexity is "Sublinear" to size of data

↳ increases as no of dimensions increases.

> Sampling is refinement of reduced data set. Such set

1.11 DATA DISCRETIZATION AND CONCEPT HIERARCHY GENERATION :-

* Data discretization techniques — reduce no of values for given continuous attribute by dividing range of attribute into intervals. Labels are used to replace actual data values. So that it simplifies original data.
* Supervised discretization
  - uses class info
* unsupervised discretization
  - doesn't use class information.
* Top down discretization (or) splitting
  - starts with one or few points (split point) to split range and repeat this recursively on resulting intervals
* Bottom up discretization (or) merging — contrast to splitting
* Discretization can be performed recursively on attribute to provide hierarchical partitioning of value known as Concept hierarchy which is useful for mining at multiple levels of abstraction
  - One form of data reduction where low level concepts are replaced by higher level concepts
* Manual definition of concept hierarchies is tedious, last and time consuming and so several discretization methods are used to automatically generate concept hierarchies for numerical attributes and for categorical it can be generated automatically.

Discretization and Concept Hierarchy Generation for Numerical Data.

Concept hierarchies for numerical attributes can be constructed automatically based on data discretization.

Binning :-

* Top - down splitting technique based on no of bins
* Attribute values are discretized by applying equal-width (or equal-frequency) and then replace each bin value by bin mean or median
* Apply these techniques recursively to resulting partitions to generate Concept hierarchies
* Doesn't use class Information and so called as unsupervised discretization technique
* Sensitive to { presence of outliers, user specified no of bins

Histogram Analysis :-

* Unsupervised discretization technique
* Histogram partition value of A into buckets with equal widths (or equal frequency).
  - Range
  - Same no of tuples
* Apply Histogram analysis algorithm recursively to generate multilevel Concept hierarchy. Recursive procedure stops when it reaches prespecified no of concept levels (or when it reaches minimum Interval size per level.
  - no of values for Each partition.

Entropy Based Discretization

* Entropy — one discretization measure
  - by Claud Shannon
  - Supervised, Top down Splitting
* To discretize a numerical attribute 'A' method selects value of 'A' that has minimum Entropy as a split point & recursively partition the resulting intervals to get hierarchical discretization.

- Its treated refined by increasing sample size.

* D ——— set of data tuples & class label attribute provides class info per tuple

* Basic method of Entropy based discretization,

① Considers split point to partition range of A
A can partitions tuples in D into 2 subsets satisfying Condition A ≤ split point and A > split point, which creates a binary discretization.

② If we want to classify tuples in D by partitioning on a attribute A & some split point
2 classes < C1, C2
1st partion ——— contain many tuples of C1 but also some of C2

Still how much info is needed for perfect classification. This info is called Expected information for classifying tuple in D based on A.

$$Info_A(D) = \frac{|D1|}{|D|} Entropy(D1) + \frac{|D2|}{|D|} Entropy(D2)$$

D1, D2 correspond to A ≤ split point & A > split point

|D| — no of tuples in D
in classes C1, C2, ---- Cm

$$Entropy(Di) = -\sum_{p=1}^{m} P_p \log_2(P_p)$$

P_p — probability of class C_p in D1

* process of determining a split point is recursively applied to each partition until stoping Criteria is met.

↓

minimum info requirements on all Candidate split point < threshold ε

Or no of intervals > threshold max. Intervals

Interval Merging by χ^2 Analysis:

* χ^2 based method, Bottom-up approach - finds best neighboring intervals & merge to form larger intervals recursively
* If two adjacent intervals have similar distribution of classes then intervals can be merged. else separate

procedure

* each value of A is considered as one interval
* χ^2 test is performed on all adjacent intervals. we merge intervals with least χ^2 value. (Similar class distribution)
  It is repeated until some stoping criteria is met.
* χ^2 used in data integration and we can construct a contingency table for our data
  - 2 columns (2 adjacent intervals)
  - m rows (no of distinct class)

  we use O_{ij} / E_{ij}
* Stopping criteria
  1. χ^2 values for all pairs exceeds some threshold (significance level)
     High value of significance — over discretization
     low " " — under discretization
     ↓ 0.10 to 0.01
  2. No of levels Intervals cannot be over prespecified maximum (10 to 15)
  3. class frequencies must be consistent within interval

Cluster Analysis:

* Discretize numerical attribute A by partioning values of A into clusters
  - considers closeness of datapoints so that it produces high quality discretization results,

* follows either Top-down or Bottom up strategy
  Each cluster forms a node
* Initial clusters may be further decomposed into several subclusters forming lower level of hierarchy.
  Or clusters are formed by repeatedly grouping neighbouring clusters to form high level concepts

Discretization by Intuitive partioning :-

* we would like to discretize numerical ranges partitioned into relatively uniform, easy-to-read intervals
  Annual salaries broken into ranges like [$50,000, $60,000] are desirable than ranges like [$51,263.98, $60,872.34]
* 3-4-5 rule —— to segment numerical data into relatively uniform, natural intervals
  ↳ partitions given range of data into 3, 4 or 5 equal width intervals recursively and level by level based on MSB

  Rule is as follows
  > If interval covers 3, 6, 7 or 9 distinct values at MSB then partition range into 3 intervals
    for 3, 6, 9 → 3 equal width intervals
    7 → 2-3-2
  > If it covers 2, 4, or 8 distinct values at MSB then partition to 4 equal width intervals
  > If it covers 1, 5 or 10 distinct values at MSB then partition range into 5 equal width intervals.

Rule is recursively applied to each interval creating concept hierarchy

−$ 351,976.00

$ 4,700,896.50

(27)

* user need automatic generation of hierarchy of profit

[−$1,000,000 ... $0]

data b/n 5th and 95th percentile are $159,876 & $1,838,761

Results are :

① Min = −$351,976.00
MAX = $4,700,896.50

Low and high values for 1st level of discretization are

5th percentile → −$159,876

95th percentile → $1838,761

② msd is at million digit position

msd = 1,000,000

Rounding LOW' = −$1,000,000

HIGH' = +2,000,000

③ We have to do 3 partitions

$$\frac{\text{HIGH}' - \text{LOW}'}{\text{msd}} = \frac{2{,}000{,}000 - (-1{,}000{,}000)}{1{,}000{,}000} = 3$$

3 equal width segments

(−$1,000,000 ... $0) ($0 ... $1,000,000] & ($1,000,000 ... $2,000,000]

CONCEPT HIERARCHY GENERATION FOR CATEGORICAL DATA:-

Categorical data — Nonordered discrete data

↓

finite no of distinct values

No ordering among values

Specification of a partial ordering of attributes Explicitly at the schema level by users or experts :-

- Involve group of attributes
- we can define concept hierarchy by specifying partial (or) total ordering of attributes at schema level

location dimension of Datawarehouse contain following attributes — street, city, province_or_state and country.

Total ordering is street < city < province_or_state < country.

Specification of portion of hierarchy by explicit data grouping

In large databases it is difficult & unrealistic to define entire concept hierarchy. So we specify explicit groupings of for a small portion of intermediate level data.

Ex After specifying province and country form hierarchy at schema level we can use some intermediate levels such as {Alberta, Saskatchewan, Manitoba} ⊂ prairies-Canada and {British Columbia, prairies-Canada} ⊂ Western Canada.

Specification of set of attributes but not of partial ordering :

User specify set of attributes forming concept hierarchy but omit their partial ordering. System automatically generate attribute ordering to construct hierarchy

ie Based on no of distinct values per attribute in given set it automatically generates hierarchy

Country with small no distinct values than street.

Attribut with most distinct values — at Low level of hierarchy

" low " — top level

Specification of only partial set of attributes :-

Sometimes users have vague idea of what should be included in hierarchy. So the can include only a small subset of relevant attributes in the hierarchy specification.

for ex the may include only street, city. To handle such partially specified hierarchies we should embed data semantics in database, so so attributi with right semantic connections can be pinned together.

ARCHITECTURE OF DATA MINING SYSTEMS :-

A good system architecture will facilitate the system to make best use of software environment which accomplish DM tasks in an efficient and timely manner interoperate and exchange the information with other information systems be adaptaple to user requirements and evolve with time.

Desired Architecture for DM systems :-

Based on different architecture designs a DM system can be integrated with a DB/DW system using following compiling systems.

a> **No Coupling :-** A DM system will not utilize any function of a DB (or) DW system. It fetches dote from particular source process data and by using some mining algorithm and then store the mining results in the another file.

b> **Loose Coupling :-** Loose coupling means that a DM system will use some facilities of a DB (or) DW system, fetching data from a data repository managed by these system performing data mining and then storing the mining results either in a File (or) in a designated place in database (or) DW.

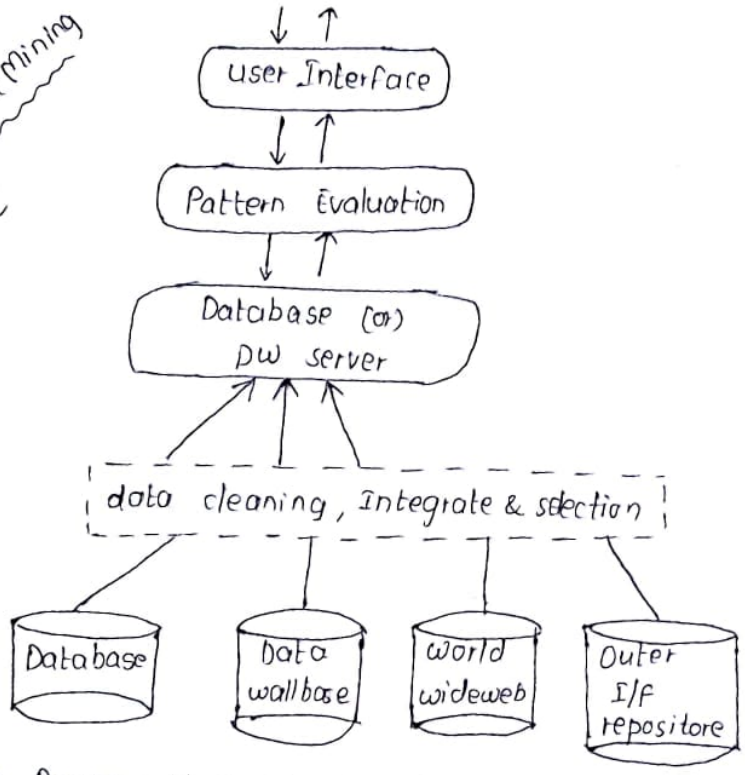
c> **Semitight Coupling :-** Semitight coupling means that besides linking a DM system to DB/DW system efficient implementation of a few essential date mining primitives can be provided in DB/DW system. These primitives can include sorting, and precomputation of some essential statistical measures such as sum, count, max and min so on

d> **Tight coupling :-** Tight coupling means that a DM systems is smoothly integrated into DB/DW system. The DM subsystem is treated as functional component of an

information system. DM Queries and functions are optimized based on mining Query analysis, data, structure, indexing schema & Query process methods of DB/DW system.

DM, DB and DW system will evolve and integrate together as one I/p system with multiple functionalities which provides a uniform I/f processing system.

Architecture of Typical data mining System



Knowledge Base:- It is the knowledge which is used to guide the search (or) evaluate the interestingnees of resulting patterns. There are dif kinds of knowledge such as concept hierarchies - To organise attribute values into different abstractions briefs - to access patern's interestingness.

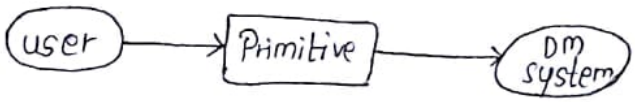
Data Mining Engine - It consists of set of functional modules for tasks such as characterization, association & correlation analysis, classification, prediction, eluctes analysis, outlier analysis and so on...

Pattern-Evaluation Module – It employs interestingness measures and interacts with DM module so as to focus the search towards interesting pattern.

User Interface:- This module communicates b/w user and data mining system allowing user to interact with system by specifying on Query (or) task, providing i/p to help focus search and performing exploratory date mining based on the intermediate DM results.

DATA MINING TASK PRIMITIVES:

* Each user will have a determining task in mind i.e some form of data analysis that he (or) she would like to have performed.
* DM task is specified in form of DM Query which is i/p to DM system.
* DM Query is defined in terms of DM task primitives.
* These primitives allow user to interactively communicate with the DM system during discovery in order to direct the mining process.



Incorporating these primitives in an DM Query Language

→ More flexible user interaction
→ foundation for design of GUI
→ Standardization of DM industry and practise

...formation system. DM Query ...

* DM primitives specify following → Task-relevant data
  → Type of knowledge to be mined
  → Background knowledge
  → Pattern interestingness (or) measurements
  → Visualization of discovered pattern

Task-relevant data to be mined

* It specifies position of database (or) set of data in which is interested.

  It includes ✓ Database (or) Dataware house name
  ✓ Database tables (or) datawarehouse cubes
  ✓ Condition for data selection
  ✓ Relevant attributes (or) dimensions
  ✓ Data grouping criteria

* Task relevant data in general known as Minable view. The portion of database to be mined is called minable view.

Example If determining task is to study association b/w items frequently purchased of All electronics by customers in canada, the task relevant data can be specified as

- name of database (or) DW to be used (eg. All electionics db)
- name of tables (or) data cubes containing relevant date (eg. item, customer, purchases, item sold)
- conditions for selecting relevant data (eg relative data pertaining to purchases made in canada for current year.

→ schema hierarchy (order among attributes in DB schema)

eg. street < city < province < country

or state

→ set-grouping hierarchy (given attribute/dimension groups)

eg {20....39} = young

{40---59} = middle aged

→ operation derived hierarchy

email address : dmbook@cs.sfu.ca

login name < dept < university < country

operations specified by user, operations can include the decoding of int encodes string, int extraction complex data objects and data cluster

→ Rule based hierarchy

how-profit-margin(x) <= price(x, P1) and

cost(x, P2) and (P1 - P2) < $50

Information about domain to be mined Usefy in discovery process.

⇒ Measurments of pattern Interestingness :-

They are used to guide mining ~~Discovery~~ process. Different kinds of knowledge may have different interesting measures

Ex Interesting-measures for association rule include support and confidence

Rules whose support & confidence value are below user-specified threshold are considered uninteresting.

information system ...

- relevant attributes (or) dimensions
  (eg name & price from itemtable
  & income age from customertable)

Type of knowledge to be mined:-

It determines data mining function to be performed such as characterization
Discrimination
Association
Classification/predictor

Outlier analysis
other DM tasks

Example

A user studying buying habits of All Electronics, customers may choose to mine association rule of the form

$P(X: customer; w) \wedge Q(X; Y)) \Rightarrow buys(X; z)$

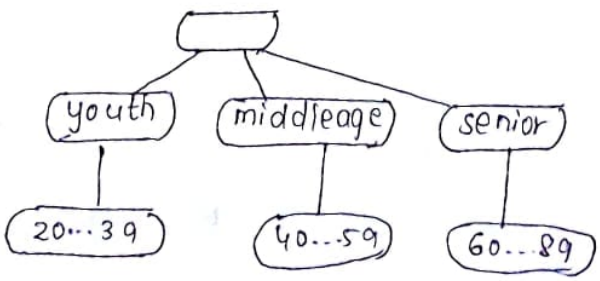
Eg:

$age(X; "30\ldots39") \wedge income(X; "40\ldots50k")) \Rightarrow buys(X; "VCR")$

[22%, 60%]

Background knowledge

This kind of knowledge is useful for guiding the knowledge discovery process and for evaluating patterns found. concept hierarchies are popular form of Background knowledge which allow data to be mined at multiple levels of abstraction.



Simplicity

Eg :- (Association) rule length, decision tree size, Can be viewed as function of pattern structure, defined pattern size in bits, (or) no of attributes (or) Operations appeared in pattern

✓ Certainity - Which access validity (or) trustworthiness of pattern.

Eg :- Confidence $p(B|A)$ — for Association Rule classification reliability (or) Accuracy

Certainty factor,

Rule Strength

Rule Quality

Discriminaty weight Etc.

✓ utility — Eg Support = $p(A \cup B) = \frac{\text{tuple (or) transaction both A \& B}}{\text{total tuple.}}$

Potential Usefulness

Support (Association)

Noise Threshold (Description)

✓ Novelty — that Contributes new info (or) increased performance to give not previously known Surprising (used to Remove redundant rule)

Visualization of Discovered patterns

This refer to form in which discovered patterns are to be displayed i.e d/f Backgsounds / usages many require d/f forms of presentation.

Eg :- Rules, Tables, Crosstabs, pie / bas Chart etc

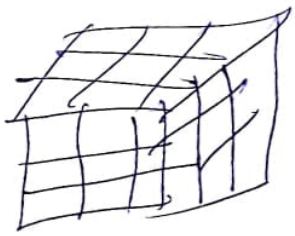
Rules

age (x, " young) and income (x, "high") ⇒ class (x, "A")
age (x, " young) and income (x, "Low") ⇒ class (x, "B")
age (x, "old) ⟶ class (x, "C")

Tables

age	income	class
young	high	A
young	low	B
old	high	C
old	low	C

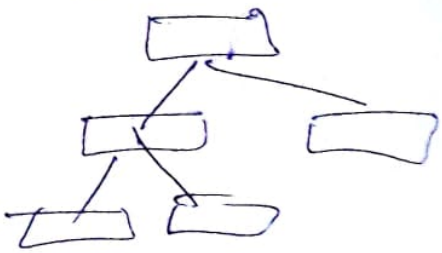
DataCube



Crosstab

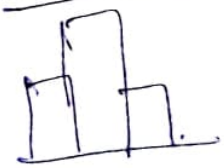
age	income		class		
	high	low	A	B	C
young	83	-			
old	-	-			
Cont					

Decision tree



pie-chart

Barchart



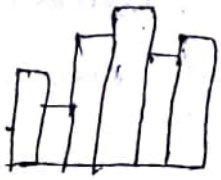
Visualization of discovered patterns in various forms can help users with diff background to identify patterns of interest and to interact can guide the system in future discovery ~~some~~ of visual form are rules, tables, Crosstabs, piechart, barcharts decision tree.

(5)

- relevant attributes (or dimensions
  (eg name & price from itemtable
  & income, age from customertable))

Type of knowledge to be mined :-

It determines data mining function to be performed such as characterization
Discrimination
Association
Classification/prediction
clustering
Outlier analysis
Other DM tasks



Example

A user studying buying habits of All electronics customers may choose to mine association rule of the form

$$P(X : customer, W) \wedge Q(X, Y) \Rightarrow buys(X, Z)$$

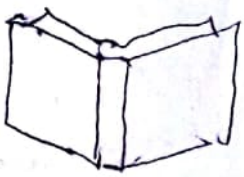
eg:

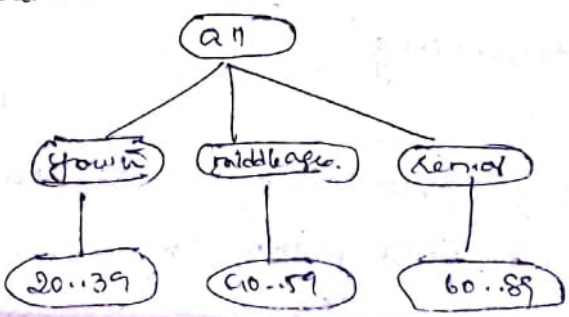
$$age(X, "30\ldots39") \wedge income(X, "40\ldots50K") \Rightarrow buys(X, "VCR")$$

[2.2%, 60%]

Background knowledge

This kind of knowledge is used for guiding the knowledge discovery process and for evaluating patterns found. concept hierarchies are popular form of Background knowledge which allow data to be mined at multiple levels of abstraction.





✓ schema hierarchy (order among attributes in DB schema)

E.g street < city < province < country
& etc

✓ Set-grouping hierarchy (given attribute / dimension groups)

E.g {20..39} = young

{40..59} = ~~Senior~~ middle aged

✓ Operation derived hierarchy

email address : dmbook @ cs.sfu.ca

login-name < dept < university < country

operations specified by users, operations can include the decoding of int encoding string, int extraction from complex data objects and data clustering

✓ Rule based hierarchy

low_profit_margin(X) <= price(X, P1) and cost(X, P2) and (P1 − P2) < $50

Diff about domain to be mined require in discovery process

Measurements of pattern Interestingness :-

They are used to guide the mining process.

Diff kinds of knowledge may have diff interesting measures.

Ex Interesting measures for association rule include support & confidence.

Rules whose support & confidence values are below user-specified threshold are considered uninteresting.

✓ Simplicity

Eg (association) rule length (simple measure. CNF — no of conjuncts), (decision) tree size (rule)

Can be viewed as fn of pattern structure, defined in terms of no of attributes (or) operations appearing in rule

* DT

✓ Certainty - which gives validity (or) trustworthiness of pattern

Eg Confidence $P(\frac{B}{A})$ - for Association rule
classification reliability (or) accuracy
Certainity factor
rule strength
rule Quality
Discriminating weight etc

✓ Utility - Eg Support = $P(A \cup B)$ = $\frac{\text{tuple cont both A \& B}}{\text{total num}}$
potential usefulness
min lot of items are min supp
support (Association), noise threshold (description)

✓ Novelty - that contribute new int (or) increased performance to give pattern set
not previously known
Surprising (used to remove redundant rules)

Visualization of Discovered patterns

This refers to form in which discovered patterns are to be displayed ie dlf backgrounds/images may require dlf forms of presentation

Eg :- Rules, tables, Crosstabs, pie/bar chart etc

Rules

age (x, "young) and income (x, "high") ⟹ class (x, "A")
age (x, "young) and income (x, "low") ⟹ class (x, "B")
age (x, "old") ⟹ class (x, "C")

Table

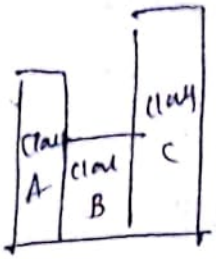
age	income	class	(conf)
young	high	A	1242
young	low	B	1342
old	high	C	856
old	low	C	752

Crosstab

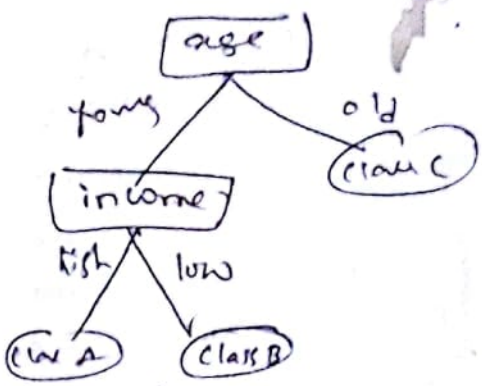
age	income		class		
	high	low	A	B	C
young	85				
old	65				
Cont					

piechart

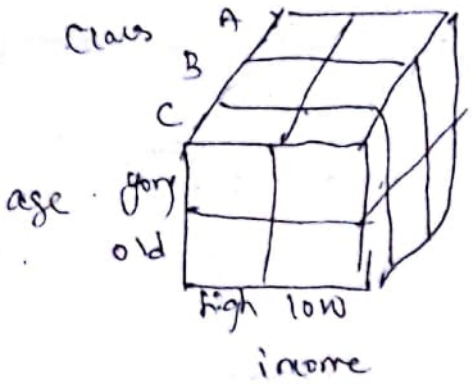
Barchart



decision tree



Data cube



Visualization of discovered patterns in various forms can help users with d/f backgrounds to identify patterns of interest and to interact or guide the system in further discovery process. Visual forms are rules, tables, crosstables, piechart, barchart, decision tree etc.

DATA MINING QUERY LANGUAGE (DMQL)

* DMQL can provide the ability to support ad-hoc and interactive data-mining in order to facilitate flexible effective DM duties
* By providing standardized language like SQL DMQL has
  - ✓ hope to achieve similar effect like SQL has on relational database
  - ✓ foundation for system development & evolution
  - ✓ facilitate info exchange, technology transfer, commercialization & wide acceptance etc,

* DMQL is designed with the primitives described earlier

Syntax for DMQL

Syntax for Specification of
- task relevant data
- kind of knowledge to be mined
- concept hierarchy specification
- interestingness measures
- pattern presentation & evaluation

* If we put all these primitives together it is called a DMQL Query
* In an inductive database (IDB) ordinary queries can be used to access and manipulate data while inductive queries can be used to generate, manipulate and apply patterns. KDD becomes an Extended Querying process

Features of DMQL

- Data mining query language satisfy the closure property
- A classical approach in database theory
- Relation algebra
- Has Optimization Strategies

* Supports Association rule mining process.

Syntax Syntax for Specification of Task relevant data:

- Use database database_name (or) Use data warehouse data warehouse_name
- from relation(s)/cube(s) [where condition]
- in relevance to attr_list (or) dim_list
- Order by order_list
- group by grouping_list
- having Condition

Example: This Example shows how to use DMQL to specify task relevant data for mining of association rules frequently purchased at AllElectronics by Canadian customers w.r.to customer income & age. In addition the user specifies that she would like data to be grouped by date. The data are retrieved from relational database

Use database AllElectronics_db
in relevance to I.name, I.price, C.income, C.age
from Customer C, item I, purchase P, items_sold S
where I.item_ID = S.item_ID and
S.trans_ID = P.trans_ID and
P.cust_ID = C.cust_ID &
C.address = "Canada"
group by P.date

Syntax: kind of knowledge to be mined:

- Characterization — specifies characteristic descriptions to be mined are to be

Mine_knowledge-Specification :::=
mine characteristics [as pattern_name]
analyze measure(s)

- Discrimination

Mine_knowledge-Specification :::=
mine comparison [as pattern_name]
for target_class where target_condition
{ versus contrast_class_1 ... /where contrast_condition_1 }
analyze measure(s)

Eg

mine comparison as purchasegroups
for bigspenders where avg(I.price) >= $100
versus budgetSpenders where avg(I.price) < $100
analyze count.

- Association

Mine_knowledge_specification :::=

mine associations [as patternname]
[matching <metapattern>]

Eg:

mine associations as buyinghabits
matching P(X:custom, W) ∧ Q(X,Y) ⇒ buys(X, Z)

- Classification

Mine_knowledge_Specification :::=

mine classification [as patternname]
analyze classifying_attribute_or_dimension

- Other patterns

Clustering, outlier analysis, predictor

Syntax: Concept hierarchy Specification

* To specify what concept hierarchies to use

use hierarchy <hierarchy> for <attribute_or_dimension>

* We use dif syntax to define dif type of hierarchies.

- Schema hierarchy

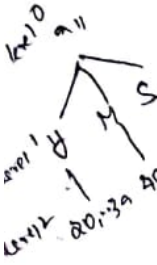
define hierarchy time_hierarchy on date as [date, month
Quarter, year]

↓ predefined concept hierarchy for schema

date(day, month, quarter, year) as hierarchy

- Set-grouping hierarchies

define hierarchy age-hierarchy for age on Customer as



{20,...,89} level 1 : {young, middle-aged, senior} < level 0 : all
level 2 : {20,...39} < level 1 : young
level 2 : {40,...,59} < level 1 : middle-aged
level 2 : {60,...,89} < level 1 : Senior

- Operation-derived hierarchies

define hierarchy age-hierarchy for age on Customer as

{age_category(1),...,age_category(5)} :=
cluster (default, age, 5) < all (age)

Here clustering alg is to be performed on all of ages values in Customer relation to form 5 clusters. age_category(1) is ranges in ascending order

- Rule based hierarchies

define hierarchy profit-margin hierarchy on item as

level-1 : low-profit-margin < level-0 : all
if (price - cost) < $50

level-1 : medium-profit-margin < level-0 : all
if (price - cost) < $50 and ((price - cost) <= $250))

level-1 : high-profit-margin < level-0 : all
if (price - cost) > $250

Specification of Interestingness Measures

- Interestingness measures and thresholds can be specified by user with Stmt

with <interest-measure-name> threshold = threshold-value

Eg with support threshold = 0.05
with confidence threshold = 0.7

Specification of pattern presentation

* Specify the display of discovered pattern

display as <result-form>

* To facilitate interactive viewing at dif concept levels the following syntax is used.

Multilevel_Manipulation := roll up on attribute_(or_dimension
| drill down on attribute (or dimension
| add attribute (or dimension
| drop attribute (or dimension

Putting it all together : A DMQL Query

Use database AllElectronics_db
Use hierarchy location_hierarchy for B.address
mine characteristics as customer purchasing
analyze count%
in relevance to C.age, I.type, I.place_made
from Customer C, Item I, purchase P, Items_Sold S,
Works_at W, branch.
where I.Item_ID = S.item_ID and S.trans_ID =
P.trans_ID and P.Cust_ID = C.Cust_ID and
P.method_paid = "AmEx"
and P.Empl_ID = W.Empl_ID and W.branch_ID=
B.branch_ID and B.address = "Canada"
and I.price >= 100
with noise threshold = 0.05
display as table

Based on all primitives we design a query language ie DMQL

Other Data mining Language & standardization Efforts:

- Association rule language specifications
  - ✓ MSQL (Imielinski & Virmani'99)
  - ✓ MineRule (Meo Psaila and Ceri'96)
  - ✓ Query flocks based on Datalog Syntax (Tsur et al'98

MSQL – Mini Structured Query Language:

Syntax

```
GetRules (c) [INTO <rulebase name>]
[WHERE <rule constraints>]
[SQL – groupby clause]
[USING encoding-clause]
```

Where c – Source table

rulebasename — name of the object

rule constraints — conditions on desired rules

MINE Rule:

It extracts association rule b/n values of attributes in relational table

Syntax:

```
MINE RULE <Tablename> AS
SELECT DISTINCT [<cardinality>] <Attributes>
AS BODY [<cardinality>] <Attributes>
AS HEAD [, SUPPORT] [, CONFIDENCE]
FROM <Table> [WHERE <Where clause>]
GROUP BY <Attributes> [HAVING <Having clause>]
[CLUSTER] BY <Attributes>
[HAVING <Having clause>]]
EXTRACTING RULES WITH
SUPPORT : <real>, Confidence <real>
```

DMQL Syntax

Use database <databasename>
Use hierarchy <hierarchyname>

<DMQL> ::= <DMQL_statement>; {<DMQL_Stmt>}

<DMQL_Statement> ::= <Data_Mining_statement>
<Concept_Hierarchy_Definition_Statement>
<Visualization_and_presentation>

<Data_Mining_Statement> ::=

Use database <database name> use datawarehouse <datawarehouse_name>
{use hierarchy <hierarchy name> for <attribute_or_dimension>}
{Mine_knowledge_specification>
in relevance to <attribute_or_dimension_list>
from <relation(s)|cube>
[where <condition>]
[order by <order list>]
[group by <grouping_list>]
[having <condition>]
{with [<interest_measure_name>] threshold = <threshold_value> [for <attribute(s)>]}

<Mine_knowledge_Specification> ::= <Mine_Char> | <Mine_Discr> | <Mine_Assoc> | <Mine_class> | <Mine_pred>

~~Which~~ all above are given before

The design of effective DMQL requires a deep understanding of power, limitation and underlying mechanism of various kinds of DM tasks

- OLEDB for DM (Microsoft' 2000)
  - ✓ Based on OLE, OLEDB, OLEDB for OLAP
  - ✓ Integrating DBMS, data warehouse & DM
- CRISP - DM (Cross-Industry Standard process for DM)
  - ✓ providing a platform and process structure for effective data mining
  - ✓ Emphasized on deploying DM technology to solve the business problems
- OLE DB for DM is an Extension of OLEDB APIs to access database systems.

Syntax:-

```
CREATE MINING MODEL <DMM name>
( <columns definition> )
USING <algorithm>
[ ( <algorithm parameters> ) ]
```

- DMQL adopts SQL-like syntax so that it can be easily integrated with relational query language
- Syntax of DMQL is defined in an Extended BNF grammar where [ ] represents zero (or) one occurrence
  - { } — 0 (or) more occurrences
- DM Query is parsed to form an SQL query which retrieves task relevant data and then proceeds with operation.

UNIT-4

* Data Mining < descriptive DataMining
  predictive DataMining

Descriptive data mining — describes dataset in concise & summarize manner and present interesting & general properties of data

predictive Data Mining — Analyzes data inorder to construct one (or) set of models and attempts to predict the behaviour of data sets

* Databases usually store large amount of data and users would like to view the data in concise & summarized (granular) descriptive terms. These descriptions provide overall picture of data, and can

* Users like the ease and flexibility of having datasets described at diff levels of granularity and from different angles. such descriptive datamining is called Conceptdescription

CONCEPT DESCRIPTION:- (CD)

* It is simplest kind of descriptive datamining.

Concept — collection of data such as frequent_buyers, graduate students & so on

CD generates descriptions for characterization & Comparison of data

* CD sometimes called as class description where concept to be described refers to class of objects.

Characterization — provides concise & summarization of given collection of data.

Discrimination — provides description comparing 2 (or more) collection of data

CD involve both characterization and discrimination

Benchmark

* CD is closely related with generalization. ie when there is large amount of data we are able to describe data in concise and generalized rather than at low levels of abstraction.
* If we allow data to be generalized at multiple levels of abstraction we can examine general behaviour of data

Example

Instead of viewing con examining individual customers. Sales manager may prefer to view data generalized to higher levels such as customer groups according to geographic regions frequency of purchase per groups and so on

multilevel data generalization = Multidimensional data analysis in DW

* The differences b/w CD in large database & OLAP involve the following
  - Complex datatypes and Aggregation
  - user control versus Automation

Complex datatypes & Aggregation :-

* DW & OLAP tools are based on Multidimensional data model that view data in form of data cube which consist of dimensions & measures. Datatypes are measures are restricted to some extent, Many current OLAP systems confine dimensions to nonnumeric data. Measures such as Count(), sum(), Average() apply only to numeric data
* CD database attributes are of various types such as Numeric, nonnumeric, spatial, text con images
* Aggregation of attributes in database include sophisticated datatypes.
* CD handles complex datatypes of their attributes & aggregation as necessary

User Control versus Automation

* OLAP in DW is user controlled process ie all operations are directed & controlled by users. Although it is user friendly it requires good understanding of role of each dimension.
* CD strives for more automated process which helps to determine which dimensions should be included in analysis, degree to which

*. If a data set should be generalized in order to produce summarization of data

(2)

DATA GENARALIZATION AND SUMMARIZATION — Based CHARACTERIZATION

The data & objects in database contains detailed int at primitive levels. ie for Example item relation in a Sales database contain item ID, brand, Category, place-made price & so on. So we have to summarize large set of data and present at higher conceptual levels.

Data generalization is a process that abstracts large set of task-relevant data in database from low conceptual level to higher Conceptual levels.

Data generalization approach include

a) Data cube — based Data aggregation

b) Attribute Oriented Induction.

(a) Data cube based on aggregation approach was discussed in 2nd unit.

(b) Attribute - Oriented Induction for Data characterization

It is to first Collect the task-relevant data using a database Query and then perform generalization based on no of distinct values of each attribute in relevant set of data. Generalization is perform by either [Attribute removal (or) Attribute generalization

* Attribute oriented Induction approach is Query Oriented, generalization based, on-line data analysis technique.

* The essential Operation of Attribute Oriented Induction is data generalization.

Attribute Removal:

If there is large set of distinct values for an attribute in initial working relation, either

(a) There is no generalization operator on attribute (i.e there is no concept hierarchy defined for attribute)

(b) Its higher level concepts are expressed interms of other attributes then attribute should be removed from working relation.

An attribute value pair represents a conjunct in a generalized tuple (or) rule. The removal of conjunct eliminates a constraint and thus generalizes the rule

In case (a) — If there is no generalization operator on attribute should be removed ∵ it can't be generalized and has large no of disjuncts

Attribute Generalization:-

If there is large set of distinct values for an attribute in initial working relation and there exist set of generalization operators on attribute, then generalization operator should be selected and applied it to attribute

Use of generalized tuple operator to generalize an attribute within tuple will cover more of original data tuples. This corresponds to rule known as climbing generalization trees in learning from examples (or) concept tree ascension

The control of how high an attribute should be generalized is typically quite subjective. Control of this process known as Attribute generalization control,

If attr is generalized too high it is over generalized and rules may not be very informative

* If attribute is not generalized to sufficiently high level then it is undergeneralized.

So some balance should be attained in Attribute Oriented Induction. There are mainly 2 ways to control a generalization process. They are
- attribute generalization threshold control
- generalized relation threshold control

Attribute generalization threshold Control :-

* We can set one generalization threshold for all attributes (or) set one threshold for each attribute
* If no of distinct values of attribute is greater than attribute threshold further attribute removal (or) attribute generalization should be performed.
* DM system have typically default threshold value b/n 2 to 8. We can modify these threshold values

If we want to further generalize relation we can reduce the threshold (roll up along attribute)
If we feel generalization reached too high we can increase the threshold (drill down along attribute)

Generalized relation threshold control :-

* This technique sets threshold for generalized relation. If
* If no of tuples in generalized relation is greater than threshold then further generalization should be performed.
* Default threshold value is b/n 10 to 30. Also it may be adjustable.

These two techniques are applied in sequence. ie 1st apply Attribute threshold control technique to generalize each attribute and then apply relation threshold control to further reduce size of generalized relation.

- Many users are interested in statistical i/f at d/f levels ; we need to accumulate (data) at count & other aggregated values. For each tuple we have count value and an initial working relation it is set to '1'.

Now we perform Attribute removal (or generalization and then Merge all identical tuples. This group count value is equal to sum of all tuples merged

Example of Attribute oriented Induction :-

Let us take initial working relation : a collection of task relevant data

name	Gender	major	birth-place	birthdate	residence	phone #	GPA
X	M	CS	Vancouver	8-12-76	AAAA-	687-4598	3.67
Y	M	CS	Newyork	28-2-75	BBBBB	253-9106	3.76
Z	F	physics	Seattle	21-3-75	CCCCC	420-5232	3.83
:	:	:	:	:	:	:	:

- for each attribute generalization proceeds as follows

1) name → It has large no of distinct values and there is no generalization operator defined for it so this attribute is removed.

2) Gender → It has only 2 distinct values and so it is retained and no generalization is performed.

3) Major → let us think major is generalized to values of {arts & science, engineering, business} and let threshold is set to 5 and there are more than 20 distinct values for major in initial working relation.
major is generalized by climbing up concept hierarchy

Birth-place → It has large no of distinct value so we need to generalize it. let us think concept hierarchy exist s city < province (or state) < country.
If no of distinct values for country is > attribute generalization threshold then attribute should be remove

If no. of distinct values for country is (4) less than threshold then birth-place can be generalized to country.

5) Birth-date → Suppose hierarchy exists that can generalize birth-date to age and age to age-range If no of age range is interval is too small w.r.t to threshold then generalization should take place

6) Residence → Residence is defined by no., street, city, residn state, residence country. No of distinct value for street, no are high since they lie at very low level ∴ attribute no, street should be removed ∴ residence is generalized to residence-city which has fewer distinct values

7) phone # → It has too many distinct values and therefore removed in generalization

8) gpa → Suppose concept hierarchy exist for gpa that groups values for grade point average into numerical intervals like {3.75–4.0, 3.5–3.75, ...} — these are grouped into descriptive values such as {excellent, very good}. The attribute can be generalized.

∴ Generalized relation is.

gender	major	birth-country	age-range	residence-city	gpa	count
M	Science	Canada	20–25	Richmond	very-good	16
F	Science	Foreign	25–30	Burnaby	excellent	22
:	:	:	:	:	:	:

Identical tuples are grouped into one and count is accumulated.

Efficient Implementation of Attribute Oriented Induction :-

Algorithm : Attribute Oriented Induction :- Mining generalized characteristics in a relational database given user's DM request.

Input :-

- ✓ DB — relational database
- ✓ DMQuery — a data mining query
- ✓ a_list — a list of attributes
- ✓ Gen(ai) — set of concept hierarchies for generalization operation on attribute a_i.
- ✓ a-gen_threshold(a_i) — attribute generalization threshold for each a_i

Output :-

P — a prime generalized relation.

Method :-

Step1 : W ← get_task_relevant_data (DMQuery, DB)

Let W be working relation that hold task relevant data.

Step2 : prepare for generalization(W)

(a) Scan W and collect distinct values for each attribute a_i. If W is large take sample of W

(b) For each attribute a_i determine whether a_i should be removed and if not compute min desired level L_i based on given threshold and determine mapping pair (v, v')

v — distinct value of a_i in W

v' — corresponding generalized value at level L_i

Step 3 : P ← generalization (W)

P is derived by replacing each value v in W by v' in mapping while accumulating count & calculating any other aggregated values

(a) For each generalized tuple, insert the tuple into sorted prime relation P by binary search — if tuple is already in P simply increase count and other aggregate value else insert into P

(b) Mostly no of distinct values at prime relation level is small so it can be coded as n-dimensional array where n is no of attributes in p. and each dimension has generalized values.

Efficiency of algorithm is as follows:-

Step 1 — It is a query to collect data relevant data into w. Its processing efficiency depends on query processing methods used

Step 2 — count statistics on w. it requires scanning w atmost once. cost for entire step is dependend on no of distinct values for each attribute

Step 3 — There are N1 tuples in w and P tuples in p

for each tuple t in w we substitute value based on mapping rules This results in generalized tuple t'

- If variation (a) is adopted each t' take O(log p) to find location for count increment. ∴ total time complexity is O(N × log p) for all generalized tuples.
- If variation (b) is adopted each t' take O(1) to find tuple for count increment. ∴ overall time complexity is O(N) for all generalized tuples

Presentation of Derived Generalization:

- Generalized descriptions are most commonly displayed in form of generalized relation (or table).
- Descriptions can be visualized in form of cross tabs (or) cross-tabulations.

In cross tab
each row — value from another attribute
column — value from another attribute

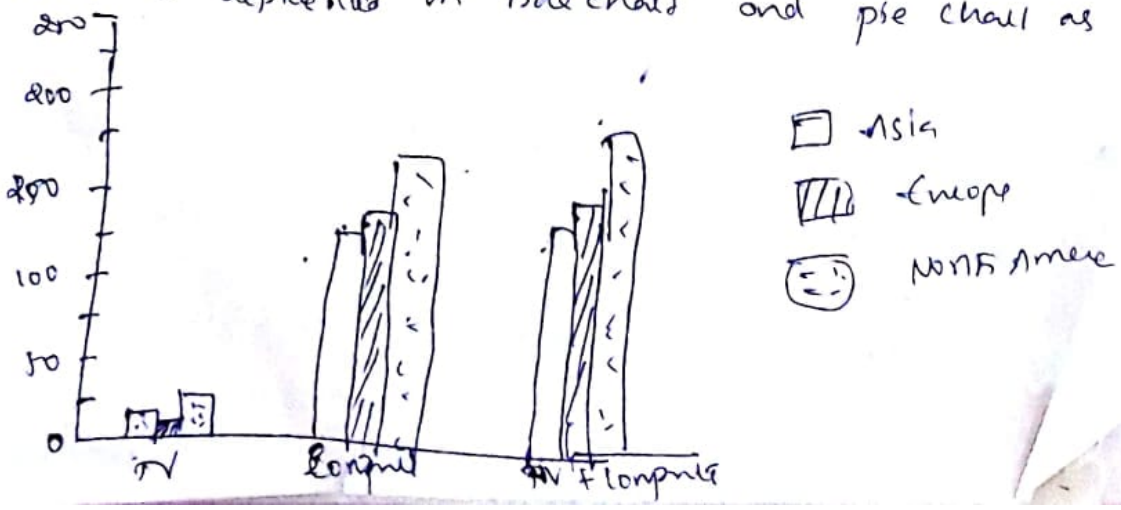
location	item	sales (in million dollars)	cost (in thousands)
Asia	TV	15	900
Europe	TV	12	250
North America	TV	28	450
Asia	computer	120	1000
Europe	Computer	150	1200
North America	computer	200	1800

A Crosstab for sales in 2004

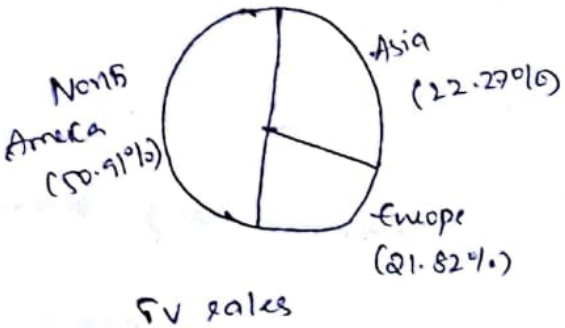
location	item					
	TV		computer		both_items	
	Sales	cost	Sales	cost	sales	cost
Asia	15	900	120	1000	135	1300
Europe	12	250	150	1200	162	1450
North America	28	450	200	1800	228	2250
all regions	45	1000	470	4000	525	5000

* Generalized data can be presented graphically using bar charts, piecharts and curves. Visualization with graphs is popular in data analysis. Such graphs can represent 2D (or 3D data.

The above table is represented in bar chart and pie chart as follow



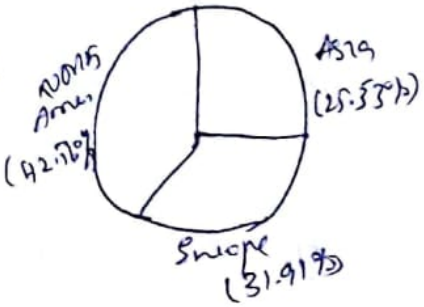
Pie-Chart



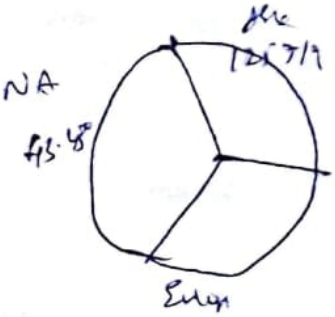
TV sales

(5)

Computer sale



TV + Computer sale



Cube View

The Cube is shown for the dimensions item, location and cost.

The size of cell represents the count of corresponding cell the brightness of cell can be used to represent another measure of the cell such as sum(sales).

* pivoting, drilling, slicing operations can be performed on the cube browser by mouse clicking.

Quantitative Rule :-

* Data in a large database usually span a diverse range of distributions a single generalization tuple is unlikely to cover (or represent 100% of initial working relation tuples or cells.

* Quantitative information such as percentage of data tuples that satisfy left and right hand side of rule should be associated with each rule.

* A logic rule that is associated with Quantitative information is called a Quantitative rule.

$$t\_weight = count(v_a) \Big/ \sum_{i=1}^{n} count(v_a)$$

where

t-weight as each tuple in corresponding generalized relation

v_a be the generalized tuple describing target class

n — no of tuples for target class in the generalized relation.

The range for t-weight is [0.0, 1.0] or [0%, 100%]

Quantitative Characteristic rule

(1) In logic form by associating corresponding t-weight value with each disjunct covering target class.

(2) In relational table or Crosstab form by changing the count values in these tables for tuples of the target class to corresponding t-weight values.

* each disjunct of rule represents a condition. The disjunction of these conditions form a necessary condition of target class.

* All tuples of target class must satisfy the condition.

Rule should be Expressed in the form

$$\forall X, target\_class(X) \Rightarrow condition_1(X)\ [t:w_1] \vee \ldots \vee condition_m(X)\ [t:w_m]$$

X — target class

w_i — t-weight value for condition

i is (1 ... m)

nec

* The Crosstab shown in the table can be transformed into logic rule form.

$\forall X$, item(X) = "Computer" $\Rightarrow$

(location(X) = "Asia")[t: 25.00%] $\vee$ (location(X) = "Europe") [t: 30.00%] $\vee$ (location(X) = "North America")[t: 45.00%]

First t-weight value of 25.00% is obtained by 1000 - the value of count slot for "(Asia, Computer)" divided by 4000 (total no of items sold)

ANALYTICAL CHARACTERIZATION:-

Analysis of Attribute Relevance:-

* We may not be sure enough that which attribute should be included for class characterization and comparison. We may specify too many attributes and slow down the system.

Measures of Attribute Relevance can be used to identify weak attributes and can be excluded from the concept description process.

* The incorporation of preprocessing step into class characterization or comparison is referred to as the Analytical characterization

Why perform Attribute Relevance Analysis:-

The first limitation of class characterization for multi-dimensional data analysis in DW and OLAP is handling of complex objects.

The second limitation is lack of automated generalization process → the user must tell system which dimensions should be included in class characterization and how high each dimension should be generalized. But it is not difficult if he can set attribute threshold and specify which level a given dimension should reach using command "generalize dimension location to the country level"

* By default there is threshold value b/w 2 to 8. If necessary we can do roll up and drill down operations
Selection of attributes should not be too high and too low. If it is too high-low ie if we include few attributes it will be incomplete.

Analytical Characterization : An Example

If the mined concept description involve many attributes Analytical characterization should be performed.
This procedure first removes irrelevant or weakly relevant attributes prior to performing generalization
Let us Examine Example

Example :- Mining general characteristics describing graduate students using Analytical characterization.

Given attributes Name, birth-place
Gender, birth-date
major, phone, gpa

Step 1 :- Target class data are collected which consist of set of graduate students
Contrasting class data is also required to perform Relevance analysis. ie Let us take as set of undergraduate students

Step 2 :- Relevance Analysis is performed via by Attribute oriented Induction
Name & are removed ∵ the no of distinct values phone # Exceeds their threshold values.
Concept hierarchies are used to generalize birth-place to birth-country and birth-date to age-range.
Major gpa are generalized to still higher levels
∴ Attributes remaining are gender
major
birth country
age range
gpa.

Candidate relation obtained for Analytical characterization : Target class (graduate students) :-

gender	major	birth_country	age_range	gpa	count
M	Science	Canada	21...25	very good	16
F	Science	Foreign	26...30	Excellent	22
M	Engineering	Foreign	26...30	Excellent	18
F	Science	Foreign	26...30	Excellent	25
M	Science	Canada	21...25	Excellent	21
F	Engineering	Canada	21...25	Excellent	18

Candidate relation Obtained for Analytical characterization : Contrasting class undergraduate students :-

gender	major	birth_country	age_range	gpa	count
M	Science	Foreign	<= 20	Very good	18
F	Business	Canada	<= 20	Fair	20
M	Business	Canada	<= 20	Fair	22
F	Science	Canada	21...25	Fair	24
M	Engineering	Foreign	21...25	Very good	22
F	Engineering	Canada	<= 20	Excellent	24

Step 3 :- Attributes are evaluated using relevance analysis measure such as Information gain

C1 be a class corresponding to graduate — 120 samples

C2 correspond to class undergraduate — 130 samples

$$I(S_1, S_2) = I(120, 130) = -\frac{120}{250}\log_2\frac{120}{250} - \frac{130}{250}\log_2\frac{130}{250}$$

$$= 0.9988$$

Next we need to compute entropy of each attribute let us do for major. ie we need to look at the distribution of graduate and undergraduate students for each value of major.

Let us calculate Expected information for each distribution

For major = "Science"

$S_{11} = 84$ $\quad I(S_{11}, S_{21}) = 0.9182$
$S_{21} = 42$

For major = "engineering"

$S_{12} = 36$
$S_{22} = 46$ $\quad I(S_{12}, S_{22}) = 0.9892$

For major = "Business"

$S_{13} = 0$ $\quad I(S_{13}, S_{23}) = 0$
$S_{23} = 42$

Expected information needed to classify a given sample

$$E(\text{major}) = \frac{126}{250} I(S_{11}, S_{21}) + \frac{82}{250} I(S_{12}, S_{22}) + \frac{42}{250} I(S_{13}, S_{23})$$

$$= 0.7873$$

∴ Gain is

$$\text{Gain (major)} = I(S_1, S_2) - E(\text{major})$$
$$= 0.2115$$

Similarly we have to calculate gain for all attributes. Then the values are sorted in increasing order.

1. gender -0.0003
2. birthcountry -0.0407
3. major -0.2115
4. gpa -0.4890
5. age range -0.5971

* Let Relevance threshold $= 0.1$ to identify weakly relevant attributes

Gain of gender, birthcountry are $<$ threshold. So they are removed.

The contrasting class is also removed finally in the final target class working relation

candidate - . .

Step 4:- Attribute oriented Induction is applied to initial target class working relation.

Mining Class Comparison: Discriminating between Different Classes

In many applications users are not interested in a single class described can characterized they would prefer to mine description that compares can distinguish one class from other comparable class.

* Target class and Contrasting class must be comparable ie both class should have similar dimensions and attributes.

Ex 1 classes person, address, item are not comparable.

Ex 2 sales in last 3 year are comparable.

* The techniques developed can be extended to handle class Comparison across several comparable classes.

ie generalization is performed synchronously among the classes compared so that attributes in all classes to be generalized at same level of abstraction.

Class Comparison Methods and Implementations:-

General procedure is as follows

1) Data collection:- The set of relevant data in database is collected by query processing and is partitioned into target class and one con set of contrasting classes.

2) Dimension Relevance Analysis:- If there are many dimensions and analytical comparison is desired then dimension relevance analysis should be performed and only highly relevant dimensions are included in further analysis

Presentation of class comparison descriptions;

Generalized relations, Crosstabs, barchart, pie chart, Curves, Rules.

Quantitative Discriminant Rule :- The discriminative features of the target and contrasting classes of a comparison description can be described quantitatively by a Quantitative discriminant rule which associates a statistical interesting measure d-weight with each generalized tuple in description.

$$\text{d-weight} = \text{count}(a_{va} \in C_j) \Big/ \sum_{i=1}^{m} \text{count}(a_{va} \in C_i)$$

where a_{va} be generalized tuple

C_j be target class

The rule of given comparison description is written in the form

$$\forall X, \text{target-class}(X) \Leftarrow \text{condition}(X) \; [d : \text{d-weight}]$$

We can present both characterization & comparison in the same table and in same rule.

Quantitative description Rule :-

A Quantitative characteristic rule and Quantitative discriminant rule for the same class can be combined to form a Quantitative description rule for the class, which displays, t-weights and d-weights associated with corresponding characteristic and discriminant rule.

A Quantitative characteristic rule provides a necessary condition for the given target class since it presents a probability measurements for each property that can occur in target class. Such a rule is of form

$$\forall X, \text{target-class}(X) \Rightarrow \text{condition}_1(X) \; [t : w_1] \vee \ldots \vee \text{condition}_m(X)$$

where each condition represents a property of target class (the rule is of form). The rule indicates that if X is in target class, the probability that X satisfies condition is value of t-weight, w_i where i is 1..m

* Quantitative discriminant rule provides a sufficient condition for target class since it presents a quantitative measurement of properties that occur in target class versus that occur in contrasting class. Such rule is of form

$$\forall X, \text{target class}(X) \Leftarrow \text{condition}_1(X)[d:w_1] \wedge \ldots \wedge \text{condition}_m(X)[d:w_m]$$

The rule indicates if X satisfies condition$_i$, there is a probability of w_i ($d\text{-weight}_{val}$) that X is in target class where i is 1..m

* A Quantitative characteristic rule and Quantitative discriminant rule for a given class can be combined as follows to form Quantitative description rule

* For each condition we have to show both t-weight and d-weight
* A bidirectional arrow should be used between the given class and conditions.

Such a rule is of form

$$\forall X, \text{target\_class}(X) \Leftrightarrow \text{condition}_1(X)[t:w_1, d:w_1'] \theta \ldots \theta \text{condition}_m(X)[t:w_m, d:w_m']$$

where θ represents logical disjunction/conjunction

The rule indicates that for i from 1 to m, if X is in target class, there is a probability of w_i that X satisfies condition$_i$ and if X satisfies condition$_i$, there is probability of w_i' that X is in in target class.

Ex:-	TV			computer			home theater		
	count	t-weight	d-weight	count	t-weight	d-weight	count	t-weight	d-weight
Europe	80	25%	40%	240	75%	80%	320		
NA	120			580					
both region	200			800					

t-weight for (Europe, TV) — 25% $= \frac{80}{(80+240)} = \frac{80}{320} = 25\%$

d-weight for (Europe, TV) — 40% $= \frac{80}{200} = 40\%$

Mining descriptive Statistical Measures in Large

Databases :-

Users are interested to learn more data characteristics regarding both central tendency and data dispersion.

Measures of central tendency include mean, median, mode, midrange

Measures of data dispersion include - Quartiles, outliers and variance

Measuring the central tendency

* The most common and most effective numerical measure of center of Center of set of data is mean,

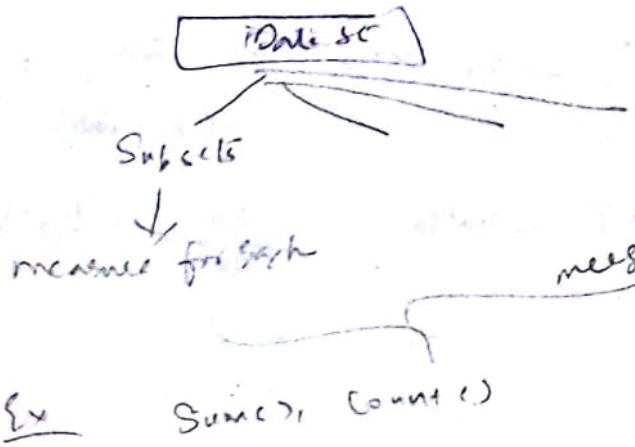
Let $x_1, x_2, \ldots x_N$ be set of N values or observations.

Mean of set of values is

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} = \frac{x_1 + x_2 + x_3 \ldots + x_N}{N}$$

This corresponds to avg() in SQL

A distributive measure is a measure that can be computed for given data by partitioning data into smaller subsets, computing measure for each subset and then merging results in order to arrive at measure's value.



Ex Sum(), count()

Algebraic measure

by applying algebraic fn to distributive measure

ex - avg() — sum() / count()

$$\bar{x} = \frac{\sum_{i=1}^{N} N_i x_i}{\sum_{i=1}^{N} N_i}$$ → weighted arithmetic mean

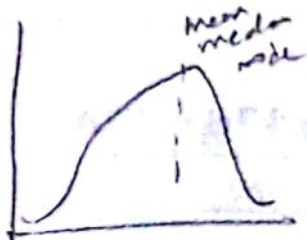
Holistic

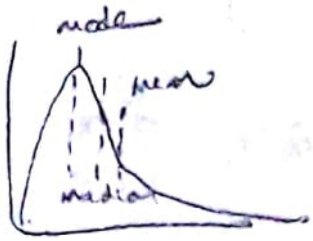
Computed on entire data set as whole

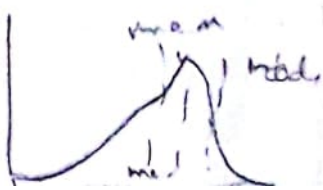
ex- $median = L_1 + \left(\frac{N/2 - (\sum freq)_l}{freq_{median}} \right) width$,

L_1 → lowerboundary of median interval

$(\sum freq)_l$ → Sum of frequencies of all intervals lower than median interval







Mode

* The mode for set of data is value that occurs most frequently in the set.
* Sometimes there occurs more than one mode : greatest frequency correspond to several values / which results in more than one mode.

$$\text{mean} - \text{mode} = 3 \times (\text{mean} - \text{median})$$

Measuring Dispersion of Data :-

The degree to which numerical data tend to spread is called the dispersion. or variance of data.

Most common measures are range, five no. summary (based on quartile) InterQuartile Range (IQR), standard deviation

* let $x_1, x_2, \ldots x_N$ be set of observations for some attribute
The range is difference b/w max() and min().
* let us assume data are sorted in the increasing numerical order

kth percentile

The k^{th} percentile of set of data in numerical order is value xi having property x^{th} percent of data entries lies at or below xi

Median is 50^{th} percentile

Other than median most commonly we use (percentiles) Quartiles.

Quartile Q_1 is first Quartile ie 25^{th} percentile

Q_3 is 75^{th} percentile.

* Quartiles, median give some shape to distribution
* Distance b/w 1st & 3rd Quartile is a simple measure that gives range covered by middle half of data, this distance is called InterQuartileRange (IQR)

$$IQR = Q_3 - Q_1$$

Candidate ...

* A common rule of thumb for identifying suspected outliers is to single out values falling at least $1.5 \times IQR$ above the 3rd Quartile (or below first Quartile.

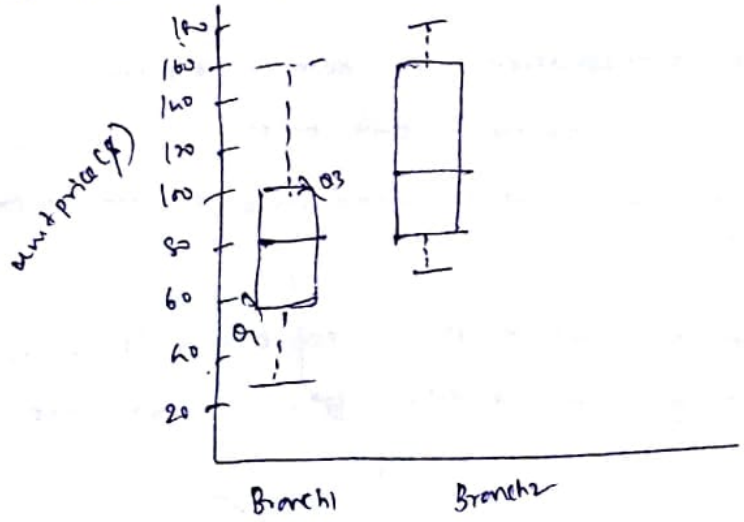
* using Q_1, Q_3, median we do not have comple information ie we don't have full shape of distribution. it is obtained by giving min & max values. This is known as five-number summary

The five-number summary consist of median, Q_1, Q_3, smallest and largest individual observation. ie Minimum, Q_1, Median, Q_3, Maximum

* Boxplots are popular form of visualizing distribution.

It consist of

* ends of box are Quartiles ie box length is equal to IQR
* Median is marked by line within box
* Two lines (whiskers) outside box extend to min & max observations



The whiskers are extended to extreme low & high observations only if values are less than $1.5 \times IQR$.

Variance and Standard deviation

Variance of N observations $x_1, x_2 \ldots x_N$ is

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2 = \frac{1}{N} \left[\sum x_i^2 - \frac{1}{N} \left(\sum x_i \right)^2 \right]$$

$\bar{x}$ = mean

* Standard deviation σ is square root of variance.

variance, standard deviation are algebraic measures ; they are computed from distributive measures.

↓

compute data by partion data set.

Graphic displays of Basic Descriptive Data Summaries

other than barchart, piechart and line graph there are popular types of graph for display of data summaries & distributions. These include Histograms

Quantile plots

q-q plots

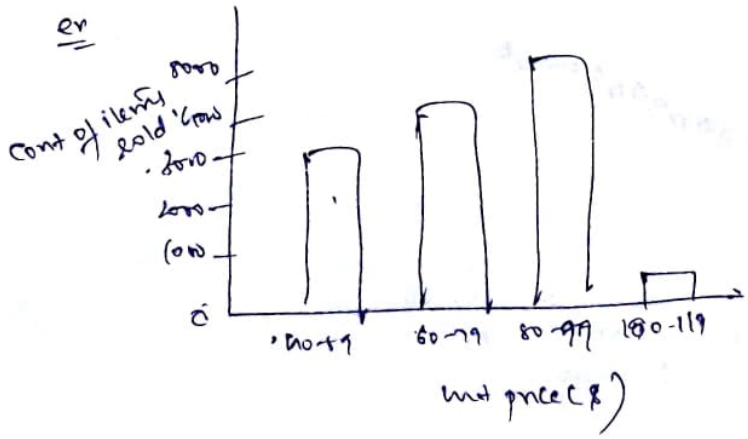
Scatter plots

loess curves.

Histogram

* It is a graphical method for summarizing distribution of given ~~data~~ attribute (A)
* It partions the distribution of A into buckets. width of each bucket is typically uniform can variable
* Each bucket is represented by rectangle whose height is equal to relative frequency of each bucket.

eg



height = Relative frequency of bucket

Quantile plot

* It is simple & effective way for univariate data distribution.
* 1st it displays all of data for given attribute.
* 2nd it plots Quantile information

i.e let x_i for $i = 1$ to N be data in increasing order.

x_1 – smallest observation
x_N – largest observation.

Each x_i is paired with f_i which indicates approximately $100 f_i\%$ of data are below or equal to value x_i

0.25 quantile – Q1
0.50 — median
0.75 — Q3

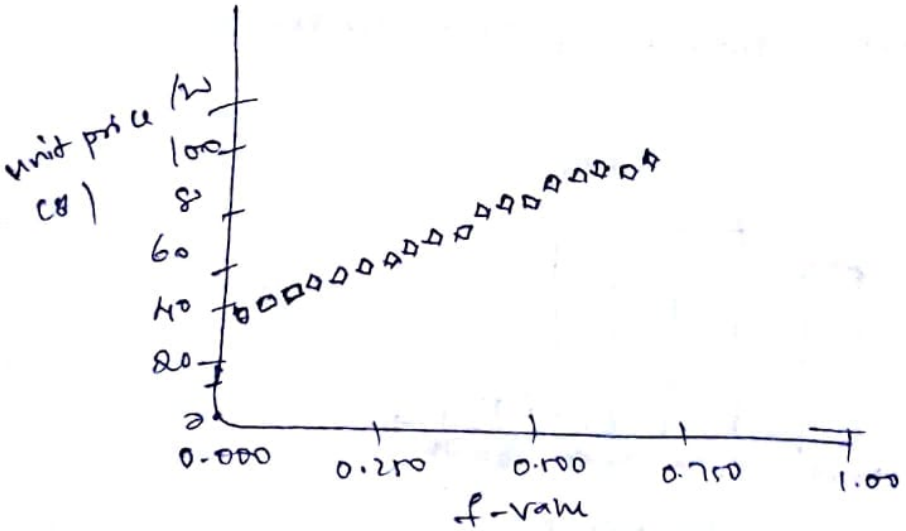
$$f_i = \frac{i - 0.5}{N}$$

Number increase in equal steps of $\frac{1}{N}$ ranging from the

$\frac{1}{2N}$ to $1 - \frac{1}{2N}$

↓ slightly above zero ↓ slightly below one

On a Quantile plot x_i is graphed against f_i



Quantile-Quantile plot (or q-q plot)

It graphs quantiles of one univariate distribution against the corresponding quantiles of another.

Let we have 2 sets of distributions for variable unit price, data from two different branch locations.

Let $x_1, x_2, \ldots x_N$ be data from first branch

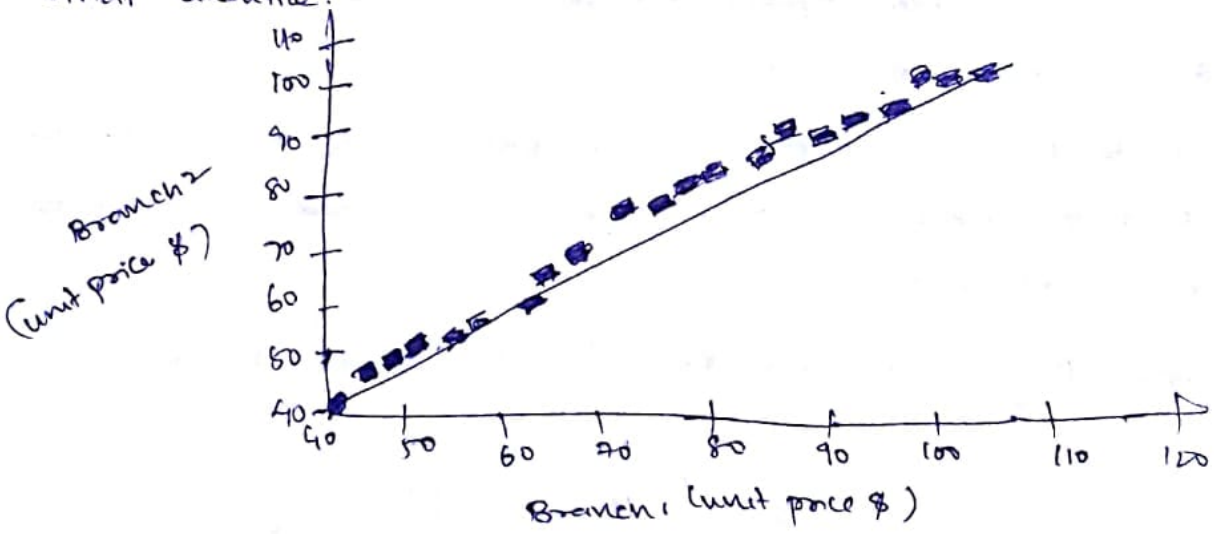
$y_1, y_2, \ldots y_M$ be data from second

$M = N$ & where each data set is sorted in increasing order.

↳ we simply plot y_i against x_i where y_i, x_i are both $\frac{i-0.5}{N}$ Quantiles of their respective data

- If $M < N$ there can be only M points on q-q plot, y_i is $\frac{i-0.5}{M}$ Quantile of y-data which is plotted against $\frac{i-0.5}{M}$ Quantile of x-data

The fig shows unit price of items sold at branch 1 verses branch 2 for that Quantile.



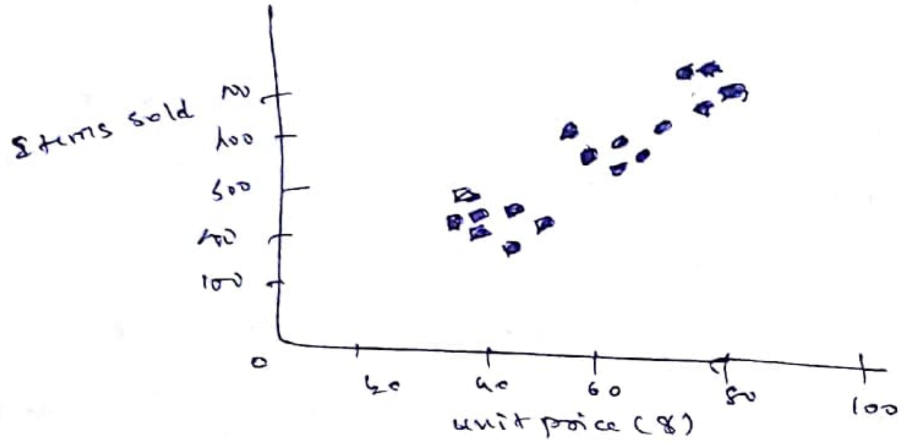
* Here lowest point in left corner corresponds to 0.03 quantile. At this quantile unit price of each of items sold at branch 1 was slightly less than at branch 2. In other words 3% of items sold at branch 1 were less than (or) equal to $40 while 3% of items at branch 2 were less than (or) equal to $42

* At the highest Quantile we see that unit price of item at branch 1

were less than cor cand to branch1.

Here there is a shift in distribution of branch1 w.r.t branch2 in unit price of items sold at branch1 tend to lower than that of branch2.

Scatter plot

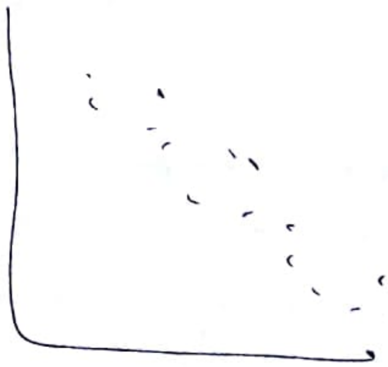


A scatter plot is one of most effective graphical methods for determining if there appear to be a relationship (or) trend b/n 2 numerical attributes.

To construct a scatter plot, each pair of values is treated as a pair of co-ordinates in algebraic sences and are plotted as points in plane.

From this figure we can see cluster of points and outliers.

positive correlation

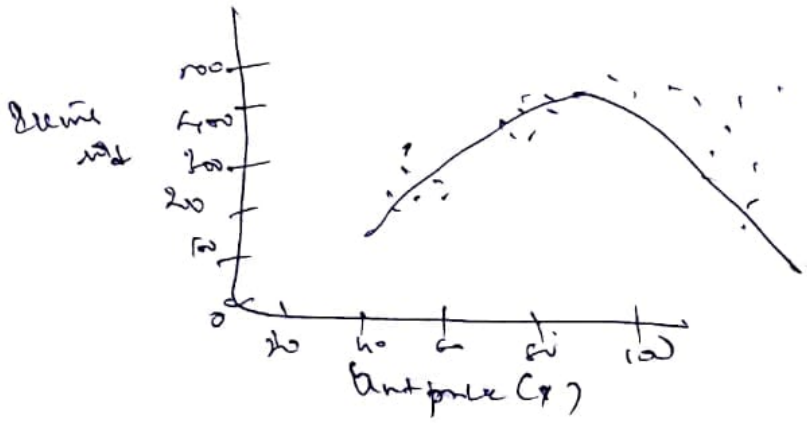


Negative correlation

Scatter plots are used to find out correlation b/n attributes

* Given n attributes a scatterplot matrix is nxn grid of (n-1) scatter plot that provides visualization of each attribute
* It becomes less effective as no of attributes grow

Loess Curve



This adds a smooth curve to a scatter plot in order to provide better perception of pattern of dependence

The word "loess" is short for "local regression"

To fit loess curve values need to be set for 2 parameters -

α — smoothy parameter

λ — degree of polynomial that are fitted by regression.

α — any +ve no. typically b/w 1/4 and 1

> can be 1 (OR) 2

Choosing α is in such a it produces smooth curve. The curve becomes smoother as α increases.

If α is very small underlying pattern is tracked.