MODULE II

DATA PREPROCESSING.

Data Preprocessing.

- Real world databases are highly susceptible to noisy, missing and inconsistent data due to their typical huge size (often several GigaBytes or more) and their origin from multiple heterogeneous sources.
- Low quality data will lead to low quality mining results.
- Data needs to be preprocessed inorder to help to improve the quality of data and the mining of results.
- There are several preprocessing techniques.

1. Data Cleaning: can be applied to remove noise and correct inconsistencies in data.

2. Data Integration: merges data from multiple sources into a coherent data source such as data warehouse.

3. Data Reduction: can reduce the size by aggregating, eliminating redundant features or by clustering.

4. Data Transformations: (eg: normalizations) may be applied, where data are scaled to fall within a smaller range like 0.0 to 1.0. This can improve the accuracy and efficiency of mining algorithms involving distance measurements.

These techniques are not mutually exclusive, they may work together. eg: Data cleaning may involve transformations to correct wrong data., such as transforming all entries for a date field to a common format.

Data has quality. if they satisfy the requirement of the intended use. The three elements of data quality are (i) Accuracy.
(ii) Completeness.
(iii) Consistency.

Inaccurate, Incomplete and Inconsistent data are commonplace properties of large real-world databases and data warehouses.

There are many possible reasons for inaccurate data (having incorrect attribute values). The data collection instruments may be faulty. There may be human or computer errors during data entry. Users may purposefully submit incorrect data values for mandatory fields when they do not wish to submit personal information. (eg. choosing default value "January 1" displayed for birthday). This is known as disguised missing data.

Errors in data transmission can also occur. There may be technology limitations such as limited buffer size for coordinating synchronized data transfer and consumption. Incorrect data may also result from inconsistencies in naming conventions or data codes

or inconsistent formats for i/p fields (eg: date). Duplicate tuples also require data cleaning.

Major Tasks in Data Preprocessing:

- The major steps involved in Data Preprocessing are.

a) Data Cleaning: Fill in missing values, smooth noisy data, identify or remove outliers and resolve inconsistencies.

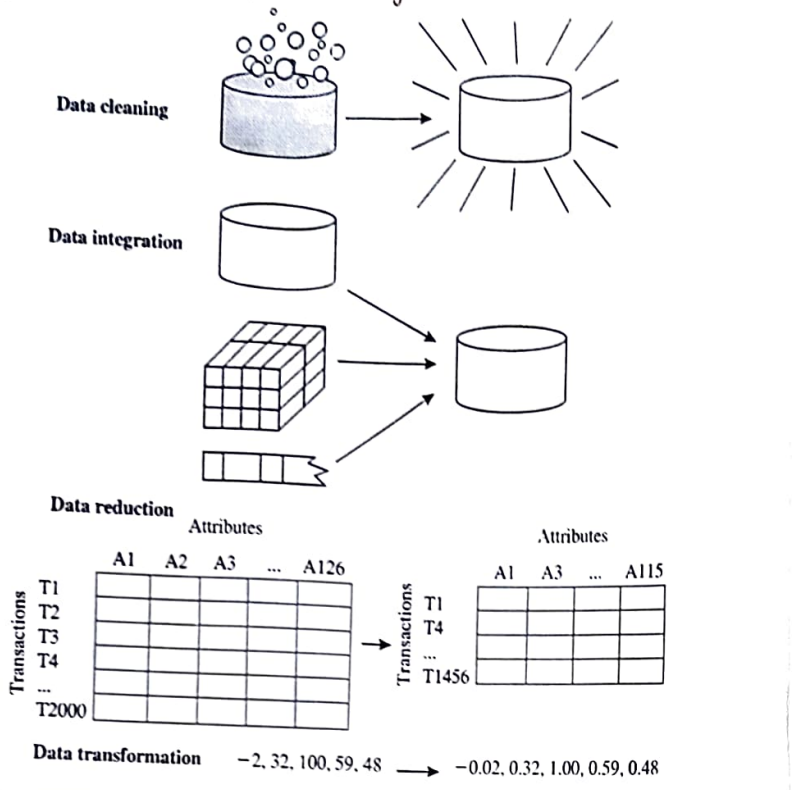
b) Data Integration: Integration of multiple database data cubes or files.

c) Data Reduction: Reduced representation of the data set ie much smaller in volume yet produces the same analytical result.

(i) Dimensionality Reduction
(ii) Numerosity
(iii) Data Compression.

d) Data Transformation and Integration: (i) Normalization
(ii) Concept hierarchy generation.

Steps in Data Preprocessing:



I Data Cleaning.

(i) Missing values
(ii) Noisy Data
(iii) Inconsistent Data.

(i) Missing Values.

Many tuples have no recorded value for several attributes.

a) **Ignore the tuple:** This is usually done when the class label is missing (if task is classification). This method is not very effective, unless the tuple contains several attributes with missing values. It is poor when the % of missing values per attribute varies considerably.

b) **Fill in missing values manually:** This approach is time consuming & may not be feasible given a large dataset with many missing values

3. Use a global constant to fill in missing values:

Replace all missing attribute values by some constant such as a label like "Unknown" or "-".

- If missing values are replaced by eg: "Unknown" then the mining program may mistakenly think that they form an interesting concept.
- Although simple this method is not recommended

4. Use the attribute mean to fill in the missing value:

Replace missing values with the mean of values of that attribute

5. Use attribute mean or median for all samples belonging to the same class as the given tuple.

6. Use the most probable value to fill in the missing value: This may be determined with regression, inference based tools using a Bayesian formalism or decision tree induction.

Methods 3 to 6 bias the data - the filled in value may not be correct. Method 6 is a popular strategy - this method uses the most information from present data to predict missing values.

(ii) Noisy Data

- Noise is a random error or variance in a measured variable.
- Data Smoothing techniques = smooth out data to remove noise.

a) Binning.

- Binning methods smooth a sorted data value ~~using~~ by consulting its "neighborhood" ie values around it.
- The sorted values are distributed into a number of "buckets" or "bins".
- Since binning methods consult the neighborhood of values, they perform local smoothing.

→ Binning techniques

- Smoothing by bin means: each bin value replaced by the bin mean
- Smoothing by bin boundaries: min & max values in a given bin are identified as bin boundaries. Each bin value replaced by the closest boundary value

- Larger the width, greater is the effect of smoothing.

eg: Sorted data for price (in dollars)

4, 8, 15, 21, 21, 24, 25, 28, 34.

Bin size = 3.

Partition into (equal-frequency) bins:

Bin 1 : 4, 8, 15

Bin 2 : 21, 21, 24

Bin 3 : 25, 28, 34.

Smoothing 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.

~~b) Clustering:~~

Q8 Suppose that the data for analysis include the attribute age. The age values for data are in the increasing order

13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

Use Smoothing by means to smooth the above data using a bin depth of 3.

Here the given data is already sorted.
Partition data into equidepth bins of depth 3.

Bin 1 : 13, 15, 16
Bin 2 : 16, 19, 20
Bin 3 : 20, 21, 22
Bin 4 : 22, 25, 25
Bin 5 : 25, 25, 30
Bin 6 : 33, 33, 35
Bin 7 : 35, 35, 35
Bin 8 : 36, 40, 45
Bin 9 : 46, 52, 70.

Calculate arithmetic mean of each bin

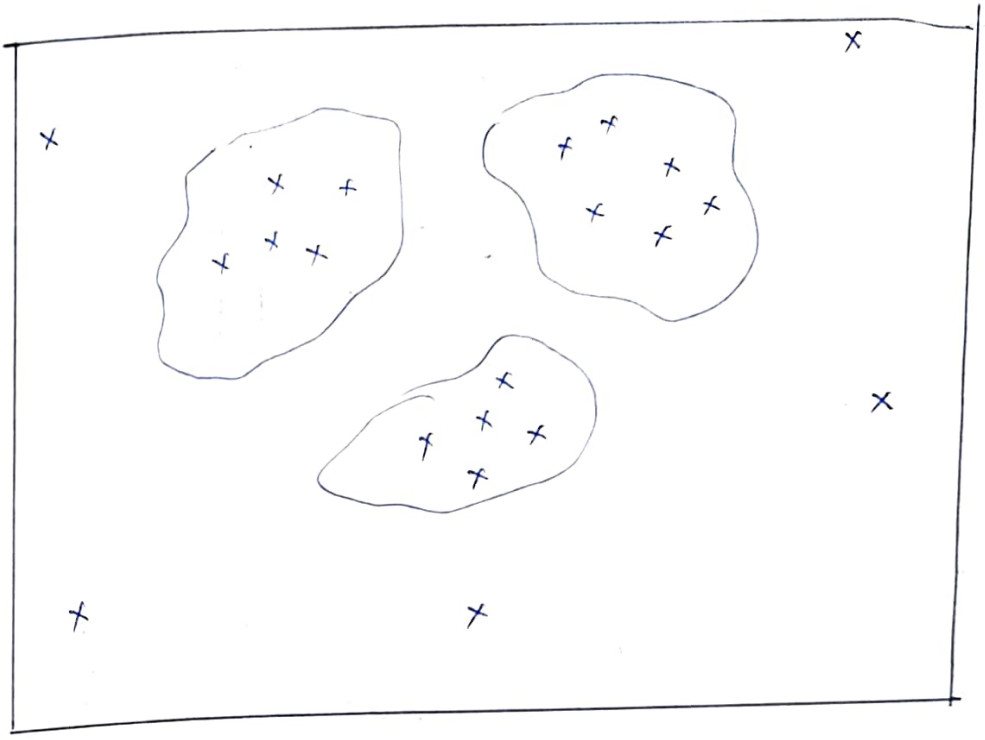
Replace each of the values in each bin by arithmetic mean calculated for the bin

Bin 1 : 14, 14, 14
Bin 2 : 18, 18, 18
Bin 3 : 21, 21, 21
Bin 4 : 24, 24, 24
Bin 5 : 26, 26, 26
Bin 6 : 33, 33, 33
Bin 7 : 35, 35, 35
Bin 8 : 40, 40, 40
Bin 9 : 56, 56, 56.

b) Clustering:

Outliers may be detected by clustering, where similar values are organized into groups or "clusters"

Intuitively, values that fall outside the set of clusters may be considered outliers.



c. Combined Computer and Human Intervention.

- Outliers may be identified through a combination of computer and human inspection.
- In one application for eg:- an information theoretic measure was used to help identify outlier patterns in a handwritten character DB for classification.
- The measure's value reflected the "surprise" content of the predicted class label w.r.t the known label.
- Outlier patterns may be informative. (eg:- identifying useful data exceptions such as different versions of the characters "0" or "7"). or "garbage" (eg:- mislabelled characters)

Patterns whose surprise content is above a threshold are output to a list. A human can then sort through the patterns in the list to identify the actual garbage ones.

d. Regression.

Data smoothing can also be done by regression, a technique that conforms data values to a function.

Linear regression: finding the "best line" to fit two attributes so that one attribute can be used to predict the other.

Multiple regression: Extension of a linear regression where more than 2 attributes are involved and the data are fit to a multi dimensional surface.

(iii) Inconsistent Data.

- There may be inconsistencies in the data recorded for some transactions.

- Some data inconsistencies may be corrected manually using external references.
- knowledge engineering tools may be used to detect the violation of known data constraints. eg: known functional dependencies b/w attributes can be used to find values contradicting the functional constraints.

II Data Integration

- combines data from multiple data sources into a coherent data store. These sources may include multiple databases, data cubes or flat files.
- Schema integration and object matching can be tricky.

Issues 1.

Entity-identification problem: How can equivalent real world entities from multiple data sources be matched up? eg: how can the data analyst/computer be sure that customer_id in one DB and cust_number in another refer to the same attribute?

MetaData can be used to help to avoid errors in schema integration. egs of metadata for each attribute include

name, meaning, data type, and range of values permitted for the attribute. and null rules for handling blank, zero or null values

Issue 2

Redundancy is another major issue in data integration.

- Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.
- Such redundancies are detected by correlation analysis. : Given two attributes, such analysis can measure how strongly one attribute implies the other based on the available data.
- For numeric attributes, we can evaluate the correlation b/w two attributes A and B by computing the correlation coefficient (Pearsons product moment coefficient).

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

(or)

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

n – no: of tuples.

a_i, b_i – respective values of A and B in tuple i.

$\bar{A}, \bar{B}$ – respective mean values of A and B

σ_A, σ_B – respective standard deviations of A and B.

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

(i) If $r_{A,B}$ is greater than 0, A and B are positively correlated, value of A increase as the value of B increase.

The higher the value, the stronger the correlation, and indicate that A (or B) may be removed as a redundancy.

(ii) if $r_{A,B} = 0$, A and B are independent and there is no correlation between them

(iii) if $r_{A,B} < 0$, then A and B are negatively correlated, where the values of one attribute increase as the values of other attribute decrease. This means that each attribute discourages the other.

Issue 3

Detection and Resolution of data value conflicts.

- For the same real world entity, attribute values from different sources may differ.
- This may be due to differences in representations, scaling or encoding.

eg: ① a weight attribute may be stored in metric units in one system and British imperial units in another.

② For a hotel chain, the prices of rooms in different cities may involve not only different currencies but also different services and taxes.

III Data Transformation.

- Data is transformed or consolidated into forms appropriate for mining. It involves the following.

a) Smoothing: which works to remove noise from data.
- binning, regression, clustering

b) Aggregation: Summary or aggregation operations are applied to data.
eg: daily sales data may be aggregated so as to compute monthly and annual amounts.

c) Generalization: Low level or Primitive (raw) data are replaced by higher level concepts through the concept of concept hierarchies.
eg: attributes such as street can be generalized to higher level concepts like city or country.

d) Attribute Construction: (Feature Construction) New attributes are constructed and added from the given set of attributes to help the mining process.
eg: we may wish to add attribute 'area' based on attributes height + width.

e) Normalization: The attribute data are scaled so as to fall within a smaller range -1.0 to 1.0 or 0.0 to 1.0

- Normalization attempts to give all attributes an equal weight.
- useful in applications like classification algorithms involving neural networks or distance measurements such as nearest neighbor classification and clustering.

Methods for Normalization

- Min-Max Normalization
- Z-score Normalization
- Normalization by decimal scaling.

(i) Min-Max Normalization.

- performs a linear transformation on the original data.
- Suppose that min_A and max_A are the minimum and maximum values of an attribute A.

Min-max normalization maps a value v_i of A to v_i' in the range $[new\_min_A, new\_max_A]$ by computing.

$$v_i' = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A.$$

- This normalization preserves the relationships among original data values.
- It will encounter an "out-of-bounds" error if a future input case for normalization falls outside of the original data range for A

eg: Suppose that minimum and maximum values for the attribute income are $12,000 and $98,000 respectively. Map a value of $73,600 to a value in the range [0.0, 1.0] using min-max normalization

$$v_i' = \frac{73600 - 12000}{98000 - 12000}(1.0 - 0.0) + 0$$

$$= 0.716$$

ii) Z-Score normalization (zero-mean normalization)

- The values of an attribute A are normalized based on the mean (ie average) and standard deviation of A.

- A value v_i of A is normalized to v_i' by computing:

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

where $\bar{A}$ and σ_A are the mean and standard deviation of attribute A.

- This method of normalization is useful when the actual minimum and maximum of attribute A are unknown or when there are outliers that dominate the min-max normalization

eg: Suppose that the mean and standard deviation for the values for attribute income are $54,000 and $16,000 respectively. using z-score normalization transform the value $73,600.

$$v' = \frac{73600 - 54000}{16000} = 1.225.$$

(iii) Normalization by decimal Scaling:

- normalizes by moving the decimal point of values of attribute A.
- The number of decimal points moved depends on the maximum absolute value of A
- A value 'v' of an attribute A is normalized to v' by computing

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

where j is the smallest integer such that

$$\max(|v_i'|) < 1$$

eg: Suppose that the recorded values of A range from −986 to 917. The maximum absolute value of A is 986. To normalize by decimal scaling, we therefore divide each value by 1000 ie' $j = 3$ so that −986 normalizes to −0.986 and 917 normalizes to 0.917

Q. Use the following methods to normalize the given set of data.

200, 300, 400, 600, 1000

1) min-max normalization by setting min=0 + max=1
2) z-score normalization
3) z-score normalization using mean absolute deviation instead of standard deviation
4) normalization by decimal scaling.

(1) min-max normalization.

$$200' = \frac{(200-200)(1-0)}{1000-200} + 0$$

$$= 0$$

$$300' = \frac{(300-200)(1-0)}{800} + 0 = \underline{0.125}$$

$$400' = \frac{(400-200)(1-0)}{800} + 0 = \underline{0.25}$$

$$600' = \frac{(600-200)(1-0)}{800} + 0 = \underline{0.50}$$

$$1000' = \frac{(1000-200)(1-0)}{800} + 0 = \underline{1.0}$$

The values after min-max normalization are (0, 0.125, 0.25, 0.50, 1.0)

ii) Z-score normalization.

$$\text{Mean} = \frac{\Sigma x_i}{n} = \frac{200+300+400+600+1000}{5}$$

$$= \frac{2500}{5} = 500$$

$$\text{Standard deviation } (\sigma) = \sqrt{\frac{\Sigma(x-\bar{x})^2}{n}}$$

$$= \sqrt{\frac{(200-500)^2+(300-500)^2+(400-500)^2+(600-500)^2+(1000-500)^2}{5}}$$

$$= 282.8$$

$$200^1 = \frac{200-500}{282.8} = -1.06$$

$$300^1 = \frac{300-500}{282.8} = -0.707$$

$$400^1 = \frac{400-500}{282.8} = -0.35$$

$$600^1 = \frac{600-500}{282.8} = 0.353$$

$$1000^1 = \frac{1000-500}{282.8} = 1.76$$

The values after Z-score normalization are

$(-1.06, -0.707, -0.35, 0.353, 1.76)$

3) Mean absolute deviation is.

$$\frac{1}{5}\left[|200-500| + |300-500| + |400-500| + |600-500| + |1000-500|\right]$$

$$= \frac{1}{5} \times 1200 = 240$$

$$\therefore 200' = \frac{200-500}{240} = -1.25$$

$$300' = \frac{300-500}{240} = -0.833$$

$$400' = \frac{400-500}{240} = -0.417$$

$$600' = \frac{600-500}{240} = 0.417$$

$$1000' = \frac{1000-500}{240} = 2.08$$

The values after normalization are

$(-1.25, -0.833, -0.417, 0.417, 2.08)$

4) The smallest integer j such that

$Max\left(\left|\frac{V_i}{10^j}\right|\right) < 1$ is 3.

∴ The values after normalization are

$(0.2, 0.3, 0.4, 0.6, 1.0)$

Q. Compute the correlation coefficient b/w X & Y.

X = 22, 25, 29, 33, 35, 40, 45, 50.

Y = 10, 14, 20, 20, 31, 35, 42, 45

Are these variables positively or negatively correlated

X	Y	$X-\bar{X}$	$Y-\bar{Y}$	$(X-\bar{X})^2$	$(Y-\bar{Y})^2$	$(X-\bar{X})(Y-\bar{Y})$
22	10	−12.88	−17.13	165.89	293.44	220.63
25	14	−9.88	−13.13	97.61	172.39	129.72
29	20	−5.88	−7.13	34.57	50.83	41.92
33	20	−1.88	−7.13	3.534	50.83	13.40
35	31	0.12	3.87	0.0144	14.97	0.464
40	35	5.12	7.87	26.21	61.93	40.29
45	42	10.12	14.87	102.41	221.11	150.48
50	45	15.12	17.87	228.6	319.34	270.194
		−0.04	−0.04	658.84	1184.84	867.098

$\bar{X} = \frac{279}{8} = 34.88$ $\quad \bar{Y} = \frac{217}{8} = 27.13$

$$\text{Correlation Coeff} = \frac{\Sigma (x-\bar{x}).(y-\bar{y})}{n.\sigma_x \sigma_y}$$

$$= \frac{\Sigma (x-\bar{x}).(y-\bar{y})}{n.\sqrt{\frac{\Sigma(x-\bar{x})^2}{n}}\sqrt{\frac{\Sigma(y-\bar{y})^2}{n}}}$$

$$= \frac{\Sigma (x-\bar{x}).(y-\bar{y})}{\sqrt{\Sigma(x-\bar{x})^2 . \Sigma(y-\bar{y})^2}}$$

$$= \frac{867.098}{\sqrt{658.84 \times 1184.84}}$$

$$= 0.9814$$

Covariance of Numeric Data.

In probability theory and statistics correlation and covariance are two similar measures for accessing how much 2 attributes are close together.

Consider 2 numeric attributes A and B

$A = \{a_1, a_2 \ldots, a_n\}$ and $B = \{b_1, b_2 \ldots, b_n\}$.

The mean values of A and B are also known as expected values of A & B

$$E(A) = \bar{A} = \sum_{i=1}^{n} \frac{a_i}{n}$$

$$E(B) = \bar{B} = \sum_{i=1}^{n} \frac{b_i}{n}$$

Covariance is defined as

$$Cov(A, B) = E\left[(A - \bar{A}).(B - \bar{B})\right]$$

$$= \sum_{i=1}^{n} \frac{(a_i - \bar{A}).(b_i - \bar{B})}{n}$$

Correlation coefficient $r_{A,B} = \dfrac{Cov(A, B)}{\sigma_A \sigma_B}$

Suppose 2 stocks A and B have the following values in one week.

(2, 5), (3, 8), (5, 10), (4, 11), (6, 14)

If the stocks are affected by the same industry ~~needs~~. trends, will their price rise or fall together.

A	B
2	5
3	8
5	10
4	11
6	14

IV 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 techniques can be applied to obtain a reduced representation of data set that is much smaller in volume, yet closely maintain the integrity of original data

- Mining on the reduced data set should be more efficient yet produce the same analytical results.

Data reduction strategies include the following.

1. Data Cube aggregation, where aggregation operations are applied to the construction of a data cube.
2. Dimension Reduction: where irrelevant, weakly relevant or redundant attributes or dimensions may be detected and removed.

3. Data compression: encoding mechanesms are used to reduce the data set size.

4. Numerosity Reduction: where original data volume are replaced by alternative, smaller forms of data representation. These techniques may be parametric or non parametric.

Parametric methods: A model is used to estimate the data, so typically only data parameters need to be stored, instead of actual data.

eg:. Regression and Log-linear models.

NonParametric Methods: store reduced representation of data

eg:. Histograms, Clustering, Sampling.

5. Discretization and Concept Hierarchy generation

- raw data values for attributes are replaced by ranges or higher conceptual levels.
- concept hierarchies allow the meneng of data at mulliple levels of abstraction and are powerful tool for data mineng.

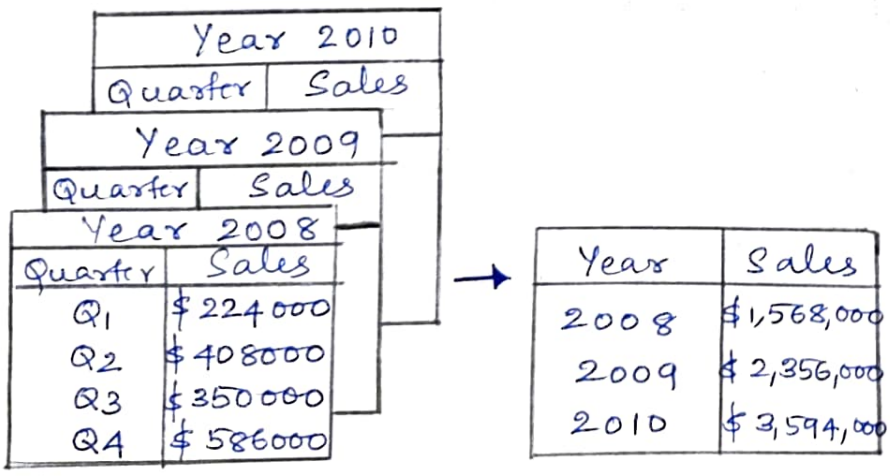
(i) Data Cube Aggregation

Imagine that data is collected for analysis. These data consist of the AllElectronics sales per quarter, for the years 2008 to 2010.

- If we are interested in the annual sales (total sales per year), rather that total per quarter.

Thus the data can be aggregated so that the resulting data summarize the total sales per year instead of per quarter.

- The resulting data is smaller in volume without loss of information necessary for analysis task.



Year 2010

Quarter	Sales

Year 2009

Quarter	Sales

Year 2008

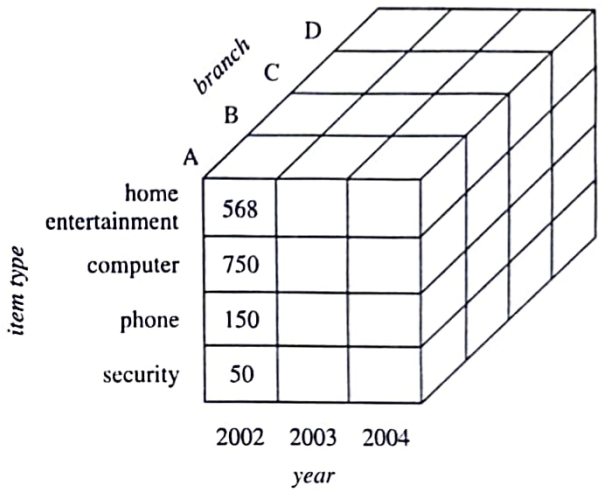
Quarter	Sales
Q1	$224000
Q2	$408000
Q3	$350000
Q4	$586000

→

Year	Sales
2008	$1,568,000
2009	$2,356,000
2010	$3,594,000

Data cubes store multidimensional, aggregated information

- The following figure shows a data cube for multidimensional analysis of sales data w.r.t annual sales per item type for each AllElectronics branch.



- Each cell holds an aggregate data value, corresponding to the datapoint in multidimensal space.

- Concept hierarchy may exist for each attribute, allowing the analysis of data at multiple abstraction levels.

eg: a hierarchy for branch could allow branches to be grouped into regions based on their address.

- Data cubes provide fast access to precomputed, summarized data. thereby benefiting online analytical processing as well as data mining.

- The cube created at the lowest abstraction level is referred to as the base cuboid. The base cuboid should correspond to an individual entity of interest such as sales or customer. — the lowest level should be usable or useful for analysis.

- A cube at the highest level of abstraction is the apex cuboid. In the figure, the apex cuboid would give one total — the total sales for all 3 years, for all item types and for all branches.

- Data cubes created for varying levels of abstraction are referred to as cuboids, data cube may be referred to as a lattice of cuboids.

Each higher abstraction level further reduces the resulting data size.

When replying to data mining requests, the smallest available cuboid relevant to the given task should be used.

(vi) Dimensionality Reduction.

- Data sets for analysis may contain hundreds of attributes, which may be irrelevant to the mining task or redundant — which can slow the mining process.
- Dimensionality reduction reduces the data size by removing such attributes or dimensions from it.
- Method of attribute subset selection: — to find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes.
- Mining on a reduced set of attributes ~~benefit~~ - reduces the number of attributes appearing in the discovered patterns, helping to make the patterns easier to understand.

Attribute Subset Selection include the following techniques.

1) Stepwise Forward Selection.

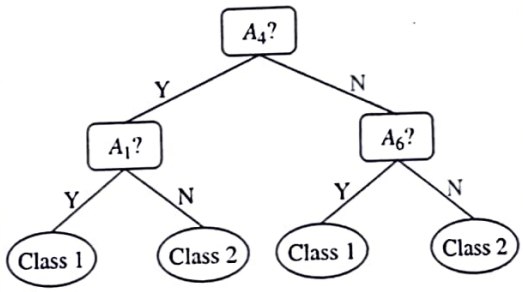
- The procedure starts with an empty set of attributes as the reduced set.
- The best of the original attributes is determined and added to the reduced set.
- At each subsequent iteration or step, the best of the remaining original attributes is added to the set.

2). Stepwise Backward ~~Selection~~ Elimination.

- The procedure starts with full set of attributes.
- At each step, it removes the worst attribute remaining in the set.

3) Combination of Forward Selection and Backward elimination:

- The stepwise forward selection and backward elimination methods are combined so that at each step, the procedure selects the best attribute and removes the worst from among the remaining attributes.

Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set: {} => $\{A_1\}$ => $\{A_1, A_4\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_4, A_5, A_6\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  => Reduced attribute set: $\{A_1, A_4, A_6\}$

4) Decision Tree Induction:

- Decision tree algorithms like ID3, C4.5 and CART are intended for classification.
- Decision tree induction constructs a flow chart like structure where each internal (non leaf) node represents a test on an attribute., each branch corresponds to an outcome of the test and each external

(leaf node) denotes a class prediction

- At each node, the algorithm chooses the "best attribute" to partition the data into individual classes.
- when decision tree induction is used for attribute subset selection, a tree is constructed from the given data.
- All attributes that do not appear in a tree are assumed to be irrelevant.

The set of attributes appearing in the tree form the reduced subset of attributes.

The stopping criteria for the methods may vary. The procedure may employ a threshold on the measure used to determine when to stop the attribute selection process.

Data Compression.

- Transformations are applied so as to obtain a reduced or compressed representation of the original data.

- Two types.

Lossless

original data reconstructed from compressed data without any info: loss.

Lossy.

we construct only an approximation of the original data.

Lossy data compression
- wavelet transform
- Principal Component Analysis.

① Wavelet Transform.

- The discrete wavelet transform (DWT) is a linear signal processing technique. when applied to a data vector X, transforms it to a numerically different vector X' of wavelet coefficients.

- The two vectors are of the same length.
- When applying this technique to data reduction, we consider that each tuple as an n-dimensional data vector. $X = \{x_1, x_2, x_3, \ldots, x_n\}$ depicting 'n' measurements made on the tuple from 'n' database attributes.

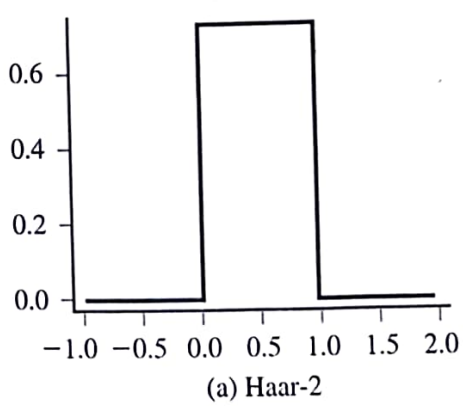
- wavelet transformed data can be truncated. A compressed approximation of data can be retained by storing only a small fraction of the strongest of the wavelet coefficients.

- eg: all wavelet coefficients larger than some user specified threshold can be retained and other coefficients set to 0.

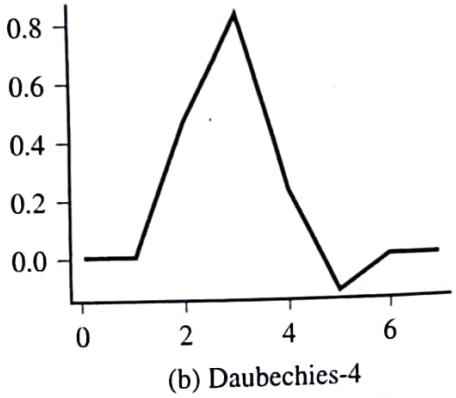
- The resulting data representation is is therefore very sparse, so that operations that can take advantage of data sparsity are computationally

very fast if performed in wavelet space.

- This technique works to remove noise without smoothing out the main features of data, making it effective for data cleaning.
- Given a set of coefficients, an approximation of original data can be constructed by applying the inverse of DWT used.
- DWT is closely related to Discrete Fourier Transform (DFT) a signal processing technique involving sines and cosines.
- DWT achieves better lossy compression and provides more accurate approximation of original data (if same no: of coefficients) (DWT vs DFT)
DWT requires less space than DFT



(a) Haar-2



(b) Daubechies-4

- Popular wavelet transforms include Haar-2, Daubechies-4, Daubechies-6 etc.

The general procedure for applying a discrete wavelet transform uses a hierarchical pyramid algorithm. that halves the data at each iteration, resulting in fast computational speed.

Hierarchical Pyramid Algorithm.

1. The length L of the input data vector must be an integer power of 2. This condition can be met by padding the data vector with zeros as necessary.
2. Each transform involves two functions. The first applies some data smoothing such as sum or weighted average. The second performs a weighted difference
3. The two functions are applied to pairs of data points in X which results in two data sets of length L/2.
4. The two functions are recursively applied to datasets obtained in the previous loop until the resulting datasets obtained are of length 2

5. Selected values from the datasets obtained in the previous operations are designated. the wavelet coefficients of the transformed data.

Equivalently, a matrix multiplication can be applied to the input data to obtain the wavelet coefficients, where the matrix used depends on the given DWT.

wavelet transformations can be applied to multi dimensional data such as a data cube.

This is done by first applying the transform to the first dimension, then to the second, and so on.

Applications of wavelet transforms.

- → compression of fingerprint images.
- → computer vision.
- → analysis of time series data.
- → Data cleaning.

A signal with 8 samples.

56, 40, 8, 24, 48, 48, 40, 16.

56	40	8	24	48	48	40	16
48	16	48	28	8	-8	0	12
32	38	16	10	8	-8	0	12
35	-3	16	10	8	-8	0	12
Use threshold of 4 35	0	16	10	8	0	0	12
if 9 35	0	16	10	0	0	0	12

The transform is invertible. We start from the bottom row. We add and subtract the difference to the mean and repeat the process up to the first row.

35	-3	16	10	8	-8	0	12
32	38	16	10	8	-8	0	12.
48	16	48	28	8	-8	0	12
56	40	8	24	48	48	40	16.

Principal Component Analysis — Refer your Machine learning Notes.

Numerosity Reduction.

- Reduce the data volume by choosing alternative smaller forms of data representation.
- Parametric or non parametric method.
- Parametric Methods.
  - → used to estimate the data.
  - → only the data parameters need to be stored instead of actual data.
  - eg: Regression analysis, Log Linear Models.
- Non Parametric Methods.
  - → for storing reduced representations of data
  - → eg: histograms, clustering, sampling.

Regression analysis and Log linear Models.

1) Linear Regression : $y = wx + b$
   - Two regression coefficients w and b specify the line and are to be

estimated using data.

- using the least squares criterion to the known values of $Y_1, Y_2 \ldots X_1, X_2 \ldots$

2) Multiple Regression:

$$Y = b_0 + b_1 X_1 + b_2 X_2$$

- many non linear functions can be transformed into the above.

Log linear Models.

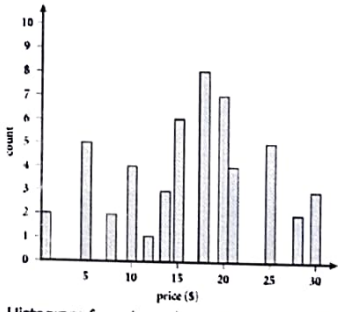
- Approximate discrete multi dimensional probability distributions
- Estimate the probability of each point (tuple) in a multidimensional space for a set of discretized attributes, based on smaller subset of dimensional combinations.
- Useful for dimension reduction and data smoothing.

1. Histograms

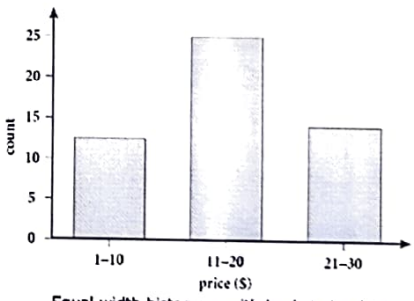
- Use binning to approximate data distributions
- Popular form of data reduction
- Histogram for an attribute , A, partitions the data distribution of A into disjoint subsets, or bucket
- Each bucket – only a single attribute-value/frequency pair – singleton buckets

Histogram Analysis - Example

- Price data:
  1,1,5,5,5,5,5,8,8,10,10,12,14,14,15,15,15,15,15,15,18,
  18,18,18,18,18,18,18,20,20,20,20,20,20,20,21,21,21,21,
  25,25,25,25,25,28,28,30,30,30



Histogram for price using singleton buckets



Equal-width histogram with bucket size $10

Histogram Analysis

- Partitioning rules:
  - Equal-width:- width of each bucket is uniform.
  - Equal-frequency (frequency- constant)
  - MaxDiff
    - Consider the difference between each pair of adjacent values
  - V-optimal
    - one with least variance
    - Histogram variance is a weighted sum of the original values that each bucket represents, where bucket weight is equal to the number of values in the bucket

2. Clustering

- Partition data set into clusters based on similarity, and store cluster representation
- The quality of a cluster may be represented by its diameter, the maximum distance between any two objects in the cluster
- Centroid distance
  - alternative measure of cluster quality
  - average distance of each cluster object from the cluster centroid
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures

3. Sampling

- Data reduction technique. Allows large data set to be represented by a much smaller random samples of the data.
- Sampling: obtaining a small sample *s* to represent the whole data set *N*
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Key principle: Choose a representative subset of the data
  - Simple random sampling may have very poor performance in the presence of skew
  - Develop adaptive sampling methods, e.g., stratified sampling:

Types of Sampling

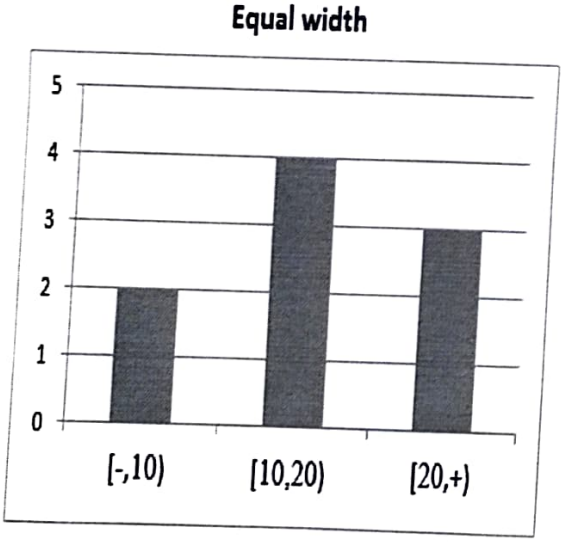
- **Simple random sampling**
  - There is an equal probability of selecting any particular item
- **Sampling without replacement**
  - Once an object is selected, it is removed from the population
- **Sampling with replacement**
  - A selected object is not removed from the population
- **Stratified sampling:**
  - Partition the data set, and draw samples from each partition (proportionally, i.e., approximately the same percentage of the data)

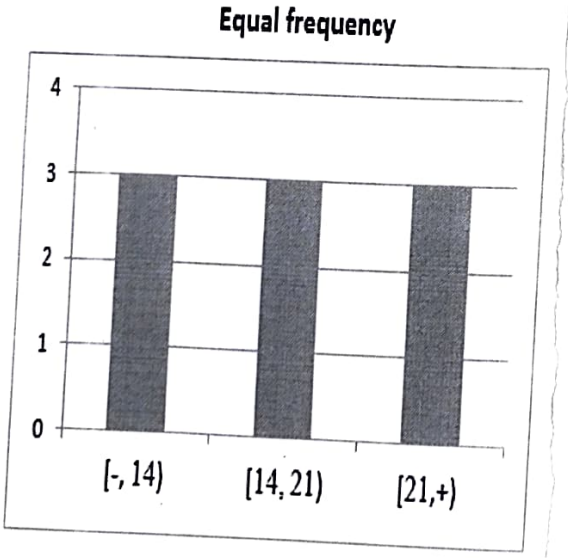
Q. Given the following set of data:
0, 4, 12, 16, 16, 18, 24, 26, 28

a) Construct a 3 bucket equal freq histogram

b) Construct a 3 bucket equal width histogram.

- **Data :** 0, 4, 12, 16, 16, 18, 24, 26, 28
- **Equal width**
  - Bin 1: 0, 4 [-,10)
  - Bin 2: 12, 16, 16, 18 [10,20)
  - Bin 3: 24, 26, 28 [20,+)
- **Equal frequency**
  - Bin 1: 0, 4, 12 [-, 14)
  - Bin 2: 16, 16, 18 [14, 21)
  - Bin 3: 24, 26, 28 [21,+)





Discretization

- Three types of attributes
  - Nominal—values from an unordered set, e.g., color, profession
  - Ordinal—values from an ordered set, e.g., military or academic rank
  - Numeric—real numbers, e.g., integer or real numbers
- Discretization: Divide the range of a continuous attribute into intervals
  - Interval labels can then be used to replace actual data values
  - Reduce data size by discretization
  - Supervised vs. unsupervised
  - Split (top-down) vs. merge (bottom-up)
  - Discretization can be performed recursively on an attribute
  - Prepare for further analysis, e.g., classification

Data Discretization Methods

- Typical methods: All the methods can be applied recursively
  - Binning
    - Top-down split, unsupervised
  - Histogram analysis
    - Top-down split, unsupervised
  - Clustering analysis (unsupervised, top-down split or bottom-up merge)
  - Decision-tree analysis (supervised, top-down split)
  - Correlation (e.g., χ^2) analysis (unsupervised, bottom-up merge)

Simple Discretization: Binning

- Equal-width (distance) partitioning
  - Divides the range into *N* intervals of equal size: uniform grid
  - if *A* and *B* are the lowest and highest values of the attribute, the width of intervals will be: $W = (B - A)/N$.
  - The most straightforward, but outliers may dominate presentation
  - Skewed data is not handled well
- Equal-depth (frequency) partitioning
  - Divides the range into *N* intervals, each containing approximately same number of samples
  - Good data scaling
  - Managing categorical attributes can be tricky

26

Binning Methods for Data Smoothing

- Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
- Partition into equal-frequency (**equi-depth**) bins:
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
- Smoothing by **bin means**:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
- Smoothing by **bin boundaries**:
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

27

Discretization by Classification & Correlation Analysis

- Classification (e.g., decision tree analysis)
  - Supervised: Given class labels, e.g., cancerous vs. benign
  - Using *entropy* to determine split point (discretization point)
  - Top-down, recursive split
- Correlation analysis (e.g., Chi-merge: χ^2-based discretization)
  - Supervised: use class information
  - Bottom-up merge: find the best neighboring intervals (those having similar distributions of classes, i.e., low χ^2 values) to merge
  - Merge performed recursively, until a predefined stopping condition

Concept Hierarchy Generation

- **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically
- usually associated with each dimension in a data warehouse
- Concept hierarchies facilitate drilling and rolling in data warehouses to view data in multiple granularity
- Concept hierarchy formation:
  - Recursively reduce the data by collecting and replacing low level concepts (such as numeric values for *age*) by higher level concepts (such as *youth*, *adult*, or *senior*)
- Concept hierarchies can be explicitly specified by domain experts and/or data warehouse designers
- Concept hierarchy can be automatically formed for both numeric and nominal data.

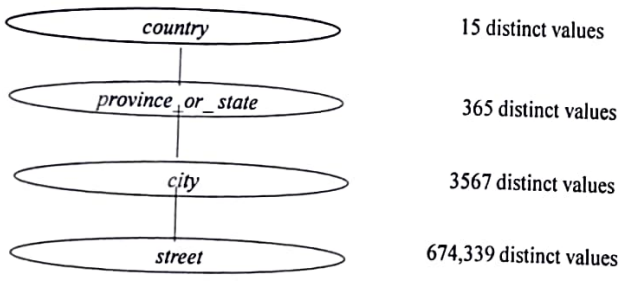
Concept Hierarchy Generation for Nominal Data

- Specification of a partial/total ordering of attributes explicitly at the schema level by users or experts
  - *street* < *city* < *state* < *country*
- Specification of a hierarchy for a set of values by explicit data grouping
  - {Urbana, Champaign, Chicago} < Illinois
- Specification of only a partial set of attributes
  - E.g., only *street* < *city*, not others
- Automatic generation of hierarchies (or attribute levels) by the analysis of the number of distinct values
  - E.g., for a set of attributes: {*street, city, state, country*}

30

Automatic Concept Hierarchy Generation

- Some hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the data set
  - The attribute with the most distinct values is placed at the lowest level of the hierarchy
  - Exceptions, e.g., weekday, month, quarter, year



31