Chapter 1: Introduction

What is Data Mining?

"The process of discovering meaningful patterns and trends often previously unknown by using some mathematical algorithm on huge amount of stored data"

"Extraction of interesting, non -trivial, implicit, previously unknown and potentially useful information or patterns from data in large database."

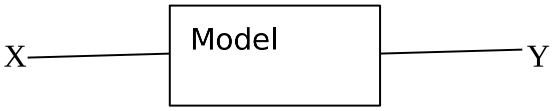
- Data mining is basically concerned with the analysis of data and the use of software techniques for finding patterns and regularities in sets of data.

Two Approaches are:

i. *Descriptive Data Mining:*
   - It characterizes the general properties of data in the database.
   - It finds patterns in data the user determinants which ones are important.
   - Mostly used during data exploration.
   - Typical questions answered by descriptive data mining are:

     . What is in the data?
     . What doesn't look like?
     . Are there any unusual patterns?
     . What does the data suggest for customer segmentation?
   - User may have no idea on which kind of patterns are interesting?
   - Functionalities of descriptive data mining are: Clustering, Summarization, Visualization, and Association.

ii. *Predictive Data Mining*:



X: Vectors of independent variables.
Y: Dependent variables
Y = f(X)

- Users don't care about the model, they simply interested in accuracy of predictions.
- Using unknown examples the model is trained and the unknown function is learned from data.
- The more data with known outcomes is available the better is the predictive power of model.

- Used to predict outcomes whose inputs are known but the output values are not realized yet.
- Never 100% accurate.
- The performance of a model on past data is not predicting the known outcomes.
- Suitable for unknown data set.
- Typical questions answered by predictive models are:
  . Who is likely to respond to next product?
  . Which customers are likely to leave in the next six months?

Data Mining Process:

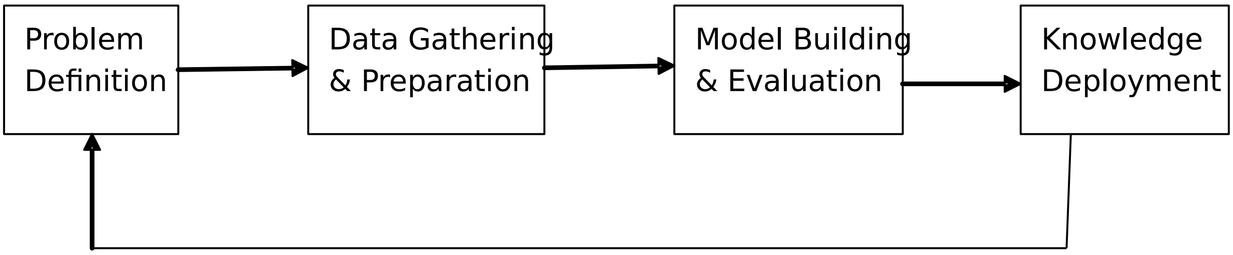


Fig: "Data mining process flow"

Problem Definition:

- Focuses on Understanding the project objectives and requirements in terms of business perspective.
  Eg: How can I sell more of my product to customer? Which customers are most likely to purchase the product?

Data Gathering and Preparation:

- Data Collection & Exploration.
- Identify data quality, patterns in data.
- Data preparation phase covers all the tasks involved to build the model.
- Data preparation tasks are likely to be performed multiple and not in any prescribed order.

Model Building and Evaluation:

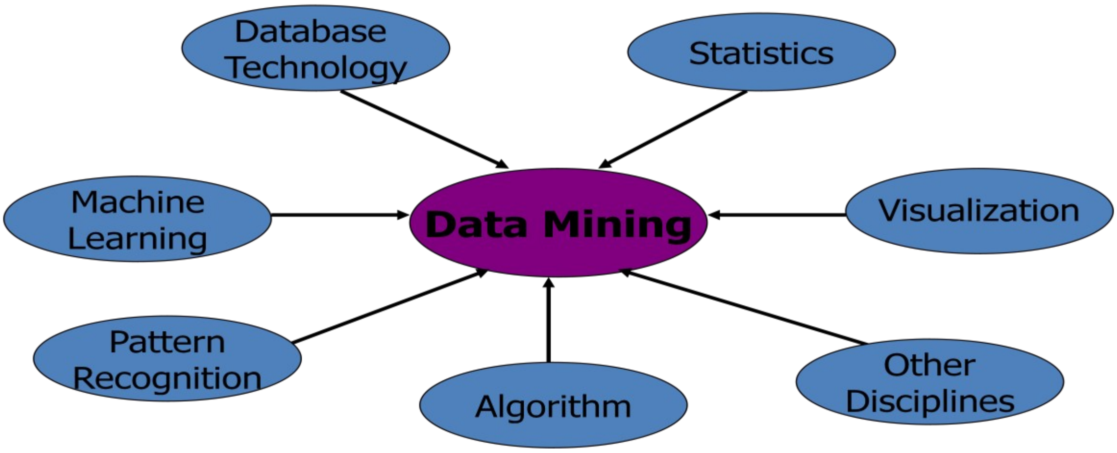
- Various modeling techniques are applied and calibrated the parameters to optimal values.
- Evaluate how well the model satisfies the originally stated business goal.

Knowledge Deployment:

- Use data mining within a target environment.
- Insight and actionable information can be derived from data.

Why Data Mining?

Data mining is a combination of multidisciplinary field. It can be applied in many fields and can be done using many algorithm and techniques.



Data Mining Vs. Query Tools

i. SQL can find normal queries from the database such as what is an average turnover? Whereas data mining tools find interesting patterns and facts such as what are the important trends in sells?
ii. Data mining is much more faster than SQL in trend and pattern analysis since it uses algorithm like machine learning, genetic algorithm.
iii. If we know exactly what we are looking for, we use SQL nut if we know only vaguely what we are looking for we use data mining.
iv. Hybrid information can't be easily be traced using SQL.

Data Warehouse

In most of the organization, there occur large databases in operation for normal daily transactions called operational database.

A data warehouse is a large database built from the operational database.
A data warehouse should be:

i. Time – dependent
   - There must be a connection between the information in the warehouse and the time when it was entered.
   - One of the most important aspect of the warehouse as it relates to data mining, because information can then be sourced according to period.

ii. Non-Volatile
- Data in a warehouse is never updated, but used only for queries.
- End-users who want to update data must use operational database.
- A data warehouse will always be filled with historical data.

iii. Subject Oriented
- Not all the information in the operational database is useful for a data warehouse.
- A data warehouse should be designed especially for decision support and expert system with specific related data.

iv. Integrated
- In an operational data, many types of information being used with different names for same entity.
- In a data warehouse, all entities should be integrated and consistent i.e. only one name must exist to describe each individual entity.

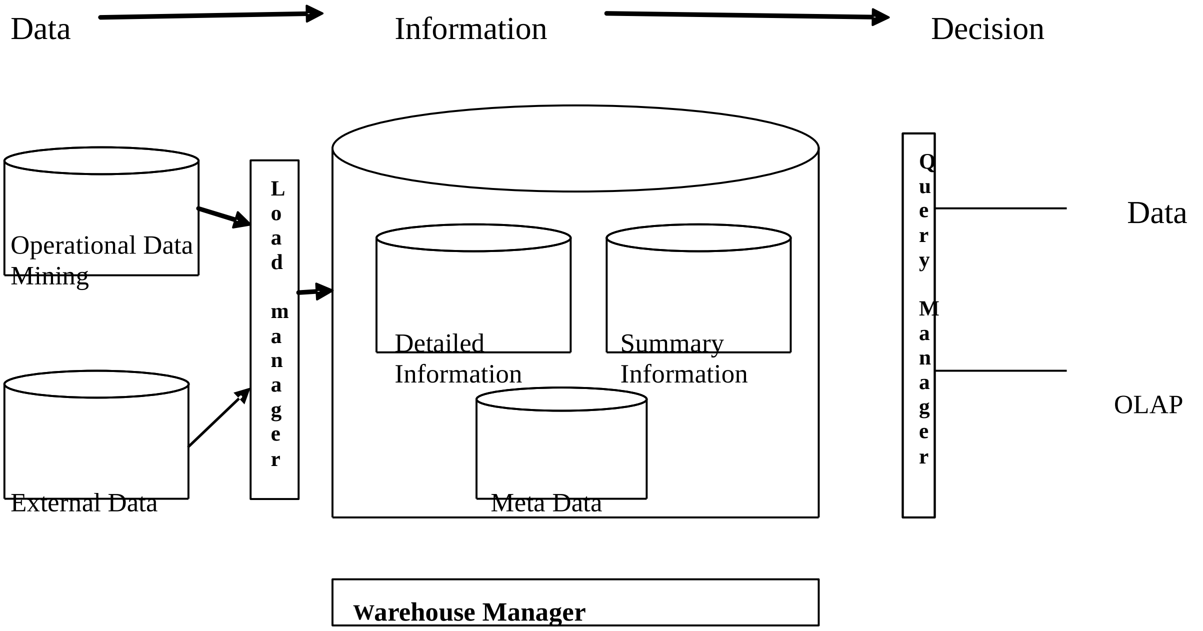


Fig: "Architecture of a Data Warehouse"

Load Manager: The system components that perform all the operations necessary to support the extract and load process. It fast loads the extracted data into a temporary data store and performs simple transformations into a structure similar to the one in the data warehouse.

Warehouse Manager: Performs all the necessary operations to support the warehouse management process. It analyzes the data to perform consistency and referential checks. It also transforms and merges the source data in the temporary data store into the published data warehouse with creating indexes and business views. Update all existing aggregations and back up data in the data warehouse.

Query Manager: Performs all the operations necessary to support the query management process by directing queries to the appropriate tables. In some cases it also stores query profiles to allow the warehouse manager to determine which indexes and aggregations are appropriate.

Detailed Information: Stores all the detailed information to determine the business requirements to analyze the level at which to retain detailed information in the data warehouse.

*Summary Informati***on:** Stores all the predefined aggregations generated by the warehouse manager. It is a transient area which will change on an ongoing basis in order to respond to changing query profiles. It is essentially a replication to detailed information.

Meta Data: Meta data is data about data which describes how information is structured within a data warehouse. It maps data stores to common view of information with the data warehouse.

Data Mart

- Data Mart is a subset of the information content of a data warehouse that is stored in its own database.
- Data mart may or may not be sourced from an enterprise data warehouse i.e. it could have been directly populated from source data.
- Data mart can improve query performance simply by reducing the volume of data that needs to be scanned to satisfy the query.
- Data marts are created along functional level to reduce the likelihood of queries requiring data outside the mart.
- Data marts may help in multiple queries or tools to access data by creating their own internal database structures.
- Eg: Departmental Store, Banking System.

Chapter 2: Data Preprocessing

What is an Attribute?

- An attribute is a property or characteristic of an object. Examples: eye color of a person, temperature, etc.
- Attribute is also known as variable, field, characteristic, or feature
- A collection of attributes describe an object. Object is also known as record, point, case, sample, entity, or instance.
- Attribute values are numbers or symbols assigned to an attribute
- Same attribute can be mapped to different attribute values. Example: height can be measured in feet or meters.
- Different attributes can be mapped to the same set of values. Example: Attribute values for ID and age are integers but properties of attribute values can be different. ID has no limit but age has a maximum and minimum value.

Types of Attributes

Approach 1:

Attribute Type	Description	Examples
Nominal	The values of a nominal attribute are just different names, i.e., nominal attributes provide only enough information to distinguish one object from another. (=,)	zip codes, employee ID numbers, eye color.
Ordinal	The values of an ordinal attribute provide enough information to order objects. (<, >)	hardness of minerals, {*good*, *better*, *best*}, grades, street numbers
Interval	For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. (+, -)	calendar dates, temperature in Celsius or Fahrenheit
Ratio	For ratio variables, both differences and ratios are meaningful. (*, /)	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current

Approach 2:

Discrete Attribute

- Has only a finite or countably infinite set of values
- Examples: zip codes, counts, or the set of words in a collection of documents
- Often represented as integer variables.
- Note: binary attributes are a special case of discrete attributes

Continuous Attribute

- Has real numbers as attribute values
- Examples: temperature, height, or weight.

- Practically, real values can only be measured and represented using a finite number of digits.
- Continuous attributes are typically represented as floating-point variables.

Approach 3:

Character: values are represented in forms of character or set of characters (string).

Number: values are represented in forms of number. Numebr may be in form of whole number, decimal number.

Types of data sets

a. Record

- Data that consists of a collection of records, each of which consists of a fixed set of attributes

i. **Data Matrix**

- If data objects have the same fixed set of numeric attributes, then the data objects can be thought of as points in a multi-dimensional space, where each dimension represents a distinct attribute
- Such data set can be represented by an m by n matrix, where there are m rows, one for each object, and n columns, one for each attribute

Projection of x Load	Projection of y load	Distance	Load	T hickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

ii. **Document Data**

- Each document becomes a `term' vector, each term is a component (attribute) of the vector, the value of each component is the number of times the corresponding term occurs in the document

	team	coach	pla y	ball	score	game	wi n	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

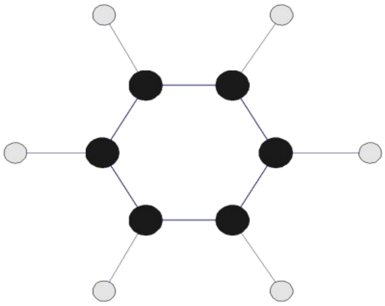
iii. **Transaction Data**

- A special type of record data, where each record (transaction) involves a set of items.
- For example, consider a grocery store. The set of products purchased by a customer during one shopping trip constitute a transaction, while the individual products that were purchased are the items

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

b. Graph

- Contains notes and connecting vertices.



Eg: World Wide Web, Molecular Structures

c. Ordered

- Has Sequences of transactions
  - i. Spatial Data

    Spatial data, also known as geospatial data, is information about a physical object that can be represented by numerical values in a geographic coordinate system.

  - ii. Temporal Data

    A temporal data denotes the evolution of an object characteristic over a period of time. Eg d=f(t).

  - iii. Sequential Data

    Data arranged in sequence.

Important Characteristics of Structured Data

a. Dimensionality

- A Data Dimension is a set of data attributes pertaining to something of interest to a business. Dimensions are things like "customers", "products", "stores" and "time".

Curse of Dimensionality

- When dimensionality increases, data becomes increasingly sparse in the space that it occupies.
- Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful

***Purpose:**

- Avoid curse of dimensionality
- Reduce amount of time and memory required by data mining algorithms
- Allow data to be more easily visualized
- May help to eliminate irrelevant features or reduce noise

***Techniques**

- Principle Component Analysis
- Singular Value Decomposition
- Others: supervised and non-linear techniques

Dimensionality Reduction:

i. PCA

- Goal is to find a projection that captures the largest amount of variation in data.
- Find the eigenvectors of the covariance matrix.
- The eigenvectors define the new space.
- Construct a neighborhoods graph
- For each pair of points in the graph, compute the shortest path distances – geodesic distances

ii. Feature Subset Selection

- Another way to reduce dimensionality of data.
- Redundant features.
  - Duplicate much or all of the information contained in one or more other attributes.
  - Example: purchase price of a product and the amount of sales tax paid.
- Irrelevant features
  - Contain no information that is useful for the data mining task at hand
  - Example: students' ID is often irrelevant to the task of predicting students' GPA

Techniques:

a. Brute-force approach:
   - Try all possible feature subsets as input to data mining algorithm
b. Embedded approaches:
   - Feature selection occurs naturally as part of the data mining algorithm
c. Filter approaches:
   - Features are selected before data mining algorithm is run
d. Wrapper approaches:
   - Use the data mining algorithm as a black box to find best subset of attributes.

Feature Creation

- Create new attributes that can capture the important information in a data set much more efficiently than the original attributes.
- Three general methodologies:
  - Feature Extraction: domain-specific
  - Mapping Data to New Space
  - Feature Construction: combining features

b. Sparsity and Density

- Sparsity and density are terms used to describe the percentage of cells in a database table that are not populated and populated, respectively. The sum of the sparsity and density should equal 100.
- Many of the cell combinations might not make sense or the data for them might be missing.
- In the relational world storage of such data is not a problem: we only keep whatever there is. If we want to keep closer to our multidimensional view of the world, we face a dilemma: either store empty space or create an index to keep track of the nonempty cells or search for an alternative solution

c. Resolution

- Scaling of data in different label and classes. Patterns depend on the scale.

Data Quality

- Real world database are highly unprotected from noise, missing and inconsistent data due to their typically huge size and their possible origin from multiple, heterogeneous sources.
- Low quality data will lead to low quality mining results.
- Data pre-processing is required to handle these above mentioned facts.
- The methods for data preprocessing are organized into
  a. Data Cleaning
  b. Data Integration
  c. Data Transformation
  d. Data Reduction
  e. Data Discritization

Data Cleaning

- Mostly concern with

i. Fill-in missing values
ii. Identify outliers and smooth out noisy data
iii. Correct inconsistent data
iv. Eliminate duplicate data

a. Missing Data

-Data is not always available because many tuples may not have recorded values for several attributes such as age, income.

- Missing data may be due to:
. Equipment Malfunction
. Inconsistent with other recorded data and thus deleted.
. Data not entered due to misunderstanding
. Certain data may not be considered important at the time of entry.
. No change in recorded data.

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing. Not effective when the percentage of missing values per attribute varies considerably.
- Fill-in missing values manually: Tedious and infeasible task.
- Use a global constant to fill-in missing values.
- Use an attribute mean fill-in missing values belonging to the same class.
- Use the most probable value to fill-in missing value.

b. Noisy Data

- Noisy data is a form of error because of random error in a measured variable.
- Incorrect attribute values may be due to:

. Faulty data collection instruments
. Data entry problem
. Data transmission problem
. Technology limitation
. Inconsistency in naming convention

How to Handle Noisy Data

- Clustering: Detect and remove outliers
- Regression: Smooth by fitting the data into regressi9on function
- Binning Method: First sort the data and partition into different boundaries with mean, median values.
- Combined computer and human inspection, doing so suspicious values are detected by human

c. Outliers

- Outliers are a set of data points that are considerably dissimilar or inconsistent with the remaining data.
- In most of the cases they are inference of noise while in some cases they may actually carry valuable information.
- Outliers can occur because of:

. Transient malfunction of data measurement
. Error in data transmission or transcription
. Changes in system behavior
. Data contamination from outside the population examined.
. Flaw in assumed theory

How to Handle Outliers

There are three fundamental approaches to the problem of outlier's detection

a. Type 1: Determine the outliers with no prior knowledge of data. This is a learning approach analogous to unsupervised learning.
b. Type 2: Model with normality and abnormality. Analogous to supervised learning.
c. Type 3: Model with normality. Semi- supervised learning approach.

Data Integration

- Combines data from multiple sources into a coherent store.
- Integrate meta data from different sources (Schema Integration)

Problem: - .Entity Identification Problem.
.Different sources have different values for same attributes.
.Data Redundancy

These problems are mainly because of different representation, different scales etc.

How to handle redundant data in data integration?

- Redundant data may be able to be detected by correlation analysis.
- Step-wise and careful integration of data from multiple sources may help to improve mining speed and quality.

Data Transformation

Changing data from one form to another form.

Approaches:

i. Smoothing: Remove noise from data.
ii. Aggregation: Summarizations of data
iii. Generalization: Hierarchy climbing of data
iv. Normalization: Scaled to fall within a small specified range.
   a. Min-Max Normalization:

   $V' = ((V\text{-}min)/(max\text{-}min)*(new\_max - new\_min)) + new\_min$

   b. Z-Score Normalization:

   $V' = (V\text{-}min)/ stand\_dev.$

   c. Normalization by decimal scaling:

   $V' = V/10^j$ where j is the smallest integer such that max (|V'|) <1

Data Aggregation:

- Combining two or more attributes (or objects) into a single attribute (or object).

Purpose

Data reduction: Reduce the number of attributes or objects

Change of scale: Cities aggregated into regions, states, countries, etc

More "stable" data: Aggregated data tends to have less variability

Data Reduction:

- Warehouse may store terabytes of data hence complex data mining may take a very long time to run on complete data set.
- Data reduction is the process of obtaining a reduced representation of data set that is much smaller in volume but yet produces the same or almost same analytical results.
- Different methods such as data sampling, dimensionality reduction, data cube, aggregation, discritization and hierarchy are used for data reduction.
- Data compression can also be used mostly in media files or data.

i. Data Sampling:

- It is one of main method for data selection i.e. sampling is the main technique employed for data selection.
- It is often used for both the preliminary investigation of the data and the final data analysis.
- Statisticians sample because obtaining the entire set of data of interest is too expensive or time consuming.
- Sampling is used in data mining because processing the entire set of data of interest is too expensive or time consuming.
- Often used for both preliminary investigation of data and the final data analysis.
- Important since obtaining entire set of data of interest is too expensive or time consuming.
- Sampling should be representative since it must represent approximately the same property as the original set of data.
- Get at least one object from each of 10 groups as sample data.

Types:

a. *Simple Random Sampling*: Equal probability of selecting any particular item.
b. *Sampling without replacement*: As each item is selected, it is removed from population.
c. *Sampling with replaceme*nt: Objects are not removed from the population as they are selected from the sample. The same objects can be picked-up more than once.
d. *Stratified Sampling*: Split the data into several partitions, then draw random samples from each partition.

ii. Dimensionality Reduction:

- Dimensionality Reduction is about converting data of very high dimensionality into data of much lower dimensionality such that each of the lower dimensions conveys much more information.
- This is typically done while solving data mining/machine learning problems to get better features for a classification or regression task.

Data Discretization:

- Convert continuous data into discrete data.
- Partition data into different classes.
- Two approaches are:

a. Equal width (distance) partitioning:

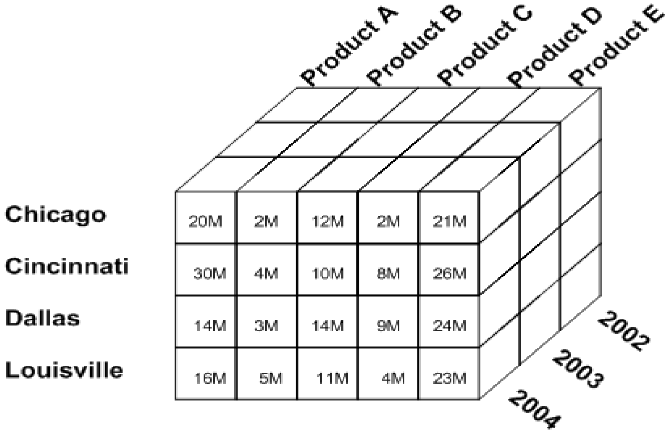
- It divides the range into N intervals of equal size.
- If A and B are the lowest and the highest values of the attribute, the width of interval will be
- $W = (A - B)/N$.
- The most straight forward approach for data discretization.

b. Equal depth (frequency) partitioning:

- It divides the range into N intervals, each containing approximately same number of samples.
- Good data scaling
- Managing categorical attributes can be tricky.

OLAP Tool

- OLAP stands for On-Line Analytical Processing.
- An OLAP cube is a data structure that allows fast analysis of data.
- OLAP tools were developed to solve multi-dimensional data analysis which stores their data in a special multi-dimensional format (data cube) with no updating facility.
- An OLAP toll doesn't learn, it creates no new knowledge and they can't reach new solutions.
- Information of multi-dimension nature can't be easily analyzed when the table has the standard 2-D representation.
- A table with n- independent attributes can be seen as an n-dimensional space.
- It is required to explore the relationships between several dimensions and standard relational databases are not very good for this.



	Product A	Product B	Product C	Product D	Product E
Chicago	20M	2M	12M	2M	21M
Cincinnati	30M	4M	10M	8M	26M
Dallas	14M	3M	14M	9M	24M
Louisville	16M	5M	11M	4M	23M

OLAP Operations:

i. **Slicing:** A slice is a subset of multi-dimensional array corresponding to a single value for one or more members of the dimensions. Eg: Product A sales.

ii. **Dicing:** Dicing operation is the slice on more than two dimensions of data cube. (More than two consecutive slice). Eg: Product A sales in 2004.

iii. **Drill-Down:** Drill-down is specific analytical technique where the user navigates among levels of data ranging from the most summarized to the most detailed i.e. it navigates from less detailed data to more detailed data. Eg: Product A sales in Chicago in 2004.

iv. **Roll-Up:** Computing of all the data relationship for more than one or more dimensions i.e. summarization of data to one o more dimensions. Eg: Total Product.

v. **Pivoting:** Pivoting is also called rotate operation. It rotates the data in order to provide an alternative presentation of data.

OLTP (Online Transaction Processing)

- Used to carry out day to day business functions such as ERP (Enterprise Resource Planning), CRM (Customer Relationship Planning)
- OLTP system solved a critical business problem of automating daily business functions and running real time report and analysis.

OLAP Vs OLTP

Facts	OLTP	OLAP
Source of Data	Operational Data	Data warehouse (From various database)
Purpose of data	Control and run fundamental business tasks	For planning, problem solving and decision support
Queries	Simple queries	Complex queries and algorithms
Processing Speed	Typically very fast	Depends on data size, techniques and algorithms
Space requirements	Can be relatively small	Larger due to aggregated databases
Database Design	Highly Normalized with many tables.	Typically denormalized with fewer tables. Use of star or snowflake schema.

Similarity and Dissimilarity

Similarity

- Numerical measure of how alike two data objects are.
- Is higher when objects are more alike.
- Often falls in the range [0,1]

Dissimilarity

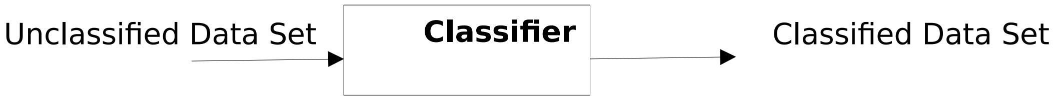
- Numerical measure of how different are two data objects
- Lower when objects are more alike
- Minimum dissimilarity is often 0
- Upper limit varies

Similarity Measure Methods:

“Refer Class Note”

Chapter-3: Classification

- Classification is a data mining technique used to predict group membership of data instances.
- Classification assigns items on a collection to target categories or classes.
- The goal of classification is to accurately predict the target class for each case in the data.



Stages in classification

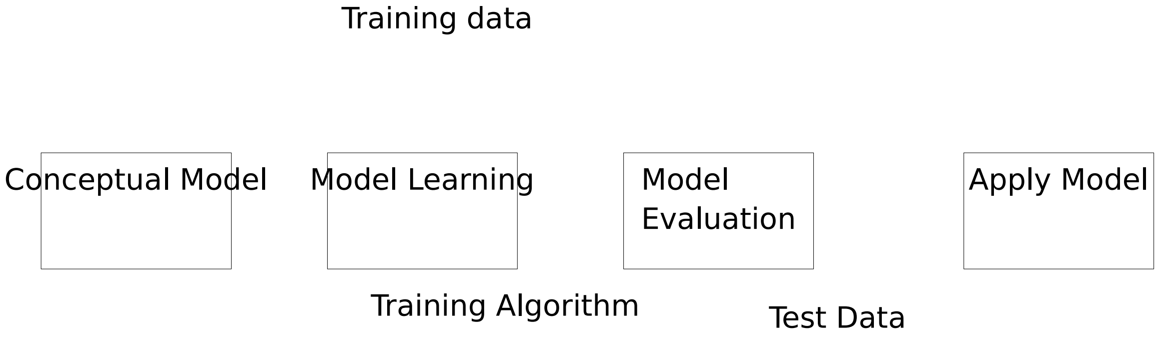


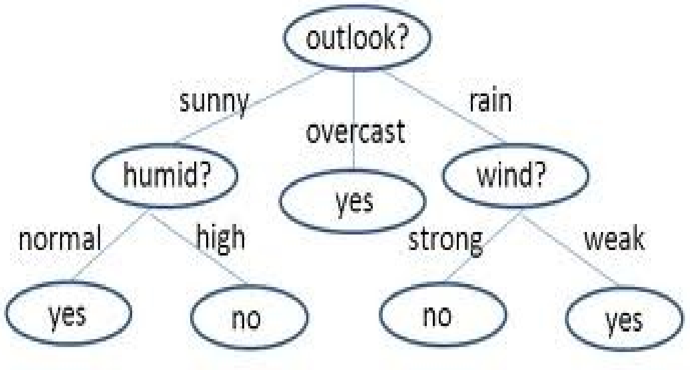
Fig: Stages in classification

Types:

i. Decision Tree classifier
ii. Rule Based Classifier
iii. Nearest Neighbor Classifier
iv. Bayesian Classifier
v. Artificial Neural Network (ANN) Classifier
vi. Others

Decision Tree classifier

- A decision tree is tree in which each branch node represents a choice between a number of alternatives and each leaf node represents a classification or decision.
- Decision tree is a classifier in the form of a tree structure where a **leaf node** indicates the class of instances, a **decision node** specifies some test to be carried out on a single attribute value with one branch and sub-tree for each possible outcome of the test.
- A decision tree can be used to classify an instance by starting at root of the tree and moving through it until leaf node. The leaf node provides the corresponding class of instance.



Decision Tree Algorithm

i. Hunt's Algorithm
ii. ID3, J48, C4.5 (Based on Entropy Calculation)
iii. SLIQ,SPRINT,CART (Based on Gini-Index)

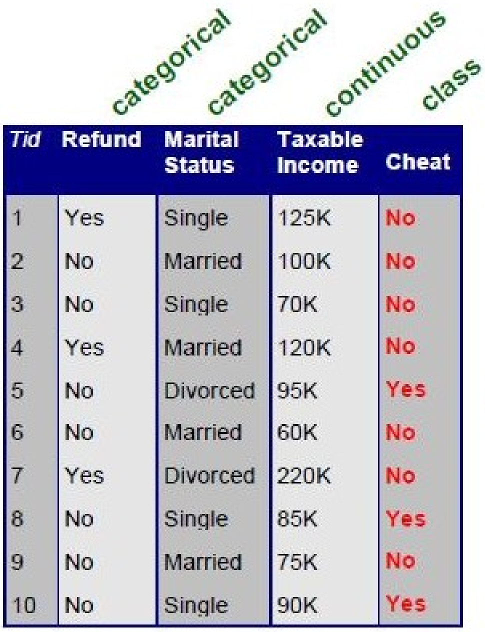
Hunt's Algorithm

- Hunt's algorithm grows a decision tree in a recursive fashion by partitioning the training data into successively into subsets.
- Let Dt be the set of training data that reach a node '**t**'. The general recursive procedure is defined as:

i. If Dt contains records that belong the same class y_t, then t is a leaf node labeled as $y_{t.}$
ii. If Dt is an empty set, then t is a leaf node labeled by the default class, y_d
iii. If Dt contains records that belong to more than one class, use an attribute test to split the data into smaller subsets.

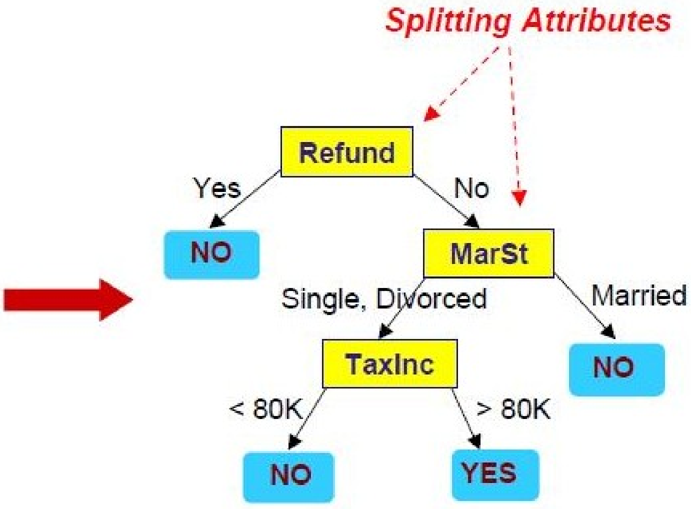
- It recursively applies the procedure to each subset until all the records in the subset belong to the same class.
- The Hunt's algorithm assumes that each combination of attribute sets has a unique class label during the procedure.
- If all the records associated with Dt have identical attribute values except for the class label, then it is not possible to split these records any future. In this case, the node is declared a leaf node with the same class label as the majority class of training records associated with this node.

  Eg:



	categorical	categorical	continuous	class
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Model: Decision Tree

Tree Induction:

Tree induction is based on Greedy Strategy i.e. split the records based on an attribute test that optimize certain criterion.

Issues:

1. **How to split the record?**
2. **How to specify the attribute test condition?**
- Depends on attribute types and number of ways to split the record i.e. 2-ways split /multi-way split.
- Depends upon attribute types. (Nominal, Ordinal, Continuous)
3. **When to stop splitting?**
- When all records are belongs to the same class or all records have similar attributes.
4. **How to determine the best split?**
- Nodes with homogenous class distribution are preferred.
- Measure the node impurity.
  i. Gini-Index
  ii. Entropy
  iii. Misclassification Error

Gini-Index

- The Gini Index measures the impurity of data set (D) as:
- Gini(D) = 1 - $\sum_{1}^{n} p_i^2$

Where,

n = Number of classes, p_i = Probability of ith class.

- It consider binary split for each attribute.
- When D is partition into D_1 and D_2 then

  Gini(D) = D_1/D Gini(D_1) + D_2/D Gini(D_2)
- The attribute that maximize the reduction in impurity is selected as splitting attribute.

Eg:

Refer Class Note

ID3 Algorithm

- The ID3 algorithm begins with the original dataset as the root node.
- On each iteration of the algorithm, it iterates through every unused attribute of the dataset and calculates the entropy (or information gain) of that attribute.
- It then selects the attribute which has the smallest entropy (or largest information gain) value.
- The dataset is then split by the selected attribute to produce subsets of the data.
- The algorithm continues to recur on each subset, considering only attributes never selected before.

  Recursion on a subset may stop in one of these cases:

Algorithm

i. Every element in the subset belongs to the same class , then the node is turned into a leaf and labeled with the class of the examples

ii. If the examples do not belong to the same class ,

- Calculate entropy and hence information gain to select the best node to split data.
- Partition the data into subset.

iii. Recursively repeat until all data are correctly classified

Throughout the algorithm, the decision tree is constructed with each non-terminal node representing the selected attribute on which the data was split, and terminal nodes representing the class label of the final subset of this branch.

Entropy

Entropy H(S) is a measure of the amount of uncertainty in the dataset (S) (i.e. entropy characterizes the dataset (S)).

$$H(S) = -\sum_{x \in X} p(x) \log_2 p(x)$$

Where,

- S The current dataset for which entropy is being calculated (changes every iteration of the ID3 algorithm)

- X - Set of classes in S
- P(x) - The probability of each set S

- When H(S) = 0, the set S is perfectly classified (i.e. all elements in S are of the same class).
- In ID3, entropy is calculated for each remaining attribute. The attribute with the smallest entropy is used to split the set S on this iteration.
- The higher the entropy, the higher the potential to improve the classification here.

Information Gain

Information gain is the measure of the difference in entropy from before to after the set S is split on an attribute A. In other words, how much uncertainty in dataset (S) was reduced after splitting dataset S on attribute A.

$$IG(A,S) = H(S) - \sum_{t \in T} p(t)H(t)$$

Where,

- H(S) - Entropy of dataset S
- T - The subsets created from splitting dataset S by attribute A.
- P(t) - The probability of class t
- H(t) - Entropy of subset t

In ID3, information gain can be calculated (instead of entropy) for each remaining attribute. The attribute with the largest information gain is used to split the set S on this iteration.

Tree Pruning

- Tree Pruning is performed in order to remove anomalies in training data due to noise or outliers.
- The pruned trees are smaller and less complex.
- Pruning is a technique in machine learning that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances.
- The dual goal of pruning is reduced complexity of the final classifier as well as better predictive accuracy by the reduction of overfitting and removal of sections of a classifier that may be based on noisy or erroneous data.
- One of the questions that arise in a decision tree algorithm is the optimal size of the final tree.
- A tree that is too large risks overfitting the training data and poorly generalizing to new samples.

- A small tree might not capture important structural information about the sample space.
- It is hard to tell when a tree algorithm should stop because it is impossible to tell if the addition of a single extra node will dramatically decrease error.
- A common strategy is to grow the tree until each node contains a small number of instances then use pruning to remove nodes that do not provide additional information.
- Pruning should reduce the size of a learning tree without reducing predictive accuracy as measured by a test set or using cross-validation.
- There are many techniques for tree pruning that differ in the measurement that is used to optimize performance.

Tree pruning approaches

i. Prepruning - The tree is pruned by halting its construction early.
ii. Postpruning - This approach removes subtree form fully grown tree.

Pre-pruning

- Based on statistical significance test.
- Stop growing the tree when there is no statistically significant association between any attribute and the class at a particular node
- Most popular test: chi-squared test
- ID3 used chi-squared test in addition to information gain.
- Only statistically significant attributes were allowed to be selected by information gain procedure.
- Pre-pruning may stop the growth process prematurely: early stopping
- Pre-pruning faster than post-pruning

Post-pruning

- First, build full tree then, prune it.
- Fully-grown tree shows all attribute interactions
- Problem: some subtrees might be due to chance effects
- Two pruning operations:
  - Subtree replacement
  - Subtree raising
- Possible strategies:
  - error estimation
  - significance testing
  - MDL principle
- Subtree replacement selects a subtree and replaces it with a single leaf.
- Subtree raising selects a subtree and replaces it with the child one ie, a "sub-subtree" replaces its parent)

Advantages of Decision Tree Classifier

– Inexpensive to construct

– Extremely fast at classifying unknown records

– Easy to interpret for small-sized trees

– Accuracy is comparable to other classification techniques for many simple data sets

Rule-Based Classifier

- It classifies records by using a collection of "If ……. Then….." rules. A rule base classifier uses a set of "If ……. Then….." rules for classification.

eg: If age = youth AND student = yes THEN buys_computer = yes.

- The 'If' part or left hand side of a rule is known as the rule antecedent or precondition where as the 'Then" part or right hand side is the rule consequent.
- In the rule antecedent, the condition consists of one or more attribute tests.
- If the condition in a rule antecedent holds true for a given tuple, the rule antecedent is satisfied and that the rule covers the tuple.
- Coverage of a rule is the fraction of records that satisfy the antecedent of a rule.

  Coverage = $\mathbf{N_{covers} / D}$

  Where,

  $\mathbf{N_{covers}}$ = number of record that can be classified by the rule.

  D = total data set.
- Accuracy of a rule is fraction of records that satisfy both the antecedent and consequent of a rule.

  Accuracy = $\mathbf{N_{correct} / N_{covers}}$

  Where,

  $\mathbf{N_{correct}}$ = Number of records that are correctly classified by the rule

  $\mathbf{N_{covers}}$ **=** Number of record that can be classified by the rule

How does Rule-Based Classifier work?

- If a rule is satisfied by a tuple, the rule is said to be triggered. Triggering doesn't always mean firing because there may be more than one rules that can be satisfied.
- Three different cases occur for classification.

Case-I: If only one rule is satisfied

- When any instances is covered by only one rule then the rule fires by returning the class prediction for the tuple defined by the rule.

Case-II: If more than one rules are satisfied

- If more than one rules are triggered, we need a conflict resolution strategy to find which rule is fired.
- Rule ordering or rule ranking or rule priority can be set in case of rules conflict. A rule ordering may be class-based or rule-based.
- Rule-based ordering: Individual rules are ranked based on their quality.
- Class-based ordering: Rules that belong to the same class appear together

- When rule-based ordering is used, the rule set is known as a decision list.

Case-III: If no rule is satisfied

- If any instance not triggered by any rule, use default class for classification. Mostly most frequent class is assigned as default class.

Eg:

S.No.	Name	Blood Type	Give Birth	Can fly	Live in water	Class
1	Lemur	Warm	Yes	No	No	?
2	Turtle	Cold	No	No	Sometimes	?
3	Shark	Cold	Yes	No	Yes	?

Rule base

R1: (Give Birth = No) ^ (Can fly = Yes) => Birds
R2: (Give Birth = No) ^ (Live in Water = Yes) => Fishes
R3: (Give Birth = Yes) ^ (Blood Type = Warm) => Mammals
R4: (Give Birth = No) ^ (Can fly = No) => Reptiles
R5: (Live in Water = Sometimes) => Amphibians

- In above example, R1 and R2 don’t have any coverage. R3, R4 & R5 have coverage.
- Instance 1 is triggered by R3, instance 2 is triggered by R4 & R5 and instance 3 is not triggered by any instances.
- Since instance 1 is triggered by only one rule (R3) so it is fired as a class mammal, instance 2 is triggered by more than two rules (R4 & R5) and hence conflict occurs. To resolve the conflict the class can be identified using priority (rule priority or class priority). Instance 3 is not triggered by any rules, to resolve this conflict default class can be used.

Characteristics of Rule-Based Classifier

1. **Mutually exclusive Rules**
   - Classifier contains mutually exclusive rules if all the rules are independent of each other.
   - Every record is covered by at most one rule.
   - Rules are no longer mutually exclusive if a record may triggered by more than one rule. To make mutually exclusive we apply rule ordering.

2. **Exhaustive Rules**
   - Classifier has exhaustive coverage if it accounts for every possible combination of attribute values (every possible rule).
   - Each record is covered by at least one rule.
   - Rules are no longer exhaustive if a record may bit trigger any rules. To make rules exhaustive use default class.

Building Classification Rules

- Two approaches are used to build classification rules.

A. Direct Method

- Extract rules directly from data. It is an inductive and sequential approach.

Sequential Covering

1. Start from an empty rule
2. Grow a rule using the Learn-One-Rule function
3. Remove training records covered by the rule
4. Repeat Step (2) and (3) until stopping criterion is met

Aspects of Sequential Covering

- Rule Growing
- Instance Elimination
- Rule Evaluation
- Stopping Criterion
- Rule Pruning

i. ***Rule Growing***

CN2 Algorithm:

– Start from an empty conjunct: {}

– Add conjuncts that minimizes the entropy measure: {A}, {A,B}, …

– Determine the rule consequent by taking majority class of instances covered by the rule

RIPPER Algorithm:

– Start from an empty rule: {} => class

– Add conjuncts that maximize FOIL's information gain measure:

R0: {} => class (initial rule)

R1: {A} => class (rule after adding conjunct)

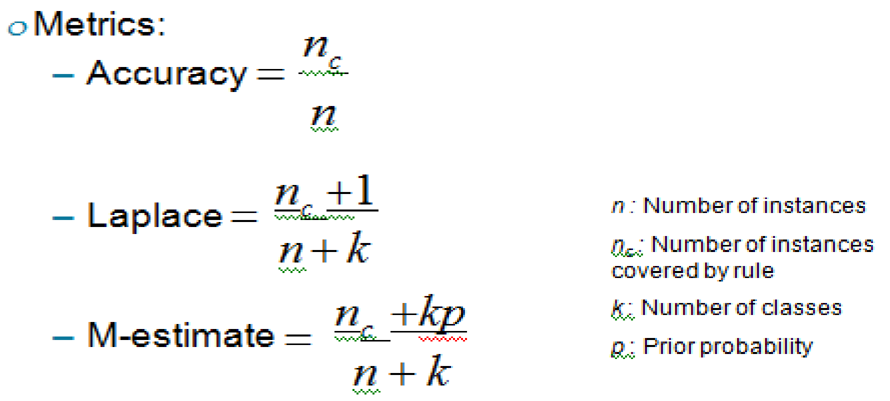
Gain (R0, R1) = t [log (p1/(p1+n1)) – log (p0/(p0 + n0))]

Where, t: number of positive instances covered by both R0 and R1 p0: number of positive instances covered by R0
n0: number of negative instances covered by R0
p1: number of positive instances covered by R1
n1: number of negative instances covered by R1

ii. ***Instance Elimination***

- We need to eliminate instances otherwise, the next rule is identical to previous rule.
- We remove positive instances to ensure that the next rule is different.
- We remove negative instances to prevent underestimating accuracy of rule

iii. ***Rule Evaluation***



- Metrics:
  - Accuracy = $\frac{n_c}{n}$
  - Laplace = $\frac{n_c + 1}{n + k}$
  - M-estimate = $\frac{n_c + kp}{n + k}$

n : Number of instances

n_c : Number of instances covered by rule

k : Number of classes

p : Prior probability

iv. ***Stopping Criterion and Rule Pruning***

<u>Stopping criterion</u>

– Compute the gain
– If gain is not significant, discard the new rule.

<u>Rule Pruning</u>

– Similar to post-pruning of decision trees.

– Reduced Error Pruning:

Remove one of the conjuncts in the rule

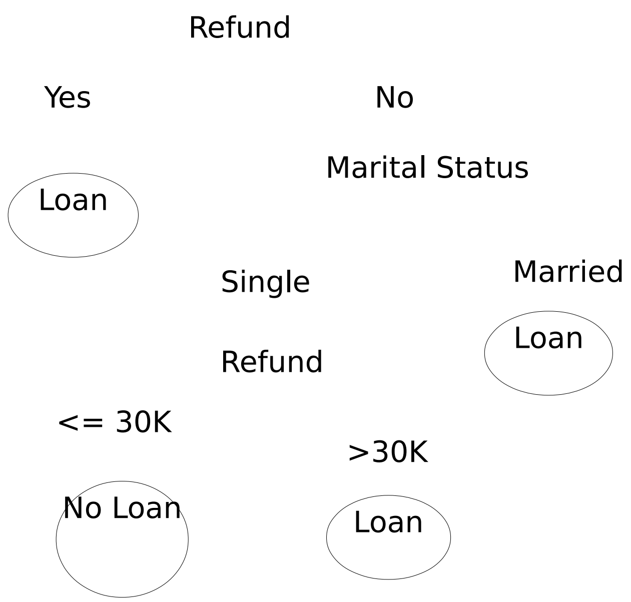
Compare error rate on validation set before and after pruning

If error improves, prune the conjunct

B. **Indirect Method:**

‹- Extract rules from other classification models (e.g. decision trees, neural networks, etc).

<u>Eg; Rule Extraction from Decision Tree</u>



Rules:

R1: (Refund = Yes) => Loan
R2: (Refund = No) ^ (Marital Status = Married) => Loan

Rule simplification
Complex rules can be simplified. In above example R2 can be simplified as:
r2: (Marital Status = Married) => Loan

Advantages of Rule-Based Classifiers

- As highly expressive as decision trees
- Easy to interpret
- Easy to generate
- Can classify new instances rapidly
- Performance comparable to decision trees

Instance Based Classifier

- It Store the training records and • Use training records to predict the class label of unseen cases.
  Examples:

i. **Rote-learner**
  - Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly

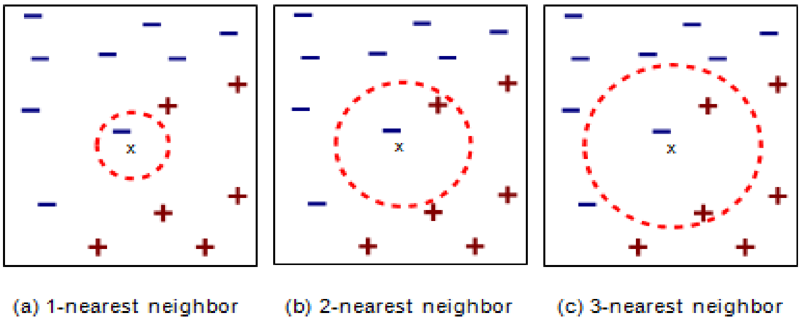
ii. **Nearest neighbor**
  - Uses k "closest" points (nearest neighbors) for performing classification. K-closet neighbor of a record 'X' are data points that have the K-smallest distance of 'X'.
  - Classification based on learning by analogy i.e. by comparing a given test tuple with training tuple that are similar to it.
  - Training tuples are described by n-attributes.
  - When given an unknown tuple, a k-nearest- neighbor classifier searches the pattern space for the k-training tuples that are closest to the unknown tuple.
  - Nearest neighbor classifier requires:
    - Set of stored records
    - Distance matric to compute distance between records. For distance calculation any standard approach can be used sch as Euclidean distance.
    - The value of 'K', the number of nearest neighbor to retrieve.
  - To classify the unknown records
    - Compute distance to other training records.
    - Identify the k-nearest neighbor.
    - Use class label nearest neighbors to determine the class label of unknown

record. In case of conflict, use majority vote for classification.

Issues of classification using k-nearest neighbor classification

i. **Choosing the value of K**

- One of challenge in classification is to choose the appropriate value of K. If K is too small, it is sensitive to noise points. If K is too large, neighbor may include points from other classes.
- With the change of value of K, the classification result may vary.



ii. **Scaling Issue**

- Attribute may have to be scaled to prevent distance measure from being dominated by one of attributes. Eg. Height, Temperature etc.

iii. **Distance computing for non-numeric data.**

- Use Distance as 0 for the same data and maximum possible distance for different data.

iv. **Missing values**

- Use maximum possible distance

Disadvantages:

- Poor accuracy when data have noise and irrelevant attributes.
- Slow when classifying test tuples.
- Classifying unknown records are relatively expensive.

Bayesian Classifier

- Bayesian classification is based on Baye's Theorem.
- It is a statistical classifier that predicts class membership probabilities such as the probability that a given tuple belongs to a particular class.

  Baye's Law

  P(A/B) = P(B/A) P(A)/ P(B)

- Has high accuracy and speed for large databases.
- Has minimum error rate in comparison to all other classifier

Types.

1. ***Bayesian Belief Networks (Graphical Method)***

- Bayesian Belief Network specifies joint conditional probability distributions.
- Bayesian Networks and Probabilistic Network are known as belief network.
- It allows class conditional independencies to be defined between subsets of variables.
- It provides a graphical model of causal relationship on which learning can be performed.
- It represents a set of random variables and their conditional dependencies via a directed acyclic graph

2. ***Naïve Bayesian Classifier***

- The Naive Bayes Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high.
- It simplifies the computational complexity.
- Naïve Bayesian Classifier assumes that the effect of an attribute value on a given class is independent of the value of other attributes i.e. class conditional independence.
- Let D be a training set of tuples and C1, C2, …….., Cm are their associated classes.
- Given a tuple X, the classifier will predict that X belongs to the class having highest posterior probability conditioned on X i.e. the Naïve Bayesian classifier predicts that tuple x belong to the class Ci if and only if

P(Ci/X) > P(Cj/X) for 1 $\leq$ j $\geq$ m, j $\neq$ i

i.e. P(Ci/X) = P(X/Ci)P(Ci)/ P(X) maximum
P(X) = Constant
P(Ci) => P(C1) = P(C2) = ………. = P(Cm)
So we need to maximize P(X/Ci)

Naïve assumption is class condition independence,

$$P(X/Ci) = \prod_{k=1}^{n} P\left(\frac{Xk}{Ci}\right)$$

= P(x1/Ci) * P (x2/Ci) * …………. * P(xn/Ci)

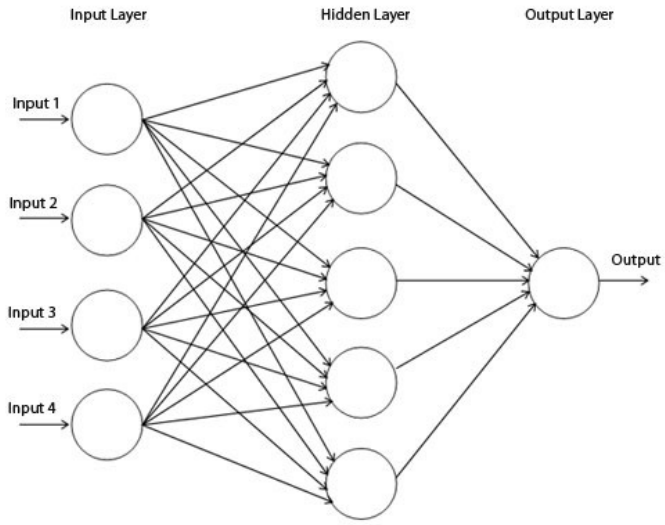
These probabilities can be calculated from training tuples.

Eg: Refer class notes

Artificial Neural Network (ANN) Classifier

- It is set of connected i/o units in which each connection has a weight associated with it.

- During the learning phase the network learns by adjusting the weights so as to be able to predict the correct class label of i/p labels.
- It also referred as connectionist learning due to connection between units.
- It has long training time and poor interpretability but has tolerance to noisy data.
- It can classify pattern on which they have not been trained.
- Well suited for continuous valued i/ps.
- It has parallel topology and processing.
- Before training the network topology must be designed by:
    i. ***Specifying number of i/p nodes/units:*** Depends upon number of independent variable in data set.
    ii. ***Number of hidden layers:*** Generally only layer is considered in most of the problem. Two layers can be designed for complex problem. Number of nodes in the hidden layer can be adjusted iteratively.
    iii. ***Number of output nodes/units:*** Depends upon number of class labels of the data set.
    iv. ***Learning rate:*** Can be adjusted iteratively.
    v. ***Learning algorithm*:** Any appropriate learning algorithm can be selected during training phase.
    vi. ***Bias value***: Can be adjusted iteratively.
- During training the connection weights must be adjusted to fit i/p values with the o/p values.



v

Back propagation algorithm

Step 1: Initialization: Set all the weights and thresholds levels of the network to random numbers uniformly distributed inside a small range.

Step 2: Activation: Activate the back propagation neural network by applying i/ps and desired o/ps.

i. Calculate the actual o/ps of the neurons in the hidden layers.
ii. Calculate the actual o/ps of the neurons in the o/p layers.

Step 3: Weight training:

i. Updates weights in the back propagation network by propagating backwards the errors associated with the o/p neurons.
ii. Calculate error gradient of o/p layer and hence of neurons in the hidden layer.

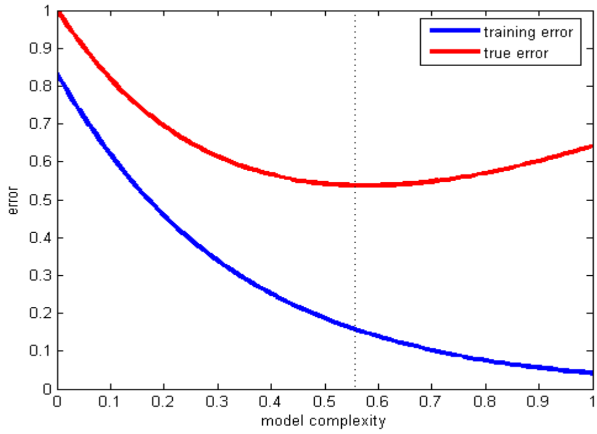
Step 4: Iteration: Increase iteration by repeating steps 2 and 3 until selected error criteria is satisfied.

(Refer textbook for mathematical equations)

Issues:

Overfitting

- Overfitting occurs when a statistical model describes random error or noise instead of the underlying relationship.
- Overfitting generally occurs when a model is excessively complex, such as having too many parameters relative to the number of observations.
- A model which has been overfit will generally have poor predictive performance.
- Overfitting depends not only on the number of parameters and data but also the conformability of the model structure.
- In order to avoid overfitting, it is necessary to use additional techniques (e.g. cross-validation, pruning (Pre or Post), model comparison,



Reason

. Noise in training data.

. Incomplete training data.

. Flaw in assumed theory.

Validation

Validation techniques are motivated by two fundamental problems in pattern recognition: model selection and performance estimation

Validation Approaches:

- One approach is to use the entire training data to select our classifier and estimate the error rate, but the final model will normally overfit the training data.

- A much better approach is to split the training data into disjoint subsets cross validation (The Holdout Method)

Cross Validation (The holdout method)

Data set divided into two groups. Training set: used to train the classifier and Test set: used to estimate the error rate of the trained classifier

Total number of examples = Training Set +Test Set

Approach:

Random Sub sampling

- Random Sub sampling performs K data splits of the dataset
- Each split randomly selects (fixed) no. examples without replacement
- For each data split we retrain the classifier from scratch with the training examples and estimate error with the test examples

K-Fold Cross-Validation

- K-Fold Cross validation is similar to Random Sub sampling.
- Create a K-fold partition of the dataset, For each of K experiments, use K-1 folds for training and the remaining one for testing.
- The advantage of K-Fold Cross validation is that all the examples in the dataset are eventually used for both training and testing.
- The true error is estimated as the average error rate

Leave-one-out Cross-Validation

- Leave-one-out is the degenerate case of K-Fold Cross Validation, where K is chosen as the total number of examples where one sample is left out at each experiment.

Model Comparison:

- Models can be evaluated based on the output using different method :
  i. Confusion Matrix
  ii. ROC Analysis
  iii. Others such as: Gain and Lift Charts, K-S Charts

Confusion Matrix (Contigency Table):

- A confusion matrix contains information about actual and predicted classifications done by classifier.
- Performance of such system is commonly evaluated using data in the matrix.
- It is also known as a contingency table or an error matrix, is a specific table layout that allows visualization of the performance of an algorithm.
- Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.

	Predicted Positive	**Predicted Negative**
Positive Examples	True Positive (TP)	False Negative (FN)
Negative Examples	False Positive (FP)	True Negative (TN)

Accuracy: (TP + TN) / Total data count

Precision: TP / (TP + FP) or TN/ (TN + FN)

True Positive Rate (TPR): TP / (TP +TN)

True Negative Rate (TNR): TN / (TP +TN)

False Positive Rate (FPR): FP / (FP +FN)

False Negative Rate (FNR): FN / (FP +FN)

Example: Refer class note

ROC Analysis

- Receiver Operating Characteristic (ROC), or ROC curve, is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied.

- The curve is created by plotting the true positive rate against the false positive rate at various threshold settings.

- The ROC curve is thus the sensitivity as a function of fall-out.

- In general, if the probability distributions for both detection and false alarm are known, the ROC curve can be generated by plotting the cumulative distribution function (area under the probability distribution from $-\infty$ to $+\infty$) of the detection probability in the y-axis versus the cumulative distribution function of the false-alarm probability in x-axis.

- ROC analysis provides tools to select possibly optimal models and to discard suboptimal ones independently from (and prior to specifying) the cost context or the class distribution.

- ROC analysis is related in a direct and natural way to cost/benefit analysis of diagnostic decision making.

Association Analysis

- Mining for associations among items in a large database of transactions is an important data mining function.
- Association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases.
- Association rules are statements of the form {X1, X_2, …, Xn} => Y, meaning that if we find all of X_1, X_2, ……… , Xn in the transaction then we have good chance of finding Y.

  Eg: The information that a customer who buys computer also tends to buy antivirus or pen drive.
- Association analysis mostly applied in the field of market basket analysis, web-based mining, intruder detection etc.

Market Basket Analysis

- Market basket analysis (also known as Affinity Analysis) is the study of items that are purchased or grouped together in a single transaction or multiple, sequential transactions.
- Understanding the relationships and the strength of those relationships is valuable information that can be used to make recommendations, cross-sell, up-sell, offer coupons, etc.
- A predictive market basket analysis can be used to identify sets of products/services purchased/events) that generally occur in sequence or something of interest to direct marketers.
- Advanced Market Basket Analysis provides an excellent way to get to know the customer and understand the different behaviors that can be leveraged to provide better assortment, design a better plan and devise more attractive promotions that can lead to more sales and profits.
- The analysis can be applied in various ways:

- Develop combo offers based on products sold together.
- Organize and place associated products/categories nearby inside a store.
- Determine the layout of the catalog of an ecommerce site.
- Control inventory based on product demands and what products sell together.

- Support of a product or group of products indicates the popularity of the product or group of products in the transaction set. Higher the support, more popular is the product or product bundle. This measure can help in identifying selling strategy of the store. Eg: if Barbie dolls have a higher support then they can be attractively priced to attract traffic to a store.
- Confidence can be used for product placement strategy and increasing profitability. Place high-margin items with associated high selling items. If Market Basket Analysis indicates that customers who bought high selling Barbie dolls also bought high-margin candies, then candies should be placed near Barbie dolls.

- Lift indicates the strength of an association rule over the random co-occurrence of Item A and Item B, given their individual support. Lift provides information about the change in probability of Item A in presence of Item B. Lift values greater than 1.0 indicate that transactions containing Item B tend to contain Item A more often than transactions that do not contain Item B.
- In order to gain better insights, Market Basket Analysis can based on
- Weekend vs weekday sales
- Month beginning vs month-end sales
- Different seasons of the year
- Different stores
- Different customer profiles
- Although Market Basket Analysis mostly applied for shopping carts and supermarket shoppers, there are many other areas in which it can be applied such as:

For a financial services company

- Analysis of credit and debit card purchases.
- Analysis of cheque payments made.
  - Analysis of services/products taken e.g. a customer who has taken executive credit card is also likely to take personal loan.

For a telecom operator

- Analysis of telephone calling patterns.
- Analysis of value-add services taken together.

Few terminologies used in association analysis

Support: The support of an association pattern is the percentage of task-relevant data transaction for which the pattern is true.

Support (A): Number of tuples containing A / Total number of tuples
Support (A = > B): Number of tuples containing A and B / Total number of tuples

- If minsup is set too high, we could miss itemsets involving interesting rare items (e.g., expensive products)
- If minsup is set too low, it is computationally expensive and the number of itemsets is very large

Confidence: Confidence is defined as the measure of certainty or trustworthiness associated with each discovered pattern.

Confidence (A = > B): Number of tuples containing A and B / Total count of A

Itemset

– A collection of one or more items. Example: {Milk, Bread, Diaper}

– An itemset that contains k items is called k-itemset.

Frequent Itemset

- An itemset whose support is greater than or equal to a minimum support threshold.

Association Rule

- An implication expression of the form X => Y, where X and Y are itemsets.

Example: {Milk, Diaper} => {Beer}

Maximal Frequent Itemset:

- An itemset is maximal if none of its immediate supersets is frequent.

Closed Itemset:

- An itemset is closed if none of its immediate supersets has same support as of the itmeset.

Lift

- Lift is a measure of the performance of a targeting model (association rule) at predicting or classifying cases as having an enhanced response with respect to the population as a whole, measured against a random choice targeting model.
- Lift can be found by dividing the confidence by the unconditional probability of the consequent, or by dividing the support by the probability of the antecedent times the probability of the consequent.
- If some rule had a lift of 1, it would imply that the probability of occurrence of the antecedent and that of the consequent are independent of each other. When two events are independent of each other, no rule can be drawn involving those two events.
- If the lift is > 1, that lets us know the degree to which those two occurrences are dependent on one another, and makes those rules potentially useful for predicting the consequent in future data sets.
- Lift = P(Y | X) /P(Y)

Association Rules Mining

- Given a set of transactions T, the goal of association rule mining is to find all rules having support ≥ minsup threshold and confidence ≥ minconf threshold.
- Some of approaches for association rules mining are:

Brute- Force Approach

- List all possible association rules.
- Compute the support and confidence for each rule.

- Prune rules that fail to minimum support and minimum confidence level.

This approach is computationally very expensive.

Frequent Itemset Generation Strategies

- Reduce the number of candidates (M): For complete search, M=2d. Use pruning techniques to reduce M.
- Reduce the number of transactions (N): Reduce size of N as the size of itemset increases.
- Reduce the number of comparisons (NM): Use efficient data structures to store the candidates or transactions. No need to match every candidate against every transaction.

Apriori Approach

- Apriori approach is two step approach: Frequent item generation and Rules generation
- Based on apriori principal

Apriori Principle:

- Supersets of non-frequent item are also non-frequent. Or, If an itemset is frequent, then all of its subset also be frequent.
- Apriori algorithm is an influential algorithm for mining frequent itemset.
- It use a level-wise search, k-itemsets are used to explore k+1 itemsets.
- At first, the set of frequent itemset is found and used to generate to frequent itemset at next level and so on.

Apriori Algorithm:

- Read the transaction database and get support for each itemset, compare the support with minimum support to generate frequent itemset at level 1.
- Use join to generate a set of candidate k-itmesets at next level.
- Generate frequent ietmsets at next level using minimum support.
- Repeat step 2 and 3 until no frequent itme sets can be generated.
- Generate rules form frequent itemsets from level 2 onwards using minimum confidence.

*****This approach has faster than Brute-Force approach but still has higher computational complexity.***

Example: Refer class note

Reducing Number of Comparisons

Candidate counting:

- Scan the database of transactions to determine the support of each candidate itemset.
- To reduce the number of comparisons, store the candidates in a hash structure
- Instead of matching each transaction against every candidate, match it against candidates contained in the hashed buckets

Hash Table

- A hash table (hash map) is a data structure used to implement an associative array, a structure that can map keys to values.

- A hash table uses a hash function to compute an index into an array of buckets or slots, from which the correct value can be found.
- Max leaf size: max number of itemsets stored in a leaf node, if number of candidate itemsets exceeds max leaf size, split the node.

Factors Affecting Complexity

i. ***Choice of minimum support threshold:*** Lowering support threshold results in more frequent itemsets.This may increase number of candidates and max length of frequent itemsets.

ii. ***Dimensionality (number of items) of the data set:*** More space is needed to store support count of each item. If number of frequent items increases, both computation and I/O costs may also increase.

iii. ***Size of database:*** Since Apriori makes multiple passes, run time of algorithm may increase with number of transactions.

iv. ***Average transaction width:*** Transaction width increases with denser data sets. This may increase max length of frequent itemsets and traversals of hash tree (number of subsets in a transaction increases with its width)

Frequent Pattern (FP) Growth Method

- Mining frequent itmesets without candidate generation.
- It is a divide and conquers strategy.
- It compress the database representing frequent items into a frequent –pattern tree (FP-Tree), which retains the itemset association information.
- Divides the compressed database into a set of conditional databases, each associated with one frequent item or pattern fragment and then mines each such database separately.
- FP-Growth method transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix.
- It uses least frequent items as suffix .
- Adv: Reduce search cost, has good selectivity, faster than apriori.
- Disadv: When the database is large, it is sometimes unrealistic toconstruct a man memory based FP-tree.

FP-Tree algorithm

- Create root node of tree, labeled with null.
- Scan the transactional database.
- The items in each transaction are processed in sorted order (Descending) and branch is created for each transaction.

FP-Tree algorithm

- Start from each frequent length pattern as an initial suffix pattern.
- Construct conditional pattern base. (Pattern base is a sub database which consists of the set of prefix paths in the FP-tree co-occurring with suffix

pattern.

- Construct its FP-tree and perform mining recursively on such a tree

Example: Refer class note

Categorical data

- Categorical data is a statistical data type consisting of categorical variables, used for observed data whose value is one of a fixed number of nominal categories.

- More specifically, categorical data may derive from either or both of observations made of qualitative data, where the observations are summarized as counts or cross tabulations, or of quantitative data.

- Observations might be directly observed counts of events happening or they might counts of values that occur within given intervals.

- Often, purely categorical data are summarized in the form of a contingency table.

- However, particularly when considering data analysis, it is common to use the term "categorical data" to apply to data sets that, while containing some categorical variables, may also contain non-categorical variables.

Potential Issues

- ***What if attribute has many possible values:*** Example: attribute country has more than 200 possible values. Many of the attribute values may have very low support.

  Potential solution: Aggregate the low-support attributes values.

- ***What if distribution of attribute values is highly skewed:*** Example: 95% of the visitors have Buy = No. Most of the items will be associated with (Buy=No) item

  Potential solution: drop the highly frequent items

Handling Categorical Attributes

- Transform categorical attribute into asymmetric binary variables. i.e If the outcomes of a binary variable are not equally important.
- Introduce a new “item” for each distinct attribute- value pair.

Sequential Pattern

- Mining of frequently occurring ordered events or subsequences as patterns. Eg: web sequence, book issued in library etc.
- Used mostly in marketing, customer analysis, prediction modeling.
- A sequence is an ordered list of events where an item can occur at most in an event of a sequence but can occur multiple times in different events of a sequence.
- Given a set of sequences, where each sequence consists of a list of events or elements and each event consists of set of items, given a minimum support threshold, sequential pattern mining finds all frequent subsequences.
- Sequence with minimum support is called frequent sequence or sequential pattern.
- A sequential pattern with length 'l' is called an l-pattern sequential pattern.
- Sequential pattern is computationally challenging because such mining may generate combinationally explosive number of intermediate subsequences.
- For efficient and scalable sequential pattern mining two common approaches are:
    i. Mining the full set of sequential patterns
    ii. Mining only the set of closed sequential pattern
- A sequence database is a set of tuples with sequence_ID and sequences. Eg:

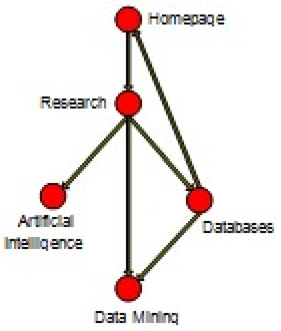
Sequence_ID	Sequence
1	{(a, (a,b,c), (a,c), (b,c)}
2	{(a,b,c), (a,d),e,(d,e)}
3	{(c,d), (a,d,e),e}
4	{ (e,f,),d,(a,b,c),f]

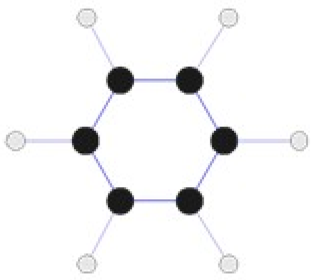
Sub-graph Patterns

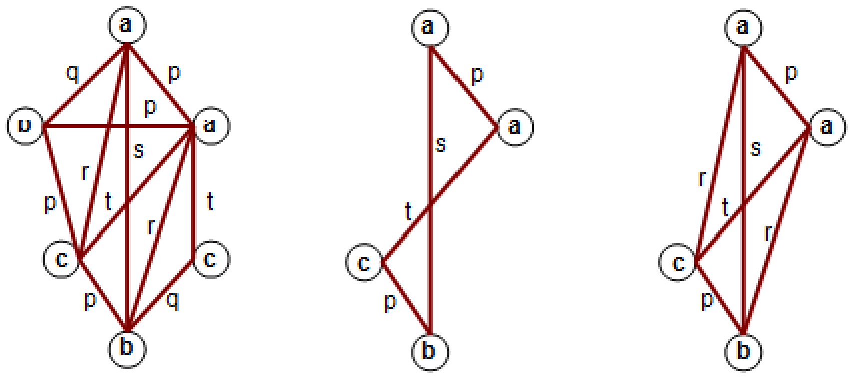
- It finds characteristics sub-graphs within the network.
- It is a form of graph search.
- Given a labeled graph data set, D = {G1, G2, …….,Gn}, a frequent graph has minimum support not less than minimum threshold support.
- Frequent sub-graph pattern can be discovered by generating frequent substructures candidate and hence check the frequency of each candidate.
- Apriori methos and frequent –growth are tow common basic methods for finding frequent sub-graph
- Extend association rule mining to finding frequent subgraphs
- Useful for Web Mining, computational chemistry, bioinformatics, spatial data sets, etc

  Eg::

  Chemical Structure, Geographical Nodes







(a) Labeled Graph (b) Subgraph (c) Induced Subgraph

Challenges

- Node may contain duplicate labels.
- How to define support and confidence?
- Additional constraints imposed by pattern structure
  - Support and confidence are not the only constraints
  - Assumption: frequent subgraphs must be connected

****Apriori-like* approach**:

– Use frequent k-subgraphs to generate frequent (k+1) subgraphs

What is frequent-pattern mining in Data Streams?

- Frequent-pattern mining finds a set of patterns that occur frequently in a data set, where a pattern can be a set of items (called an itemset), a subsequence, or a substructure.
- A pattern is considered frequent if its count satisfies a minimum support. Scalable methods for mining frequent patterns have been extensively studied for static data sets.
- Challenges in mining data streams:
  - Many existing frequent-pattern mining algorithms require the system to scan the whole data set more than once, but this is unrealistic for infinite data streams.
  - A frequent itemset can become infrequent as well. The number of infrequent itemsets is exponential and so it is impossible to keep track of all of them.

Clustering

- Cluster is a collection on data objects in which the objects are similar to one another within the same cluster and dissimilar to objects of another cluster.
- Given a database D = { t_1, t_2, t_3,…….., t_n}, a distance measure dist. (t_i, t_j,)defined between any two objects t_i and t_j and an integer value K (number of clusters), the clustering problem is to define a mapping f: D ->{ 1,2, ……., K) where each t_i is assigned ot on of cluster.
- Clustering is similar to classification where similar objects are placed together. Grouped are not predefined as in classification.
- Clustering is an example of unsupervised learning i.e. learning by observation.
- Clustering is also called data segmentation.
- Also used for outliers detection.

What is not Cluster Analysis:

- ***Supervised classification***: Have class label information
- ***Simple segmentation:*** Dividing students into different registration groups alphabetically, by last name.
- ***Results of a query:*** Groupings are a result of an external specification.
- ***Graph partitioning:*** Some mutual relevance and synergy, but areas are not identical

Distinctions between sets of clusters

Partitioning versus Hierarchical

- A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset.
- A set of nested clusters organized as a hierarchical tree.

Exclusive versus non-exclusive:

- In non-exclusive clustering, points may belong to multiple clusters.
- Can represent multiple classes or 'border' points.

Fuzzy versus non-fuzzy

- In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
- Weights must sum to 1
- Probabilistic clustering has similar characteristics

Partial versus complete

- In some cases, we only want to cluster some of the data

Heterogeneous versus homogeneous

- Cluster of widely different sizes, shapes, and densities

Types of Clusters

a. ***Well-separated clusters:*** Clusters are placed at different places.
b. ***Center-based clusters:*** The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most "representative" point of a cluster

c. ***Contiguous clusters:*** Nearest neighbor or Transitive
d. ***Density-based clusters:*** Used when the clusters are irregular or intertwined, and when noise and outliers are present.
e. ***Property or Conceptual:*** Finds clusters that share some common property or represent a particular concept
f. ***Described by an Objective Function:***

- Finds clusters that minimize or maximize an objective function.
- Enumerate all possible ways of dividing the points into clusters and evaluate the `goodness' of each potential set of clusters by using the given objective function. (NP Hard).
- Can have global or local objectives.
  - Hierarchical clustering algorithms typically have local objectives
  - Partitioned algorithms typically have global objectives
- A variation of the global objective function approach is to fit the data to a parameterized model.
  - Parameters for the model are determined from the data.
  - Mixture models assume that the data is a ‘mixture' of a number of statistical distributions.
- Map the clustering problem to a different domain and solve a related problem in that domain
  - Proximity matrix defines a weighted graph, where the nodes are the points being clustered, and the weighted edges represent the proximities between points
  - Clustering is equivalent to breaking the graph into connected components, one for each cluster.
  - Want to minimize the edge weight between clusters and maximize the edge weight within clusters

Clustering techniques:

i. Partitioning Clustering
ii. Hierarchical Clustering
iii. Density-based Clustering
iv. Grid-based Clustering
v. Model-based clustering.

- Characteristics of the input data are important for the selection of the clustering technique.
  - Type of proximity or density measure.
  - Sparseness i.e. type of similarity
  - Attribute type
  - Type of Data
  - Dimensionality
  - Noise and Outliers
  - Type of Distribution

Partitioning Clustering (Iterative relocation method)

- A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset.
- Partition the database into predefined number of cluster.
- Attempt to determine K-partition that optimizes the certain function.
- Construct a partition of a database D of n-objects into a set of K clusters such that there is the minimum sum of square distance.
- Common partitioning algorithms are: K-means, K-mode, K-medioid.

K-means algorithm (Simple k-means)

- Choose number of cluster (K) to be determined.
- Choose k objects randomly from the data as the centers of k clusters.
- Assign each of the remaining objects to the cluster whose center it is most close to using Euclidean distance.
- Computer the new cluster centers of the clusters using mean points.
- Repeat step3 and 4 until no change in cluster centers or no object change in clusters.

Example: Refer class note

Advantages:

- Relatively simple.
- Simple Implementation.

Disadvantages (Weakness):

- Need to specify number of cluster (K) in advance.
- Unable to handle noisy data and outliers.
- Complexity increases with increase in size.
- Can’t handle categorical data.
- Not efficient for highly non-uniform distributed data.
- Basic K-means algorithm can yield empty clusters.
- May generate empty cluster.

Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE). For each point, the error is the distance to the nearest cluster.
- To get SSE, we square these errors and sum them.

$$SSE = \sum_{\substack{2 \\ i=1}}^{K} \sum_{x \in C_i} dist(m_i, x)$$

- x is a data point in cluster Ci and mi is the representative point for cluster Ci i.e. mi corresponds to the center (mean) of the cluster.
- Given two clusters, we can choose the one with the smallest error.
- One easy way to reduce SSE is to increase K, the number of clusters.
- A good clustering with smaller K can have a lower SSE than a poor clustering with higher K.

Hierarchical Clustering

- A nested set of cluster is created with each level in the hierarchy. At each level it has separate set of clusters.
- At the lowest level, each item is in its own unique cluster.
- At the highest level, all items belong to the same cluster.
- Do not have to assume any particular number of clusters. Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level.

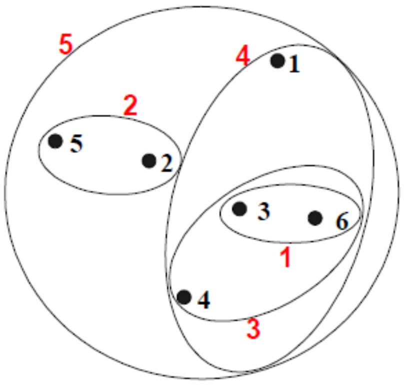
Types

i. **Agglomerative (Bottom-Up)**

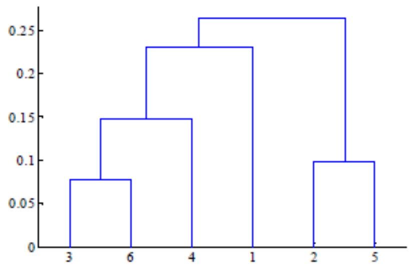
- Start from clustering individual point only, with each cluster having only one record.
- Repeat merging the cluster until a certain number of clusters are left.
- The merging is done on the basis of pair nearest to each other.
- If the merging is continued, it terminates in the hierarchy of clusters which ends into a single cluster.
- Agglomerative is more powerful.

ii. **Divisive**

- Start from a cluster including all points.
- Repeat splitting the cluster until a certain number of clusters are left.
- The splitting is done on the basis of optimization function.
- If the splitting is continued, it terminates in the hierarchy of clusters which ends into a number clusters with having one data points in each cluster.



Nested Clusters



Dendrogram

Bisecting K-means

- Variant of K-means that can produce a partitioned or a hierarchical clustering.
- Bisecting k-Means is like a combination of k-Means and hierarchical clustering.

Algorithm

1. Initialize the list of clusters to contain the cluster containing all the points

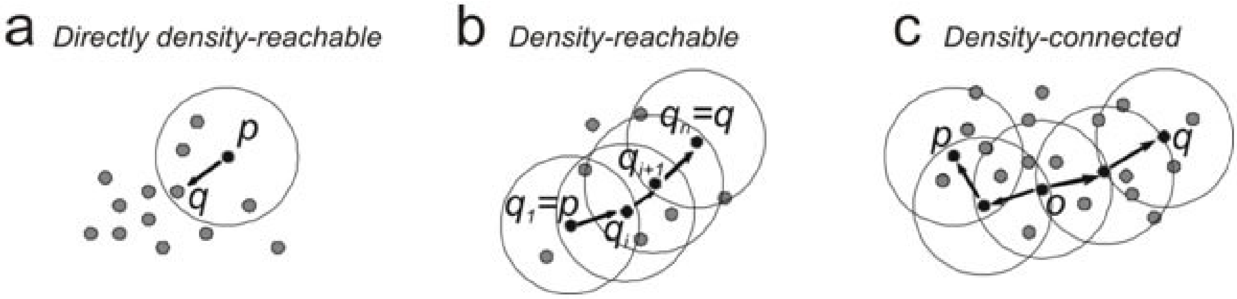
2. Pick a cluster to split.
3. Find 2 sub-clusters using the basic k-Means algorithm (Bisecting step)
4. Repeat step 2, the bisecting step, for ITER times and take the split that produces the clustering with the highest overall similarity.
5. Repeat steps 1, 2 and 3 until the desired number of clusters is reached.

Problems and Limitations

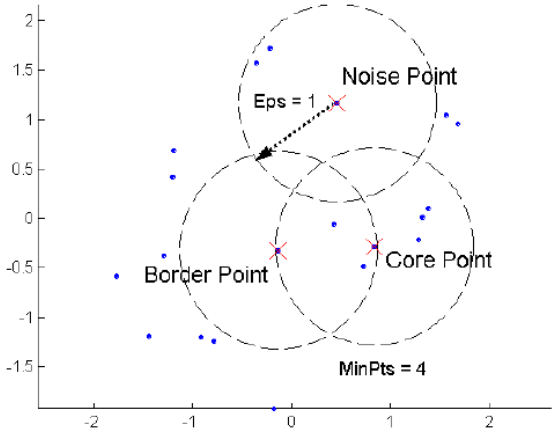
- Once a decision is made to combine two clusters, it cannot be undone.
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
    - Sensitivity to noise and outliers
    - Difficulty handling different sized clusters and convex shapes
    - Breaking large clusters

Density-Based Clustering

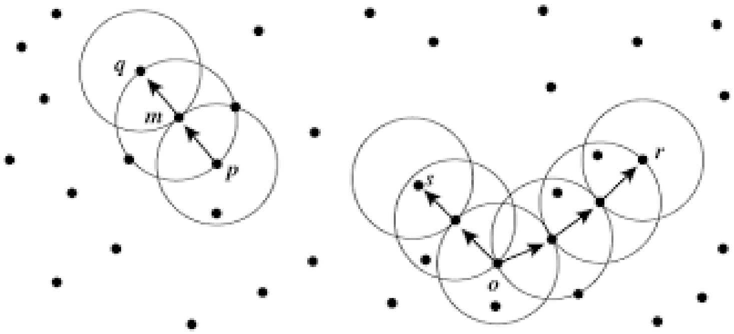
- Grows regions with sufficiently high density into clusters and discover clusters of arbitrary shape in spatial database with noise.
- The neighborhood within the radius ‘ε’ of a given object is called the ε-neighborhood of the object.
- If the ε-neighborhood of the object of an object contains at least a minimum number of points of object then the object is called ***core object***.
- Given a set of objects (D), an object ‘p’ is directly density reachable from object ‘q’ if ‘p’ is within the ε-neighborhood of q and q is a core object.
- An object ‘p’ is density reachable from object ‘q’ with respect to ε and minimum points in a set of objects (D), if there is a chain of objects p1,p2,……, pn such that p_{i+1} is density reachable from p_i with respect to ε and minimum points.
- An object ‘p’ is density reachable from object ‘q’ with respect to ε and minimum points in a set of objects (D), if there is an object O є D such that both p and q are density reachable from O with respect to ε and minimum points.
- Density reachability is the transitive closure of direct density reachability and thus relationship is asymmetric.
- Only core objects are mutually density reachable and are symmetric relation.


a Directly density-reachable
b Density-reachable
c Density-connected

- ***Border point*** has fewer than MinPts within ε-neighborhood but is in the neighborhood of a core point.
- A ***noise point*** is any point that is not a core point or a border point.



Eg:



- Above fig. shows two clusters with arbitrary shapes.
- m, p, o are core objects, each contain minimum point (4) in their ε-neighborhood.
- ‘q’ is directly density reachable from m.
- ‘m’ is directly density reachable from p and vice-versa.
- ‘q’ is density reachable from ‘p’ because ‘q’ is directly density reachable from ‘m’ and ‘m; is directly density reachable from ‘p’. However, ‘p’ is not density reachable from ‘q’ since ‘q’ is not a core object.
- Similarly ‘o’, ‘r’ and‘s’ all are density connected.

Algorithm

- Search for cluster by checking the ε-neighborhood of each point in database.
- If the ε-neighborhood of any point contains more than minimum points, a new cluster with that point as core object is created.
- Iteratively collects directly density reachable objects from these core objects which may involve the merge of a few density reachable clusters.
- Terminates when no new points can be added to any cluster.

Issues:

Cluster Evaluation

1. Intrinsic

- Measure cluster quality based on how "tight" the clusters are.
- Do genes in a cluster appear more similar to each other than genes in other clusters?

Intrinsic Evaluation Methods

a. **Cross-validation:**
 - Leave out *k* experiments (or genes) then perform clustering.
 - Measure how well clusters group in left out experiment.

b. **Rand Index**
 -The Rand index is a simple criterion used to compare an induced clustering structure (C_1) with a given clustering structure (C_2).
 -Let a be the number of pairs of instances that are assigned to the same cluster in C_1 and in the same cluster in C_2;
 - b be the number of pairs of instances that are in the same cluster in C_1, but not in the same cluster in C_2;
 - c be the number of pairs of instances that are in the same cluster in C_2, but not in the same cluster in C_1;
 - and d be the number of pairs of instances that are assigned to different clusters in C_1 and C_2.
 -The Rand index is defined as:

$$RAND = (a + d)/ (a + b + c + d)$$

 -The Rand index lies between 0 and 1. When the two partitions agree perfectly, the Rand index is 1.

c. **Sum of squares**: A good clustering yields clusters where genes have small within-cluster sum-of-squares (and high between-cluster sum-of-squares).

d. **Silhouette:** Good clusters are those where the genes are close to each other compared to their next closest cluster.

e. **Gap statistic**

Extrinsic:
Compare the results to some best standard labeled data.

Scalability

- Clustering techniques for large sets of data must be scalable, both in terms of speed and space.
- There may be millions of records, and thus, any clustering algorithm used should have linear or near linear time complexity to handle such large data sets. (Even algorithms that have complexity of $O(m_2)$ are not practical for large data sets.).
- Some clustering techniques use statistical sampling. Nonetheless, there are cases, e.g., situations where relatively rare points have a dramatic effect on the final clustering, where a sampling is insufficient.
- Clustering techniques for databases cannot assume that all the data will fit in main memory or that data elements can be randomly accessed.
- Accessing data points sequentially and not being dependent on having all the data in main memory at once are important characteristics for scalability.

Comparison

- The choice of clustering algorithm depends on type of data available and on the particular purpose of the application.
- Several algorithms can be applied on same data for descriptive or exploratory purpose cluster analysis.
- It is difficult to generalize the algorithm and techniques for clustering since some application may have clustering criteria that requires the integration of several techniques.
- Clustering techniques highly depends on dimensions and constraints.

Chapter 6: Anomaly/Fraud Detection

Anomaly Detection

- Anomaly detection is a form of classification.
- Is the process to localize objects that are different from other objects (anomalies).
- The set of data points that are considerably different than the remainder of the data are anomalies/outliers.
- Anomaly detection is the process of detecting something unusual relative to something expected.
- The goal of anomaly detection is to identify cases that are unusual within data that is seemingly homogeneous.

Why is Anomaly Detection important?

- to detect problems
- to detect new phenomenon
- to discover unusual behavior in data
-

Examples of interesting application for Anomaly Detection

- Fraud Detection - looking for buying patterns different from typical behavior
- Intrusion Detection - monitoring systems and networks for unusual behavior
- Ecosystem Disturbances - try to predict events like hurricanes and floods
- Public Health - use medical statistic reports for diagnosis
- Medicine - use unusual symptoms or test result to indicate potential health problems

Challenges

- How many outliers are there in the data?
- Method is unsupervised
- There are considerably more “normal” observations than “abnormal” observations (outliers/anomalies) in the data.

Anomaly Detection Schemes

General Steps

- Build a profile of the “normal” behavior, profile can be patterns or summary statistics for the overall population.
- Use the “normal” profile to detect anomalies, anomalies are observations whose characteristics differ significantly from the normal profile

Types of anomaly detection schemes

i. **Graphical based** : Box plot (1-D), Scatter plot (2-D), Spin plot (3-D)

ii. **Statistical-based :**

- Assume a parametric model describing the distribution of the data (e.g., normal distribution.

- A statistical test that depends on:

  . Data distribution
  . Parameter of distribution (e.g., mean, variance)
  . Number of expected outliers (confidence limit)

  a. Grubbs' Test :

  - Detect outliers in univariate data.
  - Assume data comes from normal distribution.
  - Detects one outlier at a time, remove the outlier, and repeat.

  b. Likelihood Approach

  - Assume the data set D contains samples from a mixture of two probability distributions:
  – M (majority distribution)
  – A (anomalous distribution)

  General Approach:

  – Initially, assume all the data points belong to M
  – Let Lt(D) be the log likelihood of D at time t
  - Let Lt+1 (D) be the new log likelihood.
  - Compute the difference, Δ = Lt(D) – Lt+1 (D)
  - If Δ > c (some threshold), then Xt is declared as an anomaly and moved permanently from M to A

  Limitations of Statistical Approaches

  - Most of the tests are for a single attribute
  - In many cases, data distribution may not be known
  - For high dimensional data, it may be difficult to estimate the true distribution

iii. **Distance-based:** Data is represented as a vector of features.

  Three major approaches
  – Nearest-neighbor based
  – Density based
  – Clustering based

iv. **Model-based :**

  - An anomaly detection model predicts whether a data point is typical for a given distribution or not.
  - An atypical data point can be either an outlier or an example of a previously unseen class.
  - Normally, a classification model must be trained on data that includes both examples and counter-examples for each class so that the model can learn to distinguish between them.
  - For example, a model that predicts side effects of a medication should be trained on data that includes a wide range of responses to the medication.

v. **Convex Hull Method**

  - Extreme points are assumed to be outliers. Use convex hull method to detect extreme values.
  - Major limitation is if the outlier occurs in the middle of the data.

Issues

a. **Number of Attributes**: Since an object may have many attributes, it may have anomalous values for some attributes; an object may be anomalous even if none of its attribute values are individually anomalous.
b. **Global Vs Local Perspective**: An object may seem unusual with respect to all objects, but not with respect to its local neighbors.
c. **Degree of Anomaly**: Some objects are more extreme anomalies than others;
d. **One at Time Vs Many at Once**: Is it better to remove anomalous objects one at a time or identify a collection of objects together?
e. **Evaluation**: Finding a good measure of evaluation for the process of anomaly detection when class labels are available and when class labels are not available.
f. **Efficiency**: calculate the computational cost of the process of anomaly detection scheme.

Base Rate Fallacy

- The base-rate fallacy is people's tendency to ignore base rates in favor of individuating information when such is available rather than integrate the two. This tendency has important implications for understanding judgment phenomena in many clinical, legal, and social-psychological settings.
- Base rate fallacy, also called base rate neglect or base rate bias, is a formal fallacy. If presented with related base rate information and specific information, the mind tends to ignore the former and focus on the latter.

Example

A group of policemen have breathalyzers displaying false drunkenness in 5% of the cases in which the driver is sober. However, the breathalyzers never fail to detect a truly drunk person. 1/1000 of drivers are driving drunk. Suppose the policemen then stop a driver at random, and force the driver to take a breathalyzer test. It indicates that the driver is drunk. We assume you don't know anything else about him or her. How high is the probability he or she really is drunk?

Many would answer as high as 0.95, but the correct probability is about 0.02.

To find the correct answer, one should use Bayes' theorem. The goal is to find the probability that the driver is drunk given that the breathalyzer indicated he/she is drunk, which can be represented as

$$p(drunk|D)$$

where "D" means that the breathalyzer indicates that the driver is drunk.

Using Bayes' Theorem ,

$$p(drunk|D) = \frac{p(D|drunk)\,p(drunk)}{p(D)}$$

We have,

$$p(drunk) = 0.001$$
$$p(sober) = 0.999$$
$$p(D|drunk) = 1.00$$
$$p(D|sober) = 0.05$$

$$p(D) = p(D|drunk)\,p(drunk) + p(D|sober)\,p(sober)$$
$$p(D) = 0.05095$$

Putting values into Bayes' Theorem, we get

$$p(drunk|D) = 0.019627\cdot$$

A more intuitive explanation: in average, for every 1000 drivers tested,

- 1 driver is drunk, and it is 100% certain that for that driver there is a true positive test result, so there is 1 true positive test result
- 999 drivers are not drunk, and among those drivers there are 5% false positive test results, so there are 49.95 false positive test results therefore the probability that one of the drivers among the 1 + 49.95 = 50.95 positive test results really is drunk is $p(drunk|D) = 1/50.95 \approx 0.019627$. The validity of this result does, however, hinge on the validity of the initial assumption that the policemen stopped the driver truly at random, and not because of bad driving. If that or another non-arbitrary reason for stopping the driver was present, then the calculation also involves the probability of a drunk driver driving competently and a non-drunk driver driving competently.

Chapter- 7 Advanced Application

A. Web Mining

Web mining is the application of data mining techniques to extract knowledge from Web data, i.e. Web Content, Web Structure and Web Usage data.

Web Mining Taxonomy

Web Mining can be broadly divided into three distinct categories, according to the kinds of data to be mined.

a. Web Content Mining:

- Web Content Mining is the process of extracting useful information from the contents of Web documents.
- Content data corresponds to the collection of facts a Web page was designed to convey to the users.
- May consist of text, images, audio, video, or structured records such as lists and tables.
- Web content has been the most widely researched. Issues addressed in text mining are, topic discovery, extracting association patterns, clustering of web documents and classification of Web Pages.

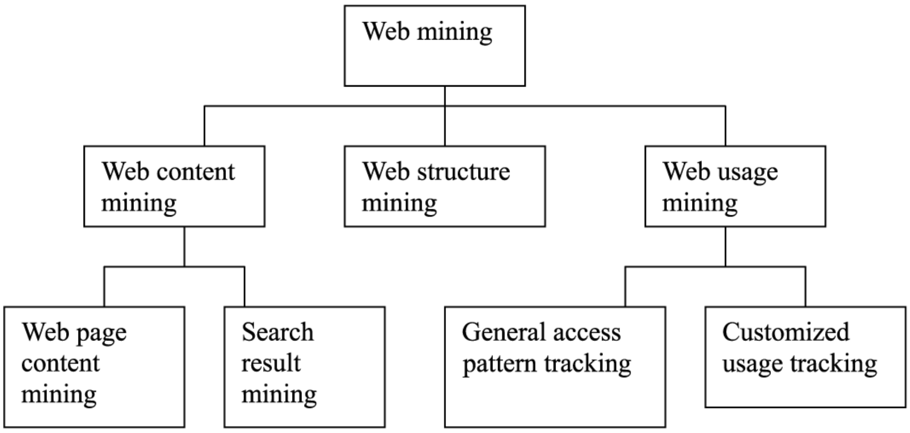
b. Web Structure Mining:

- The structure of a typical Web graph consists of Web pages as nodes, and hyperlinks as edges connecting related pages.
- Web Structure Mining is the process of discovering structure information from the Web. This can be further divided into two kinds based on the kind of structure information used.
  - ▪ Hyperlinks: A Hyperlink is a structural unit that connects a location in a Web page to different location, either within the same Web page or on a different Web page. A hyperlink that connects to a different part of the same page is called an Intra-Document Hyperlink, and a hyperlink that connects two different pages is called an Inter-Document Hyperlink.
  - ▪ Document Structure: In addition, the content within a Web page can also be organized in a tree-structured format, based on the various HTML and XML tags within the page. Mining efforts here have focused on automatically extracting document object model structures out of documents.

c. Web Usage Mining:

- Web Usage Mining is the application of data mining techniques to discover interesting usage patterns from Web data, in order to understand and better serve the needs of Web-based applications.
- Usage data captures the identity or origin of Web users along with their browsing behavior at a Web site.
- Web usage mining itself can be classified further depending on the kind of usage data considered:
  - ▪ Web Server Data: The user logs are collected by Web server. Typical data includes IP address, page reference and access time.
  - ▪ Application Server Data: Commercial application servers such as Web logic Story Server have significant features to enable E-commerce applications to be built on top of them with little effort. A key feature is the ability to track various kinds of business events and log them in application server logs.
  - ▪ Application Level Data: New kinds of events can be defined in an application, and logging can be turned on for them - generating histories of these specially defined events. It must be noted however that many end

applications require a combination of one or more of the techniques applied in the above the categories.



Challenges:

i. Too huge for effective data warehousing and data mining.
ii. Too complex and heterogeneous.
iii. Growing and changing rapidly
iv. Broad diversity of user communities.
v. Only small portion of the information on the web is truly relevant or useful.

The Page Rank Algorithm

The original Page Rank algorithm was described by Lawrence Page and Sergey Brin in several publications. It is given by

$$PR(A) = (1-d) + d\,(PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))$$

where
PR(A) is the Page Rank of page A,
PR(Ti) is the Page Rank of pages Ti which link to page A,
C(Ti) is the number of outbound links on page Ti and
d is a damping factor which can be set between 0 and 1.

- Page Rank does not rank web sites as a whole, but is determined for each page individually. Further, the Page Rank of page A is recursively defined by the Page Ranks of those pages which link to page A.
- The Page Rank of pages Ti which link to page A does not influence the PageRank of page A uniformly. Within the Page Rank algorithm, the Page Rank of a page T is always weighted by the number of outbound links C(T) on page T. This means that the more outbound links a page T has, the less will page A benefit from a link to it on page T.

- The weighted Page Rank of pages Ti is then added up. The outcome of this is that an additional inbound link for page A will always increase page A's Page Rank.
- Finally, the sum of the weighted Page Ranks of all pages Ti is multiplied with a damping factor d which can be set between 0 and 1. Thereby, the extend of PageRank benefit for a page by another page linking to it is reduced.

A Different Notation of the PageRank Algorithm

Lawrence Page and Sergey Brin have published two different versions of their Page Rank algorithm in different papers. In the second version of the algorithm, the Page Rank of page A is given as

$$PR(A) = (1-d) / N + d\ (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))$$

Where N is the total number of all pages on the web. The second version of the algorithm, indeed, does not differ fundamentally from the first one.

The Characteristics of Page Rank

The characteristics of Page Rank shall be illustrated by a small example.

We regard a small web consisting of three pages A, B and C, whereby page A links to the pages B and C, page B links to page C and page C links to page A. According to Page and Brin, the damping factor d is usually set to 0.85, but to keep the calculation simple we set it to 0.5. The exact value of the damping factor d admittedly has effects on Page Rank, but it does not influence the fundamental principles of Page Rank. So, we get the following equations for the Page Rank calculation:

$PR(A) = 0.5 + 0.5\ PR(C)$
$PR(B) = 0.5 + 0.5\ (PR(A) / 2)$
$PR(C) = 0.5 + 0.5\ (PR(A) / 2 + PR(B))$

These equations can easily be solved. We get the following Page Rank values for the single pages:

$PR(A) = 14/13 = 1.07692308$
$PR(B) = 10/13 = 0.76923077$
$PR(C) = 15/13 = 1.15384615$

It is obvious that the sum of all pages' Page Ranks is 3 and thus equals the total number of web pages. As shown above this is not a specific result for our simple example. For our simple three-page example it is easy to solve the according equation system to determine Page Rank values. In practice, the web consists of billions of documents and it is not possible to find a solution by inspection.

The Iterative Computation of Page Rank

Because of the size of the actual web, the Google search engine uses an approximate, iterative computation of Page Rank values. Each page is assigned an initial starting value and the Page Ranks of all pages are then calculated in several computation circles based on the equations determined by the Page Rank algorithm. The iterative calculation shall again be illustrated by our three-page example, whereby each page is assigned a starting Page Rank value of 1.

Iteration	PR(A)	PR(B)	PR(C)
0	1	1	1
1	1	0.75	1.125
2	1.0625	0.765625	1.1484375
3	1.07421875	0.76855469	1.15283203
4	1.07641602	0.76910400	1.15365601
5	1.07682800	0.76920700	1.15381050
6	1.07690525	0.76922631	1.15383947
7	1.07691973	0.76922993	1.15384490
8	1.07692245	0.76923061	1.15384592
9	1.07692296	0.76923074	1.15384611
10	1.07692305	0.76923076	1.15384615
11	1.07692307	0.76923077	1.15384615
12	1.07692308	0.76923077	1.15384615

We see that we get a good approximation of the real Page Rank values after only a few iterations.

B. Time Series Data Mining

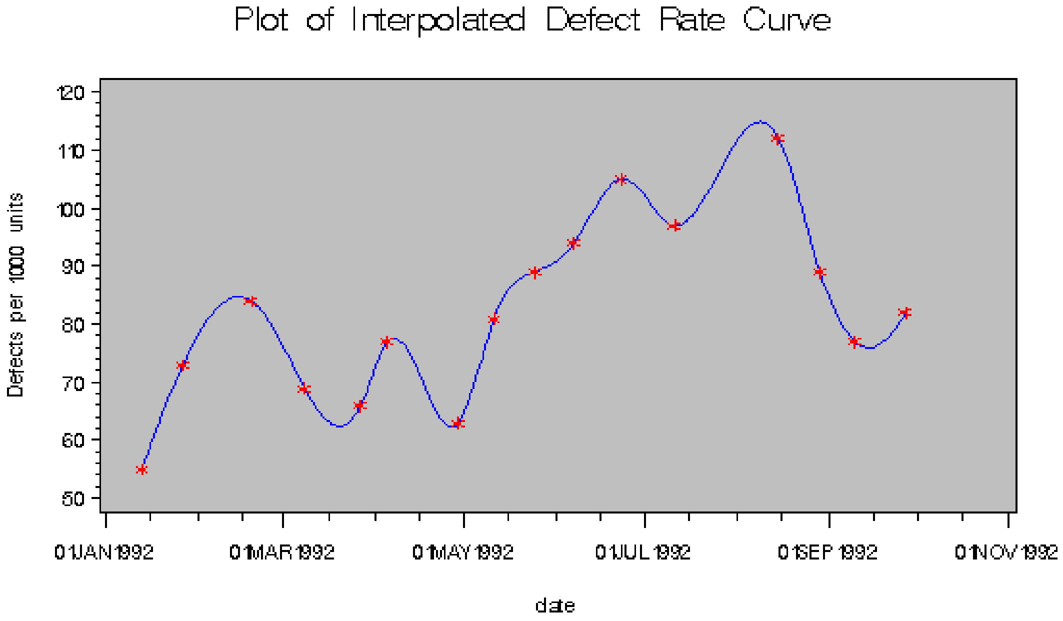
- Consists of sequences of values or events obtained over repeated measurement of time at equal time interval in most of the time.

- Used in application such as stock prediction, economic analysis etc.

- In general, there are two goals in time series analysis.

i. Modeling Time Series: Generating the time series with underlying mechanism.

ii. Forecasting Time Series: Predict the future values of the time series variables.



Major components for trend analysis in time series data

i. **Trend or Long term Movements**: Indicates the general direction in which a time series is moving over long or short interval of time through trend curve or trend line.

ii. **Cyclic Movement or Cyclic Variations**: Long term oscillations about a trend curve or line which may or may not be periodic.

iii. **Seasonal Movements or Variations**: These are systematic or calendar related. Eg. Sudden rise in sales of sweets in Tihar.

iv. **Irregular or Random Movements:** Series due to random or chance events. Eg. Price rise in crisis of supply.

Approaches for time series data analysis:

- Regression analysis is commonly used for find trend in time series data.
- Seasonal Index is used for analysis to adjust the reative values of a variable during the time series.
- Autocorrelation analysis is applied between i^{ith} element of the series and the (i-k)th element to detect seasonal patterns. Where K is referred to as the log.
- Calculating the moving average of order n is the common method for determining trend.

  Eg:

  Original Data: 3 7 2 0 4 5 9 7 2

  Moving average of order3: (3 + 7 + 2)/3 = 4, 3 2 3 6 7 6

  Weighted (1, 4, 1) average: ((1*3 +4*7 +1*2)/(1+4 +1))= 5.5, 2.5 1 3.5 5.5 8 6.5
- Free hand method is used to draw approximate curve or line to fit a set of data based on user’s judgment.
- Least square method is used to fit best curve.

C. Object/ Image/ Multimedia Mining:

- Multimedia database system stores and manages a large collection of multimedia data such as audio, video, images, graphics, speech, text etc.
- Image/multimedia mining deals with extraction of implicit knowledge, data relationship or other patterns not explicitly stored in images/multimedia
- The fundamental challenges in images mining is to determine the low-level pixel representation contained in an image or image sequence and cane be effectively and efficiently processed to identify high level spatial objects and relationships.
- Typical image/multimedia processing involves preprocessing, transformations and feature extraction mining, evaluation and interpretation of the knowledge.
- Different data mining techniques can be used such as association rules, clustering.